ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

PROVOZNĚ EKONOMICKÁ FAKULTA

Metodika návrhu multidimenzionální databáze v prostředí zemědělského podniku

disertační práce

Autor: Ing. Jan Tyrychtr

Školitel: doc. Ing. Zdeněk Havlíček, CSc.

Katedra informačních technologií

Praha 2012

OBSAH

SEZNAM VYOBRAZENÍ

SEZNAM TABULEK

Poděkování

Děkuji svým školitelům, panu doc. Ing. Zdeňku Havlíčkovi, CSc. za odborné vedení disertační práce a paní doc. PhDr. Ivaně Švarcové, CSc., In memoriam, za impuls k dosavadnímu výzkumu i cenné rady.
Poděkování patří také mým rodičům a přítelkyni Ing. Stanislavě Sýkorové za podporu v průběhu doktorského studia.

Úvod

V obecném pojetí představuje zemědělství formu podnikání, které je charakteristické určitými specifiky: hospodaření s půdou a živými tvory, biologický charakter produkce, podnikání závisející na přírodních podmínkách, nesoulad nákladů a výnosů, neelastický charakter poptávky po potravinách, socioekonomická struktura podniku a další aspekty typické pro agrární sektor. V užším pojetí lze zemědělství chápat jako systém skládající se z řady subsystémů (zemědělských podniků), ve kterých probíhají hmotně energetické a informační transformace. Právě systémové pojetí umožňuje identifikovat kvantifikovatelné vlastnosti a chování zemědělských podniků.

V současnosti mezi základní faktory konkurenceschopnosti zemědělského podniku patří efektivní vynakládání zdrojů. Coelli, et al., (2005) uvádí, že ekonomickou výkonnost lze měřit mírou efektivnosti a produktivity. Efektivnost (ekonomickou) lze dále členit: na technickou efektivnost, alokační efektivnost a efektivnost z rozsahu (Kumbhakar, et al., 2000). Studie (Matulová, 2011) prokazuje u sledovaných českých zemědělských podniků spíše konstantní výnosy z rozsahu (tedy blížící se optimálnímu rozsahu výroby). Naopak produkční faktor půda není efektivně využíván a technická efektivnost dosahuje v průměru 77%. Čechura (2010) odhaduje technickou efektivnost s ohledem na firemní heterogenitu na úroveň 90%. Výzkum provedený (Kroupová, 2010) odhaduje o 13,5 % nižší efektivnost ekologického zemědělství v komparaci s konvenčními farmami. Výzkum poukazuje na to, že v průměru se ekologické farmy pohybují na 55,1 % potenciální produkce a také, že 50 % zkoumaných ekologických subjektů dosahuje nižší než 50,1% míry technické efektivnosti.

Právě skutečnost, že zemědělské podniky nedosahují plné produkční síly a technické efektivnosti, je motivací zvoleného tématu disertační práce. Autor se domnívá, že informační technologie mohou managementu farmy pomoci interpretovat právě ta data, z kterých je možné získat relevantní poznatek o jejich ekonomické výkonosti. Takový to podnik musí mít identifikovanou produkční funkci, sbírat relevantní a objektivní data, která jsou vhodně strukturovaná a v neposlední řadě provádět analytické zpracování sumarizovaných a agregovaných dat. Právě pro tvorbu vhodných struktur bází dat a pro jejich analytické zpracování se autor v disertační práci snaží kriticky přejímat dosavadní

přístupy a vytvářet nové metody pro návrh analytických databází v zemědělském podniku.

Oblast zkoumání

Předmětem zkoumání disertační práce je návrh analytické báze dat v zemědělském podniku pro ekonometrické analýzy v rámci *Business intelligence*. Na tomto místě je důležité zmínit, že se autor explicitně nezabývá ekonometrií, tato oblast je učena ekonometrům, za které se autor rozhodně nepovažuje. Především je v práci věnována pozornost vrstvě pro ukládání dat s akcentací na analytické a plánovací potřeby uživatelů v zemědělském podniku. Do zkoumané oblasti patří nejen databázové komponenty, ale i komponenta pro on-line analytické zpracování dat (tzv. OLAP, zvýrazněné na obrázku č. 8, str. 41). Tento specifický předmět výzkumu velmi silně koreluje s oborem *informační management*, který lze podle normativního výkladu chápat jako interdisciplinární vědní obor zaměřující se na plánování, řízení a využívání zdrojů informací v rámci instituce (ČSN ISO 5127-2003). Důležitou komponentou informačního managementu je *datový management*. Jeho účelem je dosažení dostupnosti, kvality a bezpečnosti dat pro potřeby všech zainteresovaných stran (Earley, 2010). A právě management Business intelligence je jednou z mnoha funkcí datového managementu.

Autor disertační práce se dále podrobně nezabývá analýzou pojmů informačního a datového managementu a jejich komponent a funkcí. Přesto je podnětné upozornit na vzájemnou souvislost a důležitost těchto vědních disciplín, které představují vymezenou hraniční oblast zkoumání disertační práce.

Struktura disertační práce

Prvá kapitola představuje literární rešerši disertační práce. Její *prvá podkapitola* je věnována vymezení pojmů databáze a databázový systém (v současnosti stále zaměňované pojmy řady autorů vědeckých publikacích). Naznačen je jejich historický vývoj. *Druhá podkapitola* je věnována především přesnému vymezení relační databáze a jeho modelu dat. Snahou autora je použití přesných matematických formulací, které na rozdíl od neformálního výkladu přesně vymezují příslušné pojmy. *Třetí podkapitola* je věnována přesnému vymezení multidimenzionálnímu modelu dat, který je na rozdíl od pojmu multidimenzionální databáze (používán neformálně ve dvou různých významech) vymezen jednoznačně prostřednictvím operátorů datové kostky. V rámci

kritické literární rešerše je ve *čtvrté podkapitole* věnována částečná pozornost oblasti nazývající se Business intelligence. Právě prvá a druhá podkapitola představují důležitý znalostní vstup této čtvrté podkapitoly, ve které jsou popsány základní databázové komponenty Business intelligence. *Pátá a šestá podkapitola* je věnována dosavadním přístupům v návrhu provozních (relačních) a analytických (multidimenzionálních) databází. V poslední sedmé podkapitole jsou vymezeny modely formulace výrobního procesu, které jsou potencionálně použitelné jako vstup pro návrh analytické databáze. Zároveň jsou představeny dosavadní přístupy návrhu analytických databází v kontextu zemědělství.

Druhá kapitola shrnuje výsledky literární rešerše. Identifikována jsou tak zvaná „bílá místa“, která je účelné dále v disertační práci zkoumat. Jsou zde vybrány příslušné metody a modely spolu s odůvodněním jejich výběru pro další výzkum.

Třetí kapitola vymezuje cíl disertační práce spolu s dílčími cíli a pomocnými hypotézami.

Čtvrtá kapitola vymezuje prostředí metodiky, použité vědecké metody a objasňuje postup řešení vymezených cílů disertační práce.

Pátá kapitola tvoří samotný obsah vědeckého sdělení disertační práce. V této kapitole autor sděluje, co vyřešil, jakým způsobem a výsledky komentuje. Důležitým výstupem páté kapitoly je nová metodika návrhu analytické báze dat pro podporu ekonometrických analýz.

V **šesté a sedmé kapitole** autor syntetizuje nejdůležitější poznatky z předchozích kapitol. V *diskuzi* autor interpretuje dosažené výsledky a připisuje jim význam pro zkoumaný problém. Uvádí možnosti aplikace výsledků práce, upozorňuje na možné problémy a uvádí další směry svého výzkumu. *Závěrem* shrnuje postup řešení, verifikaci hypotéz a dosažené výsledky z teoretického i praktického hlediska.

1. Literární rešerše

Hlavním cílem literární rešerše je vytvoření kritického přehledu současného stavu poznání v oblasti návrhu multidimenzionálních modelů dat v kontextu zemědělského podniku. Dílčími cíli literární rešerše je:

- Analyzovat hlavní výzkumné proudy databázových technologií.
- Vymezit a kriticky zhodnotit terminologii Business intelligence a databázových komponent.
- Analyzovat procesy návrhu relační a multidimenzionální databáze.
- Formulovat výrobní proces v zemědělství.
- Analyzovat metody modelování multidimenzionálních dat v zemědělském podniku.
- Na základě syntézy formulovat cíle a hypotézy disertační práce.

1.1. Databáze

Internetové aplikace, informační systémy nebo aplikační software jsou v současnosti velmi často používány s databázovým systémem. Databázový systém a jeho struktura báze dat jsou důležitým pilířem při tvorbě software. Toto tvrzení podporuje řada autorů, například Post (2001) uvádí, že databáze jsou důležité pro většinu organizací, protože jsou základem informačních systémů.

V intuitivním pojetí se *databáze* [database] vymezuje jako „místo“ kam jsou v tištěné nebo elektronické podobě ukládána data. Takové pojetí je však velmi obecné a nepřesné. Tento základní pojem v oblasti databázové technologie je podle České terminologické databáze knihovnictví a informační vědy (dále jako TDKIV) vykládán jako „systém sloužící k modelování objektů a vztahů reálného světa (včetně abstraktních nebo fiktivních) prostřednictvím digitálních dat uspořádaných tak, aby se s nimi dalo efektivně manipulovat, tj. rychle vyhledat, načíst do paměti a provádět s nimi potřebné operace – zobrazení, přidání nových nebo aktualizace stávajících údajů, matematické výpočty, uspořádání do pohledů a sestav apod.“ (Kučerová, 2003). I takové vymezení není podle autora práce plně zdařilé. Nerozlišuje mezi pojmy databáze a databázový systém. Předpokládá, že pojmy jsou ekvivalentní.

Z etymologického výkladu (Online Etymology Dictionary, 2010) je databáze složena z pojmů data a základ (anglicky base), tudíž považovat databázi za systém není vhodné. Singh (2009) vymezuje databázi jako množinu logicky souvisejících dat uložených společně, která je navržena tak, aby splňovala informační potřeby organizace. Podle normy ČSN ISO 5127-2003 je databáze vymezena jako soubor souvisejících dat postačujících pro daný účel nebo pro daný systém zpracování dat. Tato rešerše se bude přidržovat právě vymezení pojmu databáze podle normy ČSN ISO 5127-2003.

1.1.1. Systém řízení báze dat

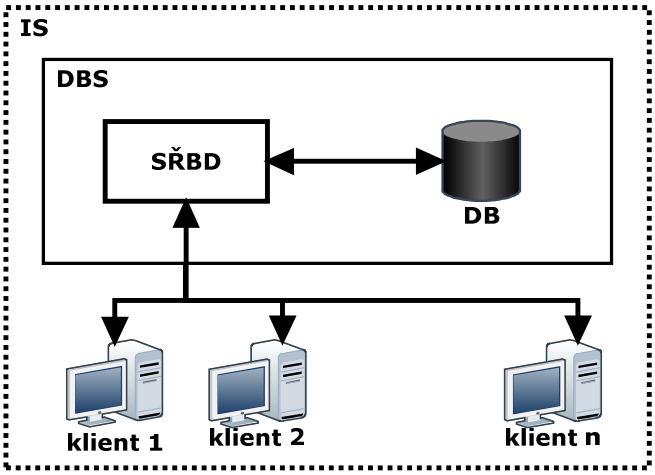
Podle normy ČSN ISO/IEC 2382-17 představuje *systém řízení báze dat* (SŘBD) [database management system] skupinu programů fungující jako rozhraní mezi daty v databázi a uživatelem, případně aplikačním programem. Helland (2009) vymezuje SŘBD jako softwarově založený systém poskytující aplikacím přístup k datům způsobem umožňujícím řízení a kontrolování. Základní účel SŘBD je definice a konstrukce databáze a manipulace s ní. Základní složky tvoří program pro definování dat, umožňující vytváření a změny datových struktur; program pro manipulaci s daty, umožňující vkládání a aktualizaci dat, vyhledávání, výběr a prezentaci dat a tvorbu formulářů a sestav; program pro řízení přístupu uživatelů k datům. Klíčovou vlastností SŘBD je vytvoření vysoké úrovně abstrakce. Ta podle Hellanda (2009) umožňuje řadu důležitých operací:

- nezávislý vývoj;
- sdílení stejných dat více aplikacemi;
- ad hoc přístup k datům;
- definici schématu;
- manipulaci s daty;
- transakce;
- zpracování dotazů;
- přístupové metody;
- souběžné řízení a obnovení dat.

Výčet těchto operací není úplný a ani závazný, databázové systémy se odlišují právě implementací SŘBD. Proto je již v současnosti možné některé operace považovat za samozřejmé, jako výstup dat na obrazovku nebo tiskárnu apod. a naopak některé operace, jako triggery, procedurální podpora atd. považovat za „nadstandardní“.

1.1.2. Databázový systém

Singh (2009) *databázový systém* [database system] nazývá SŘBD a vymezuje jej jako obecný softwarový systém pro manipulaci s databází. Stejný přístup k pojmu databázový systém má celá řada autorů (například Silberschatz, Korth a další). TDKIV označuje databázový systém ekvivalentní s pojmem databáze. Ovšem vhodnější vymezení je následující: databázový systém je pojem, který se obvykle užívá k zapouzdření konstrukce datového modelu, SŘBD a databáze (Beynon-Davies, 2004). „Princip databázového systému lze charakterizovat rovnicí: DBS= DB + SŘBD, kterou lze interpretovat následovně: data jsou organizována v databázi (DB) a jsou řízena systémem řízení báze dat (SŘBD)“ (Vostrovský, 2004). Na obrázku č. 1 jsou znázorněny složky databázového systému spolu s klienty (uživateli), kteří tak tvoří informační systém.



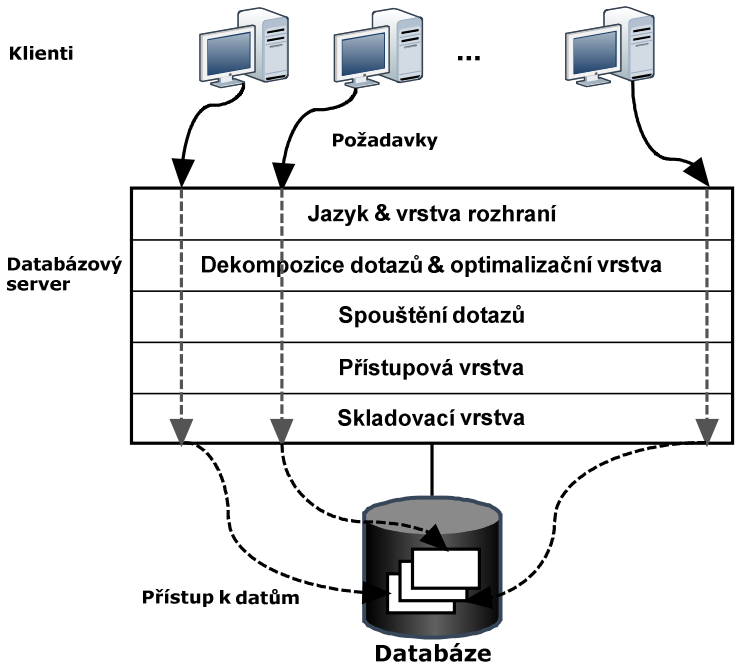
Obrázek č. 1 - Elementy databázového systému (podle Vostrovský, 2004).

Databázový systém se od souborového systému liší vysokou úrovní abstrakce. Mezi hlavní rozdíly patří (Silberschatz, et al., 2001):

- Oba systémy obsahují kolekci dat a soubor programů, které přistupují k datům. SŘBD koordinuje jak fyzický, tak logický přístup k datům, zatímco zpracování dat souborového systému je realizováno pouze prostřednictvím fyzického přístupu.
- Databázový systém snižuje duplicitu dat, tak aby fyzický údaj byl k dispozici všem oprávněným programům přistupujícím k němu. Zpracování dat u souborového systému nemusí být čitelné jiným programem.

- Databázový systém je navržen tak, aby umožnil flexibilní přístup k datům (tj. tvorba dotazů), zatímco u zpracování dat souborového systému je nutné mít předem vytvořen přístup k datům (tj. prostřednictvím sestavených programů).
- Databázový systém je navržen tak, aby umožňoval přístup více uživatelům ke stejným datům ve stejnou dobu.

Obrázek č. 2 zobrazuje vrstvy databázového systému a jejich funkce. Taková architektura je společná v podstatě pro všechny databázové servery v té či oné podobě. Když od klienta přijde požadavek na server, server spustí kód, který převádí žádosti do jedné nebo více operací v každé ze spodních vrstev. Nakonec dochází k přístupu na disk (pokud přístupu nezabraňuje caching), (Vossen, 2009).



Obrázek č. 2 - Funkce vrstev databázového systému (podle Vossen, 2009).

1.1.3. Historický vývoj

Tato kapitola nastiňuje historický vývoj databázových systémů s cílem krátce shrnout dosavadní výzkum. Hierarchické, síťové, relační, sémantické, objektově orientované a objektově-relační systémy řízení jsou popsány.

Děrné štítky a papírové pásky byly historicky prvním způsobem počítačového ukládání dat. Jejich potenciál však nikdy nebyl dostatečně velký, aby je bylo možné použít pro zpracování dat. Situace se změnila s příchodem magnetického záznamového média.

Vynález magnetického záznamového média, jako jsou magnetické pásky a magnetické disky, umožnil trvalé uložení „velkého“ množství dat, které již umožňovalo počítačové zpracování. Logika každého programu byla stejná, vše co bylo možné pozměnit, byly vstupní a výstupní operace. Záznamy se skládaly z řady souborů, ve kterých byly jednotlivé informace. Záznamy v souboru byly seřazeny, aby bylo možné rychlé zpracování a vyhledávání. Taková sémantika již byla možná k implementaci systémů pro zpracování dat. Ovšem nutné bylo vytvořit poměrně komplexní programy, které byly náročné na údržbu. Nevýhodou těchto souborových systémů je, že nepodporují různé způsoby zpracování. (Jackson, 1999)

1.1.3.1. Hierarchický model dat

Člověk používá hierarchické rozložení po tisíce let pro klasifikaci objektů a artefaktů, organizační struktury, plánování činnosti, popis fyzického a koncepčního uspořádání, nebo organizaci znalostí (Kamfonas, 1992). Tento princip vztahů struktury dat v reálném světě, bylo velmi nepraktické realizovat prostřednictvím papírové pásky nebo děrného štítku.

Se zavedením adresovatelných magnetických disků bylo možné vložit do záznamů ukazatele. Tyto ukazatele představovali vztahy mezi daty. Kolem roku 1960 byla magnetická páska stále hlavní médium pro ukládání dat (Jackson, 1999). Ovšem nevýhodou byla neexistence flexibility magnetického disku, která je nutná pro datový model podporující postupný přístup k datům. Tento požadavek vedl k vývoji hierarchického modelu dat realizovaného společností IBM pod označením IMS (IMS/360, 1971).

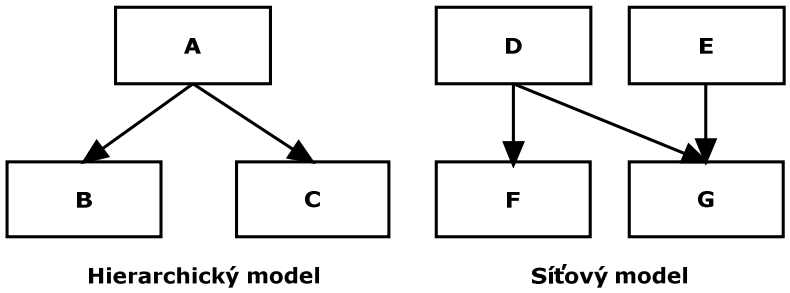
Jakékoliv hierarchické záznamy mohly být reprezentovány jako posloupnost a tyto sekvence mohly být uloženy na magnetickém pásku. První datový model vznikl pouze s ohledem na fyzické paměti, s kterými musel pracovat. Původní použití určené pro IMS bylo „účtování zpracovaného materiálu“ a vybraný datový model byl pro tento účel ideální. Tento typ aplikace se zabýval fakty, jako „část A je vyrobena z části B a C, část B je postavena z části D, E a F“. Díky adresovatelnosti magnetických disků byla do IMS později přidána možnost přímého přístupu (prostřednictvím hašovací funkce a B-stromu) do kořene záznamů tak, aby byla umožněna podpora transakčního zpracování (Hainaut, 2009(a)).

Přesto IMS nezachytil sémantiku uložených dat kromě schopnosti reprezentovat vztahy mezi záznamy. Jednotlivá pole nebyla identifikována v systému pro správu databází, záznam byl definován jednoduše jako počet bytů, do kterých mohou být data umístěna. V důsledku toho, nebyly podporovány ad hoc dotazy. Zpracování sémantiky bylo zcela vložené do programů vytvořených aplikací a bylo třeba psát programy umožňující přístup k databázi (Jackson, 1999).

IMS patří mezi starší technologie, ale existuje stále mnoho velkých firemních databází, které IMS používají. Používá se především pro stabilní, pomalu se vyvíjející aplikace a to zejména v bankovních společnostech (Hainaut, 2009(a)). Nedostatek pružnosti a složitost hierarchického modelu vede organizace v současnosti k implementaci datových skladů.

1.1.3.2. Síťový model dat

Síťový model odstraňuje omezení ve vztazích, které nemohou být zastoupeny v hierarchickém modelu. V čistě hierarchických modelech je možné realizovat vztah pouze jeden k mnoha (jeden záznam typu A je spojen s mnoha záznamy typu B). Důvodem je, že v hierarchii může uzel mít pouze jeden uzel rodičovský. V síťovém modelu může uzel souviset s více než jedním dalším uzlem. (Jackson, 1999) Obrázek č. 3 znázorňuje logická schémata vztahů v hierarchickém a síťovém modelu.



Obrázek č. 3 - Komparace hierarchického a síťového modelu dat.

Hierarchické systémy doporučované výborem CODASYL jsou popsány v literatuře CODASYL Data Base Task Group Report (1971). V roce 1960 se stal COBOL převládající jazyk pro zpracování dat a CODASYL byl orgánem, který odpovídal za rozvoj standardů COBOL. Databázové systémy byly předurčeny stát se významným hráčem na zpracování dat a bylo považováno za vhodné, aby byly přezkoumány možnosti, jak by mohly být integrovány databázové technologie s jazykem COBOL.

V roce 1965 na konferenci CODASYL byl vytvořen výbor Database Task Group (DBTG), který měl za úkol vytvořit koncepci databázových systémů. (Jackson, 1999)

Hlavní inovace ve zprávě DBTG bylo oddělení různých problémů správy dat. Na základě toho byl definován *jazyk pro definici dat* (DDL) umožňující návrháři databáze kompletně definovat databázi bez ohledu na aplikace, které mohou databázi využívat. Syntaxe byla podobná jazyku COBOL. *Jazyk pro manipulaci s daty* (DML) dovolující programátorovi vytvořit vztahy v databázi, aniž by si byl vědom toho, že záznamy byly připojeny přes adresu ukazatele. CODASYL také představil omezenou formu fyzické datové nezávislosti. Nebylo nutné, aby záznamy byly uloženy tak, jak byly popsány ve schématu. (Jackson, 1999) Definování rozhraní pro aplikační programy se nazývá subschéma DDL (Hainaut, 2009(b)).

CODASYL nepochybně ovlivnil závěry ANSI / SPARC studijní skupiny, která se sešla k diskusi k databázové technologii. Výsledkem byl framework architektury DBS, skládající se ze tří schémat: vnitřní, vnější a konceptuální. (Tsichritizis, et al., 1978)

1.1.3.3. Relační model dat

Koncepce síťového a relačního modelu dat byla vyvinuta ve zhruba stejnou dobu. Síťový model byl však mnohem rychleji obsažen v komerčních databázových systémech, než model relační. Důvodem vzniku relačního modelu byla řada otázek, které si položili v IBM. Problémy, které bylo nutné řešit, byly následující (Jackson, 1999):

- je potřeba zvýšit nezávislost dat v systémech pro správu databází;
- je potřeba matematického přístupu k ukládání a načítání dat;
- je potřeba podporovat ad hoc zpracování dotazu.

První bod je cílem každého DBS, tedy poskytnutí nezávislosti dat a skrytí pro programátora aplikací skladovací a vyhledávací operace. Síťové a hierarchické systémy netrpí datovou závislostí, ale závislostí „přístupové cesty“. Přístup k datům v IMS vyžaduje, aby programátor vstupoval do databáze přes záznam v horní části hierarchie. Jestliže programátor neví, kterou z nejvyšší úrovně záznamu zvolit, pak musí celou databázi prohledávat. Možností bylo identifikovat vstupní body do databáze pomocí klíče haše nebo „zvláštních vztahů“. Taková databáze podporovala pouze několik přístupových cest. Návrhář databáze se u takového přístupu musel snažit předvídat

všechny možné aplikace, které mohly být s databází provozovány a vytvořit vhodné přístupové cesty. Pokud se následně objevila aplikace, která nebyla podporována, pak musela být databáze přepracována. Relační model se pokusil o zlepšení této situace tím, že nemá žádné předdefinované přístupové cesty. (Jackson, 1999)

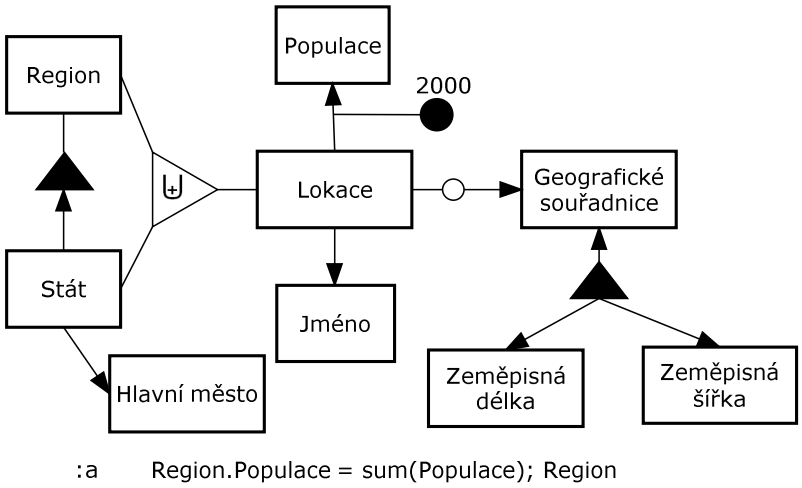
Důležitým milníkem se stal dokument (Codd, 1970), který zavádí relační model jako model založený na *n*-árních vztazích. Potřebu matematického základu pro DBS si lze vysvětlit na principu získávání dat z databáze ještě před vznikem relačního systému. Principy byly podobné těm, které známe v každém programovacím prostředí. V první fázi byla sepsána specifikace, v další fázi byl vytvořen návrh a poté byla vytvořena konstrukce programu. Tato sekvence je plná příležitostí pro zavedení chyby. U relačního modelu je možné vytvořit dotaz vyjádřený v predikátu, podle kterého systém automaticky načítá data. Problémem zajištění toho, aby dotaz odpovídal reálnému světu, zůstává, ale problém transformace dotazu do spustitelného kódu odpadá. (Jackson, 1999)

První rozsáhlou implementací relačního modelu se stal výzkumný projekt IBM pod názvem System R, který byl dokončen v roce 1976 (Astrahan, et al., 1976). System R je experimentální databázový systém, který byl realizován, aby prokázal možnost realizace relačního datového modelu (Chamberlin, et al., 1981). Zároveň se podobný projekt vyvinul na University of California v Berkeley pod názvem INGRES (Stonebraker, et al., 1976). Mnoho myšlenek v System R bylo včleněno do prvního komerčního relačního systému Oracle, který byl vydán v roce 1979 (Jackson, 1999). Trvalým odkazem Systemu R se stal jazyk SQL, který prostřednictvím relačního kalkulu umožňuje manipulovat s databází (Chamberlin, et al., 1974(b)).

1.1.3.4. Sémantický model dat

Logická datová struktura DBS, ať již hierarchických, síťových nebo relačních, nemůže zcela uspokojit požadavky na definici dat. *Sémantický model dat* reprezentuje data v pojmenovaných množinách objektů, hodnot, vztahů a omezeními mezi těmito množinami objektů, hodnot a vztahů (Embley, 2009(b)). Nejjednodušší z těchto modelů je Entity Relationship (ER) model (Chen, 1976). Tento model nabízí více sémantiky, než relační model. ER model byl později rozšířen o vyjádření sémantiky dat, kterou nelze přímo reprezentovat pomocí relační databáze (Teorey, et al., 1986). Vhodným

příkladem sémantických modelů je sémantický databázový model (Hammer, et al., 1981). Na obrázku č. 4 je znázorněn obecný sémantický model dat.



Obrázek č. 4 - Obecný sémantický model podle Embley (2009(b)).

1.1.3.5. Objektový model dat

Objektový model dat poskytuje podporu objektům modelovaných v databázových aplikacích. Objekt je jedním z nejzákladnějších pojmů objektového modelu dat, kde objekt představuje entitu zájmu v konkrétní aplikaci. Objekt má stav popisující specifické strukturální vlastnosti objektu. Objekt má také chování, definuje metody, které se používají k manipulaci s objekty. Implementace metody lze změnit bez vlivu na rozhraní třídy a způsob, jakým je rozhraní používané v kódu aplikace. (Urban, et al., 2009) Rozdíl mezi klasickým a objektovým přístupem tvorby modelu dat lze vysvětlit následovně: „Při tvorbě datového modelu klasickým způsobem se snažíme prvky reálného světa zobrazit do předem připravených struktur pevně daného druhu. U objektů je tomu obráceně; pro prvky reálného světa si vytváříme nové objekty, které se jim podobají.“ (Merunka, 2006)

První objektové systémy se začaly objevovat v první polovině osmdesátých let (Copeland, et al., 1984). Manifest k tomu, co by měla objektový databázový systém (ODBS) podporovat, byl předložen v roce 1989 (Atkinson, et al., 1989). Práce na normalizacích pro objektové databáze začala v roce 1991 (Kaufmann, M., 1993). Normalizačním orgánem k objektové databázi je znám pod zkratkou ODMG (Object Data Management Group). Současný standard objektových databází je ve verzi 3.0 (Cattell, et al., 2000).

Typickými příklady využití ODBS jsou geografické informační systémy (GIS), počítačové navrhování (CAD) a počítačová podpora výroby (CAM). Některé datové struktury by mohly být mapovány do relační databáze, nicméně většina relačních systémů dostupných v roce 1980 podporovala velmi málo datových typů. Schopnost vytvořit potřebné datové typy je právě vlastnost objektově orientovaných programovacích jazyků. (Jackson, 1999)

1.1.3.6. Objektově-relační model dat

Zveřejnění manifestu ODBS (Atkinson, et al., 1989) vedlo k téměř okamžité odezvě od těch, které zajímal další vývoj relačních DBS (Stonebraker, et al., 1976). Zastánci relační technologie, tvrdí, že ODBS byly vyvinuty především k podpoře programovacího jazyku a jeho interakcí s daty. Objektově-relační tábor identifikuje hlavní slabost tradičních relačních systémů v neschopnosti podporovat komplexní data. (Jackson, 1999) Rowe a Stonebraker vyvinuli jako první *objektově-relační databázový systém* (ORDBS), který přidává podporu zpracování komplexních dat na již existující SQL. Stonebraker (1996) uvádí, že je nutné přidat podporu:

- pro rozšíření základních datových typů v rámci SQL;
- pro složité objekty v kontextu SQL;
- dědění v kontextu SQL;
- pro tvorbu systémových pravidel.

Základními datovými typy v tradičních relačních systémech jsou obvykle znak, řetězec, celé číslo, číslo s plovoucí desetinnou čárkou, datum a čas. ORDBS umožňují návrháři databáze definovat nové základní typy. Aby bylo možné plně využít rozšiřitelnost datových typů, musí systém rovněž umožnit tvorbu uživatelem definovaných funkcí a operátorů. (Jackson, 1999)

V relačním modelu jsou atributy tradičně atomové. ORDBS podporují komplexní objekty, které tvoří shluky hodnot jiných datových typů. V ORDBS existují mechanismy pro definici složitých objektů. Dědičnost je jedním z klíčových pojmů objektového paradigma. Dědičnost byla zavedena do objektově-relačních systémů, aby mohl být opět použit definovaný komplexní objekt a uživatelem definované funkce. Je také možné definovat podtyp stávajícího typu. Nový typ zdědí data a funkce od jeho nadtypu. (Jackson, 1999) Větší a složitější aplikace vyžadují další integritní omezení.

To je řešeno prostřednictvím pravidel, která jsou spojena s událostí. Když dojde k události, operace spojené s pravidlem se provedou. Pravidla se používají k zajištění konzistentního stavu databáze.

V současnosti existuje na trhu několik ODBS a zároveň mnoho relačních databázových produktů již zajišťuje nějakou formu podpory ORDB (Urban, et al., 2009). Autoři (Feuerlicht, et al., 2009) upozorňují, že je zapotřebí nových návrhových metod pro podporu objektově-relačních funkcí, které jsou často implementovány pomocí čistě relačního návrhu.

1.1.3.7. Další DBS a jejich vývoj

Souběžně s vývojem výše popsaných modelů dat probíhal výzkum dalších databázových systémů. Níže uvedené databáze se zaměřily na aplikace, které byly do značné míry ignorovány v komerčním prostředí. Tato část obsahuje stručný souhrn těchto oblastí.

Deduktivní DBS [Deductive DBS]

Důležitým omezením relačního kalkulu / algebry je, že nedokáže vyjádřit dotazy týkající se „cesty“ skrz instance, jako například nad tranzitivním uzávěrem binární relace (Karvounarakis, 2009). *Deduktivní DBS* poskytují mechanismy, kterými lze pomocí pravidel odvodit z dat uložených v databázi nové skutečnosti (Jackson, 1999). Jazyk používaný ke specifikaci faktů, pravidel a dotazů v deduktivních DBS se nazývá Datalog. Jazyk Datalog byl pojmenován Maierem v roce 1980 (Abiteboul, et al., 1995).

Aktivní DBS [Active DBS]

Aktivní DBS podporuje reaktivní chování na základě pravidel ECA (Event Condition Action). Aktivní databáze může automaticky reagovat včas a efektivně na události, jako jsou databázové přechody, čas události a externí signály. (Berndtsson, et al., 2009) Pojem aktivní databáze byl poprvé použit v roce 1980 (Geppert, et al., 1997).

Temporální DBS [Temporal DBS]

Temporální DBS jsou předmětem intenzivního studia od začátku roku 1980 (Jensen, et al., 2009). Temporální DBS se zabývá situacemi, kde fakta jsou spojeny s časem (Jackson, 1999). Rozlišuje se složka vyjadřující období, ve kterém byl fakt (v modelovaném světě) pravdivý (tzv. Valid time) a složka vyjadřující období, po které je fakt uložen v databázi (tzv. Transaction time).

Distribuované DBS [Distributed DBS]

Distribuované DBS se začali propagovat se System R a Ingres na začátku roku 1980 (Risch, 2009(a)). Distribuované DBS mají SŘBD rozděleny do několika uzlů (počítačů) v počítačové síti. V centrálním SŘBD jsou data řízena v jednom uzlu, zatímco v distribuovaném SŘBD jsou data řízena několika uzly. (Risch, 2009(a)) Výzkum distribuovaných DBS se snaží vyřešit problémy, při kterých jsou globální data zaznamenávána v mnoha zeměpisně oddělených lokalitách (Sheth, et al., 1990).

Multimediální DBS [Multimedia DBS]

Multimediální DBS obsahují a umožňují klíčové operace s multimediálními daty. První multimediální DBS, které se začaly objevovat na konci roku 1980, umožňovaly pouze zpracování obrazu (tzv. image database). Relační model dat se stal nejrozšířenějším modelem pro řešení strukturovaných dat a byl používán pro ukládání obrázků prostřednictvím datového typu BLOB. Komerční řešení bylo použito v tradičních databázových systémech v polovině roku 1990. IBM použila Qbic technologii v databázovém systému DB2 a Oracle, Sybase a Illustra použili technologii vyvinutou společností Virage. S ohledem na současný trend, je pravděpodobné, že většina databází se pomalu stane multimediálními. (Ramesh, 2009)

Prostorové DBS [Spatial DBS]

Prostorová databáze je založena na datovém modelu definující vlastnosti a operace statických objektů v prostoru (Schneider, 2009(a)). Zájem o ukládání geometrických dat v databázích začal již v roce 1970. Vzhledem k rostoucímu úspěchu relačních databází byly první přístupy k ukládání prostorových objektů řešeny uložením v tabulkách. (Jackson, 1999) Tento přístup nebyl vhodný pro ukládání prostorových dat. To vedlo k návrhu prostorových datových typů, které jsou reprezentovány jako abstraktní datové typy a mohou být použity stejným způsobem jako standardní datové typy. (Schneider, 2009(b))

Multidimenzionální DBS [Multidimensional DBS]

Multidimenzionální databáze nemají původ v databázové technologii, ale vychází z vícerozměrné maticové algebry, která byla použita pro (ruční) analýzy dat od konce devatenáctého století. V roce 1960, dvě společnosti IRI a Comshare nezávisle na sobě začaly vývoj systémů, který později představoval začátek multidimenzionálních DBS. (Pedersen, 2009(c)) Oblast řízení statistických dat se od roku 1980 nejprve zaměřovala

na modelování a řízení statistických dat v kontextu společenských věd, jako jsou údaje ze sčítání lidu. Mnohé důležité pojmy vícerozměrného modelování, jako například sumarizabilita (více se tímto pojmem zabývá Lenz, et al. (1997)), mají své kořeny v této oblasti. Přehled této problematiky uvádí Shoshani (1997).

V roce 1990 byl zaveden operátor datové kostky (Gray, et al., 1997), který vyvolal značný zájem ve výzkumu modelování multidimenzionálních databází. V roce 1998 Microsoft představil první multidimenzionální systém (MS OLAP Server) zaměřený na prodej širšímu spektru zákazníků. To vedlo k současné situaci, kdy multidimenzionální systémy se stále častěji stávají produkty, které jsou dodávány spolu s předními relačními DBS. Podrobnější pokrytí historie multidimenzionální databáze uvádí Thomsen (1997). Průzkumy vícerozměrných modelů dat lze nalézt v literatuře (Pedersen, et al., 2001), (Vassiliadis, et al., 1999). Multidimenzionálnímu pohledu na data je v této rešeršní práci podrobněji věnováno v kapitole č. 1.3.

NoSQL databáze

V souvislosti s databázemi určených pro webové prostředí a cloud computing se začátkem roku 2009 začaly objevovat tak zvané NoSQL databáze. Účelem NoSQL databází je dosažení horizontální škálovatelnosti databázového zpracování v dynamickém prostředí distribuovaných databází, které obsahují semistrukturovaná data bez pevného databázového schématu. Termín NoSQL databáze byl zvolen pro volně specifikovanou třídu nerelačních datových úložišť. (Pokorný, 2012)

Pokorný (2011) poukazuje na fakt, že vzhledem k horizontálnímu škálování je slabší podpora vlastností ACID [atomicity, consistency, isolation, durability]. Přesto existuje v současnosti široké využití NoSQL databází především ve specializovaných projektech zejména s nestrukturovanými daty a vysokými požadavky na škálování. Do NoSQL databází se zahrnují i XML databáze, grafové databáze či databáze dokumentů nebo objektové databáze. (Pokorný, 2012)

1.2. Relační databáze

Jednoduše lze relační databázi označit za soubor tabulek (Churcher, 2008). Ovšem přesněji *relační databáze* [relational database] představuje databázi založenou na relačním modelu dat.

