Systém analýzy dat

Systém analysis of data

Bakalářská práce

Zdenka Roudenská

Vedoucí práce: Ing Václav Novák, CSc.

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

2008

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracovala samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne

Anotace

Tato bakalářská práce se věnuje systému analýzy dat. Konkrétněji se ve své práci zabývám analýzou dat nad informačním systémem HELIOS firmy LCS. Tento informační systém je provozován na Microsoft SQL serveru. Proto se zde jako nejúčelnější pro analýzu jeví použití produktu Microsoft Analysis Services SQL 2005 a jeho metody datových kostek. Z hlediska dalšího šíření výstupů je vhodné využít webového rozhraní z důvodu obecnosti nebo Microsoft SharePoint Server pro jeho interaktivnost. Obě varianty mají své výhody i nevýhody.

Abstract

The thesis deals with systém analysis of data. Particular attention is paid to data analysis of information system Helios, firm LCS. This information systems is provided on Microsoft SQL server. Therefore seems the most effective to provide the analysis the usage of product called Microsoft Analysis Services SQL 2005 and methods of data cubes. To spread further outcomes it is most convenient to use web borderline because of its generality or Microsoft Sharepoint Server because of its interactivity. Both versions have their advantages and disadvantages which are compared and evaluated.

Poděkování

Ráda bych poděkovala panu Ing. Václavu Novákovi, CSc. za to, že se ochotně ujal mé práce. Dále bych chtěla poděkovat panu Ing. Dominovi za předané zkušenosti a rady při zpracování bakalářské práce.

Obsah

1 Úvod a cíl práce

1.1 Úvod

V dnešní době, která je v celém světě charakterizována jako „věk informatiky", je zpracovávání informací a jejich analýza důležitým předpokladem vysoké prosperity firmy.

Potřeba analyzovat data existuje již velmi dlouho a každý z nás se s ní již v životě setkal. Mohlo jít třeba o jednoduchou analýzu rodinných výdajů. Pro firmy je taková analýza daleko potřebnější. Ať už pro interní potřeby firmy a nebo analýzy dané zákonem.

V době kdy ještě firmy neměly počítače a tedy ani možnost zpracovávat data elektronicky se vše muselo dělat ručně a na papír. Toto bylo vždy velmi zdlouhavé a mohlo se stát, že výsledky už pro firmu nebyly aktuální.

Dnešní doba nám poskytuje daleko více prostředků a možností jak data analyzovat. Dat je také čím dál větší množství. Jednoduchý a rychlý pohled na globální výsledky je mnohdy důležitější než detail. Jednou z podmínek dobrého fungování se tak stává kvalitní informační systém. Zavedení informačního systému ve firmě sice na jedné straně přináší konkurenční výhodu, ale na druhé straně se nejedná o malou investici, která obzvláště pro malé firmy a podnikatele je, bohužel, mnohdy z finančních důvodů nerealizovatelná.

Informační systémy slouží především ke shromažďování dat jako jsou evidence zákazníků a organizací, expediční příkazy, výdejky, faktury, účetnictví, evidence zakázek, atd. Lze v něm realizovat jednoduché sestavy a analýzy. Avšak sestavy složitější jako jsou například hospodářské výsledky v jednotlivých obdobích, grafické výstupy, jsou v informačních systémech velmi složité ne-li nemožné. Proto je třeba využít takzvaných nadstaveb.

Ve své práci se budu konkrétně věnovat informačnímu systému Helios IQ. Jedná se o český systém běžící na SQL serveru, který je cenově dostupný i menším firmám. Každý uživatel v tomto systému může vytvořit pouze tabulkové (textové) sestavy. Ovšem jakékoliv složitější sestavy, jakou je například sestava Hospodářských výsledků, bude neúměrně zatěžovat a zpomalovat celý informační systém. Systém, bohužel, nedisponuje takovými prostředky, které by tuto nevýhodu odstranily. Proto je dobré využít dalších prostředků a možností.

1.2 Cíl práce

Mým cílem je navrhnout jiný způsob analýz bez zbytečného zatěžování databázového serveru. Důležitým parametrem je také kompatibilnost s SQL serverem na kterém systém Helios IQ běží. Nesmí omezovat ostatní uživatele systému. Důležitým požadavkem je také jednoduchost práce s analýzami.

Rozhodla jsem se využít MS OLAP kostek, které při správném návrhu vrátí velice rychle požadované výsledky. A také z důvodu, že tyto analytické prostředky jsou součástí MS SQL serveru, který je pro běh informačního systému důležitý. Pro grafické a textové zobrazení následně využiji webového rozhraní nebo Microsoft Sharepoint.

2 Přehled literatury

Při tvorbě datových kostek jsem nejvíce využila knihy SQL Server 2005 Analysis services, která je sice v angličtině, ale jsou v ní velice podrobně popsány postupy tvorby kostky pro různé případy. Krok za krokem jsem s touto knihou vytvořila datovou kostku, jak potřebuji. Tuto knihu bych tedy doporučila každému, kdo se chce tvorbě datových kostek věnovat. Další knihu, kterou jsem také využila se jmenuje Datové sklady, analýza OLAP a dolování dat. Tato kniha je vhodná pro seznámení se samotným systémem OLAP a možnostmi zobrazení hotové kostky. Ovšem postupy tvorby kostky v ní popsané jsem v mém případě již nemohla použít. Postupy v této knize jsou určené pro SQL 2000 a nižší. S příchodem SQL serveru 2005 došlo v tvorbě datových kostek k velké změně.

Při zpracování své práce jsem téměř nevyužila žádných internetových stránek. Alespoň malým pomocníkem mi byly stránky http://datamining.xf.cz, kde jsem nalezla přehlednou teorii, co jsou datové kostky a k čemu se využívají.

Více jsem využila znalostí a rad mých kolegů.

3 Metodika

Kritéria pro výběr vhodného způsobu tvorby analýz nad informačním systémem Helios IQ jsou:

- Musí pracovat s SQL serverem.
- Nesmí nadměrně zatěžovat server při generování analýzy.
- Nesmí omezovat ostatní uživatele informačního systému.
- Jednoduchá práce s analýzami.

Po zvážení všech těchto kritérií se jako nejlepší způsob jeví použití MS OLAP a jeho tvorby analýz formou datových kostek.

Způsob vytvoření datové kostky se odvíjí od potřeb konkrétní firmy. Pro jednoduché a standardní analýzy jsem zvolila metodu předdefinovaných formulářů. Při složitějších a unikátních analýzách je nutno zvolit metodu programování.

Pro zobrazení výsledků analýz jsem zvolila způsob za použití hotových komponent pro jejich jednoduchost a snadnou práci. U složitých analýz jsem musela zvolit opět formu programování.

4 Informační systémy

Co jsou to informační systémy? V podstatě lze říci, že jde o soubor prostředků zajišťující sběr, přenos a uchovávání dat pro účely jejich pozdější prezentace, zpracování a poskytování potřebných informací.

V informačních systémech můžou organizace zpracovávat veškeré pohyby ve skladech, fakturace, evidence zaměstnanců i zákazníků, účetnictví, mzdy a mnoho dalšího. Jedná se tedy i o systémy podporující systém řízení ve firmách, podnicích a v dalších.

Na českém trhu se objevuje mnoho takových systémů. Od jednoduchých systémů určených malým firmám a podnikatelům až po robustní a rozsáhlé systémy pro velké firmy. Převážně jde o systémy z různých států. Příkladem takových systémů může být informační systém SAP nebo iScala. Tyto systémy, podle mého názoru, za sebou táhnou velký kus historie jejich vývoje. To se však nyní stává jejich nevýhodou, jelikož přichází o potřebnou dynamiku. A často nastávají i další problémy při jejich dalším vývoji.

Dnes již existují i české varianty informačních systémů. Většinou jde o nové systémy postavené již na novějších technologiích a tedy i rychleji a snadněji se rozvíjející. Příkladem českého informačního systému je Helios IQ od firmy LCS International a. s., kterému se budu nyní více věnovat. Jelikož naše firma je obchodním zprostředkovatelem tohoto systému a měla jsem tudíž možnost tento systém více prozkoumat.

4.1 Informační systém Helios IQ

Jak jsem již zmínila výše, jedná se o český informační systém od firmy LCS International a.s. Tento systém je distribuován v různých verzích. Nabízí verzi pro malé, střední, ale i velké firmy.

- Helios GREEN – je určen velkým a středním firmám
- Helios ORANGE – pro střední a menší firmy
- Helios RED – pro podnikatele a malé firmy

Tyto verze se pochopitelně liší složitostí ukládání dat a celkovou robustností. Každé verze nabízejí také rozdílné zpracování dat co do formulářů a modulů.

Naše firma se zabývá distribucí verze Helios Orange, která je určená středním a menším firmám.

Obrázek 1 - úvodní obrazovka informačního systému Helios Orange

4.1.1 Více o informačním (ERP) systému Helios Orange

Tento systém poskytuje přehled o situaci uvnitř podniku, automatizaci rutinních operací a zefektivňuje provoz ve firmě. Pomáhá snižovat náklady a

zlepšovat celkovou komunikaci. Pomáhá managementu v řízení v potřebných oblastech, včetně například CRM.

Systém je zcela modulární. Firma, pořizující si tento systém, si sama vybere jaké moduly a oblasti bude pro svou činnost potřebovat. Toto je pro takové firmy výhodné jelikož neinvestují do celkového robustního systému, ale platí pouze za moduly a oblasti, které budou pro jejich práci aktuální.

I nainstalovaný systém lze v budoucnosti rozšiřovat podle potřeb dané firmy. Helios Orange lze rozšiřovat o celé specializované oblasti, zvyšovat počet uživatelů pracujících v systému, ale lze do něj implementovat i samostatně programované moduly na míru každé společnosti. Takové samostatné programované moduly se přidávají pomocí definovaných přehledů nebo externích akcí.

4.1.2 Technologie informačního systému

Aplikace informačního a ekonomického systému Helios Orange je určena pro provoz v systémovém prostředí firmy Microsoft. Technologie klient-server, na které je tento informační systém vystavěn, zajišťuje potřebnou stabilitu a bezpečnost dat.

Systém je provozován na Microsoft SQL serveru. Ten zajišťuje maximální rychlost a neomezenou práci s uloženými daty. Lze využívat celou škálu nástrojů, které databázový server MS SQL poskytuje. Orientace na jeden SQL server umožňuje využít i všech specifických vlastností a veškerých možností serveru včetně přímého přístupu k datům.

Avšak MS SQL server je pro malé firmy cenově příliš nákladný. Proto je možno provozovat systém Helios Orange na databázovém stroji MSDE do verze MS SQL 2000 nebo v současné době Microsoft SQL Server Express edition pro verze MS SQL 2005 a 2008, které jsou distribuovány zdarma. Jde o výrazně zmenšenou verzi MS SQL se základními funkcemi. Tato verze nepos-

kytuje takový komfort. Je omezena i na počet zároveň přistupujících uživatelů, současně zpracovávaných požadavků a objemem dat v databázi. V případě potřeby rozšíření, například umožnění přístupu více uživatelům, je nutno zakoupit plnou verzi MS SQL. S tím je spojená reinstalace databázového serveru a přenos databáze informačního systému.

4.1.3 Výhody systému HELIOS Orange

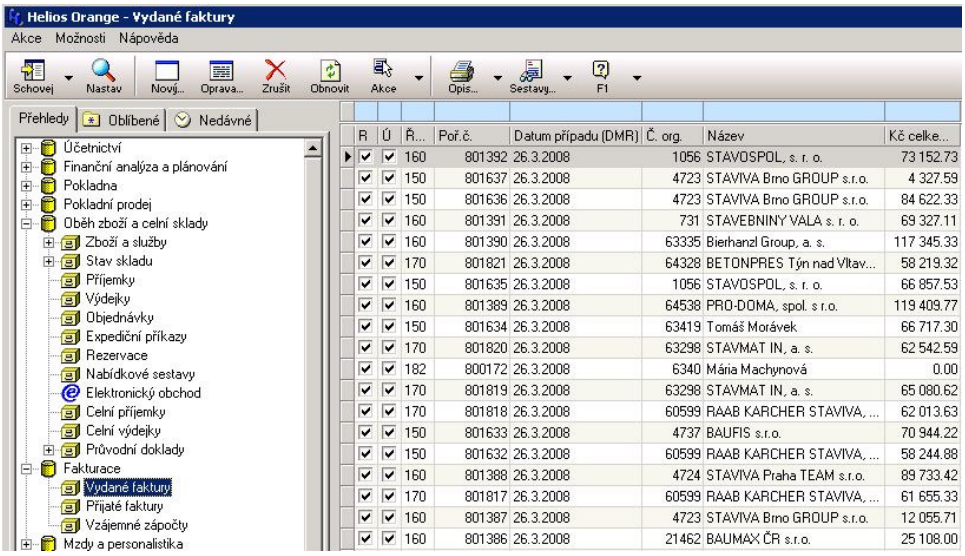
- Díky tomu, že je firma LCS je ryze českou společností je zde i výborná podpora při změně českých zákonů a české legislativy.
- Poměrně jednoduchá instalace informačního systému snižuje náklady na implementaci.
- Snadný síťový provoz bez nutnosti instalace na jednotlivých stanicích.
- Možnost i vzdálené správy a automatizace zálohovacích procesů.
- Podpora světových jazyků v celém prostředí informačního systému, legislativ a standardů.
- Podpora mezinárodních účetních standardů US, GAAP a IFRS.
- Manažerské vyhodnocování umožňuje vytvářet v prostředí MS Office libovolné reporty a přehledy.
- Podpora standardů elektronické komunikace jako např. EDI, XML, XSL, SSL, HTTP, HTTPS a další.
- Možnost šifrování a elektronického podpisu dat.
- Integrace s ostatními aplikacemi včetně MS Office aplikací.
- Nabízí možnosti návazných řešení v oblasti softwarových řešení např. Internetový obchod, Rezervační systém pro hotely a cestovní kanceláře a mnoho dalších. V oblasti hardwarových řešení např. Řešení pro mobilní obchodníky, Čtečky a tiskárny čárových kódů, Terminály docházkových systémů a mnoho dalších.

4.2 Možnosti analýzy dat v rámci systému Helios IQ

Analýzu dat lze provést i pomocí standardně implementovaných nástrojů v informačním systému. Tyto nástroje však poskytují informace pouze v textové podobě.

Zadávání dat do Heliosu IQ obvykle začíná zadáním nové organizace nebo zákazníka do evidence. Následuje zadání objednávky popřípadě již Expedičního příkazu, Příjemky na sklad, Dodacího listu a Výdejky. Posledním krokem je obvykle vystavení faktury a zaúčtování dokladu. Jednotlivé doklady mají mezi sebou vazbu a lze je zpětně dohledat. Avšak zobrazení například prodejů v daném roce nebo měsíci je již komplikované.

Uživatel systému standardně vidí seznam dokladů v daném období (viz. Obrázek 2).

