Microsoft® SQL Server™ 2008

Business Intelligence

Ľuboslav Lacko

2. aktualizované vydanie (SQL Server 2008 – finálna verzia)

Microsoft®

Microsoft SQL Server 2008

Business Intelligence na platforme Microsoft SQL Server 2008

Obsah:

Pre praktické zoznámenie sa s črtami Business Intelligence na platforme nového databázového servera MS SQL Server 2008 odporúčame ako prvý krok jeho inštaláciu. Popis je v prílohe č. 1. Ako veľmi zaujímavú alternatívu pre testovanie a zoznamovanie sa so serverom SQL Server 2008 odporúčame nainštalovať ho na virtuálny počítač vytvorený pomocou nástroja Microsoft Virtual PC 2007. Ako operačný systém odporúčame pre vývojárov a študentov Windows Vista, a pre migrujúcich administrátorov Windows Server 2008.

Kapitola 1: Microsoft SQL Server 2008 ako platforma pre BI

Termín Business Intelligence (BI) prvýkrát v roku 1989 definoval Howard Dresner zo spoločnosti Gartner Group ako „množinu konceptov a metodík, ktoré zlepšujú rozhodovací proces za použitia metrík alebo systémov založených na metrikách." Je to proces transformácie údajov na informácie a prevod týchto informácii na poznatky prostredníctvom objavovania. Účelom procesu je konvertovať veľké objemy údajov na poznatky, ktoré sú potrebné pre koncových používateľov. Tieto poznatky môžeme potom efektívne využiť napríklad v procese rozhodovania. S problematikou BI sú úzko spojené údajové sklady. **Údajový sklad** je podnikovo štruktúrovaný depozitár subjektovo orientovaných, integrovaných, časovo premenných, historických údajov použitých na získavanie informácií a podporu rozhodovania. V údajovom sklade sú uložené atomické a sumárne údaje. Údaje sa získavajú a ukladajú do produkčných (operačných) databáz, ktoré môžu byť v rôznych oddeleniach firiem, alebo dokonca v rozličných geografických lokalitách.

Tieto údaje v pravidelných intervaloch zozbierame, predspracujeme a zavedieme do údajového skladu. Údajový sklad je v podstate tiež databáza, len je organizovaná podľa trochu iných pravidiel, tabuľky napríklad nemusia byť normalizované.

Údaje sa do údajového skladu zapisujú skôr podľa predmetu záujmu, než podľa aplikácie, v ktorej boli vytvorené. Pri orientácii na subjekt sú údaje v údajovom sklade kategorizované podľa subjektu, ktorým môže byť napr. zákazník, dodávateľ, zamestnanec, výrobok a podobne.
Údajový sklad musí byť jednotný a integrovaný. To znamená, že údaje týkajúce sa konkrétneho predmetu sa do údajového skladu ukladajú len raz. Preto musíme zaviesť jednotnú terminológiu, jednotné a konzistentné jednotky veličín. Nie je to úloha jednoduchá, pretože údaje prichádzajú do údajového skladu z nekonzistentného a neintegrovaného operačného prostredia. Preto musia byť údaje v etape prípravy a zavedenia upravené, vyčistené a zjednotené. Ak údaje nie sú konzistentné a dôveryhodné, údajový sklad stráca význam. Údaje sa ukladajú do údajového skladu ako séria snímok, z ktorých každá reprezentuje určitý časový úsek. Na rozdiel od operačného prostredia, kde sú údaje platné v okamihu prístupu, v údajových skladoch sú údaje platné pre určitý časový moment, časový snímok. Zatiaľ čo v operačnom databázovom prostredí sa údaje ukladajú za kratšie časové obdobie dní, maximálne mesiacov, v údajovom sklade sú údaje za dlhšie časové obdobie, typicky niekoľko rokov. Kľúčové atribúty v údajovom sklade obsahujú čas, ktorý v operačných databázach nemusí byť uvádzaný. Len čo je v údajovom sklade zaznamenaná konkrétna snímka údajov z operatívnej databázy, nemôžu byť už tieto údaje v údajovom sklade modifikované.

SQL 2008 borí mýtus o nevhodnosti transakčných databáz pre analýzy

Transakčné OLTP databázy sú určené na ukladanie operačných údajov. Výsledkom dopytovania sú databázové tabuľky, súhrny získané pomocou agregačných funkcií, rôzne zostavy a podobne. OLTP databázy sú z dôvodu jednoduchého dopytovania a vylúčenia redundancie spravidla normalizované. Takéto systémy dosahujú vysoké výkony skôr pri online transakciách než pri zložitých analýzach, ktoré sú veľmi náročné na výpočtovú kapacitu. Komplexná analýza vyžaduje iné techniky návrhu databáz, napríklad použitie multidimenzionálnych a hviezdicových schém s tabuľkami faktov, ktoré obsahujú merné jednotky obchodovania a vysoko nenormalizované tabuľky dimenzií. Azda najväčšou prekážkou použitia databázových systémov OLTP pre analýzy je skutočnosť, že tieto systémy nemajú k dispozícii integrovaný zdroj údajov zo všetkých operačných systémov v rámci podniku tak, aby umožnili tvorbu komplexných analýz, to znamená, že potrebné údaje, alebo údaje ktoré by mali slúžiť ako podklady pre analýzy sú roztrúsené v rôznych spravidla heterogénnych OLTP systémoch a musia sa zakaždým prácne integrovať skôr, ako je možné získať požadované informácie. Časová náročnosť prípadných analýz, hoci nemusí ísť ani o príliš zložité ani príliš komplexné analýzy je preto pomerne vysoká. Niekedy sa dokonca ani nepodarí koordinovať údaje medzi jednotlivými systémami, takže vlastne ani nemôžeme získať globálny obraz o stave podnikania.

Nevýhody použitia transakčných databáz pre analytické účely by sme mohli zhrnúť do niekoľkých bodov:
- nie je jednoduché nájsť príčiny a vysvetlenia problémov,
- zložité hľadanie závislosti jednotlivých veličín,
- príliš rozsiahle výstupy z transakčných systémov,
- dlhý čas výpočtu a degradácia výpočtového výkonu databázového stroja transakčnej databázy,
- transakčný systém neuchováva historické údaje,

– nehomogénna štruktúra údajov,
– dlhý čas prípravy údajov.

Výhody
Na druhej strane je snaha o komplexný prístup k údajom v údajovom sklade pri akceptovaní hlavnej požiadavky, ktorú prináša dnešná hektická globalizovaná ekonomika – „všetko v reálnom čase". Kým v minulosti sa údaje z údajových skladov spracovávali v dávkach pre nejaké obdobie z blízkej minulosti (včerajší deň, minulý týždeň), v súčasnosti manažéri a analytici požadujú prístup k výsledkom analýz v reálnom čase. V nadväznosti na túto požiadavku už začína byť minulosťou aj oddelenie transakčných systémov a údajových skladov. Súčasným trendom vyplývajúcim z citovaných požiadaviek je viacúrovňový podnikový údajový sklad s možnosťou prístupu pomocou integrovaného aplikačného prostredia. Vznikne tak akýsi komplexný BI ekosystém, do ktorého sú na strane vstupov spravidla zahrnuté aj externé zdroje údajov a ktorý na výstupe poskytuje informácie nielen pre firmu, ktorá je vlastníkom, budovateľom a prevádzkovateľom BI ekosystému, ale aj pre jej partnerov, dodávateľov a zákazníkov.

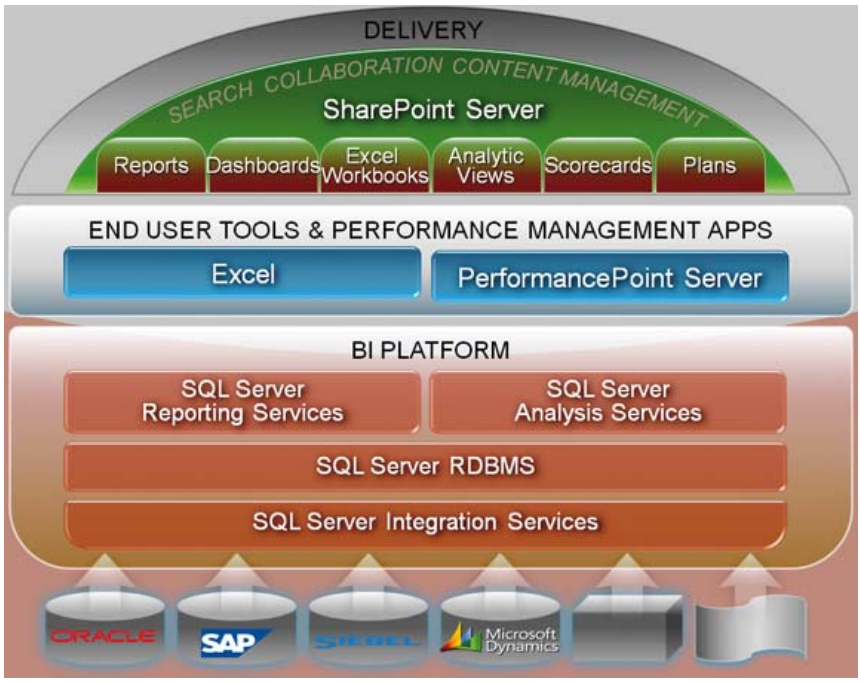
BI systémy pracujúce v reálnom čase

BI systémy pracujúce v reálnom čase sú založené na troch primárnych faktoroch:
– paralelný priebeh etapy ETL, teda extrakcie, transformácie a zavedenia údajov, pričom všetky ETL procesy musia byť dostatočne nadimenzované na predpokladané objemy spracovávaných údajov a to aj v dobe špičiek, ako sú napríklad predvianočný predaj, koncoročné uzávierky alebo marketingové kampane,
– relačné systémy musia okrem operačnej prevádzky, teda spracovávania transakcií zvládnuť aj zložité a kapacitne náročné BI dopyty,
– doručovanie výsledkov BI procesov, teda reportov a výsledkov analýz a to buď periodicky, alebo na vyžiadanie.

Charakteristickou črtou údajových skladov je veľké množstvo údajov v niektorých databázových tabuľkách. V takýchto prípadoch systémov reálneho času je výhodné rozdeliť veľkú tabuľku na niekoľko logických oddielov (partícií), podľa nejakého pravidla vyplývajúceho z povahy údajov, napríklad jeden logický oddiel môže obsahovať údaje za nejaké časové obdobie (mesiac, rok), prípadne údaje podľa iného pravidla, napríklad v zozname osôb, osoby, ktorých priezvisko začína príslušným písmenom a podobne. Takéto delenie je potom logické aj z hľadiska následného spracovania údajov, pretože trebárs reklamácie a fakturácie sa obvykle vykonávajú za príslušné kalendárne obdobie.

Pri analýze možností integrácie jednotlivých podnikových systémov a procesov, či už ekonomických, alebo výrobných, nachádzame ako hlavný integračný prvok databázu, kam všetky integrované a konsolidované systémy ukladajú svoje údaje. Firma Microsoft v tejto oblasti profituje z tradície jedného z najvýznamnejších dodávateľov databázových systémov. Ich nový databázový produkt SQL Server 2008 je moderná komplexná serverová platforma pre ukladanie a správu údajov v databázach a údajových skladoch a balíky nástrojov pre Business Intelligence vrátane pokročilého reportovania. IT Ekosystém podnikových aplikácií využívajúci SQL Server 2008 by mal byť koncipovaný tak, že na najvyššej úrovni architektúry je SharePoint Server pre spoluprácu, a vrstva Content management sa využíva na vyhľadávanie a správu obsahu.

SQL Server 2008 je vyšším evolučným stupňom databázových serverov. Umožňuje prelínanie a integráciu podnikových procesov aj na úrovni analytických služieb. Analytické služby implementované priamo v databázovom serveri sa tak môžu využívať buď izolovane v jednotlivých odvetviach alebo komplexne. Nemusí to byť len podpora rozhodovania na strategickej úrovni týkajúca sa top manažmentu. Na jednej strane analytických procesov sú podklady pre analýzy uložené v databázach alebo údajových skladoch. Rovnako dôležitý je na druhej strane aj spôsob prezentovania výsledkov analýz. Pracovníci pred špecializovanými aplikáciami preferujú používanie nástrojov na ktoré sú zvyknutí zo svojej bežnej práce, napríklad programy kancelárskych balíkov.



Business Intelligence na platforme Microsoft

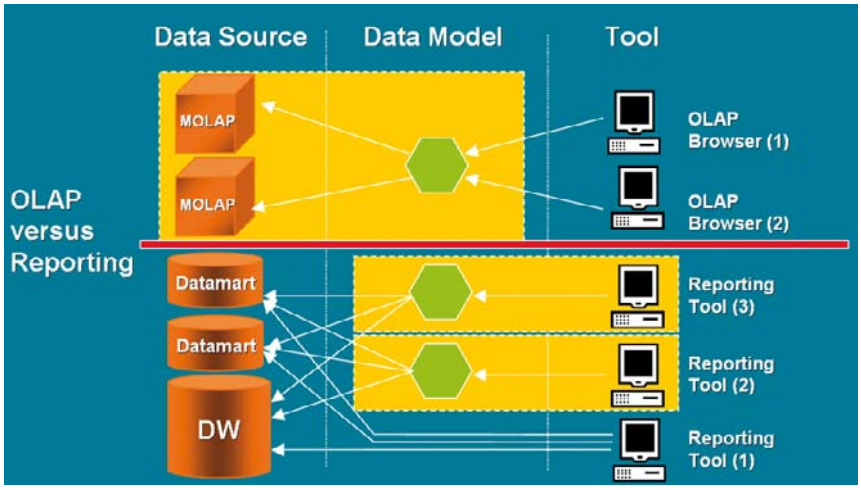
Kapitola 2: Modelovanie Business Intelligence projektov

Jedným z preferovaných trendov pre budovanie podnikových projektov a aplikácií je model driven development, kedy sa budovanie projektu začne fázou modelovania. Túto filozofiu podporuje aj SQL Server 2008, ktorý pre modelovanie projektov typu Business Intelligence využíva technológiu UDM (Unified Dimension Model). Táto technológia bola po prvý krát použitá vo verzii SQL Server 2005. Ťažisko vytvorenia projektu spočíva v modelovaní a následne na základe vytvoreného modelu BI aplikácie bude vygenerovaná štruktúra pre údajový sklad, vytvorené dávky pre jeho plnenie prostredníctvom možností Integračných služieb. Budovanie BI projektu pokračuje návrhom mierok, dimenzií, kociek a podobne.

Prvý predpoklad pre unifikovaný prístup k modelovaniu je unifikácia technológií. Tento predpoklad je splnený integráciou databázových analytických a reportovacích služieb do jedného balíka. Druhým predpokladom je unifikované používateľské rozhranie, čo v prípade databázového servera predstavujú nástroje pre administráciu, dopytovanie a návrh štruktúr a čiastkových modelov. Aj táto požiadavka je v prípade SQL Servera 2008 splnená. K dispozícii je komfortný nástroj **SQL Server Business Intelligence Development Studio**.

Základné princípy UDM

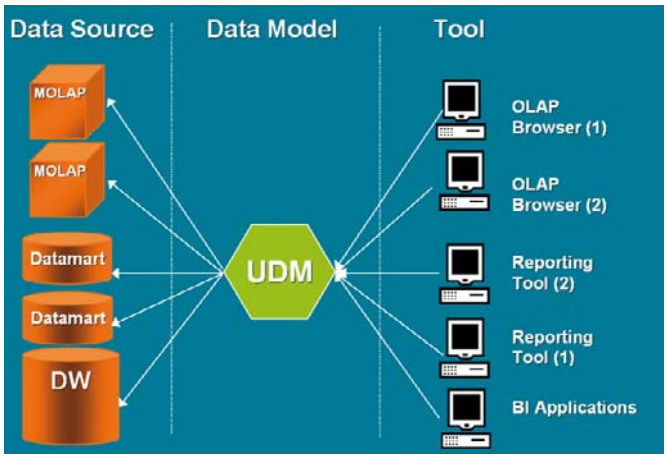
Do verzie servera SQL Server 2005 boli vrstvy databázových, analytických a reportovacích služieb pomerne striktne oddelené, aj keď aj tu by sme už našli množstvo zjednocujúcich prvkov – je možné analyzovať údaje z relačných databáz, prípadne generovať reporty z relačných aj analytických databáz.



Oddelenie vrstvy OLAP a reportovacích služieb v serveri SQL Server 2000

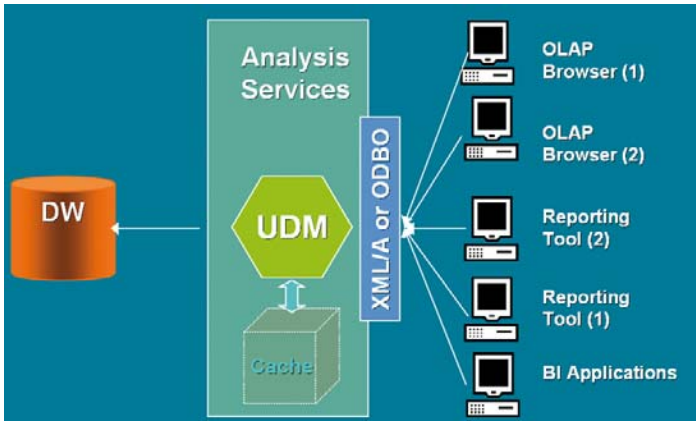
Hlavná nevýhoda architektúry predchádzajúcej verzie – redundancia údajov a modelov je zrejmá už na prvý pohľad. Na obrázku v ľavej časti vidíme už na prvý pohľad, že dochádza k redundancii údajov, pretože tie isté údaje sú jednak v relačných a jednak v multidimenzionálnych databázach. Pri hlbšej analýze údajových tokov zistíme, že situácia z hľadiska redundancie je ešte komplikovanejšia. Tie isté údaje sa prenášajú z relačných databáz do údajových skladov, odtiaľ v mnohých prípadoch do údajových tržníc a odtiaľ do multidimenzionálnych OLAP kociek. Taktiež nezanedbateľná redundancia a s ňou spojené množstvo práce vysokokvalifikovaných špecialistov je vo vrstve modelov. Iným problémom takejto oddelenej architektúry sú rozdielne návyky pracovníkov čo sa týka práce s údajmi.

Modelovanie s využitím UDM umožňuje využitie výhod relačných aj multidimenzionálnych databáz a do istej miery eliminuje ich nevýhody. Od verzie SQL Servera 2005 sú vrstvy Business Intelligence zjednotené do jednotného modelu Unified Dimensional Model (UDM). Tento model prevzal to najlepšie z reportovania a OLAP analýz. K zjednoteniu dochádza jednak na úrovni modelov, kedy je potrebný len jeden dimenzionálny model pre generovanie reportov aj OLAP kociek.



Unified Dimensional Model – zjednotenie na úrovni modelov

Klasický postup analýzy začínal výberom tabuliek z relačných databáz alebo z údajového skladu. Tieto tabuľky slúžili na vytvorenie faktov a dimenzií. Podobne je to aj pri UDM, ale údaje zostávajú v pôvodných úložiskách Mechanizmus MOLAP (multidimenzionálne online analytické spracovanie) potom uloží analytické údaje vo vlastných údajových štruktúrach a sumároch. Počas tohto procesu sa napočíta toľko predbežných výsledkov, koľko je z technického a časového hľadiska možné. Údaje v úložisku typu MOLAP sa teda budú ukladať ako vopred vypočítané pole. Hodnoty údajov aj indexov sa uchovávajú v jednotlivých poliach multidimenzionálnej databázy. Databáza je organizovaná tak, aby umožnila rýchle získavanie príslušných údajov z viacerých dimenzií. Preto je hlavnou výhodou MOLAP maximálny výkon vzhľadom na dopyty používateľa, nevýhodou je redundancia údajov, nakoľko tieto sú uložené jednak v relačnej databáze, jednak v multidimenzionálnej databáze. Požiadavky na úložnú kapacitu môžu v prípade použitia viacerých dimenzií extrémne narastať. Východiskom z tejto situácie pri serveri SQL Server 2005 je vyrovnávacia MOLAP cache pamäť, kam sa ukladajú najčastejšie používané alebo očakávané výsledky agregácií. Navyše toto proaktívne cachovanie je pri serveri SQL Server 2005 plne automatizované. Údaje teda zostávajú v relačných databázach a napočítané agregácie sa ukladajú do multidimenzionálnych štruktúr. Pri dopytovaní sa údaje vyberajú do multidimenzionálnej pamäte cache.



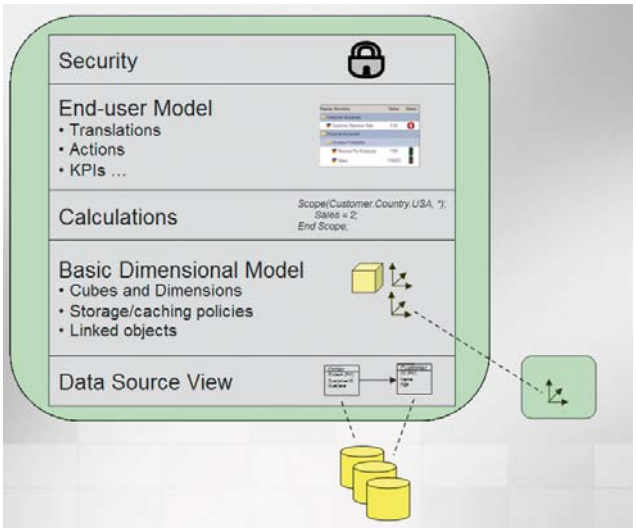
Unified Dimensional Model – cache

Významnou novinkou je zavedenie novej skupiny služieb s názvom Integration Services. Na úrovni tejto vrstvy sa budú integrovať údaje z rôznych údajových zdrojov vrátane ich požadovaných transformácií.

Ak by sme mali zhrnúť dosiaľ prezentované fakty do jednoduchej definície UDM, dospejeme k záveru, že UDM je akýmsi pomysleným mostom medzi používateľom a jeho údajmi.