1.2.1. Relační model dat

„Pojem *relace* vznikl ve snaze formalizovat to, co chápeme v běžné řeči jako „vztah“ mezi objekty. (Vaníček, 2004)“. Relační model dat [relational data model] navrhl Codd. Termín relace Codd používá v matematickém smyslu. Tedy máme-li množiny S_1, $S_2, \dots, S_n$ (nemusí být nutně odlišné), *R* představuje relaci na *n* množin, kde každá *n*-tice má první prvek z S_1, druhý prvek z S_2, a tak dále. (Codd, 1970) Matematici mají pro množinu všech *n*-tic S_1, $S_2, \dots, S_n$, kde $s_j \in S_j$, název kartézský součin množin S_1, $S_2, \dots, S_n$ a označení $S_1 \times S_2 \times \dots \times S_n$ (Vaníček, 2004). Tedy *R* je podmnožinou kartézského součinu. Pokud platí, že $s_j \in S_j$, v takovém případě říkáme, že *R* je relací s doménami S_j. Pro každou složku *n*-tice, je možné definovat jeho doménu a jeho jedinečný název, který se nazývá atribut.

Definice: Relací s doménami S_1, $S_2, \dots, S_n$ nazýváme libovolnou podmnožinu kartézského součinu $S_1 \times S_2 \times \dots \times S_n$; *n*-arní relací na *S* nazýváme libovolnou podmnožinu kartézské mocniny S^n, definované indukcí vztahy: $S^1 = S$; $S^{(n+1)} = S^n \times S$ pro *n* = 1, 2, … . (Vaníček, 2004)

Pole, které představuje *n*-ární vztah *R* má následující vlastnosti (Codd, 1970):

- Každý řádek představuje *n*-tici z *R*.
- Pořadí řádků je nevýznamné.
- Všechny řádky jsou odlišné.
- Uspořádání sloupců je významné a odpovídá uspořádání domén S_1, $S_2, \dots, S_n$, na které je definována *R* (platí pro daný vztah).
- Význam jednotlivých sloupců je částečně dán názvem příslušné domény.

Relační model popisuje data jako pojmenované vztahy označených hodnot (Embley, 2009(a)). Například, zákaznické ID se vztahuje se zákazníkovým jménem a adresou. Relační model dat je možné intuitivně zobrazit na příkladu relace *Zákazník: {<(ZákazníkID, 100), (Jméno, Jan), (Příjmení, Novák), (Adresa, Kamýcká 123)>, <(ZákazníkID, 101), (Jméno, Alena), (Příjmení, Nováková), (Adresa, Kamýcká 123)>}*. V tomto příkladu je pojmenován vztah: *Zákazník*; označeny hodnoty dvojic: (*ZákazníkID, 100)*; a n-tice: *<(ZákazníkID, 100), (Jméno, Jan), (Příjmení, Novák), (Adresa, Kamýcká 123)>*.

V RMD je relace reprezentovaná tabulkou. Tabulkové znázornění relace má tyto vlastnosti (Vostrovský, 2004):

- Hodnoty v tabulkách jsou atomické.
- Hodnoty jsou skalární.
- Hodnoty v tabulkách existují jako prvky jednotlivých domén. Všechny prvky dané domény musejí být mezi sebou porovnatelné a musejí náležet jednomu datovému typu.
- Pro práci s tabulkami se používá operací výrokové logiky.
- V každé tabulce slouží hodnoty v jedné nebo více doménách k jednoznačné identifikaci řádků mezi sebou (tzv. primární klíče).
- V některých tabulkách mají hodnoty v jedné nebo více doménách vztah k hodnotám v jiných tabulkách (tzv. cizí klíč).
- V tabulkách lze definovat podmnožiny řádků anebo podmnožiny sloupců.
- Více tabulek lze kombinovat mezi sebou pomocí operací relační algebry.

1.2.2. Pravidla E. F. Codda

Codd publikoval pravidla pro relační model dat (Codd, 1985(a), 1985(b)). Ve většině publikací se nepřesně uvádí 12 pravidel. Tato záměna byla způsobena tím, že Codd číslovал pravidla od nuly. Databázový systém, označený jako plně relační, musí splňovat **13 pravidel**, která uvádí Whitehorn, et al. (2007).

1.2.3. Relační algebra

V roce 1970 Codd navrhl použití relační algebry jako základu pro dotazovací jazyk. Od té doby relační algebra nachází výrazné uplatnění při vývoji dotazovacích jazyků, nejpopulárnější z nich je SQL. (Mitra, 2009)

Relační algebra [relational algebra] je procedurální dotazovací jazyk se základními operacemi (Silberschatz, et al., 1997). Tyto operace zahrnují množinové operace: sjednocení, průnik, rozdíl a kartézský součin. Kromě toho se relační algebra skládá z dalších odvozených nebo pomocných operací typických pro relační databáze, jako jsou selekce, projekce, přejmenování, spojení a rozdělení. Operace relační algebry jsou tedy operace teorie množin s dalšími operátory, které berou v úvahu specifickou povahu vztahů (Krishna, 1992).

Množinové operace

- **Sjednocení, průnik a rozdíl**

Binární operace sjednocení je označována v teorii množin symbolem ∪. *Sjednocení* má za cíl sjednotit všechna fakta z argumentů. Relační operátor sjednocení záměrně není tak obecný, jako operátor sjednocení v matematice (Codd, 1990). Není povoleno sjednocení binární a ternární relace, protože důsledek takového spojení není relací. Tedy pro $R_1 \cup R_2$ musí být splněny následující podmínky (Group, 2005):

1. Relace R_1 a R_2 jsou stejné arity (tedy stejný počet atributů).
2. Domény *i*-tého atributu R_1 a *i*-tého atributu R_2 jsou stejné.

Výsledkem operace *průniku* je relace, která zahrnuje všechny *n*-tice, které jsou v obou R_1 a R_2. Průnik operací je označován $R_1 \cap R_2$.

Operace *rozdílu* je označována $R_1 - R_2$. Výsledkem operace je relace, která obsahuje všechny *n*-tice v R_1, ale ne v R_2.

Sjednocení i průnik jsou operace komutativní a asociativní. To znamená, že následující výrazy jsou pravdivé:

$A \cup B = B \cup A$ a $A \cap B = B \cap A$

$A \cup (B \cup C) = (A \cup B) \cup C$ a $A \cap (B \cap C) = (A \cap B) \cap C$.

- **Kartézský součin**

Kartézský součin dvou vztahů, R_1 a R_2, je zapsán v infixovém zápisu jako $R_1 \times R_2$. K definici konečných schémat relací, je nutné použít plně kvalifikované názvy atributů. To znamená, připojit název relace před atribut. Tímto způsobem je možné rozlišit relace $R_1.A$ a $R_2.A$.

Definice: Je-li $R_1(A_1, \dots, A_n)$ a $R_2(A_1, \dots, A_n)$, pak kartézský součin $R_1 \times R_2$ je relace se schématem obsahující všechny plně kvalifikované názvy atributů z R_1 a R_2: $(R_1.A_1, \dots, R_1.A_n, R_2.A_1, \dots, R_2.A_n)$. (Group, 2005)

Operace pro relační databáze

- **Operace projekce**

Projekce je unární operace označována řeckým písmenem PI (Π). Intuitivně slouží projekce k potlačení sloupců. Tedy přesněji projekce relace na podmnožinu atributů je relace tvořená vynecháním některých atributů. Syntaxe zápisu je následující:

$$\Pi_{\text{čárkou oddělený seznam atributů}}(relace).$$

- **Operace selekce**

Selekce je unární operace, která vybere *n*-tice, které splňují daný predikát. Podobně jako projekce, která vybere podmnožinu atributů, selekce vybere podmnožinu *n*-tic. Malé řecké písmeno sigma (σ) se používá k označení selekce:

$$\sigma_{podmínka\ výběru}(relace).$$

Podmínka selekce může zahrnovat (Group, 2005):

- konstanty,
- názvy atributů,
- aritmetické porovnávání ($=, \neq, <, \leq, >, \geq$),
- logické operátory (AND, OR, NOT).

- **Operace přejmenování**

Přejmenování je unární operace označována řeckým písmenem ró (ρ). Výsledkem použití operátoru přejmenování na relaci je relace shodná s původní kromě toho, že relace a její atributy dostávají nová jména. Nová jména pro relace a její atributy se zapisují jako index za ρ. Nové pojmenování relací je v seznamu první, následuje seznam nových názvů atributů oddělených čárkou a uvedených v závorkách. Obecně lze operaci přejmenování použít v těchto formách (Group, 2005):

$\rho_{nová\ relace(nová\ jména\ atributů)}$(relace),

$\rho_{nová\ relace}$(relace),

$\rho_{(nová\ jména\ atributů)}$(relace).

- **Operace spojení [Join]**

Operace spojení je binární operace (označována ⋈) používající se pro kombinaci souvisejících *n*-tic ze dvou relací (tabulek) do jedné *n*-tice (Sirangelo, 2009). Operace spojení je velmi důležitá pro všechny relační databáze s více než jednou tabulkou,

protože umožňuje zpracovat více než jednu tabulku najednou. Operace spojení odpovídá datům ze dvou nebo více tabulek, na základě hodnot jednoho nebo více sloupců v jednotlivých tabulkách. Mezi typy operací spojení patří přirozené spojené, vnější spojení (levé a pravé) a theta spojení:

Theta spojení [Theta join]

Theta spojení umožňuje kombinovat selekci a kartézský součin do jediné operace.

Definice: Máme-li dvě relace $r_1(R_1)$a $r_2(R_2)$. Potom formální zápis je:

$$r_1 \bowtie_\Theta r_2 = \sigma_\Theta(r_1 \times r_2).$$

Přirozené spojení [Natural join]

Definice: Máme-li dvě relace $r_1(R_1)$a $r_2(R_2)$. Přirozené spojení r_1 a r_2 je relace na schématu $R_1 \cup R_2$. Potom formální zápis je:

$$r_1 \bowtie r_2 = \Pi_{R_1 \cup R_2}(r_1 \bowtie r_1.A_1 = r_2.A_1 \wedge \ldots \wedge r_1.A_n = r_2.A_n\, r_2).$$

Vnější spojení [Outer join]

Definice: Máme-li dvě relace $r_1(R_1)$a $r_2(R_2)$. Potom formální zápis pravého spojení je:

$$r_1 \supset\!\bowtie r_2 = (r_1 \bowtie r_2) \cup ((r_1 - \Pi_{r_1.A_1,\ldots,r_1.A_n}(r_1 \bowtie r_2)) \times \{(null, \ldots, null)\}).$$

Definice: Máme-li dvě relace $r_1(R_1)$a $r_2(R_2)$. Potom formální zápis levého spojení je:

$$r_1 \bowtie\!\sqsubset r_2 = (r_1 \bowtie r_2) \cup ((r_2 - \Pi_{r_2.A_1,\ldots,r_2.A_n}(r_1 \bowtie r_2)) \times \{(null, \ldots, null)\}).$$

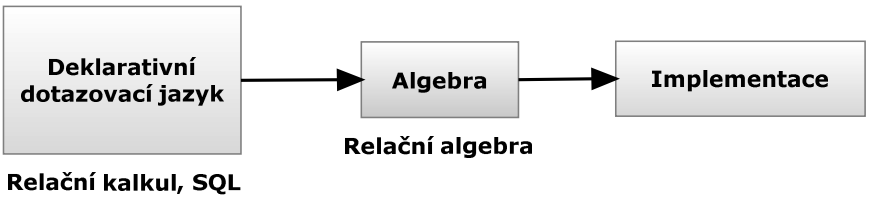
1.2.4. Relační dotazovací jazyky

Dotazovací jazyk [relational query language] je specializovaný jazyk pro vyhledávání a změnu obsahu databáze. I přesto, že termín se odkazuje na jazyky umožňující pouze vyhledávání (dotazování) obsahu databáze, současné dotazovací jazyky, jako je SQL, jsou všeobecné jazyky pro interakci se SŘBD, včetně definování schématu, naplnění obsahu, vyhledávání, aktualizaci, definování integritních omezení, definování procedur, pravidel apod. (Risch, 2009(b))

Jazyky pro formulaci požadavků na výběr dat z relační databáze (dotazovací jazyky) se dělí do dvou skupin (Šarmanová, 2007):

1. Jazyky založené na relační algebře, kde jsou výběrové požadavky vyjádřeny jako posloupnost speciálních operací prováděných nad daty. Dotaz je tedy zadán algoritmem, jak vyhledat požadované informace;
2. Jazyky založené na predikátovém kalkulu, které požadavky na výběr zadávají jako predikát charakterizující vybranou relaci. Úlohou překladače jazyka je nalézt odpovídající algoritmus; tyto jazyky se dále dělí na:
    - *n*-ticové relační kalkuly,
    - doménové relační kalkuly.

Oba typy jazyků jsou ekvivalentní, vzhledem k možnostem formulace výběrových podmínek. Každý požadavek formulovaný v relační algebře se dá vyjádřit v relačním kalkulu a naopak. (Šarmanová, 2007) Vztah mezi relační algebrou a kalkulem je vyjádřen na obrázku č. 5.



Obrázek č. 5 - Relační algebra (vlastní zpracování).

1.2.4.1. Relační kalkul

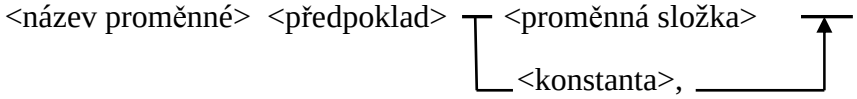
V roce 1972 Codd představil termíny relační algebra a relační kalkul (Codd, 1972(b)). Později se stalo zvykem mluvit o tzv. doménovém relačním kalkulu, který úzce souvisí se syntaxí prvně jmenovaného a o *n*-ticovém relačním kalkulu, který je ve skutečnosti ten, který publikoval Codd (1972(b)). Tyto dva kalkuly jsou rovnocenné, prostřednictvím jednoduchých převodů tam a zpět.

Oba kalkuly umožňují formulaci doménově závislých dotazů nevhodných pro databázové jazyky. Doménová nezávislost je v těchto jazycích neřešitelná, ovšem prostřednictvím sub-jazyka je možné definovat bezpečné dotazy, které jsou samy o sobě doménově nezávislé. (Tannen, 2009)

_N_-ticový relační kalkul

N-ticový relační kalkul je založen na použití *n*-tic proměnných. *N*-ticová proměnná je proměnná, která je určena pojmenováním relace, to znamená, že to je proměnná, jejíž hodnoty jsou dány příslušnou *n*-tic (Singh, 2009).

Definice: Pokud *n*-tice proměnných *R* reprezentuje *n*-tici *r* v nějakém bodě, pak $R.A$ reprezentuje složku *A* z *r*, kde *A* je atributem z *R*. Term lze zapsat jako (Singh, 2009):



kde <název proměnné> = <*n*-ticová proměnná>.<název atributu>

$= R.A$

<předpoklad> = binární operace

= NOT, $<, \leq, >, \geq$

Ve vymezení správného tvaru formule (WFF) [well-formed formula] jsou následující symboly, které se běžně používají v predikátu:

$\neg =$ negace;

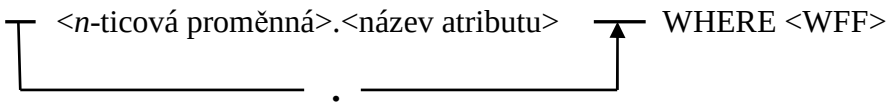
$\exists =$ existenční kvantifikátor používaný ve formuli, který musí být pravdivý alespoň pro jeden objekt;

$\forall =$ universální kvantifikátor používaný ve výrocích o všech objektech.

N-tice proměnných, které jsou kvantifikovány pomocí $\forall$ nebo $\exists$ se nazývají vázané proměnné a ty, které nejsou, se nazývají volné proměnné. Codd (1972(b)) definuje WFF následovně:

- Jakýkoli *term* je WFF.
- Jestliže *x* je WFF, pak $(x) = \neg x$. Všechny volné *n*-ticové proměnné v *x* zůstávají volné v (x) a $\neg x$, a všechny vázané *n*-ticové proměnné v *x* zůstávají vázané v (x) a $\neg x$.
- Jestliže *x*, *y* jsou WFF, pak jsou $x \wedge y$ a $x \vee y$. Všechny volné *n*-ticové proměnné v *x* a *y* zůstávají volné v $x \wedge y$ a $x \vee y$.
- Jestliže *x* je WFF a obsahuje volnou *n*-ticovou proměnnou *T*, pak $\exists T(x)$ a $\forall T(x)$ jsou WFF. *T* se stává vázanou *n*-ticovou proměnnou, ale ostatní volné *n*-ticové proměnné zůstávají volné. Všechny vázané termy v *x* zůstávají vázané v $\exists T(x)$ a $\forall T(x)$.
- Žádné jiné formule nejsou WFF.

Výraz kalkulu může být ve tvaru níže uvedeného tak, aby všechny *n*-ticové proměnné předcházely slovo WHERE, za kterým jsou volné *n*-ticové proměnné v WFF (Singh, 2009).



Doménový relační kalkul

Doménový relační kalkul byl navržen Lacroixem a Pirottem v roce 1977. V doménovém relačním kalkulu proměnné odvozují své hodnoty z domén atributů místo z *n*-tic relací. Výraz pro doménový relační kalkul má následující obecný tvar (Singh, 2009):

$\{d_1, d_2, \dots, d_n \mid F(d_1, d_2, \dots, d_m)\}\ \ m \geq n,$

kde $d_1, d_2, \dots, d_n$ reprezentují doménové proměnné a $F(d_1, d_2, \dots, d_m)$ reprezentují formule složené z atomů.

Doménový relační kalkul používá stejné operátory jako *n*-ticový kalkul. Rozdíl je v tom, že v doménovém kalkulu, namísto použití *n*-ticových proměnných, jsou používány doménové proměnné reprezentující *n*-ticové složky. Výraz *n*-ticového kalkulu může být přeměněn na vyjádření v doménovém kalkulu nahrazením každé *n*-ticové proměnné *n* doménovými proměnnými. Zde *n* je arita *n*-ticové proměnné. (Singh, 2009)

1.2.4.2. Strukturovaný dotazovací jazyk

Strukturovaný dotazovací jazyk (SQL) [structured query language] je světově nejpoužívanější databázový dotazovací jazyk. Podporuje vyhledávání, zpracování a správu dat uložených ve formě tabulek. SQL bylo uživatelské rozhraní definované v rámci projektu výzkumu System R. Mezi hlavní cíle, které ovlivnily konstrukci SQL, patří následující (Chamberlin, 2009):

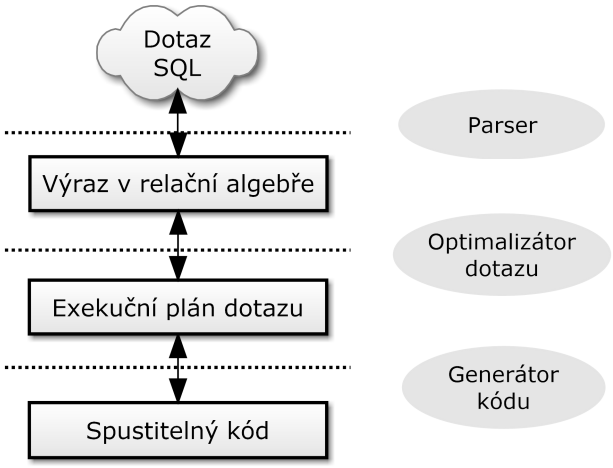
1. SQL patří mezi vyšší jazyky. Je to neprocedurální jazyk určený pro zpracování optimalizace kompilátoru. Je navržen tak, aby byl ekvivalentní s relačními jazyky, které původně navrhl Codd.
2. SQL má být přístupný uživatelům bez formálního vzdělání v matematice a programování.

3. SQL se pokouší sjednotit datové dotazy a úkoly pro správu databází, jako je vytváření a úpravy tabulek a pohledů, řízení přístupu k datům a definování omezení k ochraně integrity databáze.
4. SQL je určen pro podporu rozhodování a on-line zpracování transakcí.

První specifikace SQL byla zveřejněna v květnu 1974 (Chamberlin, et al., 1974(b)). V tomto dokumentu byl jazyk pojmenován SEQUEL, zkratka pro „anglický strukturovaný dotazovací jazyk“. Poslední významná revize normy je SQL: 2003.

Procesor dotazů

Procesor dotazů [query processor] přijímá dotaz, ověřuje jej, optimalizuje procedurální datový tok prostřednictvím exekuční plánu a realizuje jej pro získání výsledků dotazu (Pitoura, 2009(c)). Podobně vymezuje pojem Ailamaki (2009); *Procesor dotazů* v SŘBD přijímá jako vstupní požadavek dotazy ve formě textu SQL, analyzuje je, vytváří exekuční plán. Následovně dokončí zpracování provedením plánu a vrácením výsledku klientovi. *Dotazové* nebo *exekuční plány* přesně určují, jak má být dotaz vykonán (Pitoura, 2009(a)).



Obrázek č. 6 - Proces zpracování SQL dotazu (podle Pitoura, 2009).

Zpracování dotazů se skládá z několika fází. V první fázi parser dotazů [query parser] kontroluje, zda je dotaz správně uveden, řeší veškeré názvy a odkazy, ověřuje konzistenci a provádí autorizační testy. Pak přepisovací modul [query rewrite modul], zjednodušuje dotaz a přeměňuje jej na odpovídající formu tím, že provádí několik optimalizací, které nejsou závislé na fyzickém stavu systému. V dalším kroku optimalizátor dotazu transformuje interní reprezentaci dotazu na efektivní plán dotazu. (Pitoura, 2009(c)) *Optimalizace dotazů* je zaměřena na co nejefektivnější výběr

přístupových cest pro daný dotaz. Úkolem optimalizace dotazů je najít nejefektivnější celkové provedení dotazu. (Markl, 2009) Generátor kódu transformuje plán vytvořeného optimalizátorem do exekučního (spustitelného) plánu. Nakonec exekuční stroj zpracuje plně specifikovaný plán dotazu. (Pitoura, 2009(c)) Celý proces zpracování SQL dotazů je znázorněn na obrázku č. 6.

Příklad dotazu v relační algebře a SQL

Je stanoveno následující nenormalizované relační schéma:

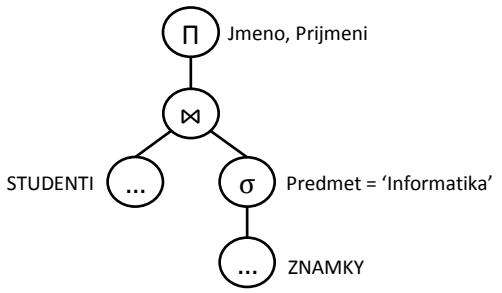
```
STUDENTI(Jmeno, Prijmeni, RC, Datum narozeni, Predmet);
ZNAMKY(Hodnoceni, Předmět);
```

Dotaz vyhledá všechny studenty, kteří mají známku z předmětu Informatika a zobrazí jejich jména a příjmení. Na schéma je aplikován dotaz v relační algebře a SQL následovně:

Zápis v relační algebře:

```
π{Jmeno, Prijmeni}(STUDENTI ⋈
σ{Predmet='Informatika'}(ZNAMKY)
                )
```

Strom:



Zápis v SQL:

```
SELECT Jmeno,Prijmeni FROM STUDENTI NATURAL JOIN (SELECT * FROM ZNAMKY
WHERE Predmet = 'Informatika').
```

1.2.5. Relační SŘBD

SŘBD se skládá z komponent (modulů), které mohou být definovány na různé úrovni. *Komponentou* je myšlen samostatný software, který je součástí databázového systému. Komponenty relačního databázového systému se skládají z klienta pro řízení komunikace, řízení procesů, procesoru pro tvorbu dotazů, transakčního správce úložiště a nástrojů (Hellerstein, et al., 2007).

Klient pro řízení komunikace se skládá z místních a vzdálených klientských protokolů. Řízení procesů se skládá z přijímací, odesílací a plánovací částí. Procesor pro tvorbu

relačních dotazů se skládá z analýzy dotazů a autorizace přepisovacích dotazů, optimalizačních dotazů, plánu realizace [execution plan], DDL a procesních nástrojů. Transakční správce úložiště se skládá z přístupových metod, řízení vyrovnávací paměti, správce zámků a log manažera. Mezi dílčí komponenty, které jsou součástí nástrojů, patří katalog, správce paměti a replikační služby (Gehrke, 2009)

1.3. Multidimenzionální databáze

Pojem *multidimenzionální data* [multidimensional data] se používá ve dvou různých významech v oblasti řízení dat. V prvním smyslu pojem představuje data souhrnných ukazatelů vytvořených různým seskupením relačních dat určených pro on-line analytické zpracování (OLAP). *OLAP* [on-line analytical processing] popisuje přístup pro podporu rozhodování, jehož cílem je získat znalosti z datového skladu, nebo z datových tržišť (Abelló, et al., 2009). V druhém smyslu se pojem *multidimenzionální data* odkazuje na data, která lze popsat jako pole heterogenních datových typů spolu s metadaty k jejich popisu. (Gupta, 2009) *Dimenze* je hierarchicky uspořádaný soubor rozměrových hodnot, které poskytují kategorické informace charakterizující určitý aspekt dat uložených v multidimenzionální kostce. (Pedersen, 2009(b)) *Datová kostka* [cube] je datová struktura pro ukládání a analýzu velkého množství vícerozměrných dat (Pedersen, 2009(a)). Datová kostka je všeobecně přijímána jako základní logická struktura k popsání multidimenzionální databáze, stejně jako relace pro relační databáze.

1.3.1. Multidimenzionální model dat

Datová kostka je základní konstrukt multidimenzionálních databází a slouží jako základní jednotka vstupu a výstupu pro všechny operátory definované na multidimenzionální databázi (Datta, et al., 1999).

Definice: Máme-li čtveřici $< D, M, A, f >$, kde čtyři složky indikují vlastnosti kostky. Pak tyto vlastnosti jsou (Datta, et al., 1999):

1. Množina n dimenzí (rozměrů) $D = \{d_1, d_2, \dots, d_n\}$, kde každé d_i je jméno dimenze získané z domény $dom_{\dim(i)}$.
2. Množina k měr $M = \{m_1, m_2, \dots, m_k\}$, kde každé m_i je jméno míry získané z domény $dom_{\text{measure}(i)}$.
3. Množina jmen dimenzí a měr je disjunktní; tj. $D \cap M = 0$.

4. Množina *t* atributů $A = \{a_1, a_2, \dots, a_t\}$, kde každé a_i je jméno atributu získané z domény $dom_{\text{attr}(i)}$.
5. Zobrazení jeden k mnoha $f: D \rightarrow A$, existuje pro každou dimenzi a množinu atributů. Zobrazení je takové, že množina atributů příslušející k dimenzi je v páru disjunktní, tj. $\forall i, j, i \neq j, f(d_i) \cap f(d_j) = 0$.

Definice uvedená výše představuje abstraktní strukturu. Pro připisování hodnot jednotlivých měr v rámci všech rozměrů je nutné definovat šestici $< D, M, A, f, V, g >$, kde elementy D, M, A a f jsou zděděné od rodičovské kostky, V představuje množinu hodnot, které byly použity k realizaci kostky. Tedy element $v_i \in V$ je k-tice $< \mu_1, \mu_2, \dots, \mu_k >$, kde μ_i je instance *i*-té míry m_i. Nakonec g reprezentuje zobrazení $g: dom_{\text{dim}(1)} \times dom_{\text{dim}(2)} \times \dots \times dom_{\text{dim}(n)} \rightarrow V$. Intuitivně tedy zobrazení g ukazuje, které hodnoty jsou spojeny s konkrétní „buňkou“. Takové naplnění kostky se nazývá *instance kostky* [cube-instance]. (Datta, et al., 1999)

1.3.2. Pravidla E. F. Codda pro OLAP

OLAP je vymezen pomocí 12 pravidel stanovených Coddem (1993). Vzhledem k tématu disertační práce považuje autor práce za důležité tato základní pravidla neopomenout a naopak akcentovat jejich stálý význam. Stručný výčet pravidel je uveden v příloze č. 1.

1.3.3. OLAP algebra

OLAP algebra je sestavená z množiny operací podobně jako relační algebra. Tyto OLAP operace jsou definovány následovně (Datta, et al., 1999):

Restrikce (σ)

Definice: Operace restrikce omezuje hodnoty na jeden nebo více rozměrů. Pokud máme atomový predikát p, pak logický výraz zahrnuje jednu dimenzi. Pokud máme kombinaci predikátů P, pak výraz obsahuje množinu atomových predikátů $\{p_1, p_2, \dots, p_l\}, l \geq 1$ ve tvaru:

$$P = p_1 < op > p_2 < op > \cdots < op > p_l,$$

kde $< op >$ reprezentuje logický operátor (and, or).

Matematická notace je: $\sigma_P(C_I) = C_o$, pro:

- Vstup: kostka $C_I =< D, M, A, f, V, g >$ a kombinace predikátů P.
- Výstup: kostka $C_o =< D_o, M_o, A_o, f_o, V_o, g_o >$, kde $D_o = D$; $M_o = M$; $A_o = A$; $f_o = f$; $V_o \subseteq V$; a $g_o = g_o$;

Agregace (α)

Definice: Operátor agregace provede agregace na jednu nebo více dimenzí. Pokud h je agregační funkce definovaná na jediné míře m_i a S je množina seskupení dimenzionálních atributů $\{a_1, a_2, \dots, a_q\}$ taková, že $S \subseteq A$, pak je možné stanovit jedna k jedné zobrazení $\delta: A \to D$, kde δ představuje zobrazení atributů a_i na jméno dimenze d_i.

Matematická notace je: $\alpha_{h,m_i,S}(C_I) = C_o$, pro:

- Vstup: kostka $C_I =< D, M, A, f, V, g >$, míra agregace m_i a množina seskupení dimenzionálních atributů S.
- Výstup: kostka $C_o =< D_o, M_o, A_o, f_o, V_o, g_o >$, kde $D_o = \{d_1, d_2, \dots, d_q\}$; $q = |S|$; $\forall a_i \in S, d_i = \delta(a_i)$; $M_o = \{m_i\}$; $A_o = \cup_{\forall d_i \in D_o} f(d_i)$; a $f_o = f$. V_o reprezentuje hodnoty získané použitím agregační funkce h na prvky V a g_o představuje zobrazení $g_o: dom_{\dim(1)} \times dom_{\dim(2)} \times \dots \times dom_{\dim(q)} \to V_o$.

Kartézský součin ($\times$)

Definice: Kartézský součin je binární operací, která může být použita na jakékoliv dvě kostky.

Matematická notace je: $C_{I1} \times C_{I2} = C_o$, pro:

- Vstup: kostka $C_{I1} =< D_1, M_1, A_1, f_1, V_1, g_1 >$ a kostka $C_{I2} =< D_2, M_2, A_2, f_2, V_2, g_2 >$.
- Výstup: kostka $C_o =< D_o, M_o, A_o, f_o, V_o, g_o >$, kde $D_o = D_1 \cup D_2$; $M_o = M_1 \cup M_2$; $A_o = A_1 \cup A_2$; $V_o = V_1 \times V_2$, a $|V_o| = |V_1| \times |V_2|$ reprezentuje hodnoty získané použitím agregační funkce h na prvky V. f_o je odvozena z f_1 a f_2; g_o představuje zobrazení $g_o: dom_{\dim(1)} \times dom_{\dim(2)} \times \dots \times dom_{\dim(q)} \to V_o$, kde $q = |D_o|$.

Spojení (⋈)

Definice: Dvě kostky můžeme spojit právě když $D_1 \cap D_2 \neq 0$ a $\forall d_i \in (D_1 \cap D_2)$, $f_1(d_i) = f_2(d_i)$. Dále $D_1 \cap D_2$ je označena jako společná dimenze (cd), tedy $D_1 \cap D_2 = \{cd_1, cd_2, \dots, cd_l\}$. Algebru spojení můžeme vyjádřit prostřednictvím následující identity:

$C_1 \bowtie C_2 = \sigma_P(C_1 \times C_2)$, kde P je predikát ve formě $P = [(C_1 cd_1 = C_2 cd_1) \wedge (C_1 cd_2 = C_2 cd_2) \wedge \dots \wedge (C_1 cd_1 = C_2 cd_1)]$.

Kompatibilní sjednocení

Neformálně vyjádřeno, pokud mají dvě kostky stejný počet dimenzí a měr, soulad 1:1 mezi dimenzemi a mírami napříč dvěma kostkami, pak jsou kompatibilní pro sjednocení.

Sjednocení (∪)

Operace sjednocení najde sjednocení dvou kostek. Matematická notace je: $C_{I1} \cup C_{I2} = C_o$.

Rozdíl (−)

Operace rozdíl najde rozdíl dvou kostek. Matematická notace je: $C_{I1} - C_{I2} = C_o$.

Průnik (∩)

Operace průniku není základní operací, proto může být vyjádřena jako: $C_{I1} - (C_{I1} - C_{I2}) = C_o$, nebo také $C_{I1} \cap C_{I2} = C_o$.

Pull (ϕ)

Definice: Operace *pull* převádí míry na dimenze. Matematická notace je: $\phi_{R,D_{R,K}}(C_I) = C_o$, pro:

- Vstup: kostka $C_I =< D, M, A, f, V, g >$, množina měr k transformaci R, množina jmen dimenzí D_R a zobrazení jmen dimenzí κ.
- Výstup: kostka $C_o =< D_o, M_o, A_o, f_o, V_o, g_o >$, kde $D_o = D \cup \kappa(d_{Ri})$; $M_o = M - R$; $A_o = A \cup f_o(\kappa(d_{Ri}))$; f_o je odvozeno z f; $\forall m_i \in R, f_o(m_i) \rightarrow d_{Ri}$.

Push (ψ)

Definice: Operace *push* převádí dimenze na míry. Matematická notace je: $\psi_{d_i}(C_I) = C_o$, pro:

- Vstup: kostka $C_I =< D, M, A, f, V, g >$, jména dimenzí k transformaci d_i.
- Výstup: kostka $C_o =< D_o, M_o, A_o, f_o, V_o, g_o >$, kde $D_o = D - d_i$; $M_o = M \cup f(d_i)$; $A_o = A - f(d_i)$; $f_o = f$. Pro V_o, $dom_{V_o} = dom_{\mu_1} \times \ldots \times dom_{\mu_p}$ takové, že $f(d_i) = \{\mu_1, \ldots, \mu_p\}$ a $|V_o| = |V|$.

Partition (ψ)

Definice: Operace *partition* mapuje body kostky do smysluplných skupin. Takové rozdělení je třeba pro určité typy agregace. Matematická notace je: $\gamma_{t,R}(C_I) = C_o$, pro:

- Vstup: kostka $C_I =< D, M, A, f, V, g >$, množina oddělených atributů dimenze R a množina oddělených funkcí t.
- Výstup: kostka $C_o =< D_o, M_o, A_o, f_o, V_o, g_o >$, kde $D_o = D$; $M_o = M$; $\forall a_i \in R$ $A_o = A \cup t(a_i)$ a $f_o = f$. Pro V_o, $|V_o| = |V|$.

1.3.4. Typické OLAP operace

Bohužel neexistuje žádná shoda ohledně multidimenzionálních operací a jejich pojmenování. Nicméně, příspěvek (Romero, et al., 2007) poskytuje srovnání algebraických návrhů OLAP operací. V současnosti jsou nejčastěji v literatuře prezentovány níže uvedené OLAP operace. Následující operace jsou vyjádřeny v jazyku OCL podle (Pardillo, et al., 2010):

Selekce nebo řezy [selection or dice]

Tato operace umožňuje pomocí logiky predikátu přes atributy dimenze vybrat uživatelům podmnožinu bodů zájmu z celé *n*-dimenzionálního prostoru (Abelló, et al., 2009). Zápis v jazyku OCL je: $sliceAndDice : Cube \times (Cell \times Dice) \rightarrow Cube$.

Roll-up (nebo také drill-up)

Operace seskupí buňky v kostce na základě agregace hierarchie. Tedy změní granularitu dat pomocí vztahu M:1, který se týká dvou agregačních úrovní ve stejné dimenzi (Abelló, et al., 2009). Zápis v jazyku OCL je: $rollUp : Cube \times (Axis \times Rolling \times Additivity) \rightarrow Cube$.

Drill-down

Operace dril-down je opakem operace roll-up, tedy: $drillDown : Cube \times (Axis \times Drilling \times Additivity) \rightarrow Cube$.

Drill-across

Operace změní předmět analýzy kostky zobrazením míry týkající se nového faktu. Zápis v OCL je: $drillAcross : Cube \times (Conformity \times Cube) \rightarrow Cube$.

Projekce

Vybere podmnožinu měr z dostupných měr v kostce. Zápis v OCL je: $dimensionalProject : Cube \times (Measure) \rightarrow Cube$.

1.3.5. Multidimenzionální deklarativní dotazovací jazyky

Datové sklady obvykle ukládají obrovské množství dat. Tento aspekt je výhodný a přitom náročný zároveň, protože konkrétní dotazování/aktualizování/modelování vlastností zpracování dotazu je poměrně obtížné vzhledem k vysokému počtu stupňů volnosti (Lehner, 2009). Proto bylo nutné navrhnout jazyk podobný SQL v relačních databázích. Cabibbo a Torlone (1998) navrhli grafický dotazovací jazyk, Gyssens a Lakshmanan (Gyssens, et al., 1997) navrhli kalkulus. Ovšem v současnosti nejrozšířenější jazyk se stal MDX (Microsoft, 2011), představený v roce 1997 společností Microsoft. Syntaxe se podobá SQL (Microsoft, 2011):

```
WITH
   MEMBER [Measures].[Special Discount] AS
   [Measures].[Discount Amount] * 1.5
SELECT
   [Measures].[Special Discount] on COLUMNS,
   NON EMPTY [Product].[Product].MEMBERS  ON Rows
FROM [Adventure Works]
WHERE [Product].[Category].[Bikes]
```

Nicméně, sémantika MDX je zcela odlišná. Zjednodušeně řečeno, dotaz MDX získá instanci dané kostky uvedené v klauzuli FROM a umístí ji do prostoru definované klauzulí SELECT. Kromě toho mohou být definovány složité výpočty v klauzuli WITH

a dimenze, které nejsou použity v klauzuli SELECT, mohou být rozděleny v klauzuli WHERE. (Abelló, et al., 2009)

Zpracování dotazů [query processing]

OLAP databáze se vyznačují v analytické oblasti širokou škálou metod zpracování dotazu. Například různé metody jsou představeny v příspěvcích autorů jako Chaudhuri (1997), Kalnisn (2003), Pardillo (2010). Následující přehled poskytuje důležité aspekty zpracování dotazů v systémech datových skladů podle Lehnera (2009):

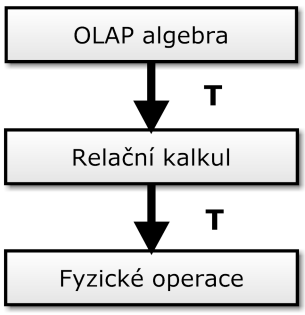
1. **Požadavky a specifika analytického kontextu**

Zpracování dotazů v prostředí datových skladů se zabývají dotazovací jazyky, jako jsou MDX, DMX, nebo SQL / XML s analytickou funkcí v XPath / XQuery. Složitost zpracování doménově-specifických dotazovacích jazyků a zobrazení těchto výrazů je vysoce náročné vzhledem ke struktuře dat. Analytické dotazy obvykle provádí agregace ve velkém počtu řádků. Přesto mohou existovat analytické dotazy vykazující velmi selektivní predikáty.

2. **Potenciál a řešení efektivního zpracování dotazů datového skladu**

Specifické charakteristiky, stejně jako konkrétní okolnosti vytvářejí širokou škálu možností, jak provádět zpracování dotazů v datovém skladu efektivním způsobem.