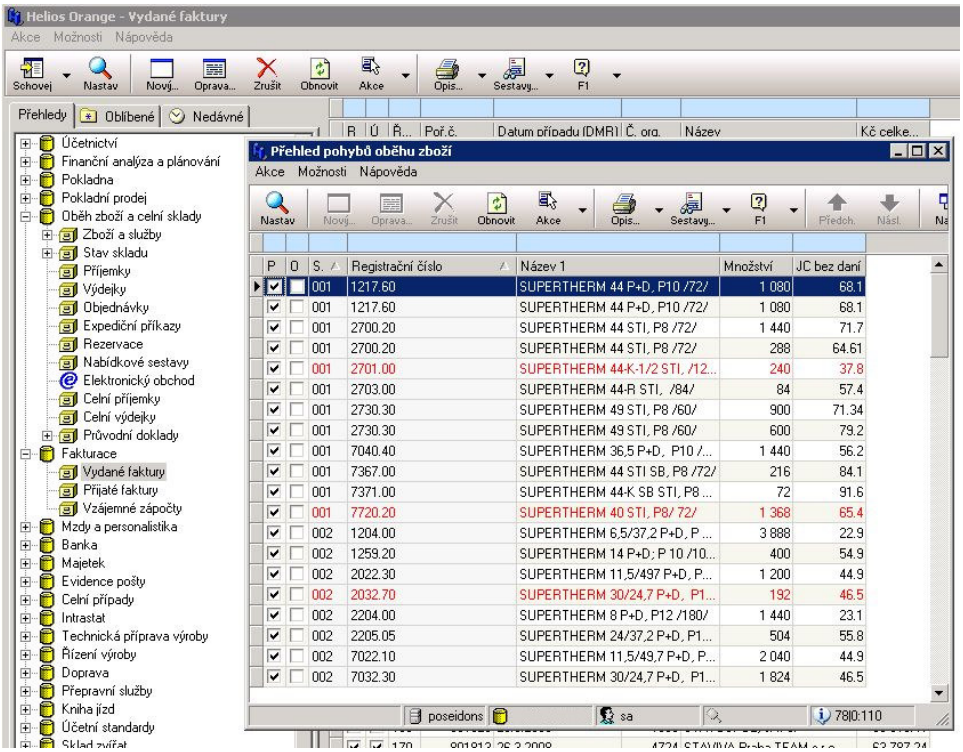


Obrázek 2 - Seznam dokladů v daném období

Jak je vidět z tohoto seznamu nelze zjistit jednotlivé prodeje zboží. Zobrazuje se zde pouze hlavička dokladu s názvem odběratele a cena celkové faktury. Toto by teoreticky mohlo stačit pro analýzu prodejů konkrétnímu zákaz-

níkovi. Zde je možnost si daného zákazníka vyfiltrovat za pomoci filtrů umístěných nad hlavičkami sloupců nebo povolit seskupování.

Jednotlivé položky pak lze zobrazit, pokud označím doklady a přes pravé tlačítko vyberu z menu Přehled položek (viz. Obrázek 3).



Obrázek 3 - Zobrazení položek

Zde již vidíme prodané zboží a stejně jako v předchozím seznamu můžeme i tento seskupovat podle vybraných sloupců. Avšak nevýhody tohoto zobrazení jsou:

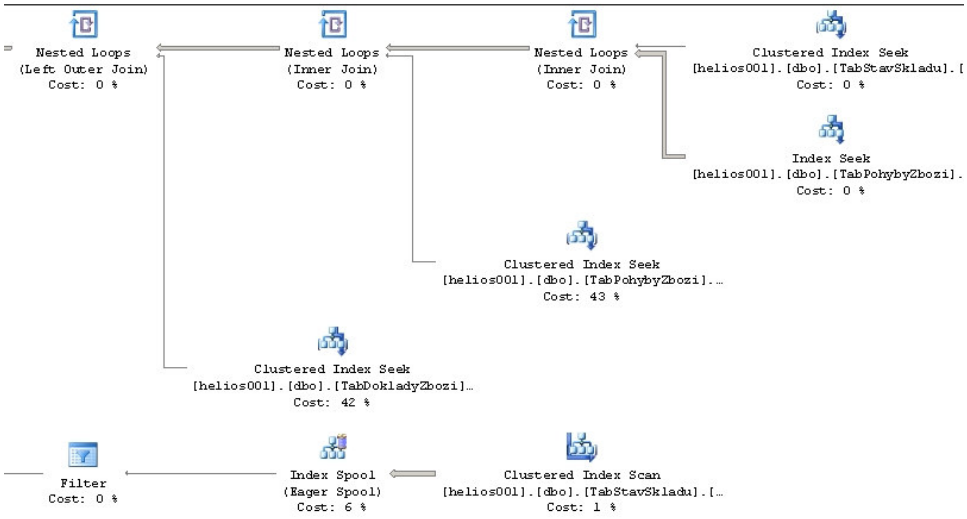
- Pokud chci vidět všechny prodeje ve vybraném období, musím označit všechny doklady v seznamu a následné zobrazení položek trvá neúměrně dlouho.
- Nelze zobrazit jiné období než předem vybrané.
- Zobrazuje prodej pouze na aktuálním skladu.

Poslední zmíněnou nevýhodu lze eliminovat pomocí přehledu přes všechny sklady. Bohužel, i v tomto případě musím opět označit všechny položky a vybrat přehled přes všechny sklady. Ale opět se ukazuje další nevýhoda těchto přehledů. Jejich vygenerování může trvat i více než několik minut, jelikož jde obvykle o velké množství dat a také o velmi složité dotazy.

```
SELECT

    CAST(CASE when (( Select ( TabStavSkladu.MnozBezVyd -
TabStavSkladu.InventuraMn - TabPohybyZbozi.Mnozstvi) from TabStavSkladu
WITH(NOLOCK) Where TabPohybyZbozi.IDZboSklad=TabStavSkladu.ID
AND TabPohybyZbozi.DruhPohybuZbo=11 ) <= 0 ) AND
(TabPohybyZbozi.SkupZbo <> '999')  THEN 1 else

    CASE when ((Select (TabStavSkladu.MnozBezVyd -
TabStavSkladu.InventuraMn) from TabStavSkladu WITH(NOLOCK) Where

    TabPohybyZbozi.IDZboSklad=TabStavSkladu.ID  ) <= 0 ) AND
(TabPohybyZbozi.SkupZbo <> '999') THEN 1 ELSE  NULL END end AS
TINYINT) AS _HeIQ_Barva FROM TabPohybyZbozi

LEFT OUTER JOIN TabStavSkladu VPohybyStavSkladu ON
TabPohybyZbozi.IDZboSklad=VPohybyStavSkladu.ID

LEFT OUTER JOIN TabDokladyZbozi VDokladyZboziPohybyZbozi ON
VDokladyZboziPohybyZbozi.ID=TabPohybyZbozi.IDDoklad

WHERE  (VPohybyStavSkladu.IDKmenZbozi=4675) ORDER BY 11 DESC
```

Jak je vidět z tohoto hodně zkráceného dotazu spojuje hned několik tabulek a samotný je poměrně nečitelný. Celý dotaz viz příloha A. Takový dotaz velmi

zatěžuje server. Jelikož jde o on-line požadavek do databáze, musí server také on-line vytvořit takzvaný prováděcí plán.

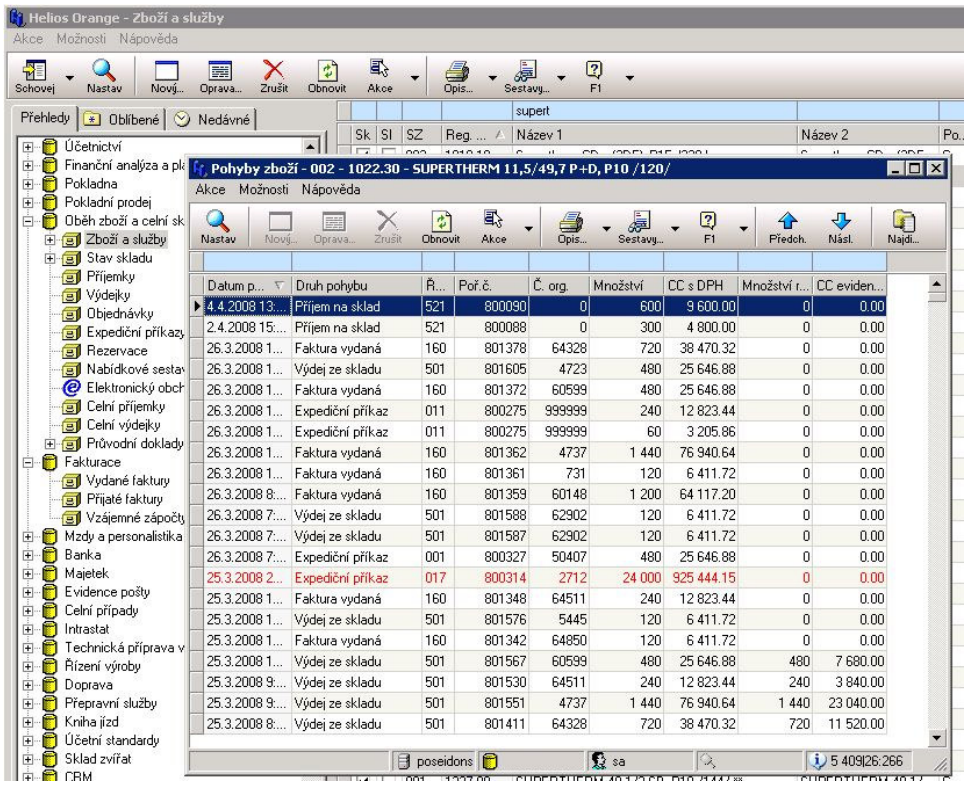


Obrázek 4 – Výřez prováděcího plánu

SQL server každý dotaz rozčlení do menších uzlů podle tabulek, omezovací podmínky, počítaných sloupců. Vznikne jakýsi strom, podle kterého dotaz následně vykoná.

Další možností, jak zobrazit jednotlivé prodeje za položku, je ze seznamu „Zboží a služby“. Výběrem položky pro kterou chceme zobrazit prodeje a zvolíme „Pohyby aktuálního zboží“. A opět musíme čekat, než se seznam zobrazí. I zde platí stejný problém jako v předchozím případě. Opět jde o on-line požadavek a s tím spojená i velká zátěž serveru. Pak si teprve vybereme druh dokladu, který nás zajímá (viz obrázek 5).

Pokud chceme vidět prodeje za určité měsíce, musíme přes volbu „Nastav“ zobrazit sloupec „Datum Případu (M)“, který obsahuje pouze hodnotu měsíce 1 až 12 a podle tohoto sloupce všechny položky seskupit.



Obrázek 5 - Pohyby aktuálního zboží

Dovedu si představit, že pro manažera, který potřebuje mít data rychle a přehledně toto není dostačující. Nehledě na vysokou zátěž serveru, která znemožní v dané chvíli práci ostatním uživatelům.

Další možností, která se nabízí jsou definované přehledy, ale ani ty neposkytují takový komfort jaký je potřeba pro rychlou práci. Pro manažera je navíc detailní pohled zbytečný. A aby z něj dostal globální pohled, musí ho přenést například do aplikace Excel a dále upravit.

Stejným způsobem jako v předchozím případě bychom zobrazili i další požadované výstupy. Jako jsou například přehledy zásob na skladě, pohledávky, závazky a další. Tyto bychom již hledali v jiných sekcích informačního systému.

4.3 Manažerské rozhraní v rámci Helios IQ

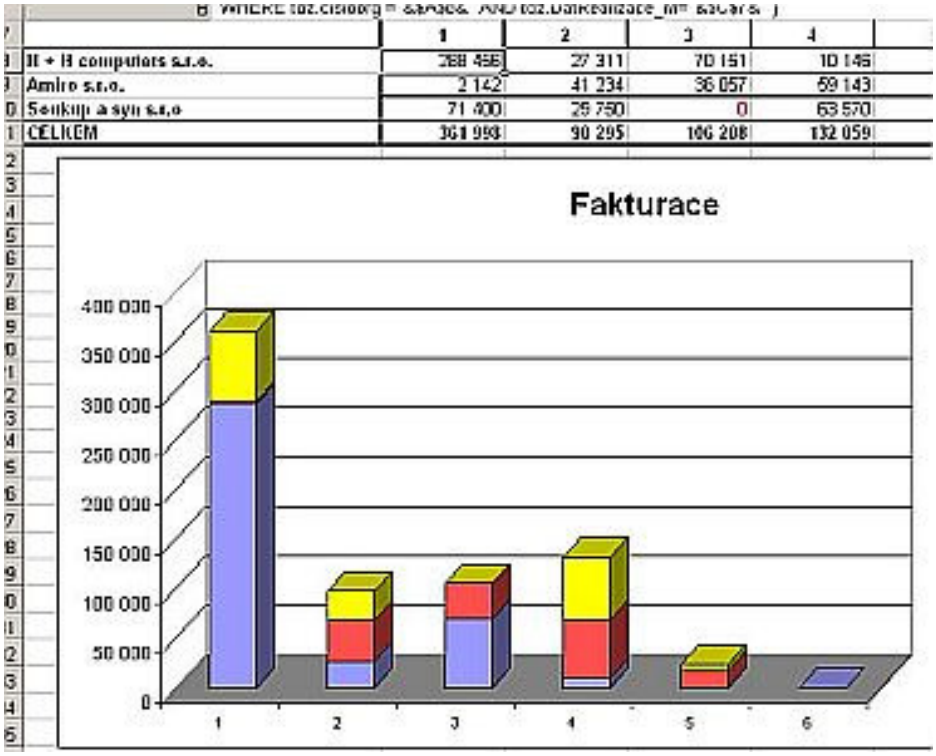
Helios IQ nabízí také modul manažerského rozhraní. Pro funkci tohoto modulu je však nutné mít nainstalovanou sadu MS Office. Jednotlivé analytické přehledy se tvoří v aplikaci MS Excel.

Helios IQ poskytuje pomocné funkce pro tuto aplikaci pomocí kterých se zadávají dotazy do informačního systému. Pomocné funkce se zobrazují jako menu v liště pracovních tlačítek v aplikaci Excel (viz. Obrázek 6).

Obrázek 6 - pomocná tlačítka

Je zde nabídka od různých výpočetních funkcí přes funkce pro tvorbu SQL dotazů. Nejpoužívanější funkcí v tomto případě bude nejspíše IHEQuery do které se dává přímý dotaz pro databázi informačního systému. Po natažení dat již s nimi můžeme dále pracovat a tvořit případné grafy (viz obrázek 7).

Ani toto rozhraní není zcela vyhovující, jelikož vyžaduje dostatečné znalosti v oblasti jazyka SQL. Je nutná také znalost databázové struktury informačního systému. Měla jsem možnost seznámit se s touto strukturou podrobněji a musím podotknout, že jde o poměrně složitou databázi. Vyznat se v ní vyžaduje dost času a pochopit jednotlivé vazby mezi jednotlivými tabulkami. Tím pádem i samotná tvorba takového přehledu je poměrně náročná a zdlouhavá.



Obrázek 7 - Tvorba v Excel

Ukázka zadání dotazu pro vybrání celkové ceny z faktury v MS Excel:

='C:\Program Files\Microsoft Office\Office\LIBRARY\HELIQMR.XLA'!IQQUERY("SELECT sum(tdz.sumakcpozao) FROM Tabdokladyzbozi tdz WHERE tdz.cisloorg ="&A8&" AND tdz.DatRealizace_M="&C7&"")

Velkou nevýhodou tohoto řešení jsou on-line pohledy do databáze. Jelikož jde o poměrně složité dotazy. Opět tedy jako v předchozím řešení dochází k neúměrnému zatížení serveru při zpracovávání požadavků. Takové reporty zatěžují i celý informační systém a omezují i ostatní uživatele.