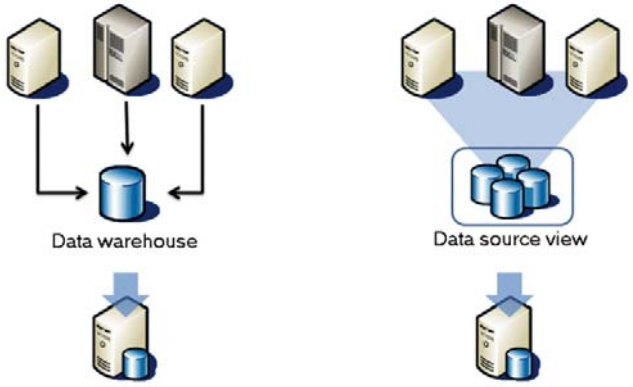
Implementácia UDM na platforme SQL Server 2008

Azda najlepšie pochopíme filozofiu modelovania na hierarchickej schéme. Celá vrstva slúžiaca na vytvorenie výstupu (reportu), je z architektonického hľadiska postavená nad údajmi v databázach. Samozrejme je výhodné, aby tieto údaje boli čo najviac vyčistené, prípadne do rozumnej miery normalizované bez zbytočných redundancií a neprehľadných komplikovaných relačných vzťahov. Samozrejme, nie každý deň sú Vianoce a tak často pri budovaní BI vrstvy musíme vychádzať z údajov aké sú k dispozícii, prípadne aké vyhovujú logike predmetu podnikania, alebo k akej štruktúre údajov sme sa dopracovali postupným budovaním informačných systémov.



Unified Dimensional Model

Samotná BI nadstavba nad databázou je rozdelená do štyroch vrstiev. Nad databázami je vrstva údajových pohľadov, v ktorej už presne vyšpecifikujeme tabuľky, pohľady a to až na úroveň jednotlivých atribútov a relačných väzieb. Modelovaním vrstvy **Data Source View** do určitej miery prispejeme síce nie k fyzickému vyčisteniu údajov, no k nadefinovaniu priamejšieho prístupu k nim. Zjednodušene povedané – na tejto úrovni vytvárame relačné schémy, na základe ktorých budú vytvárané dimenzie a kocky. Veľmi dôležitou úlohou úrovne Data Source View je zjednotenie údajov z heterogénnych zdrojov.



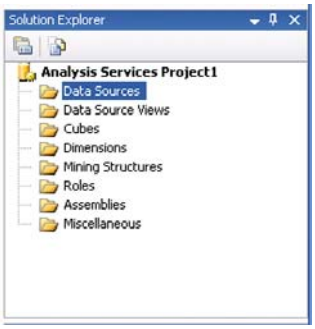
Data source view abstrahuje údaje pre analýzy od údajových zdrojov pri analýzach

Nasleduje úroveň dimenzií kde špecifikujeme, ktoré atribúty sa viažu k merným jednotkám obchodovania. Na úrovni **základného dimenzionálneho modelu** do značnej miery ovplyvňujeme aj spôsob ukladania a cachovania multidimenzionálnych údajov. Okrem faktov a dimenzii vstupujú do hry aj **kalkulácie**, to znamená údaje vypočítané podľa jednoduchých, prípadne aj zložitých vzťahov z hodnôt iných atribútov. Napríklad, ak by sme mali v tabuľke ako atribút rodné číslo, dokážeme z neho určiť dátum narodenia a teda aj vek príslušnej osoby.

Vrstva **End-user Model** definuje manipulácie s údajmi a výsledkami súhrnov na najvyššej úrovni. Je potrebné si uvedomiť, že výsledky analýz a agregované údaje obsahujú oveľa citlivejšie informácie, než surové údaje. Preto na úrovni výsledkov analýz je oveľa dôležitejšie zabezpečenie cieleného prístupu k informáciám.

Vytváraniu analytických projektov je prispôsobená aj koncepcia Business Intelligence projektov vo vo vývojovom prostredí. Unifikovaný prístup k BI úlohám podľa tejto viacvrstvovej schémy je naznačený aj priečinkami v okne vývojového prostredia Solution Explorer v pravom hornom rohu:

- Data Sources
- Data Source Views
- Cubes
- Dimension
- Mining Models
- Roles
- Assemblies
- Miscellanous



SQL Server Business Intelligence Development Studio – okno Solution Explorer

Pri vytváraní modelu vlastne s väčšou alebo menšou pomocou interaktívnych sprievodcov budujeme v záložkách jednotlivé hierarchické vrstvy modelu.

Kapitola 3: Integračné služby

Fáza zavádzania údajov je jednou z najdôležitejších etáp budovania analytických projektov a údajového skladu. SQL Server 2008 má pre túto fázu implementované integračné služby. Môžeme ich využiť pri inštalácii databázovej časti informačných systémov, migrácil na inú platformu alebo pri zavádzaní údajov do údajového skladu. Údaje, ktoré chceme integrovať sú umiestnené spravidla v rôznych nehomogénnych operačných prostrediach, hardvérových platformách (PC, mainframe, iMac...), operačných systémoch (Windows, Unix, Linux, Sun, Solaris...), databázových systémoch (SQL Server, Oracle, Informix, IBM DB2...) archívnych systémoch, podnikových systémoch (SAP, PeopleSoft, Baan...) a čo je ešte horšie, môžu sa vyskytovať v rozličných formátoch.

Hlavným rozdielom medzi importom údajov do informačných systémov a zavádzaním údajom do údajového skladu spočíva v tom, že import je záležitosťou jednorazovou, ale zavádzanie údajov do údajového skladu sa odohráva periodicky v určitých, napríklad 24-hodinových intervaloch. Aj pri týchto dvoch scenároch je začiatok veľmi podobný, import údajov do informačného systému a prvotné naplnenie údajového skladu.

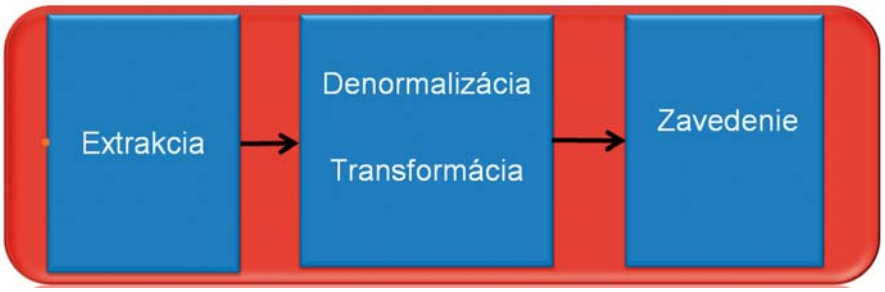
Okrem interných údajov z nášho podnikateľského prostredia je pomerne často potrebné pracovať aj s externými údajmi. Tieto údaje môžeme získať napríklad ich nákupom, analýzou konkurenčného prostredia, môžeme využiť aj údaje voľne prístupné na Internete. Nevýhodou vyplývajúcou so samotnej povahy externých údajov je, že nemôžeme periodicky odoberať vzorky, ako je tomu pri interných údajch. Zdroje externých údajov preto vyžadujú nepretržité monitorovanie za účelom určenia, kedy sú dostupné.

Kapitolou samou o sebe sú na jednej strane chýbajúce údaje a na strane druhej záznamy duplicitné. Duplicita údajov je menším problémom, pretože duplicitné záznamy je možné odstrániť, no v niektorých prípadoch to môže byť pomerne časovo náročné. Väčším problémom sú chýbajúce údaje. Pri malom objeme ich môžeme v niektorých prípadoch ignorovať, alebo doplniť ručne z iných zdrojov. Problémy môžu nastať aj ak sú údaje uložené v rôznych druhoch formátov. Týka sa to rodných čísel, PSČ, čísiel účtov a podobne. Okrem hodnôt samotných sú v údajoch skryté aj rôzne vzťahy, napríklad master – detail, organizačná štruktúra firmy, hierarchická štruktúra zamestnancov a podobne. Údaje sú vo väčšine prípadov dynamické, a v niektorých prípadoch sa neuváženým vymazaním údajov naruší integrita údajov.

Extrakcia – transformácia – zavedenie

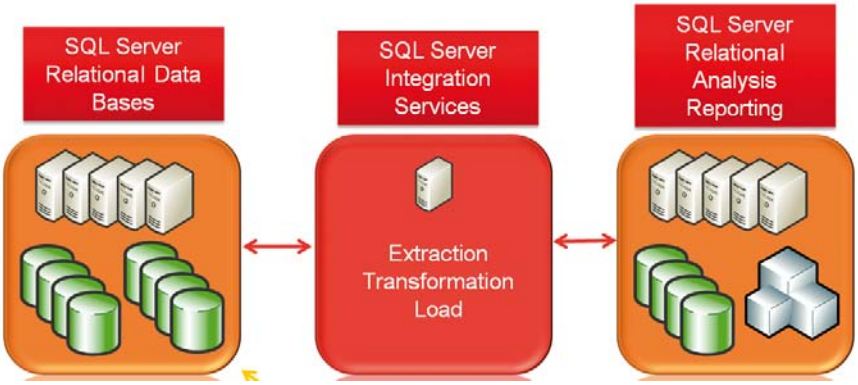
Podľa teórie budovania údajových skladov sa tento proces zvykne označovať ETL(Extrakcia, transformácia, Loading) alebo ETT(Extrakcia, Transformácia, Transfer) Hlavnou úlohou tohto procesu je naplnenie údajového skladu určenými údajmi v požadovanom čase.

Jednotlivé etapy procesu ETL sú:
- **Extrakcia** – výber údajov prostredníctvom rôznych metód,
- **Transformácia** – overenie, čistenie, integrovanie a časové označenie dát,
- **Loading (zavedenie)** – premiestnenie údajov do údajového skladu.



Klasická schéma zavádzania údajov z relačných databáz do údajového skladu

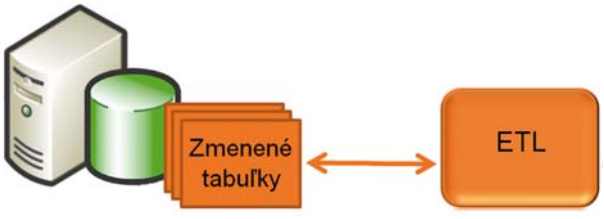
Pojmy ETT alebo ETL popisujú rad procesov, ktorých úlohou je extrakcia údajov zo zdrojových systémov, ich transformácia a vyčistenie a prenos údajov do údajového skladu. Údaje sa teda v tejto etape nielen prenášajú, ale aj spracovávajú, napríklad indexujú, sumarizujú, zisťujú sa prípadné zmeny štruktúr zdrojových údajov potrebných pre údajový sklad, podľa potreby sa menia štruktúry kľúčov a udržiavajú sa metaúdaje, teda údaje o dátach, v tomto prípade predpisov a definícií pre prenos a spravovanie údajov. Počiatočným naplnením údajového skladu údajmi z operačných databáz sa úloha ETL samozrejme nekončí, údajový sklad sa predsa v pravidelných intervaloch napĺňa aktualizovanými dátami. Proces je komplexný a vo väčšine prípadov časovo pomerne náročný. Pri niektorých implementáciách môže zabrať aj viac ako polovicu celkového času, úsilia a teda aj veľkú časť nákladov potrebných na vytvorenie informačného systému alebo údajového skladu. Hlavným cieľom etapy ETL je centralizácia údajov, t. j. ich zhromaždenie z mnohých spravidla nehomogénnych a rôznorodých zdrojov z OLTP databáz a naplnenie údajového skladu určenými údajmi v požadovanom čase. Je dôležité si uvedomiť, že na úrovni ETL sa pracuje s údajmi, z ktorých sa neskôr stanú informácie. Preto by tieto údaje mali byť vysoko kvalitné, presné, predmetné a aktuálne a teda užitočné pre používateľov. Okrem kvality údajov je samozrejme dôležitá aj ich dostupnosť, aby mohli jednotliví používatelia údajového skladu, napríklad manažéri, obchodníci a analytici používať údajový sklad účinne a efektívne.



Integračné služby ako most medzi relačným a analytickým prostredím

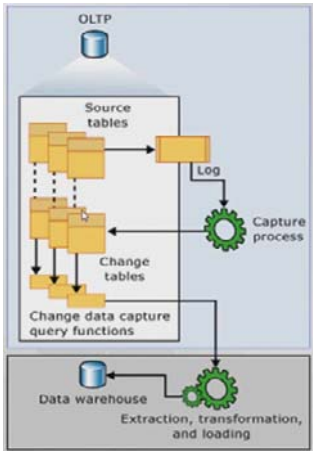
Zachytávanie zmien pomocou CDC (Change data capture)

Pri periodickom zavádzaní údajov zo zdrojových do cieľových databázových tabuliek je najvýhodnejšie a najefektívnejšie zavádzať len tie údaje, ktoré boli zmenené. Pre zachytávanie zmien môžeme využiť pomocné atribúty, napríklad dátum a čas poslednej zmeny a potom pri prenose údajov prenášame len údaje zmenené od posledného prenosu. Ak údaje nepridávame, len meníme, stačí nám jeden bit ako príznak, či boli záznamy zmenené. Po prenose zmenených záznamov tento príznak vymažeme a nastavíme ho až pri editovaní údajov. Spoločným menovateľom týchto metód je prácnosť – musíme si to urobiť sami. Preto bola do servera SQL Server 2008 pridaná nová funkcionalita pre zachytávanie zmien údajov v databázových tabuľkách. CDC nám umožňuje podchytiť zmeny v databázovej tabuľke, napríklad pridanie, vymazanie, alebo zmeny údajov v zdrojových tabuľkách.



Zachytávanie zmien v databázových tabuľkách OLTP systémov pre účely ETL

Keď sa zmenia údaje v zdrojovej tabuľke, tieto údaje sú zapísané do transakčných logov. Proces monitorovania zmien pracuje s týmto transakčným logom a zapisuje zmenené údaje do tabuliek zmien.



Princíp zachytávania zmien v databázových tabuľkách

Najlepšie bude predstaviť túto novinku na praktickom príklade. Pre fungovanie CDC je potrebné spustiť službu SQL Server Agent. V implicitnej inštalácii je totiž zastavená s príznakom, že sa spúšťa ručne. Môžeme ju spustiť napríklad prostredníctvom nástroja Sql Server Configuration Manager

Vytvorme novú databázu. Môžeme to urobiť cez kontextové menu v nástroji SQL Server Management Studio alebo SQL príkazom:

```
CREATE DATABASE CDCDatabaza
```

Režim zachytávania zmien v aktuálnej databáze aktivujeme spustením uloženej procedúry

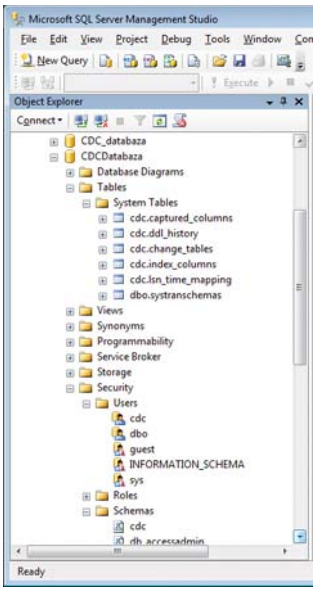
```
EXEC sp_cdc_enable_db
```

Pre spustenie tejto uloženej procedúry musíme mať privilégiá administrátora. Preto sa musíme prihlásiť s parametrami používateľa, ktorého sme v procese inštalácie definovali ako administrátora. Ak chceme zistiť, či je pre existujúce databázy táto črta povolená, môžeme vytvoriť dopyt do katalógového pohľadu sys.databases, kde nás zaujíma názov databázy, jej ID a príznak is_cdc_enabled. Ak je tento príznak nastavený na hodnotu 1, zachytávanie zmien je povolené.

```
select name, database_id, is_cdc_enabled from sys.databases

name                    database_id is_cdc_enabled
----------------------- ----------- --------------
master                  1           0
tempdb                  2           0
model                   3           0
msdb                    4           0
ReportServer            5           0
ReportServerTempDB      6           0
AdventureWorksLT2008    7           0
AdventureWorks2008      8           0
Pokusy                  9           0
Test                    10          0
FileStreamDB            12          0
CDCDatabaza             13          1
```

V nástroji SQL Server Management Studio sa môžeme presvedčiť, že zavolaním uloženej procedúry pre aktiváciu CDC pribudol do zoznamu používateľov v priečinku „Security" nový používateľ CDC a v záložke „Schemas" rovnomenná schéma. V zozname systémových tabuliek pribudli nové tabuľky pre zachytávanie zmien



V zozname systémových tabuliek pribudli nové tabuľky pre zachytávanie zmien a v zozname používateľov používateľ „cdc"

Pre pokusy zo zachytávaním zmien vytvoríme jednoduchú tabuľku zamestnancov. Ak chceme nastaviť parameter `supports_net_changes` na hodnotu 1 je potrebné, aby tabuľka mala definovaný primárny kľúč

```
CREATE TABLE zamestnanci
(
   ID int PRIMARY KEY,
   meno nvarchar(30),
   email nvarchar(30)
)
```

CDC je potrebné povoliť a nastaviť separátne pre každú tabuľku pomocou uloženej procedúry

```
exec sys.sp_cdc_enable_table
@source_schema = N'dbo',
@source_name = N'zamestnanci',
@role_name = N'cdc_Admin',
@capture_instance = N'dbo_zamestnanci',
@supports_net_changes =1;
```

Po spustení tejto uloženej procedúry sa vytvoria dve úlohy pre SQL Server Agent. Jedna pre zachytávanie zmien z transakčného logu a pre ich synchrónny zápis do asociovaných tabuliek určených pre zachytávanie zmien a druhá pre čistenie:

```
Job 'cdc.CDCDatabaza_capture' started successfully.
Job 'cdc.CDCDatabaza_cleanup' started successfully.
```

Ak chceme zistiť, pre ktorú tabuľku je povolené zachytávanie zmien, môžeme vytvoriť dotaz do katalógového pohľadu sys.tables, kde nás zaujíma názov tabuľky a príznak is_tracked_by_cdc. Ak je tento príznak nastavený na hodnotu 1, zachytávanie zmien je povolené.

```
select name, is_tracked_by_cdc from sys.tables

name                 is_tracked_by_cdc
-------------------- -----------------
ddl_history          0
lsn_time_mapping     0
captured_columns     0
index_columns        0
zamestnanci          1
dbo_zamestnanci_CT   0
systranschemas       0
change_tables        0
```

Do tabuľky pridáme niekoľko záznamov:

```
INSERT INTO dbo.zamestnanci VALUES (1, 'Mickey Mouse','mickey@disney.com');
INSERT INTO dbo.zamestnanci VALUES (2, 'Donald Duck','donald@disney.com');
INSERT INTO dbo.zamestnanci VALUES (3, 'Snehurka','snehurka@disney.com');
```

Vypíšeme obsah tabuľky pre zachytávanie zmien:

```
SELECT * FROM cdc.dbo_zamestnanci_CT;
```

Výpis je z dôvodu tlače rozdelený na dve časti

```
__$start_lsn           __$end_lsn    __$seqval
---------------------- ------------- ----------------------
0x0000001C000000480013 NULL          0x0000001C000000480012
0x0000001C0000004C0003 NULL          0x0000001C0000004C0002
0x0000001C0000004D0003 NULL          0x0000001C0000004D0002


__$operation __$update_mask   ID    meno            email
------------ ---------------- ---- --------------- --------------------
2            0x07             1    Mickey Mouse    mickey@disney.com
2            0x07             2    Donald Duck     donald@disney.com
2            0x07             3    Snehurka        snehurka@disney.com
```

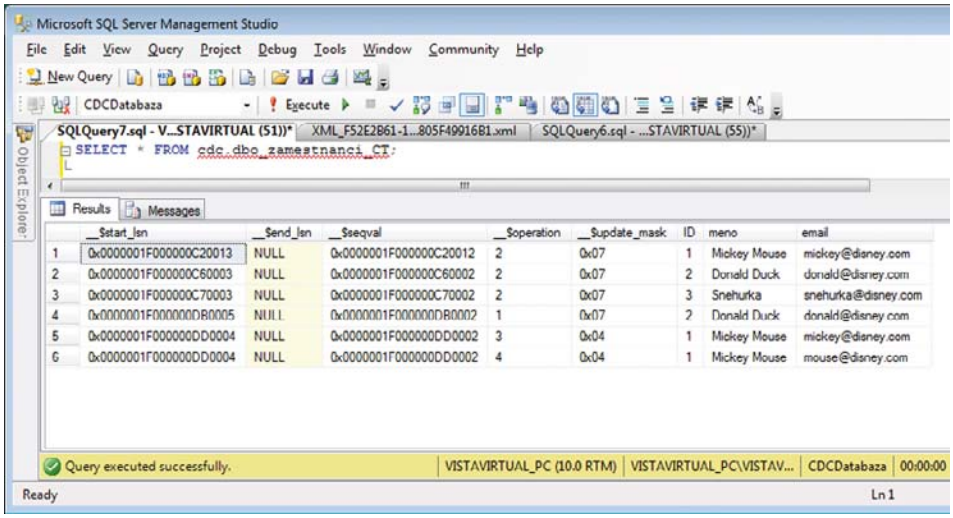
Atribúty `__$start_lsn` a `_$end_lsn` udávajú polohu zachytenej zmeny v redo-log súbore. Atribút `__$operation` udáva typ operácie, ktorá bola nad zachyteným záznamom vykonaná. Hodnota 1 znamená operáciu DELETE, hodnota 2 operáciu INSERT a 3 je pre UPDATE

Aby ste sa presvedčili o fungovaní zachytávania zmien, urobte ešte ďalšie dve operácie. Jeden záznam vymažte a jeden zeditujete:

```
DELETE FROM dbo.zamestnanci WHERE ID=2
UPDATE dbo.zamestnanci SET email = 'mouse@disney.com'  WHERE ID=1
```