Proces zpracování SQL dotazů pro datové sklady je znázorněn na obrázku č. 7. Toto znázornění je podobné relačnímu databázovému zpracování dotazů. Ovšem nutné je pro zpracování dotazu transformovat OLAP algebru do relačního kalkulu.



Obrázek č. 7 - Proces zpracování OLAP dotazu (vlastní zpracování).

Datové kostky zachycují všeobecné trendy agregovaných multidimenzionálních dat z kategorických vztahů. Trendy nebo zvláštní jevy nesouvisející v jedné datové krychli

mohou souviset v jiných datových krychlích. Pro zachycení této změny trendu byla Nedjarem (2011) navržena koncepce založená na SQL pod názvem *formující se kostka* [emerging cube]. (Nedjar, et al., 2011)

Mnoho autorů se zabývá problematikou minimalizace nákladů plánu datové kostky. Lee (2010) vyvinul heuristický algoritmus, který garantuje, že minimální počet *delta* kvádru zachovává 2^n kvádrů pro n dimenzionálních atributů. Problematikou efektivní roll-up a drill-down operace s cílem maximalizovat výkon se zabývá Doka (2011).

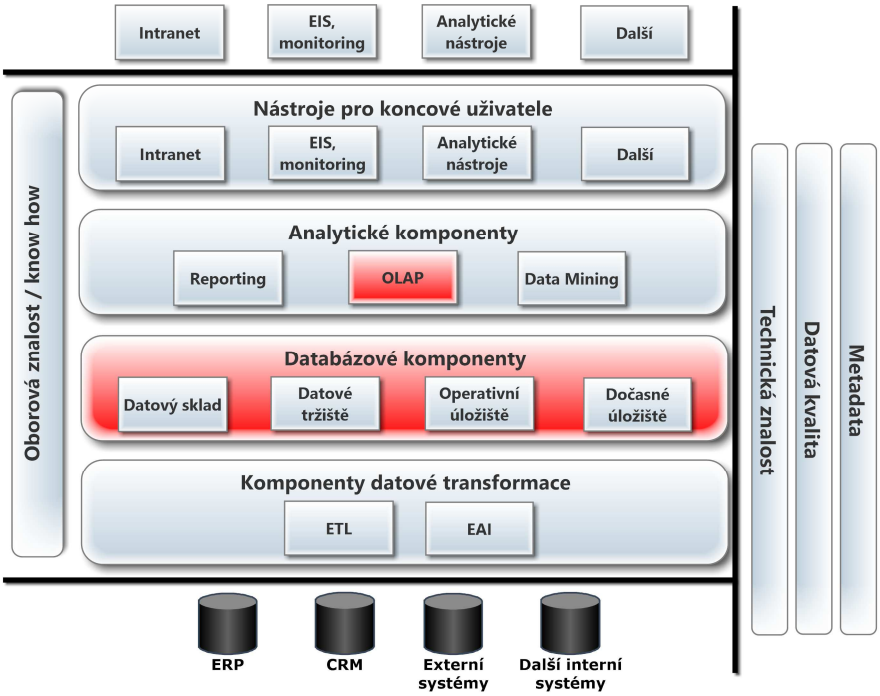
1.3.6. Úložiště multidimenzionálních dat

Datta (1999) uvádí dva hlavní způsoby k podpoře OLAP aplikací: multidimenzionální OLAP (MOLAP) servery a relační OLAP (ROLAP) servery. Přístup *MOLAP* fyzicky ukládá data v poli jako struktury, které jsou podobné datové kostce uvedené na obrázku č. 10. V přístupu *ROLAP*, jsou data uložena v relační databázi pomocí speciálního schématu namísto tradičního relačního schématu. Více o této problematice je pojednáno v části 1.6.4.

1.4. Business Intelligence

Multidimenzionální databáze představují v současnosti důležitou komponentu pro OLAP, která je v rámci obecné koncepce (Novotný, a další, 2005) součástí analytické vrstvy business intelligence (BI), tak jak ilustruje obrázek č. 8. Jednotlivé komponenty v řešení se mohou výrazně lišit a nejsou v literatuře pevně vymezeny.

BI je možné charakterizovat jako systémy poskytující schopnost analyzovat podnikové informace s cílem podpořit a zlepšit řízení rozhodování v celé řadě podnikatelských aktivit. (Elbashir, et al., 2008) BI je proces, jehož prostřednictvím organizace využívají virtuální a digitální technologie pro shromažďování, správu a analýzu strukturálních nebo nestrukturálních dat. BI představuje zpracování technologických a obchodních postupů v rozhodování podniku podporovaných prostřednictvím extrakce, integrace a analýzy dat. BI je nástrojem analýzy, poskytuje automatizované rozhodování o podmínkách podnikání, prodeji, poptávce zákazníků, produktů a tak podobně. Používá „velké“ databáze (datové sklady) pro analýzy, stejně jako matematické a statistické postupy, umělou inteligenci, dolování dat a on-line analýzy zpracování. (Rouhani, et al., 2012)



Obrázek č. 8 - Obecná koncepce architektury BI (podle Novotný, a další, 2005).

Výše uvedené vymezení pojmů BI si nějak významně nerozporují. Ovšem autor práce se bude přidržovat vymezení pojmu BI podle České společnosti pro systémovou integraci, která BI vymezuje jako „sadu procesů, aplikací a technologií, jejichž cílem je účinně a účelně podporovat rozhodovací procesy ve firmě. Podporují analytické a plánovací činnosti podniků a organizací a jsou postaveny na principech multidimenzionálních pohledů na podniková data.“ (Novotný, a další, 2005) Efekty BI pro výkonnost podniku a jeho řízení jsou představeny Pourem, a dalšími (2004). Měření účinků systémů BI jsou publikována v příspěvku (Elbashir, et al., 2008). Rozvoj trhu s BI je možné nalézt v příspěvku Poura (2010). Zajímavým přístupem v současnosti jsou tak zvané samoobslužné business intelligence [self-service BI]. Samoobslužné BI umožňují okamžitý přístup koncových uživatelů k vícerozměrným datům z desktopových aplikací, jako jsou kontingenční tabulky v tabulkových procesorech, bez nutnosti zvláštních technických znalostí transformačních procesů (Thanisch, a další, 2012).

1.4.1. Hlavní komponenty Business Intelligence a jejich vazby

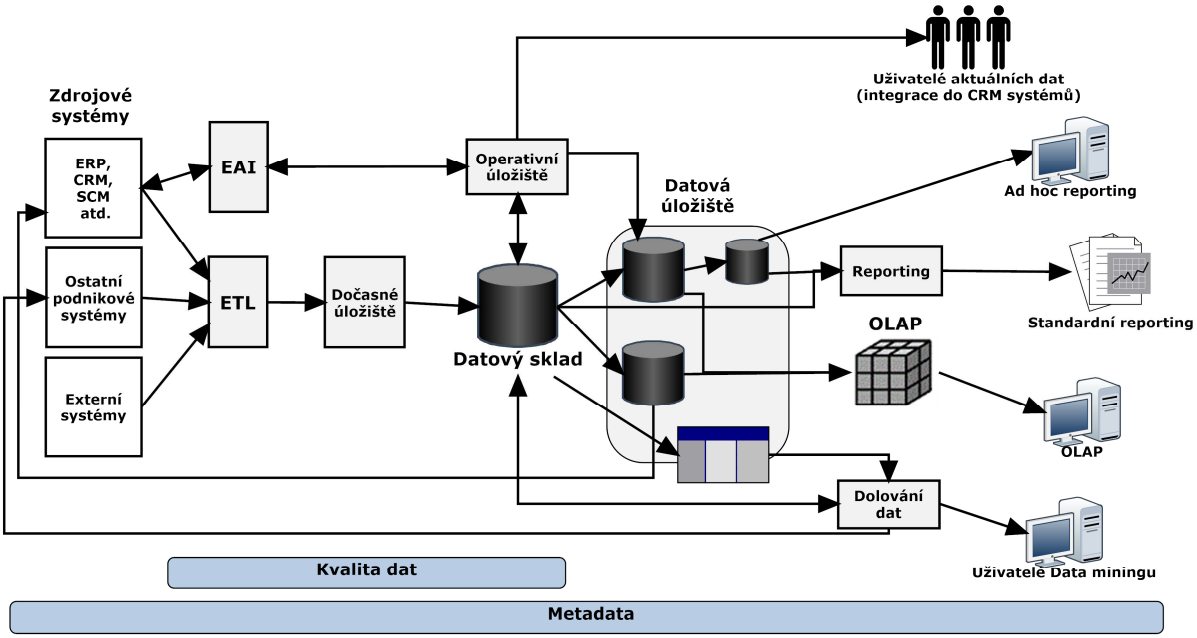
Z pohledu návrhu multidimenzionální databáze je důležité vymezení pojmů objevujících se v obecné koncepci BI (obrázek č. 8) a jejich vazby (obrázek č. 9). Podle (Novotný, a další, 2005) je možné identifikovat několik vrstev s tímto obsahem:

- **Vrstva pro extrakci, transformaci, čištění a nahrávání dat** (komponenty transformace dat), která pokrývá oblast sběru / přenosu dat ze zdrojových systémů do vrstvy pro ukládání dat v řešení BI:
    - ETL [Extract, Transform and Load] systémy - neboli systémy pro extrakci, transformaci a přenos dat.
    - EAI [Enterprise Application Integration] systémy - neboli systémy pro integraci aplikací.
- **Vrstva pro ukládání dat** (databázové komponenty), která zajišťuje procesy ukládání, aktualizace a správy dat pro řešení BI:
    - Datové sklady [Data Warehouse] - základní databázová komponenta řešení BI.
    - Datová tržiště [Data Marts] - subjektově orientované analytické databáze, součást nebo nadstavba datového skladu.
    - Operativní uložiště dat [Operational Data Store] - podpůrné analytické databáze.
    - Dočasná úložiště dat [Data Staging Areas] - databáze pro dočasné uložení dat před jejich vlastním zpracováním do databázových komponent řešení BI.
- **Vrstva pro analýzy dat** (analytické komponenty), pokrývající činnosti spojené s vlastním zpřístupněním dat a analýzou dat:
    - Reporting - analytická vrstva, zaměřená na standardní nebo ad hoc dotazovací proces do databázových komponent řešení BI.
    - Systémy OLAP [On-Line Analytical Processing] - vrstva zaměřená na pokročilé a dynamické analytické úlohy.
    - Dolování dat [Data Mining] - systémy zaměřené na sofistikovanou analýzu velkého množství dat.
- **Prezentační vrstva** (nástroje pro koncové uživatele), zajišťující komunikaci koncových uživatelů s ostatními komponentami řešení BI, tedy zejména sběr požadavků na analytické operace a následnou prezentaci výsledků:
    - Portálové aplikace založené na technologiích WWW.

- Systémy EIS [Executive Information Systems].
- Různé analytické aplikace.

- **Vrstva oborové znalosti** (oborová znalost/know-how), zahrnující oborovou znalost a tzv. best-practices nasazování řešení BI pro konkrétní situaci v organizaci.
- Aplikace BI kromě toho využívají následující **obecné komponenty pro správu a manipulaci s daty**:
  - Nástroje pro zajištění kvality dat, tedy nástroje zajišťující, že data přesně odpovídají realitě.
  - Nástroje pro správu metadat, zabývající se popisem a dokumentací systémů i probíhajících procesů.
  - Technickou znalost, zahrnující programovací a technologicky závislé schopnosti implementačního týmu.

Pohled na hlavní vazby v řešení BI je ilustrován na obrázku č. 9.



Obrázek č. 9 - Hlavní komponenty BI a jejich vazby (podle Novotný, a další, 2005).

1.4.2. Datový sklad

Datový sklad je integrované úložiště dat ve formě, která může být snadno srozumitelná, interpretovatelná a analyzovatelná lidmi, pro jejich podporu rozhodování. (Song,

2009(a)) Nejčastěji citované vymezení je od Inmona, který vymezuje datový sklad jako „předmětově orientovaný, integrovaný, stálý, a časově rozlišený sběr dat pro podporu rozhodování managementu“. (Inmon, 2002) Tyto vlastnosti znamenají: (Song, 2009(a))

Předmětově (subjektově) orientovaná vlastnost znamená, že data v datovém skladu jsou organizována kolem významných předmětů zájmů organizace. Příklady předmětů, jsou zákazníci, produkty, prodeje a prodejci. Tato vlastnost dovolí uživatelům datového skladu analyzovat každý předmět do hloubky pro taktické a strategické rozhodování.

Integrovaný znamená, že data v datovém skladu jsou integrována nejen ze všech operačních systémů, databází, ale také z některých metadat a dalších souvisejících externích dat. Jsou-li údaje přesunuty z provozních databází do datového skladu, jsou extrahovány, očištěny, transformovány, a pak nahrány. To dělá z datového skladu centralizované úložiště všech podnikových dat se společnou sémantikou a formátem.

Stálý znamená, že data v datovém skladu nejsou obvykle aktualizována. Jakmile jsou data načtena do datového skladu, nejsou odstraněna. Jakákoliv změna dat, které již byly přesunuty do datového skladu, je zaznamenána v podobě snímku. To umožňuje v datovém skladu sledovat historii dat.

Časově rozlišený znamená, že datový sklad obvykle obsahuje historická data. Není neobvyklé, že datové sklady obsahují data za více než deset let. To umožňuje uživatelům datových skladů analyzovat trendy, vzory, korelace, pravidla a výjimky z historického hlediska.

Velký počet aplikací může být postaven na datovém skladu, například aplikace zaměřené na podporu OLAP analýzy, dolování dat [data mining], what-if analýzy, prognózy, příprava BSC [balanced scorecards], geoprostorové analýzy, click-stream analýzy a další. Architektura může být doplněna o reaktivní tok dat, který se hodí pro sledování časově kritických provozních procesů podporující real-time aplikace. (Rizzi, 2009)

Přesto hlavním důvodem použití datových skladů je především nevhodnost provozních databází pro analýzy. Provozní databáze (OLTP), jsou v podnicích využívány především pro transakční operace. Prostřednictvím nich jsou ukládány operační údaje v podobě normalizovaných tabulek (obvykle ve třetí normální formě). „Takové systémy dosahují vysokých výkonů při on-line transakcích než při složitých analýzách, které jsou velmi náročné na výpočetní kapacitu procesorů.“ (Lacko, 2006)

Datový sklad je fyzicky oddělen od provozních databází a to z těchto důvodů (Singhal, 2007):

1. OLTP databáze jsou optimalizovány pro 3 normální formu, dobu odezvy transakcí a propustnost. OLAP databáze jsou tržně orientovaná a optimalizovaná pro analýzu dat podle manažerů a vedoucích pracovníků.
2. OLTP systémy se zaměřují na aktuální data, bez odkazu na historické údaje. OLAP se zabývá historickými daty, pocházejících z různých míst v organizaci.
3. Přístupový vzor pro aplikace OLTP se skládá z krátkých atomických transakcí. OLAP systémy jsou primárně navrženy pouze pro čtení. Transakce jsou založeny na provádění složitých dotazů.

Ovšem autor disertační práce se domnívá, že takovýto přístup vede k redundanci dat a větší složitosti navrhování BI. Důsledkem jsou pak mnohem větší náklady na BI. Wolfgang Lehner (2010) na panelu "Merging OLTP and OLAP" vyslovil otázku: „Jsou v současnosti potřeba datové sklady?“

1.4.3. Datová tržiště

Datové tržiště [data mart] je datový sklad malých rozměrů zaměřený na konkrétní předmět. Zatímco datový sklad je určen pro celý podnik. Datová tržiště jsou postavena tak, aby údaje v nich umožnila řešit konkrétní analýzy potřeb podnikatelské jednotky. Proto je možné datová tržiště definovat jako „datový sklad malé velikosti, který obsahuje podmnožinu podnikového datového skladu nebo omezený objem agregovaných údajů pro konkrétní analytické potřeby podnikatelské jednotky, spíše než pro potřeby celého podniku“. (Song, 2009(b))

Rozdíly mezi datovým skladem a datovým tržištěm uvádí Song (2009(b)) následující:

- Cílem datového skladu je zaměřit se na potřeby celého podniku, cílem datového tržiště je zaměřit se na potřeby podnikatelské jednotky (oddělení).
- Data v datovém skladu jsou pořizována ze systémů OLTP, data v datových tržištích, jsou pořizována z podnikového datového skladu.
- Granularita (zrnitost) datového skladu je na úrovni OLTP. Datové tržiště je obvykle lehce agregovatelné pro optimální analýzu oddělení.

- Pokrytí datového skladu je plně historické pro potřeby celého podniku. Datové tržiště je omezené na specifické potřeby oddělení.

V druhém bodě existuje ovšem rozpor. Existují totiž odlišná pojetí datového skladu Ralpha Kimballa a Billa Inmona. Tyto rozdíly jsou uvedeny v publikaci (Novotný, a další, 2005):

Datový sklad jako množina datových tržišť

Tvrzení Ralph Kimballa, že „datový sklad není nic jiného než sjednocení datových tržišť“ znamená, že namísto vytváření jednoho datového skladu jsou postupně budována jednotlivá datová tržiště. Logickým sjednocením těchto datových tržišť je pak datový sklad.

Integrovaný datový sklad

Při této koncepci se data z provozních systémů ukládají do centrálního datového skladu. Centrální celopodnikový datový sklad, se tak stává „srdcem“ podnikové architektury pro podporu rozhodování. Nad tímto datovým skladem jsou pak budována datová tržiště, která slouží pro podporu rozhodovacích procesů jednotlivých útvarů podniku. Tato architektura byla navržena Billem Inmonem s myšlenkou vytvořit architekturu, která by minimalizovala redundantní data a zároveň počet rozhraní mezi produkčními systémy a datovým skladem.

1.4.4. Přístupy k řešení Business Intelligence

V současnosti existuje několik přístupů k řešení BI. Novotný, a další (2005) uvádí tyto tři přístupy:

- přístup postupného budování datových tržišť, založený na architektuře nezávislých datových tržišť;
- přístup jednorázového vybudování celkového řešení, založený na architektuře konsolidovaného datového skladu;
- přírůstkový přístup založený na architektuře konsolidovaného datového skladu.

Postupné budování datových tržišť

Princip tohoto přístupu vytvořeným Kimballem spočívá v relativně nezávislém vytváření jednotlivých datových tržišť pro specifické útvary podniku (divize, oddělení,

pobočky, závody). V současnosti byl tento přístup přepracován do tzv. sběrnicové architektury (Kimball, et al., 2002). Rozdíl oproti předchozímu chápání je pouze ve snaze budovat jednotlivá nezávislá datová tržiště integrovaně. Integračním prvkem jsou tzv. sdílené dimenze, tedy dimenzionální tabulky, které jsou opakovaně použity v různých datových tržištích. (Novotný, a další, 2005)

Přístup postupného budování datových tržišť je následující (Novotný, a další, 2005):

1. První datové tržiště je vybudováno na základě analytických potřeb oddělení firmy, pro které je určeno. V rámci databázového návrhu se identifikují potenciální sdílené dimenze a jejich modelování probíhá na základě předpokladu, že budou použity i pro jiná datová tržiště.
2. Další datová tržiště se budují tak, aby maximálně využila již existující dimenze.
3. Ostatní komponenty řešení (ETL prvky, reporty, OLAP kostky a další) se budují v rámci každého tržiště nezávisle na ostatních.
4. Vzhledem k denormalizaci modelu dat datových tržišť do podoby tzv. hvězdy nebo sněhové vločky, nejsou potenciální podobné či shodné ukazatele sdíleny, ale jsou umístěny v datovém tržišti.

Jednorázové vybudování celkového řešení

Tento přístup spočívá v jednorázovém vybudování celkového řešení. Přístup se skládá z následujících hlavních kroků (Novotný, a další, 2005):

1. Celková analýza a dokumentace všech relevantních uživatelských potřeb.
2. Návrh a implementace celkového řešení. Týká se zejména vybudování konsolidovaného datového skladu, pokrývajícího zmapované potřeby uživatelů a tvorby základních závislých datových tržišť.
3. V případě dalších uživatelských potřeb jsou tyto nároky pokryty tvorbou nových datových tržišť.

Přírůstkový přístup

Přírůstkový přístup je, stejně jako předchozí, spojený s architekturou konsolidovaného datového skladu. V rámci tohoto přístupu (Novotný, a další, 2005) je:

1. vytvořena celková koncepce BI řešení podniku. Celková koncepce obsahuje nejen souhrn všech uživatelských požadavků, včetně stanovení jejich priorit z pohledu celé firmy, ale také návrh architektury řešení a hrubý časový harmonogram tvorby celkového řešení. Obsahuje rovněž identifikaci jednotlivých projektů (přírůstků) a jejich návazností (nejen časových, ale i obsahových). V rámci koncepce by neměl chybět způsob financování projektů a algoritmus výpočtu návratnosti investic pro jejich pozdější kontrolu. Celková koncepce tak tvoří jednotný rámec řešení BI ve společnosti.

2. Tato koncepce se následně naplňuje v jednotlivých, časově i finančně omezených krocích (přírůstcích). V rámci každého přírůstku je vybudováno kompletní řešení (podobně jako v případě přístupu nezávislých datových tržišť), které je otevřené a rozšiřitelné v rámci dalších projektů.

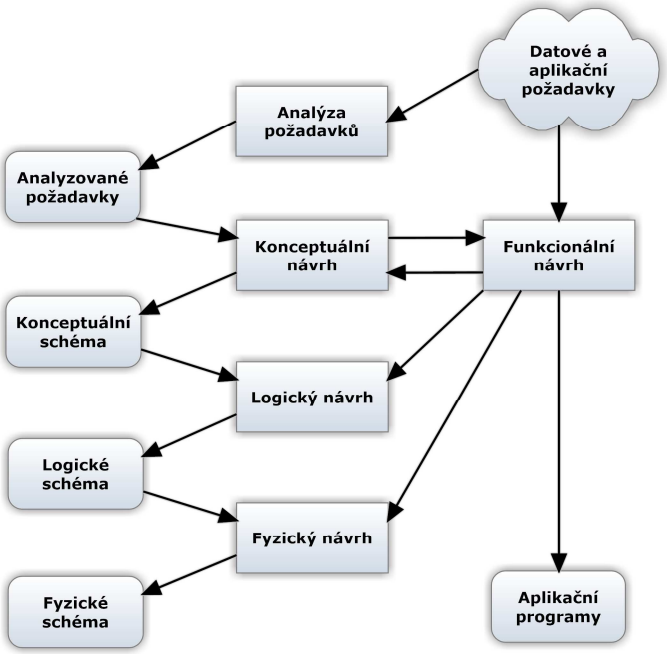
Výsledkem tohoto přístupu je celopodnikový datový sklad se závislými datovými tržišti, stejně jako v předchozím případě. Ovšem problémem tohoto přístupu je nutnost vytvářet v návrhu různá konceptuální schémata (pro každý přírůstek). To nezřídka vede k určitým dodatečným úpravám schémat předchozích přírůstků. Výsledkem, pak může být mnohem složitější návrh. V současnosti je možné tento problém řešit prostřednictvím metodiky tvorby různých verzí schémat popsané v příspěvku (Golfarelli, et al., 2006).

Každý z těchto tří přístupu k řešení BI má svoje výhody a nevýhody. Přístup postupného budování datových tržišť je vhodné použít tam, kde není možné vytvářet integrační činnosti pro tvorbu konsolidovaného datového skladu. Důvodem mohou být technologická, ale i finanční omezení, nebo prostě jen potřeba vybudovat rychlé řešení. Přístup jednorázového vybudování celkového řešení je vhodný v situaci, kde lze zmapovat uživatelské požadavky najednou a zároveň neexistuje velké riziko změn nebo rozšíření těchto požadavků. Tento přístup je vhodný pro celopodnikové analýzy díky konsolidovanému datovému skladu. Přírůstkový přístup má podobné výhody jako přístup jednorázového vybudování, ovšem s možností budovat malá řešení přinášející okamžitý efekt.

1.5. Proces návrhu relační databáze

Vytvoření modelu dat je zapotřebí k tvorbě každé databáze. *Model dat* je množina konceptů, kterou lze použít k popisu struktury a operací na databázi (Navathe, 1992). U většiny databázových řešení je model dat nutné postupně formulovat (obrázek č. 9), to znamená, nejdříve vymezit jeho strukturu obecně. Obecným úkolem návrhu databáze je zmapovat daný reálný svět aplikace do formálního datového modelu daného systému pro správu databází (Fahrner, et al., 1995). *Návrh databáze* je proces, který produkuje sérii databázových schémat pro určitou aplikaci (Mylopoulos, 2009). Mylopoulos (2009) uvádí následující čtyři fáze návrhu databáze: požadavky elicitace, konceptuální návrh schémat, logický návrh schémat a fyzický návrh schémat.

Tato rešeršní práce se bude přidržovat procesu návrhu databáze podle Fahrnera, který ve své interpretaci odlišuje mezi pojmy schéma a návrh. Obrázek č. 10 znázorňuje proces databázového návrhu podle Fahrnera (1995).



Obrázek č. 10 - Proces databázového návrhu (Fahrner, et al., 1995).

1.5.1. Konceptuální návrh

V současnosti je konceptuální schéma nejčastěji znázorňováno prostřednictvím ER-modelu. Tento model a jeho notace byly publikovány Chenem v roce 1976 (Chen, 1976). Od té doby vznikaly různé alternativní notace (Martinova notace, Bachmanova notace (1969), IDEF1X a další) a modely – nejznámější je EER model (v současnosti

velmi často používán). Smysl samotného konceptuálního návrhu se začal postupně přesouvat do pozadí a novým cílem tvůrců notací a modelů se stala schopnost zaznamenat co nejvíce různých typů vztahů, typů atributů, hierarchií a jejich vzájemnou korelaci (EER model, AER model a další), (Tyrychtr, a další, 2010). V článku (Genero, et al., 2007) je navržen soubor metrik pro měření strukturálních vlastností ER diagramů pro predikci vnější kvality modelu. Článek (Artale, et al., 2007) poukazuje na to, že přítomnost vztahu hierarchie je hlavním zdrojem složitosti modelu. Všechny tyto poznatky jsou důvodem pro používání srozumitelných notací pro tvorbu konceptuálních modelů, bez přídavných prvků, které srozumitelnost modelu snižují.

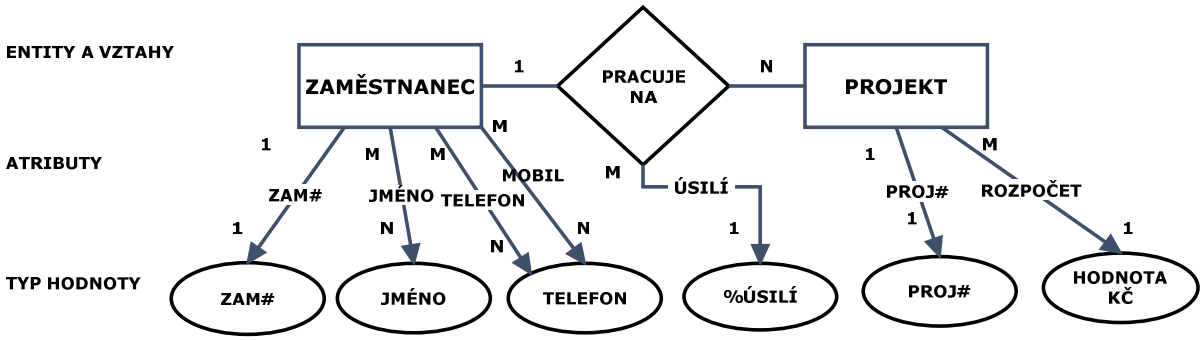
ER model

ER model je možné použít jako základ pro sjednocení odlišných pohledů na data: síťový model, relační model, a model množin entit. (Chen, 1976) Pro grafickou reprezentaci ER modelu je používán ER diagram. Technika tvorby *ER diagramů* představuje grafický způsob zobrazování typů entit, typů vztahů a atributů. (Chen, 2002)

Formální definice konceptu entit a množin je (Chen, 2002):

ENTITA	e
MNOŽINA ENTIT	$E; e \in E$
HODNOTA	v
MNOŽINA HODNOT	$V; v \in V$
VZTAH	r
MNOŽINA VZTAHŮ	$R; r \in R$

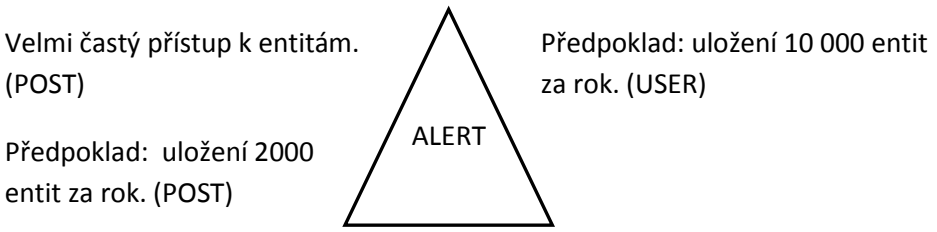
Obrázek č. 11 zobrazuje ER diagram dvou entit. Typy entit ZAMĚSTNANEC a PROJEKT jsou reprezentovány obdélníkem, typy vztahů jako PRACUJE NA jsou reprezentovány kosočtvercem. Množina hodnot (doména) jako ZAM#, JMÉNO, a TELEFON jsou reprezentovány kružnicí, zatímco atributy jsou mapovány z entit a vztahů na množinu hodnot. (Chen, 2002) Kardinalita vztahu mezi relacemi je v diagramu znázorněna u spojovací čáry, například "1" a "N".



Obrázek č. 11 - ER diagram podle (Chen, 2002).

ER-A model

Autorem práce byl v článku (Tyrychtr, a další, 2010) představen inovovaný Chenův ER model. Tento model využívá klasickou Chenovu notaci, modifikovanou o množinu upozornění (ES-A) [Entity Set of Alert]. ES-A přidává do modelu důležitou informaci o entitách, která by jinak pro členy vývojového týmu byla nedostupná nebo podléhala zkreslení (vlastní představy o entitách). ES-A je graficky vyjádřena trojúhelníkem. Hodnoty atributů jsou napsány po stranách trojúhelníku. Názvy atributů jsou vyjádřeny v závorkách, které jsou připsány ke každé hodnotě. (Tyrychtr, a další, 2010)



Obrázek č. 12 - Množina upozornění (podle Tyrychtr, a další, 2010)

Navržená inovace konceptuálního modelu dat má za cíl zlepšit kvalitu vytvářených databázových projektů a to způsobem, který eliminuje komunikační šum v rámci přenosu informací mezi jednotlivými členy vývojového týmu. Konceptuální datový model vyjádřený pomocí ER diagramu s množinou upozornění se nazývá Entity-Relationship Alert (ER-A) model.

1.5.2. Logický návrh

Návrh logického schématu často zahrnuje normalizační kroky, kde je počáteční logické schéma s asociovanými funkčními závislostmi transformováno v normalizované schéma za použití normálních forem. Funkcionální závislosti představil Codd (1970). Ovšem

Armstrong (1974) byl první, který axiomatizoval funkční závislosti. V současnosti jsou pravidla rozšířena do funkcionální a vícehodnotové podoby závislostí podle Beeriho (1978).

Normální forma definuje stav závislostí množiny dat, nebo sémantické omezení, které bylo zadáno jako součást databázového schématu. Tyto podmínky se používají ke kontrole, zda návrh databáze má žádoucí vlastnosti (například databáze neukládá redundantní informace). (Arenas, 2009) Normalizační přístup byl navržen na začátku 70. let Coddem (Codd, 1972(a)). Od té doby se mnohé studie zabývaly problémem normalizace relačních databází a dalších datových modelů. V současnosti je možné najít normální formy (NF) pro různé typy datových závislostí: 3NF (Codd, 1972(a)), BCNF (Codd, 1974) pro funkční závislosti, 4NF (Fagin, 1977) s více závislostmi, PJ / NF (Fagin, 1979), 5NFR (Vincent, 1997) pro spojené závislosti a DK / NF (Fagin, 1981) pro obecná omezení. Tyto normální formy, spolu s normalizačními algoritmy pro převod špatně navržené databáze na „správně“ navrženou databázi, lze nalézt v řadě publikací. První tři normální formy lze jednoduše popsat následovně (Conolly, a další, 2009):

1NF: Tabulka, v níž každý průsečík sloupce a záznamu obsahuje jen jedinou hodnotu.

2NF: Tabulka, která je v 1NF a ve které jsou hodnoty každého sloupce, který není součástí primárního klíče, determinovány všemi hodnotami sloupců, které tvoří primární klíč.

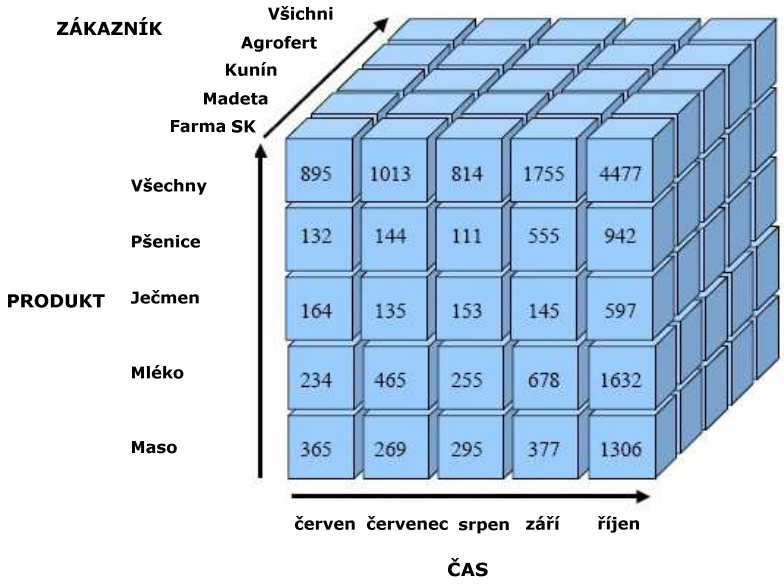
3NF: Tabulka, která již je v 1NF a 2NF a ve které všechny hodnoty ve sloupcích které nepatří k primárnímu klíči, jsou determinovány pouze sloupci primárního klíče a nejsou determinovány žádnými jinými sloupci.

1.5.3. Fyzický návrh

Fyzická fáze návrhu je koncepčně odlišná, protože se zaměřuje na výkonová hlediska (Storey, et al., 1995). Batini (1986) vymezuje *fyzický návrh* jako mapování logického schématu databáze do příslušného úložiště v databázovém systému, včetně nových fyzických parametrů k optimalizaci výkonu databáze pro práci s transakcemi.

1.6. Proces návrhu analytické databáze

Multidimenzionální modelování je proces modelování dat pomocí modelovacích konstrukcí zajišťující vícerozměrný model dat. Multidimenzionální modely kategorizují data, buď jako fakta asociovaná s numerickou mírou, nebo jako dimenze, která charakterizují fakta a jsou většinou textová (obrázek č. 13). *Fakta* jsou objekty, které představují předmět požadované analýzy, který má být analyzován pro lepší pochopení jeho chování. (Pedersen, 2009(c)) Multidimenzionální modely v současné době nejčastěji vycházejí z relačního modelu dat, popřípadě jsou založeny na multidimenzionální datové kostce. (Zádová, 2009)



Obrázek č. 13 - Příklad datové kostky (vlastní zpracování).

Multidimenzionální model dat vycházející z relačního modelu odlišuje dva základní typy relací, které se nazývají tabulky dimenzí a tabulky faktů. Oba typy tabulek jsou databázové relace s určitými specifiky, které zohledňují cíl, pro který jsou určeny. Mohou vytvářet hvězdicové struktury [star schema], různé formy sněhových vloček [snowflake schema] a souhvězdí [constellation schema]. (Zádová, 2009) Problematika výběru vhodné struktury je řešena v příspěvku Leveneho, et al., (2003).

Multidimenzionální model dat představující složku BI systému je nutné zavádět v souladu s návrhem BI systému, Závodný (2011) uvádí následující fáze návrhu BI systémů:

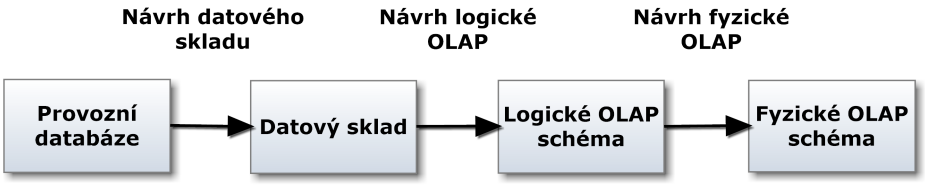
1. Analýza informačních potřeb uživatelů.

2. Analýza datové základny.
3. Návrh řešení a jeho architektury

Multidimenzionální modelování vyžaduje specializované návrhové metody. Ačkoli bylo napsáno mnoho o tom, jak by měl být navržen datový sklad, neexistuje dosud jednotný názor na metody návrhu. Multidimenzionální modelování bylo nejprve představeno Kimballem (1996). Kimballův přístup byl dobře přijat v tomto odvětví a zároveň představil první metodu pro odvození logického schématu datového skladu (Romero, et al., 2011). Kimball (1998), (2002) doporučuje čtyři kroky v procesu multidimenzionálního modelování:

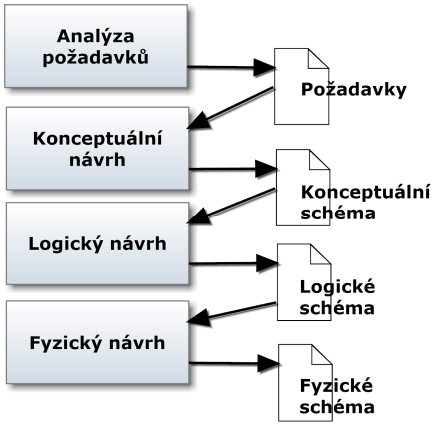
1. Výběr podnikových procesů.
2. Výběr části z podnikového procesu.
3. Výběr rozměrů (dimenzí).
4. Výběr měr.

Jiný přístup ukazuje obrázek č. 14. Jedná se o proces návrhu OLAP a datového skladu podle (Niemi, et al., 2003).



Obrázek č. 14 - Proces návrhu OLAP databáze (podle Niemi, et al., 2003).

Je zřejmé, že multidimenzionální modelování vyžaduje specializované návrhové metody. V příspěvku (Rizzi, et al., 2006) vyplývajícího ze široké diskuze, byla představena metoda návrhu datových skladů, která je zobrazena na obrázku č. 15.



Obrázek č. 15 - Základní fáze návrhu datového skladu (podle Rizzi, et al., 2006).

Autor disertační práce považuje tento přístup za podnětný, ovšem ne úplný. Příspěvek (Rizzi, et al., 2006) předkládá spíše otázky než odpovědi. Sice je další část textu (v rámci kapitoly 1.6) strukturovaná podle dílčích fází tohoto návrhu datového skladu (podle Rizziho), ale obsahově s tímto příspěvkem explicitně nesouvisí.