5 OLAP

Z anglického „On-line Analytical Processing“. Při hledání nejvhodnějšího způsobu pro výpočet analýz, jsem jako nejvhodnější zvolila způsob tvorby datových kostek pomocí OLAP analýz. Tento způsob se mi zdál nejvhodnější z toho důvodu, že je součástí licence SQL serveru. Při tvorbě kostky se data uloží do „skladiště“ mimo informační systém a jakákoliv práce s daty nijak neomezuje práci a ani nezatěžuje server.

5.1 Historie OLAP

5.1.1 Bussines Intelligence

BI je relativně nový termín. Konceptem je jednoduše využívat informace, které jsou již k dispozici ve firmě pomocí rozhodovacích pravomocí, jako pomoc k lepšímu a rychlejšímu rozhodování.

Analýzy se vlastně dělaly odjakživa. Jen s tím rozdílem, že dříve nebyla k dispozici potřebná technika a tak se veškeré analytické procesy vypracovávaly papírově. Počátkem roku 1980 se začaly používat první tabulkové kalkulační programy.

5.1.2 Systémy OLTP (Online Transaction Processing)

Jejich úkolem bylo umožnit uživatelům vykonávat co nejvíce transakcí on-line. Cílem těchto systémů je co největší automatizace každodenních činností v podnikání. Tyto systémy jsou velmi oblíbené. K velkému množství dat zde může přistupovat v reálném čase velké množství uživatelů.

Nad tímto systémem lze vytvořit aplikace OLAP, avšak je zde problém s mnoha relačně svázanými tabulkami a velkým množstvím takto uložených dat. Analýzy v těchto systémech jsou neefektivní a pomalé.

5.1.3 Systémy MIS (Management Information Systems)

Hlavní funkcí tohoto systému je poskytovat kvalitní informace řídícím pracovníkům. Do tohoto systému vstupují data z transakčních systémů. Požadavky na určité sestavy se odesílaly vývojářům systému MIS, kteří je vytvořili a poskytli manažerům. Avšak toto bylo velmi zdlouhavé a vytvořené sestavy již často byly neaktuální a tedy i nepoužitelné.

5.1.4 Systémy DSS (Decision-Support Systems)

Jedná se o systémy pro podporu rozhodování. Obvykle se jedná o nadstavbu nad systémy MIS. Poskytují manažerům výsledky poměrně složitých analýz.

5.1.5 Systémy EIS (Executive Information Systems)

Jeho aplikace byly velmi populární. EIS umožňovalo kopírování klíčových dat z různých reportů do "Dashboard", takže manažer mohl vidět data na první pohled. Jde o systémy pro vrcholové řízení. Ale cíl byl stále stejný. Pomáhat v rozhodování tvůrcům. Později, EIS aplikace byly nahrazeny DSS (Rozhodovacím Systémem Podpory) aplikací, které dělaly v podstatě totéž.

5.1.6 OLAP (On-line Analytical Processing)

Typické využití systémů OLAP je pro analýzu velkého množství dat. Výsledkem jsou souhrny a reporty, které slouží jako podklady manažerům pro rozhodování nebo řízení v různých odvětvích firem. Využívá takzvaného multidimenzionálního databázového modelu.

5.1.7 Multidimenzionální databáze

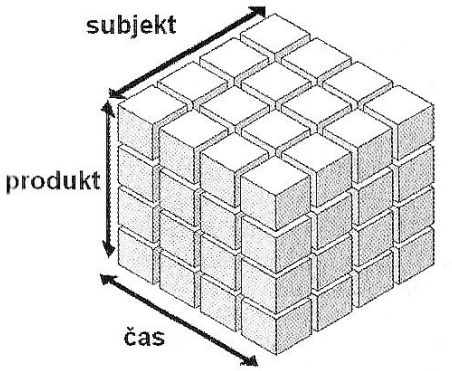
Většina údajů v Informačních systémech je organizována v dvourozměrných relačních databázích. Výsledkem analýzy dat a agregace je multidimenzionální datová struktura nebo-li krychle, která slouží jako podklad pro analy-

tické aplikace. V podstatě jde o ekvivalent tabulky v relační databázi. Do těchto databází se ukládají již upravená a očištěná data od nepotřebných informací.

5.2 Struktura krychle OLAP

Každá krychle OLAP je tvořena dvěma druhy údajů. Tabulkou faktů a přidružených dimenzí. Společně vytváří určitá schémata, například schéma hvězdy (Star schema) (více v kapitole 5.4.1), schéma sněhové vločky (snowflake schema) (více v kapitole 5.4.2) a jiné.

Každá krychle má několik dimenzí (více v kapitole 5.2.2), ale oproti geometrické krychli může mít i více než tři dimenze (viz. Obrázek 8).



Obrázek 8 - Zobrazení dimenzí OLAP

Další důležitou hodnotou krychle jsou takzvané „Measures“, tedy měrné jednotky (více v kapitole 5.2.3). Jde o hodnoty jako je množství, cena a další.

5.2.1 Tabulka faktů

Tato tabulka obsahuje největší množství dat ze všech tabulek v multidimenzionální databázi. Pro mou práci jako tabulka faktů poslouží vytvořený pohled „ZRProdej“. Struktura pohledu viz příloha B. Jedná se o tabulku, která obsahuje data na nejnižší možnou úroveň. Obsahuje také cizí klíče, jež se vážou k jednotlivým dimenzím. Například k dimenzi organizací nalezneme v tabulce faktů sloupec IDOrg nebo-li ID organizace. Obsahuje ale také dále-

žité informace o prodejích, které využijeme jako měrné jednotky nebo-li „Measures“. Těmito informacemi jsou například množství, cena bez DPH nebo cena s DPH.

5.2.2 Dimenze

Tabulky dimenzí obvykle obsahují číselníky. Například dimenzí může být číselník organizací nebo číselník zboží. Dimenze kostky reprezentují rozdílné kategorie pro analýzu dat. Datová kostka není omezena jen na pár dimenzí. Například kostky vytvořené v MS OLAP services mohou obsahovat až 64 dimenzí. Hodnoty v dimenzních tabulkách se obvykle příliš často nemění.

Dimenze obvykle mají stromovou strukturu skládající se z několika podúrovní například:

Produkt

- *Druh Produktu*
    - *Kategorie*
        - *Název produktu*

Jedna z nejdůležitějších dimenzí obvykle bývá dimenze časová. Ta může mít následující strukturu:

Čas

- *Rok*
    - *Kvartál*
        - *Měsíc*
            - Týden
                - Den

Každá dimenze musí obsahovat sloupec s jedinečnou hodnotou. V relačních databázích se těmto hodnotám říká primární klíč. Každá z těchto hodnot se může v tabulce faktů vyskytnout libovolněkrát. To se nazývá cizí klíč. Avšak v tabulce dimenze musí být hodnota jedinečná. Jde tedy o vazbu 1:N.

5.2.3 Measures (měrné jednotky)

Měrné jednotky jsou kvantitativní informace v databázi, které mají být analyzovány. Jejich vytvoření je důležité. Určují konkrétní hodnoty například ceny, množství a další. Zdrojové sloupce s hodnotami se nachází v tabulce faktů.

5.3 Způsoby uložení dat v OLAP systému

5.3.1 Relační databáze (ROLAP)

Při použití způsobu uložení dat ROLAP data zůstávají v původních relačních databázích. Oddělená sada relačních tabulek je použita k uložení agregací. ROLAP je vhodný pro rozsáhlé databáze nebo na stará data, která nejsou často analyzována a měněna.

Výhodou této databáze je možnost nasazení pro různé software, používání vývojových nástrojů, generování reportů a použitelnost v transakčních databázích a datových skladech.

Nevýhoda je omezení údajů ke kterým můžeme přistoupit v reálném čase a chybějící komplexní analytické nástroje.

5.3.2 Multidimenzionální databáze (MOLAP)

Je to multidimenzionální způsob uložení dat s vysokým výkonem. Data jsou ukládána na OLAP server. MOLAP poskytuje nejlepší výkon ve fázi dotazování (analýzy), je speciálně optimalizován pro mnoho dimenzionálních dotazů. Je vhodný pro malé až středně velké objemy dat, kdy kopírování všech dat

do multidimenzionálního formátu nevyžaduje výrazně dlouhou dobu nebo nespotřebovává příliš mnoho diskového prostoru.

Výhodou tohoto způsobu je tedy rychlý přístup k velkému objemu údajů. Možnost komplexních analýz a přístup k multidimenzionálním a relačním strukturám dat.

Nevýhodou je vyšší nárok na kapacitu úložiště a problémy při změnách dimenzí.

5.3.3 Hybridní databáze (HOLAP)

HOLAP slučuje prvky z předešlých dvou přístupů MOLAP a ROLAP. Ponechává původní data v relačních tabulkách, ale ukládá agregace v multidimenzionálním formátu. HOLAP poskytuje propojení mezi rozsáhlými objemy dat v relačních tabulkách a zároveň nabízí výhodu rychlejšího výkonu multidimenzionálně uložených agregací.

Tzv. „Příčky“ rozdělují kostku do částí, které mohou být optimalizovány individuálně, ale následně může být kostka analyzována jako celek. Každá kostka se skládá alespoň z jedné části, ale může být rozdělena i do více částí. Každá část potom může být uložena rozdílným způsobem. Např. kostka, která má tři části. Jedna používá ROLAP, další HOLAP a třetí MOLAP.

5.4 Schéma uložení dat v OLAP

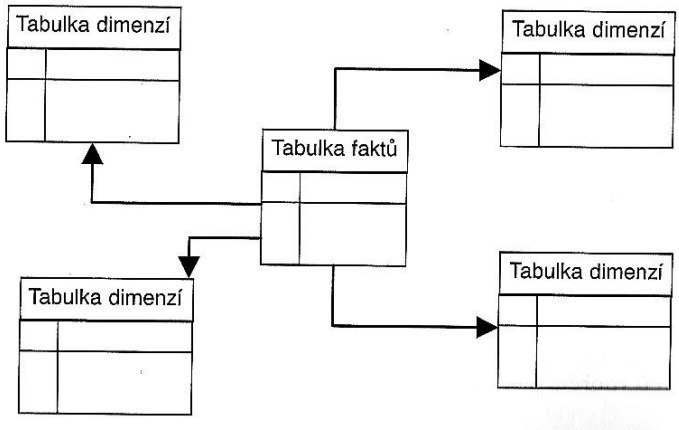
Každá dimenze má s tabulkou faktů propojení pomocí primárních a cizích klíčů. Vzniká tak uspořádání, kterému říkáme schéma. Nejčastěji se využívá schéma „Hvězdy“ (viz 5.4.1) nebo „Sněhové vločky“ (viz. 5.4.2).

5.4.1 Star (Hvězda)

Skládá se z tabulky faktů a dimenzí vzájemně propojených pomocí primárních klíčů v dimenzích a cizích klíčů v tabulce faktů. Grafické vyjádření sché-

matu připomíná hvězdu, s tabulkami dimenzí zobrazenými v paprskovité struktuře okolo centrální tabulky faktů.

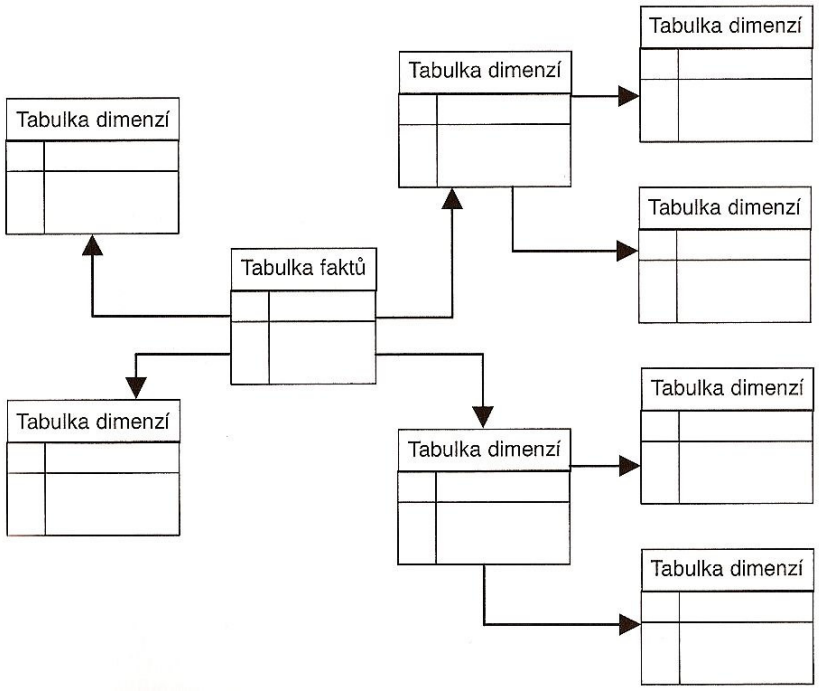
Ve hvězdicovém schématu je každá dimenze reprezentována právě jednou tabulkou. A každá tabulka obsahuje několik atributů. Např. dimenze „čas“ může mít tyto atributy: den, měsíc, kvartál, rok.



Obrázek 9 - Schéma hvězda

5.4.2 Snowflake (Sněhová vločka)

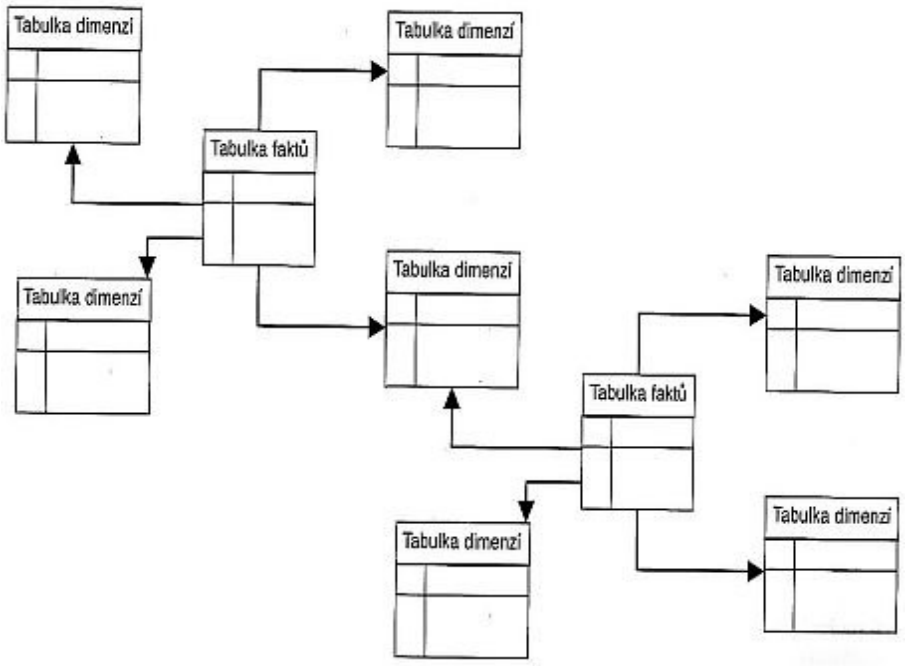
Skládá se z tabulky faktů a dimenzí, které mohou mít přes relační vazby připojené další dimenze. Výsledné grafické schéma pak vytváří tvar podobný sněhové vločce. Hlavní rozdíl mezi těmito dvěma modely spočívá v tom, že tabulky dimenzí jsou rozdělené do dalších tabulek, aby snížily redundance v uložených datech. Takováto tabulka je snadno udržovatelná a šetří diskový prostor. Ovšem tato úspora je zanedbatelná ve srovnání s typickou velikostí tabulky faktů. Navíc toto schéma může snižovat efektivnost analýz dat, neboť je zapotřebí provést více spojení tabulek, aby mohl být dotaz proveden. Proto může být výkon systému nepříznivě ovlivněn. Z tohoto důvodu není schéma sněhové vločky tak časté, při návrhu datového skladu, jako hvězdicové schéma.