Následne si pozrite aké údaje obsahuje tabuľka zamestnancov a čo sa zapísalo do tabuľky zachytávania zmien. Jej obsah môžete znova vypísať pomocou SQL príkazu

```
SELECT * FROM cdc.dbo_zamestnanci_CT;
```



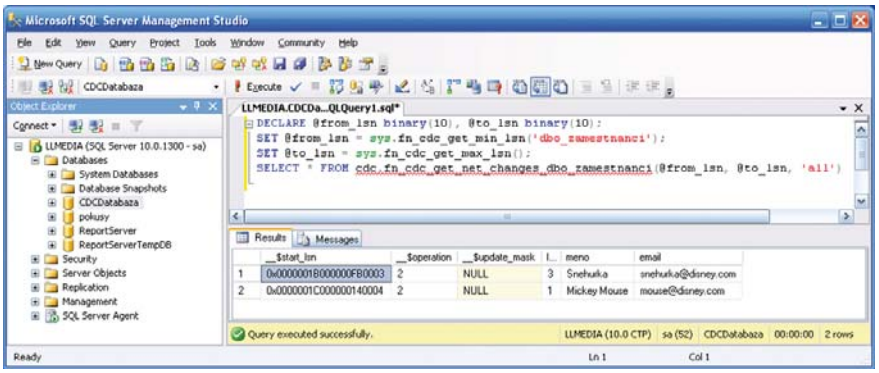
Obsah tabuľky pre zachytávanie zmien po zmenách údajov

Pre zistenie zmien máme k dispozícii aj dve systémové funkcie

`cdc.fn_cdc_get_all_changes_<capture_instance>` – vypíše všetky zmeny
`cdc.fn_cdc_get_net_changes_<capture_instance>` – len zmeny

Ukážeme príklad ich použitia:

```
DECLARE @from_lsn binary(10), @to_lsn binary(10);
SET @from_lsn = sys.fn_cdc_get_min_lsn('dbo_zamestnanci');
SET @to_lsn = sys.fn_cdc_get_max_lsn();
SELECT * FROM cdc.fn_cdc_get_all_changes_dbo_zamestnanci(@from_lsn, @to_lsn, 'all')
```



Výpis zmien pomocou systémových funkcií

Údaje v tabuľke zachytávania môžeme zmazať a poznačiť si čas tejto operácie

```
DECLARE @end_time datetime;
DECLARE @to_lsn binary(10);
SET @end_time = GETDATE();
SELECT @to_lsn = sys.fn_cdc_map_time_to_lsn('largest less than or equal', @end_time);
exec sys.sp_cdc_cleanup_change_table @capture_instance = 'dbo_zamestnanci', @low_
water_mark=@to_lsn
```

Príkaz MERGE

Novinkou jazyka SQL využiteľnou v integračných projektoch vo verzii 2008 je aj príkaz MERGE. Je v zhode so štandardom SQL 2006. Slúži na synchronizáciu obsahu tabuliek. Ak pri vkladaní záznamu zo zdrojovej do cieľovej tabuľky takýto záznam ešte neexistuje, uloží sa do databázy, alebo do údajového skladu ako nový. Ak záznam už existuje, tak sa pomocou príkazu MERGE aktualizuje už existujúci záznam. Pomocou tohto príkazu môžeme nahradiť sekvenciu viacerých príkazov typu INSERT, UPDATE, DELETE, nehovoriac už o zisťovaní či taký záznam existuje, prípadne o ošetrovaní výnimky pri klasickom vkladaní už existujúceho záznamu.

Príkaz MERGE spája dve tabuľky podľa kritéria:

```
MERGE target USING source ON
```

Podľa výsledku spojenia vykonáva INSERT/UPDATE/DELETE.

V príkaze môžeme využívať dodatočné klauzuly:

```
WHEN MATCHED THEN
   Riadky spĺňajúce kritéria. (INNER JOIN)
WHEN [TARGET] NOT MATCHED THEN
  Nevyhovujúce riadky v TARGET tabuľke (LEFT OUTER JOIN)
WHEN NOT MATCHED BY SOURCE THEN
  Nevyhovujúce riadky v SOURCE tabuľke (RIGHT OUTER JOIN)
```

Použitie príkazu MERGE ukážeme na praktickom príklade. Pomocou príkazu CERATE TABLE vytvoríme dve tabuľky – zdrojovú a cieľovú, ktoré budeme synchronizovať.

```
CREATE TABLE zdroj (id INT, meno NVARCHAR(20), skore_hry INT);
CREATE TABLE ciel (id INT, meno NVARCHAR(20), skore_hry INT);
```

Do nich vložíme niekoľko záznamov:

```
INSERT INTO zdroj VALUES (1, 'Peter', 100);
INSERT INTO zdroj VALUES (2, 'Ignac', 200);
INSERT INTO ciel VALUES (1, 'Mirka', 100);
INSERT INTO ciel VALUES (2, 'Ignac', 100);
```

Pomocou príkazu MERGE zosynchronizujeme cieľovú tabuľku so zdrojovou:

```
MERGE ciel AS c
    USING zdroj as z
    ON c.id = z.id
    WHEN MATCHED AND
       (c.meno != z.meno OR c.skore_hry!= z.skore_hry) THEN
        -- Ak riadok existuje no obsahuje rozdielne udaje
        UPDATE SET c.meno = z.meno, c.skore_hry = z.skore_hry
    WHEN NOT MATCHED THEN
        -- Ak riadok existuje len v zdrojovej tabulke
       INSERT VALUES (z.id, z.meno, z.skore_hry)
    WHEN NOT MATCHED BY SOURCE THEN
        -- Ak riadok existuje len v cielovej tabulke vymazeme ho
        DELETE
    OUTPUT $action, inserted.id, deleted.id;
```

V poslednom riadku príkazu sme nechali vypísať ID vložených a vymazaných záznamov:

```
$action    id          id
---------- ----------- -----------
UPDATE     1           1
UPDATE     2           2
```

Po vykonaní príkazu MERGE naše cvičné tabuľky obsahujú údaje:

zdroj

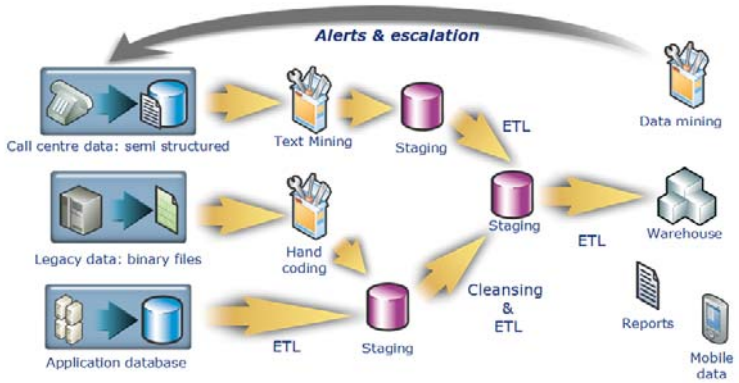
```
id          meno                 skore_hry
----------- -------------------- -----------
1           Peter                100
2           Ignac                200
```

cieľ

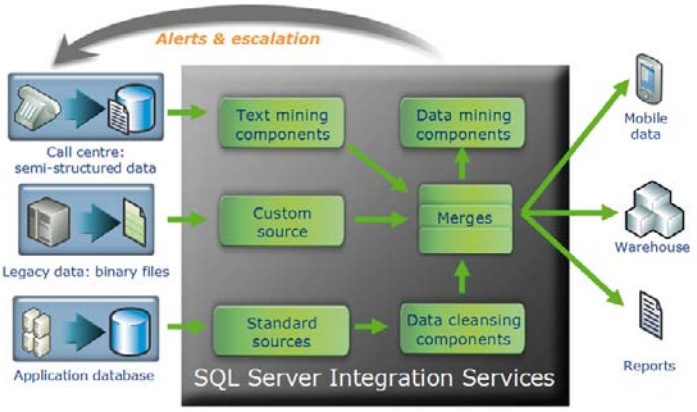
```
id          meno                 skore_hry
----------- -------------------- -----------
1           Peter                100
2           Ignac                200
```

Využitie integračných služieb pre BI projekty

Business Intelligence projekty sa spravidla nezačínajú budovať úplne „nanovo", teda spôsobom, že doteraz tu nebolo nič a našim cieľom je vybudovať BI systém pre podporu rozhodovania. Spravidla aj v týchto prípadoch používa zákazník na získavanie informácií pre podporu rozhodovania nejaké provizórne riešenie, napríklad sa generujú výstupné zostavy z transakčných systémov a tieto sú potom buď ručne alebo pomocou automatizačných prvkov softvéru typu Office spracovávané do manažérskych podkladov poskytovaných manažérom pre podporu rozhodovania. Vo všeobecnosti údaje, ktoré chceme aby vstupovali do procesu business intelligence pochádzajú z rôznych nehomogénnych zdrojov. Môžu to byť údaje zo súborových databáz (Access, dBase...), údaje z databáz spravovaných niektorým databázovým serverom (Oracle, Informix, Microsoft SQL Server, Sybase, Interbase, Ingres...), alebo údaje vyexportované nejakou databázovou platformou alebo informačným systémom, napríklad pobočkovou telefónnou ústredňou do tzv. flat file, XML dokumentu a podobne. Príprava a zavedenie údajov je dôležitou súčasťou každého BI riešenia aj riešenia údajového skladu. Údaje z operačného prostredia je potrebné pred zavedením do údajového skladu vyextrahovať, vyčistiť, upraviť a až následne vo vhodnej forme do údajového skladu zaviesť.



Typický prípad schémy zavádzania údajov z nehomogénnych zdrojov do údajového skladu bez použitia integračných služieb

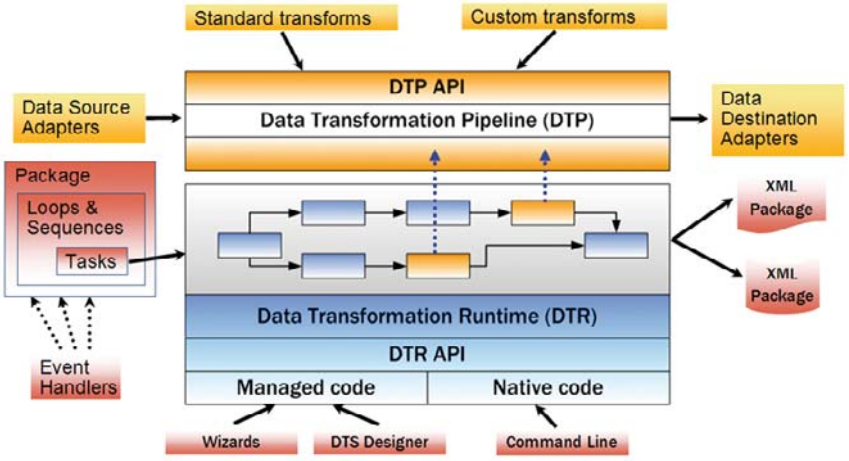


Scenár zavádzania údajov z nehomogénnych zdrojov do údajového skladu s využitím integračných služieb servera SQL Server 2008

Okrem už spomínaného zavádzania údajov z produkčných databáz do údajových skladov nájdu Integračné služby uplatnenie aj pri zbere údajov z rôznorodých systémov a reorganizácii štruktúr údajov, ktoré vyplývajú napríklad zo zmeny organizačnej štruktúry, alebo zo zmeny prípadne reštrukturalizácie predmetu podnikania.

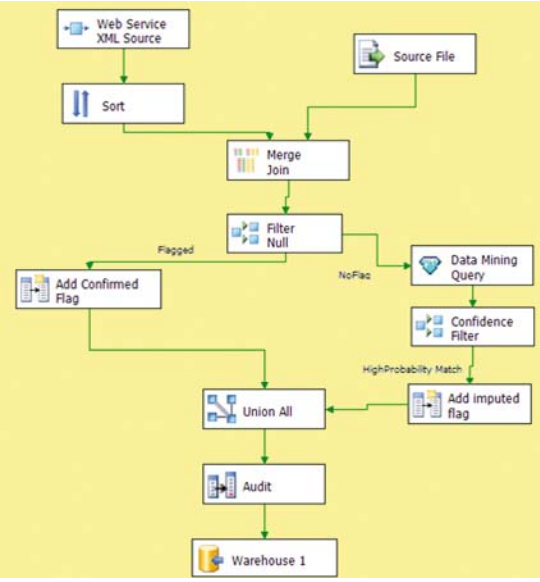
Správne zvládnutá etapa ETL je nevyhnutná v obidvoch popisovaných scenároch, teda aj pre rýchlu a úspešnú migráciu údajov v databázových projektoch aj pre nasadzovanie a prevádzkovanie projektov Business Intelligence a údajových skladov. V obidvoch prípadoch sa proces ETL zapojí do určitého stavu informačného systému. K dispozícii sú rôzne zostavy, dokumenty a údaje z primárnych transakčných systémov **OLTP** (On-line Transaction Processing).

Princíp činnosti Integračných služieb je zrejmý z architektonickej schémy. Hlavným blokom architektúry je Data Transformation Pipeline (DTP), ktorý prepája zdrojový a cieľový údajový adaptér. Na nižšej vrstve sa vykonávajú komplexné úlohy, z ktorých pozostáva proces extrakcie a transformácie.



Architektúra Integračných služieb servera SQL Server

Základné prvky tejto vrstvy (zreťazené bloky v strede architektonickej schémy) tvoria Tasks, teda samostatne vykonateľné jednoduché čiastkové úlohy podieľajúce sa na komplexnom procese transformácie. Mohli by sme to prirovnať k príkazom v ľubovoľnom programovacom jazyku. Po zreťazení úloh a ich doplnení o rôzne cykly získame komplexnejšie samostatné bloky – balíčky (Packages). Zreťazenie jednotlivých úloh je definované pomocou Task Flow diagramu. Farby čiar so šípkami medzi úlohami znázorňujú v akej podmienkovej vetve sa dané bloky nachádzajú. Napríklad vetva predpokladaného úspešného priebehu je vyznačená zelenou farbou a procesný tok v prípade nesplnenia nejakej podmienky je znázornený červenou farbou.



Pomocou Integračných služieb je možné vytvárať aj veľmi sofistikované projekty pre spracovanie údajov z heterogénnych zdrojov a ich zavádzanie do údajového skladu

Integračné služby v prostredí Business Intelligence Development Studio

Najskôr sa zoznámime s možnosťami a usporiadaním obrazovky vývojového prostredia v režime Integration Services Project. Zoznámenie vykonáme na praktickom príklade importu flat súboru, teda textového súboru s hodnotami oddelenými čiarkami. Ako námet pre príklad využijeme trochu žartovné údaje o tom ako jednotlivé plemená psov hrajú poker. Je to paródia na hru dogsplayingpoker.org. Mimochodom s týmito údajmi pracuje aj tutorial pre datamining na platforme Oracle, takže po naštudovaní problematiky dataminingu môžete porovnávať.

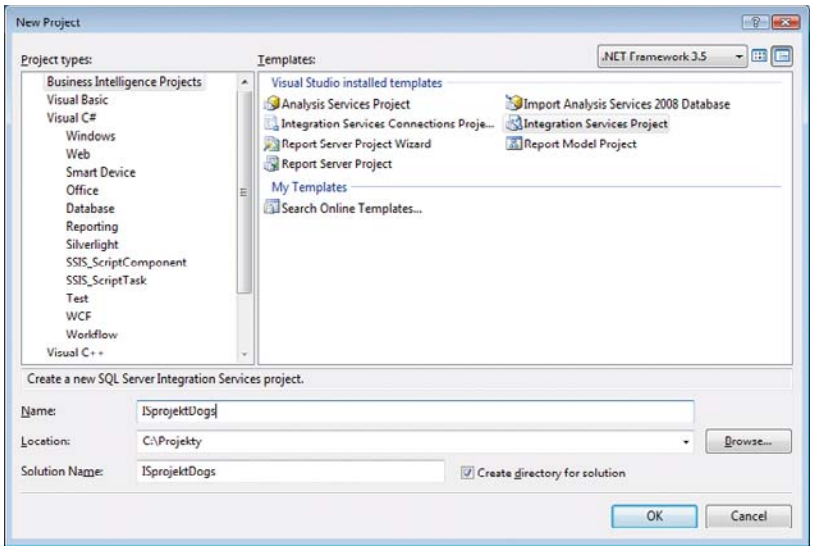
Súbor DogsPlayingPoker_Results.csv obsahuje údaje v tvare:

```
ID,PLEMENO,VEK,ZRADLO,KUPIROVANY,HRA,VYSLEDOK
1,Affenpinscher,2,CANNED,YES,5 CARD DRAW,1
2,Afghan Hound,7,STORE BRAND,NO,5 CARD STUD,0
3,Airedale Terriers,14,HIGH END,YES,4 CARD GUTS,0
4,Akbash Dogs,8,TABLE SCRAPS,NO,TEXAS HOLD'EM,0
..
5906,Xoloitzcuintli,12,TABLE SCRAPS,NO,OMAHA,0
5907,Yorkshire Terrier,5,CANNED,YES,5 CARD DRAW,0
```

Prvý riadok súboru obsahuje názvy atribútov.

Trochu nelogicky začneme od konca. Údaje potrebujeme umiestniť do databázy, preto ak nemáme vhodnú databázu, musíme ju najskôr pomocou nástroja SQL Server Management Studio vytvoriť.

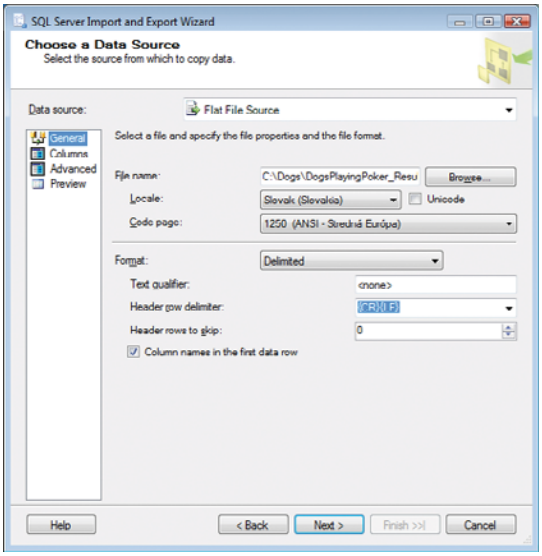
Vytvoríme nový projekt podľa šablóny Business Intelligence Projects. Nazveme ho napríklad ISprojektDogs. V priečinku SSIS Packages bude automaticky vytvorený balíček Package.dtsx. V tomto prvom príklade však tento balíček potrebovať nebudeme, takže ho môžeme vymazať. Použijeme kontextové menu SSIS Import and Export Wizard priečinka SSIS Packages. Aktivujeme tým rovnako pomenovaného sprievodcu.



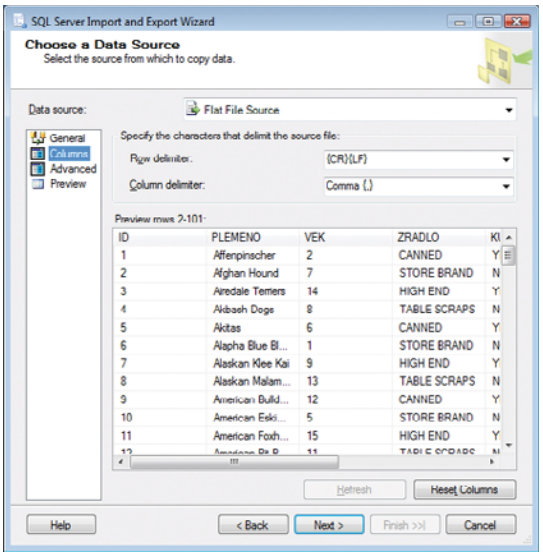
Vytvorenie nového projektu

Podľa pokynov sprievodcu najskôr vyšpecifikujeme zdroj údajov. Vyberieme ako druh pripojenia „Flat File Source“, pomocou tlačidla „Browse“ otvoríme súbor ktorý bude pre nás zdrojom údajov. Všimnite si, že po jeho otvorení sa automaticky doplní znaková sada, údaj o použití delimitovaného formátu a podobne. Nástroj „Flat File Connection Manager Editor“ akceptuje tri typy formátovania textových súborov:

- Delimited – Stĺpce sú od seba oddelené znakom delimitátora. Delimitátor (čiarka, bodkočiarka, tabulátor) je definovaný v režime dialógu „Columns“.
- Fixed width – stĺpce majú pevnú šírku.
- Ragged right – stĺpce majú pevnú šírku s výnimkou posledného, ktorý môže mať premenlivú šírku. Posledný stĺpec je potom ukončený znakmi riadkového delimitátora,



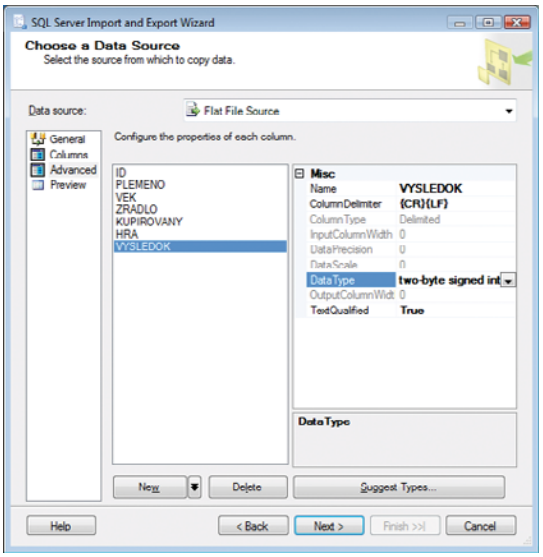
Výber zdroja údajov – záložka General



Výber zdroja údajov – záložka Columns

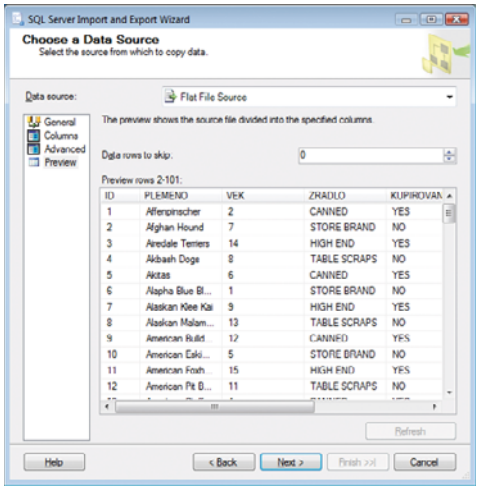
V záložke Advanced môžeme definovať údajový typ pre jednotlivé atribúty.