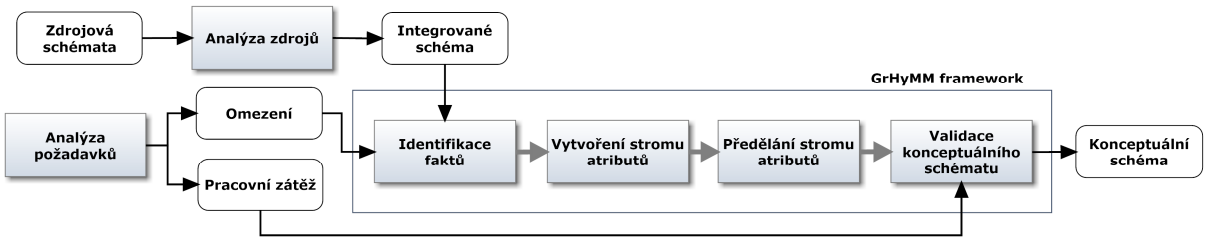
1.6.1. Analýza požadavků

Podobně jako u modelování tradičních informačních systémů je i Kimballova metoda řízena požadavky (tzv. *requirement-driven*). Začíná stanovením obchodních požadavků organizace a postupně krok za krokem dochází k odvození multidimenzionálního schématu. (Romero, et al., 2011)

Podle (Winter, et al., 2003) je možné metody analýzy požadavků zařadit do *demand-driven*, *supply-driven* nebo *hybridního* rámce:

- *Supply-driven* přístup (SDA): Známý jako *data-driven*, vychází z podrobné analýzy datových zdrojů pro stanovení multidimenzionálních konceptů v procesu reengineeringu.
- *Demand-driven* (DDA): Známý jako *requirement-driven* nebo *goal-driven*, se zaměřuje na stanovení uživatelských multidimenzionálních požadavků (podobně jako u návrhu informačních systémů) a až pak je mapován do datových zdrojů.
- *Hybridní přístupy*: Kombinuje oba vzory pro návrh DS z datových zdrojů, a to tak, že bere na zřetel požadavky koncových uživatelů.

Poslední přístup je v současnosti velmi podporován v příspěvcích (Tria, et al., 2011), (Tria, et al., 2012). Autoři těchto příspěvků představili konceptuální model, který je založen na grafově-orientované reprezentaci datových zdrojů. Jejich GrHyMM [Graph-Oriented Hybrid Multidimensional Model] framework je ilustrován na obrázku č. 16.



Obrázek č. 16 - Framework pro návrh datového skladu (podle Tria, et al., 2011).

Činnosti GrHyMM frameworku jsou prováděny postupně takto:

1. *Analýza požadavků*: V tomto kroku jsou zkoumány potřeby koncových uživatelů. Za tímto účelem je použit i* framework (Mazón, et al., 2011), který umožňuje přímo vyjádřit podnikové cíle s odkazem na několik aktérů systému.

2. *Analýza zdrojů*: V tomto kroku je třeba analyzovat různá schémata zdrojů dat a následně je sladit s cílem získat globální a integrované schéma.

3. *Konceptuální návrh*: Tento krok je založen na GrHyMM a grafově-orientovaném multidimenzionálním modelu.

4. *Logický návrh*:V tomto kroku je konceptuální schéma přeměněno na logické schéma v závislosti na modelu dat systému cílové databáze.

5. *Fyzický návrh*: Proces návrhu končí implementací logického schématu definováním fyzických vlastností databáze na základě specifických rysů, které poskytuje databázový systém, jako je indexování, dělení [partitioning], a tak dále.

1.6.2. Konceptuální návrh

Pro zobrazení konceptuálního schématu multidimenzionálních modelů je možné použít klasické přístupy jako u relačního modelu. Ovšem z důvodu hierarchie multidimenzionálních dat je vhodné použít sofistikovanější modely. Elmasri a Navathe (1994) byli první, popsali konceptuální modelování operačních a analytických dat. Následně vznikalo mnoho návrhů pro tvorbu konceptuálních schémat multidimenzionálních modelů. Ucelený přehled je uveden v tabulce č. 1.

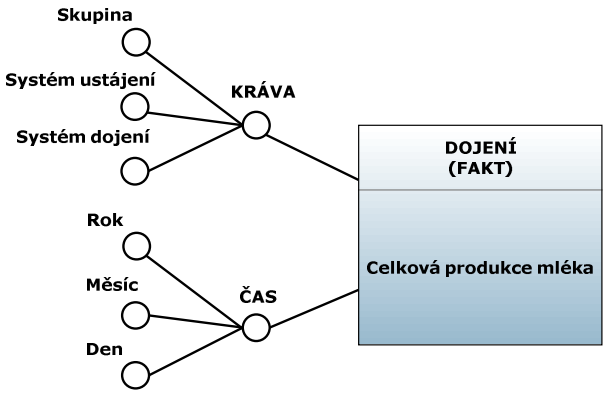
Tabulka č. 1: Přehled konceptuálních přístupů (vlastní zpracování).

	Konceptuální návrh	Logický návrh
ER	(Elmasri, et al., 1994), (McGuff, 1998)	(Ballard, et al., 1998)
EVER	(Bækgaard, 1999)	
StarER	(Tryfona, et al., 1999)	
mER	(Sapia, et al., 1998), (Hahn, et al., 2000)	
DWCDM	(Franconi, et al., 1999)	
MAC	(Tsois, et al., 2001)	
Star	(Wu, et al., 1997),	(Chaudhuri, et al., 1997), (Ballard, et al., 1998), (McGuff, 1998), (Boehnlein, et al., 1999)
Snowflake		(Chaudhuri, et al., 1997), (Ballard, et al., 1998), (Boehnlein, et al., 1999)
DFM	(Golfarelli, et al., 1998)	
MDS	(Thomsen, 2002)	
MultiDimER	(Malinowski, et al., 2006)	(Malinowski, et al., 2006)
UML	(Prat, et al., 2006)	(Prat, et al., 2006)

V další části textu se autor disertační práce omezuje pouze na přehled některých modelů.

DFM (Dimensional Fact Model)

DFM představil v příspěvku (Golfarelli, et al., 1998) a slouží jako podpora tvorby multidimenzionálních modelů prostřednictvím schéma faktu. Na obrázku č. 17 ilustrován příklad DFM. Konečné uzly představují atributy a otcovské uzly dimenze.

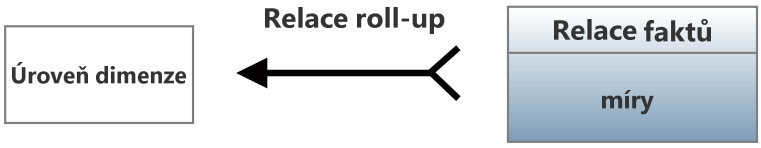


Obrázek č. 17 - DFM model (vlastní zpracování).

mER model

mER model představil Sapia (1998) rozšířením ER modelu pro muldidimenzionální paradigma. Tento model je se svojí srozumitelnou notací stále používaný i v současných projektech. Elementy mER modelu jsou (obrázek č. 18):

- Úroveň dimenzí [dimension level] (agregovaná úroveň kvalifikovaných dat).
- Roll-up vztah (agregace relací mezi úrovněmi).
- Vztah faktů [fact relationship] (*n*-ární relace mezi úrovní dimenze, jejíž atributy modelu kvantifikují data jako míry faktů).

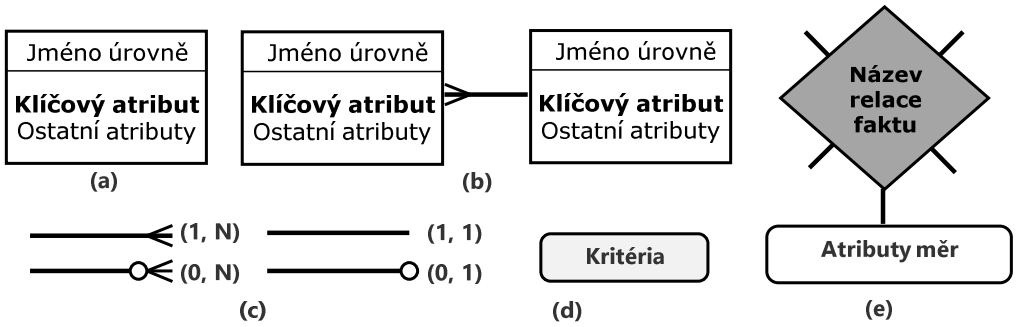


Obrázek č. 18 - Notace elementů mER modelu (podle Sapia, 1998).

MultiDimER model

MultiDimER (Malinowski, et al., 2006) je založený na modelu ER. Malinowski navrhl několik úprav pro generalizované a nestriktní hierarchie. Hierarchie reprezentovány v modelu MultiDimER používají notaci, která umožňuje jednoznačně rozlišovat každý typ hierarchie s ohledem na jejich rozdíly ve schématu, tak úrovní instance. Nicméně, i když je zobrazení logické úrovně založeno na dobře známých pravidlech, není možné pomocí relačního modelu vyjádřit sémantiku každé hierarchie: obecných, hromadných nebo najednou závislých hierarchií. Prostřednictvím MultiDimER modelu, je možné zachytit lepší význam dat pro datové sklady nebo OLAP systémy. Konceptuální

grafický model pro reprezentaci faktů, měr a dimenzí, včetně různých druhů hierarchie umožňuje návrhářům lépe porozumět datovému modelu.

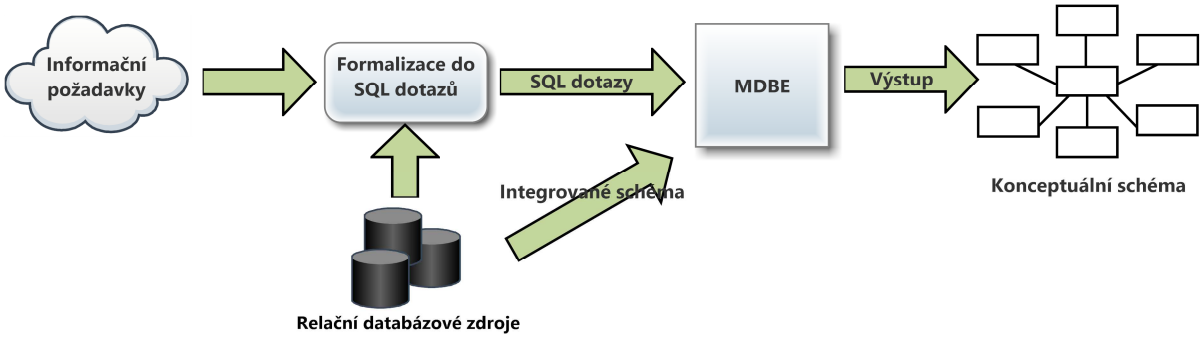


Obrázek č. 19 - Notace pro MultiDimER model (podle Malinowski, et al., 2006).

Notace pro multidimenzionální model je znázorněna na obrázku č. 19: (a) představuje úroveň, (b) hierarchii, (c) kardinalitu, (d) kritéria analýzy, a (e) relaci faktů.

Další přístupy k multidimenzionálnímu modelování

Inovovaný přístup k tvorbě multidimenzionálních konceptuálních schémat z relačních zdrojů představuje Romero (2010) tak jak je znázorněno na obrázku č. 20. Tato metoda MDBE [Multidimensional Design By Examples] využívá hybridní přístup pro automatické generování multidimenzionálních schémat od koncových uživatelů a relačních datových zdrojů. (Romero, et al., 2010)

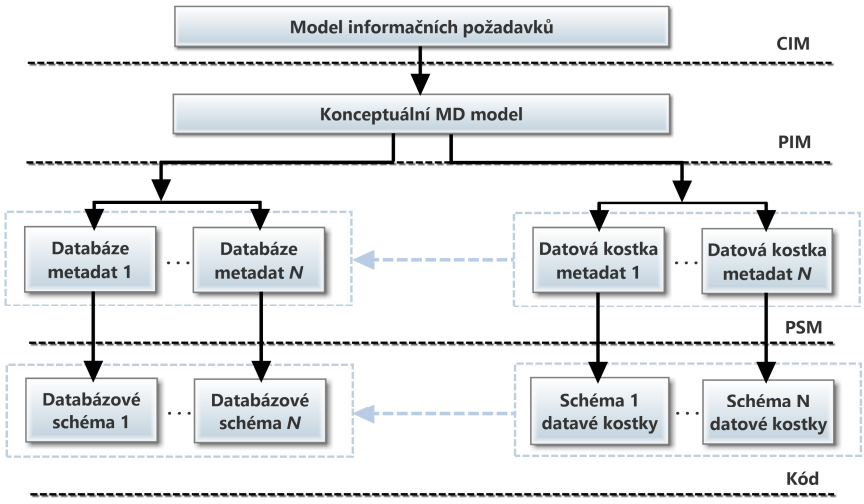


Obrázek č. 20 - Metoda MDBE (podle Romero, et al., 2010).

Je všeobecně přijímána teze, že vývoj datových skladů musí být řízen definicí konceptuálních vícerozměrných modelů dat. Technologická nezávislost na konceptuálním modelu je základem pro implementaci databázového schématu datového skladu pomocí logického modelu na míru jedné konkrétní technologii. Většina

současných výzkumů se zaměřuje na odvození (automatické) různých druhů databázových schémat z těchto konceptuálních modelů.

Nicméně, tyto snahy překvapivě zanedbávají automatické odvození metadat OLAP. Integrovaný způsob řízení metadat je nesmírně důležitý, neboť umožňuje koncovému uživateli správně vyhledávat v databázi schéma datového skladu a usnadňovat jeho řízení. Pardillo (2011) navrhuje model transformace architektury, který umožňuje získání jak schématu databáze tak požadovaných metadat. (Pardillo, et al., 2011)



Obrázek č. 21 - MDA pro návrh multidimenzionálního modelu dat (podle Pardillo, 2011).

1.6.3. Logický návrh

V kapitole 1.3.6 jsou představeny úložiště multidimenzionálních dat. V případě realizace uložení multidimenzionálních dat v relační databázi je vhodným řešením využití databázových schémat hvězdy nebo sněhové vločky. Princip takového modelování je založen na rozlišení dvou základních tabulek. V relaci k OLAP kostkám (Novotný, a další, 2005):

- buňky kostky, tj. jednotlivé ukazatele a jejich souřadnice reprezentuje *tabulka faktů* [fact table];
- hrany kostky, tj. dimenze, jejich prvky a další atributy reprezentují *tabulky dimenzí*.

K dispozici jsou 3 typy schémat:

Schéma hvězda [star schema]

Schéma hvězda představuje tabulku faktu umístěnou uprostřed a obklopenou tabulkami dimenzí. Tabulky dimenzí denormalizované a tabulka faktu je normalizovaná. Toto schéma může být tvořeno dvěma nebo více tabulkami faktu a dvěma nebo více tabulkami dimenzí, které se vztahují k cizímu klíči.

Schéma sněhová vločka [snowflake schema]

Schéma sněhové vločky představuje denormalizované tabulky dimenzí, které lze rozdělit do dvou nebo více normalizovaných dimenzí. Tabulky faktů i dimenzí jsou pak normalizované.

Schéma konstelace (nebo také galaxie/integrované/hybridní schéma) [constellation schema]

Hybridní schéma je tabulka dimenzí sdílená dvěma nebo více tabulkami faktů.

Ačkoli je obecně známé, že návrh multidimenzionálních modelů a schémat (hvězdy a sněhové vločky) představuje netriviální problém, přesto existují metody pro odvození takového schématu z provozní databáze (Chen, et al., 2007), (Romero, et al., 2007b). Obecně jsou MD modely oproti relačním modelům v nenormalizovaném tvaru (provádí se tzv. *denormalizace*). Ovšem v příspěvku Lechtenbörgera, et. al., (2003) jsou definovány tzv. *multidimenzionální normální formy* umožňující úvahy o kvalitě konceptuálních schémat podle pevně stanovených pravidel.

První tři multidimenzionální normální formy (MNF) lze neformálně popsat následovně:

1MNF: Cílem 1MNF je zajistit, aby funkcionální závislosti vyplývající ze schématu skutečně odrážely funkční závislosti v rámci aplikace domény.

2MNF: Pokud existuje schéma v 1MNF, pak další podmínkou pro 2MNF je syntaktická kontrola týkající se volitelných úrovní a souvisejících závislostí.

3MNF: Pokud existuje schéma ve 2MNF, pak další podmínkou pro 3MNF je, že každá úroveň dimenze (povinná, či nikoliv) obsahující platné souvislosti, by měla odpovídat vybraným atributům pro nějakou podtřídu shluku.

1.6.4. Fyzický návrh

V tomto vývojovém procesu je důležité udělat rozhodnutí o implementaci multidimenzionálního modelu dat. Ta se prakticky realizují prostřednictvím následujících designových vzorů:

MOLAP [Multidimensional OLAP]

MOLAP systémy ukládají data ve specializovaných multidimenzionálních strukturách. Nejjednodušeji jsou implementovány prostřednictvím vícerozměrných polí pro uložení datové kostky. MOLAP implementace není závislá na relační databázi. (Kimball, et al., 2011) Obecně platí, že MOLAP systémy umožňují rychlejší odezvy na dotazy a prostorově úsporné ukládání, zatímco ROLAP systémy jsou flexibilnější, pokud jde o redefinici datové kostky a zajištění lepší podpory pro časté aktualizace (Jensen, et al., 2010). Naopak (Fišer, et al., 2004) uvádí, že existují dva hlavní problémy pro použití této metody. Za prvé, na rozdíl od Jensena, Fišer uvádí, že MOLAP vytváří obrovské velikosti datové kostky. Ve svém příspěvku Fišer (2004) toto tvrzení ukazuje na příkladu s pěti dimenzemi a osmi bitovém rozlišení pro každou dimenzi. Dále předpokládá, že každá buňka datové kostky uloží pouze jeden bajt dat. Pak tedy je možné vypočítat: $256\ pozic^{5\ dimenzí} = 1\ Tbyte \cong 10^{12}\ bytů$.

Dalším problémem, který Fišer (2004) uvádí je, že paměť, bez ohledu na to, jestli se jedná o hlavní paměť počítače nebo pevný disk, umožňuje obecně jednorozměrné ukládání. Proto je nutné k uložení jednorozměrné projekce multidimenzionální kostky použít různé indexovací techniky.

ROLAP [Relational OLAP]

Ve velké míře jsou v současnosti relační DBS používány pro aplikace OLAP. Přístup *ROLAP* umožňuje ukládat multidimenzionální data v relační databázi. Na konceptuální úrovni je pak multidimenzionální pohled všeobecně přijímán jako standardní model dat pro OLAP aplikace (Ramsak, et al., 2001). ROLAP aplikace jsou nejčastěji vyzdvihovány pro svoji flexibilitu ve vytváření ad-hoc dotazů a pro svoji schopnost pracovat s OLTP databází, namísto vytváření datových skladů. Jejich nevýhodou je účinnost aplikace, tedy vzrůstající doba odezvy, která je závislá na uloženém množství dat v OLTP databázi. (Fišer, et al., 2004) Tento problém je v současnosti řešen speciálními indexy, jako jsou UB-stromy (Ramsak, et al., 2001) pro velké agregované

dotazy. Podrobný přehled indexů a metod je možné najít v příspěvku (Morfonios, et al., 2007).

HOLAP [Hybrid OLAP]

Hybridní OLAP kombinuje vlastnosti ROLAP a MOLAP. Využívá vynikající zpracování MOLAP se schopností ROLAP pracovat s větším objemem dat. HOLAP ukládá data jak v relační databázi, tak v multidimenzionální databázi. V HOLAP jsou data agregována pomocí MOLAP strategie, zatímco zdrojová data, která jsou determinována objemem, jsou uložena pomocí ROLAP strategie. Tato konfigurace umožňuje velmi rychlé zpracování a minimalizaci požadavků na ukládání dat. (Khan, 2005)

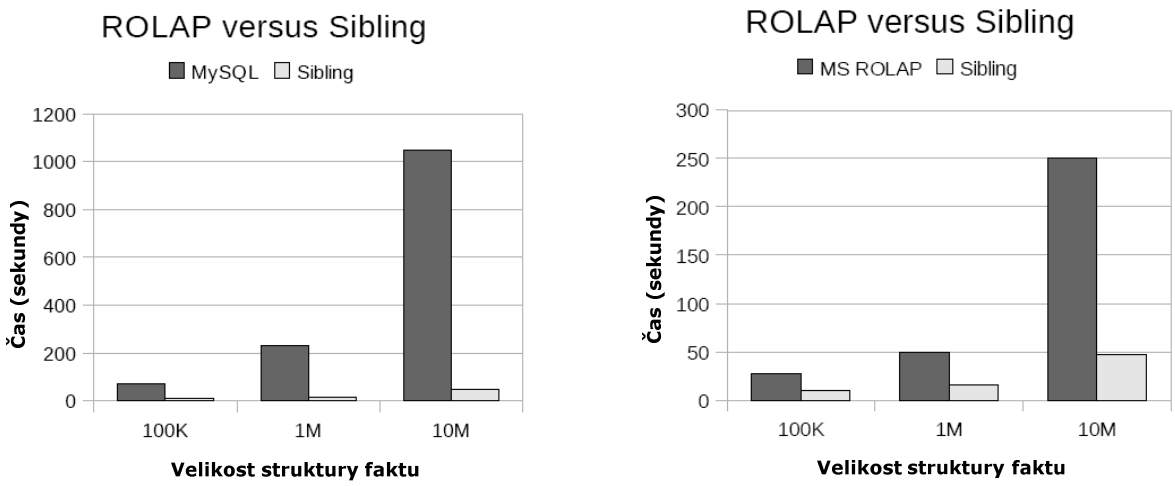
DOLAP [Desktop OLAP]

DOLAP umožňuje připojit se k centrálnímu úložišti OLAP dat a stáhnout si potřebnou podmnožinu kostky na lokální počítač. Veškeré analytické operace jsou prováděny nad touto lokální kostkou. (Novotný, a další, 2005)

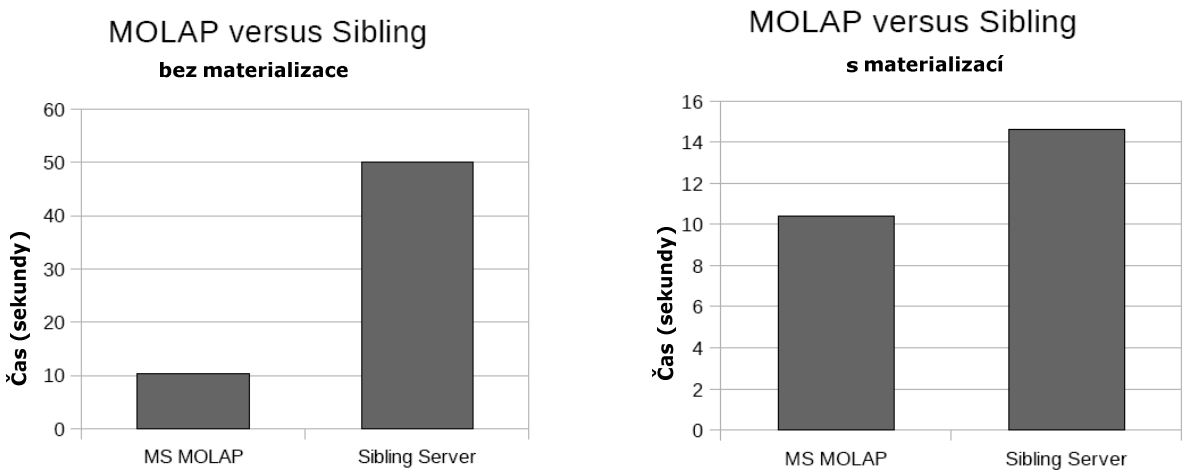
V současnosti se otázka fyzického návrhu OLAP omezuje na to, zda zvolit ROLAP nebo MOLAP. MOLAP systémy skutečně poskytují působivý výkon analytických dotazů, ale mají tendenci mít omezenou škálovatelnost. Naopak, ROLAP jako tabulkově orientovaný model má dobrou škálovatelnost, ale nabízí průměrný výkon oproti MOLAP systémům.

Pro zpracování multidimenzionálních dat jsou používány speciální indexovací techniky, jako je bitmapový index a různé varianty stromů. Ačkoli mnoho vhodných datových struktur bylo vyvinuto během posledních 20 let, jen některé z nich, například UB-stromy a M-stromy, jsou integrovány do komerčních DBMS. (Pokorný, 2006)

Problematika účinnosti OLAP je v současnosti velmi inovativním způsobem řešena v práci (Eavis, et al., 2012). Eavis v tomto dokumentu výrazně rozšířil původní model, který zahrnuje jak R-strom, tak bitmapové indexování. Konkrétně jsou integrovány open source knihovny Berkeley DB na server tak, že jsou zapouzdřeny oba indexy a datová kostka v jednom úložišti dat. Dimenze datové kostky a jejich hierarchie jsou také efektivně uloženy v Berkeley DB. Ne-hierarchické atributy jsou uloženy jako soubor FastBit bitmapových indexů. Jeho výkonnost oproti ROLAP je zobrazena na obrázku č. 22 a výkonnost oproti MOLAP na obrázku č. 23.



Obrázek č. 22 - Sibling Server versus MySQL / MS Analysis Services (ROLAP), (podle Eavis, et al., 2012).



Obrázek č. 23 - MOLAP versus ne-materializovaný Sibling / MOLAP versus materializovaný Sibling, (podle Eavis, et al., 2012).

Takováto integrovaná architektura představuje velmi efektivní OLAP, který poskytuje vysoký výkon, který by se dal očekávat od systémů MOLAP, se škálovatelností typicky spojenou s relačně orientovanými servery. Tento paralelní přístup je v příspěvku (Eavis, et al., 2012) uveden pod názvem "Sibling Server".

1.7. Modelování dat v zemědělském podniku

Výběr dat, tvorba souvisejících modelů dat a příslušné struktury představují důležitý a v současnosti stále málo akcentovaný proces tvorby IS. Stanovení potřebných dat vychází v podstatě ze dvou přístupů (Duchoň, 2007): inženýrského a statistického. Inženýrský přístup je založen na znalosti konkrétního podnikového systému oproti tomu

statistická analýza využívá data více podniků. Inženýrský přístup je právě vhodný tam, kde má být zkoumána vlastní podniková produkční funkce, jelikož jsou data velmi dobře srozumitelná. Nevýhodou je, že výchozí data mohou popisovat pouze jeden aspekt technologického procesu.

Využívání informačních a komunikačních technologií a zemědělských manažerských informačních systémů pro podporu rozhodování představuje velký příslib pro dosažení lepšího stavu zemědělského podniku čelícímu tlaku na snižování marží ze zisku. (Nikkila, et al., 2010) Nové koncepce návrhů a řízení informací znamená, že zemědělci musí být připraveni přijmout nové pracovní návyky a možná také další účast ve vzdělávání. (Sörensen, et al., 2011) Murakami, et al. (2007) zjistili, že nejdůležitější požadavky na zemědělské manažerské informační systémy zahrnují:

- návrh zaměřený na specifické potřeby zemědělců,
- jednoduché uživatelské rozhraní,
- automatizace a snadno dostupné metody pro zpracování dat,
- uživatelské rozhraní umožňující počítačový přístup k procesním a analytickým funkcím,
- integrace,
- lepší integraci standardizovaných počítačových systémů,
- užší integrace a interoperabilita,
- škálovatelnost,
- schopnost výměny dat mezi aplikacemi a
- nízké náklady.

Charvát, et al. (2011) doporučuje pro budoucnost farmy využít servisně orientovanou architekturu (SOA), která poskytuje metody vývoje a integrace systémů, kde jsou systémy funkcionálně seskupeny kolem podnikových procesů jako balík interoperabilních služeb.

Kavka, et al. (2011) vyvinuli informační systém zemědělských výrobních technologií (ISZVT) pro podporu rozhodovacího procesu a poradenství na úrovni zemědělských podniků. Obsah databáze se skládá ze standardních pěstebních ukazatelů, technologií chovu zvířat a standardních výpočtů práce, energie, nákladů na výrobu a prodej v hotovosti. (Kavka, et al., 2011) ISZVT je především vhodné použít jako provozní

systém farmy, jelikož je založen na relační databázi pro OLTP. Nevýhodou je, že neposkytuje multidimenzionální pohled na data a jejich kategorizaci v čase.

1.7.1. Formulace výrobního procesu v zemědělství

Model produkční funkce může v zemědělském podniku sloužit jako nástroj ekonomické analýzy. Podle Duchoně (2007) *produkční funkce* vyjadřuje závislost mezi vytvořenou produkcí a základními výrobními faktory. Nejčastěji je v odborné literatuře využívána Cobb-Douglesova produkční funkce, kterou lze charakterizovat konstantní elasticitou výrobních faktorů, konstantní elasticitou substituce výrobních faktorů, neměnností výnosů z rozsahu mezi jednotlivými podniky ve zkoumaném souboru a konvexností izokvantové funkce směrem k počátku. (Kroupová, 2010) Cobb-Douglesova produkční funkce má následující obecný tvar (Filipe, et al., 2005), (Kroupová, 2010):

$y = \alpha x_l^{\beta_j} x_p^{\beta_p} x_k^{\beta_k}$, kde

y… množství výstupu,

$x_{l,p,k}$ … množství *l*-tého, *p*-tého a *k*-tého vstupu,

α, β … parametry produkční funkce.

Uvedenou mocninou funkci lze jednoduchou logaritmickou transformací převézt do log-lineárního tvaru. Alternativně je používána translogaritmická produkční funkce. Na rozdíl od Cobb-Douglasovy produkční funkce umožňuje translogaritmická funkční forma přímé testování vzájemných vztahů mezi výrobními faktory a jejich sdruženého vlivu. (Kroupová, 2010) Funkční předpis translogaritmické funkce je následující:

$$\log y = \alpha + \sum_{j=1}^{J} \beta_j \log x_j + \frac{1}{2} \sum_{j=1}^{J} \sum_{m=1}^{M} \gamma_{jm} \log x_j \log x_m, \text{ kde}$$

y… množství produkce,

x_j … množství jednotlivých výrobních faktorů,

α … konstanta,

β_j… parametry vyjadřující vliv jednotlivých vstupů,

γ_{jm}… parametry skupinových agregátů vstupů,

$j = 1,2, \dots J, m = 1,2, \dots M$.

Pro modelování produkčních funkcí a stanovení jejích parametrů lze užít inženýrského i statistického přístupu s využitím průřezových dat nebo časových řad. Formulovat výrobní proces je možné také prostřednictvím strukturního modelu. Charvat, et al. (2011) představili nový matematický model pro optimální výrobu a využití půdy pro další období. Účelem je maximalizovat očekávaný zisk zemědělců prostřednictvím lineárního programovacího modelu. Tento maximalizační model je definován následující formou (Charvat, et al., 2011):

$z =$
$\sum_{i=1}^{m}\sum_{j=1}^{n}\left[a_j\left(h_{ij}p_i + s_{ij} - c_{ij}\right)x_{ij} + a_j\left(h_{ij}^{VA}p_i + s_{ij}^{VA} - c_{ij}^{VA}\right)x_{ij}^{VA}\right] - \sum_{i=1}^{m} f_i^{prod}y_i -$
$(VA)C^{VA} - \sum_{j=1}^{n}\gamma_j^{VA}f_j^{landVA}$, kde

m - počet rostlinných druhů, z nichž zemědělec vybírá výrobní kombinace,

n - počet zemědělských pozemků,

a_j - plocha půdy j (v hektarech),

c_{ij} - náklady na jeden hektar, jestliže výrobek i pěstované na půdě j,

C^{VA} - variabilní použití fixních nákladů,

f_i^{prod} - produkt j fixních nákladů,

f_j^{landVA} - fixní náklady na variabilní použití na půdě j,

s_{ij} - dotace na jeden hektar, jestliže výrobek i je pěstován na půdě j,

h_{ij} - očekávaný výnos produktu, jestliže výrobek je pěstován na pozemku j,

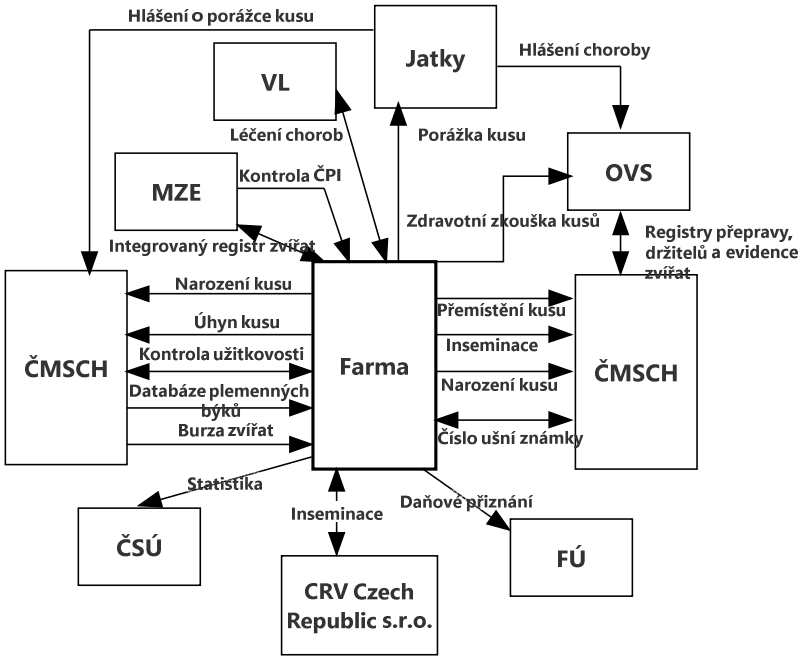
p_i - očekávaná prodejní cena jedné tuny výrobků i.

1.7.2. Informační toky v zemědělském podniku

Proces databázového návrhu začíná analýzou požadavků, která odráží datové a aplikační požadavky. V této fázi je vhodné identifikovat a charakterizovat informační toky farmy. Tato fáze je velmi důležitá pro další procesní vývoj databáze. Opomenutí identifikace relevantních dat se promítne do konceptuálního, logického a v konečné fázi do fyzického návrhu databáze, která se pak stává afunkční pro další vývoj OLAP řešení nebo jiných systému pro podporu rozhodování.

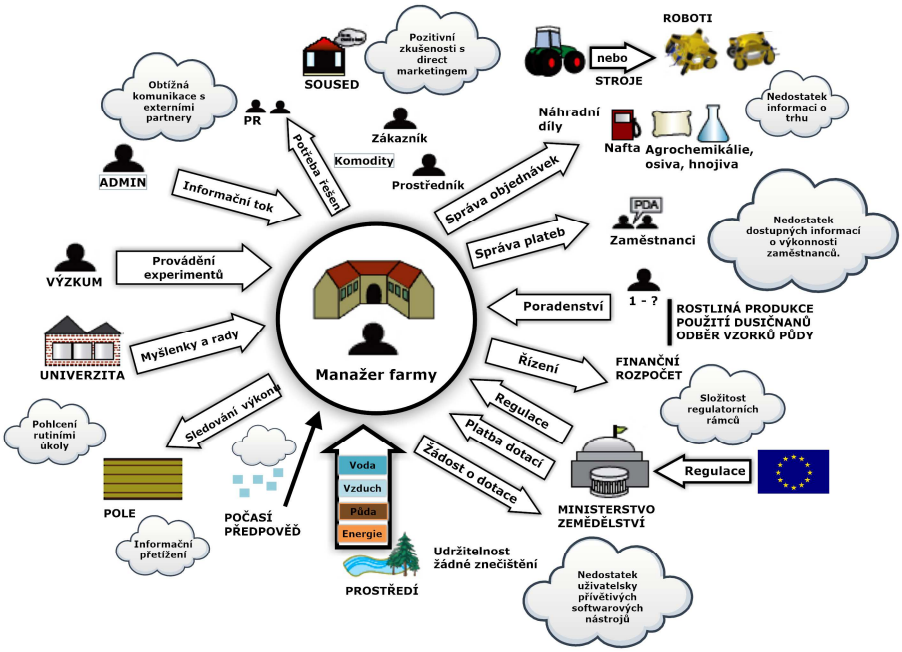
Ulman (2009) ve svém příspěvku identifikoval a charakterizoval základní informační toky farem ve vztahu ke státní správě (obrázek č. 24). Na základě toho sestavil katalog

základních informačních toků, přehled důležitých kritérií měření a jejich hodnocení. Ovšem ve výsledné identifikaci nejsou zahrnuty informační toky mezi farmou a celní správou (tedy problematika uplatnění vrácení daně z minerální olejů používaných pro zemědělskou prvovýrobu upravenou § 57 zákona č. 353/2003 Sb., o spotřebních daních) a katastrálním úřadem (tedy problematika prodeje a evidence zemědělských pozemků).



Obrázek č. 24 - Informační toky malé farmy specializované na chov masného skotu (podle Ulmana, 2009).

Další konceptuální model identifikující informační toky farmy představil Sörensen, et al. (2010) pod názvem „bohatý obrázek“ [rich picture]. V rámci specifikace a návrhu zemědělského manažerského informačního systému byl konceptuální model odvozen s cílem zachytit procesy a zájmy zemědělců z hlediska nakládání s informacemi. Tento model je podle autora disertační práce možné považovat za přijatelný k identifikaci globálních informačních toků pro návrh zemědělského informačního systému.



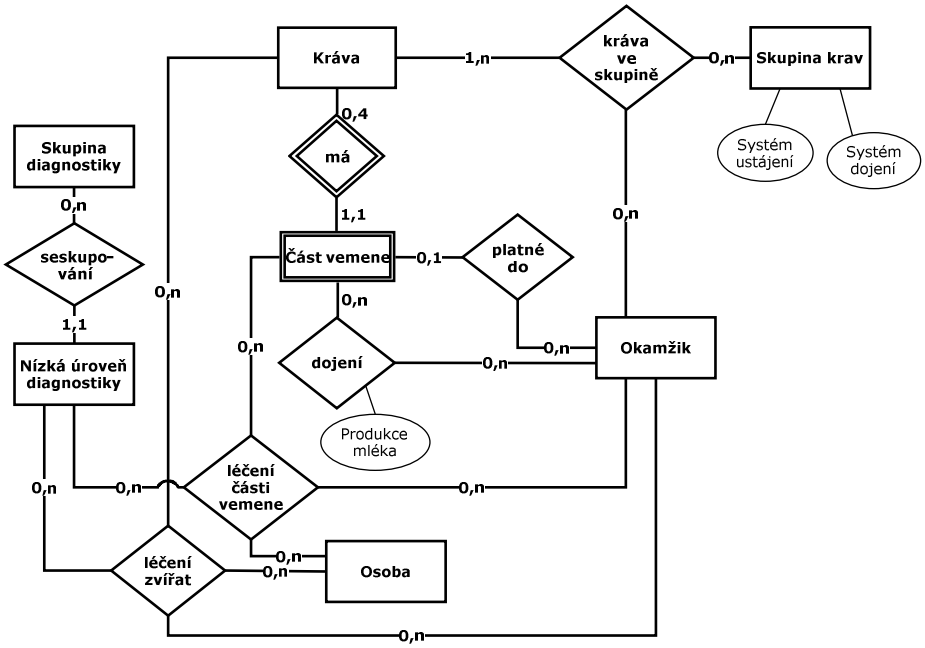
Obrázek č. 25 - Management farmy, "rich picture" (podle Sörensen, et al., 2010).

Nicméně nelze opomenout, že úspěšné zhodnocení informací v podniku vyplývají z obchodních pochopení podnikových procesů. Autorem práce je doporučeno využít například WCA Framework popsaný v příspěvku Raina (2010).