Obrázek 10 - Schéma sněhová vločka

5.4.3 Fact Constellation (Souhvězdí)

V některých případech se můžeme dostat do situace, že aplikace bude vyžadovat více tabulek faktů, které budou sdílet mezi sebou tabulky dimenzí. Toto schéma může být zobrazeno jako soubor hvězd a proto se nazývá „Souhvězdí“ (Constellation).



Obrázek 11 - Souhvězdí

6 Tvorba OLAP kostek

Existují dva způsoby, jak datovou kostku vytvořit. Každý je vhodný pro různé typy analýz.

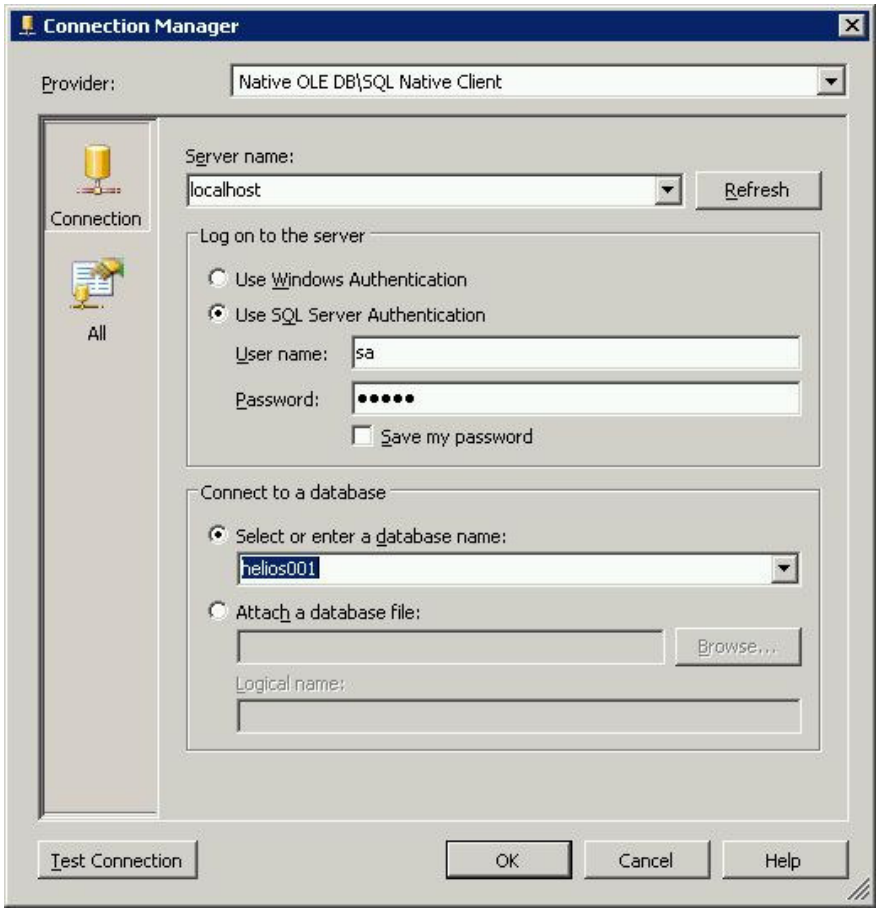
Prvním způsobem je tvorba pomocí předdefinovaných formulářů. Tento způsob je vhodný pro standardní řešení analýz. Ve starších verzích SQL se datové kostky tvořily přímo v programu Analysis services. Od verze SQL 2005 slouží tento program pouze k prohlížení vytvořených kostek a datové kostky se tvoří pomocí programu Visual Studio 2005. Více v kapitole 6.1.

Druhý způsob je vhodný pro složitá a nestandardní řešení. Vyžaduje již znalost programování. Tvorba datové kostky probíhá pomocí jazyka MDX. Pomocí tohoto způsobu lze také vytvořit lokální kostku. Více v kapitole 6.2.

6.1 Tvorba OLAP pomocí předdefinovaných formulářů

Důležitou informací při tvorbě kostky je připojení ke zdrojovým datům. Tímto určíme k jaké zdrojové databázi se kostka připojí a odkud bude brát svá data.

Ve formuláři nejprve vybereme server, na kterém se nachází databáze SQL s informačním systémem Helios IQ. Dále zvolíme uživatele s právem pro přístup a čtení na databázi. Já jsem zvolila hlavní administrátorský účet na SQL serveru, jelikož jde o můj testovací server, ke kterému se nikdo kromě mě připojovat nebude. Není tedy nutné zde tvořit speciální uživatele. Pokud by šlo o server se skutečnými přístupy je záhodno vytvořit speciálního uživatele z důvodu potřebného zabezpečení serveru.



Obrázek 12 - Připojení k databázi Helios

Měrné jednotky

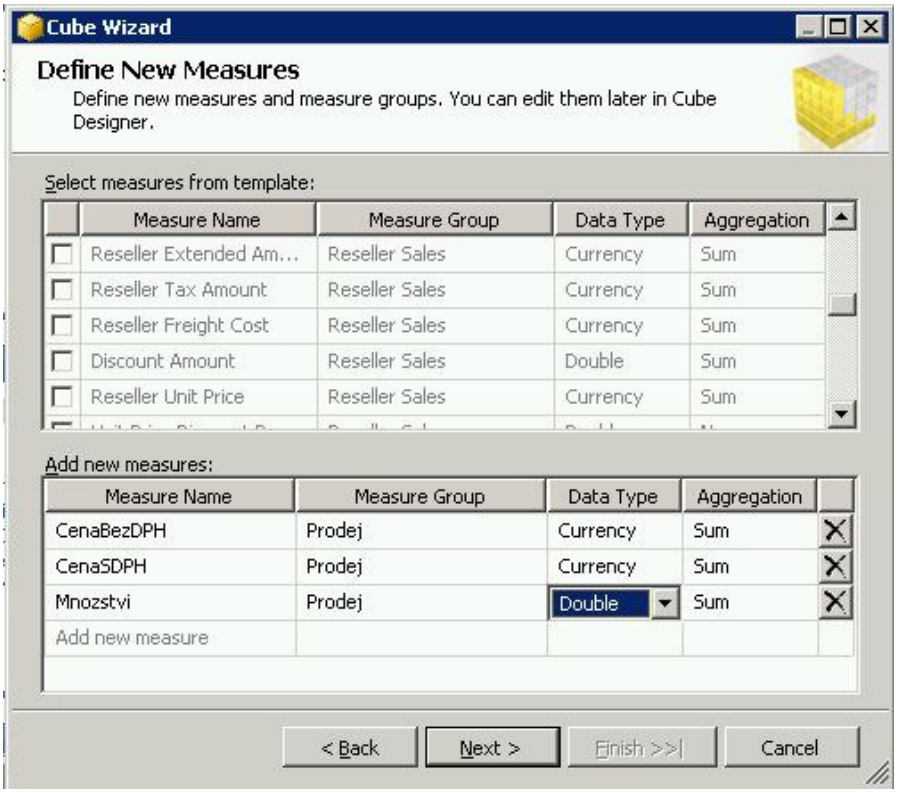
Dalším důležitým krokem je vytvoření měrných jednotek (Measures). Jde o hodnoty pro analýzu v podstatě nejdůležitější. Například ceny s DPH, ceny bez DPH, množství a další. Záleží na potřebě dané firmy, jaké hodnoty budou důležité pro zobrazení reportů.

Například pro zobrazení prodejů bude manažera asi nejvíc zajímat za kolik v určitém období která organizace nakupovala. Nebo také, jaké množství se kterého zboží prodalo. Proto jsem vytvořila 3 měrné jednotky. Tento počet pro mou ukázku tvorby OLAP kostky bude dostačující. Ve skutečném prostředí

firmy je nutná analýza potřeb oné firmy a na jejich základě se vytvoří i příslušný počet měrných jednotek i dimenzí.

První vytvořená měrná jednotka určuje cenu bez DPH, druhá je cena s DPH a třetí je Množství. Samozřejmě je možné, kdykoliv v průběhu tvorby i později, přidat další měrné jednotky podle potřeby manažerů.

Na obrázku 13 je vidět definice měrných jednotek. U každé se nadefinuje její typ a můžeme je také seskupit do různých skupin. Měrné jednotky můžeme vybrat i z předdefinovaných variant. Já jsem zvolila postup vytvoření vlastních jednotek, které jsem seskupila do skupiny nazvané „Prodej“.



Obrázek 13 - Definice měrných jednotek

Dimenze

Jednotlivé dimenze nám poslouží pro různé filtrování dat a zobrazování k analýze prodejů. Pro tento příklad jsem zvolila pouze 4 dimenze z důvodu jednoduchosti a přehlednosti:

- **Faktura Typ** – rozděluje data na Fakturu a Dobropisy
- **Organizace** – číselník organizací podle nějž si můžeme zobrazit jednotlivé prodeje jednotlivých zákazníků (organizací)
- **Sortiment** – číselník zboží, které daná firma prodává
- **Datum** – toto bude časová dimenze, která bude udávat jednotlivá období rozdělená podle roků, měsíců, kvartálů, dnů a týdnů. Pro tuto dimenzi jsem, pomocí jednoduchého dotazu, vytvořila zvlášť pohled z důvodu rychlejší práce s touto dimenzí. Data pro tuto dimenzi se nacházejí přímo v tabulce faktů.

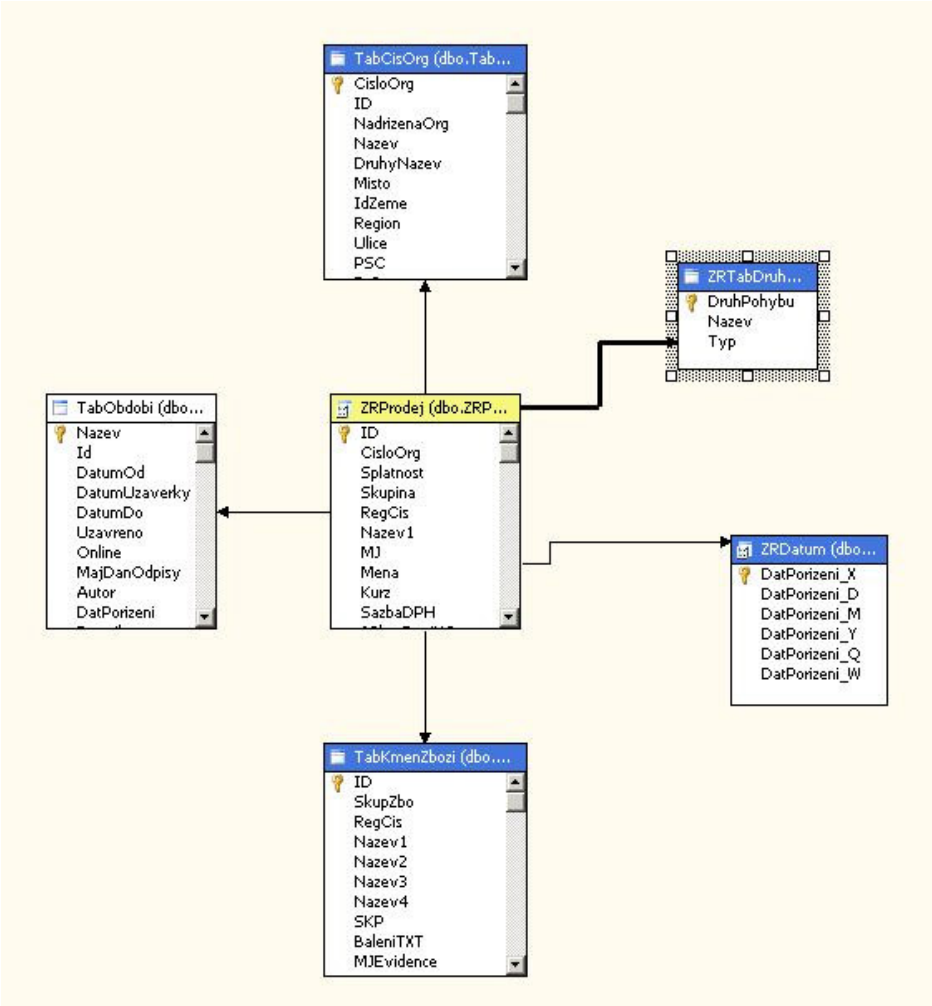
Samozřejmě dimenzí můžeme mít mnoho. Pro tento příklad bych mohla mít například ještě dimenzi Skladů, pro rozlišení prodejů na různých skladech. Celkový návrh dimenzí se odvíjí od požadavků jednotlivé firmy.

Jelikož jsem vytvořila Časovou dimenzi přímo z tabulky, která zároveň slouží i jako tabulka faktů, musela jsem zajistit, aby se jednotlivé datumové položky v pohledu neopakovaly. To se zajistí v dotazu jediným slovíčkem „DISTINCT“. Toto slovo zajistí zobrazení jedinečných položek.

```
SELECT DISTINCT
DatPorizeni_D, DatPorizeni_M, DatPorizeni_Y, DatPorizeni_Q,
DatPorizeni_W, DatPorizeni_X
FROM    dbo.TabDokladyZbozi
```

Některé firmy také pracují podle „hospodářského roku“, který začíná i končí jinak než běžný rok. Například může začít 1. června a končit 31. května. Informační systém Helios IQ umožňuje použití tohoto přístupu a tato data udržuje v jedné tabulce, která se nazývá TabObdobi. Pokud bychom chtěli zobrazovat data podle „Hospodářského roku“, stačí přidat výše zmíněnou tabulku jako další dimenzi.

Na obrázku 14 je vidět rozmístění tabulek a jejich relace vůči hlavní tabulce faktů. Z obrázku je také jasné, že jde o použité schéma „Hvězdy“.



Obrázek 14 – Rozmístění tabulek s vazbami

Pro různé druhy datových kostek se budou lišit i jejich dimenze podle jejich požadavků a možností. Podle složitosti jednotlivých dimenzí se bude měnit i použité schéma.