V našom príklade zmeníme údajový typ atribútov ID, VEK a VYSLEDOK na dvojbajtový integer.



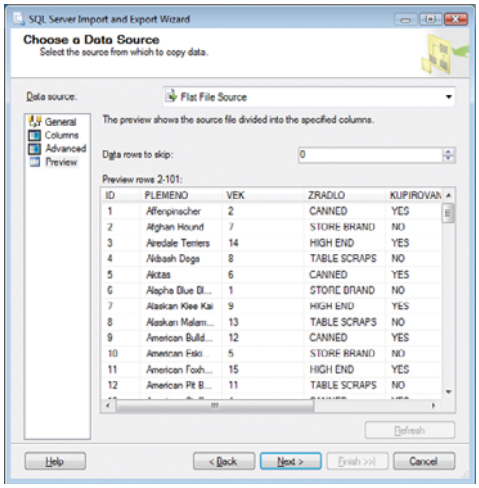
Výber zdroja údajov – v záložke Advanced zmeníme atribúty

Nakoniec si môžeme hodnoty určené pre konverziu skontrolovať v záložke „Preview". Ak je preview údajov v poriadku a príslušné atribúty majú očakávané hodnoty, je veľmi vysoká pravdepodobnosť, že sa podarí aj celý prevod flat – súboru do databázovej tabuľky.



Preview údajov

Ako cieľovú databázu vyberieme napríklad databázu Test:

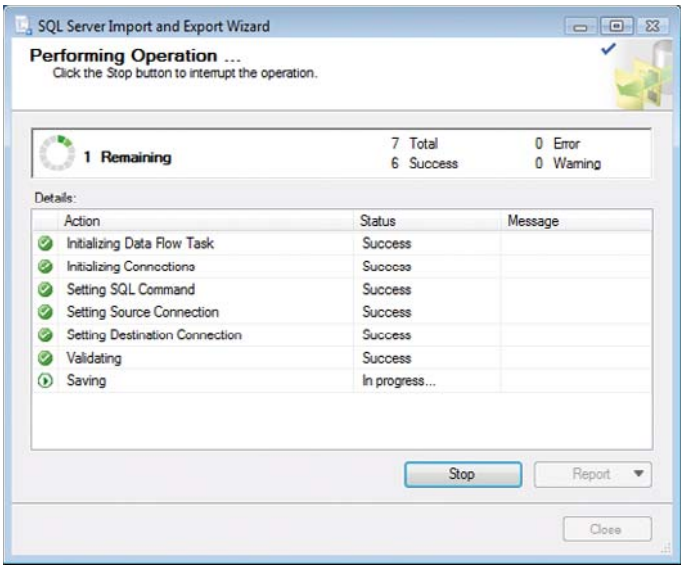


Mapovanie dátových typov

Skontrolovať a v prípade potreby aj upraviť môžeme aj SQL príkaz pre vytvorenie tabuľky v cieľovej databáze.

```
CREATE TABLE [dbo].[DogsPlayingPoker_Results]
(
  [ID] smallint,
  [PLEMENO] varchar(50),
  [VEK] smallint,
  [ZRADLO] varchar(50),
  [KUPIROVANY] varchar(50),
  [HRA] varchar(50),
  [VYSLEDOK] smallint
)
```

Finálnym krokom sprievodcu je vytvorenie IS (Information Services) úlohy. Priebeh tohto procesu, teda vynášania metaúdajov môžeme sledovať v dialógu Performing Operation. Tu si treba uvedomiť, že v tejto fáze sa ešte s údajmi samotnými zatiaľ nepracuje.

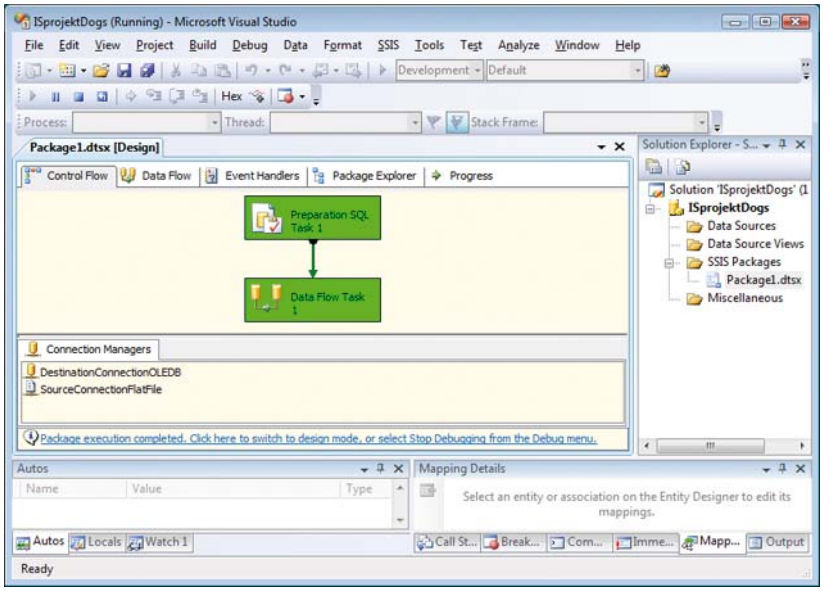


Priebeh vynášania metadát

Výsledok zostavovania úlohy je zobrazený na pracovnej obrazovke vývojového prostredia v záložkách Control Flow a DataFlow.

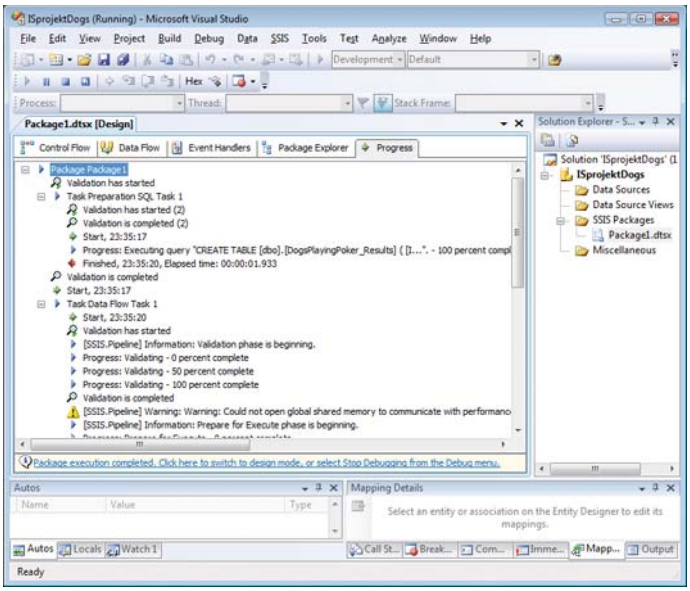
Po spustení IS projektu môžeme v okne Control Flow sledovať priebeh jeho vykonávania. Stav vzhľadom na jednotlivé objekty je signalizovaný farbou ich symbolov. Význam jednotlivých farieb je nasledovný:

- Sivá – čakanie na spustenie
- Žltá – aktuálne spustené
- Zelená – úspech
- Červená – chyba



Sledovanie prebehu v Záložke Control Flow

Zároveň môžeme podľa čísiel nad blokmi cieľovej destinácie informatívne sledovať počet spracovaných a prenesených záznamov. Ak spúšťame alebo ladíme zložitý projekt a navyše vo veľkej databáze, môže beh integračnej úlohy trvať aj pomerne dlho. Priebeh úlohy môžeme sledovať v záložke Progress:



Sledovanie priebehu úlohy

... úloha pokračuje úvodným príkladom v kapitole OLAP analýzou údajov z údajového tržiska.

Záložka ControlFlow

Záložka Control Flow je orientovaná na procesy. V nej je možné definovať ľubovoľné operácie nad určitými objektmi. V režime návrhu jednoduchého IS projektu sú v záložke ControlFlow umiestnené ikony jednotlivých čiastkových úloh, z ktorých sa projekt skladá.

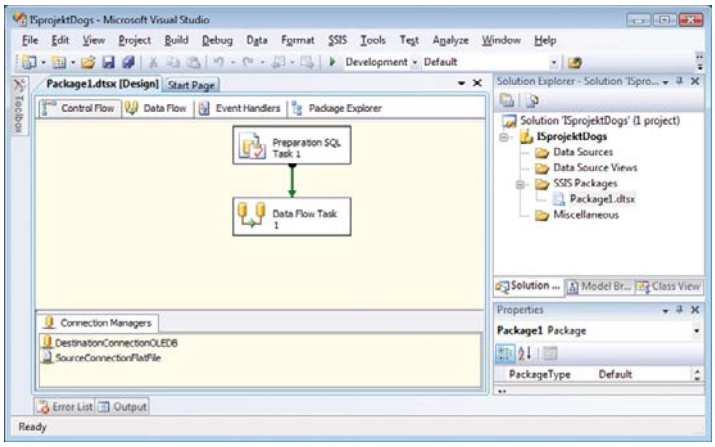
V záložke Control Flow sa zobrazuje priebeh spustenia úlohy po jednotlivých funkčných blokoch. V našom jednoduchom príklade kopírovania údajov zo zdrojových tabuliek do cieľovej databázy má tento diagram len dva bloky:
- Preparation SQL Task,
- Data Flow Task.

Pracovné okno vývojového prostredia obsahuje dva oddelené editory, jeden pre návrh údajových tokov a jeden pre návrh riadiacich tokov. Tomu je prispôsobená aj pracovná plocha vývojového prostredia. Jej hlavné okno je rozdelené na štyri záložky:
- Control Flow,
- Data Flow,
- Event Handlers,
- Package Explorer.

Približné poradie úkonov pri návrhu integračného projektu nám naznačujú priečinky v okne Solution Explorer. Filozofia usporiadania a nadväznosti priečinkov v tomto okne vyplýva z koncepcie UDM modelovania (viď predchádzajúcu kapitolu). Okno obsahuje štyri priečinky:
- Data Sources,
- Data Source Views,
- SSIS Packages,
- Miscellaneous.

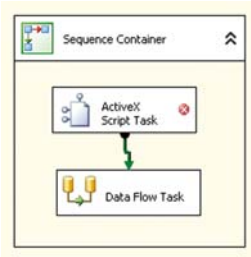
Každá operácia môže skončiť v jednom z troch základných stavov:
- Success,
- Fail,
- Completion result.

Medzi týmito výsledkami je možné definovať logické operácie AND a OR. Paletu možností Integračných služieb servera SQL Server 2008 dokresľuje ponuka symbolov modulov v okne Toolbox.



Záložka Control Flow

Na začiatku zoznamu prvkov sú tri kontajnery. Pomocou prvku **For Loop Container** je možné vytvárať opakované cykly v diagrame Control Flow. Cyklus je riadený počítadlom cyklu. Definujeme počiatočnú podmienku, koncovú podmienku a inkrementáciu počítadla.
Pri **Foreach Loop Containeri** je koniec cyklu daný enumerátorom, napríklad vyčerpaním všetkých prvkov z danej množiny. **Sequence Container** slúži na zapuzdrenie niekoľkých blokov Control Flow diagramu, napríklad rozdielnych tokov po výstupe z bloku transformácie Conditional Split.



Sequence Container

Symbol kontajnera integračných služieb obsahuje okrem záhlavia, aj kontajnerovú plochu na ktorú je možné umiestňovať symboly ďalších prvkov, s ktorými v rámci kontajneru pracujeme.

Prípravné úlohy pracujú so súbormi, priečinkami, umožňujú preberať webový obsah a načítať obsah XML dokumentov. Sú len prípravné, s vlastným obsahom údajov nepracujú. Do tejto skupiny môžeme zaradiť úlohy **File System Task**: súbor úloh pre prácu so súbormi a priečinkami. Umožňuje ich vytvárať premenovávať, alebo vymazávať:
- **FTP Task** – umožňuje prenos súborov prostredníctvom FTP,
- **Web Service Task** – spúšťa metódy webových služieb,
- **XML Task** – práca s XML dokumentom v súbore, parsovanie, vyhľadávanie uzlov a podobne.

Workflow úlohy komunikujú s procesmi, operačným systémom a službami, kde spúšťajú balíčky, programy, posielajú správy medzi balíčkami, e-maily, čítajú údaje z Windows Management Instrumentation (WMI) a dohliadajú na WMI udalosti. Do tejto skupiny patria úlohy:
- **Execute Package Task**: umožňuje spúšťanie IS balíkov z vnútra balíka, čím podstatne zvyšuje modularitu možností integračných služieb.
- **Execute Process Task**: spúšťanie programových modulov mimo balíka,
- **Message Queue Task**: odosiela alebo prijíma správy Microsoft Message Queue (MSMQ).
- **Send Mail Task**: poslanie správy prostredníctvom SMTP protokolu. Napríklad ak rozbehneme úlohu, ktorá má trvať veľmi dlho, a po hodine dôjde k chybe, je potrebné oznámiť to obsluhe aby operatívne zasiahla.

Úlohy pre SQL Server umožňujú prístup k údajom pod správou SQL Servera, napríklad ich kopírovanie, vkladanie, vymazávanie a modifikáciu. Taktiež môžeme vykonávať úkony s objektmi pod správou SQL Servera. Sem môžeme zaradiť úlohy:

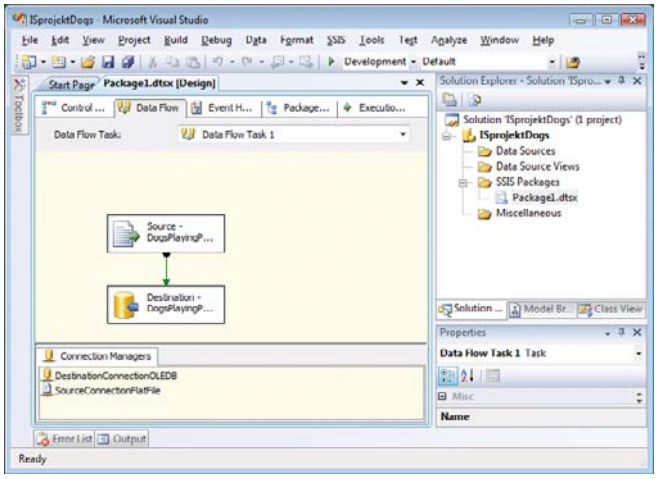
- **Bulk Insert Task** – načítanie údajov zo súborov pomocou príkazu BULK INSERT SQL,
- **Execute SQL Task** – spúšťa SQL príkaz alebo uloženú procedúru.

Niektoré úlohy môžu pracovať s objektmi a procesmi analytických služieb, napríklad:

- **Analysis Services Processing Task** – spúšťa úlohu pod analytickými službami, napríklad kocku alebo data miningový model,
- **Analysis Services Execute DDL Task** – spúšťa úlohu typu DDL (Data Definition Language – jazyk pre definovanie údajov) v analytických službách. Tak ako sa v relačných databázach používajú DDL príkazy pre vytváranie relačných objektov (tabuliek, indexov...), v analytických službách a tieto príkazy využívajú pre vytváranie, zmeny, alebo odstraňovanie kociek.
- **Data Mining Query Task**: povoľuje spúšťanie prediktívnych dopytov prostredníctvom analytických služieb na data miningových modeloch.

Záložka Data Flow

Data Flow Task je komplexná úloha pozostávajúca s viac alebo menej blokov pre načítanie údajov, ici transformáciu a uloženie do cieľovej databázy. Tento proces je orientovaný na prácu s údajmi. Podľa režimu práce produkuje alebo konzumuje riadky s údajmi.



Záložka DataFlow

Symboly Toolboxu sú logicky rozdelené do troch skupín:

- Data Flow Sources
- Data Flow Transformations
- Data Flow Destinations

Niektoré často používané transformácie predstavíme podrobnejšie.
Agregačná transformácia umožňuje aplikovať agregačné funkcie (Min(), Max(), Average(), Sum()...) na množinu údajov, ktorá vstupuje do transformácie. Agregačné funkcie operujú nad množinou záznamov, pričom vracajú jeden výsledok pre celú vstupnú množinu údajov. **Audit** transformácia umožňuje operatívne získavanie údajov z prostredia, kde je Data Flow úloha spustená. Transformácia **Conditional Split** slúži na rozdelenie množiny údajov na viac podmnožín podľa určitých kritérií. Napríklad usporiadať záznamy osôb do skupín podľa prvého písmena, rozdelenie do intervalov podľa hodnoty niektorého atribútu a podobne. Transformácia **Copy Column** umožňuje pridanie nových stĺpcov, ktoré sú kópiami stĺpcov zo vstupnej množiny údajov. **Data Conversion** prenáša údaj zo špecifikovaných vstupných stĺpcov do výstupných stĺpcov, pričom je možné zmeniť ich údajový typ. Transformácia **Derived Column** umožňuje vytvoriť stĺpec, ktorého obsah je vytvorený z iných stĺpcov za pomoci SSIS operátorov a funkcií.

Import údajov z databázových tabuliek

V predchádzajúcom príklade sme ukázali jednoduchý import údajov bez akýchkoľvek úprav. V tejto stati sa pokúsime oveľa viac priblížiť realite zavedenia údajov z relačných OLTP databáz do údajových skladov. Využijeme cvičnú OLTP databázu AdventureWorks2008. Aby sme mohli s touto cvičnou databázou efektívne pracovať, mali by sme sa s ňou aspoň letmo zoznámiť. Databáza je organizovaná podľa schém.

Schéma	Popis objektov	Tabuľky
HumanResources	Zamestnanci spoločnosti Adventure Works Cycles.	Employee Department
Person	Mená a adresy zákazníkov, predajcov a zamestnancov.	Contact Address StateProvince
Production	Produkty vyrábané a predávané spoločnosťou Adventure Works Cycles.	BillOfMaterials Product WorkOrder
Purchasing	Dodávatelia súčiastok.	PurchaseOrderDetail PurchaseOrderHeader Vendor
Sales	Údaje týkajúce sa obchodu a zákazníkov	Customer SalesOrderDetail SalesOrderHeader

Námetom príkladu nad databázou AdventureWorks2008 bude vytvorenie operačného údajového tržiska nad údajmi z OLTP databázy. Technicky zjednodušene povedané potrebujeme preniesť niekoľko tabuliek zo zdrojovej do cieľovej databázy. Pre tento účel môžeme z databázy AdventureWorks2008 využiť napríklad tabuľky zo schém SALES, kde sú uložené údaje z oblasti marketingu a predaja firmy AdventureWorks. Spoločnosť má dva typy zákazníkov, ktorí sa odlišujú pomocou atribútu CustomerType:
- individuálnych, ktorí nakupujú cez webový on-line obchod (CustomerType = 'I')
- obchody s bicyklami a cyklistickými potrebami (CustomerType = 'S')

Nakoľko firemné procesy nie sú izolované ale navzájom previazané, potrebujeme pre niektoré výpisy týkajúce sa predaja a marketingu aj tabuľky z iných schém. Do všetkých procesov vstupujú osoby (zamestnanci, obchodníci, zákazníci), ich údaje sú uložené v tabuľkách schémy Person. Pre náš príklad využijeme z tejto schémy tabuľky objednávok:
- Sales.SalesOrderHeader,
- Sales.SalesOrderDetail.

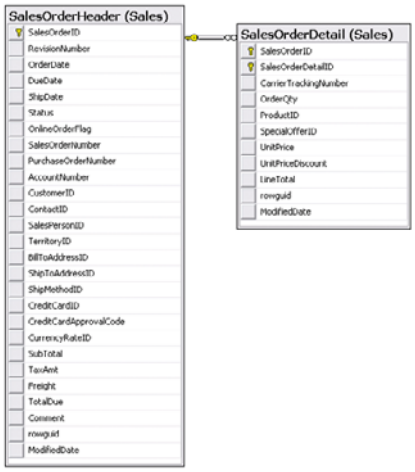


Diagram väzieb tabuliek pre výpis údajov o objednávkach AdventureWorks použitých v príklade

Druhá schéma, ktorú v príklade využijeme bude PRODUCTION. Firma vyrába bicykle a rôzne cyklistické príslušenstvo. Údaje ohľadne produktov firmy sú uložené v tabuľkách schémy Production.. V príklade využijeme z tejto schémy tabuľky:

- Production.Product,
- Production.ProductSubcategory,
- Production.ProductCategory.

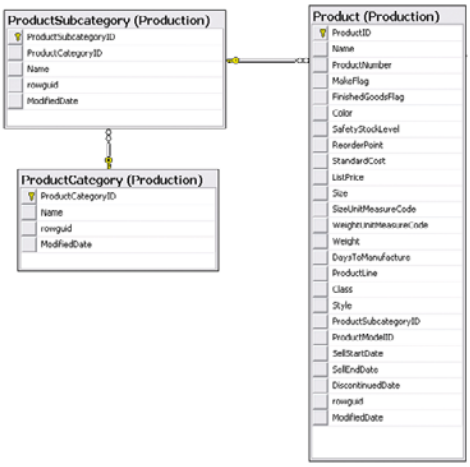


Diagram väzieb tabuliek pre výpis produkcie firmy AdventureWorks použitých v príklade

Schémy použitých tabuliek sme nepublikovali náhodou. Všimnite si jednu z teoretického hľadiska dosť podstatnú záležitosť. Chýba tu tabuľka, ktorá by bola jednoznačným podkladom pre časovú dimenziu. Z vybraných tabuliek však časové údaje obsahuje tabuľka OrderHeader. Takto to býva aj v praxi. Každý si ukladá do databáz údaje o predmete svojho podnikania a tieto samozrejme obsahujú aj dátumové a časové údaje.