1.7.3. Modelování multidimenzionálních dat v zemědělském podniku

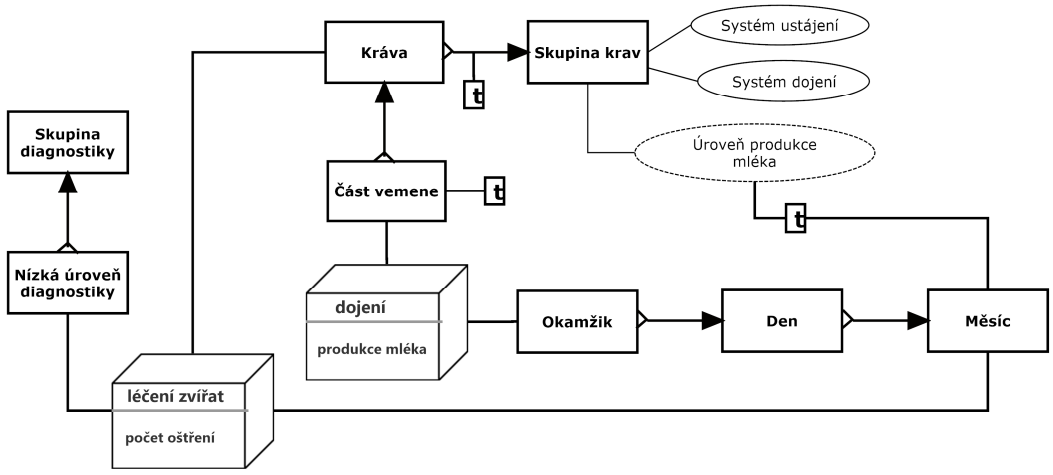
Interdisciplinární pojetí zemědělství vyžaduje velmi vysoké standardy pro správu dat. Zvláštní pozornost je nutné věnovat tvorbě provozních a analytických podkladů pro použití OLAP. Právě tento přístup je rozdílný oproti modelování pouze databází pro OLTP. V zemědělském podniku jsou databáze pro OLAP spíše výjimkou. Přesto je v současnosti možné najít literaturu, která se tématem návrhu OLAP a multidimenzionálních databází v zemědělství zabývá. Příkladem je příspěvek Schulze (2007) a Raie (2008), kteří popsali návrh OLAP v odlišném aplikačním kontextu.

Konkrétní aplikační scénář příspěvku (Schulze, et al., 2007) zahrnuje léčbu chorob u krav s podmnožinou ošetření vemene s informací o čase, o osobě, která poskytla ošetření, a o diagnostice. Při dojení krávy v určitém okamžiku je dojivost měřena individuálně pro každou část vemene. Příslušný ER model pro tento aplikační scénář na úrovni podniku je znázorněn na obrázku č. 26.



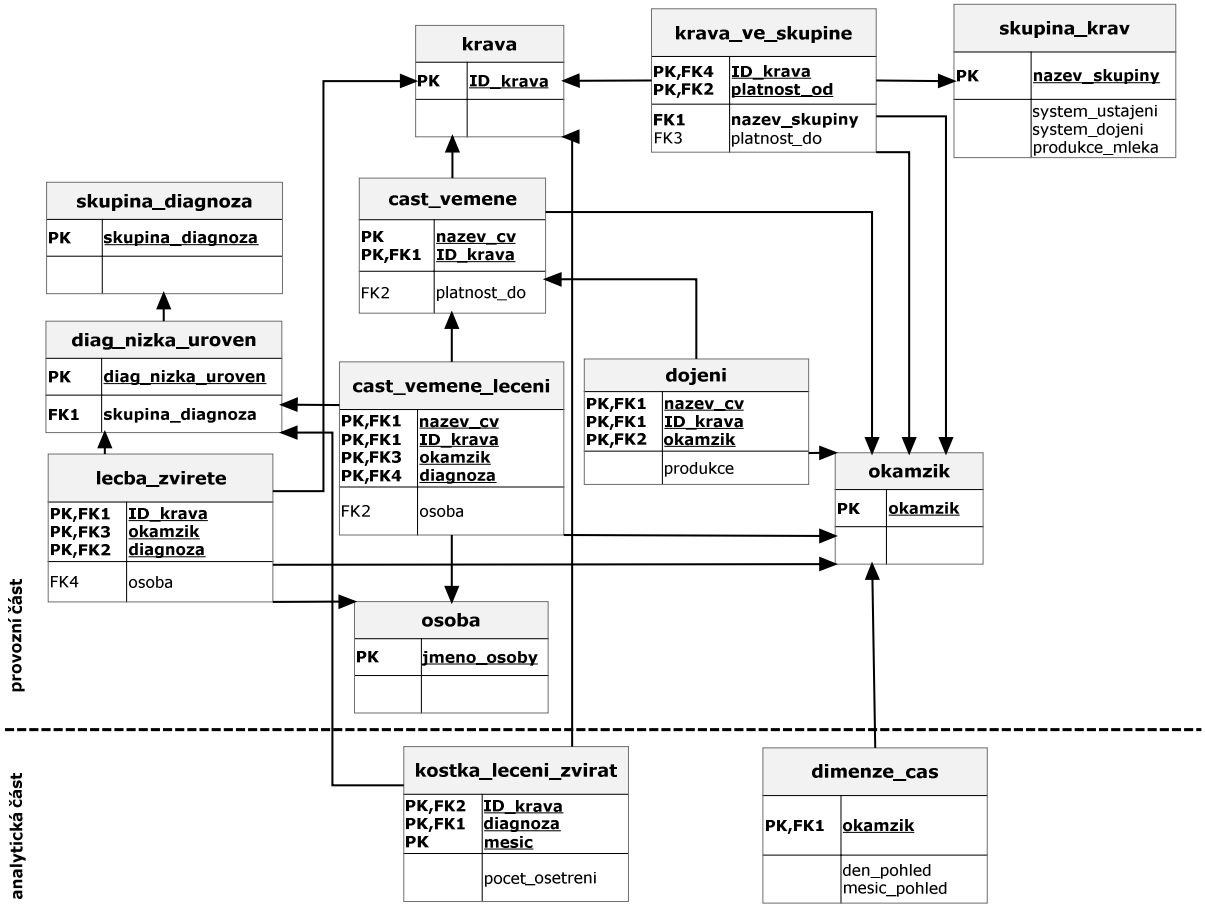
Obrázek č. 26 - ER model pro data skotu (podle Schulze, et al., 2007).

ER model (obrázek č. 26) zahrnuje typy entit, jejich vztahy a nezbytné atributy. Vztah „dojení“ je modelován jako typ vztahu mezi typy entit „časový okamžik“ a „část vemene“ s atributem „produkce mléka“ atd.



Obrázek č. 27 - Rozšířený mER model pro data skotu (podle Schulze, et al., 2007).

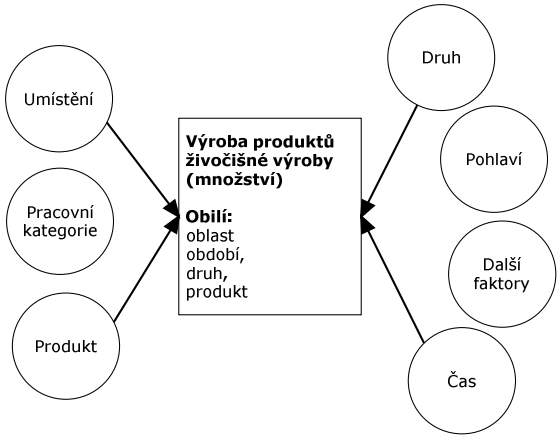
Obrázek č. 27 ukazuje analyticky orientovaný pohled potřebný pro aplikaci výše uvedeného scénáře prostřednictvím mER. Akce „dojení“ představuje fakt související s mírou „produkce mléka“. V analytickém pohledu, může být léčba onemocnění buď krávy, nebo části vemene chápána jako agregovaná míra (počet ošetření) a k ní vztah „léčba zvířete“.



Obrázek č. 28 - Implementace ER a mER modelu do logického návrhu (podle Schulze, et al., 2007).

Kromě konceptuálního návrhu multidimenzionální databáze v zemědělském podniku je nutné příslušné modely transformovat do relačního modelu. Časové omezení v klasifikaci hierarchie představuje časovou dobou platnosti konkrétní dimenzionální úrovně nebo roll-up vztahu. Obrázek č. 28 zobrazuje implementaci konceptuálních modelů do jednoho logického návrhu.

Další možný přístup k modelování multidimenzionálních dat v kontextu zemědělství je představen v příspěvku (Rai, et al., 2008). V příspěvku jsou nejprve identifikovány tři důležité dimenze pro databázi zvířat: zvířata, umístění a čas. Následně navržena tabulka faktů. Obrázek č. 29 ukazuje diagram faktů živočišné výroby. V tomto případě je mírou „množství živočišných produktů“, jako jsou mléko, maso, vlna a vejce. Tato produkce je měřena buď v hmotnosti, nebo podle počtu kusů v závislosti na druhu produktu. Například množství mléka, masa, vlny v daném roce se měří v jednotkách hmotnosti, zatímco vejce s odkazem na stejné rozměry jsou měřeny počtem kusů. (Rai, et al., 2008)



Obrázek č. 29 - Diagram faktu živočišné výroby (podle Raie, et al., 2008).

Oba dva přístupy návrhu OLAP databází Schulze (2007) a Raie (2008) jsou odlišné. Autor disertační práce považuje přístup Raie (2008) ne zcela zdařilý. Z diagramu faktů není snadno rozpoznatelné, co je míra, atribut nebo dimenze. Jiný přístup představil v příspěvku Schulze (2007). Na rozdíl od klasických konceptů (tedy i přístupu Raie (2008)) je předpokládán paralelní proces modelování pro oba pohledy (provozní a analytický), což má za následek společné logické schéma dat. Schulze (2007) se domnívá, že to je jediný způsob, jak účinně zabránit redundanci a nesrovnalostem jak na datové úrovni, tak na úrovni schématu. V příspěvku Schulze (2007) je použit ER model pro modelování provozních dat. Pro modelování analytických dat je použit mER model. Oba konceptuální modely (ER a mER) mají společný logický model pro jedno společné relační schéma.

Z výše představených přístupů je zřejmé, že aplikace OLAP v zemědělském podniku je potřebná. Jsou to nejen OLAP v rámci precizního zemědělství (Schulze, et al., 2007), (Rai, et al., 2008), (Nilakanta, et al., 2008), ale i Agri Data Mining (Abdullah, et al., 2003), zemědělské systémy pro podporu rozhodování (tzv. ADSS-OLAP) (Abdullah, 2009), prostorové OLAP (Rivest, et al., 2005), (McGuire, et al., 2008) nebo geografické datové sklady (da Silva, et al., 2010).

2. Výsledky syntézy literární rešerše

Interdisciplinární pojetí zemědělství vyžaduje velmi vysoké standardy pro správu dat. Zvláštní pozornost je vhodné věnovat nejen tvorbě provozních, ale i analytických podkladů pro použití OLAP, který představuje přístup pro podporu rozhodování, jehož cílem je získat znalosti z analytických databází. V současnosti existuje několik přístupů pro uložení analytických dat. Mezi nejvýznamnější patří tak zvané multidimenzionální, relační, hybridní anebo desktop OLAP. Autor disertační práce považuje technologii ROLAP za právě vhodnou pro zemědělský podnik díky svojí flexibilitě ve vytváření ad-hoc dotazů a pro svoji schopnost pracovat s provozní databází, namísto vytváření datových skladů, a to i za cenu degrese účinnosti aplikace. V přístupu ROLAP, jsou data uložena v relační databázi pomocí speciálního multidimenzionálního schématu namísto tradičního relačního schématu. Základem návrhu ROLAP řešení jsou přístupy multidimenzionálního modelování (Datta, et al., 1999), (1993), (Pardillo, et al., 2010), (Abelló, et al., 2009). Právě tyto přístupy jsou rozdílné oproti modelování provozních (OLTP) databází.

V zemědělském podniku jsou analytické databáze spíše výjimkou. Přesto je v současnosti možné najít literaturu, která se tématem návrhu multidimenzionálních databází v zemědělství zabývá. Příkladem je příspěvek (Schulze, et al., 2007) a (Rai, et al., 2008), kteří popsali návrh databáze pro OLAP v odlišném aplikačním kontextu.

Autoři zabývající se problematikou návrhu analytických databází v zemědělství neuvažují návrh v kontextu produkční (popř. nákladové) funkce. Autor disertační práce považuje za podnětné uvažovat v návrhu OLAP produkční funkci zemědělského podniku. Produkční funkce představuje vztah mezi velikostí vstupů (výrobních faktorů) a velikostí výstupů, které zemědělský podnik produkuje. Autor se domnívá, že na základě identifikované produkční funkce zemědělského podniku, je možné vytvořit multidimenzionální databázi pro ekonometrické analýzy. Pro tvorbu takové databáze je ovšem nutné transformovat výrobní funkci (ekonometrický model) do konceptuálního modelu dat.

Důvod výběru ekonometrických modelů při použití s OLAP je, že zemědělský podnik může provádět analytické zpracování sumarizovaných a agregovaných dat a pomoci odpovědět například na otázky:

- Jak velká byla celková produkce [v tis. Kč] zemědělského podniku v roce 2011?
- Jak velká byla produkce mléka [v kg] v zemědělském podniku v roce 2011?
- Jak velká byla produkce mléka [v kg] u holštýnského skotu v měsíci březnu roku 2011?

Autor disertační práce se domnívá, že znalost výrobní funkce v rámci BI může zemědělským podnikům pomoci například v otázkách:

- O kolik se změní produkce při změně pracovní síly o jednotku?
- O kolik se změní produkce při změně výměry obhospodařované půdy o jednotku?

Všechny tyto otázky mohou být zemědělským podnikem řešeny efektivně prostřednictvím technologie ROLAP založené na multidimenzionálním modelu dat vytvořeném z ekonometrických modelů. Vzhledem k tomu, že existuje absence takové metodiky, je na základě těchto skutečností v práci stanovena základní výzkumná otázka: *Je možné vytvořit metodiku návrhu multidimenzionální databáze pro ekonometrické analýzy zemědělského podniku?*

3. Cíl disertační práce

Hlavním cílem předkládané disertační práce je navrhnout novou metodiku návrhu multidimenzionální databáze pro zajištění ekonometrických analýz za účelem zvýšení efektivnosti zemědělských podniků. Výsledná metodika musí podporovat návrh multidimenzionální struktury z produkčních funkcí v zemědělství a umožňovat realizaci OLAP prostřednictvím relační databáze.

Uvedený cíl disertační práce je dále diferencován na následující dílčí cíle:

- Navrhnout novou metodu pro transformaci ekonometrického modelu do konceptuálního a logického modelu dat.
    - Provést komparaci multidimenzionálních schémat podle měr měření kvality datových skladů a
    - vytvořit formální pravidla metody transformace ekonometrického modelu.
- Vytvořit prototyp multidimenzionální databáze zajišťující ekonometrické analýzy produkční funkce v zemědělství.
    - Provést návrh konceptuálního schématu prototypu,
    - vytvořit logické schéma prototypu a následně
    - vytvořit a implementovat fyzický návrh prototypu prostřednictvím technologie relační OLAP.

Z výše vymezených cílů jsou odvozeny následující operační hypotézy:

$H1$: Existuje významný rozdíl v hodnotách míry **měření srozumitelnosti** mezi schématy navrženými z analogie.

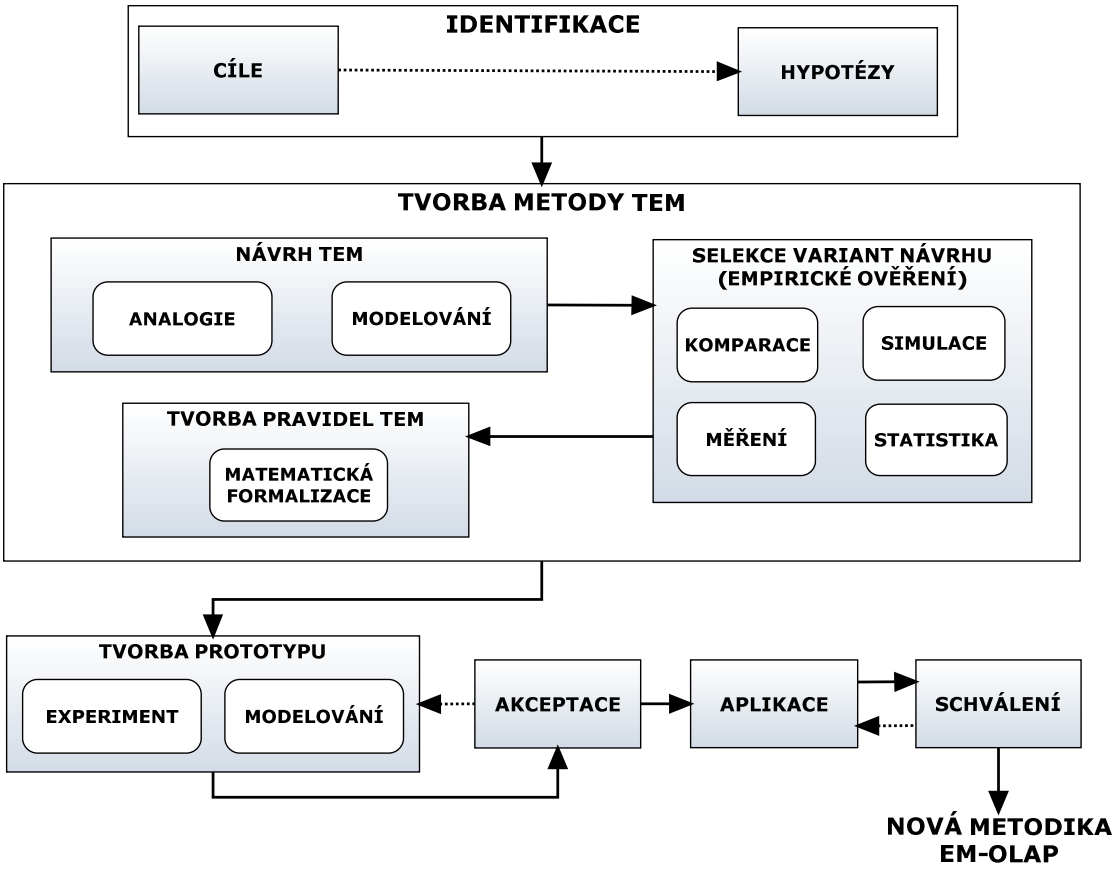
$H2$: Existuje významný rozdíl v hodnotách míry **měření kvality** mezi schématy navrženými z analogie.

$H3$: Existuje *přípustná transformace* ekonometrického modelu do **fyzického** schématu relačního OLAP.

Přípustnou transformací je v disertační práci myšlena taková transformace, při jejíž aplikaci nedojde k žádné ztrátě nebo zkreslení ekonometrických proměnných.

4. Metodika disertační práce

V rámci samotného zkoumání jsou použity obecně teoretické metody (analýza, syntéza, indukce, komparace a analogie) a metody empirické (měření a experiment). Kromě uvedených empirických a obecně teoretických metod jsou v práci použity také specifické vědecké metody (simulace, metody matematické a statistické), které umožní přesné exaktní vyjádření jevů a vztahů mezi nimi. Další významnou specifickou metodou použitou v disertační práci je multidimenzionální modelování, které představuje jak nástroj, tak předmět výzkumu. Celý vymezený rámec metodiky, použitých vědeckých metod a postupů řešení vymezených cílů objasňuje obrázek č. 30.



Obrázek č. 30 - Metodika disertační práce (vlastní zpracování).

Obdélníky znázorňují jednotlivé fáze metodiky disertační práce, zaoblené obdélníky představují použité vědecké metody při řešení příslušných cílů práce. Šipky znázorňují posloupnost metodického řešení.

Nová metodika návrhu multidimenzionální databáze pro použití ekonometrických modelů pro on-line analytické zpracování dat v zemědělství (EM-OLAP) je vytvořena na základě *identifikace* cílů a odvozených hypotéz. Pro její tvorbu je použita syntéza následujících kroků:

Návrh transformace ekonometrického modelu (TEM). V této fázi je vytvořena analogie ekonometrického modelu s multidimenzionálním schématem, tak aby bylo možné transformovat ekonometrický model do konceptuálního a logického schématu multidimenzionálního modelu dat.

Selekce variant návrhů TEM. Na základě navržených přístupů k transformaci ekonometrického modelu do multidimenzionálního schématu je věnována pozornost měření jejich kvality. V této části je komparováno, zdali míry kvality a míry srozumitelnosti multidimenzionálního schématu jsou pro navržené přístupy ekvivalentní nebo naopak diferentní. V rámci *empirického ověření* jsou ověřovány statistické hypotézy $H1$ a $H2$. Pro jejich přijetí nebo zamítnutí je použit dvouvýběrový test průměrů. Pro tvorbu výběrového souboru jsou náhodně generovány ekonometrické modely. Smyslem generování ekonometrických modelů je simulovat náhodné formální struktury modelů, které mohou v reálných podmínkách vzniknout. Pro vytvoření souboru náhodných generovaných ekonometrických modelů je vytvořen generátor. Jeho algoritmus je vyjádřen v pseudokódu. Výsledkem simulace a měření je statisticky podložené rozhodnutí, jaký z přístupu je vhodný pro transformaci ekonometrických modelů.

Tvorba pravidel metody TEM. V této fázi jsou vytvořena formální pravidla nové metody TEM prostřednictvím matematického aparátu.

Tvorba prototypu. V této fázi je vytvořen prototyp multidimenzionální databáze a aplikace OLAP pro zajištění ekonometrických analýz. Vytvoření prototypu umožní získat výhody a omezení pro vytvoření konečné metodiky EM-OLAP. Tvorba prototypu umožňuje validovat novou metodu TEM. Vytvořen je prototyp konceptuálního a logického multidimenzionálního schématu podle metody TEM. Pro návrh fyzického schématu multidimenzionální databáze je experimentováno s různými podobami integrovaných dat.

Akceptace. Cílem této fáze je systematické experimentování s dosaženým postupem návrhu metodiky EM-OLAP. Tento způsob je používán v souvislosti s tvorbou prototypu.

Aplikace. Přijatá metodika EM-OLAP je použita v reálných případech.

Schválení. Toto je poslední fáze metodiky disertační práce. Je to dynamická fáze, která probíhá společně s aplikační fází. Cílem této fáze je údržba nové metodiky EM-OLAP, tak aby byla přizpůsobena měnícímu se aplikačnímu prostředí.

Níže jsou představeny použité specifické modely, metody a příslušné metody měření.

4.1. Ekonometrické modely

Ekonometrický model představuje matematický model, který je matematicko-statistickou formulací ekonomické hypotézy. Vyjadřuje závislost ekonomických veličin na veličinách, které je podle hypotézy vysvětlují. V zemědělském prostředí je ekonometrický model velmi často tvořen více jak jednou rovnicí. Pokud se v modelu vyskytuje mezi vysvětlujícími proměnnými endogenní proměnné, pak se jedná o *soustavu simultánních rovnic*. (Cipra, 2008) Rovnice obsahující pouze endogenní proměnné se nazývá *identitní*.

Zápis ekonometrického modelu v symbolickém tvaru je: (Tvrdoň, 2006)

$$y_{1t} = \gamma_{11}x_{1t} + \gamma_{12}x_{2t} + \gamma_{13}x_{3t} + \gamma_{14}x_{4t} + u_{1t}$$

$$y_{2t} = \beta_{21}y_{1t} + \gamma_{21}x_{1t} + \gamma_{25}x_{5t} + u_{2t}$$

$y_{3t} = y_{1t} + y_{2t}$, pak (1)

y_s je endogenní proměnná *s*-tého druhu a její hodnota v období $t -\ y_{st}$, index *s* = (1, 2, … g), *t* = (1, …, *n*). x_r je *r*-tá exogenní proměnná s hodnotou v období $t -\ x_{rt}$, kde počet exogenních proměnných je roven *k*, pak *r* = (1, 2, … *k*). Endogenní proměnné zpožděné vyjadřují působení proměnných z období $t - z$, kde z = (1, 2, …, *t*−*z*). u_{st} je náhodná proměnná v *s*-té rovnici vysvětlované endogenní proměnné v období *t*. β_{is} je strukturní parametr v *i*-té rovnici modelu *s*-té nezpožděné endogenní proměnné a γ_{ir} v *i*-té rovnici modelu *r*-té predeterminované proměnné.

Fáze konstrukce simultánního ekonometrického modelu (Čechura, a další, 2009):

1. Vytvoření maticového zápisu modelu, přičemž obsah jednotlivých matic a vektorů je následující:
    - matice B obsahuje parametry endogenních proměnných modelu,
    - matice Γ obsahuje parametry predeterminovaných proměnných modelu,
    - vektor y_t obsahuje endogenní proměnné modelu,
    - vektor x_t obsahuje predeterminované proměnné modelu,
    - vektor u_t obsahuje stochastické proměnné modelu.

2. Provedení identifikace modelu podle podmínky:

    $k_{**} \geq g_{\Delta} - 1$, kde g je počet endogenních proměnných v modelu celkem, k je počet predeterminovaných proměnných v modelu celkem, symbol *, Δ znamenají, že proměnná je zahrnuta v identifikované rovnici, symbol **, ΔΔ znamenají, že proměnná v rovnici, pro niž se provádí identifikace, není obsažena v jiných rovnicích modelu.

Pro tvorbu prototypu (kapitola č. 5.2) byla vybrána produkční funkce konvenčního zemědělství nabývající následující mocninné formy podle (Kroupová, 2010):

$$y_{kt} = 234{,}25{L_{kt}}^{0{,}244}{WU_{kt}}^{0{,}616}{K_{kt}}^{0{,}100} \qquad (2)$$

Uvedená forma produkční funkce je aplikována za účelem komplexní analýzy vlivu základních výrobních faktorů konvenčního zemědělství na výslednou produkci y, která byla odhadována ve stálých cenách roku 2005 v tisících korunách. Vysvětlující proměnné představují následující výrobní faktory:

- Půda (L), hektarová výměra obhospodařované zemědělské půdy;
- Práce (WU), průměrný počet pracovníků;
- Kapitál (K), vyjádřený v podobě souhrnu hmotného a nehmotného dlouhodobého majetku v tisících korunách.

4.2. Metody měření kvality datových skladů

Pro komparaci multidimenzionálních schémat jsou provedena měření kvality datových skladů. Použity jsou metody měření podle Serrano (2008) a Gupta, et al., (2010). Oba tyto metodické přístupy jsou založeny na měření složitosti struktury datových skladů (Calero, et al., 2001):

Tabulka č. 2: Míry měření kvality datových skladů

NFT (Sc). Počet tabulek faktů ve schématu.
NDT (Sc). Počet tabulek dimenzí ve schématu.
NSDT (Sc). Počet sdílených tabulek dimenzí. Počet tabulek dimenzí společných pro více než jednu tabulku faktů ve schématu.
NAFT (Sc). Počet atributů tabulek faktů ve schématu.
NADT (Sc). Počet atributů tabulek dimenzí ve schématu.
NASDT (Sc). Počet atributů sdílených tabulek dimenzí ve schématu.
NFK (Sc). Počet cizích klíčů ve všech tabulkách faktů v schématu.

V příspěvku (Gupta, et al., 2010) je prostřednictvím lineární regresní analýzy zjištěno, že ze sedmi ukazatelů jsou nejlepšími prediktory pouze dva (NFT a NSDT), které mohou působit jako indikátory kvality modelu datových skladů.

V empirické studii (Serrano, et al., 2008) je navržen a ověřen soubor měr měření struktury jako ukazatelů srozumitelnosti, a tím i jako ukazatelů kognitivní složitosti schémat datových skladů. Mezi míry srozumitelnosti patří:

Tabulka č. 3: Míry měření srozumitelnosti schématu datových skladů

NFT(Sc). Počet tabulek faktů ve schématu.
NDT(Sc). Počet tabulek dimenzí ve schématu.
NFK(Sc). Počet cizích klíčů ve všech tabulkách faktů ve schématu. $NFK(Sc) = \sum_{i=1}^{NFT} NFK(FT_i)$, kde $NFK(FT_i)$ je počet cizích klíčů v tabulce faktů i ze schématu Sc.
NMFT(Sc). Počet měr (ukazatelů) v tabulkách faktů; Počet atributů v tabulkách faktů, které nejsou cizím klíčem $NMFT(Sc) = NA(Sc) - NFK(Sc)$, kde $NA(Sc)$ počet atributů v tabulkách faktů ve schématu Sc.

4.3. Pseudokód

Pro popis jednotlivých algoritmů autor disertační práce zavádí pseudokód (Robertson, 2003), kterým jsou popisovány jednotlivé algoritmy. Jedná se o formální zápis algoritmu, který není možné zkompilovat do spustitelné podoby. Základní notace použitá v této práci je následující (upraveno podle Chu (2006)):

1. Blok kódu je oddělen pomocí odsazení a zároveň ukončen pomocí konce bloku (například *konec if*, *konec for*).

2. Pro iterační konstrukce jsou k dispozici výrazy *while*, *for* a *repeat*. Podmínková konstrukce je k dispozici jedna, a to podmínka *if-then-else*. Význam těchto konstrukcí je ekvivalentní jazykům Pascal, C apod. *While* je cyklus s podmínkou (testem) na začátku (tedy jeho blok nemusí být vykonán ani jednou), *repeat* je cyklus s podmínkou na konci (jeho blok je vykonán minimálně jednou) a *for* je cyklus s automatickou inkrementací řídicí proměnné.

3. Komentáře jsou uvedeny symbolem //. Tento symbol indikuje, že zbytek řádku za ním je komentářem.

4. Operace přiřazení je zapsána symbolem ←. Levý operand je proměnná, které je hodnota přiřazována, pravý operand je výraz, který po vyhodnocení dá přiřazovanou hodnotu. Je možné použít i vícenásobné přiřazení, například ve formě i ← j ← k. Operace přiřazení je vyhodnocována vždy zprava.

5. Proměnné jsou vždy lokální pro danou funkci. V případě využití hodnoty proměnné i mimo funkci je používán návratový příkaz *Vrať*.

6. Přístup k jednotlivým elementům pole je možné specifikováním jména pole a indexu prvku v hranatých závorkách. Například i-ty prvek pole A je zapsán jako A[i]. Pole je indexováno od indexu 0.

4.4. PowerPivot

Doplněk Microsoft® PowerPivot for Microsoft® Excel 2010 je nástroj pro analýzu dat, který poskytuje uživatelům výpočetní možnosti OLAP přímo v aplikaci MS Excel.

PowerPivot umožňuje provádět zpracování rozsáhlých souborů dat (řádově miliony řádků) s přibližně stejnou výkonností, jako by zabralo zpracování několika set řádků (v kontextu MS Excelu). Součástí je podpora integrace dat z mnoha zdrojů, včetně databází, tabulek, sestav a textových souborů. Umožňuje tvorbu relací mezi tabulkami. Pro práci s daty používá jazyk DAX [Data Analysis Expressions]. Umožňuje provádět interaktivní prohlížení, analýzy a tvorbu sestav pomocí nativních funkcí aplikace Excel 2010, jako jsou kontingenční tabulky [PivotTables], průřezy a další analytické funkce typické pro OLAP.

5. Materiál a výsledky

V této části autor disertační práce představuje vědecké výsledky disertační práce a způsob jejich řešení. Autor se snaží hledat odpovědi na otázky, zdali je možné a jak transformovat ekonometrický model do konceptuálního schématu pro návrh multidimenzionální databáze. Ověřovat stanovené operační a statistické hypotézy, vytvářet prototypy řešení. Výstupem této části disertační práce je nová metodika návrhu multidimenzionální databáze pro podporu ekonometrických analýz v zemědělském podniku.

5.1. Transformace ekonometrického modelu

Cílem této části je navržení nové metody pro *transformaci ekonometrického modelu (TEM)*. Nová metoda TEM umožní transformovat ekonometrický model v zemědělství do konceptuálního a logického modelu dat v procesu návrhu multidimenzionální databáze.

5.1.1. Návrh TEM

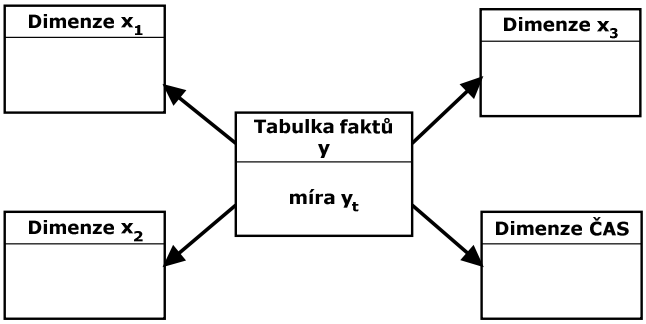
Autor uvažuje ekonometrický model (EM) s jednou rovnicí podle pravidel (2):

$$y_t = \gamma_1 x_{1t} + \gamma_2 x_{2t} + \gamma_3 x_{3t}. \quad (3)$$

Může se jednat například o produkční, nákladovou nebo jakoukoliv jinou funkci v zemědělství. Exogenní proměnné mohou představovat stavy hospodářských zvířat, výše podpory dotací atp. V této fázi není podstatné, jakého významu budou jednotlivé proměnné nabývat. Nyní autor bude hledat odpověď na otázku, zdali lze takový ekonometrický model transformovat do konceptuálního schématu. Vzhledem k vymezenému hlavnímu cíli disertační práce a jeho specifikaci je důležité zvolit typ schématu vhodný i pro návrh multidimenzionální databáze vytvořené prostřednictvím relační OLAP technologie.

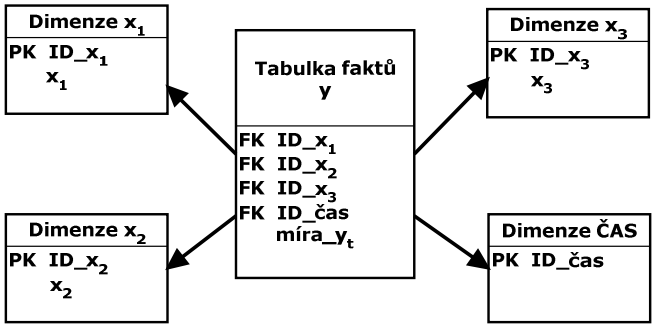
V prvním fázi konceptuálního návrhu multidimenzionální databáze je nutné vytvořit tabulku faktů do prázdného konceptuálního schématu. Z rovnice ekonometrického modelu (3) je zřejmé, že za fakt je možné považovat endogenní proměnnou y. Tedy y bude představovat tabulku faktů. Exogenní proměnné x_1, x_2, a x_3 pak dimenze. Jelikož model (3) obsahuje časovou proměnnou *t*, pak schéma bude obsahovat ještě dimenzi

času. Tabulka faktů bude asociována vztahem roll-up se všemi příslušnými dimenzemi, proměnnými na pravé straně rovnice (3). Celý zápis rovnice (3) bude představovat míru, tedy sledovaný ukazatel, která bude součástí tabulky faktů. Takto vytvořené konceptuální schéma je znázorněno na obrázku č. 31.



Obrázek č. 31 - Konceptuální schéma TEM (vlastní zpracování).

V rámci převodu do logického schématu bude každá dimenze opatřena číselným primárním klíčem a asociována prostřednictví cizího klíče s tabulkou faktů (obrázek č. 32).

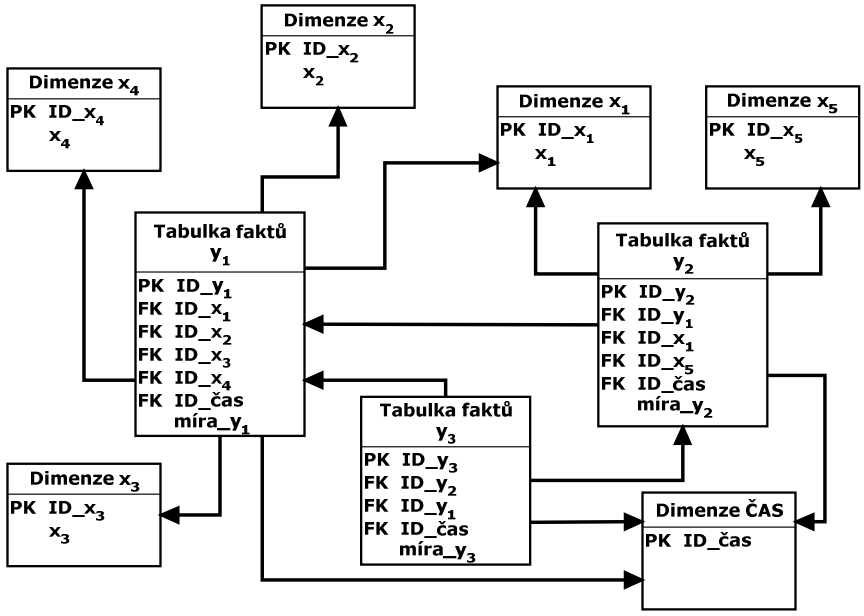


Obrázek č. 32 - Logické schéma TEM pro EM s 1 rovnicí (vlastní zpracování).

Výše uvedená transformace byla provedena pro model s 1 rovnicí. Je tedy vhodné uvažovat složitější model, například model (1) se třemi rovnicemi popsaný v části 4.1. Na tento ekonometrický model je aplikován výše uvedený postup následovně.

V prvním fázi je vytvořena tabulka faktů do prázdného schématu pro y_1, y_2 a y_3. Následně jsou vytvořeny dimenze do schématu pro každou exogenní proměnnou v ekonometrickém modelu (x_1, x_2, x_3, x_4 a x_5). Jelikož model obsahuje časovou proměnnou *t*, pak je také vytvořena dimenze času. Vytvořena je roll-up asociace mezi tabulkou faktů a dimenzemi. Tedy například rovnice $y_{2t} = \beta_{21}y_{1t} + \gamma_{21}x_{1t} + \gamma_{25}x_{5t} + u_{2t}$ vyjadřuje, že dimenze x_1 a x_2 jsou v relaci s tabulkou faktů y_2. Ovšem v této druhé

rovnici se vyskytuje i endogenní proměnná y_1. Tedy musí vzniknout roll-up asociace i mezi tabulkou faktů y_1 a tabulkou faktů y_2. V rámci převodu do logického schématu bude každá dimenze opatřena číselným primárním klíčem a asociována prostřednictví cizího klíče s tabulkami faktů. Pro každou ze tří rovnic je vhodné sledovat diferentní míry, které budou součástí každé tabulky faktů. Náhodné složky u_{1t}, u_{2t} nejsou v konceptuálním ani v logickém schématu znázorněny. Takto vytvořené logické schéma je znázorněno na obrázku č. 33.



Obrázek č. 33 - Logické schéma TEM pro EM se 3 rovnicemi, 1. varianta (vlastní zpracování).

Je nutné podotknout, že takto vzniklé schéma je fyzicky realizovatelné pouze při použití ROLAP technologie. V případě MOLAP realizace by nebylo možné propojit mezi s sebou datové kostky (tabulky faktů v terminologii ROLAP). Proto v případě simultánních ekonometrických modelů (1) je vhodné model převést do redukované formy. V případě modelu (1) by taková redukovaná forma pro 2. rovnici vypadala následovně:

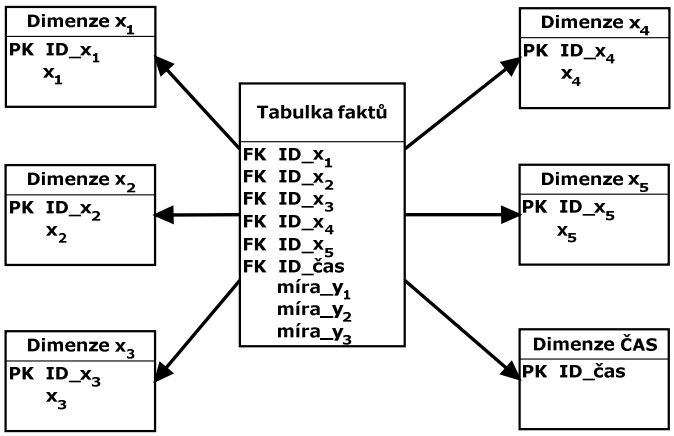
$$y_{1t} = \gamma_{11}x_{1t} + \gamma_{12}x_{2t} + \gamma_{13}x_{3t} + \gamma_{14}x_{4t} + u_{1t}$$

$$y_{2t} = \beta_{21}(\gamma_{11}x_{1t} + \gamma_{12}x_{2t} + \gamma_{13}x_{3t} + \gamma_{14}x_{4t}) + \gamma_{21}x_{1t} + \gamma_{25}x_{5t} + u_{2t} \qquad (4)$$

Tedy prostou substitucí je možné vyjádřit rovnici bez endogenních proměnných na pravé straně rovnice. Důsledkem je, že v konceptuálním (logickém) schématu nebudou

propojeny tabulky faktů mezi s sebou. Tím bude možné model realizovat v MOLAP úložišti dat, ovšem za cenu nárůstu propojení mezi tabulkami faktů a dimenzí.