Dalším příkladem datové kostky by mohly být finanční zásoby firmy. Tedy stav na bankovních účtech. Tabulka, obsahující požadovaná data je TabKontaD. Zde bychom tvořily jiné dimenze než u prodejů. Jednou z dimenzí v tomto případě bude číselník účtů. Jelikož tabulka obsahuje data až na úroveň jednotlivých středisek, můžeme vytvořit dimenzi „Střediska“ pro možnost podrobných výstupů. Dále můžeme vytvořit i dimenzi „Měna“. Z této kostky tedy můžeme následně získat data, jak o stavu na účtu v bance, ale i finanční spotřebu a přínosy uvnitř ve firmě.

Ve verzi SQL 2000 a nižší bylo možné mít u každé kostky jak vlastní dimenze, tak i jednotlivé dimenze, které bylo možno mezi jednotlivými kostkami sdílet. Tedy využívat jednu dimenzi na více místech. Takovou dimenzí v obou těchto příkladech by mohla být dimenze „Datum“. Obě datové kostky pracují hlavně s časovými údaji. Nabízí se zde tedy využití právě sdílené dimenze. Od verze SQL 2005 jsou již všechny dimenze sdílené. Na to je potřeba dát si při tvorbě datové kostky pozor.

Při vytváření je také nutné nastavit oprávnění pro přístup uživatelů k vytvořené datové kostce. Způsob autorizace závisí na nastavení (SQL, Doména, Lokální účty, jiná LDAP kompatibilní autorita nebo zprostředkovatel).

6.2 Tvorba OLAP pomocí jazyka MDX

Jazyk MDX - Multidimenzional Expresion (Multidimenzionální výrazy) je ekvivalentem jazyka SQL. Cílem tohoto jazyka je navigace v multidimenzionálních údajích. Pomocí dotazu MDX můžeme vypsat vybrané údaje do dvourozměrné tabulky a nebo tvořit celé kostky.

Struktura jazyka MDX je velmi podobná jazyku SQL. Avšak oproti jazyku SQL je podstatně jednodušší. Například syntaxe příkazu SELECT se skládá pouze z klauzulí FROM a WHERE. Lze jí zapsat takto:

SELECT <<set>> ON COLUMNS,
<<set>> ON ROWS,
<<set>> ON PAGE
FROM <<cube>>
WHERE <<tuple>>

Dotaz zobrazující jednoduchou tabulku by tedy mohl například vypadat takto:

SELECT
{[Druh Pohybu].[Faktura],[Druh Pohybu].[Dobropis]} ON COLUMNS,
{[Organizace].[Nazev]} ON ROWS
FROM Helios001

Pomocí klauzulí ON COLUMNS a ON ROWS se definují dimenze. Jazyk MDX může pracovat až se 128 dimenzemi. Lze je definovat buď slovně (ROW, COLUMNS, PAGES,…) a nebo číselně pomocí klíčového slova AXIS.

Definice příkazu by mohla pak vypadat například takto:

{[Organizace].[Nazev]} ON AXIS(1)

Důležitým a také povinným příkazem je výběr příslušné krychle pomocí klauzule FROM.

V případě potřeby, lze výsledná data omezit pomocí klauzule WHERE. Pokud bych například chtěla omezit výsledná data na určitý sortiment, pak bych k předchozímu příkladu přidala příkaz:

WHERE ([Sortiment].[Baterka])

Pomocí jazyka MDX lze také vytvořit lokální krychle OLAP. Toto se může hodit v případě, kdy manažer hodně cestuje a potřebuje prezentovat datové analýzy i mimo svou firmu. Jednou možností je, že se manažer připojí pomocí VPN (Virtual Private Network) do své firmy, ale to vyžaduje, aby v daném místě, kde hodlá prezentovat své reporty, měl připojení k internetu a povolený vzdálený přístup do firmy. Lepší možností je tedy tvorba lokální kostky.

V podstatě jde o vytvoření souboru s příponou CUB. Tento soubor si může manažer stáhnout například z intranetu než odjede z firmy a následně ho může připojit k aplikaci MS Excel na svém notebooku aniž by musel být připojen k internetu. S lokální krychlí může pracovat úplně stejně, jako by se díval na krychli uloženou na serveru.

Proces vytvoření lokální kostky se skládá z několika kroků.

Prvním krokem je vytvoření struktury samotné kostky. K tomu slouží příkaz CREATE CUBE. Definují se zde jednotlivé dimenze, měrné jednotky (measures) a případné počítané položky.

Struktura tohoto příkazu vypadá takto:

CREATE CUBE <název krychle> (
DIMENSION <název> [TYPE <typ>]
LEVEL <název>[TYPE <typ>], LEVEL <název>[TYPE <typ>]…
MEASURE <název> FUNCTION <název>…)

V části příkazu DIMENSION se definují jednotlivé dimenze. Jednou z vlastností u definice dimenze je úroveň dimenzí. Definuje se zde hierarchická struktura dané dimenze. Důležité je pořadí definice. Úrovně je třeba definovat od nejvyšší po nejnižší. Například hierarchická struktura u dimenze organizace by mohla vypadat takto:

…
DIMENSION [Organizace],
, LEVEL [All Organizace] TYPE ALL
, LEVEL [Nazev]
, LEVEL [Cislo Org]
…

V případě časové dimenze, je třeba u jednotlivých úrovní nadefinovat i standardní časovou hierarchii. Definice časové úrovně bude mít následující tvar:

DIMENSION [Datum] TYPE TIME,
, LEVEL [All Datum] TYPE ALL
, LEVEL [Rok] TYPE YEAR
, LEVEL [Kvartal] TYPE QUARTER
, LEVEL [Mesic] TYPE MONTH
, LEVEL [Den] TYPE DAY

V každé krychli vedle dimenzí musí existovat i měrné jednotky. Bez nich by krychle vlastně ani neměla smysl. Na měrné jednotky se aplikují agregační funkce jako jsou SUM, COUNT, MAX a MIN.

MEASURE [Mnozstvi] FUNCTION SUM,

Takto vytvořená kostka však neobsahuje žádná data. K naplnění údaji definované krychle pomocí příkazu CREATE CUBE, slouží příkaz INSERT INTO. Samotný příkaz se opět od jazyka SQL příliš neliší.

Syntaxe tohoto příkazu je:

INSERT INTO <název krychle>
(
<název dimenze>.<název úrovně>,[<název dimenze>.<název úrovně>,…]
<název měrné jednotky>, [<název měrné jednotky>,…]
)
OPTIONS <seznam voleb>
SELECT <seznam sloupců>
FROM <seznam tabulek>
WHERE <omezovací podmínky>

Pomocí těchto příkazů lze tedy vytvořit skripty, které budou generovat lokální kostky podle potřeby jednotlivých manažerů. Ukázka jak by mohl vypadat kompletní funkční skript v jazyce VBScript, generující lokální kostku včetně dat, je v příloze C.

6.3 Zhodnocení obou způsobů

První způsob tvorby kostky pomocí předdefinovaných formulářů je vhodný pro standardně tvořené analýzy. Mohou je také tvořit lidé neznalý programování. Bohužel, tímto způsobem nelze tvořit velmi složité analýzy ani přenositelné lokální kostky.

Druhý způsob tvorby pomocí jazyka MDX je již určen pro velmi složité analýzy. Pro tvorbu kostky tímto způsobem je již nutná znalost programování a také velmi dobrá orientace ve struktuře tabulek v informačním systému. Výhodou tohoto způsobu je možnost vytvoření lokální přenositelné kostky.

Oba způsoby mají své pro a proti. Pokud není tvůrce dostatečně znalý programování, doporučuji využít prvního způsobu. Ve velké většině analýz je tento způsob dostačující a tvůrce se v něm snadno zorientuje. Pro vytvoření lokální kostky je nutné použít druhý způsob. Doporučuji držet se vytvořeného příkladu v příloze C. Skript byl již otestován v praxi a osvědčil se.

7 Práce s OLAP kostkami

V této kapitole se budu zabývat způsoby zobrazení výsledků z OLAP kostek. Je mnoho možností jak zobrazit data. Jedním ze způsobů zobrazení a práce je přímo v samotném programu Visual Studio (viz. 7.1) nebo SQL Management Studio (viz. 7.2). Tyto programy jsou určené vývojářům a běžní uživatelé je nemají.

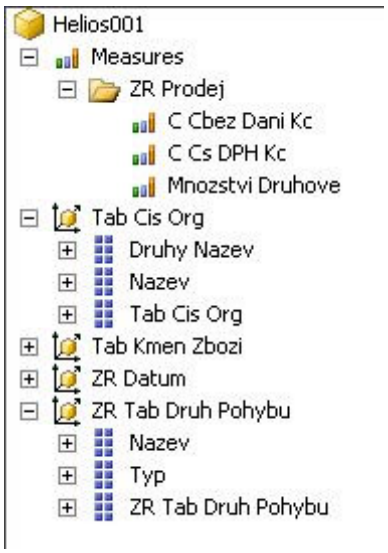
Jednou z dalších dostupných možností je zobrazení přes www rozhraní (viz. 7.5). Zde máme dva způsoby. Lze využít již hotové komponenty, která je vhodná pro standardní a jednoduché analýzy (viz. 7.5.2). Nebo za pomoci jazyka MDX vytvořit vlastní výstupy (viz. 7.5.1). Toto řešení je vhodné pro složité analýzy a náročné výstupy.

Další možností, které se také budu věnovat, je zobrazení v Sharepoint (viz. 7.6). Toto řešení je vhodné pro implementaci do velkých firem, kde jsou již vyšší nároky na samotný provoz serverů i software.

7.1 Zobrazení ve Visual Studio

Zobrazení přes vývojový program je určeno výhradně lidem, kteří datové kostky vytvářejí. Běžní uživatelé nemají tuto aplikaci nainstalovanou a pro neznalého uživatele je práce s programem zbytečně složitá.

Data si může vývojář zobrazit přímo v aplikaci Visual studio pod záložkou „Browser“. Zobrazí se pracovní plocha, kde vlevo jsou vytvořené dimenze. Ty lze myší přetáhnout do pracovního okna na požadované místo.



Obrázek 15 - Zobrazení dimenzí

Na obrázku 15 jsou zobrazeny vytvořené dimenze. Z těchto dimenzí můžeme vybírat jaké informace chceme zobrazit. Můžeme například vybrat Název z dimenze „Tab Druh Pohybu“. Ta obsahuje položky Dobropis a Faktura. Při přetažení této položky do řádků nebo sloupců uvidíme kolik z měrné jednotky připadá na fakturu nebo na dobropis. Tyto výběry jsou zcela individuální a záleží na konkrétní potřebě manažera nebo obchodníka co potřebuje v dané situaci vidět.

Na dalším obrázku 16 je vidět už konkrétní hodnoty přetažené do okna „Browseru“ tak, jak by si je mohl zobrazit uživatel v jiných aplikacích. Tyto výběry se provádí jednoduchým přetažením myši, proto je pro uživatele tato varianta přehledů velice jednoduchá a intuitivní.

Jak už jsem zmínila v začátku této kapitoly, pro běžného uživatele toto zobrazení nepřipadá v úvahu a je tedy nutné najít jiné řešení.

Sem přetáhněte pole filtrů.

	Nazev		
	Faktura	Dobropis	Celkový součet
Nazev	C Cbez Dani Kc	C Cbez Dani Kc	C Cbez Dani Kc
Václav Budín stavební firma	105724		105724
VARIOS plus - Hoffman Petr	35920		35920
"BALTIC" ZACZKIEWICZ DOROTA	249408		249408
ABADIA a.s.	3113386.64	-219336.6	2894050.04
ADAM CAR s.r.o.	48615.68		48615.68
AG Konstrukt s.r.o.	205919.37		205919.37
Agra Březnice s.r.o.	17731.91	-1080	16651.91
Agra Deštná	9000		9000
AGRO Brno-Tuřany a.s.	309923.58		309923.58
Agro Popice, zemědělská výroba, služby□□Petr Korous	52000		52000
AGRODRUŽSTVO Žimutice	19269.92		19269.92
Agros Vraný, družstvo vlastníků	87041.66		87041.66
Agrospol Hrádek plus družstvo	39550		39550
AGROSTA Mlékojedy s.r.o.	77870.58		77870.58
ALDAST,spol. s.r.o.	512101		512101
Aleš Šorfa - ALSTAV	250952.8	-5940	245012.8
Ambrož Jan	231488.8		231488.8
Antonín Buman	10809385.63	-4574.02	10804811.61
Antonín Smola STAVEBNÍ A ŠTUKATÉRSKÁ FIRMA	2400		2400

Obrázek 16 - Zobrazený výstup

7.2 Zobrazení přes SQL Management Studio

Někteří vývojáři dávají přednost zobrazování výsledků pomocí dotazů. I zde je toto možné za použití jazyka MDX. Součástí SQL nástrojů je Management Studio pomocí nějž se můžeme na data v kostce podívat právě pomocí jazyka MDX.

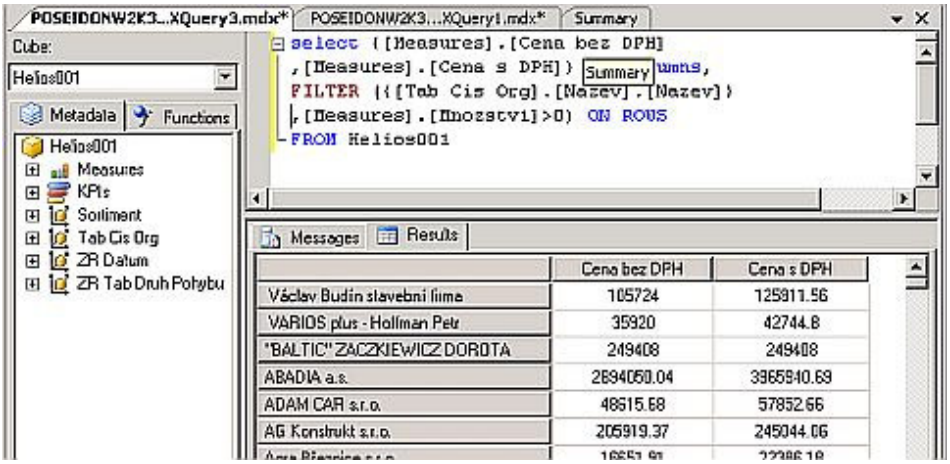
Dejme tomu, že si potřebuji zobrazit za kolik určitá organizace v naší firmě nakoupila. Budou mě tedy zajímat měrné jednotky Cena bez DPH a Cena s DPH. Tyto jednotky zobrazím do sloupců. Do řádek pak zobrazím do organizace. Dotaz na datovou kostku bude tedy vypadat následovně:

select {[Measures].[Cena bez DPH],[Measures].[Cena s DPH]} ON Columns,

{[Tab Cis Org].[Nazev].[Nazev]} ON ROWS

FROM Helios001

Výsledek tohoto dotazu je zobrazen formou tabulky viz.obrázek 17.