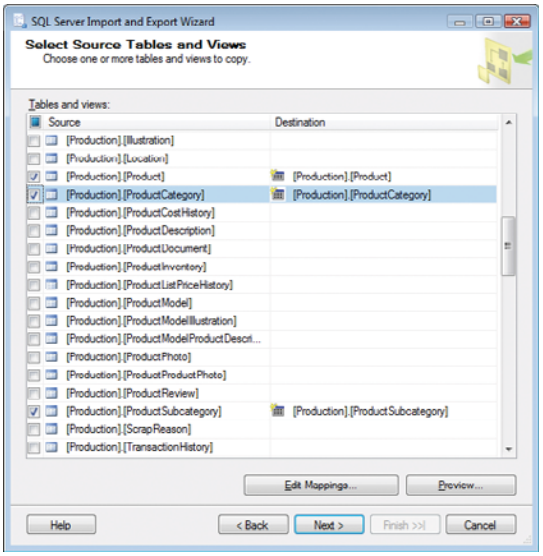
Ako prvý prípravný krok vytvoríme cieľovú databázu napríklad s názvom ISmarket. Následne vytvoríme projekt typu Information Services Project. Ani v tomto príklade nebudeme potrebovať automaticky vytvorený balíček Package.dtsx.

Podľa pokynov sprievodcu vyberieme zdrojovú databázu AdventureWorks 2008. Ako cieľovú databázu vyberieme zatiaľ prázdnu nami vytvorenú databázu ISmarket.

V nasledujúcom kroku máme možnosť zvoliť si metódu pre výber množiny údajov, ktoré potrebujeme preniesť a prípadne pretransformovať. Máme na výber dve možnosti:
– kopírovať vybrané tabuľky a pohľady ako celky zo zdroja údajov do cieľovej databázy,
– pomocou Query dotazu špecifikovať podmnožiny údajov, ktoré potrebujeme pretransformovať.

Budeme prenášať tabuľky:
– Sales.SalesOrderHeader,
– Sales.SalesOrderDetail,
– Production.Product,
– Production.ProductSubcategory,
– Production.ProductCategory.

Alternatívne by sme nemuseli prenášať údaje z celých tabuliek, ale využiť možnosť vytvorenia SQL dopytu pre výber podmnožiny údajov zo zdrojovej databázy do cieľovej.



Výber tabuliek pre prenos údajov zo zdrojovej do cieľovej databázy

Výsledok zostavovania úlohy je zobrazený v záložkách Control Flow a DataFlow.

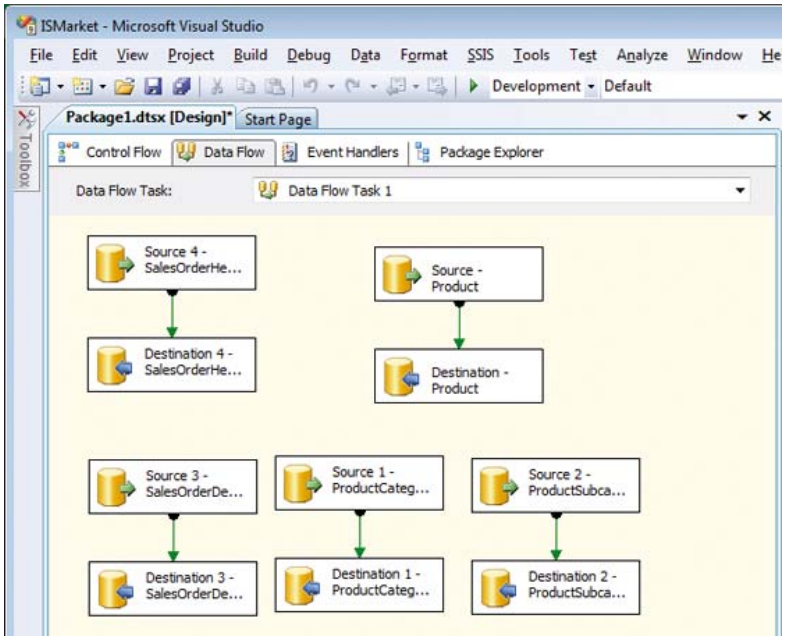
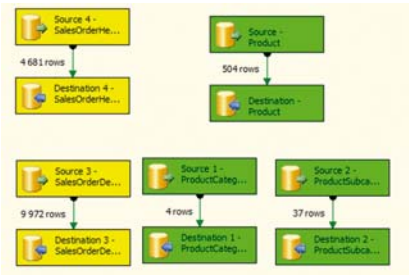


Diagram v záložke Data Flow

V záložke Data Flow vidíme päť vzájomne nesúvisiacich čiastkových úloh. Pre každú prevádzanú tabuľku je znázornený tok údajov zo zdrojového do cieľového úložiska. Tieto úložiská sú znázornené obdĺžnikmi. Veľmi dôležitým symbolom je aj šípka medzi nimi, kde sú nadefinované transformačné operácie. Kliknutím na túto šípku aktivujeme Data Flow Path editor, kde si môžeme prehliadnuť metaúdaje. Kliknutím na symbol zdroja alebo cieľa údajov vyvoláme Source Editor, prípadne Destiantion Editor. V tejto etape môžeme napríklad pomocou Source Editora v záložke Columns vynechať z transformácie stĺpce, ktoré v cieľových tabuľkách nepotrebujeme.
Samotnú úlohu spustíme zelenou šípkou na toolbare vývojového prostredia. Na procesnom diagrame môžeme podľa farieb jednotlivých symbolov sledovať priebeh vykonávania úlohy, pričom aktuálne bežiace úlohy sú znázornené žltým pozadím symbolu, úspešne ukončené úlohy zeleným pozadím a prípadné chybne ukončené úlohy červeným pozadím symbolu. Zároveň môžeme podľa čísiel nad blokmi cieľovej destinácie informatívne sledovať počet spracovaných a prenesených záznamov.



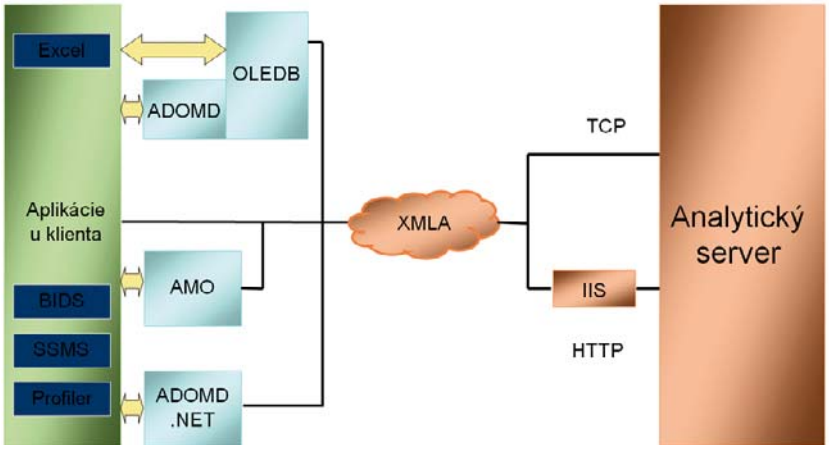
Priebeh vykonávania IS tasku je signalizovaný farbou pozadia jednotlivých blokov

Ak nám diagram po dobehnutí všetkých úloh „ozelenie", transformácia údajov prebehla úspešne. V prípade problémov nám pomôže napríklad protokol o priebehu IS úlohy, ktorý sa zobrazuje v okne Output v dolnej časti obrazovky vývojového prostredia.

...úloha pokračuje úvodným príkladom v kapitole OLAP analýzou údajov z údajového tržiska.

Kapitola 4: Analytické služby

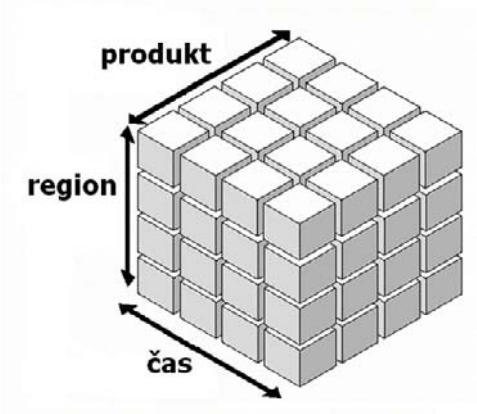
Pre úspešné zvládnutie problematiky multidimenzionálnych analýz je potrebné nielen uvedomiť si rozdiel medzi klasickou relačnou a multidimenzionálnou analytickou databázou, ale najmä poznať aspoň teoretické základy OLAP kociek.Základom pre vytvorenie OLAP kocky sú fakty a dimenzie.



Klient – server architektúra analytických služieb

Fakty sú numerické merné jednotky obchodovania. Tabuľka faktov je spravidla najväčšia tabuľka v databáze a obsahuje veľký objem dát. Niektoré jednoduchšie databázy zvyčajne obsahujú len jednu tabuľku faktov, iné, hlavne DSS schémy môžu obsahovať viaceré tabuľky faktov. Prvotné fakty, napríklad objem predaja, sa môžu kombinovať alebo vypočítať pomocou iných faktov a vytvoriť tak merné jednotky. Merné jednotky sa môžu uložiť v tabuľke faktov, prípadne vyvolať, ak je to nevyhnutné, na účely vykazovania.

Dimenzie obsahujú logicky alebo organizačne hierarchicky usporiadané údaje. Sú to vlastne textové popisy obchodovania. Tabuľky dimenzií sú zvyčajne menšie ako tabuľky faktov a údaje v nich sa nemenia tak často. Tabuľky dimenzií vysvetľujú všetky „prečo" a „ako" pokiaľ ide o obchodovanie a transakcie prvkov. Kým dimenzie vo všeobecnosti obsahujú relatívne stabilné dáta, dimenzie zákazníkov sa aktualizujú častejšie. Veľmi často sa používajú časové, produktové a geografické dimenzie. Narastajúcim počtom dimenzií geometrickým radom narastá veľkosť (množstvo údajov) OLAP kocky.



Príklad dimenzií

Tabuľky dimenzií obvykle obsahujú stromovú štruktúru. Napríklad dimenzia vytvorená na základe geografických informácii, teda regionálna dimenzia sa člení na jednotlivé úrovne podľa konkrétneho územnosprávneho členenia danej geografickej oblasti, napríklad (počet bodiek znamená úroveň vnorenia jednotlivých atribútov dimenzie):

Región
- Kontinent,
- • Krajina,
- • • Územný celok,
- • • • Mesto.

Hierarchicky býva usporiadaná aj produktová klasifikácia. Produkty sa začleňujú do druhov, kategórií, skupín a podobne, napríklad:

Produkt
- Druh produktu,
- • Kategória,
- • • Subkategória,
- • • • Názov produktu.

Takmer vždy sa používa ako dimenzia čas, kde sú jednotlivé úrovne definované podľa kalendárnych alebo fiškálnych zvyklostí napríklad:

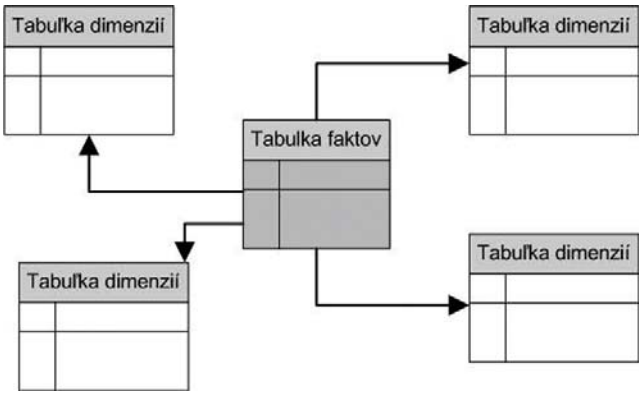
Čas
- Rok,
- • Kvartál,
- • • Mesiac,
- • • • Týždeň.

Ak uvedené údaje usporiadame do vhodne logicky organizovanej multidimenzionálnej štruktúry, budeme pracovať s oveľa menšími rozmermi dimenzií a údaje budú mať oveľa väčšiu vypovedaciu schopnosť. Multidimenzionálna informácia je navyše v čistej podobe, to znamená, že je orientovaná na predmet podnikania a nie je viazaná na konkrétny systém, ktorý je určený na zber a spracovanie základných údajov.

Tabuľky faktov a dimenzií môžu vytvárať určité schémy, napríklad hviezdicovú schému (star schema) alebo schému snehovej vločky (snowflake schema).

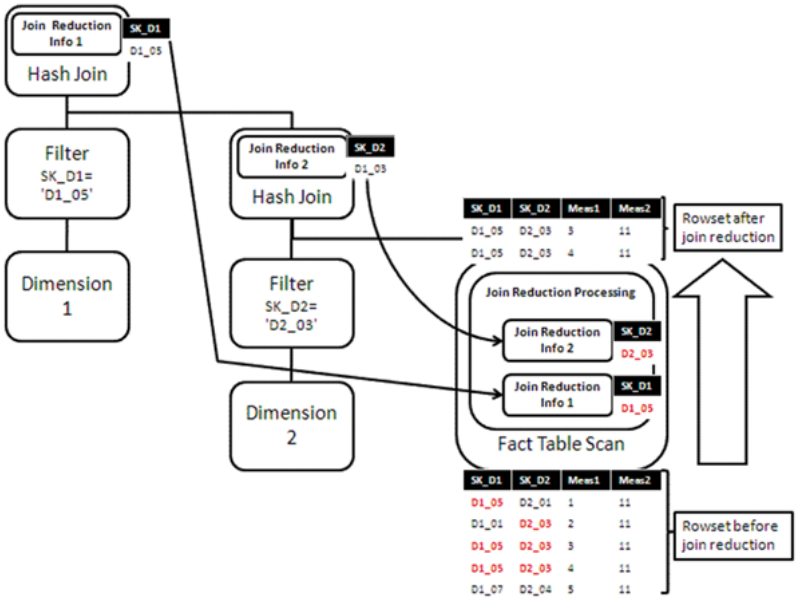
Schémy tabuliek faktov a dimenzií
Multidimenzionálne kocku (príklad vidíme na obrázku 1) vytvárame na základe dimenzionálneho modelu, ktorý má určité topologické usporiadanie, ktorému hovoríme schéma. Najčastejšie používame hviezdicovú schému (star schema), alebo schému „snehovej vločky" (snowflake schema).

Hviezdicová schéma sa skladá z tabuľky faktov obsahujúcich cudzie kľúče, ktoré sa vzťahujú k primárnym kľúčom v tabuľkách dimenzií. Hviezdicová schéma nemá normalizované dimenzie ani relačné prepojenie medzi tabuľkami dimenzií, preto je veľmi ľahko pochopiteľná, ale v dôsledku nenormalizovaných dimenzií je vytvorenie takéhoto modelu relatívne pomalé, ale na druhej strane, tento model poskytuje vysoký „dopytovací výkon".



Hviezdicová schéma

SQL Server 2008 prináša novú črtu nazývanú „Star Join", pomocou ktorej je možné výrazne urýchliť typické dopyty do hviezdicovej schémy relačne zviazaných tabuliek. Táto technológia je založená na Bloom filtri (podrobnejšie viď **http://en.wikipedia.org/wiki/Bloom_filter**) Bloom filter je špeciálny bitmapový filter a SQL server 2008 ho s výhodou využíva na eliminovanie riadkov z tabuľky faktov, ktoré sa nepodieľajú na výsledku. „Star Join" prináša 15–25 percentné zrýchlenie týchto dopytov.



Princíp „Star Join"

Schéma „snehovej vločky" obsahuje niektoré dimenzie zložené z viacerých relačne zviazaných tabuliek. Tento model umožňuje rýchlejšie zavedenie údajov do normalizovaných tabuliek, ale má podstatne nižší dopytovací výkon, lebo obsahuje väčšie množstvo spojení tabuliek.

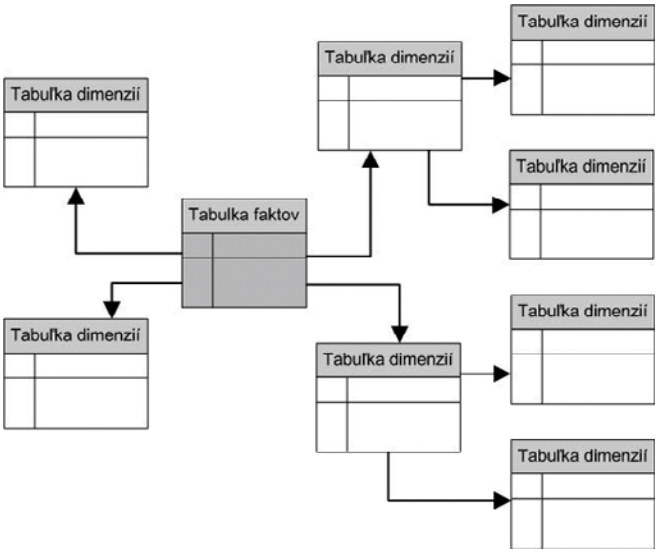
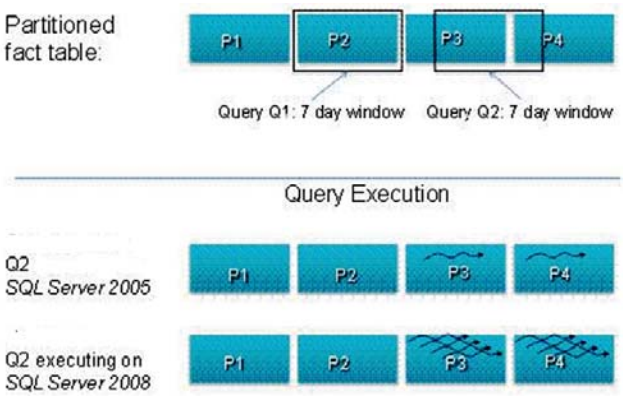


Schéma snehovej vločky (snowflake schema)

Paralelné spracovanie tabuľky rozdelenej na partície

Ak databázová aplikácia pracuje s veľkým množstvom údajov, je z hľadiska manipulácie s údajmi výhodné ukladať tieto údaje do databázových tabuliek rozdelených na niekoľko častí – partícií. Tabuľku môžeme na partície rozdeliť podľa rôznych kritérií, najčastejšie však podľa časového hľadiska. Pri databázach nad ktorými chceme robiť OLAP analýzy je najväčšou tabuľkou tabuľka faktov, preto práve táto tabuľka býva často rozdelená na viac partícii. Ak sa údaje vyhľadávajú v rámci jednej partície, nie je žiadny problém. Ak vyhľadávacie okno presahuje partíciu, situácia sa komplikuje. SQL Server pre vyhľadávanie v každej partícii alokoval jeden thread, takže potom pri finalizácii výsledkov dopytu odovzdávanie údajov medzi threadmi nie je optimálne. Nová verzia, SQL Server 2008 alokuje pre každú partíciu viac threadov, ktoré môžu pri finalizácii výsledkov dopytovania cez viac partícii efektívne spolupracovať.



Princíp paralelného spracovania tabuľky faktov rozdelenej na partície

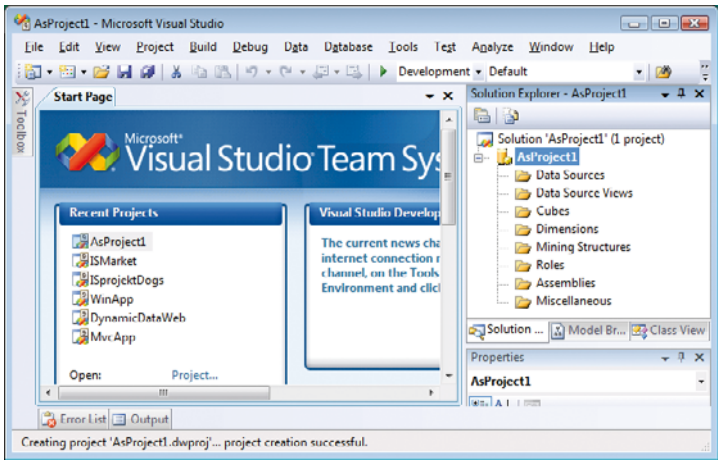
Na úvod praktickej časti kapitoly venovanej vytváraniu OLAP kociek apelujeme na čitateľa, ktorý doteraz nečítal tretiu kapitolu venovanú UDM (Unified Dimension Model), aby sa najskôr vrátil k tejto niekoľkostranovej kapitole, ktorá popisuje základné princípy a filozofiu vytvárania BI štruktúr. Na základe princípov UDM sú totiž koncipované všetky čiastkové úkony, ktoré vedú k vytvoreniu OLAP kociek.

Úvodný príklad pre vytvorenie OLAP kocky

... pokračovanie úvodného príkladu z predchádzajúcej kapitoly...
V predchádzajúcej kapitole sme vytvorili jednoduché údajové tržisk (samostatnú databázu) ISmarket, do ktorej boli prenesené z databázy AdventureWorks2008 tabuľky:
- Sales.SalesOrderHeader,
- Sales.SalesOrderDetail,
- Production.Product,
- Production.ProductSubcategory,
- Production.ProductCategory.

Tabuľky boli zo zdrojovej databázy do cieľovej prenesené bez akýchkoľvek úprav a transformácií, preto tento príklad môžu pokojne urobiť aj čitatelia, ktorí príklad z predchádzajúcej kapitoly vynechali. Vtedy pracujeme s originálnou zdrojovou databázou Adventure Works2008.

V o vývojovom prostredí Business Intelligence Development Studio vytvoríme nový projekt typu Analysis Services Project napríklad s názvom AsProject1. Postup v akom budeme novú aplikáciu vytvárať (vlastne modelovať) je naznačený poradím priečinkov v okne vývojového prostredia Solution Explorer v pravom hornom rohu. OLAP analýzy sa týkajú priečinkov:
- Data Sources,
- Data Source Views,
- Cubes,
- Dimension.