Model s jednou rovnicí vyjadřuje typické schéma hvězdy, naopak model s více rovnicemi odpovídá schématu konstelace (v případě simultánních modelů) nebo souhvězdí. Je tedy zřejmé, že vytvořené logické schéma z ekonometrického modelu se 3 rovnicemi (obrázek č. 33) je oproti ekonometrickému modelu s 1 rovnicí subjektivně „složitější“ (obrázek č. 32). Proto je podnětné dále uvažovat, zdali neexistuje jiný strukturní návrh konceptuálního schématu s ekvivalentní sémantikou. Tedy výše uvedené tvrzení, že y_t představuje tabulku faktů, lze poupravit. Je možné uvažovat, že bude existovat pouze jedna tabulka faktů ve schématu a jednotlivé endogenní proměnné nebudou modelovány. Namísto toho bude každá rovnice představovat jednu míru tabulky faktů. Výsledek takové varianty návrhu je zobrazen na obrázku č. 34.



Obrázek č. 34 - Logické schéma TEM pro EM se 3 rovnicemi, 2. varianta (vlastní zpracování).

Takto vytvořené schéma umožňuje zaznamenat stejné ekonometrické proměnné, jako původně uvažovaný přístup (obrázek č. 33). Ovšem vyvstává otázka, která varianta schématu vzniklé působením analogie je vhodnější pro návrh multidimenzionální databáze.

5.1.2. Komparace multidimenzionálních schémat

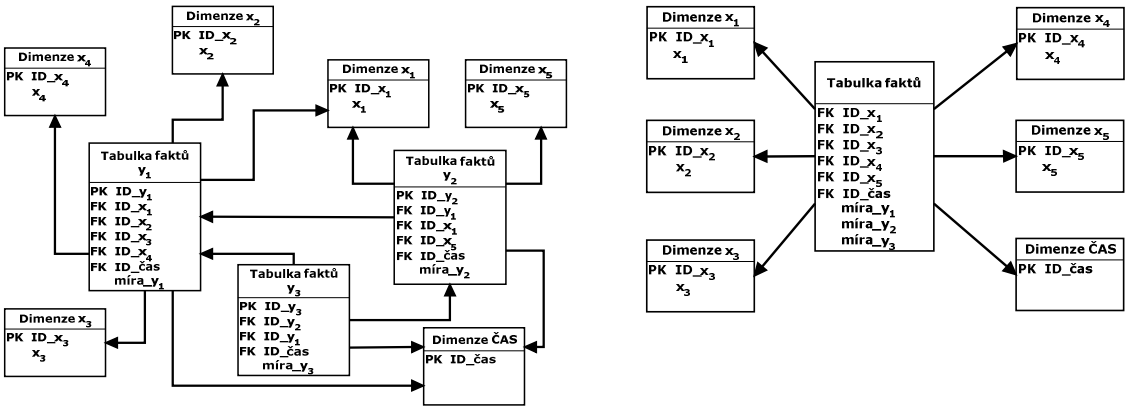
Výše uvedené dvě varianty transformace ekonometrického modelu jsou posibilní pro návrh multidimenzionálního schématu. Vzhledem k obecně rostoucí složitosti datových skladů je vhodné věnovat pozornost hodnocení jejich kvality v průběhu jejich vývoje. V této části bude ověřeno, zdali míra kvality a míra srozumitelnosti multidimenzionálního schématu je pro obě výše představené varianty transformace

ekvivalentní nebo naopak diferentní. To je důležité pro rozhodnutí, jaký z těchto dvou přístupů vybrat pro transformaci ekonometrického modelu.

5.1.2.1. Kvantitativní srovnání

Pro hodnocení kvality schémat jsou použity míry měření kvality datových skladů (NFT, NSDT) uvedené v části 4.2 podle (Gupta, et al., 2010). Hodnocení je provedeno pro prvou a druhou variantu návrhu. Souhrnně zobrazeno na obrázku č. 35.

1. varianta návrhu TEM **2. varianta návrhu TEM**



Obrázek č. 35 - Logické schéma TEM pro EM se 3 rovnicemi, 1. a 2. varianta (vlastní zpracování).

Výsledky hodnocení jsou uvedeny v tabulce č. 4. Z výsledků je patrné, že vytvořené schéma podle prvé varianty TEM je strukturálně složitější (celková hodnota měr je vyšší než u druhého způsobu).

Tabulka č. 4: Výsledek hodnocení kvality schématu (vlastní zpracování).

1. varianta návrhu TEM		**2. varianta návrhu TEM**	
Míra měření	Hodnota míry	Míra měření	Hodnota míry
NFT(Sc)	3	*NFT(Sc)*	1
NSDT(Sc)	2	*NSDT(Sc)*	0
Celkem	**5**	**Celkem**	**1**

Dalším možným ukazatelem kvality výsledných schémat je míra měření srozumitelnosti uvedená v části 4.2. Hodnocení je opět provedeno pro prvou a druhou variantu TEM. Výsledky hodnocení jsou uvedeny v tabulce č. 5.

Tabulka č. 5: Výsledek hodnocení srozumitelnosti schématu (vlastní zpracování).

1. varianta návrhu TEM		**2. varianta návrhu TEM**	
Míra měření	Hodnota míry	Míra měření	Hodnota míry
NFT(Sc)	3	*NFT(Sc)*	1
NDT(Sc)	6	*NSDT(Sc)*	0
NFK(Sc)	12	*NFK(Sc)*	6
NMFT(Sc)	3	*NMFT(Sc)*	3
Celkem	**24**	**Celkem**	**10**

Z výsledků měření (tabulka č. 5) je zřejmé, že prvá varianta je z hlediska srozumitelnosti výrazně horší. Opět druhá varianta je pro TEM vhodnější. Vzhledem k faktu, že výše uvedené hodnocení proběhlo pouze na jednom schématu, nelze tvrzení, že druhá varianta TEM je pro návrh multidimenzionálního modelu dat vhodnější, považovat za významné. Proto je v další části disertační práce provedeno statistické ověření rozdílu mezi prvou a druhou variantou TEM.

5.1.2.2. Empirické ověření

V této části je empiricky ověřován rozdíl mezi kvalitou schématu navrženým podle prvé a druhé varianty transformace ekonometrického modelu.

Statistické hypotézy

Operační hypotéza $H1$ tvrdí, že existuje významný rozdíl v hodnotách míry *měření srozumitelnosti* mezi schématy navrženými z analogie. Tato operační hypotéza je převedena na statistickou hypotézu $H1_0$, která tvrdí, že existuje shoda průměrů obou souborů. Proti ní stojí alternativní hypotéza $H1_1$, která tvrdí opak. Nulová a alternativní hypotéza pro dvouvýběrový test průměrů je:

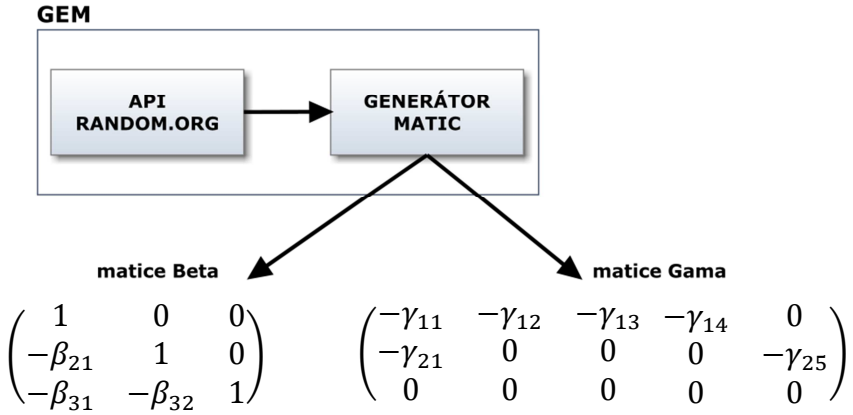
$H1_0: \mu = \mu_0$, $H1_1: \mu \neq \mu_0$.

Hypotéza $H2$ tvrdí, že existuje významný rozdíl v hodnotách míry *měření kvality* mezi schématy navrženými z analogie. Tato operační hypotéza je převedena na statistickou hypotézu $H2_0$, která tvrdí, že existuje shoda průměrů obou souborů. Proti ní stojí alternativní hypotéza $H2_1$, která tvrdí opak. Nulová a alternativní hypotéza pro dvouvýběrový test průměrů je:

$H2_0: \mu = \mu_0, H2_1: \mu \neq \mu_0.$

Experimentální materiály

Pro ověření stanovených hypotéz jsou náhodně generovány ekonometrické modely. Výběrový soubor obsahuje 30 náhodně generovaných ekonometrických modelů splňující podmínku identifikace (viz kapitola 4.1). Smyslem generovaných ekonometrických modelů je odrážet náhodné formální struktury modelů, které mohou v reálných podmínkách vzniknout. Tedy generované ekonometrické modely neobsahují odhady parametrů, ani jejich verifikaci (ekonomickou, ekonometrickou, statistickou nebo matematickou).



Obrázek č. 36 - Generátor ekonometrických modelů (vlastní zpracování).

Pro vytvoření souboru náhodných generovaných ekonometrických modelů je vytvořen *generátor ekonometrických modelů (GEM)*. Princip vytvořeného generátoru je zobrazen na obrázku č. 36. Jeho základním principem je vytvořit náhodné matice Beta (B) a Gama (Γ), které popisují vztahy mezi proměnnými v modelu. Náhodnost je řešena prostřednictvím API [Application Programming Interface] RANDOM.ORG, která představuje službu pro generování opravdových náhodných čísel (na rozdíl od pseudonáhodných čísel generovaných typickými programovacími jazyky). Náhodnost čísel z API RANDOM.ORG je na rozdíl od pseudo-generátorů náhodných čísel řešena

prostřednictvím atmosférického šumu. Takto náhodně vygenerované matice B a Γ jsou použity pro tvorbu ekonometrických modelů ve strukturální formě: B$y_t + \Gamma x_t = u_t$.

Algoritmus GEM (v pseudokódu) je zobrazen na obrázku č. 37, kde:

- *generuj(parametr1, parametr2)* je funkce vracející náhodné číslo od hodnoty *parametr1* do *parametr2* generované z random.org,
- *TvorbaMaticeB(g)* je funkce, která generuje matici *B* ve tvaru $[g \times g]$,
- *TvorbaMaticeG(g, k)* je funkce, která generuje matici Γ ve tvaru $[g \times k]$.

```
Proměnné:
maticeB      // matice Beta (pole)
maticeG      // matice Gama (pole)
radek        // inkrement (číslo)
sloupec      // inkrement (číslo)
g            // počet endogenních proměnných (číslo)
k            // počet predeterminovaných proměnných (číslo)

Generování:
g ← generuj(1, 3)
k ← generuj(2, 10)

Proveď funkci TvorbaMaticeB(g)
Proveď funkci TvorbaMaticeG(g, k)

Vypiš matici(maticeB)
Vypiš matici(maticeG)

funkce TvorbaMaticeB(g)
 for radek ← 0 to g-1 do
       for sloupec ← 0 to g-1 do
             if radek = sloupec then
                   maticeB[radek][sloupec] ← 1
             else
                   maticeB[radek][sloupec] ← generuj(0,-β)
             konec if
       konec for
 konec for
vrať maticeB
konec funkce

funkce TvorbaMaticeG(g, k)
 for radek ← 0 to g-1 do
       for sloupec ← 0 to k do
             maticeG[radek][sloupec] ← generuj(0,-γ)
       konec for
 konec for
vrať maticeG
konec funkce
```

Obrázek č. 37 - Algoritmus GEM v pseudokódu (vlastní zpracování).

Tento algoritmus generuje modely s jednou až maximálně třemi rovnicemi. Počet predeterminovaných proměnných v rovnici je omezen na 11. Výsledné generované

rovnice jsou testovány, zda splňují podmínku identifikace $k_{**} \geq g_{\Delta} - 1$ (kapitola 4.1). Pokud ano, jsou generované ekonometrické modely zařazeny do souboru k statistickému testování. K tvorbě statistických výstupů je použit statistický software STATISTICA 10.

Výsledky a interpretace experimentu

Pro ověření statistické **hypotézy $H1$** je použit t-test pro nezávislé proměnné. Interpretaci samotného t-testu (tabulka č. 6) předchází posouzení průkaznosti rozdílu mezi rozptyly $(s_1{}^2, s_2{}^2)$ s použitím F-testu ($F = 5{,}426334$). K výsledku F-testu přísluší vypočtená hladina významnosti $p = 0{,}000018$, která je nižší než stanovená hladina významnosti $\alpha = 0{,}05$. Rozptyly se tedy na hladině významnosti $\alpha = 0{,}05$ statisticky významně liší, není tedy splněna podmínka homogenity rozptylů.

Tabulka č. 6: Statistické hodnocení schémat (vlastní zpracování).

Skup. 1 vs. skup. 2	Průměr skup. 1	Průměr skup. 2	Hodnota t.	sv	p	Poč. plat. skup. 1	Poč. plat. skup. 2	Sm.odch. skup. 1	Sm.odch. skup. 2	F-poměr Rozptyly	p - Rozptyly
1. způsob vs. 2. způsob	18,1333	12,9	3,660596	58	0,000545	30	30	7,195465	3,088912	5,426334	0,000018

Na základě nehomogenních rozptylů je proveden výpočet p pro dvouvýběrový test. Výsledky testu jsou uvedeny v tabulce č. 7.

Tabulka č. 7: Statistické hodnocení schémat (vlastní zpracování).

Skup. 1 vs. skup. 2	t separ. odh. prom.	sv	p oboustr.
1. způsob vs. 2. způsob	3,66059592	39,3375373	0,000738545

Hodnota $p \cong 0{,}000738$ (tabulka č. 7) je výrazně nižší než zvolená hladina významnosti $\alpha = 0{,}05$ (i pro hladinu významnosti $\alpha = 0{,}01$). Nulová hypotéza o shodě průměrů obou souborů je zamítnuta a přijata je hypotéza alternativní.

Na základě provedeného statistického testování autor disertační práce konstatuje, že je s 99% pravděpodobností prokázán statisticky významný rozdíl v *hodnotách míry měření srozumitelnosti* mezi schématy navrženými z analogie.

Pro ověření statistické **hypotézy $H2$** je stejně jako pro $H1$ použit t-test pro nezávislé proměnné. Výsledek použitého F-testu je $F = 0$ a příslušná vypočtená hladina významnosti $p = 1$, která je vyšší než stanovená hladina významnosti $\alpha = 0{,}01$ (i $\alpha = 0{,}05$). Rozptyly se tedy na hladině významnosti $\alpha = 0{,}05$ statisticky významně neliší, je tedy splněna podmínka homogenity rozptylů.

Tabulka č. 8: Statistické hodnocení schémat (vlastní zpracování).

Skup. 1 vs. skup. 2	Průměr skup. 1	Průměr skup. 2	Hodnota t.	sv	p	Poč. plat. skup. 1	Poč. plat. skup. 2	Sm.odch. skup. 1	Sm.odch. skup. 2	F-poměr Rozptyly	p - Rozptyly
1. způsob vs. 2. způsob	4,1667	1	6,63509	58	0	30	30	2,614065	0	0	1

K výsledku samotného t-testu $t \cong 6{,}6351$, pak přísluší vypočtená hladina významnosti $p = 0$. Jelikož $p < \alpha = 0{,}01$, pak zamítáme nulovou hypotézu $H2_0$ a přijímáme hypotézu alternativní $H2_1$. Rozdíl mezi průměry obou souborů je statisticky významný.

Na základě provedeného statistického testování autor disertační práce konstatuje, že byl s 99% pravděpodobností prokázán statisticky významný rozdíl v *hodnotách míry měření kvality* mezi schématy navrženými z analogie.

5.1.3. Metoda TEM

Na základě výsledků kvantitativního srovnání a empirického ověření je vytvořena a formalizovaná nová metoda TEM, která je založena na druhé variantě návrhu multidimenzionálního schématu. Z obrázku č. 34 (se třemi rovnicemi) je patrné, že zvolený způsob odpovídá návrhovému typu *hvězda*.

5.1.3.1. Formální reprezentace

Pro formální definici pravidel metody TEM mějme množiny Y a X, kde:

$Y = \{y_s\} \cup \{y_{st}\}$ je konečná množina všech endogenních proměnných,

$X = \{x_r\} \cup \{x_{rt}\}$ je konečná množina všech exogenních proměnných a

$Rel \subseteq (X \times Y) \cup (Y \times Y)$ je množina strukturálních relací v ekonometrickém modelu.

Schéma *Hvězda* je určeno pěticí (*Ent*, *Key*, *Att*, *Ass*, *getKey*), kde:

Ent je konečná neprázdná množina entit ve schématu,

Key je konečná neprázdná množina klíčů ve schématu,

Att je konečná neprázdná množina atributů ve schématu,

$Fact \subseteq Ent$ je konečná množina faktů ve schématu,

$Dim \subseteq Ent$ je konečná množina dimenzí ve schématu,

$Measure \subseteq Fact$ je konečná množina měr ve schématu.

Každá entita $e \in Ent$ je popsána kolekcí klíčů a atributů a platí pro ni:

$\forall e \in Ent: \exists(\{k \in Key\} \cup \{a \in Att\})$.

$getKey$ je funkce, která vrací klíče entit ve schématu hvězda a platí pro ni:

$\forall e \in Ent: getKey(e): Ent \rightarrow Key_e \subseteq Key$.

$Ass \subseteq (Dim \times Fact)$ je konečná množina vztahů entit.

5.1.3.2. Návrh metody TEM

Fáze 1: Vytvoření základního schématu hvězda.

Pravidlo 1.1: Vytvoření měr do prázdného schématu pro každou endogenní proměnnou z ekonometrického modelu je definován vztahem:

$\forall y_s \in Y : m_s \in Measure$ a $\forall y_{st} \in Y : m_{st} \in Measure$

Pravidlo 1.2: Vytvoření dimenzí do schématu pro každou exogenní proměnnou z ekonometrického modelu je definován vztahem:

$\forall x_r \in X : d_s \in Dim$ a $\forall x_{rt} \in X : d_{rt} \in Dim$

Pravidlo 1.3: Pokud existuje časová proměnná v ekonometrickém modelu, pak se vytvoří dimenze času podle vztahu:

$\forall x_{rt} \in X : \ d_{rt} \in Dim_{time}$

Fáze 2: Vytvoření vztahů mezi entitami ve schématu hvězda.

Pravidlo 2.1: Pokud existuje vztah mezi exogenní proměnnou x a endogenní proměnnou y a funkce $getKey$, která vrací množinu klíčů těchto proměnných, pak se vytvoří asociace mezi faktem a dimenzí, pro kterou platí:

$\forall (x, y) \in Rel: (d, c, K) | (d \in Dim) \wedge (c \in Fact) \wedge ((d, c) \in Ass) \wedge$
$(K \subseteq K_d \cup K_c | (K_d = getKey(d)) \wedge (K_c = getKey(c)))$

5.1.3.3. Aplikace pravidel metody TEM

Pro ověření pravidel autor práce uvažuje ekonometrický model (1) a k němu zjednodušený významový příklad:

y_{1t} … hrubá produkce rostlinné výroby v období *t*,

y_{2t} … hrubá produkce živočišné výroby v období *t*,

y_{3t} … hrubá produkce zemědělská v období *t*,

x_{1t} … výše dotací (podpory),

x_{2t} … základní výrobní fondy v rostlinné výrobě,

x_{3t} … množství práce v rostlinné výrobě,

x_{4t} … klimatické podmínky,

x_{5t} … stavy hospodářských zvířat,

u_{1t}, u_{2t} … náhodná složka v období t,

Uvedený příklad představuje situaci, kdy celková produkce zemědělského podniku je závislá na rostlinné produkci a živočišné produkci a pro každou z těchto tří produkcí je vhodné sledovat diferentní míry. Výsledek aplikace metody TEM (po provedení jednotlivých formalizovaných pravidel) pro ekonometrický model (1) je demonstrován na obrázku č. 34. V první fázi jsou vytvořeny míry do tabulky faktů v prázdném schématu hvězdy pro hrubou produkci zemědělskou, hrubou produkci rostlinné a živočišné výroby (pravidlo 1.1). Následně jsou podle pravidla 1.2 vytvořeny dimenze do schématu hvězda pro každou exogenní proměnnou v ekonometrickém modelu (výše dotací, základní výrobní fondy v rostlinné výrobě, množství práce v rostlinné výrobě, klimatické podmínky, stavy hospodářských zvířat). Jelikož model (1) obsahuje časovou

proměnnou *t*, pak je vytvořena dimenze času. V poslední fázi (pravidlo 2.1) je vytvořena asociace prostřednictvím vygenerovaných klíčů mezi tabulkou faktů a dimenzemi. Tedy například rovnice $y_{2t} = \beta_{21} y_{1t} \gamma_{21} x_{1t} + \gamma_{25} x_{5t} + u_{2t}$ vyjadřuje, že výše dotací a stavy hospodářských zvířat mají relaci s hrubou produkcí živočišné výroby (tedy s mírou y_{2t} v tabulce faktů). V aplikačním kontextu může být rovnice vyjádřená například ve tvaru: $y_{1t} = 3{,}45x_{1t} + 1{,}32x_{2t} + 1{,}07x_{3t} + 0{,}43x_{4t} + 284{,}36$. Tedy náhodné složky u_{1t}, u_{2t} a parametry β, γ jsou již vyjádřeny číselně. Proto náhodné složky u_{1t}, u_{2t} (popř. jiné proměnné, které nejsou uvedeny v pravidlech metody TEM) nejsou ve schématu hvězdy znázorňovány.

5.2. Tvorba prototypu

Pro získání přesného náhledu budoucího řešení OLAP je prostřednictvím metody TEM vytvořen prototyp multidimenzionální databáze. Vytvoření prototypu umožní získat výhody a omezení pro vytvoření konečné metodiky návrhu multidimenzionální databáze v prostředí zemědělského podniku. Pro tvorbu prototypu je v disertační práci uvažován návrh datových skladů podle Rizzi, et al., 2006 (obrázek č. 14). Předpokladem výzkumného návrhu prototypu je, že z analýzy požadavků vyplývá pouze a jenom požadavek na analytickou aplikaci ekonometrického modelu v zemědělství. Pro tvorbu prototypu je použita produkční funkce konvenčního zemědělství (2).

5.2.1. Konceptuální návrh prototypu

Pro vytvoření konceptuálního schématu jsou aplikována pravidla metody TEM. Aplikace metody TEM na produkční funkci (2) vede k identifikaci míry a dimenzí. Tabulka faktů je pro celé schéma pouze jedna. Míra identifikovaná pravidlem 1.1 je tedy podmnožinou tabulky faktů.

Tabulka č. 9: Popis výsledku TEM (vlastní zpracování).

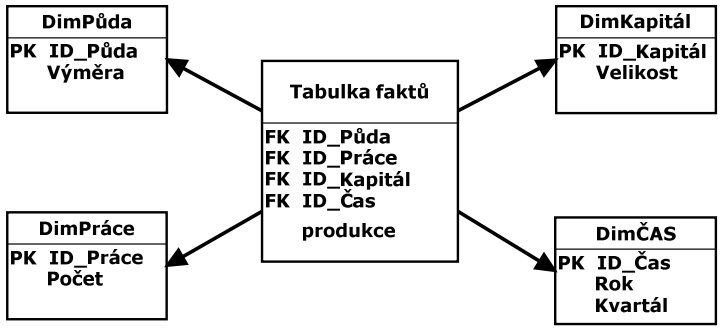
Pravidlo 1.1	Míra (ukazatel)	$y_{kt} = 234{,}25 {L_{kt}}^{0{,}244} {WU_{kt}}^{0{,}616} {K_{kt}}^{0{,}100}$
Pravidlo 1.2	Dimenze	Půda (L)
		Práce (WU)
		Kapitál (K)

Pravidlo 1.3	Dimenze	Čas

Výsledek aplikovaných pravidel (tabulky č. 9) umožňuje vytvoření konceptuálního schématu.

5.2.2. Logický návrh prototypu

V logickém návrhu je aplikováno pravidlo 2.1, které identifikovaným dimenzím přiděluje příslušnou asociaci s tabulkou faktů. Jelikož metoda TEM nepostihuje tvorbu atributů vzhledem k sémantice proměnných v modelu, je vhodné po provedení metody TEM doplnit do schématu další případné atributy dimenzí. Pro dimenzi Půda je vytvořen atribut Výměra, který bude obsahovat data o hektarové výměře obhospodařované zemědělské půdy. Pro dimenzi Práce je vytvořen atribut Počet, který bude obsahovat průměrný počet pracovníků. V dimenzi Kapitál bude atribut Velikost vyjádřený v podobě souhrnu hmotného a nehmotného dlouhodobého majetku v tisících korunách. Do dimenze Čas jsou přidány atributy, které budou umožňovat ekonometrické analýzy v dlouhodobém období. Z ekonomické podstaty nemá význam provádět analýzy z krátkodobého hlediska, jelikož většina faktorů v rovnici je v krátkodobém období neměnná. Zároveň je nutné zvolit typ granularit dat. Pro ekonometrickou analýzu je autorem doporučena snímková granularita. Data vstupují do databáze ve stejných časových úsecích (jednou za čtvrt roku). Dimenze času tedy zohledňuje jak rok, tak kvartál. V těchto intervalech je možné identifikovat změny jednotlivých dimenzí (proměnných). Výsledné logické schéma je na obrázku č. 38



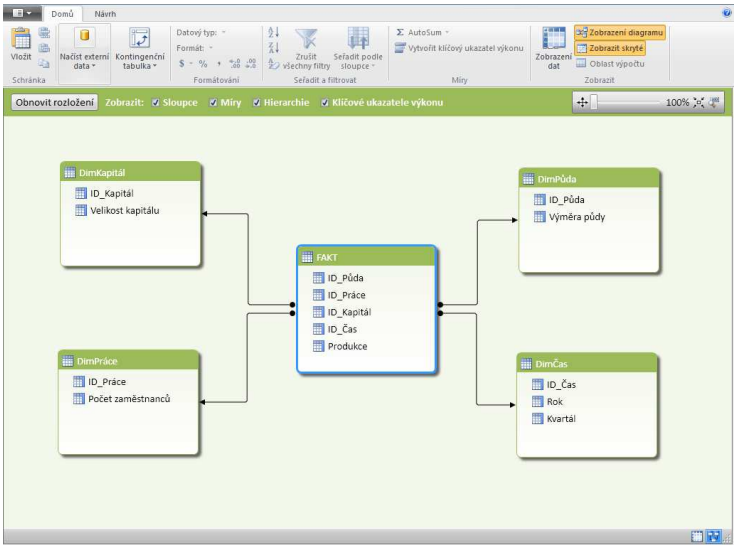
Obrázek č. 38 - Logické schéma prototypu (vlastní zpracování).

5.2.3. Fyzický návrh prototypu

Po vytvoření logického schématu je dalším krokem návrh fyzického schématu. Podstatou je doplnění logického modelu o fyzické charakteristiky, které jsou typické

pro danou technologii OLAP a konkrétní databázový systém. Ovšem při fyzickém návrhu prototypu nejsou důležitá optimální specifická nastavení navrhovaného databázového řešení, ale především možnost verifikace navrženého logického modelu. Proto je pro fyzický návrh prototypu použit Microsoft® Excel 2010 a doplněk PowerPivot (kapitola č. 4.4), který je pro tvorbu prototypu dostačující.

Nejprve jsou v doplňku PowerPivot integrována data tabulky faktů a jednotlivých dimenzí a vytvořeny relace podle navrženého logického schématu (obrázek č. 39). Integrovaná data nepředstavují konkrétní data zemědělského podniku. Data jsou pouze teoretická a představují středně velký zemědělský podnik v období 2010 – 2012 s průměrným počtem pracovníků v intervalu <3; 6>, kapitálem (mil. Kč) v intervalu <2,5; 4> a výměrou půdy (ha) v intervalu <90; 120>. Integrované tabulky obsahují atributy navržené v logickém schématu. Další atributy, především ty, které by mohly přidávat jednotlivým dimenzím hierarchii, nejsou do prototypu zahrnuty.



Obrázek č. 39 - Relace v PowerPivot pro prototyp (vlastní zpracování).

Fyzický přístup návrhu prototypu nejen umožní ověřit, že logický model navržený podle metody TEM je realizovatelný, ale také ověří, zda řešená problematika transformace ekonometrického modelu pro OLAP má praktický smysl. Především podoba integrovaných dat do doplňku PowerPivot je nejednoznačná. Proto je pro realizaci fyzického modelu prototypu předpokládáno několik variant integrovaných dat z produkčních zdrojů (provozních databází), mající vliv na interpretaci výstupů OLAP.

5.2.3.1. Integrovaná data - A

Varianta s označením A představuje integrovaná data, která odráží aktuální stav faktorů v zemědělském podniku. Například v 1. kvartálu 2012 je hodnota výměry zemědělského podniku na úrovni 125 ha. Ve 2. kvartálu 2012 je hodnota výměry 128 ha. Takto pořizovaná data do dimenzí odráží momentální skutečný stav faktorů v zemědělském podniku v době pořizování. Přehled obsahu jednotlivých tabulek je v příloze č. 2.

Maximální hodnota z Produkce	Čas							
	2010				Max 2010	2011	2012	Maximum
Půda	I. kvartál	II. kvartál	III. kvartál	IV. kvartál				
90	6 029	6 140			6 140			6 140
100			6 299		6 299			6 299
110				6 548	6 548	7 817		7 817
120						9 021	10 251	10 251
Maximum	**6 029**	**6 140**	**6 299**	**6 548**	**6 548**	**9 021**	**10 251**	**10 251**

Obrázek č. 40 - Kontingenční tabulka, faktor – čas (vlastní zpracování).

Pro praktické ověření prototypu multidimenzionální databáze je vytvořena v PowerPivot kontingenční tabulka (obrázek č. 40). Tento typ výstupu je většinou podporován všemi klientskými aplikacemi pro OLAP. V rámci fyzického návrhu je vybrána míra (produkce), která bude zobrazena uvnitř kontingenční tabulky a dimenze v řádcích a sloupcích (Půda a Čas) přes které jsou jejich hodnoty vypočteny. Vzhledem k charakteristice produkčního zdroje není možné provádět agregace typu součet. Například sumarizaci hodnot výsledné produkce pro 1. kvartál 2010 a 2. kvartál (6 029 + 6 140) nelze provést. Důvodem je, že výsledek produkce odráží aktuální stav disponibilních faktorů za dané období. Nelze tedy interpretovat výsledek tak, že velikost produkce za první dva kvartály je na úrovni 12 169 Kč (s. c.). Proto namísto sumarizace je použita maximalizace v následujícím tvaru:

```
=MAX('FAKT'[Produkce])
```

Takovýto přístup umožňuje přehledně zobrazit jednotlivé velikosti produkce podle vybraného faktoru a vývoje času. Ovšem pro ekonometrické analýzy je účelné, aby řešení OLAP umožnovalo analýzy typu faktor – faktor. To je v tomto přístupu možné pouze pro skutečné kombinace zaznamenané v tabulce faktů. Například na obrázku č. 41 je vidět, že matice výsledné kontingenční tabulky je řídká. V rámci analýzy faktor – faktor je možné sledovat množství faktoru použitého pro tvorbu konkrétní produkce. Například hodnota produkce 6029 Kč (s. c.) je vytvořena při použití 90 ha půdy

a 2,5 mil. Kč kapitálu. Ovšem není možné zjistit velikost produkce, které by bylo dosaženo při použití například 100 ha půdy.

Maximální hodnota z Produkce	Kapitál			
Půda	2500000	3000000	3500000	Maximum
90	6 029	6 140		6 140
100		6 299		6 299
110			7 817	7 817
120		9 021	10 251	10 251
Maximum	**6 029**	**9 021**	**10 251**	**10 251**

Obrázek č. 41 - Kontingenční tabulka, faktor – faktor (vlastní zpracování).

Tato varianta integrovaných dat je prakticky realizovatelná, ale její omezení spočívá v ekonomické analýze faktor – faktor, která je pro zemědělský podnik podnětná.

5.2.3.2. Integrovaná data - B

Data pořízená do dimenzí odráží změny oproti předchozímu stavu faktorů v zemědělském podniku. Například při prvním pořízení dat je výměra zemědělského podniku na úrovni 125 ha v 1. kvartálu 2012. Při dalším pořízení dat je hodnota výměry 3 ha ve 2. kvartálu 2012. To znamená, že došlo ke změně +3 ha oproti předchozímu období. Přehled obsahu použitých tabulek je uveden v příloze č. 2.

I když se v tomto produkčním zdroji, jeví data jako plně aditivní, implementace prototypu potvrdila opak. Data jsou neaditivní. Z matematické povahy ekonometrického modelu není možné provést agregaci dat za období 1. – 4. kvartálu, kvůli konstantě, která je v modelu zahrnuta. Takovéto agregace obsahují navýšení produkce o konstantu za každý kvartál. Sumarizace by odpovídala hodnotě 48,6 namísto 32,2.

Půda	Pr...	Kapitál	Rok	Kvartál	Produkce
90	3	3,5	2010	I. kvartál	29,6515
0	0	0,5	2010	II. kvartál	5,5066
10	0	0	2010	III. kvartál	7,9004
0	0	0,5	2010	IV. kvartál	5,5066
100	3	4,5	2011	I. kvartál	32,1959

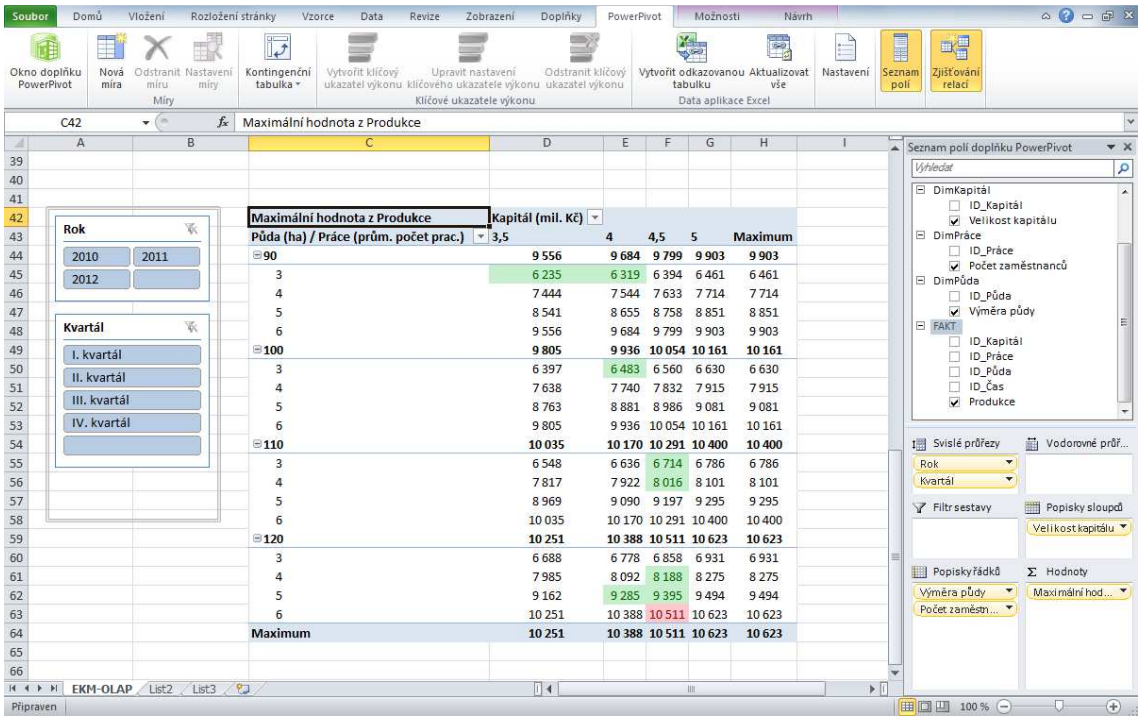
Obrázek č. 42 - Příklad tabulky faktů v PowerPivot (vlastní zpracování).

Nutné je také produkci počítat prostřednictvím linearizované produkční funkce. Důvodem je, že v případě nulové změny množství faktoru by nebylo možné mocninou funkci matematicky vypočítat. Autorem disertační práce je použití tohoto přístupu pořizování dat (formou diferencí) z výše uvedených důvodů zamítnuto.

5.2.3.3. Integrovaná data - C

U varianty C jsou data integrována tak, že data v dimenzích odráží aktuální stav faktorů v zemědělském podniku a zároveň jsou vytvářeny do tabulky faktů veškeré možné kombinace faktorů, které mohou teoreticky nastat. Podobně jako u varianty A odráží jednotlivé dimenze momentální stav zemědělského podniku v době pořizování. Ovšem na rozdíl od varianty A umožňuje tento typ integrovaných dat provádět ekonometrické analýzy typu faktor – faktor i pro teoretické kombinace faktorů. Podstatou řešení je, že se do tabulky faktů zaznamenávají nejen skutečné kombinace faktorů a příslušné vypočtené produkce, ale také veškeré možné kombinace faktorů, kterých může podnik dosahovat.

Na obrázku č. 43 je vytvořena kontingenční tabulka z dat uvedených v příloze č. 2. Popisky řádků kontingenční tabulky jsou tvořeny atributem Výměra půdy a Počet zaměstnanců. Popisek sloupců je tvořen atributem Velikost kapitálu. Vytvořená míra představuje maximum z produkce. Celým multidimenzionálním modelem dat je možné provádět řezy podle roku a kvartálu. V kontingenční tabulce je vytvořené pravidlo, které barevně vyznačí záznamy odpovídající reálným kombinacím faktorů použitých v zemědělském podniku. Červenou barvou je označena poslední vypočtená hodnota produkce na základě reálně použitého množství faktorů (hodnota 10 511 Kč). Zelenou barvou jsou označeny hodnoty produkce z předchozích období. Ostatní barvou neoznačené položky představují teoretické hodnoty produkce. Například hodnoty produkce 6 235 Kč je dosaženo při použití faktoru půdy o výměře 90 ha, průměrném počtu pracovníků rovných 3 a 3,5 mil. Kč kapitálu. Management zemědělského podniku může na základě takto vytvořené kontingenční tabulky odvodit, že při nákupu 10 ha půdy a konstantních ostatních faktorech je možné dosahovat produkce 6 397 Kč. Nebo při zvýšení průměrného počtu pracovníku na 4 by bylo dosahováno 7 444 Kč.