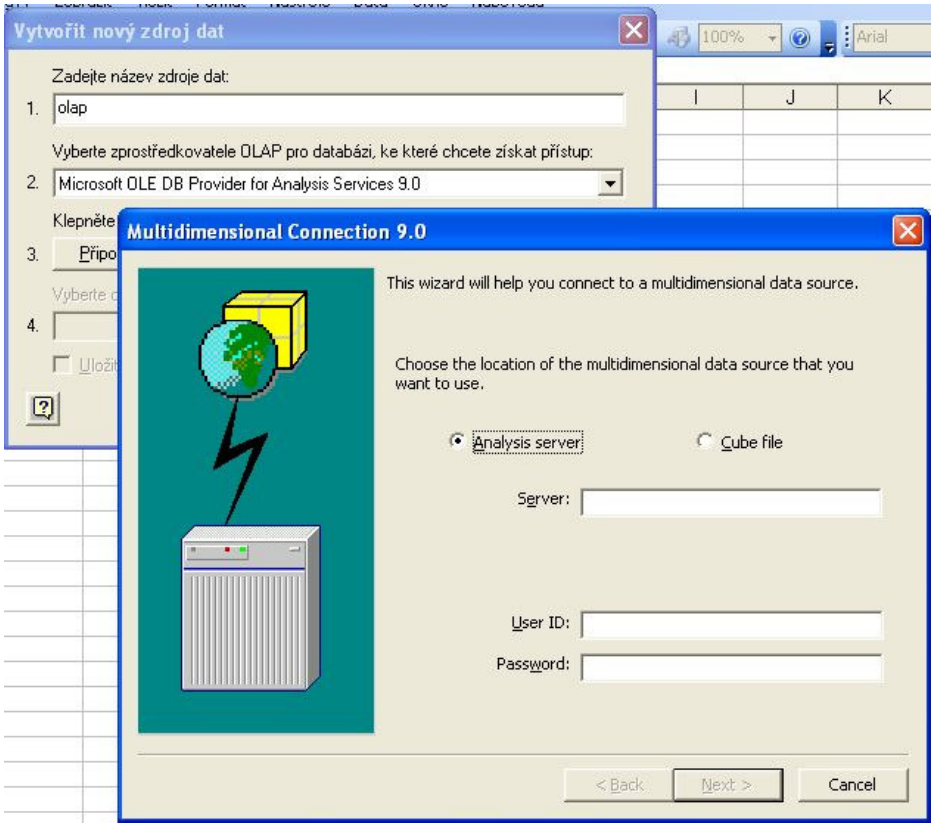
Obrázek 17 - Zobrazení přes SQL Management Studio

Ani toto řešení není určeno pro běžného uživatele. Vývojář si zde může vyzkoušet zda datovou kostku vytvořil správně.

7.3 Zobrazení v aplikaci Excel

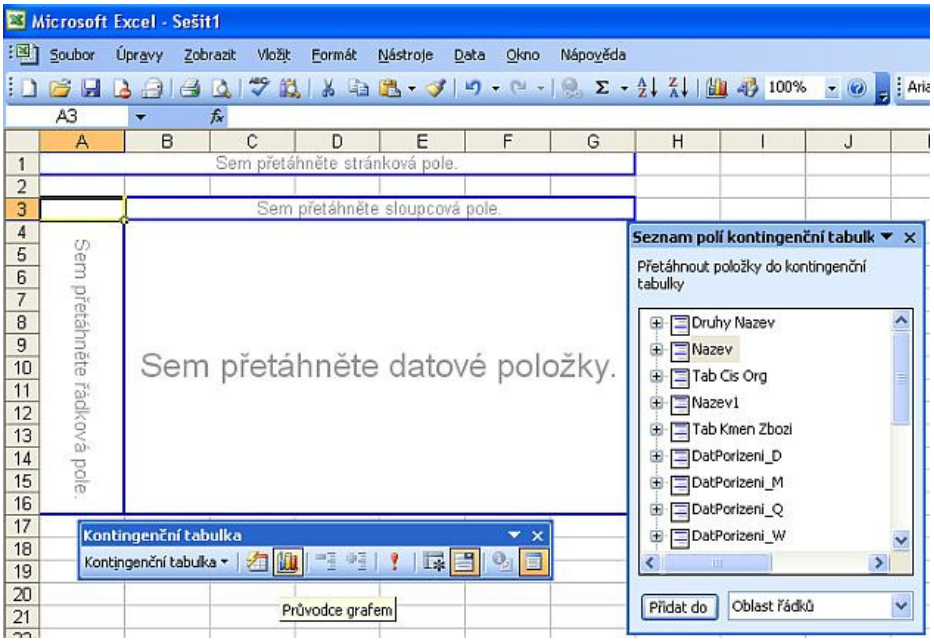
K datové kostce se můžeme připojit i pomocí programu Microsoft Excel, který je součástí balíku Microsoft Office. Práce s datovou kostkou probíhá pomocí takzvaných „Kontingenčních tabulek“.

Abych se mohla připojit k datové kostce, musím při vytváření „Kontingenční tabulky“ vybrat z nabídky místo, kde se nacházejí data, volbou „Externí zdroj dat“. Na další obrazovce už nadefinuji konkrétní připojení k datové kostce. Důležité je vybrat, jako zprostředkovatele OLAP pro databázi, Microsoft OLE DB Provider for Analysis Services 9.0. Starší verze tohoto ovladače se nedokáží k databázi serveru SQL 2005 připojit. Definici připojení zobrazuje obrázek č. 18.



Obrázek 18 - Definice připojení kostky

Po načtení dat se uživateli zobrazí prázdná tabulka a okýnko se seznamem dimenzí až do jejich detailů a měrné jednotky (cena bez DPH, cena s DPH a množství). Uživatel si může libovolně pouze přetažením myši nadefinovat tabulku podle potřeby. Například přetažením názvu organizace do levého sloupečku, datumové dimenze do horní řádky a ceny s DHP do středu. Získáme analýzu jednotlivých firem a jejich nákupů v určitém časovém období v naší firmě. Jednoduchým přepnutím si uživatel může zobrazit data i ve formě grafu. Je pouze na jeho vůli a na možnostech navržené kostky jaká data chce zobrazit.



Obrázek 19 - Nastavení výstupu v Excel

7.4 Zobrazení pomocí Report serveru

Jednou z dalších možností zobrazení výstupů z datové kostky je pomocí Report serveru, který je součástí licence MS SQL serveru. K této aplikaci přistupujeme pomocí prohlížeče. Musí být nakonfigurována na stejném serveru jako je instalace SQL serveru.

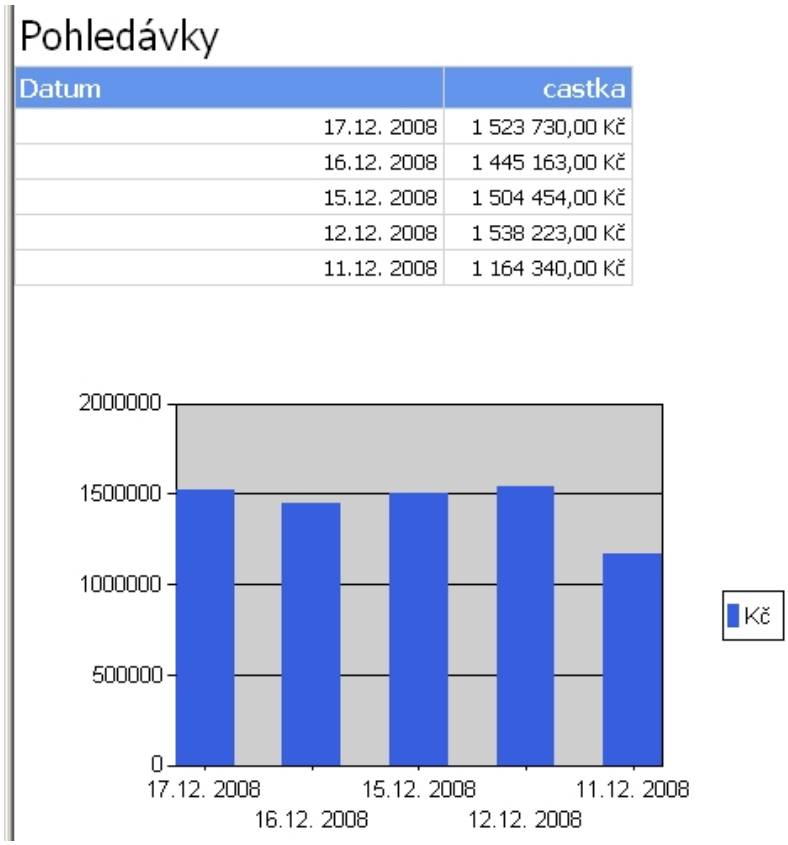
Reporty můžeme tvořit dvěma různými způsoby. Jednou z možností je aplikace Report builder k níž se lze dostat odkazem ze stránky Report serveru. Bohužel, tento program se mi nepodařilo rozchodit ani za pomoci odborně vzdělaného kolegy. Tudíž nemohu porovnat, jaké poskytuje možnosti a co jí chybí.

Druhou možností tvorby reportů je přes Visual Studio. Tato varianta je rozsáhlejší i co do možnosti samotné tvorby. Lze zde využít také předdefinovaných formulářů a nebo, pokud má tvůrce reportu vyšší znalost prog-

ramování, může si celý výstup vytvořit podle potřeby. Po vytvoření reportu se vypublikuje na web Report serveru.

Report server má ještě jednu výhodu, kterou bych ráda zmínila. Lze v něm tvořit i on-line reporty, které lze naplánovat třeba na noční hodinu, kdy je provoz v informačním systému nejmenší nebo žádný. Takový report se pak uloží do paměti serveru než si jej uživatel odtud stáhne. Lze tím částečně nahradit výstupy z OLAP kostek.

Výsledné reporty lze exportovat do několika formátů (pdf,tif,xls,...). Nevýhodou je, že nelze přenést celá data a pracovat s filtry reportů jinde než na serveru. Jiné prohlížeče než Internet Explorer mají trochu problém s grafikou těchto výstupů. Nicméně jsou reporty funkční i v nich.



Obrázek 20 - Možné výsledné zobrazení reportu

7.5 Zobrazení přes www rozhraní

Pro zobrazení OLAP datové kostky můžeme využít dvou způsobů zobrazení. Jedním ze způsobů je využití jazyka MDX. Kdy přesně definujeme jak má výstup vypadat. Pokud chceme uživateli povolit, aby si sám mohl definovat výstupy musíme již rozhraní naprogramovat. Lze také využít komponenty OWC, která uživateli umožní definici vlastního výstupu.

7.5.1 Zobrazení OLAP kostky pomocí jazyka MDX

Jak už jsem zmínila v kapitole 6.2 lze k práci s datovou kostkou využít jazyka MDX. Toho zde využiji. Pomocí jednoduchých dotazů získáme potřebná data a následně za pomoci například jazyka ASP je „naservírovat“ uživateli již v přijatelné podobě. Toto je jeden z nejjednodušších přístupů k datové kostce.

Jeho výhodou je, že není nutné mít instalované další podpůrné aplikace na klientském počítači. Zobrazit výsledná data si může ve svém prohlížeči každý, který má oprávnění danou kostku vidět.

Dejme tomu, že si uživatel přeje vidět za kolik určitá organizace v naší firmě nakoupila. Budou mě tedy zajímat měrné jednotky Cena bez DPH a Cena s DPH. Tyto jednotky zobrazím do sloupců. Do řádek pak zobrazím organizace. Dotaz na datovou kostku bude tedy vypadat následovně:

select {[Measures].[Cena bez DPH],[Measures].[Cena s DPH]} ON Columns,

{[Tab Cis Org].[Nazev].[Nazev]} ON ROWS

FROM Helios001

Tento dotaz má však jednu vadu na kráse. A to, že zobrazuje všechny organizace z číselníku bez ohledu na to jestli u nás nakoupily nebo ne. A tabulka je

tak zbytečně dlouhá. Proto jsem do dotazu ještě doplnila filtr, který zobrazí v tabulce pouze organizace, které mají hodnotu větší než nula. Výsledný dotaz tedy vypadá takto:

select {[Measures].[Cena bez DPH],[Measures].[Cena s DPH]} ON Columns,

FILTER ({[Tab Cis Org].[Nazev].[Nazev]},[Measures].[Mnozstvi]>0) ON ROWS

FROM Helios001

Výsledné zobrazení vypadá pak takto:

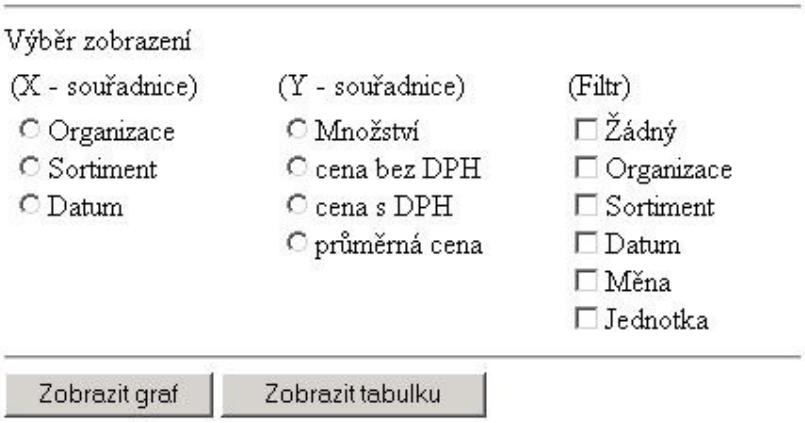
	Mnozstvi	Cena bez DPH	Cena s DPH
Václav Budín stavební firma	118	105724	125811.56
VARIOS plus - Hoffman Petr	602	35920	42744.8
"BALTIC" ZACZKIEWICZ DOROTA	6984	249408	249408

Obrázek 21 - zobrazení stránky pomocí MDX

Ukázka celého skriptu viz. Příloha D. Nevýhodou je, že takto vytvořené přehledy uživatel nemůže měnit ani nastavovat si podle vlastní potřeby. Je nutné všechna zobrazení definovat předem.

Pokud bych chtěla uživateli umožnit volbu v nastavování přehledů, musím naprogramovat vlastní rozhraní. Toto rozhraní se skládá z jednoduchého formuláře, kde si uživatel vybere jaké dimenze a v jaké ose chce použít.

Rozhraní by mohlo vypadat například takto:



Obrázek 22 - návrh rozhraní pro ovládání OLAP kostek

Po odeslání se požadavek zpracuje a podle zvolených nastavení se vytvoří MDX dotaz na datovou kostku. Zobrazení výsledku pak již probíhá stejně jako v předchozím případě.

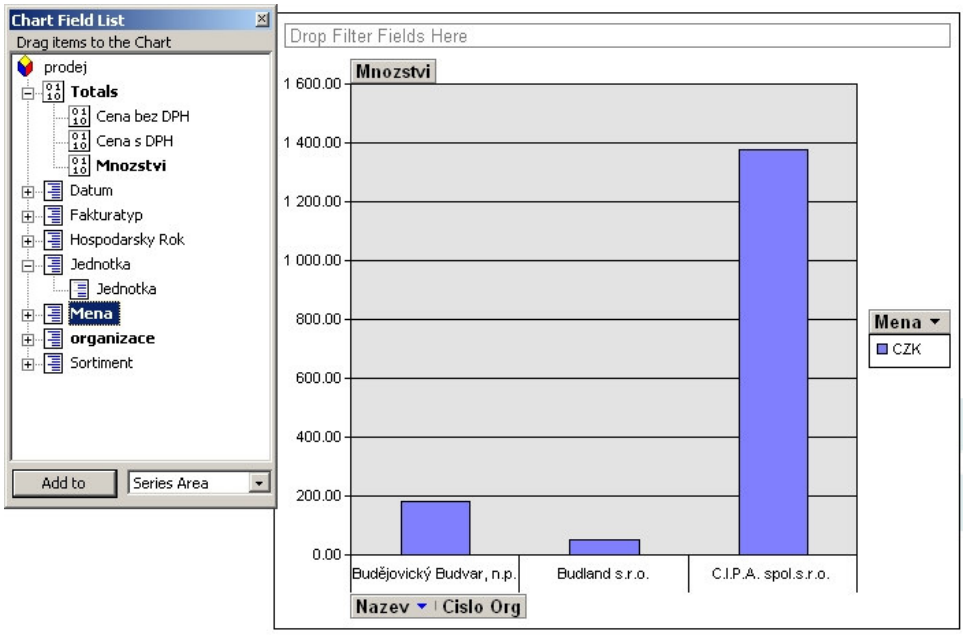
V obou předchozích příkladech jde pouze o textové zobrazení. Pokud chci uživateli umožnit komfortnější zobrazení pomocí grafů, musím si vypomoci již dalším produktem.