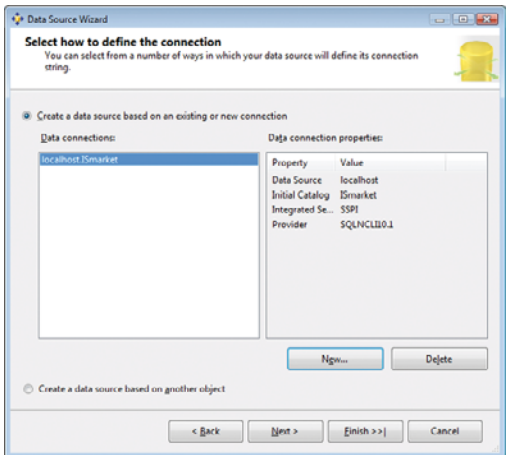
Organizácia návrhového prostredia

Príklad teda môžeme rozdeliť do štyroch hlavných krokov:
1. definovanie údajových zdrojov,
2. definovanie pohľadov na údajové zdroje,
3. návrh dimenzie,
4. návrh kocky.

Aj keď kocka ako geometrický pojem evokuje predstavu trojdimenzionálneho priestoru, skutočné OLAP kocky majú spravidla viac dimenzií ako tri a naopak naša pravdepodobne najjednoduchšia OLAP kocka akú je vôbec možné vymyslieť má len jednu produktovú dimenziu.

Definovanie údajových zdrojov

Prvým krokom je definovanie zdroja údajov, ktorým je spravidla údajový sklad, prípadne jedna, alebo viacero relačných databáz. V priečinku Data Sources definujeme ako údajový zdroj databázu ISmarket. Využijeme pri tom jednoduchého sprievodcu Data Source Wizard, kde si môžeme vybrať niektoré z už nadefinovaných pripojení na databázy, alebo ako v našom prípade, vytvoríme nové pripojenie na databázu ISmarket.

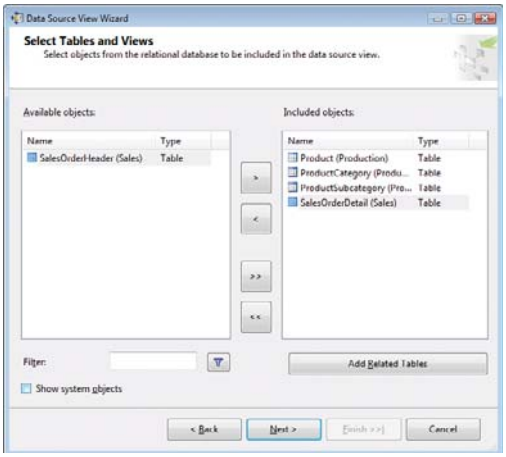


Data Source Wizard – výber zdroja údajov

Definovanie pohľadov na údajové zdroje

Databáza, alebo údajový sklad, ktorý poslúži ako zdroj údajov pre analýzu obsahuje veľa rôznych relačne zviazaných databázových tabuliek. Aj pre túto akciu máme k dispozícii sprievodcu – tentoraz s názvom Data Source View Wizard. V jednom z úvodných dialógov môžeme špecifikovať vytvorenie logických relačných vzťahov na základe určitých pravidiel, napríklad ak sa stĺpec v niektorej tabuľke volá rovnako ako primárny kľúč inej tabuľky a podobne.

V dialógu pre výber tabuliek a pohľadov vyberieme tabuľky: Sales.SalesOrderDetail, Production.Product, Production.ProductSubcategory a Production.ProductCategory – zjednodušene povedané všetky tabuľky, ktoré sme v etape nasadenia integračných služieb preniesli okrem Sales.SalesOrderHeader. Veľmi praktické je tlačidlo Add Related Tables pre pridanie relačne zviazaných tabuliek do pohľadu.



Sprievodca pre definovanie pohľadu na údajové zdroje – výber databázových tabuliek pre OLAP kocku

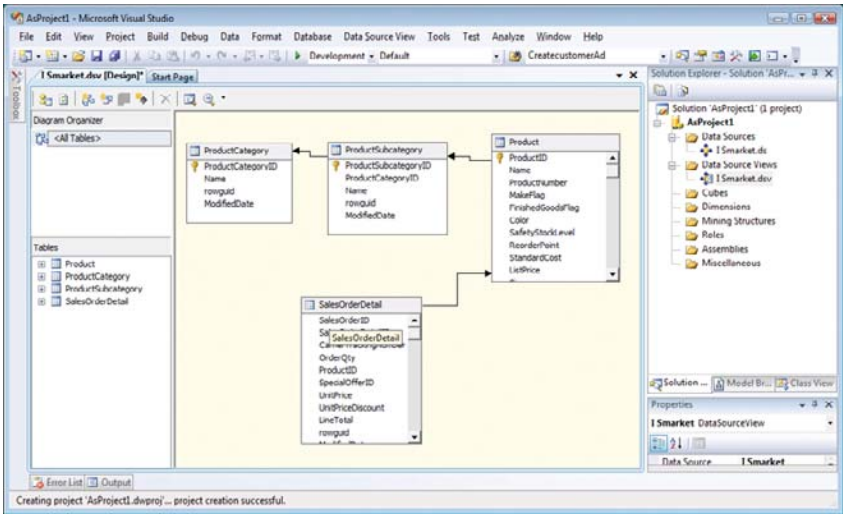
Po výbere tabuliek je potrebné nadefinovať väzby medzi nimi. Navrhujeme usporiadať v zobrazenom diagrame na pracovnej ploche tabuľky zľava doprava takto:

ProductCategory — ProductSubcategory — Product — SalesOrderDetail

Väzby budeme definovať zas opačne sprava doľava (podľa logiky relácií tabuliek) takto:

– medzi tabuľkami Product a SalesOrderDetail bude väzba pomocou kľúča ProductID,
– medzi tabuľkami Product a ProductSubcategory bude väzba pomocou kľúča ProductSubcategoryID,
– medzi tabuľkami ProductCategory a ProductSubcategory bude väzba pomocou kľúča ProductCategoryID.

V tomto procese zároveň nadefinujeme aj primárne kľúče, nakoľko v IS etape sme prenášali len tabuľky bez relačných väzieb. Najnázornejšie to vidíme na obrázku:



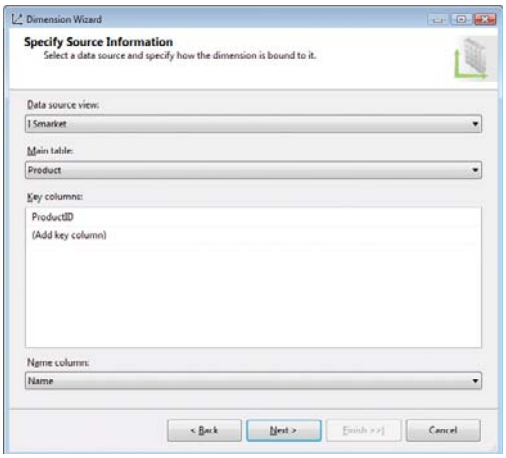
Definovanie relačných vzťahov medzi tabuľkami

Návrh dimenzie
Pre návrh dimenzií pomocou sprievodcu Dimension Wizard môžeme využiť metódu Auto Build, ktorá analyzuje údajové zdroje a navrhuje definície dimenzií, prípadne si dimenzie navrhnúť sami. V cvičnom príklade uprednostníme vlastný návrh dimenzií. Navrhneme produktovú dimenziu **PRODUCT** – ktorá bude mať takúto hierarchickú štruktúru:

- **CATEGORY**,
- • **SUBCATEGORY**,
- • • **PRODUCT**.

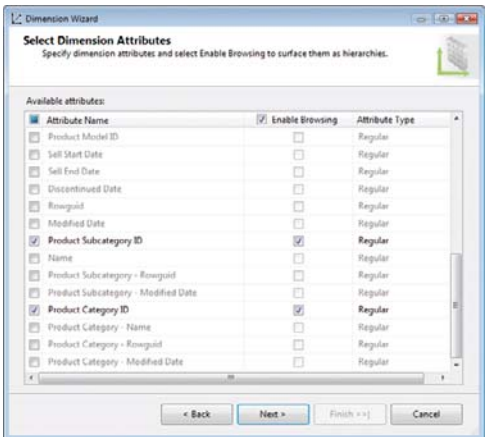
Po výbere metódy zostavenia dimenzie nasleduje výber jej typu. Dialóg sprievodcu ponúka tri typy:
– štandardnú dimenziu,
– časovú dimenziu,
– časovú serverovú dimenziu.

Pre produktovú dimenziu vyberieme typ štandardná dimenzia. Hlavnou tabuľkou dimenzie bude tabuľka Production.Product. Ako kľúčový stĺpec nastavíme Product ID a stĺpec Name ako Name Column.



Výber hlavnej tabuľky dimenzie

Sprievodca v nasledujúcom dialógu podľa nami nadefinovaných cudzích kľúčov sám správne určí relačné tabuľky Subcategory a Category. Ako atribúty vyberieme stĺpce Color, Size, Product Subcategory, a Product Category.



Výber atribútov dimenzie

Pomocou presúvania atribútov dimenzie navrhneme jej hierarchiu:

- CATEGORY
- SUBCATEGORY
- PRODUCT

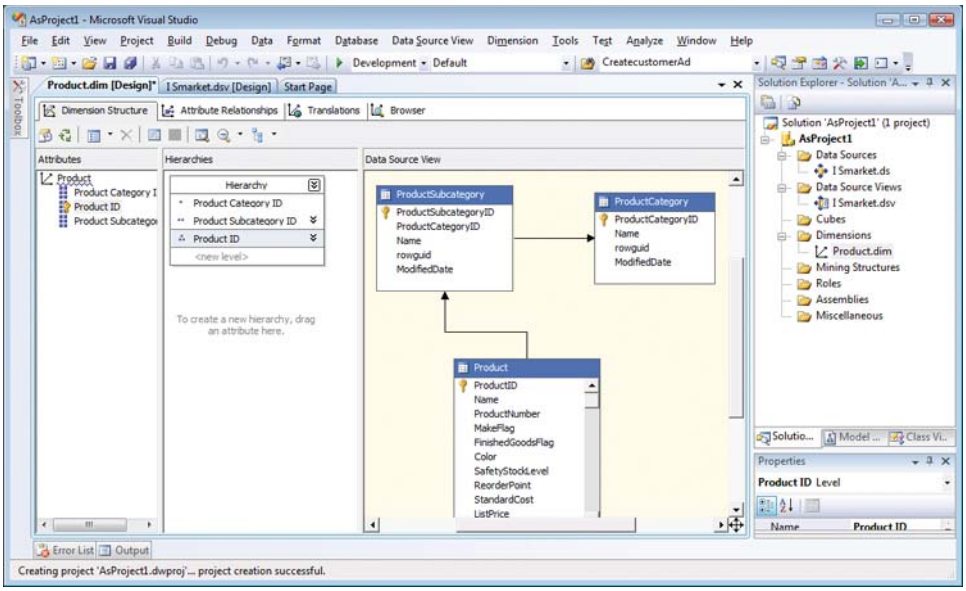
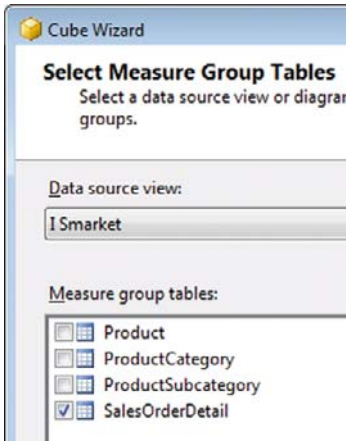


Diagram produktovej dimenzie zobrazený vo vývojovom prostredí

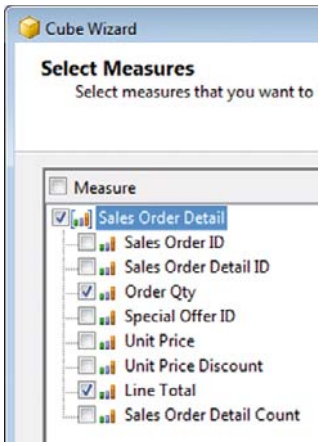
Návrh kocky

Po prípravných prácach spočívajúcich v špecifikovaní údajov, ktoré budú slúžiť ako podklady pre vytvorenie tabuľky faktov a tabuliek dimenzií pristúpime k návrhu OLAP kocky. Využijeme sprievodcu – Cube Wizard. Aktivujeme ho pomocou položky New Cube kontextového menu zložky Cubes v okne nástroja Solution Explorer.Vyberieme tabuľku SalesOrderDetails ako tabuľku faktov (mierok), pretože obsahuje merné jednotky obchodovania.



Výber tabuľky faktov §§

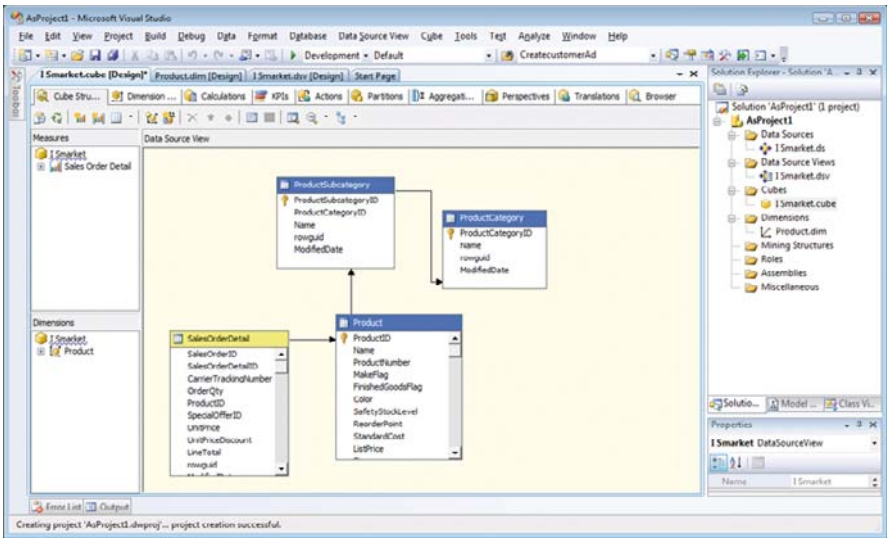
Nasleduje výber mierok (merných jednotiek obchodovania) z tabuľky faktov. Spravidla je to počet predaných kusov, prípadne výška úhrady za tovar a služby a podobne. V tomto príklade budeme pracovať s mierkami:
- Order Qty,
- Line Total.



Výber mierok

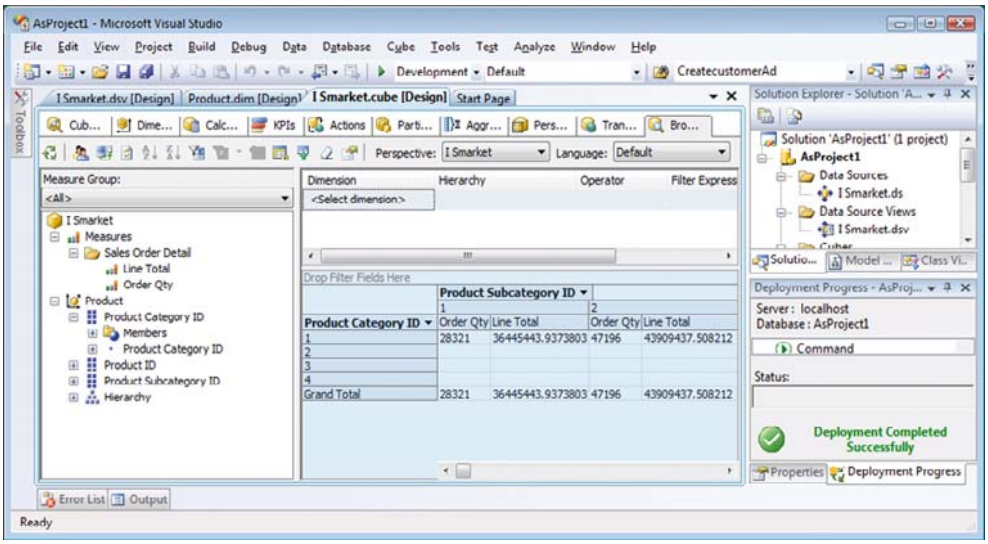
Ako dimenziu vyberieme nami vytvorenú dimenziu Products. Po vytvorení OLAP kocky a napočítaní agregácií sa pracovná plocha nástroja Business Intelligence Development Studio rozdelí do záložiek:

- Cube Builder – záložka pre vytváranie a editovanie mierok,
- Dimension Usage pre definovanie použitia dimenzií v OLAP kockách,
- Calculations – vytváranie a editovanie kalkulácií,
- KPIs – Key Performance Indicators,
- Actions – definovanie akcií pre navrhnutú kocku,
- Partitions – prezeranie a editovanie partícii kocky,
- Perspectives Translations – budovanie a editovanie perspektívnych prehľadov,
- Browser – návrh kontingenčnej tabuľky pre prezeranie údajov.



Hierarchia kocky zobrazená vo vývojovom prostredí

Zelenou šípkou na Toolbare zahájime zostavenie a zavedenie projektu. O priebehu tejto činnosti získame podrobný protokol. Finálnym krokom bude prehliadanie vytvorenej a zostavenej OLAP kocky. V záložke Browser môžeme interaktívne vytvoriť požadovanú kontingenčnú tabuľku. Do obdĺžnika pre polia údajov umiestnime fakty a do obdĺžnikov pre polia riadkov a stĺpcov umiestnime príslušné dimenzie. Z každej dimenzie sa tak stane stáva množina polí, v ktorých možno rozbaliť a zbaliť podrobnosti na jednotlivých úrovniach hierarchií. Tento extrémne jednoduchý príklad OLAP kocky s jednou jedinou dimenziou poslúžil hlavne na zoznámenie sa z návrhovými nástrojmi a postupom modelovania a návrhu.



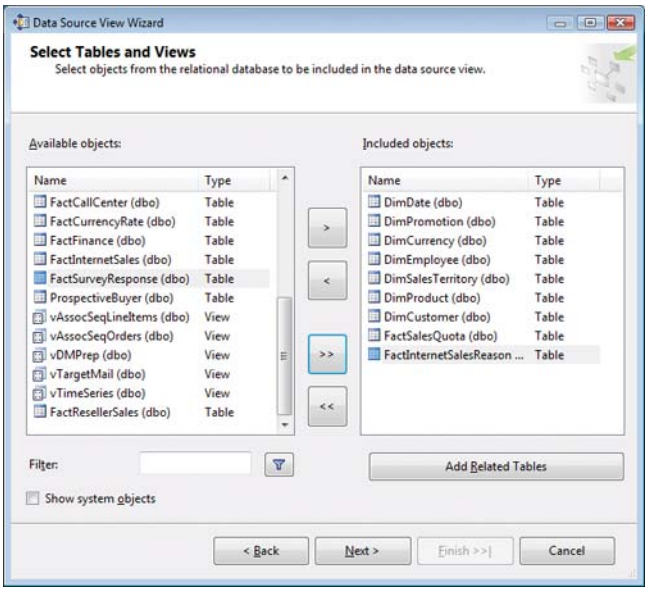
Prezeranie OLAP kocky vo forme kontingenčnej tabuľky

Vytvorenie OLAP kocky z údajového skladu

Predchádzajúci príklad mal k reálnej OLAP kocke asi tak ďaleko ako aplikácia typu „Hello World" k reálnej aplikácii, slúžila skôr ako praktický úvod do problematiky a zoznámenie sa s vývojovým prostredím v režime BI projektov, hlavne s množinou jeho sprievodcov. OLAP kocka vytváraná v tomto prípade sa už bude realite približovať oveľa viac, bude mať viac dimenzií, pričom podobne ako pri kockách v reálnej praxi jedna dimenzia bude časová. Budeme pracovať s cvičnou databázou AdventureWorksDW2008. Písmenká v názve (DW – Data Warehouse) dávajú tušiť, že táto databáza je organizovaná ako typický údajový sklad. V databáze sú tabuľky obsahujúce podklady pre vytvorenie mierok, teda údaje obsahujúce merné jednotky obchodovania (názvy začínajú prefixom Fact) a tabuľky tabuliek dimenzií (začínajú prefixom Dim). Určenie každej tabuľky sa dá jednoznačne identifikovať už z jej názvu. Proste vzorový údajový sklad aký je snom každého IT oddelenia.

Definovanie údajových zdrojov
Ako údajový zdroj definujeme cvičnú databázu AdventureWorksDW.

Definovanie pohľadov na údajové zdroje
V príklade vyberieme tabuľky:
- DimDate,
- DimPromotion,
- DimCurrency,
- DimEmployee,
- DimSalesTerritory,
- DimProduct,
- DimCustomer,
- FactInternetSalesReason,
- FactSalesQuota.



Definovanie pohľadu na údajové zdroje – výber faktov a dimenzií

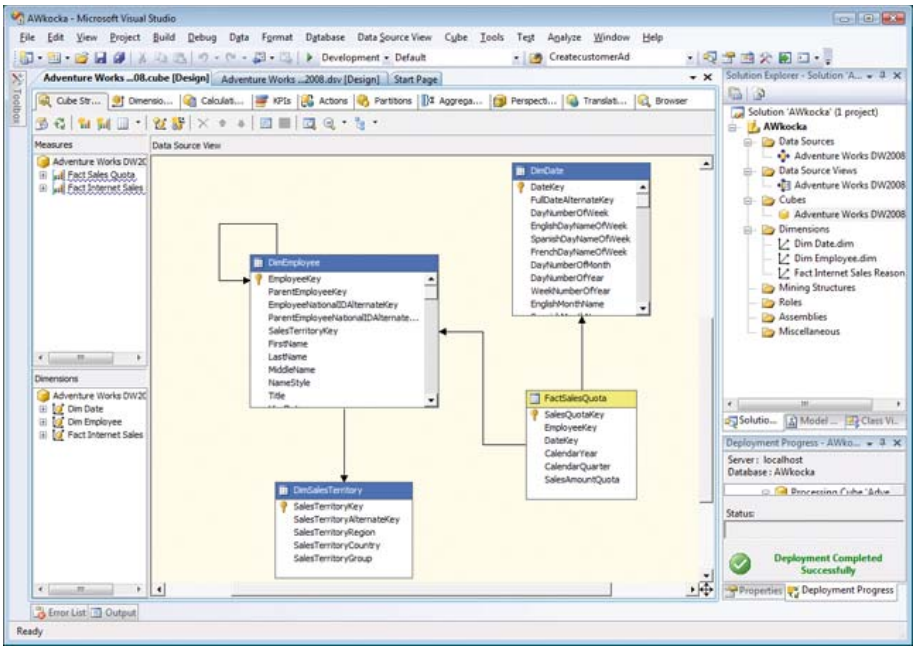
Tabuľky dimenzií môžu okrem stĺpcov tabuľky obsahovať aj vypočítané atribúty.