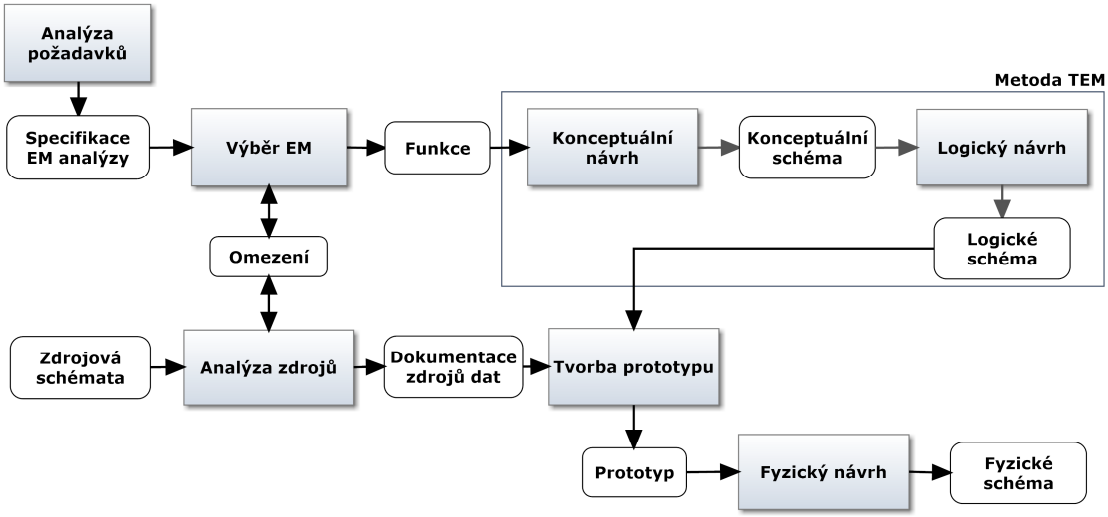
Obrázek č. 43 - Kontingenční tabulka v PowerPivot (vlastní zpracování).

Takovýto přístup umožní nalézt managementu zemědělského podniku kombinace množství faktorů, které vedou k přibližně shodné úrovni produkce. Identifikovat maximální hodnotu produkce ve sledovaném období. Odvodit procentní změnu množství jednoho faktoru při změně množství druhého faktoru a při konstantní úrovni produkce.

Na základě provedeného konceptuálního, logického a fyzického návrhu prototypu multidimenzionální databáze a již přijatých statistických hypotéz $H1$ a $H2$ je **přijímána hypotéza $H3$** o existenci přípustné transformace ekonometrického modelu do fyzického schématu relačního OLAP.

5.3. Metodika EM-OLAP

Na základě syntézy jednotlivých dílčích fází provedeného výzkumu disertační práce je představena nová ***metodika EM-OLAP*** *(ekonometrické modely pro online analytické zpracování dat)*. Metodika EM-OLAP podporuje návrh multidimenzionální databáze pro zajištění ekonometrických analýz za účelem zvýšení efektivnosti zemědělských podniků. Nová metodika je zaměřena na návrh multidimenzionální struktury z produkčních, nákladových anebo poptávkových funkcí v zemědělství a podporuje realizaci OLAP prostřednictvím relační databáze. Činnosti metodiky jsou vyobrazeny na obrázku č. 44.



Obrázek č. 44 - Metodika EM – OLAP (vlastní zpracování).

Tyto činnosti jsou prováděny postupně takto:

1. *Analýza požadavků*. V této etapě jsou zkoumány potřeby koncových uživatelů v kontextu ekonometrických analýz. Identifikuje se požadavek na typ ekonometrické analýzy v zemědělském podniku (analýza produkce, nákladů nebo poptávky). Identifikuje se požadavek na analýzu vztahů mezi činiteli výroby nebo výsledky výroby (vztah faktor-faktor, produkt-faktor apod.) Zařazují se požadavky na vyjádření charakteristik průběhu funkce (jednotková, mezní) a její pružnosti. Tato etapa může být v průběhu životního cyklu návrhu zpřesňována a končí spolu s fyzickým návrhem multidimenzionální databáze.

2. *Výběr EM*. Cílem etapy výběru ekonometrického modelu je volba vhodného typu ekonometrického modelu pro následné ekonometrické analýzy. Model

může být vyjádřen ve strukturální nebo redukované formě. Musí ovšem být již odhadnutý a použitelný pro ekonomickou interpretaci v zemědělském podniku. Všechny proměnné v modelu musí být jednoznačně interpretovatelné a musí být splněna podmínka, že je možné získat hodnoty těchto proměnných z produkčních zdrojů (provozních databází apod.) zemědělského podniku. Model může představovat produkční, nákladovou nebo například poptávkou funkci a může být vyjádřen v mocninném tvaru za předpokladu, že žádná z proměnných nemůže nabýt nulové hodnoty. Pokud tato podmínka není splněna, je nutné volit lineární podobu funkce. Výsledkem této fáze je přehled ekonometrických funkcí spolu s podrobnou dokumentací významu proměnných a charakteristik jejich průběhů.

3. *Analýza zdrojů*. V této etapě je třeba analyzovat různá schémata zdrojů dat a následně je sladit s cílem získat dokumentaci zdrojů dat k integraci. Tato fáze probíhá v součinnosti s fází výběru ekonometrického modelu. Je nutné znát ekonometrickou funkci, pro kterou budou data integrována a naopak je potřeba znát disponibilní zdroje dat pro výběr vhodné ekonometrické funkce. Výsledkem této fáze je dokumentace zdrojů dat k integraci do prototypu multidimenzionální databáze. Pro data určená k integraci je nutné splnit následující podmínky:

    a. Data určená k integraci do dimenzí musí odrážet aktuální stav faktorů v zemědělském podniku.

    b. Do tabulky faktů musí být vytvořeny skutečné kombinace faktorů, které tvoří výslednou hodnotu produkce zemědělského podniku a zároveň

    c. musí být v tabulce faktů vytvořeny veškeré možné kombinace faktorů, které mohou teoreticky nastat.

4. *Tvorba modelu dat.* Cílem této etapy je vytvořit konceptuální a logické schéma (podle metody TEM). Pro tvorbu *konceptuálního schématu* je nutné provést následující kroky:

    a. Vytvořit míry pro každou endogenní proměnnou z ekonometrického modelu (pravidlo 1.1).

b. Vytvořit dimenze pro každou exogenní proměnnou z ekonometrického modelu (pravidlo 1.2).

c. Pokud existuje časová proměnná v ekonometrickém modelu, pak je nutné vytvořit dimenzi času (pravidlo 1.3).

Pro tvorbu *logického schématu* je nutné provést následující krok:

a. Pokud existuje vztah mezi exogenní proměnnou a endogenní proměnnou, pak se vytvoří asociace prostřednictvím vygenerovaných klíčů mezi tabulkou faktů a dimenzemi (pravidlo 2.1).

5. *Tvorba prototypu.* V této fázi je vytvářen prototyp pro validaci navrženého logického schématu. Jsou integrována data do tabulky faktů a jednotlivých dimenzí. Vytvářeny jsou relace mezi tabulkou faktů a dimenzemi. Následně mohou být vytvářeny kontingenční tabulky, grafy a další výstupy pro OLAP. Při implementaci výpočtu míry je nutné zohlednit typ funkce ekonometrického modelu. Pokud modelovaná funkce je lineární a neobsahuje žádnou konstantu, pak je možné pro výpočet měr použít součet. V ostatních případech dávají ekonomický smysl pouze agregace typu maximum nebo průměr. Výsledné zobrazení dat v kontingenční tabulce musí umožňovat analýzu typu faktor – faktor. Pokud některá z výše uvedených činností skončila neúspěchem, nebo výsledek není validní podle analýzy požadavků, pak je nutné se vrátit zpět k přemodelování konceptuálního nebo logického schématu.

6. *Fyzický návrh.* Akceptace prototypu ukončuje proces navrhování logického schématu. V této fázi jsou stanoveny fyzické vlastnosti databáze založené na konkrétních poskytovaných funkcích databázového systému, jako je indexování, dělení a podobně.

5.4. Aplikace a schválení metodiky EM-OLAP

Přijatá metodika EM-OLAP byla aplikována v reálném prostředí zemědělského podniku. Požadavkem zemědělského podniku bylo provádět ekonometrické analýzy produkční funkce ekologického zemědělství se zohledněním vlivu dotací. Celá aplikace metodiky EM-OLAP v zemědělském podniku je založena na open source softwarovém řešení Pentaho Business Analytics, která je připojena k relační databázi MySQL.

Takovéto řešení nepřináší zemědělskému podniku další náklady na softwarové vybavení. Výsledná klientská aplikace umožňuje managementu zemědělského podniku získat informace o celkové produkci vyvíjené v čase prostřednictvím kontingenční tabulky a grafu. Provádět analýzu faktor – faktor, která je naimplementována, tak aby uživatel mohl volit různé kombinace faktorů (práce, půda, kapitál, přímé platby, dotace ekologického zemědělství, ostatní dotace). Označovány jsou v OLAP nástroji klíčového indikátory, které představují kombinace faktorů, které vedou k přibližně shodné úrovni produkce. Vytvářeny byly další speciální výstupy, které umožňují pro jednotlivé faktory získat informace o jednotkové produkci, mezní produkci a produkční pružnosti.

Výsledná nová metodika EM-OLAP a její aplikace byla schválena společností AgroKonzulta Žamberk spol. s r.o., která se v oblasti zemědělství pohybuje již přes 20 let a zabývá se nejen zemědělskou výrobou, ale i poradenstvím a vývojem software pro zemědělce. Zástupci společnosti AgroKonzulta Žamberk konstatují, že *„metodika EM-OLAP je samozřejmě v praxi použitelná. Provedené výpočty mohou sloužit k optimalizaci ekonomiky zemědělského podniku. Pokud jsou výpočty správně nastaveny, mohlo by nám to významně pomoci v bilancování ekonomiky podniků a navrhování různých modelových situací při rozhodování managementu.“*

6. Diskuze

V disertační práci byla představena nová metodika EM-OLAP, která byla autorem akceptovaná a nakonec aplikována a schválena v zemědělské praxi. Níže je posouzena validita výsledků jednotlivých fází metodiky disertační práce:

Návrh metody TEM (transformace ekonometrického modelu). Vzhledem k povaze použité metody analogie jako myšlenkového procesu je zřejmé, že závěry analogie jsou pouze pravděpodobné, nemají charakter nevyvratitelných tvrzení. Proto může existovat i jiná přípustná transformace ekonometrického modelu do konceptuálního a logického schématu.

Selekce variant návrhů TEM. Měření kvality navržených schémat byla provedena na základě vědeckých metod měření datových skladů podle (Serrano, et al., 2008) a (Gupta, et al., 2010). V rámci *empirického ověření* byly hodnoceny statistické hypotézy prostřednictvím dvouvýběrového testu průměrů. Výběrový soubor obsahoval 30 náhodně generovaných ekonometrických modelů splňující podmínku identifikace. Náhodnost byla řešena prostřednictvím generování opravdových náhodných čísel (na rozdíl od pseudonáhodných čísel generovaných typickými programovacími jazyky). Statistický software Statistica 10 předložil u obou testů hodnoty p výrazně nižší než zvolená hladina významnosti $\alpha = 0{,}05$ (i pro hladinu významnosti $\alpha = 0{,}01$).

Tvorba pravidel metody TEM. Formální zápis metody TEM byl vytvořen prostřednictvím matematického aparátu. Výsledná formální pravidla byla autorem prezentována v recenzovaném článku (Tyrychtr, a další, 2012).

Tvorba prototypu. Vytvoření prototypu konceptuálního a logického schématu (podle metody TEM) a následné vytvoření fyzického schématu multidimenzionální databáze umožnilo přijetí hypotézy $H3$. Pro návrh fyzického schématu bylo experimentováno s různými variantami integrovaných dat. Všechny varianty byly založené na datech produkční funkce. Je zřejmé, že právě fyzický návrh vykazoval možnost identifikace odlišných přístupů v případě návrhu jiného typu ekonometrického modelu. V dalším výzkumu je vhodné zohlednit návrh fyzického přístupu v kontextu nákladové a poptávkové funkce.

Akceptace. Autorem byla na základě vytvořeného prototypu akceptována varianta podporující ekonometrické analýzy typu faktor – faktor i pro teoretické kombinace faktorů.

Aplikace. V disertační práci byla schválena metodika EM-OLAP a její aplikace v zemědělském podniku. Ovšem výsledná metodika je ověřena pouze na jednom zemědělském podniku. Překážkou aplikování metodiky na více typů zemědělských podniků byla především vysoká nákladnost a časová pracnost realizace celkového řešení EM-OLAP. Na základě stanovených formalizovaných pravidel TEM je v současnosti autorem vyvíjen software pro automatizaci celého procesu návrhu EM-OLAP. Tento software si klade za cíl zefektivnit vývoj EM-OLAP snížením potřebného času na návrh a dosáhnout tak snížení počátečních nákladů do řešení EM-OLAP v zemědělském podniku. Další překážkou validace metodiky EM-OLAP je její srozumitelnost a náročnost, která nebyla exaktně měřena. Autorem bude v dalším výzkumu měřena jak srozumitelnost metodiky EM-OLAP, tak její koheze, ekonomický dopad a také testování účinnosti použité technologie na výsledné řešení. Postup návrhu EM-OLAP je také podnětné v dalším výzkumu integrovat do stávajících metod návrhu, tak aby bylo možné paralelně navrhovat jak klasické OLAP systémy tak systémy EM-OLAP. Dopad ostatních komponent a celkových přístupů návrhů v business intelligence na EM-OLAP není v metodice exaktně vyjádřen. I když metodika EM-OLAP vychází z přístupů návrhu business intelligence, je přesto nutné tuto problematiku dále podrobně zkoumat.

I přes některé překážky validace metodiky EM-OLAP umožňuje výsledná metodika EM-OLAP provádět ekonometrické analýzy v zemědělském podniku bez jejich hlubokých znalostí managementem zemědělského podniku. V disertační práci byl především kladen důraz na aplikaci produkční funkce pro OLAP. Autor tedy může konstatovat, že hlavní cíl disertační práce byl splněn. Je patřičné neopomenout, že výsledná metodika je zobecnitelná i na technologii MOLAP. Zaměření cíle na ROLAP vychází ze syntézy literární rešerše. Autor se domnívá, že ROLAP jako tabulkově orientovaný model bude postupně nahrazovat MOLAP systémy, které v současnosti poskytují rychlejší odezvu, ale menší škálovatelnost. Vzhledem k novým provedeným výzkumům a inovativním přístupům v návrhu indexů je zřejmé, že účinnost ROLAP se začne s MOLAP srovnávat.

7. Závěr

Skutečnost, že zemědělské podniky nedosahují plné produkční síly a technické efektivnosti, byla motivací zvoleného tématu předkládané disertační práce. Autor na základě literární rešerše identifikoval mezery současného stavu poznání v oblasti návrhu multidimenzionálních modelů dat s akcentací na zemědělství. Smyslem bylo prostřednictvím kritického přehledu, hledat takové přístupy, které mohou managementu zemědělského podniku pomoci interpretovat právě ta data, z kterých je možné získat relevantní poznatek o jejich ekonomické výkonosti. Na základě výsledků syntézy literární rešerše byla položena výzkumná otázka, zdali *je možné vytvořit metodiku návrhu multidimenzionální databáze pro ekonometrické analýzy zemědělského podniku?*

Výsledky literární rešerše, vymezená oblast zkoumání a stanovená výzkumná otázka umožnily *identifikovat* hlavní a dílčí cíle disertační práce spolu s odvozenými hypotézami. Pro splnění cílů a ověření hypotéz byl použit následující metodický přístup:

Návrh TEM (Transformace Ekonometrického Modelu). V této fázi byla vytvořena analogie ekonometrického modelu s multidimenzionálním paradigmatem, tak aby bylo možné transformovat ekonometrický model do konceptuálního a logického schématu multidimenzionálního modelu dat. Byly identifikovány dvě možné varianty transformace multidimenzionálního schématu s ekvivalentní sémantikou.

Podle *prvé varianty* byla z rovnic ekonometrického modelu transformována endogenní proměnná y_s na tabulku faktů. Tedy y_s představovala tabulku faktů. Exogenní proměnné x_r pak dimenze. U ekonometrických modelů, které obsahovaly časovou proměnnou *t*, pak schéma obsahovalo ještě dimenzi času. Tabulka faktů byla asociována vztahem roll-up se všemi příslušnými dimenzemi podle proměnných na pravé straně rovnice. Celý zápis rovnice představoval míru, jako sledovaný ukazatel tabulky faktů. V rámci převodu do logického schématu byla každá dimenze opatřena číselným primárním klíčem a asociována prostřednictvím cizího klíče s tabulkou faktů. Výsledný způsob transformace byl vyobrazen na obrázku č. 33. Takto vzniklé schéma je fyzicky realizovatelné pouze při použití ROLAP technologie. V případě MOLAP realizace není možné propojit mezi s sebou datové kostky. V případě použití tohoto způsobu

transformace ekonometrického modelu do multidimenzionálního schématu je vhodné simultánní ekonometrický model převést nejprve do redukované formy.

Ve *druhé variantě* byla transformace obdobná. Došlo především k upravení tvrzení, že y_t představuje tabulku faktů. Uvažována byla pouze existence jedné tabulky faktů ve schématu. Každá rovnice v modelu pak představovala jednu míru tabulky faktů. Výsledek takového způsobu transformace byl vyobrazen na obrázku č. 34.

Selekce variant návrhů TEM. Na základě navržených přístupů k transformaci ekonometrického modelu do multidimenzionálního schématu byla provedena měření kvality navržených schémat. Měření kvality a srozumitelnosti byla provedena na základě vědeckých metod měření datových skladů. Hodnoty měr měření (tabulka č. 4 a č. 5) potvrdily, že druhá varianta je z hlediska kvality a srozumitelnosti pro návrh transformace ekonometrického modelu do konceptuálního a logického modelu dat vhodnější. Vzhledem k faktu, že výše uvedené hodnocení proběhlo pouze na jednom schématu, nebylo možné přijmout tvrzení, že druhá varianta TEM je vhodnější.

Proto byly následně ***empiricky*** ověřovány hypotézy $H1$ a $H2$ (o existenci významného rozdílu v hodnotách míry měření srozumitelnosti a kvality mezi schématy navrženými z analogie) prostřednictvím dvouvýběrového testu průměrů. Výběrový soubor obsahoval 30 náhodně generovaných ekonometrických modelů splňující podmínku identifikace (příloha č. 3). Smyslem generování ekonometrických modelů bylo simulovat náhodné formální struktury, které mohou v reálných podmínkách vzniknout. Pro vytvoření souboru náhodných generovaných ekonometrických modelů byl v disertační práci vytvořen algoritmus zapsaný v pseudokódu (obrázek č. 37) na jehož principech byl vytvořen generátor ekonometrických modelů (obrázek č. 36).

Výsledkem simulace a měření byly statisticky podložená rozhodnutí, jaký z přístupu je vhodný pro transformaci ekonometrického modelu. Na základě provedeného statistického testování (softwarem Statistica 10) byla **přijata** hypotéza $H1$ a $H2$. Bylo tedy možné konstatovat, že s 99% pravděpodobností byl prokázán statisticky významný rozdíl v *míře kvality a srozumitelnosti modelu dat* mezi schématy navrženými z analogie. To umožnilo přijmout tvrzení, že druhá varianta transformace ekonometrického modelu je vhodnější.

Tvorba pravidel metody TEM. Formální zápis nové metody TEM byl vytvořen prostřednictvím matematického aparátu, který přesně vymezuje pravidla jejího použití.

V prvé fázi je vytvářeno základní schéma typu hvězda podle pravidel:

Pravidlo 1.1: Vytvoření měr do prázdného schématu pro každou endogenní proměnnou v ekonometrickém modelu je definován vztahem:

$\forall y_s \in Y : m_s \in Measure$ a $\forall y_{st} \in Y : m_{st} \in Measure$

Pravidlo 1.2: Vytvoření dimenzí do schématu pro každou exogenní proměnnou v ekonometrickém modelu je definován vztahem:

$\forall x_r \in X : d_s \in Dim$ a $\forall x_{rt} \in X : d_{rt} \in Dim$

Pravidlo 1.3: Pokud existuje časová proměnná v ekonometrickém modelu, pak se vytvoří dimenze času podle vztahu:

$\forall x_{rt} \in X : d_{rt} \in Dim_{time}$

Ve druhé fázi jsou vytvářeny vztahy mezi dimenzemi a tabulkou faktů ve schématu hvězda podle pravidla:

Pravidlo 2.1: Pokud existuje vztah mezi exogenní proměnnou x a endogenní proměnnou y a funkce $getKey$, která vrací množinu klíčů těchto proměnných, pak se vytvoří asociace mezi faktem a dimenzí, pro kterou platí:

$$\forall (x, y) \in Rel: (d, c, K) | (d \in Dim) \wedge (c \in Fact) \wedge ((d, c) \in Ass) \wedge$$
$$(K \subseteq K_d \cup K_c | (K_d = getKey(d)) \wedge (K_c = getKey(c)))$$

Tvorba prototypu. Pro ověření hypotéz $H3$ (o existenci přípustné transformace ekonometrického modelu do fyzického schématu relačního OLAP) a pro získání výhod a omezení tvorby nové metodiky EM-OLAP byl vytvořen prototyp multidimenzionální databáze a aplikace OLAP pro zajištění ekonometrických analýz. Prostřednictvím nové metody TEM byly vytvořeny prototypy konceptuálních a logických schémat. Pro návrh fyzického schématu bylo experimentováno s různými podobami integrovaných dat. Varianta s označením A představovala integrovaná data, která odrážela aktuální stav faktorů v zemědělském podniku. U varianty typu B odrážela data změny oproti předchozímu stavu faktorů v zemědělském podniku. U varianty C data v dimenzích

odrážela aktuální stav faktorů v zemědělském podniku a zároveň byly do tabulky faktů zaznamenávány teoretické kombinace faktorů.

Vytvořením fyzického prototypu multidimenzionální databáze a aplikace OLAP umožnilo **přijetí** hypotézy $H3$.

Akceptace. Cílem této fáze bylo systematické experimentování s dosaženým postupem návrhu metodiky EM-OLAP. Za vyhovující byla vybrána varianta typu C, podle které byla data integrována tak, že data v dimenzích odrážela aktuální stav faktorů v zemědělském podniku a zároveň byly vytvářeny do tabulky faktů veškeré možné kombinace faktorů, které mohly teoreticky nastat. Jednotlivé dimenze tak odráží momentální stav faktorů zemědělského podniku a zároveň je možné provádět ekonometrické analýzy typu faktor – faktor i pro teoretické kombinace faktorů. Podstatou takového řešení je, že se do tabulky faktů zaznamenávají nejen skutečné kombinace faktorů a příslušné vypočtené míry, ale také veškeré možné kombinace faktorů, kterých může podnik dosahovat.

Aplikace. Přijatá metodika EM-OLAP byla *aplikována* v reálném prostředí zemědělského podniku. Požadavkem zemědělského podniku bylo provádět ekonometrické analýzy produkční funkce ekologického zemědělství se zohledněním vlivu dotací. Celá aplikace EM-OLAP byla založena na softwaru Pentaho Business Analytics a na relační databázi MySQL. Výsledná klientská aplikace následně umožňovala managementu zemědělského podniku získat informace o celkové produkci a provádět analýzy faktor – faktor.

V poslední fázi metodiky disertační práce byla cílová metodika EM-OLAP a její aplikace ***schválena*** společností AgroKonzulta Žamberk spol. s r.o., která se v oblasti zemědělství pohybuje již přes 20 let a zabývá se nejen zemědělskou výrobou, ale i poradenstvím a vývojem software pro zemědělce.

Na základě syntézy jednotlivých dílčích fází metodického přístupu disertační práce byl splněn hlavní cíl disertační práce a představena nová **metodika EM-OLAP** *(ekonometrické modely pro online analytické zpracování dat).* Metodika EM-OLAP podporuje návrh multidimenzionální databáze pro zajištění ekonometrických analýz za účelem zvýšení efektivnosti zemědělských podniků. Metodika je zaměřena na návrh multidimenzionální struktury z produkčních, nákladových anebo poptávkových funkcí

v zemědělství a podporuje realizaci OLAP prostřednictvím relační databáze. Jednotlivé fáze metodiky EM-OLAP (obrázek č. 44) jsou souhrnně vyjádřeny takto:

1. *Analýza požadavků*. V prvé fázi jsou analyzovány potřeby koncových uživatelů v kontextu ekonometrických analýz.

2. *Výběr ekonometrického modelu.* Ve druhé fázi je vybírán vhodný typ ekonometrického modelu pro následné ekonometrické analýzy.

3. *Analýza zdrojů*. V této fázi jsou analyzována disponibilní schémata zdrojů dat s cílem získat dokumentaci zdrojů dat k integraci do multidimenzionální databáze.

4. *Tvorba modelu dat.* V této fázi je vytvářeno konceptuální a logické schéma multidimenzionální databáze (podle metody TEM).

5. *Tvorba prototypu.* Na základě provedené analýzy zdrojů a vytvořených modelů dat je vytvářen prototyp pro validaci navrženého logického schématu.

6. *Fyzický návrh.* Akceptace prototypu ukončuje proces navrhování logického schématu. V tomto kroku jsou stanovovány fyzické vlastnosti databáze.

Navržená metodika EM-OLAP poskytuje návrhářům metodický rámec pro návrh struktury multidimenzionálních databází pro zajištění ekonometrických analýz. Na tomto přístupu založené klientské aplikace OLAP umožní managementu zemědělského podniku získávat analytická data prostřednictvím kontingenčních tabulek, grafů a dalších speciálních výstupů. Sledovat informace o celkové produkci, nákladech nebo spotřebě vyvíjené v čase a to za celý podnik nebo jeho část. Klíčovým přínosem je možnost provádět analýzu faktor – faktor, kterou je možné implementovat způsobem, který uživateli umožní zvolit různé kombinace faktorů (například počet pracovníků, velikost obhospodařované plochy, množství kapitálů, velikost dotací apod.). Sledovat kombinace faktorů, které vedou k přibližně shodné úrovni produkce a vytvářet další speciální výstupy, které poskytnou pro jednotlivé faktory další ekonomické informace (například o jednotkové produkci, mezní produkci nebo produkční pružnosti).

Vzhledem k faktu, že podobný výzkum týkající se ekonometrických analýz prostřednictvím OLAP nebyl proveden, výsledky a přínosy disertační práce přinášejí nové poznatky do oblasti návrhu multidimenzionálních databází.

Praktickým přínosem disertační práce je právě analýza faktor – faktor, která je pro zemědělský podnik považována za podnětnou. Důvodem je, že ekonomické výsledky zemědělských podniků je možné ovlivňovat vzájemnou substitucí výrobních faktorů, která umožňuje reagovat na jejich cenové výkyvy. Hledat optimální kombinace faktorů pro maximalizaci produkce nebo naopak minimalizaci nákladů a pomoci tak zvýšit efektivnost zemědělských podniků.

Přínosem pro vědu je především vytvořená:

- nová metoda TEM a její formální pravidla, která jsou použitelná pro další vědecký rozvoj (například v oblasti automatizace transformace ekonometrických modelů).
- nová metodika EM-OLAP, která odráží výsledky výzkumu disertační práce a současné poznatky z oblasti návrhů multidimenzionálních databází. Metodika poskytuje teoretický rámec pro další zpřesňování a integraci do stávajících přístupů (například paralelní návrh provozních a analytických modelů dat).

Závěrem disertační práce je vhodné zopakovat, že předmětem nové metodiky není konstrukce ekonometrického modelu, ale jeho transformace do multidimenzionální databáze. Vstupním prvkem metodiky jsou právě ekonometrické rovnice, které jsou prostřednictvím metody TEM transformovány do konceptuálního a logického schématu. Spolu s analýzou zdrojů dat je hledán právě takový prototyp, který umožní realizovat požadavky ekonometrických analýz. Výstupem metodiky EM-OLAP je fyzické schéma multidimenzionální databáze založené na ekonometrických modelech v kontextu zemědělství, které je použitelné pro online analytické zpracování dat.

Citovaná literatura

Abdullah, Ahsan. 2009. Analysis of mealybug incidence on the cotton crop using ADSS-OLAP (Online Analytical Processing) tool. *Comput. Electron. Agric.* s.l. : Elsevier Science Publishers B. V., 2009. Vol. 69, 1. 0168-1699.

Abdullah, Ahsan, Brobst, Stephen and Pervaiz, Ijaz. 2003. Agri Data Mining/Warehousing:Innovative Tools for Analysis of Integrated Agricultural & Meteorological Data. 2003.

Abelló, Alberto and Romero, Oskar. 2009. On-Line Analytical Processing. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer, 2009. pp. 1949-1954. 978-0-387-39940-9.

Abiteboul, S., Hull, R. and Vianu, V. 1995. Foundations of Databases. *Addison-Wesley.* MA, USA : s.n., 1995.

Ailamaki, Anastasia and Pandis, Ippokratis. 2009. Query Processor. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 2307-2308. 978-0-387-39940-9.

Antunes, J. F. G., Rodrigues, L. H. A. and Oliveira, S. R. de M. 2011. Data mining for sugarcane crop classification using MODIS data. [ed.] K. Charvát E. Gelb. *EFITA/WCCA '11.* Prague : Czech Centre for Science and Society, 2011. 978-80-904830-3-3.

Arenas, Marcelo. 2009. Normal Forms and Normalization. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 1917-1920. 978-0-387-35544-3.

Armstrong, W. W. 1974. Dependency structures of data base relationships. *Proc. IFIP Congress 74.* 1974. pp. 580–583.

Artale, A., et al. 2007. Reasoning over Extended ER Models. 2007. pp. 277-292. 3-540-75562-4 978-3-540-75562-3.

Astrahan, M.M., et al. 1976. System R: relational approach to database management. *ACM Transactions on Database.* 1976. Vol. 1, 2. 97-137.

Atkinson, M., et al. 1989. The object-oriented database system manifesto. 1989.

Bækgaard, L. 1999. Event-Entity-Relationship Modeling in Data Warehouse Environments. *Proceedings of the ACM DOLAP99 Workshop.* Missouri : s.n., 1999.

Bachman, C. W. 1969. Data Structure Diagrams. *Database.* 1969, Vol. 1, 2, pp. 4-10.

Ballard, C., et al. 1998. Data Modeling Techniques for Data Warehousing, , l. *IBM Redbook.* s.l. : IBM International Technical Support Organization, 1998. 0738402451.

Batini, C., Lenzerini, M. and Navathe, S. B. 1986. A Comparative Analysis of Methodologies for Database Schema Integration. s.l. : ACM COMPUTING SURVEYS, 1986. Vol. 18, 4, pp. 323-364.

Beeri, C., Fagin, R. and Howard, J. H. 1978. A complete axiomatization for functional and multivalued dependencies. *Proc. ACM SIGMOD Int. Conf. on Management of Data.* 1978. pp. 47–61.

Berndtsson, M. and Mellin, J. 2009. Active Database, Active Database (Management) System. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. 978-0-387-39940-9.

Beynon-Davies, P. 2004. *Database Systems 3rd Edition.* s.l. : Palgrave, Basingstoke, UK, 2004. 1-4039-1601-2.

Boehnlein, M. and Ende, A.U. 1999. Deriving initial data warehouse structures from the conceptual data models of the underlying operational information systems. *Proceedings of DOLAP.* Kansas City, USA : s.n., 1999.

Burstein, F. and Holsapple, C.W. 2008. Handbook on Decision Support Systems. *International Handbooks on Information Systems.* s.l. : Springer, 2008. Vol. 1. 9783540487128.

Cabibbo, L. and Torlone, R. 1998. From a procedural to a visual query language for OLAP. *Proc. 10th Int. Conf. on Scientific and Statistical Database Management.* 1998. pp. 74-83.

Calero, C., et al. 2001. Towards DW Quality metrics. *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW 2001).* Interlaken, Switzerland : s.n., 2001.

Cattell, R. G.G., et al. 2000. *The Object Data Management Standard: ODMG 3.0.* s.l. : Morgan Kaufmann, 2000. 1-55860-647-5.

Cipra, Tomáš. 2008. *Finanční ekonometrie.* Praha : Ekopress, 2008, 2008. 978-80-86929-43-9.

CODASYL. 1971. CODASYL Data Base Task Group Report. s.l. : CODASYL, ACM, 1971.

Codd, E. F. 1970. A relational model for large shared data banks. *Commun.* s.l. : ACM, 1970. Vol. 13, 6. 377-487.

—. 1985(a). Does Your DBMS Run By the Rules. s.l. : ComputerWorld, 21 October 1985(a).

—. 1972(a). Further normalization of the data base relational model. *Data base systems.* Englewood Cliffs, NJ, USA : Prentice-Hall, 1972(a). pp. 33–64.

—. 1985(b). Is Your DBMS Really Relational? s.l. : ComputerWorld, 14 October 1985(b).

—. 1974. Recent investigations in relational data base systems. *IFIP Congress.* North-Holland, Amsterdam : s.n., 1974. pp. 1017–1021.

—. 1972(b). Relational Completeness of Database Sublanguages. [ed.] R. Rustin. *Courant Computer Science Symposium 6: Data Base Systems.* Englewood Cliffs, NJ : Prentice-Hall, 1972(b).

—. 1990. *The Relational Model for Database Management: Version 2.* s.l. : Addison-Wesley, 1990. p. 538. 9780201141924.

Codd, E. F., Codd, S. B. and Salley, C. T. 1993. Providing OLAP (On-line Analytical Processing). San Jose : Codd & Date, Inc, 1993.

Coelli, T.J., et al. 2005. *An Introduction to Efficiency and Productivity Analysis.* 2. ed. s.l. : Springer, 2005. Vol. XVII. 978-0-387-24265-1.

Conolly, Thomas, Begg, Carolyn a Holowczak, Richard. 2009. *Mistrovství - databáze, Profesionální průvodce tvorbou efektivních databází.* Brno : Computer Press, a. s., 2009. 978-80-251-2328-7.

Copeland, G. and Maier, D. 1984. Making smalltalk a database system. 1984. Vol. 4 (1), pp. 107-131.

Correa, F. E., et al. 2011. Data Mining Applied on Grain Data Mart. *EFITA/WCCA '11.* Prague : Czech Centre for Science and Society, 2011. 978-80-904830-3-3.

Čechura, Lukáš. 2010. Estimation of technical efficiency in Czech agriculture with respect to firm heterogeneity. *Agricultural Economics (Zemědělská ekonomika), roč. 56, č. 4.* 2010. pp. 183 - 191. 0139-570X.

Čechura, Lukáš, a další. 2009. Cvičení z ekonometrie. Praha : Česká zemědělská univerzita v Praze, Provozně ekonomická fakulta, 2009. str. 102. 978-80-213-1976-9.

ČSN ISO 5127 (01 0162). 2003. Informace a dokumentace - slovník. Praha : Český normalizační institut, 2003.

da Silva, Joel, et al. 2010. Modelling and querying geographical data warehouses. *Information Systems.* 2010. Vol. 35, 5. 0306-4379.

Datta, Anindya and Thomas, Helen. 1999. The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. *Decision Support Systems.* 1999, Vol. 27, 3, pp. 298-301.

Doka, Katerina, Tsoumakos, Dimitrios and Koziris, Nectarios. 2011. Online querying of d-dimensional hierarchies. *Journal of Parallel and Distributed Computing.* s.l. : Elsevier, 2011. Vol. 71, 3. 0743-7315.

Duchoň, B. 2007. *Inženýrská ekonomika.* Beckovy ekonomické učebnice. místo neznámé : Nakladatelství C H Beck, 2007. 9788071797630.

Earley, Susan. 2010. *The DAMA Guide to the Data Management Body of Knowledge.* s.l. : Technics Publications LLC, 2010. 9781935504023..

Eavis, Todd and Taleb, Ahmad. 2012. Towards a scalable, performance-oriented OLAP storage engine. *Proceedings of the 17th international conference on Database Systems for Advanced Applications - Volume Part II.* Busan, South Korea : Springer-Verlag, 2012. 978-3-642-29034-3.

Elbashir, M., Collier, P. A. and Davern, M. J. 2008. Measuring the effects of business intelligence systems: The relationship between business process and organizational performance. [ed.] S. G. Sutton. *International Journal of Accounting Information Systems.* New York, NY : Elsevier Science , 2008. Vol. 9, 3, pp. 135–153. 1467-0895 .

Elmasri, R. and Navathe, S. 1994. Fundamentals of Database Systems. s.l. : Benjamin Cummings, 1994.

Embley, David W. 2009(a). Relational Model. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* 2009(a). pp. 2372-2376. 978-0-387-39940-9.

—. 2009(b). Semantic Data Model. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(b). pp. 2559-2561. 978-0-387-39940-9.

Fagin, R. 1981. A normal form for relational databases that is based on domians and keys. s.l. : ACM Trans. Database Syst., 1981. Vol. 6, 3, pp. 387–415.

—. 1977. Multivalued dependencies and a new normal form for relational databases. *Database Systems.* s.l. : ACM Trans, 1977. Vol. 2, 3, pp. 262–278.

—. 1979. Normal forms and relational database operators. *Proc. ACM SIGMOD Int. Conf. on Management of Data.* 1979. pp. 153–160.

Fahrner, C. and Vossen, G. 1995. A survey of database design transformations based on the Entity-Relationship model. *Data and Knowledge Engineering.* 1995, Vol. 15, 3, pp. 213-250.

Feuerlicht, G., Pokorný, J. and Richta, K. 2009. Object-Relational database design: Can your application benefit from SQL:2003? *ISD Challenges in Practice, Theory, and Education.* Galway : Springer Scence+ Business Media, LLC, 2009. pp. 975-988. 978-0-387-78577-6.

Filipe, J. and Adams, G. 2005. The Estimation of the Cobb Douglas Function. *Eastern Economic Journal.* 2005. Vol. 31, 3, pp. 427–445.

Fišer, B. and Brezany, P. 2004. Approaches to the Development of OLAP Systems. *Gridminer Tr.* Vienna : University of Vienna, 2004.

Franconi, E. and Sattler, U. 1999. A Data Warehouse Conceptual Data Model for Multidimensional Aggregation . *In Proceedings of the Workshop on Design and Management of Data Warehouses (DMDW'99).* 1999.

Gehrke, Johannes. 2009. DBMS Component. [ed.] Ling Liu and M. Tamer Özsu. s.l. : Springer US, 2009. p. 755. 978-0-387-39940-9.

Genero, M., Poels, G. and Piattini, M. 2007. Defining and validating metrics for assessing the understandability of entity-relationship diagrams. October 2007.

Geppert, A. and Berndtsson, M., [ed.]. 1997. s.l. : Springer, 1997. Proc. 3rd International Workshop on Rules in Database Systems. Vol. 1312.

Golfarelli, M. and Rizzi, S. 1998. A Methodological Framework for Data Warehouse Design. *Proceedings of the ACM DOLAP98 Workshop.* Washington, D.C. : s.n., 1998.

—. 2011. Data warehouse testing: A prototype-based methodology. *Information and Software Technology.* Bologna, Italy : Elsevier B.V., 2011. Vol. 53.

Golfarelli, M., et al. 2006. Schema versioning in data warehouses: Enabling cross-version querying via schema augmentation. *Data Knowledge Engineering.* s.l. : Elsevier, 2006. Vol. 59, 2. 0169-023X.

Gray, J., et al. 1997. Data cube: A relational aggregation operator generalizing group-by, cross-tab and subtotals. *Data Mining Knowl. Dis.* 1997. Vol. 1, 1, pp. 29–54.

Group, Isrd. 2005. *Introduction to Database Management Systems.* s.l. : Tata McGraw-Hill Education, 2005. 9780070591196.

Gupta, Amarnath. 2009. Multidimensional Data Formats. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 1776-1777.

Gupta, Rolly and Gosain, Anjana. 2010. Analysis of Data Warehouse Quality Metrics Using LR. Communications in Computer and Information Science *Information and Communication Technologies.* Berlin Heidelberg : Springer, 2010. Vol. 101. 978-3-642-15766-0.

Gyssens, M. and Lakshmanan, L. V. S. 1997. A foundation for multidimensional databases. *Proc. 23rd Int. Conf. on Very Large Data Bases.* 1997. pp. 106-115.