Jako nejvhodnější pro grafické zobrazování se jeví SVG (Scalable Vector Graphics). Jde o grafický formát původně navržený specielně pro www grafiku. Dnes se již používá i v jiných oblastech. Jde o XML aplikaci pomocí níž můžeme vytvořit jednoduché grafy. Zde si již, ale musí každý instalovat tento modul do svého prohlížeče. Stále již, ale přibývá prohlížečů, které SVG podporují nativně a není již tedy nutná instalace pomocného modulu.

7.5.2 Zobrazení OLAP kostky pomocí komponenty OWC

Office Web Components se skupinou OLE tříd je implementován jako ovládací prvek ActiveX v aplikacích Microsoft Office 2000, Office XP a Office 2003. Tyto ovládací prvky ActiveX můžou být zapojeny do webových

stránek, Visual Basic a VBA. Díky této komponentě můžeme tvořit tabulky, grafy a další. Práce s ní je velmi jednoduchá a individuální stejně jako u zobrazení přes Excel nebo Visual Studio.



Obrázek 23 - Zobrazení pomocí komponenty OWC

Vývoj Office Web Components byl přerušen s příchozím Office 2007 a nejsou v něm již zahrnuty. Pouze jako součást sady Office Project Server 2007. Avšak jsou i nadále k dispozici pro stažení na internetových stránkách společnosti Microsoft. Společnost Microsoft nemá zatím kompletní náhradu za Office Web Components. Nicméně lze používat libovolnou kombinaci produktů třetích stran, Excel služby nebo Visual Studio Tools pro sadu Office, které poskytují podobnou funkčnost.

Verze této komponenty se liší podle klientů. Někteří klienti vyžadují komponentu verze 10 a jiní verzi 11. Obvykle to záleží na verzi instalace balíčku Microsoft Office. Na serveru můžeme mít nainstalované obě verze komponenty. Avšak je potřeba dát pozor při vývoji a toto zohlednit. Pro správný běh této komponenty je nutné, aby uživatel měl na svém počítači nainstalované MS

Office a přístup pro čtení u serverů s aplikací a SQL . Tato podmínka tak zamezuje libovolnému použití na internetu. Jde tedy o čistě interní aplikace v rámci firem.

Bohužel, tato komponenta nefunguje ani s jinými prohlížeči než je Internet Explorer. Toto lze obejít programováním pomocí jazyka MDX. Tím, ale uživateli již pevně definujeme co uvidí a co ne. Případně můžeme umožnit částečné možnosti volby pomocí standardních formulářových výběrů a tak upravovat dotaz do datové kostky a její následné zobrazení.

7.6 Zobrazení v Sharepoint

7.6.1 Co je Sharepoint

Už z názvu Sharepoint (sdílený bod) je vidět, že jde o technologii na způsob intranetových aplikací. Snaží se sdružovat informace na jedno místo a tím nejen umožnit snadný přístup, ale i díky vyhledávacím technologiím není problém potřebné informace nalézt velmi rychle.

Je to webová informační služba běžící na technologii .NET. K ukládání všech svých dat využívá MS SQL server. Nástroj spolupracuje se serverovými službami. Umožňuje uživatelům tvořit si vlastní prostředí www stránek. Jde tedy o aplikaci server-klient. Klientská část je „tenký klient“ v podobě www stránek přístupných přes internetový prohlížeč.

7.6.2 Předpoklady pro instalaci Sharepoint

Sharepoint pro svůj běh potřebuje .NET framework a IIS verze 6. Tyto předpoklady splňuje Windows 2003 Server a vyšší. Instalace na nižší verze Windows serverů nebude úspěšná. Tato aplikace tedy vyžaduje vyšší pořizovací náklady jak na Hardware tak i na Software.

Další důležitá instalace je MS SQL 2000 a vyšší. V případě mé práce je toto kritérium splněno. Instalaci jsem provedla na MS SQL 2005.

7.6.3 Windows Sharepoint services (WSS)

Tato služba je součástí licence operačního systému Windows 2003 server. Není však součástí standardní instalace. Pokud chceme tuto službu využívat musíme jí dodatečně nainstalovat. Poskytuje základní platformu pro tvorbu webových aplikací.

7.6.4 Windows Sharepoint portal server (SPS)

Je to portálové řešení využívající služby Windows Sharepoint services. Jde tedy v podstatě o nástroj rozšiřující možnosti výše zmíněné služby. Umožňuje její komfortnější ovládání a práci s aplikací. Toto řešení již není součástí licence Windows 2003 server. Pokud bychom jí chtěli využívat musíme tuto část produktu dokoupit.

7.6.5 Zobrazení OLAP kostek v Sharepoint services

Stejně jako v případě zobrazení přes www rozhraní, existuje i pro Sharepoint komponenta pro zobrazování OLAP kostek. Jmenuje se Office 2003 Web Parts and Components add-in. Tato komponenta však funguje pouze do verze Sharepoint 2.0. Pro novější verzi Sharepointu byl vývoj této komponenty zastaven. Tuto skutečnost však lze obejít ruční instalací komponenty.

Nejprve je třeba získat starou komponentu ze stránek Microsoft nebo z CD dodávaného k Sharepoint 2.0. Ze staženého souboru komponenty jsem získala 3 soubory. A to:

- Gplib.cab_LocaleID
- Quickqt.Cab_LocaleID
- Microsoft_Office_DataParts.Cab_LocaleID

Tyto soubory je třeba přejmenovat na:

- Microsoft.Sharepoint.Solutions.GreatPlains.cab
- Microsoft.Sharepoint.WebParts.QuickQuote.cab
- Microsoft.Office.DataParts.cab

Následně jsem využila aplikace stsadm.exe, která je jednou z důležitých součástí instalace Sharepoint. Tato aplikace zajistí instalaci připravené komponenty do Sharepointu. Nabízí funkce, které jsou přístupné i přes webové rozhraní, ale má i další možnosti, které jinak nelze zavolat než z příkazové řádky. Nabízí plno parametrů. Pro instalaci stažené komponenty Office 2003 Web Parts and Components add-in mě však zajímal jen jeden z nich.

Parametr Addwpack zajistí přidání webové části do webové galerie virtuálního serveru. Každý z jednotlivých parametrů utility má ještě své vlastní parametry:

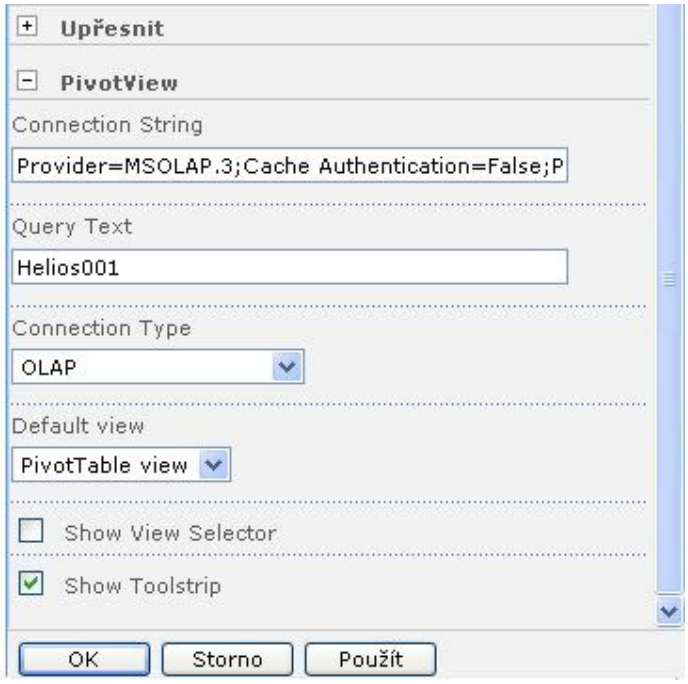
- **filename** – název souboru webové části
- **url** – adresa webového serveru, kam se má webová část uložit
- **globalinstall** – tento parametr zpřístupní webovou část všem virtuálním serverům v rámci instalace Sharepoint
- **force** – pokud již webová část již je nainstalovaná, nelze ji přepsat bez tohoto parametru

Všechny 3 soubory jsem nainstalovala následujícím způsobem:

stsadm.exe -o addwppack -filename Drive:\microsoft.office.dataparts.cab -globalinstall –force

Zvolila jsem globální instalaci, aby jí bylo možné použít ve všech vytvořených virtuálních webech. Po provedení těchto kroků již mohu přes webové rozhraní přidat komponentu na stránku, kde chci údaje z datové kostky zobrazovat.

Po přidání je ještě nutné nastavit parametry pro připojení k datové kostce viz. Obrázek 24.

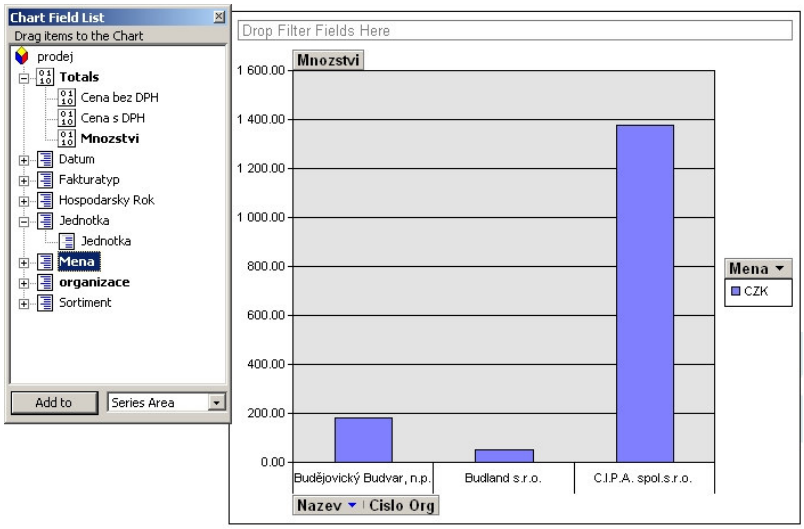


Obrázek 24 - Nastavení OLAP kostky v Sharepoint

Nejdříve je třeba nastavit Connection String nebo-li definici připojení k OLAP databázi. Důležité je nastavit Providera podle verze použitého SQL

serveru. V případě použití SQL 2005 je nutno nastavit providera MSOLAP.3. Dále do pole Query Text se zapíše jméno OLAPové kostky. Ve výběrovém poli Default view lze vybrat, zda chci zobrazovat kostku formou tabulky nebo grafu.

Po té co je vše nastaveno již lze s kostkou pracovat, stejně jako v případě komponenty OWC nebo jako s Kontingenční tabulkou v Excelu. Můžeme myší přetahovat jednotlivé dimenze tam, kam potřebujeme a vytvořit si tak pohled na jednotlivá data. Jednoduchým přepnutím v nastavení této webové části můžeme tabulku změnit na graf a zobrazit si grafické vyjádření nastavených hodnot.



Obrázek 25 - Práce s komponentou v Sharepoint

Práce s komponentou je téměř stejná jako v případě komponenty OWC. Výhodou oproti OWC je zde možnost uložení nastaveného výstupu pro znovu zobrazení této webové části. Nemusíme tedy jako v případě předchozí komponenty vše znovu nastavovat při každém spuštění aplikace.

Nevýhodu v řešení skrze Sharepoint vidím i zde, že jde o záležitost, která správně funguje pouze v prohlížeči Internet Explorer. Ladění pro jiné prohlížeče jako je Firefox, Opera a další je problematické, ale je to možné. Bohužel,

jelikož i u této komponenty byl zastaven její vývoj a tudíž i její podpora, setkávám se zde s problémy kompatibility.

Je možné samozřejmě i v případě Sharepoint naprogramovat vlastní rozhraní, ale to vyžaduje velmi dobré znalosti programování v .NET a velmi dobré znalosti aplikace Sharepoint. Jelikož jde z hlediska programátorského o poměrně složitou aplikaci, není jednoduché se v ní vyznat. I v dnešní době v naší republice existuje jen málo takových programátorů. Já jsem zatím do programování Sharepoint nepronikla a proto si na sestavení rozhraní pro Sharepoint netroufám.

7.7 Porovnání všech variant práce s datovou kostkou

V následující tabulce jsem shrnula všechny výhody a nevýhody jednotlivých typů zobrazení.

Typ zobrazení	**Výhody**	**Nevýhody**
Visual Studio	• Možnost zobrazení při vývoji kostky. • Jednoduché nastavování.	• Pouze vývojová aplikace.
SQL Management Studio	• Zobrazení výstupu pomocí MDX jazyka.	• Pouze vývojová aplikace.
MS Excel	• Jednoduché načtení kostky a práce s ní. • Možnost načtení lokální kostky.	• Nutnost instalace balíku MS Office.

Report Server	• Variabilnost výstupů. • Možnost naplánování generování výstupu na určitou hodinu i den.	• Přístup pouze přes webové rozhraní na příslušném SQL serveru. • Nutná alespoň částečná znalost programování.
WWW rozhraní OWC	• Jednoduchá práce s komponentou.	• Nutná instalace MS Office. • Vývoj komponenty zastaven. • Nekompatibilita s jinými prohlížeči.
WWW rozhraní MDX	• Není nutné instalovat další Software. • Kompatibilita i s jinými prohlížeči.	• Předdefinované výstupy. • Nutná znalost jazyka MDX a programování.
Sharepoint	• Vhodné řešení pro větší firmy. • Jednoduchá práce s komponentou.	• Vývoj komponenty ukončen. • Problém s kompatibilitou.

Z tabulky vyplývá, že pokud firma nevyžaduje speciální funkce a je schopna zajistit na všech počítačích instalaci MS Office, bude stačit využití komponenty OWC, případně Excel. Pro složitější výstupy a při požadavku na kompatibilnost s prohlížeči je nutno využít programování výstupů pomocí jazyka MDX. Zobrazení v Sharepoint je vhodné pouze pro větší firmy, které vyžadují rozdílnou správu serverů. Server v takovém případě bývá umístěn i v jiném státě.

8 Závěr

Výhodou integrovaných nástrojů pro analýzy v rámci informačního systému Helios IQ je zobrazení aktuálního stavu. Bohužel, jsou zde i nevýhody, které použití těchto nástrojů v reálném čase prakticky znemožňují. Mezi tyto nevýhody patří příliš složité dotazy na MS SQL server, jejichž zpracování vyžaduje dostatek času a příliš zatěžují server. Znesnadňují tak práci i ostatním uživatelům informačního systému. I samotné výstupy nejsou příliš praktické a jejich potřebné zpracování vyžaduje dostatečné znalosti struktur tabulek a jazyka SQL.