Návrh kocky
V sprievodcovi Cube Wizard vyberieme tabuľky faktov (začínajú prefixom Fact). Dimenzie budú priradené automaticky.

Po ukončení návrh vykonáme deploy aplikácie pomocou zeleného tlačidla v tvare šípky na toolbare.

Práca s OLAP kockou v prostredí BI Dev Studio
Zatiaľ čo v predchádzajúcom dalo by sa povedať „zoznamovacom" príklade sme kládli dôraz na postup pre modelovanie OLAP štruktúry v tomto príklade to bude naopak. Postup vytvorenia OLAP kocky z efektívne usporiadaného údajového skladu sme zhrnuli do niekoľkých heslovitých bodov a o to väčšiu pozornosť budeme venovať možnostiam práce s OLAP kockou v prostredí nástroja BI Dev Studio. Postupne si prezrieme jednotlivé záložky.

Cube Builder
V záložke je možné prezerať a upravovať model OLAP kocky. V prehľadnom diagrame sú znázornené relačné vzťahy medzi tabuľkami faktov a dimenzií.

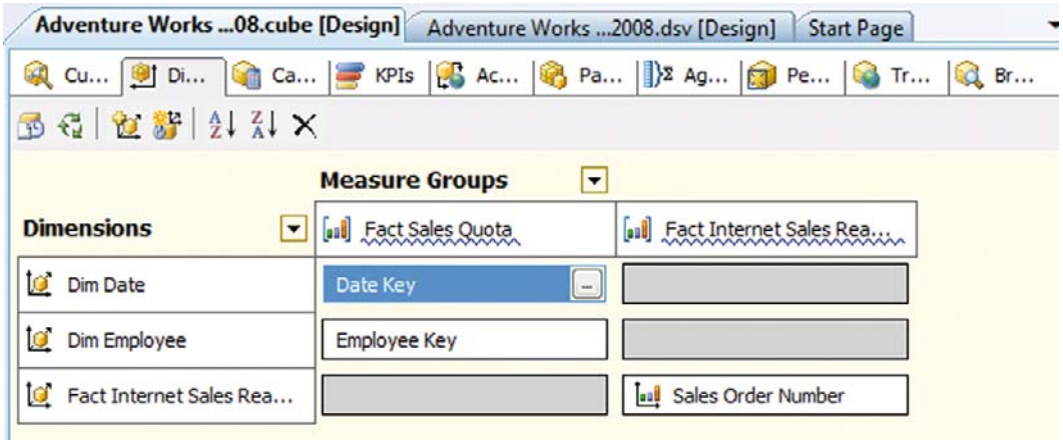


Model OLAP kocky vo vývojovom prostredí

Jednou z užitočných záležitostí, ktoré je možné nastavovať v tomto okne je formát zobrazenia jednotlivých mierok.

Dimension Usage

Záložky slúžia na definovanie použitia dimenzií v OLAP kockách, prípadne editovanie relačných vzťahov medzi tabuľkami faktov a dimenzií. V samostatných dialógoch pre jednotlivé dimenzie, ktoré sa aktivujú tlačidlom (...) je možné okrem typov relačných vzťahov (Regular, Fact, Referenced, Many-to-many) meniť aj granularitu jednotlivých atribútov.



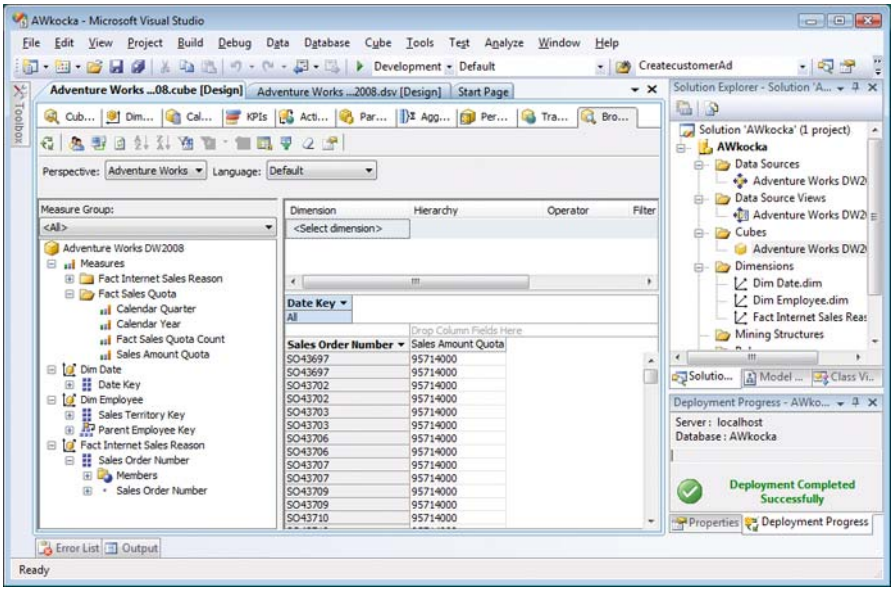
Záložka Dimension Usage

Význam ostatných záložiek uvedieme vzhľadom na priestor v tejto publikácii len heslovito, odporúčame ich podrobne preskúmať a vyskúšať si možnosti, ktoré ponúkajú.

Calculations – vytváranie a editovanie kalkulácií
KPIs – Key Performance Indicators
Actions – definovanie akcií pre navrhnutú kocku
Partitions – prezeranie a editovanie partícii kocky
Perspectives Translations – budovanie a editovanie perspektívnych prehľadov

Browser
Po úspešnom zostavení OLAP kocky a jej zavedení na analytický server nastala prvá príležitosť na prehliadnutie a skontrolovanie výsledkov analýzy. Nie je náhoda, že rozmiestnenie pracovného okna je veľmi podobné ako pri návrhu kontingenčnej tabuľky pre prezeranie údajov v programe Excel. Vývojové prostredie totiž využíva komponent Office Web Components. Okrem plochy v strede tabuľky pre údaje, kam umiestnime mierky sú v návrhovom liste kontingenčnej tabuľky ešte polia:
– riadkové polia,
– stĺpcové polia,
– polia filtrov,
kam spravidla umiestňujeme dimenzie.

Ako prvý krok pre zamýšľanú zostavu prenesieme symbol mierky Sales Amount z ľavého okna Data Source View do vnútra kontingenčnej tabuľky. Do riadkových polí presunieme symbol Product line (atribút produktovej dimenzie). Do stĺpcového poľa prenesieme celú hierarchiu dimenzie OrderDate (symbol OrderData.Fiscal Quarter).



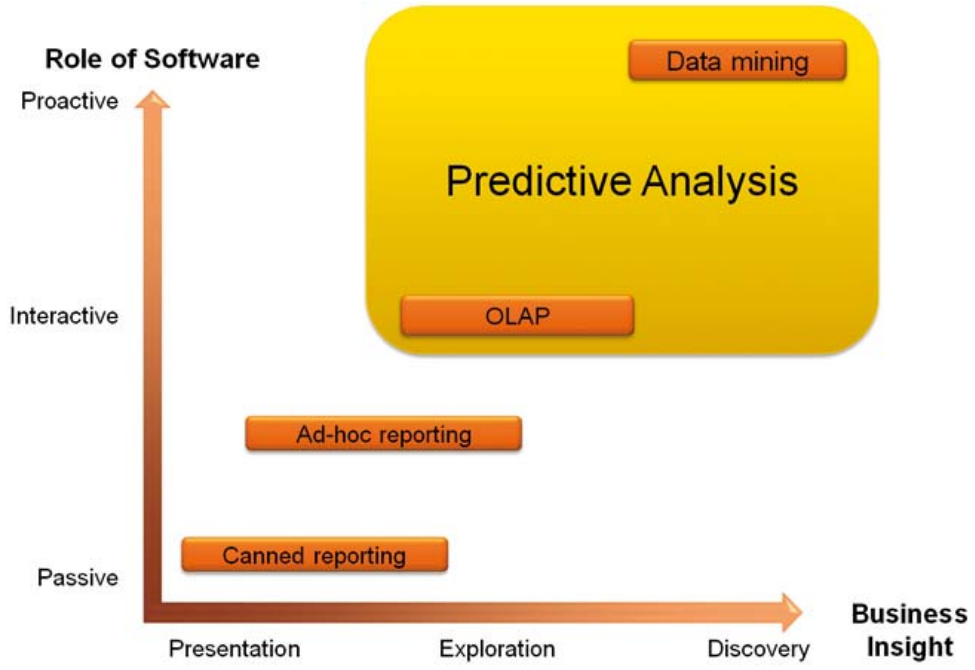
Prehliadanie údajov OLAP kocky v okne Browser

Ak nie sme spokojní s formátom zobrazovania, tento môžeme zmeniť v záložke Cube Structure. Ak v ktorejkoľvek záložke a v okne Solution Explorer označíme ľubovoľnú dimenziu, môžeme si prehliadať jej štruktúru aj hierarchiu.Tu môžeme meniť úrovne hierarchie, prípadne atribúty na jednotlivých úrovniach. Stačí interaktívne presúvať atribúty medzi oknami Hierarchies and Levels a Attributes.

Kapitola 5: Dolovanie údajov – data mining

Data mining je momentálne jednoznačne najrýchlejšie rastúci segment Business Intelligence, nakoľko dokáže vykonávať sofistikované prediktívne analýzy. Je to prienik umelej inteligencie do oblasti databáz. S trochou nadsádzky by sa dalo povedať, že ide o odvetvie niekde na rozhraní vedy a mágie. Ako vyplýva z názvu, ide v princípe o ťažbu údajov, presnejšie o proces analýzy údajov z rôznych perspektív a ich premenu na užitočné informácie. Z matematického a štatistického hľadiska ide o hľadanie korelácií, teda vzájomných vzťahov alebo vzorov v údajoch v kombinácii so štatistikou. V niektorých prípadoch býva technológia dolovania údajov označovaná aj skratkou KDD (Knowledge discovery in databases).

Databázy v dnešnej informatickej spoločnosti skrývajú nesmierne informačné bohatstvo, problémom však je ako ho odkryť alebo expresívnejšie povedané „vydolovať". Databázové tabuľky obsahujú milióny až miliardy záznamov, ktoré sú rôzne usporiadané a členené. Data mining umožňuje v týchto údajoch vyhľadávať vzory informácií. Na základe nazbieraných údajov sa pokúša zisťovať (odkrývať) rôzne závislosti, ktoré tieto údaje v sebe skrývajú. Takto vytvorené znalosti potom využíva pri prediktívnej analýze. V kombinácii s OLAP analýzou dokážeme riešiť aj veľmi náročné typy úloh.



Data mining umožňuje vykonávať prediktívnu analýzu

Je založený na heuristických algoritmoch, umelej inteligencii, neurónových sieťach a iných pokročilých databázových softvérových technológiách a metódach umelej inteligencie. Technológie pre dolovanie údajov pomáhajú jednak sledovať a analyzovať trendy a taktiež predvídať udalosti. V ekonomickej oblasti sa data mining využíva napríklad pri analýze a predikcii úverového rizika, predikcii rizika pri vydávaní kreditných kariet, odhaľovaní bankových podvodov, predikcii kurzu meny. V poisťovníctve na odhaľovanie poistných podvodov v oblasti likvidácie škôd, ale aj pri získavaní klientov.

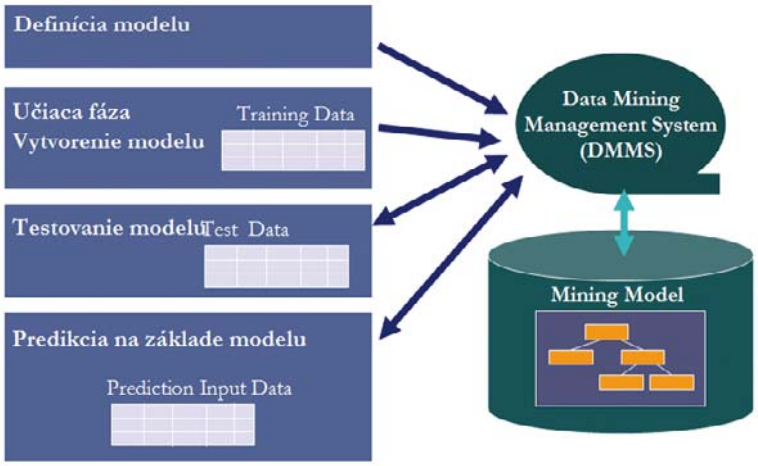
V podnikovej informatike sa využíva najmä v oblasti marketingu (Direct marketing), v oblasti styku so zákazníkmi (CRM) ale aj oblasti kontroly kvality výrobkov (Quality Control/Quality Management). Používa sa na vytipovanie vhodných zákazníkov pre marketingovú kampaň a podobne. Pomocou dataminingových modelov dokážeme z histórie správania klientov, ktoré máme uložené vo forme údajov v databáze predpovedať správanie konkrétneho klienta v danej situácii. Napríklad je možné identifikovať, kedy klient zvažuje odísť ku konkurencii, odhaliť pokus o bankový podvod, či ohraničiť cieľovú skupinu pre ponuku konkrétneho produktu alebo služieb.

Čoraz častejšie sa však výhody tejto technológie využívajú aj v iných oblastiach, napríklad pri analýze laboratórnych vzoriek, v medicíne pri vyhľadávaní skupín príznakov, vyhľadávaní skupín liekov, vyhľadávanie skrytých vzťahov medzi jednotlivými diagnózami, v energetike na predikciu zaťaženia distribučnej siete alebo predikciu odberu elektriny a podobne.

Procesná schéma data miningu

Proces data miningu je možné rozčleniť do troch etáp:

- výber algoritmu a modelu,
- učiaca fáza aplikovaná na doteraz existujúcich prípadoch,
- analýza a predikcia nových prípadov.



Procesná schéma data miningu

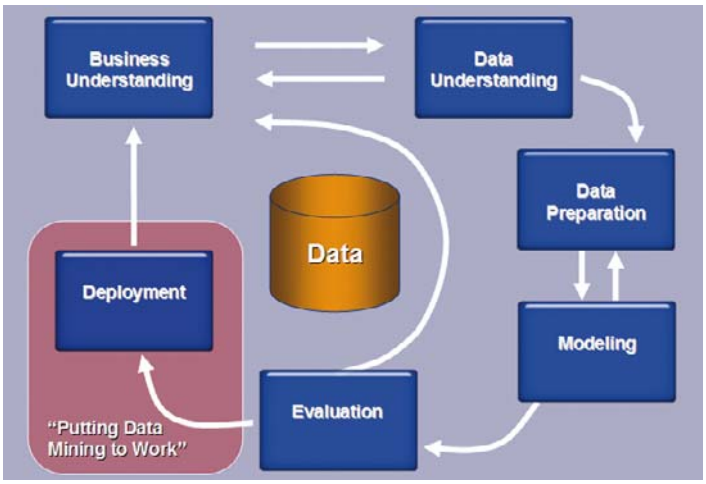
Alebo inými slovami, na základe prieskumu údajov sa v nich vyhľadajú určité vzory a na základe týchto vzorov sa vypracujú modely pre predikciu budúcich prípadov. Pochopiteľne vzory sa podarí odhaliť len vtedy, ak nejaké existujú. Preto napríklad nemá význam aplikovať data mining nad rôznymi cvičnými databázami obsahujúcimi náhodne vygenerované údaje.

Učiaca fáza

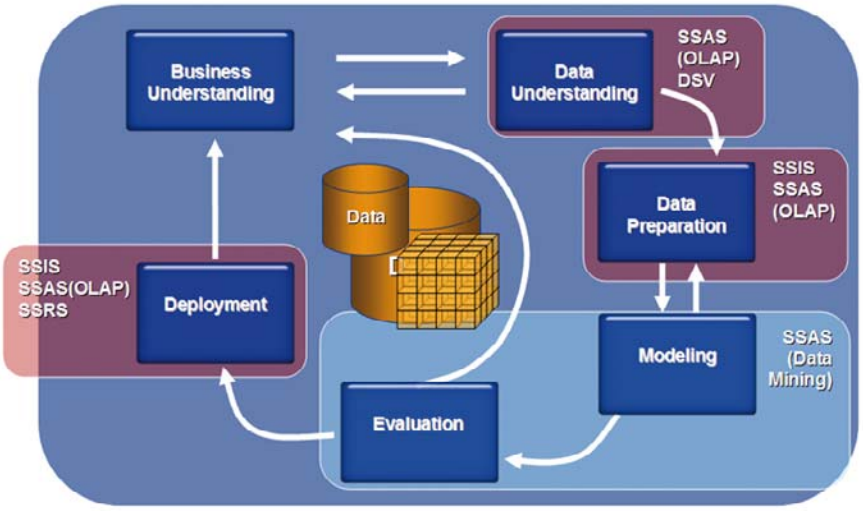
V tejto fáze sa vybraný data miningový model učí a spresňuje svoje parametre na množine údajov získaných z doteraz existujúcich prípadov. Ako podklad pre učiacu fázu slúžia doteraz zozbierané (namerané) a vyhodnotené údaje. Výberu a príprave údajov, ktoré budú slúžiť ako podklad pre učiacu fázu musíme venovať veľkú pozornosť, aby boli dostatočne reprezentatívne a očistené od prípadných chýb. Najvýhodnejšie je pripraviť a predspracovať tieto údaje v procese ETL. Vidíme, že učiaca fáza je veľmi náročná na výber správneho modelu a taktiež na výber a predspracovanie údajov.

Analýza a predikcia nových prípadov

V etape analýzy a predikcie aplikujeme už naučený data miningový model na množinu vstupných údajov, z ktorých potrebujeme získať súvislosti. Na rozdiel od učiacej fázy je táto fáza skôr náročná na výpočtovú kapacitu použitého hardvéru a softvéru.



Procesná schéma data miningu údajov z relačnej databázy



Procesná schéma data miningu údajov z OLAP kocky

Algoritmy pre data mining

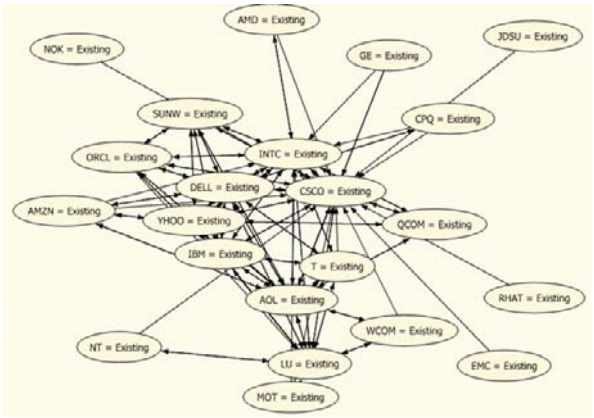
Data mining je proces analýzy údajov z rôznych perspektív a ich premena na užitočné informácie. Z matematického a štatistického hľadiska ide o hľadanie korelácií, teda vzájomných vzťahov alebo vzorov v údajoch. SQL Server 2008 má implementované tieto algoritmy:

- asociačné pravidlá,
- nevyvážené rozpadové stromy,
- zhlukovanie (clustering),
- naive Bayes,
- neuronové siete,
- sekvenčné zhlukovanie,
- časové série.

Jednoznačné odporúčanie pre výber najvodnejšieho algoritmu pre tú ktorú úlohu neexistuje, vždy to závisí od každého konkrétneho prípadu. Pre uľahčenie orientácie najskôr popíšeme jednotlivé algoritmy a potom sa pokúsime priradiť vhodné algoritmy k jednotlivým typom úloh. Pri popise algoritmov uvádzame aj zoznam parametrov, pomocou ktorých špecifikujeme úlohu, ktorú chceme pomocou algoritmu riešiť.

Asociačné pravidlá

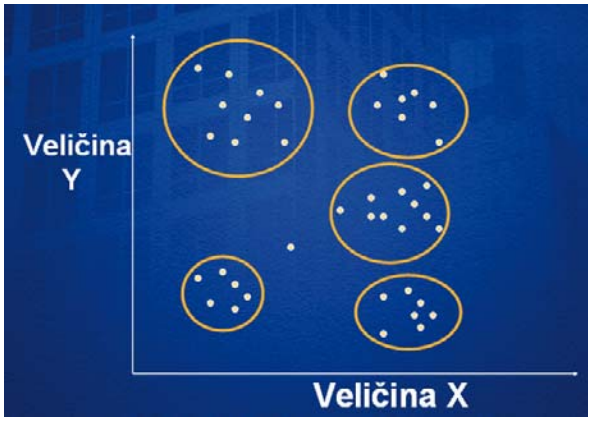
Analýza z hľadiska asociačných pravidiel je zameraná na odhaľovanie rôznych súvislostí priraďovania. Najčastejšie sa pravdepodobne používa pri hľadaní odpovede na otázku, aké tovary si zákazníci kupujú najčastejšie spolu. Poznanie týchto súvislostí umožní efektívnejšie rozmiestnenie tovarov v supermarketoch, prípadne zostavovanie akciových ponúk. Z hľadiska matematickej štatistiky ide o skúmanie korelácie, či už pozitívnej, alebo negatívnej. Pozitívna korelácia udáva, že vysoká úroveň jednej premennej bude sprevádzaná vysokou úrovňou korelačnej premennej. Naopak negatívna korelácia udáva, že vysoká úroveň jednej premennej bude sprevádzaná nízkou úrovňou korelačnej premennej. Poznanie pozitívnej korelácie je dôležité napríklad pri rozmiestňovaní tovaru alebo pri marketingových rozhodnutiach, ktoré tovary je možné predávať spolu, prípadne čo je potrebné ktorému zákazníkovi ponúknuť. Význam však má aj negatívna korelácia, napríklad pri výmene skladby firemnej produkcie. Môžeme to demonštrovať napríklad záujmom výrobcov klasických fotoaparátov o fotoaparáty digitálne.