Hahn, K., Sapia, C. and Blaschka, M. 2000. Automatically Generating OLAP Schemata from Conceptual Graphical Models. *Proceedings of the ACM DOLAP 2000 Workshop.* New York, NY, USA : ACM, 2000. 1-58113-323-5.

Hainaut, Jean-Luc. 2009(a). Hierarchical Data Model. [ed.] Ling Liu and M. Tamer Özsu. s.l. : Springer US, 2009(a). pp. 1294-1300. 978-0-387-39940-9.

—. 2009(b). Network Data Model. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(b). pp. 1900-1906. 978-0-387-39940-9.

Hammer, M. and McLeod, D. 1981. Database description with SDM: a semantic database model. s.l. : ACM Transactions on Database Systems, 1981. Vol. 3, 6, pp. 351-386.

Helland, Pat. 2009. Database Management System. [ed.] Ling Liu and M. Tamer Özsu. s.l. : Springer US, 2009. pp. 714-719. 978-0-387-39940-9.

Hellerstein, J.M., Stonebraker, M. and Hamilton, J. 2007. Architecture of a Database System. *Foundations and Trends in Databases.* 2007, Vol. 1, pp. 141-259.

Chamberlin, D.D and Boyce, R.F. 1974(a). SEQUEL: A structured english query language. *ACM SIGFIDET Workshop.* 1974(a). pp. 249-264.

Chamberlin, D.D. 2009. SQL. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* 2009. pp. 2753-2760. 978-0-387-39940-9.

Chamberlin, D.D. and Boyce, R.F. 1974(b). SEQUEL: A structured English query language. *Proceedings of the 1974 ACM SIGMOD Workshop on Data Description.* s.l. : Access and Control, 1974(b). 249–264.

Chamberlin, D.D., et al. 1981. A history and evaluation of system R, Communications of the ACM. 1981. Vol. 24, 10. 632–646.

Charvat, K. (junior), Gnip, P. and Charvat, K. 2011. COIN – Module for Tactical planning of agriculture production. *EFITA/WCCA ´11.* Prague : Czech Centre for Science and Society, 2011. 978-80-904830-3-3.

Charvat, K. and Gnip, P. 2011. Future Farm for a Farm of the Future. *EFITA/WCCA ´11.* Prague : Czech Centre for Science and Society, 2011. 978-80-904830-3-3.

Chaudhuri, S. and Dayal, U. 1997. An Overview of Data Warehousing and OLAP Technology. *ACM SIGMOD Rec.* 1997. Vol. 26, 1, pp. 65-74.

Chen, P. P. 2002. Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned. [ed.] M. Broy and E. Denert. *Software Pioneers: Contributions to Software Engineering.* June 2002, pp. 100-114.

—. 1976. The entity-relationship model—toward a unified view of data. s.l. : ACM Transactions on Database Systems, 1976. Vol. 1, 1. 9-36.

Chen, Yen-Ting and Hsu, Ping-Yu. 2007. A grain preservation translation algorithm: From ER diagram to multidimensional model. *Information Sciences.* s.l. : Elsevier, 2007. Vol. 177, 18. 0020-0255.

Chu, Andrej. 2006. Matematické základy informatiky. [Online] 2006. [Citace: 6. září 2012.] hroch.sk/skola/mzi/download/1_pseudokod.doc.

Churcher, Clare. 2008. Beginning SQL Queries From Novice to Professional. Berkeley : Apress, 2008. 978-1-4302-0550-0.

IMS/360, Information Management System. 1971. Application Description Manual GH20-0765-1. New York : IBM Corporation, White Plains, 1971.

Inmon, W.H. 2002. *Building the DataWarehouse.* New York : Wiley, 2002. 0-471-08130-2.

Jackson, M. 1999. Thirty years (and more) of databases. *Information and Software Technology.* 1999, Vol. 41, 14, pp. 969-978.

Jensen, C.S. and Snodgrass, R.T. 2009. Temporal Database. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 2957-2960. 978-0-387-39940-9.

Jensen, C.S., Pedersen, T.B. and Thomsen, C. 2010. *Multidimensional Databases and Data Warehousing.* Synthesis Lectures on Data Management. s.l. : Morgan & Claypool Publishers, 2010. 9781608455379.

Kalnisn, Panos and Papadias, Dimitris. 2003. Multi-query optimization for on-line analytical processing. *Information Systems.* 2003, Vol. 28, pp. 457–473.

Kamfonas, M. J. 1992. Recursive Hierarchies: The Relational Taboo! *The Relation Journal.* 1992.

Karvounarakis, G. 2009. Datalog. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 751-754. 978-0-387-39940-9.

Kaufmann, M. 1993. The Object Database Standard: ODMG-93. [ed.] R.G.G. Catell. Los Altos, CA : s.n., 1993.

Kavka, M., et al. 2011. Information system of agricultural production technologies based on standard database. *EFITA/WCCA '11.* Prague : Czech Centre for Science and Society, 2011. 978-80-904830-3-3.

Khan, Arshad. 2005. *SAP and BW Data Warehousing: How to Plan and Implement.* s.l. : Khan Consulting and Publishing, LLC, 2005. 9780595340798.

Kimball, R. 1996. The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. Chichester : John Wiley & Sons, Inc., 1996. 978-0471153375.

Kimball, Ralph and Ross, Margy. 2002. *The Data Warehouse Toolkit, 2nd ed.* New York : Wiley, 2002.

—. 2011. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling.* s.l. : John Wiley and Sons, 2011. 9781118082140.

KimballL, Ralph, et al. 1998. *The data warehouse lifecycle toolkit expert methods for designing, developing & deploying data warehouse.* s.l. : Wiley, 1998. 978-0471255475 .

Krishna, S. 1992. Introduction to Database and Knowledge-based Systems. s.l. : World Scientific, 1992.

Kroupová, Zdeňka. 2010. Produkční schopnost a technická efektivnost ekologického zemědělství České republiky. *Disertační práce na Provozně ekonomické fakultě České zemědělské univerzity na katedře ekonomiky.* 2010. Vedoucí disertační práce Doc. Ing. Eva Rosochatecká, CSc..

—. 2010. Technická efektivnost ekologického zemědělství České republiky. Ostrava : Ekonomická revue, 2010. Sv. 2, 13, stránky 61 - 73. 1212-3951.

Kučerová, Helena. 2003. Databáze. *KTD: Česká terminologická databáze knihovnictví a informační vědy (TDKIV).* [Online] 2003. [Citace: 8. Srpen 2011.] http://aleph.nkp.cz/F/?func=direct&doc_number=000000089&local_base=KTD.

Kumbhakar, S.C. and Lovell, C.A.K. 2000. *Stochastic Frontier analysis.* Cambridge : Cambridge University Press, 2000.

Lacko, Luboslav. 2006. *Business Intelligence v SQL Serveru 2005 - Reportovací, analytické a další datové služby.* Brno : Computer Press, a.s., 2006. 80-251-1110-5.

Lee, Ki Yong, Chung, Yon Dohn and Kim, Myoung Ho. 2010. An efficient method for maintaining data cubes incrementally. *Information Sciences.* s.l. : Elsevier, 2010. Vol. 180, 6. 0020-0255.

Lehner, W. 2010. Merging OLTP and OLAP – Back to the Future. *Enabling Real-Time Business Intelligence.* s.l. : Springer Berlin Heidelberg, 2010. Vol. 41, pp. 171-173. 978-3-642-14559-9.

Lehner, Wolfgang. 2009. Query Processing in Data Warehouses. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 2297-2301. 978-0-387-39940-9.

Lechtenbörger, Jens and Vossen, Gottfried. 2003. Multidimensional normal forms for data warehouse design. s.l. : Elsevier, 2003. Vol. 28, pp. 415–434. 0306-4379/03.

Lenz, Hans-J. and Shoshani, Arie. 1997. Summarizability in OLAP and Statistical Data Bases. s.l. : IEEE Computer Society, 1997. pp. 132--143.

Levene, Mark and Loizou, George. 2003. Why is the snowflake schema a good data warehouse design? *Information Systems.* 2003, Vol. 28, pp. 225–240.

Malinowski, E. and Zimányi, E. 2006. Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Data & Knowledge Engineering.* 2006, Vol. 59, pp. 348–377.

Markl, Volker. 2009. Query Processing (in Relational Databases). [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 2288-2293. 978-0-387-39940-9.

Matulová, Kateřina. 2011. Přímé platby, technologická změna a technická efektivnost: české zemědělství po vstupu do Evropské unie. *Think Together 2011.* Praha : PEF ČZU v Praze, 2011. 978-80-213-2169-4.

Mazón, J.N., et al. 2011. Designing Data Warehouses: from Business Requirement Analysis to Multidimensional Modeling. *In International Workshop on Requirements Engineering for Business Need and IT Alignment.* 2011. 0733422764.

McGuff, F. 1998. Designing the Perfect Data Warehouse. [Online] 1998. http://members.aol.com/fmcguff/dwmodel/index.htm.

McGuire, Michael, et al. 2008. A user-centered design for a spatial data warehouse for data exploration in environmental research. *Ecological Informatics.* s.l. : Elsevier, 2008. 1574-9541.

Merunka, Vojtěch. 2006. *Datové modelování.* Praha : Alfa Publishing, 2006. 80-86851-54-0.

Microsoft. 2011. Multidimensional Expressions (MDX) Reference. *MSDN Library.* [Online] 2011. [Cited: 25 August 2011.] http://msdn2.microsoft.com/en-us/library/ms145506.aspx.

Mitra, Pitram. 2009. Relational Algebra Learning Tool. 2009. p. 93.

Morfonios, Konstantinos, et al. 2007. ROLAP implementations of the data cube. *ACM Comput. Surv.* New York, NY, USA : ACM, 2007. Vol. 39, 4. 0360-0300.

Murakami, E., et al. 2007. An infrastructure for the development of distributed service-oriented information systems for precision agriculture. *Computers and Electronics in Agriculture.* 2007. Vol. 58, 1, pp. 37–48.

Mylopoulos, John. 2009. Database Design. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 708-710. 978-0-387-35544-3.

Navathe, S. B. 1992. Evolution of data modeling for databases. *Commun ACM.* 1992, Vol. 35, 9, pp. 112-123.

Nedjar, Sébastien, Cicchetti, Rosine and Lakhal, Lotfi. 2011. Extracting semantics in OLAP databases using emerging cubes. *Information Sciences.* s.l. : Elsevier, 2011. Vol. 181, 10, pp. 2036-2059. 0020-0255.

Niemi, Tapio, Nummenmaa, Jyrki and Thanisch, Peter. 2003. Normalising OLAP cubes for controlling sparsity. *Data Knowl. Eng.* 2003, Vol. 46, 3, pp. 317-343.

Nikkila, R., Seilonen, I. and Koskinen, K. 2010. Software architecture for farm management information systems in precision agriculture. *Computers and Electronics in Agriculture.* 2010. Vol. 70, 2, pp. 328–336.

Nilakanta, Sree, Scheibe, Kevin and Rai, Anil. 2008. Dimensional issues in agricultural data warehouse designs. *Comput. Electron. Agric.* s.l. : Elsevier Science Publishers B. V., 2008. Vol. 60, 2. 0168-1699.

Novotný, Ota, Pour, Jan a Slánský, David. 2005. *Business Intelligence.* Praha : Grada Publishing, a.s., 2005. 80-247-1094-3.

Online Etymology Dictionary. 2010. Database. *Online Etymology Dictionary.* [Online] 2010. [Citace: 8. Srpen 2011.] http://www.etymonline.com/index.php?search=database&searchmode=none.

Pardillo, Jesús and Mazón, Jose-Norberto. 2011. Model-driven development of OLAP metadata for relational data warehouses. *Computer Standards & Interfaces.* s.l. : Accepted Manuscript, 2011. 0920-5489.

Pardillo, Jesús, Mazon, Jose-Norberto and Trujillo, Juan. 2010. Extending OCL for OLAP querying on conceptual multidimensional models of data warehouses. *Information Sciences.* 2010. Vol. 180, 5, pp. 584-601. 0020-0255.

Pedersen, T.B. 2009(a). Cube. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(a). pp. 538-539. 978-0-387-39940-9.

—. 2009(b). Dimension. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(b). p. 836. 978-0-387-39940-9.

—. 2009(c). Multidimensional Modeling. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(c). pp. 1777-1784. 978-0-387-39940-9.

—. 2009. Multidimensional Modeling. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 1777-1784. 978-0-387-39940-9.

Pedersen, T.B., Jensen, C.S. and Dyreson, C.E. 2001. A foundation for capturing and querying complex multidimensional data. *Inf. Syst.* 2001. Vol. 26, 5.

Pitoura, Evaggelia. 2009(a). Query Plan. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(a). p. 2282. 978-0-387-39940-9.

—. 2009(c). Query Processing. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(c). 978-0-387-39940-9.

Pokorný, Jaroslav. 2006. Database architectures: Current trends and their relationships to environmental data management. *Environmental Modelling & Software.* s.l. : Elsevier Ltd, 2006. Vol. 21, 11, pp. 1579 - 1586. 1364-8152.

—. 2011. NoSQL Databases: a step to database scalability in Web environment. *Proceeding The 13th International Conference on Information Integration and Web-based Applications & Services (iiWAS2011).* Ho Chi Minh City, Vietnam : ACM, 2011. 978-1-4503-0784-0.

—. 2012. NoSQL databáze - současný stav vývoje. *MODERNÍ DATABÁZE 2012 ARCHITEKTURA INFORMAČNÍCH SYSTÉMŮ.* Praha : KOMIX s.r.o., 2012. stránky 1-10. 978-80-905231-0-4.

Post, Gerald and Kagan, Albert. 2001. Database management systems: design considerations and attribute facilities. *The Journal of Systems and Software.* 2001, Vol. 56, 2, pp. 183-193.

Pour, Jan a Slánský, David. 2004. Efekty a rizika Business Intelligence. *Systémová integrace.* Praha : ČSSI, 2004. 1210-9479.

Pour, Jan. 2010. Business intelligence – prostor trvalého rozvoje. *Systémová integrace: Role ICT při zvyšování konkurenceschopnosti ČR.* Praha : VŠE, 2010. stránky 114-122. 978-80-245-1660-8.

Prat, Nicolas, Akoka, Jacky and Comyn-Wattiau, Isabelle. 2006. A UML-based data warehouse design method. *Decis. Support Syst.* s.l. : Elsevier Science Publishers B. V., 2006. Vol. 42, 3. 0167-9236.

Rai, A., et al. 2008. Design and development of data mart for animal resources. *Computers and electronics in agriculture.* s.l. : Elsevier, 2008. 64, pp. 111-119.

Rain, T. and Švarcová, I. 2010. WCA Framework. *AGRIS on-line Papers in Economics and Informatics.* Prague : FEM CULS Prague, 2010. 3, pp. 63 - 67. 1804-1930.

Ramesh, Jain. 2009. Multimedia Databases. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 1817-1820. 978-0-387-39940-9.

Ramsak, Frank, et al. 2001. Interactive ROLAP on Large Datasets: A Case Study with UB-Trees. *Proceedings of IDEAS 2001 in Grenoble, France.* Piscataway, NJ 08855-1331, USA. : IEEE, 2001.

Risch, Tore. 2009(a). Distributed Architecture. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(a). pp. 875-879. 978-0-387-39940-9.

—. 2009(b). Query Language. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(b). pp. 2260-2261. 978-0-387-39940-9.

Rivest, S., et al. 2005. SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. *ISPRS Journal of Photogrammetry and Remote Sensing.* s.l. : Elsevier, 2005. Vol. 60, 1. 0924-2716.

Rizzi, S. 2009. Business Intelligence. *Encyclopedia of Database Systems.* USA : Springer Science+Business Media, LLC, 2009. p. 288. 978-0-387-35544-3.

Rizzi, Stefano, et al. 2006. Research in data warehouse modeling and design: dead or alive? *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP.* Arlington, Virginia, USA : ACM, 2006. 1-59593-530-4.

Robertson, Lesley Anne. 2003. Pseudocode. *Encyclopedia of Information Systems.* New York : Elsevier, 2003. 9780122272400.

Romero, Oscar and Abelló, Alberto. 2011. A comprehensive framework on multidimensional modeling. *Proceedings of the 30th international conference on Advances in conceptual modeling: recent developments and new directions.* Brussels, Belgium : Springer-Verlag, 2011. 978-3-642-24573-2.

—. 2010. Automatic validation of requirements to support multidimensional design. *Data & Knowledge Engineering.* s.l. : Elsevier, 2010. Vol. 69, 9, pp. 917-942. 0169-023X.

—. 2007b. Automating multidimensional design from ontologies. *Proceedings of the ACM tenth international workshop on Data warehousing and OLAP.* Lisbon, Portugal : ACM, 2007b. 978-1-59593-827-5.

—. 2007. On the need of a reference algebra for OLAP. *Data Warehousing and Knowledge Discovery.* s.l. : Springer Berlin / Heidelberg, 2007. Vol. 4654. 978-3-540-74552-5.

Rouhani, S, Ghazanfari, M and Jafari, M. 2012. Evaluation model of business intelligence for enterprise systems using fuzzy TOPSIS. *Expert Systems with Applications: An International Journal.* Tarrytown, NY, USA : Elsevier Ltd., 2012. Vol. 39, 3, pp. 3764-3771 . 0957-4174.

Sapia, C., et al. 1998. Extending the E/R model for the multidimensional paradigm. *Lecture Notes in Computer Science.* 1998, Vol. 1552.

Seják, J. and Zavíral, J. 2007. Growing inequalities in added-value distribution in the Czech agri-food chains. 53 *AGRIC. ECON. – CZECH.* 2007. 5, pp. 235–245. 0139-570X.

Serrano, Manuel Angel, et al. 2008. Empirical studies to assess the understandability of data warehouse schemas using structural metrics. *Software Quality Control.* Hingham, MA, USA : Kluwer Academic Publishers, 2008. Vol. 16, 1. 0963-9314.

Sheth, A. and Larson, J. 1990. Federated database systems. *ACM Computing Surveys.* 1990. Vol. 23, 3, pp. 183–236.

Shoshani, A. 1997. OLAP and statistical databases: similarities and differences. 1997. pp. 185–196.

Schneider, Markus. 2009(a). Spatial and Spatio-Temporal Data Models and Languages. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(a). pp. 2681-2685. 978-0-387-39940-9.

—. 2009(b). Spatial Data Types. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009(b). pp. 2698-2702. 978-0-387-39940-9.

Schulze, Ch., Spilke, J. and Lehner, W. 2007. Data modeling for Precision Dairy Farming within the competitive field of operational and analytical tasks. *Computers and Electronics in Agriculture.* s.l. : Elsevier, 2007. 59, pp. 39–55.

Silberschatz, A., Korth, F. H. and Sudarshan, S. 1997. Database System Concepts. s.l. : McGraw-Hill, 1997.

—. 2001. *Database System Concepts Fourth Edition.* s.l. : Bell Laboratories, 2001. Instructor Manual Version 4.0.0.

Singh, S. K. 2009. *Database Systems: Concepts, Design and Applications.* s.l. : Pearson Education India, 2009. p. 896. 9788177585674.

Singhal, Anoop. 2007. An Overview of Data Warehouse, OLAP and Data Mining Technology. *Data Warehousing and Data Mining Techniques for Cyber Security.* s.l. : Springer US, 2007. Vol. 31. 978-0-387-47653-7.

Sirangelo, Cristina. 2009. Join. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 1580-1581. 978-0-387-39940-9.

Song, I.-Y. 2009(b). Data Mart. *Encyclopedia of Database Systems.* USA : Springer Science+Business Media, LLC, 2009(b). p. 594. 978-0-387-35544-3.

—. 2009(a). Data Warehouse. *Encyclopedia of Database Systems.* USA : Springer Science+Business Media, LLC, 2009(a). p. 657. 978-0-387-35544-3.

Sörensen, C.G., et al. 2010. Conceptual model of a future farm management information system. *Computers and Electronics in Agriculture.* s.l. : Elsevier, 2010. Vol. 72, pp. 37–47.

Sörensen, C.G., et al. 2011. Functional requirements for a future farm management information system. *Computers and Electronics in Agriculture.* s.l. : Elsevier B.V., 2011. Vol. 76.

Stonebraker, M. and Moore, D. 1996. Object-Relational DBMSs The Next Great Wave. Los Altos, CA : Morgan Kaufmann, 1996.

Stonebraker, M., et al. 1976. The design and implementation of INGRES. *ACM Transactions on Database Systems.* 1976. Vol. 1, 3. 189–222.

Stonebraker, M., et al. 1990. Third generation database systems manifesto. *ACM SIGMOD Record.* 1990, Vol. 19, 3, pp. 31-44.

Storey, Veda C., Thompson, Cheryl Bagley and Ram, Sudha. 1995. Understanding database design expertise. *Data & Knowledge Engineering 16.* s.l. : Elsevier Science B.V., 1995. pp. 97-124.

Střeleček, F., Lososová, J. and Zdeněk, R. 2011. Economic results of agricultural enterprises in 2009. 57 *Agric. Econ. – Czech.* 2011. 3, pp. 103–117. 0139-570X.

Šarmanová, Jana. 2007. *TEORIE ZPRACOVÁNÍ DAT.* Ostrava : VŠB – Technická univerzita Ostrava, 2007. 2, str. 169. 978-80-248-1498-8.

Šilerová, E. and Kučírková, L. 2010. Information and Comunication Technologies and their Influence and the Management. *Economy&Business.* Bulgaria : s.n., 2010. Vol. 4, pp. 224-229. ISSN 1313-2555.

Tannen, Val. 2009. Relational Calculus. [ed.] Ling Liu and M. Tamer Özsu. *Encyclopedia of Database Systems.* s.l. : Springer US, 2009. pp. 2370-2371. 978-0-387-39940-9.

Teorey, T.J., Yang, D. and Fry, J.P. 1986. A logical design methodology for relational databases using the extended entity-relationship model. s.l. : ACM Computer Surveys, 1986. Vol. 18, 2. 197-222.

Thanisch, Peter, a další. 2012. Mapping a Resource Description Framework OLAP Ontology to the Business Intelligence Semantic Model. *Practical Applications of Intelligent Systems.* Berlin / Heidelberg : Springer, 2012. Sv. 124, stránky 419-428. 978-3-642-25657-8.

Thomsen, E. 2002. OLAP Solutions. *Building Multidimensional Information Systems.* s.l. : John Wiley, 2002. 0-471-40030-0.

—. 1997. OLAP Solutions: Building Multidimensional Information Systems. New York : Wiley, 1997.

Tria, Francesco, Lefons, Ezio and Tangorra, Filippo. 2011. GrHyMM: A Graph-Oriented Hybrid Multidimensional Model. *Advances in Conceptual Modeling. Recent Developments and New Directions.* s.l. : Springer Berlin Heidelberg, 2011. 978-3-642-24573-2.

—. 2012. Hybrid methodology for data warehouse conceptual design by UML schemas. *Inf. Softw. Technol.* s.l. : Butterworth-Heinemann, 2012. Vol. 54, 4. 0950-5849.

Tryfona, N., Busborg, F. and Christiansen, J. G. B. 1999. StarER: A Conceptual Model for Data Warehouse Design. *Proceedings of the ACM DOLAP99 Workshop.* Missouri : s.n., 1999.

Tsichritizis, D.C. and Klug, A. (Eds.). 1978. The ANSI/X3/SPARC DBMS framework: report of the study group on database management systems. *Information Systems.* 1978. 3.

Tsois, A., Karayannidis, N. and Sellis, T. 2001. MAC: Conceptual Data Modeling for OLAP . *3rd International Workshop on Design and Management of Data Warehouses (DMDW 2001).* 2001.

Tvrdoň, Jiří. 2006. *Ekonometrie.* Praha : Česká zemědělská univerzita v Praze, 2006. 80-213-0819-2.

Tyrychtr, J., Buchtela, D. a Havlíček, Z. 2012. Návrh ROLAP databáze v zemědělském podniku: Transformace ekonometrického modelu do konceptuálního modelu dat. *Systémová integrace.* Praha : VŠE ČSSI, 2012. Sv. 12, 2. 1210-9479.

Tyrychtr, Jan, Poláčková, Julie a Švarcová, Ivana. 2010. Akcentace a inovace konceptuálního modelu při vývoji databázových aplikací. *Systémová integrace.* 2010, Sv. 17, 4, stránky 51-62.

Ulman, M. 2009. Informační toky farmy. Praha : IPC, ČZU v Praze, 2009. 978-80-213-1932-5.

Urban, Susan D. and Dietrich, Suzanne W. 2009. Object Data Models. [ed.] Ling Liu and M. Tamer Özsu. s.l. : Springer US, 2009. pp. 1929-1935. 978-0-387-39940-9.

Vaníček, Jiří. 2004. *Měření a hodnocení jakosti informačních systémů.* Praha : PEF, ČZU v Praze, 2004. 80-213-1206-8.

Vassiliadis, P. and Sellis, T.K. 1999. A survey of logical models for OLAP databases. s.l. : ACM SIGMOD Rec., 1999. Vol. 28, 4.

Vincent, M. 1997. A corrected 5NF definition for relational database design. *Theor. Comput. Sci.* 1997. Vol. 185, 2, pp. 379–391.

Vossen, Gottfried. 2009. Transaction Management. [ed.] Ling Liu and M. Tamer Özsu. s.l. : Springer US, 2009. pp. 3153-3157. 978-0-387-39940-9.

Vostrovský, Václav. 2004. *Vytváření databází v ORACLE.* Praha : PEF, Česká zemědělská univerzita v Praze, 2004. str. 134. 80-213-1191-6.

Whitehorn, Mark and Marklyn, Bill. 2007. Inside Relational Databases with Examples in Access. s.l. : Springer, 2007. 1-84628-394-9.

Winter, R. and Strauch, B. 2003. A Method for Demand-Driven Information Requirements Analysis in DW Projects. *Proc. of 36th Annual Hawaii Int. Conf. on System Sciences.* Los Alamitos : IEEE, 2003. 0-7695-1874-5.

Wolfgang, Lehner. 2009. Query Processing in Data Warehouses. [ed.] Ling Liu and M. Tamer Özsu. s.l. : Springer US, 2009. pp. 2297-2301. 978-0-387-39940-9.

Wu, M.C. and Buchmann, A.P. 1997. Research Issues in Data Warehousing. *Intl. Conference on Databases in Office, Engineering and Science (BTW'97).* Ulm, Germany : s.n., 1997.

Zádová, Vladimíra. 2009. Multidimenzionální modelování v rámci analýzy a návrhu IS/ICT. *Systémová Integrace.* 2009, 4, stránky 66-76.

Závodný, Martin a Pergl, Robert. 2011. Business Intelligence pro univerzitní prostředí. *Systémová integrace, roč. 12, č. 2.* 2011, stránky 288-296.

Přílohy

Seznam příloh

Příloha č. 1 – 12 pravidel E. F. Codda pro OLAP

OLAP je vymezen pomocí 12 pravidel stanovených Coddem (1993):

Pravidlo 1: Multidimenzionalita. Uživatelsko-analytický pohled v organizaci musí mít multidimenzionální povahu. Multidimenzionální datové modely umožňují jednodušší a intuitivní manipulaci s daty uživatelů, včetně „slicing“ a „dicing“.

Pravidlo 2: Transparentnost. Pokud OLAP nástroje jsou součástí uživatelova běžného prostředí (tabulkového kalkulátoru, grafického balíčku), pak musí být pro uživatele transparentní. OLAP by měl být součástí otevřené architektury systémů, který může být vložen do libovolného místa podle potřeb uživatele, aniž by měl vliv na funkčnost hostitelského nástroje. Uživateli by neměli být odkryty zdroje dat poskytnutých OLAP nástrojem.

Pravidlo 3: Dostupnost. Nástroj OLAP by měl být schopen uplatnit vlastní logickou strukturu pro přístup k heterogenním zdrojům dat a provádět jakékoliv převody nutné k předložení uceleného pohledu uživateli. Nástroj (a nikoli uživatel) by měl být zainteresován v tom, odkud fyzická data pochází.

Pravidlo 4: Konzistentní výkon reportování. Výkon OLAP nástrojů nesmí záviset na počtu dimenzí.

Pravidlo 5: Architektura klient-server. Serverová část OLAP nástrojů by měla být dostatečně inteligentní, aby různí klienti mohli být připojeni s minimálním úsilím. Tento server by měl být schopen zobrazení a konsolidace dat mezi jednotlivými databázemi.

Pravidlo 6: Rovnoprávnost dimenzí. Všechna dimenzionální data by měla být ekvivalentní strukturou a operační schopností.

Pravidlo 7: Dynamické řízení řídkých matic. OLAP server by měl ve fyzické struktuře optimálně zpracovávat řídké matice.

Pravidlo 8: Víceuživatelská podpora. OLAP nástroje musí zajistit souběžný přístup k vyhledávání, aktualizaci a jejich zabezpečení.

Pravidlo 9: Neomezené operace napříč dimenzemi. Výpočetní prostředky musí umožňovat výpočty a manipulaci s daty v libovolném počtu dimenzí dat, a nesmí omezovat žádný vztah mezi datovými buňkami.

Pravidlo 10: Intuitivní ovládání. Manipulace s daty, jako „drilling down“ nebo „zooming“, by měla být provedeno prostřednictvím přímého působení na analytický model buněk, a nevyžadovat použití menu nebo více cest v uživatelském rozhraní.

Pravidlo 11: Flexibilní reporting. Reporting by měl prezentovat informace na základě požadavků uživatelů.

Pravidlo 12: Neomezený počet dimenzí a úrovní v hierarchiích. Počet dimenzí dat by měl být neomezený. Každý obecný rozměr by měl umožnit v podstatě neomezený počet uživatelem definovaných úrovní agregace v dané pozici.

Příloha č. 2 – Data v tabulkách při návrhu prototypu

Dimenze Čas:

ID_Čas	Rok	Kvartál
1	2010	I. kvartál
2	2010	II. kvartál
3	2010	III. kvartál
4	2010	IV. kvartál
5	2011	I. kvartál
6	2011	II. kvartál
7	2011	III. kvartál
8	2011	IV. kvartál
9	2012	I. kvartál
10	2012	II. kvartál

Integrovaná data - A

Tabulka faktů:

ID_Půda	ID_Práce	ID_Kapitál	ID_Čas	Produkce
1	1	1	1	6 029
1	1	2	2	6 140
2	1	2	3	6 299
3	1	3	4	6 548
3	2	3	5	7 817
3	2	3	6	7 817
4	2	3	7	7 985
4	3	2	8	9 021
4	3	3	9	9 162
4	4	3	10	10 251

Dimenze Půda:

ID_Půda	Výměra půdy
1	90
2	100
3	110
4	120

Dimenze Práce:

ID_Práce	Počet zaměstnanců
1	3
2	4
3	5
4	6

Dimenze Kapitál:

ID_Kapitál	Velikost kapitálu
1	2 500 000
2	3 000 000
3	3 500 000
4	4 000 000

Integrovaná data - B

Tabulka faktů:

ID_Půda	ID_Práce	ID_Kapitál	ID_Čas	Produkce
1	1	1	1	30
2	2	3	2	6
3	2	2	3	8
2	2	3	4	6
4	1	4	5	32

Dimenze Půda:

ID_Půda	Výměra půdy
1	90
2	0
3	10
4	100

Dimenze Práce:

ID_Práce	Počet zaměstnanců
1	3
2	0
3	1

Dimenze Kapitál:

ID_Kapitál	Velikost kapitálu
1	3,5
2	0,0
3	0,5
4	4,5

Integrovaná data - C

Tabulka faktů:

ID_Půda	ID_Práce	ID_Kapitál	ID_Čas	Produkce
1	1	1	1	6 235
1	1	2	2	6 319
1	1	3		6 394
1	1	4		6 461
1	2	1		7 444
1	2	2		7 544
1	2	3		7 633
1	2	4		7 714
1	3	1		8 541
1	3	2		8 655
1	3	3		8 758
1	3	4		8 851
1	4	1		9 556
1	4	2		9 684
1	4	3		9 799
1	4	4		9 903
2	1	1		6 397
2	1	2	3	6 483
2	1	3		6 560
2	1	4		6 630
2	2	1		7 638
2	2	2		7 740
2	2	3		7 832
2	2	4		7 915

2	3	1		8 763
2	3	2		8 881
2	3	3		8 986
2	3	4		9 081
2	4	1		9 805
2	4	2		9 936
2	4	3		10 054
2	4	4		10 161
3	1	1		6 548
3	1	2		6 636
3	1	3	4	6 714
3	1	4		6 786
3	2	1		7 817
3	2	2		7 922
3	2	3	5	8 016
3	2	4		8 101
3	3	1		8 969
3	3	2		9 090
3	3	3		9 197
3	3	4		9 295
3	4	1		10 035
3	4	2		10 170
3	4	3		10 291
3	4	4		10 400
4	1	1		6 688
4	1	2		6 778
4	1	3		6 858

4	1	4		6 931
4	2	1		7 985
4	2	2		8 092
4	2	3	7	8 188
4	2	4		8 275
4	3	1		9 162
4	3	2	8	9 285
4	3	3	9	9 395
4	3	4		9 494
4	4	1		10 251
4	4	2		10 388
4	4	3	10	10 511
4	4	4		10 623

Dimenze Půda:

ID_Půda	Výměra půdy
1	90
2	100
3	110
4	120

Dimenze Práce:

ID_Práce	Počet zaměstnanců
1	3
2	4
3	5
4	6

Dimenze Kapitál:

ID_Kapitál	Velikost kapitálu
1	3 500 000
2	4 000 000
3	4 500 000
4	5 000 000

Příloha č. 3 – Generované ekonometrické modely a hodnoty měr měření

	Identifikace		Schéma konstelace									Schéma hvězda						
Modely	**k****	**g*-1**	**NFT**	**NDT**	**NFK(FT1)**	**NFK(FT2)**	**NFK(FT3)**	**NFK**	**NMFT**	**NSDT**	**CELKEM 1**	**CELKEM 2**	**NFT**	**NDT**	**NFK**	**NMFT**	**NSDT**	**CELKEM 1**	**CELKEM 2**
1. y1 = x5 x1 x2 x4	0	0	1	5	5	0	0	5	1	0	12	1	1	5	5	1	0	12	1
2. y1 = y2 x3 x1	3	1	2	6	4	5		9	2	3	19	5	1	6	6	2	0	15	1
y2 = x1 x3 x4 x2 x6	0	0																	
3. y1 = x2	0	0	1	2	2	0	0	2	1	0	6	1	1	2	2	1	0	6	1
4. y1 = y2 x1	4	1	2	6	2	5		7	2	2	17	4	1	6	6	2	0	15	1
y2 = x5 x6 x3 x4 x1	0	0																	
5. y1 = y2 x1 x4 x5 x2	2	1	3	7	6	7	7	20	3	7	33	10	1	7	7	3	0	18	1
y2 = y1 x5 x6 x3 x2 x4	1	1																	
y3 = y2 x3 x2 x1 x5 x6	1	1																	
6. y1 = y3 y2 x2 x1	3	2	3	6	5	6	4	15	3	3	27	6	1	6	6	3	0	16	1
y2 = y3 x6 x2 x3 x1	1	1																	
y3 = y2 x1 x2 x4	2	1																	
7. y1 = x3 x1 x4	2	0	3	5	3	2	5	10	3	4	21	7	1	5	5	3	0	14	1
y2 = x2 x1	3	0																	
y3 = y1 x1 x4 x2	2	1																	
8. y1 = x1 x2	0	0	1	3	3	0	0	3	1	0	8	1	1	3	3	1	0	8	1
9. y1 = x2 x3 x4	2	0	3	5	4	5	5	14	3	4	25	7	1	5	5	3	0	14	1
y2 = y3 x4 x2 x1	2	1																	
y3 = y2 x4 x2 x3	2	1																	
10. y1 = y3 x2 x3	1	1	3	4	4	4	4	12	3	3	22	6	1	4	4	3	0	12	1
y2 = y1 x3 x2	1	1																	
y3 = x3 x2 x4	0	0																	
11. y1 = x1	4	0	3	6	2	6	3	11	3	3	23	6	1	6	6	3	0	16	1
y2 = x3 x2 x5 x1 x4	0	0																	
y3 = x1 x2	3	0																	
12. y1 = x1	1	0	2	3	2	3	0	5	2	2	12	4	1	3	3	2	0	9	1
y2 = x2 x1	0	0																	
13. y1 = y2 y3 x4 x5 x6 x1	2	2	3	7	7	3	3	13	3	2	26	5	1	7	7	3	0	18	1
y2 = x2 x3	4	0																	

	y3 = y1 x2	5	1																
14.	y1 = x1 x2	0	0	1	3	3	0	0	3	1	0	8	1	1	3	3	1	0	8	1
15.	y1 = y2 y3 x3 x1	1	1	3	4	5	4		9		3	16	6	1	4	4	3	0	12	1
	y2 = y1 x3 x1	1	1																	
	y3 = x2	2	0																	
16.	y1 = x2 x1 x3	0	0	1	4	4	0	0	4	1	0	10	1	1	4	4	1	0	10	1
17.	y1 = x2 x4 x5 x3	1	0	3	6	5	3	6	14	3		26	3	1	6	6	3	0	16	1
	y2 = x2 x1	3	0																	
	y3 = x2 x1 x5 x4 x3	0	0																	
18.	y1 = x5 x4 x3 x1 x2	0	0	2	6	6	3	0	9	2	3	19	5	1	6	6	2	0	15	1
	y2 = x1 x3	3	0																	
19.	y1 = y2 x4 x3 x1	1	1	3	5	5	3	5	13	3	3	24	6	1	5	5	3	0	14	1
	y2 = y3 x2	3	1																	
	y3 = y1 x2 x3 x4	1	1																	
20.	y1 = x2 x4 x1 x5	0	0	1	5	5	0	0	5	1	0	12	1	1	5	5	1	0	12	1
21.	y1 = x4 x3 x1	0	0	1	4	4	0	0	4	1	0	10	1	1	4	4	1	0	10	1
22.	y1 = x2 x3 x1 x5 x4	0	0	1	6	6	0	0	6	1	0	14	1	1	6	6	1	0	14	1
23.	y1 = x2 x1	0	0	1	3	3	0	0	3	1	0	8	1	1	3	3	1	0	8	1
24.	y1 = x1	3	0	3	5	2	4	5	11	3	3	22	6	1	5	5	3	0	14	1
	y2 = y1 x2 x1	2	1																	
	y3 = y2 x3 x2 x4	1	1																	
25.	y1 = x3 x4 x5 x1	1	0	3	6	5	5	6	16	3	5	28	8	1	6	6	3	0	16	1
	y2 = y3 x4 x3 x1	2	1																	
	y3 = y2 x2 x5 x4 x3	1	1																	
26.	y1 = x1 x2 x3	0	0	3	4	4	3	3	10	3	2	20	5	1	4	4	3	0	12	1
	y2 = y1 x2	2	1																	
	y3 = y1 y2	3	2																	
27.	y1 = y2 x1 x3 x4	1	1	2	5	5	3	0	8	2	2	17	4	1	5	5	2	0	13	1
	y2 = x2 x1	2	0																	
28.	y1 = y2 x2	2	1	3	4	3	4	4	11	3	2	21	5	1	4	4	3	0	12	1
	y2 = x2 x3 x1	0	0																	
	y3 = y2 x1 x2	1	1																	
29.	y1 = x2 x5 x3 x4	0	0	1	5	5	0	0	5	1	0	12	1	1	5	5	1	0	12	1
30.	y1 = y3 x2 x3 x4	2	1	3	6	5	3	6	14	3	4	26	7	1	6	6	3	0	16	1
	y2 = y1 x1	4	1																	
	y3 = y1 x4 x5 x3 x1	1	1																	