Cílem mé práce tedy bylo, nalézt a navrhnout jiný způsob tvorby analýz nad tímto informačním systémem. Helios IQ je postavený na MS SQL serveru, proto se jeví jako nejlepší použití MS OLAP, který je součástí licence MS SQL serveru. Tento způsob použití analýz sice nemůže zajistit aktuální zobrazení stavu dat v informačním systému. Ale v tomto případě pro nás „čerstvá“ data ani nejsou příliš důležitá a výsledky analýz ani příliš neovlivní. Data se do datových kostek generují v předem určených časech. Avšak právě díky generování dat do speciálního úložiště, lze s analýzami pracovat kdykoliv je potřeba, aniž by práce zatěžovala informační systém. Z toho vyplývá požadavek na určité volné místo na disku. Toto ale v dnešní době nevidím jako velký problém. Při použití standardních prostředků, práce s OLAP kostkou již pro uživatele není složitá a nevyžaduje ani velkou znalost struktur tabulek ani jazyka SQL jako u integrovaných nástrojů informačního systému.

Zhodnotím-li obě možnosti analýz, jeví se jako nejlepší použití OLAP datových kostek i přes svou „nevýhodu“ předem určených generování dat. Hlavně z důvodu menší zátěže serveru a neomezuje ostatní uživatele informačního systému.

Pro tvorbu jednoduchých a standardních řešení datových kostek bych doporučila tvorbu pomocí předdefinovaných formulářů a následné zobrazení za vy-

užití komponenty OWC pro její jednoduchost. Ovšem při použití speciálních modulů jsem vždy narážela na problém kompatibility s ostatními prohlížeči. Nehledě na to, že firma Microsoft zastavila vývoj těchto modulů a tak ve vyšších verzích MS SQL již nastal problém i se samotným zobrazením vůbec. Doporučuji brát tedy ještě na zřetel hledisko do jaké míry je firma schopna splnit podmínky pro správnou funkčnost této komponenty. Pokud některá z podmínek nebude splněna je programování výstupu pomocí MDX nevyhnutelné. Tedy využít kombinaci tvorby kostky pomocí předdefinovaných formulářů a zobrazení pomocí programování za pomoci MDX.

Pro tvorbu složitých řešení a speciálních požadavků na práci s datovou kostkou doporučuji využít tvorbu i zobrazení za pomoci jazyka MDX. Díky jeho univerzálnosti co se týče kompatibility prohlížečů. Při zobrazení v podobě tabulek není nutná další instalace podpůrných modulů či aplikací. U zobrazení grafiky je to již složitější a doporučila bych použití SVG grafiky, která se v dnešní době stává součástí www prohlížečů.

Co se týče zobrazení v Sharepoint doporučuji využít hotové komponenty. Vlastní programování do této aplikace je velmi složité a proto běžným uživatelům bych prozatím nedoporučovala. Popřípadě si takovou aplikaci nechat naprogramovat od specializované firmy na Sharepoint aplikace.

Reference

[1] J. Reed, M. Stacia, Hitachi Consulting. SQL Server 2005 Analysis services.

[2] Lacko Luboslav. Datové sklady, analýza OLAP a dolování dat. Computer Press, 2003.

[3] Kutěj Tomáš, Sobotka Miloš, Lávička Jan. Technologie Sharepoint 2003 – Implementace, administrace a vývoj. Computer Press, 2000

[4] Pasch Ondřej. Windows Sharepoint Services – hotová řešení. Computer Press, 2005

[5] Malina Patrik. PowerShell – Podrobný průvodce skriptováním

[6] Homer Alex, Sussman Dave, Francis Brian. Active Server Pages 3.0 profesionálně.

[7] Mikle Pavol. XCSS – rozšířené kaskádové styly CSS.

[8] Mikle Pavel. DHTML – referenční příručka.

[9] Šimůnek Milan. SQL – kompletní kapesní průvodce. Computer Press, 2001.

[10] http://support.microsoft.com/

[11] http://www.dbsvet.cz

[12] http://datamining.xf.cz

Přílohy

Příloha A. Ukázka SQL dotazu

SELECT TabPohybyZbozi.ID, TabPohybyZbozi.BlokovaniEditoru, CAST(CASE when ((Select (TabStavSkladu.MnozBezVyd - TabStavSkladu.InventuraMn -TabPohybyZbozi.Mnozstvi) from TabStavSkladu WITH(NOLOCK) Where TabPohybyZbozi.IDZboSklad=TabStavSkladu.ID AND TabPohybyZbozi.DruhPohybuZbo=11) <= 0) AND (TabPohybyZbozi.SkupZbo <> '999') THEN 1 else

CASE when ((Select (TabStavSkladu.MnozBezVyd - TabStavSkladu.InventuraMn) from TabStavSkladu WITH(NOLOCK) Where

TabPohybyZbozi.IDZboSklad=TabStavSkladu.ID) <= 0) AND (TabPohybyZbozi.SkupZbo <> '999') THEN 1 ELSE NULL END end AS TINYINT) AS

_HeIQ_Barva,TabPohybyZbozi.IDZboSklad,TabPohybyZbozi.IDDoklad,TabPohybyZbozi.IDOldDoklad,TabPohybyZbozi.Obal,TabPohybyZbozi.IDOldPolozka,TabPohybyZbozi.DruhPohybuZbo,TabPohybyZbozi.SkutecneDatReal,VDokladyZboziPohybyZbozi.DatPorizeni,VDokladyZboziPohybyZbozi.DruhPohybuZbo,VDokladyZboziPohybyZbozi.RadaDokladu,VDokladyZboziPohybyZbozi.PoradoveCislo,VDokladyZboziPohybyZbozi.CisloOrg,TabPohybyZbozi.Mnozstvi,TabPohybyZbozi.CCsDPHKc,TabPohybyZbozi.MnozstviReal,TabPohybyZbozi.CCevid,VPohybyStavSkladu.IDSklad

FROM TabPohybyZbozi LEFT OUTER JOIN TabStavSkladu VPohybyStavSkladu ON TabPohybyZbozi.IDZboSklad=VPohybyStavSkladu.ID

LEFT OUTER JOIN TabDokladyZbozi VDokladyZboziPohybyZbozi ON VDokladyZboziPohybyZbozi.ID=TabPohybyZbozi.IDDoklad

WHERE (VPohybyStavSkladu.IDKmenZbozi=4675) ORDER BY 11 DESC

Příloha B. Příprava pro tvorbu krychle

```
CREATE VIEW dbo.ZRProdej
AS
SELECT   TOP 100 PERCENT
dbo.TabDokladyZbozi.CisloOrg, dbo.TabDokladyZbozi.Splatnost,
dbo.TabPohybyZbozi.SkupZbo AS Skupina, dbo.TabPohybyZbozi.RegCis,
dbo.TabPohybyZbozi.Nazev1, dbo.TabPohybyZbozi.MJ,
dbo.TabPohybyZbozi.Mena, dbo.TabPohybyZbozi.Kurz,
dbo.TabPohybyZbozi.SazbaDPH, dbo.TabPohybyZbozi.JCbezDaniKC,
dbo.TabPohybyZbozi.JCsDPHKc,
case when
        (dbo.TabDokladyZbozi.DruhPohybuZbo = 14) then -
dbo.TabPohybyZbozi.CCbezDaniKc else
        dbo.TabPohybyZbozi.CCbezDaniKc end as CCbezDaniKc,
dbo.TabPohybyZbozi.CCsDPHKc, dbo.TabPohybyZbozi.DruhPohybuZbo,
dbo.TabDokladyZbozi.ID, dbo.TabPohybyZbozi.IDDoklad,
dbo.TabDokladyZbozi.DUZP, dbo.TabPohybyZbozi.MnozstviDruhove,
dbo.TabPohybyZbozi.CCsSDKcPoSDruhova, dbo.TabDokladyZbozi.Sleva,
dbo.TabDokladyZbozi.Obdobi
FROM        dbo.TabDokladyZbozi INNER JOIN
dbo.TabPohybyZbozi ON dbo.TabPohybyZbozi.IDDoklad =
dbo.TabDokladyZbozi.ID
WHERE    (dbo.TabDokladyZbozi.DruhPohybuZbo = 13) OR
            (dbo.TabDokladyZbozi.DruhPohybuZbo = 14)
ORDER BY dbo.TabDokladyZbozi.DruhPohybuZbo DESC
```

Příloha C. Ukázka skriptu tvorby lokální kostky

Dim sSourceDSN,sCreateCube, sInsertInto,sql,Con, s

sSourceDSN = "SOURCE_DSN=""DRIVER=SQL Server; SERVER= ZDENKA;"

sSourceDSN = sSourceDSN & "UID=sa;DATABASE=HELIOS001"""

sCreateCube = "CREATECUBE=CREATE CUBE [Helios001] ("

sCreateCube = sCreateCube & "DIMENSION [Fakturatyp]"

sCreateCube = sCreateCube & ", LEVEL [All FakturaTyp] TYPE ALL"

sCreateCube = sCreateCube & ", LEVEL [Typ Faktury]"

sCreateCube = sCreateCube & ", LEVEL [Druh Pohybu]"

sCreateCube = sCreateCube & ", DIMENSION [Organizace]"

sCreateCube = sCreateCube & ", LEVEL [All Organizace] TYPE ALL"

sCreateCube = sCreateCube & ", LEVEL [Nazev]"

sCreateCube = sCreateCube & ", LEVEL [Cislo Org]"

sCreateCube = sCreateCube & ", DIMENSION [Sortiment]"

sCreateCube = sCreateCube & ", LEVEL [All Sortiment] TYPE ALL"

sCreateCube = sCreateCube & ", LEVEL [Nazev Sortiment]"

sCreateCube = sCreateCube & ", LEVEL [Reg Cis]"

sCreateCube = sCreateCube & ", DIMENSION [Datum] TYPE TIME"

sCreateCube = sCreateCube & ", LEVEL [All Datum] TYPE ALL"

sCreateCube = sCreateCube & ", LEVEL [Rok]"

sCreateCube = sCreateCube & ", LEVEL [Kvartal]"

```
sCreateCube = sCreateCube & ", LEVEL [Mesic]"

sCreateCube = sCreateCube & ", LEVEL [Den]"

sCreateCube = sCreateCube & ", MEASURE [Mnozstvi] Function Sum
Format '#.#'"

sCreateCube = sCreateCube & ", MEASURE [Cena bez DPH] Function Sum
Format '#.#'"

sCreateCube = sCreateCube & ", MEASURE [Cena s DPH] Function Sum
Format '#.#'"

sCreateCube = sCreateCube & ")"


sInsertInto = "INSERTINTO=INSERT INTO [Helios001]("

sInsertInto = sInsertInto & "[Fakturatyp].[Typ Faktury],[Fakturatyp].[Druh
pohybu]"

sInsertInto = sInsertInto & ", [Organizace].[Nazev], [Organizace].[Cislo
Org]"

sInsertInto = sInsertInto & ", [Sortiment].[Nazev Sortiment],[Sortiment].[Reg
Cis]"

sInsertInto = sInsertInto & ", [Datum].[Rok], [Datum].[Kvartal],
[Datum].[Mesic]"

sInsertInto = sInsertInto & ", [Datum].[Den]"

sInsertInto = sInsertInto & ", Mnozstvi,[Cena bez DPH],[Cena s DPH]"

sInsertInto = sInsertInto & ")"


sql = " select ZRTabDruhPohybu.Nazev,ZRTabDruhPohybu.DruhPohybu"
```

```
sql = sql & " ,TabCisOrg.Nazev,TabCisOrg.CisloOrg"
sql = sql & ",TabKmenZbozi.Nazev1,TabKmenZbozi.RegCis"
sql = sql & " ,Year(ZRProdej.DUZP), DATENAME(quarter,ZRProdej.DUZP)"
sql = sql & " ,Month(ZRProdej.DUZP),DAY(ZRProdej.DUZP)"
sql = sql & " ,ZRProdej.MnozstviDruhove, ZRProdej.CCbezDaniKc"
sql = sql & " ,ZRProdej.CCsSDKcPoSDruhova"
sql = sql & " from ZRProdej ,ZRTabDruhPohybu, TabCisOrg,TabKmenZbozi"
sql = sql & ", TabSkupinyZbozi"
sql = sql & " where
ZRProdej.DruhPohybuZbo=ZRTabDruhPohybu.DruhPohybu"
sql = sql & " and ZRProdej.CisloOrg=TabCisOrg.CisloOrg"
sql = sql & " and ZRProdej.RegCis=TabKmenZbozi.RegCis"
set Con = wscript.CreateObject ("ADODB.Connection")
s="PROVIDER=MSOLAP;DATA SOURCE=kostka.cub;"
s=s & sSourceDSN & ";" & sCreateCube & ";" & sInsertInto & sql & ";"
on error resume next
con.Open s
if err<>0 then
   err_popis=err.description
   wscript.echo err.number & err.description
else
  con.Close
end if
```

on error goto 0

set Con = Nothing

Příloha D. Ukázka zobrazení OLAP přes MDX

```
<%@ Language=VBScript %>

<html><head></head><body>

<%Dim cat,cst,i,j,sSelect,csw,iDC0,iDC1,iPC0,iPC1

set cat=server.createobject("ADOMD.Catalog")

set cst=server.createobject("ADOMD.CellSet")

OLAPServerName="http://POSEIDONW2K3/"

cat.ActiveConnection="Provider=MSOLAP.3;Data Source="&
OLAPSERVERNAME & ";Initial Catalog=Helios;User
Id=sa;Password=sa159;"

sSelect = "SELECT"

sSelect = sSelect & "{[Measures].[Mnozstvi],[Measures].[Cena bez
DPH],[Measures].[Cena s DPH]} ON Columns,"

sSelect = sSelect & "FILTER ({[Tab Cis
Org].[Nazev].[Nazev]},[Measures].[Mnozstvi]>0) ON ROWS"

sSelect = sSelect & " FROM Helios001"

cst.Source = sSelect

set cst.ActiveConnection = cat.ActiveConnection

cst.Open

iDC0=cst.Axes(0).DimensionCount-1

iDC1=cst.Axes(1).DimensionCount-1

iPC0=cst.Axes(0).Positions.Count-1

iPC1=cst.Axes(1).Positions.Count-1
```

```
response.write "<table border=1>"
for h=0 to iDC0
  response.write "<tr>"
  for c=0 to iDC1
    response.write "</td><td>"
  next
  for i=0 to iPC0
    response.write "<th>"
    response.write cst.Axes(0).Positions(i).Members(h).Caption
    response.write "</th>"
  next
  response.write "</tr>"
next
for j=0 to iPC1
  response.write "<tr>"
  for h=0 to iDC1
    response.write "<td><b>"
    response.write cst.Axes(1).Positions(j).Members(h).Caption
    response.write "</b></td>"
  next
  for k=0 to iPC0
    response.write "<td align=right>"
```

```
    response.write cst(k,j).FormattedValue
    response.write "</td>"
  next
  response.write "</tr>"
next
response.write "</table>"
%>
</body>
</html>
```