Asociačné pravidlá

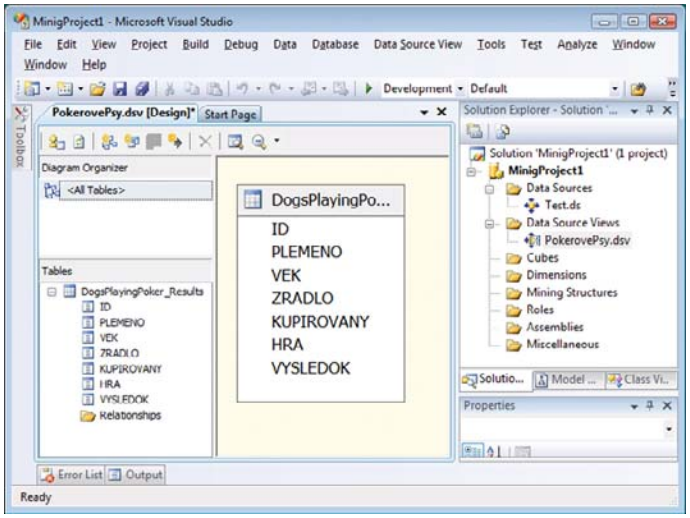
Viacrozmerné zhlukové diagramy

Tento algoritmus sa používa na odhaľovanie zhlukov údajov v multidimenzionálnych priestoroch. Pomocou neho môžeme rozdeliť množiny prípadov na čo najohraničenejšie skupiny, takzvané „ostrovy podobnosti". Vhodný je napríklad na identifikáciu zákazníckych segmentov, ktoré sú založené na spoločných charakteristikách, napríklad demografických, sociálnych, profesijných a podobne.



Odhaľovanie zhlukov

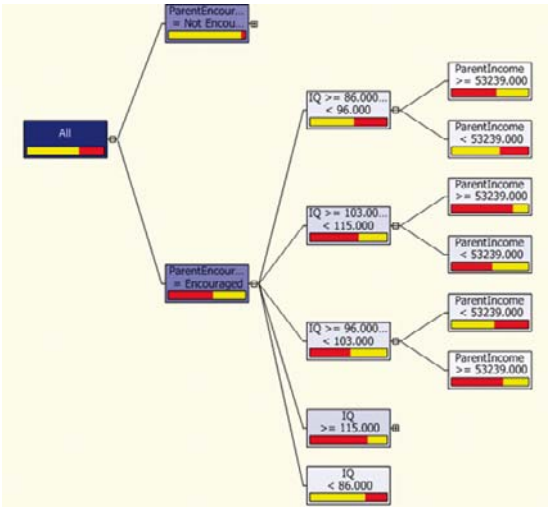
Zhluky sú odhaľované na základe aplikovania a analýzy kriviek pravdepodobnosti, pričom sa skúma, či jednotlivé údaje, alebo skupiny údajov nespĺňajú podmienku napríklad Gaussovho normálneho rozdelenia a podobne.



Princíp vyhľadávania segmentov

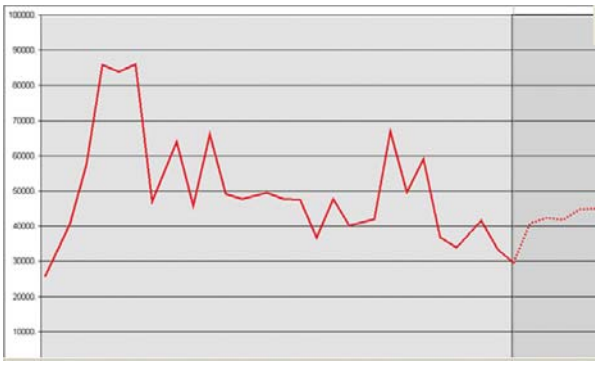
Sekvenčné zhlukovanie
Špecifickým prípadom zhlukovania je sekvenčné zhlukovanie. Z matematického hľadiska ide o aplikáciu Markových procesov a modelov.

Nevyvážené rozhodovacie stromy
Tento typ diagramov odhaľuje závislosti a vyhľadáva špecifické vlastnosti, ktoré potom poslúžia na zostavenie predikčného modelu rozhodovania na jednotlivých úrovniach hierarchickej štruktúry. Algoritmus podporuje predikciu diskrétnych aj spojitých atribútov.



Nevyvážené rozpadové stromy

Časové rady
Algoritmus analýzy časových radov sa spravidla vzťahuje k nejakej premennej, napríklad obratu, zisku, nákladom a podobne. Na základe analýzy údajov z minulosti je možné definovať určité pravidlá, pomocou ktorých potom predpovedáme budúci trend danej premennej. V princípe ide o regresiu časových úsekov, takže predpovede trendov v sebe zahrňujú aj krátkodobé cyklické fluktuácie, napríklad ak predikujeme trend vývoja nezamestnanosti pre budúci rok, na základe údajov z minulých rokov, určite sa v dobrej predpovedi objavia aj logické fluktuácie, napríklad nárast nezamestnanosti po ukončení školskej dochádzky, jej pokles pri sezónnych prácach a podobne.



Časové rady

Neurónové siete Spracovanie pomocou neurónových sietí nevychádza zo žiadneho štatistického rozdelenia ale pracuje, podobne ako ľudský mozog na princípe rozpoznávania vzorov a minimalizácie chýb. Tento proces si môžeme predstaviť ako prijímanie informácií a poučenie sa z každej skúsenosti. Neurónovú sieť tvoria uzly usporiadané do vrstiev. Predtým, než sa začne vlastný proces, údaje sa rozdelia do tréningovej a testovacej množiny. Počas každej iterácie sú vstupy spracovávané systémom a sú porovnávané so skutočnou hodnotou. Zmeria sa chyba a odovzdá sa na spracovanie systému, aby upravil pôvodné váhy. Proces sa končí spravidla v okamihu dosiahnutia dopredu určenej minimálnej chyby. Obrázok ilustruje jednoduchú neurónovú sieť z jednou skrytou vrstvou.

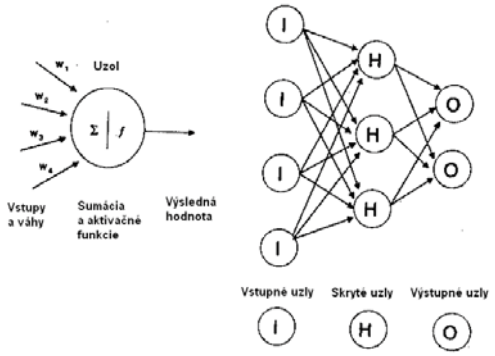


Schéma neurónovej siete

Naïve Bayes
Tento veľmi rýchly a pritom pomerne presný algoritmus je založený na Bayesovej vete, pomocou ktorej je možné pracovať priamo s pravdepodobnosťami.

Pravdepodobnosť výskytu náhodnej udalosti A za podmienky, že nastala náhodná udalosť B sa rovná podielu pravdepodobnosti prieniku náhodných udalostí A a B a pravdepodobnosti B.

Teda platí: $P(A \cap B) = P(A|B) * P(A)$
$P(B \cap A) = P(B|A) * P(B)$
$P(A \cap B) = P(B \cap A) \Rightarrow P(A|B) * P(A) = P(B|A) * P(B)$

Bayesovu vetu potom môžeme zapísať v tvare:

$P(A|B) * P(B) = P(B|A) * P(A)$

Tento algoritmus sa používa pre zložitejšie analýzy, napríklad pre výpočet paralelnej korelácie a podobne. Algoritmus podporuje predikciu len diskrétnych alebo diskretizovaných atribútov.

Typické okruhy úloh a výber algoritmov pre ich riešenie

Výber algoritmov pre riešenie typických okruhov úloh nie je jednoduchý a už zďaleka nie jednoznačný. Vymenovanie a stručný popis algoritmov nám nie vždy pomôže pri základnej orientácii, ktorý algoritmus sa hodí na aký typ úloh. Možno bude užitočnejší pohľad z druhej strany – vymenujeme typické úlohy a k nim priradíme vhodné algoritmy. Nesmie vás prekvapiť, že na väčšinu typov úloh je možné použiť viac druhov algoritmov.

Klasifikácia
Klasifikácia zaraďuje objekty do kategórií na základe cieľovej premennej. Každý objekt je charakterizovaný množinou premenných, z ktorých je jedna cieľová premenná. Cieľom je nájsť model, ktorý opisuje cieľovú premennú ako funkciu vstupných premenných (prediktorov). Trénovanie klasifikačného modelu vyžaduje znalosť hodnôt cieľovej premennej a prediktorov, typicky sú to historické dáta. Do tejto skupiny úloh patrí hľadanie odpovedí na otázky typu „Aké typy kampaní a členských kariet má ponúkať moja obchodná firma?" Aký okruh záujemcov osloví reklamná a marketingová kampaň? Prečo strácame zákazníkov? Bude zamýšľaný produkt úspešný?

Regresia
Na rozdiel od klasifikácie je pri regresii cieľová premenná číselná. Zo štatistického hľadiska ide o metódu, ktorá kvantifikuje závislosť medzi dvoma spojitými premennými: závislou premennou, ktorú sa snažíme predikovať a nezávislou teda prediktívnou premennou. Hľadáme priamku prechádzajúcu medzi jednotlivými bodmi, pre ktorú platí, že súčet druhých mocnín odchýlok od každého bodu je minimálna. Ak je vzťah medzi závislou a nezávislou premennou nelineárny, spravidla je potrebné transformovať prediktívnu premennú tak, aby umožnila nájsť lepšie preloženie. Logistická regresia je veľmi podobná lineárnej regresii, s tým rozdielom, že závislá premenná nie je spojitá, ale diskrétna. Preto ju transformujeme na spojitú hodnotu, ktorá je funkciou pravdepodobnosti výskytu udalosti. Regresiu je možné použiť na predikovanie výsledkov dvoch, alebo viacerých úrovní, napríklad prečo nastane jav nesplácania pôžičiek, akú tržbu môžem očakávať od ktorého zákazníka a podobne. Regresný model môže predpovedať napr. rýchlosť vetra v závislosti od teploty, tlaku a vlhkosti alebo cenu bytu v závislosti od plochy, lokality, vybavenia a pod.

Zhlukovanie (segmentácia)
Zhlukovanie sa používa na identifikáciu prirodzených skupín objektov charakterizovaných určitými premennými. Objekty vnútri zhluku sú si podľa posudzovaných premenných podobné, medzi zhlukmi sú rôzne. Rozdelenie nejakej množiny entít, napríklad zákazníkov do rôznych skupín a segmentov podľa určitých kritérií umožňuje dosiahnuť cielený a personalizovaný prístup ku každému segmentu. Iným okruhom využitia segmentácie je organizácia údajov, aby boli čo najefektívnejšie použiteľné.

Priraďovanie (asociácia)
Typickým asociačným problémom je analýza nákupného koša, teda identifikácia produktov, ktoré sa často predávajú spoločne. Cieľom je odhalenie asociačných pravidiel (ak zákazník kúpi A a B, potom s pravdepodobnosťou 60 % kúpi aj C). Získané pravidlá možno využiť viacerými spôsobmi: na krížový marketing, cielenú ponuku, optimalizáciu katalógu a štruktúry obchodu, ale aj na segmentáciu zákazníkov s rovnakým správaním sa pri nakupovaní určenom asociačnými pravidlami.

Prognózovanie
Prognózovanie odpovedá napr. na otázky: Aký bude obrat podniku o mesiac? Aké budú ceny akcií o týždeň? Vstupné údaje predstavujú časový rad – postupnosť čísel, zaznamenaných v čase, ktorých hodnoty sú vzájomne závislé. Prognózovanie sa zaoberá trendmi, cyklami a filtrovaním šumu.

Analýza sekvencií
Analýza sekvencií sa používa na nájdenie vzorov v radoch pozostávajúcich z diskrétnych hodnôt (alebo stavov). Napr. sekvencia webových kliknutí pozostáva z radu webových adries. Sekvencia je podobná časovému radu s tým rozdielom, že sekvencia obsahuje diskrétne stavy a časový rad obsahuje spojité čísla.

Analýza odchýlok
Umožňuje nájdenie veľmi zriedkavých objektov, ktoré sú veľmi odlišné od ostatných. Typicky sa používa pri odhaľovaní podvodov a detekciu výrobných chýb. Napr. v bankovníctve sa bez vopred formulovanej definície podozrivej operácie z miliónov operácií klientov identifikuje pár desiatok takých, ktoré sa od zvyšných (zoskupených do niekoľkých zhlukov) pri použití viacerých premenných (napr. obrat, typ operácie, konštantný symbol, čas od zadania po jej splatnosť atď.) výrazne odlišujú.

Analýzy textu
Textové záznamy sú typickým príkladom neštruktúrovaných údajov, aj keď v nich určité pravidlá dokážeme identifikovať týkajú sa skôr syntaxe a sémantiky prirodzeného jazyka (čeština, angličtina...) v ktorom bol text vytvorený. Niektoré heuristické data miningové algoritmy si však poradia aj s bežným neštruktúrovaným textom.

Úvodný príklad pre data mining

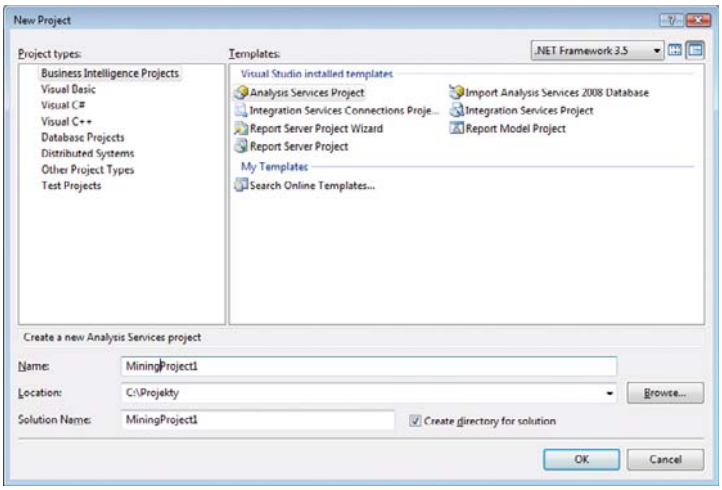
Aj napriek tomu, že data mining sa oprávnene považuje za vrchol databázových a analytických metód, jeho zvládnutie vďaka unifikácii na úrovni UDM modelovania nie je príliš zložité, preto vás povzbudzujeme urobiť nasledujúci príklad aj v prípade, že problematike data miningu zatiaľ príliš nerozumiete. Príklad je možné urobiť takmer mechanicky podľa predpísaných krokov, a je takmer isté že na jeho konci budete mať o data miningu a jeho možnostiach oveľa jasnejšiu predstavu.

Námetom pre prvý zoznamovací príklad z oblasti data miningu bude trochu žartovná (ale veľmi jednoduchá) úloha na báze tabuľky o tom, ako jednotlivé plemená psov hrajú poker.

```
SELECT [ID],[PLEMENO],[VEK],[ZRADLO],[KUPIROVANY],[HRA],[VYSLEDOK]
  FROM [pokusy].[dbo].[DogsPlayingPoker]
```

```
ID   PLEMENO                        VEK   ZRADLO          KUPIR HRA              VYSLEDOK
---- ------------------------------ ----- --------------- ----- ---------------  --------
1    Affenpinscher                  2     CANNED          YES   5 CARD DRAW      1
2    Afghan Hound                   7     STORE BRAND     NO    5 CARD STUD      0
3    Airedale Terriers              14    HIGH END        YES   4 CARD GUTS      0
4    Akbash Dogs                    8     TABLE SCRAPS    NO    TEXAS HOLD'EM    0
5    Akitas                         6     CANNED          YES   3 CARD MONTE     1
6    Alapha Blue Blood Bulldogs     1     STORE BRAND     NO    BLACK JACK       0
7    Alaskan Klee Kai               9     HIGH END        YES   7 CARD STUD      0
8    Alaskan Malamutes              13    TABLE SCRAPS    NO    OMAHA            0
...
5907 rows
```

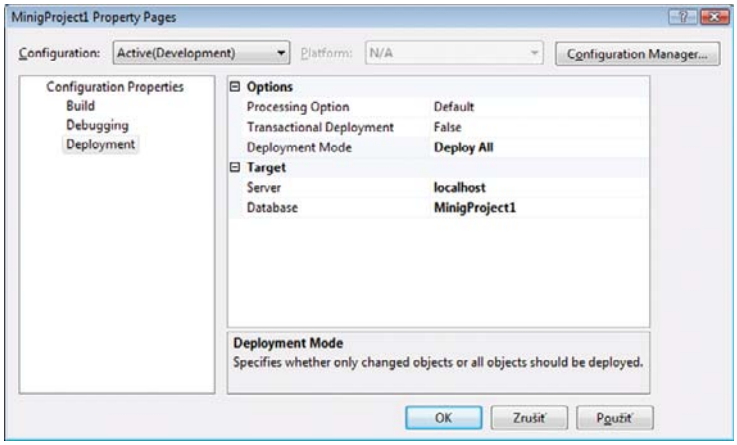
V prostredí Business Intelligence Development Studio vytvoríme nový projekt typu Analysis Services Project z priečinka BI projektov. V našom prípade sme zvolili názov MiningProject1.



Vytvorenie projektu typu „Analysis Services Project" pre data mining

Všimnite si, že v zhode s filozofiou UDM vytvárame rovnaký typ analytického projektu ako pri vytváraní OLAP kociek. Rozdiel bude len v niektorých krokoch návrhu, ktoré sú dané záložkami. Vynecháme záložku Cubes a namiesto toho budeme vytvárať data miningový model. Vytváranie údajových zdrojov a pohľadov na údajové zdroje zostane nezmenené.

Pre zrýchlenie deploymentu projektu na analytický server a zrýchlenie ladenia odporúčame metódu deploymentu. V okne nástroja Solutions Explorer cez kontextové menu aktivujeme dialóg Properties a v ňom priečinok Deployment. V ňom nastavíme Deployment Mode na hodnotu Deploy All.



Nastavenie Deployment Mode

Podobne ako pri OLAP analýzach, aj v tomto prípade je postup budovania projektu naznačený poradím priečinkov v okne vývojového prostredia Solution Explorer v pravom hornom rohu. Pracovať budeme hlavne zo záložkami:

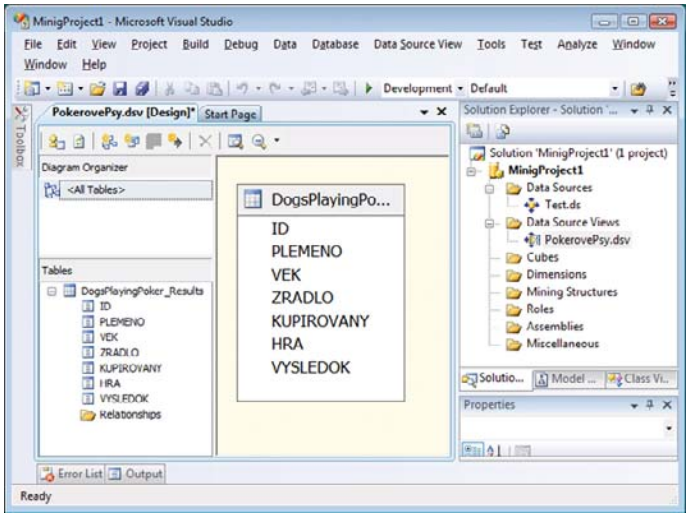
- Data Sources,
- Data Source Views,
- Mining Structures.

Data Sources – definovanie údajových zdrojov

Ako údajový zdroj poslúži tabuľka `DogsPlayingPoker`. Je v tej databáze, do ktorej sme ju umiestnili v cvičnom príklade z kapitoly „Integračné služby". V našom prípade je táto tabuľka v databáze Test.

Definovanie pohľadov na údajové zdroje

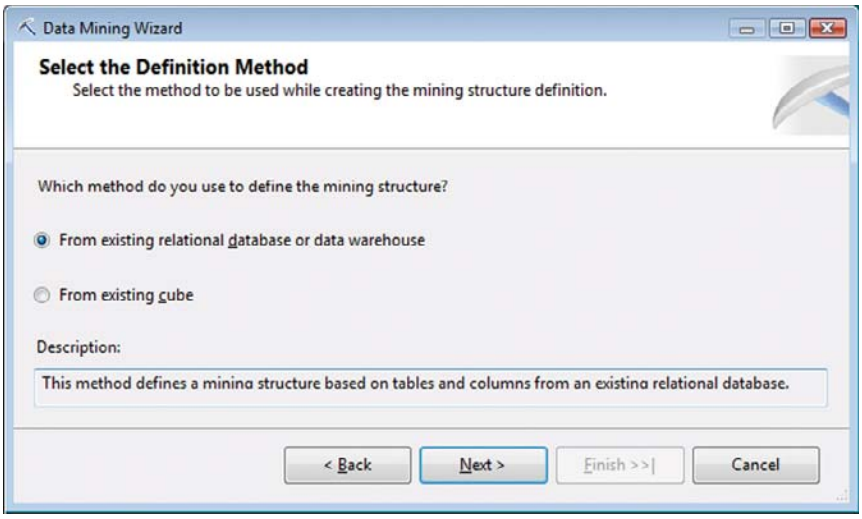
V tejto záložke vyberieme tabuľku DogsPlayingPoker. Pohľad môžeme nazvať napríklad PokerovePsy. V strednom okne vývojového prostredia sa zobrazí diagram návrhovej štruktúry vybraného pohľadu na údajové zdroje. Pomocou položky „Explore Data" kontextového menu si môžeme prezrieť vybrané údaje.



Definovanie pohľadov na údajové zdroje

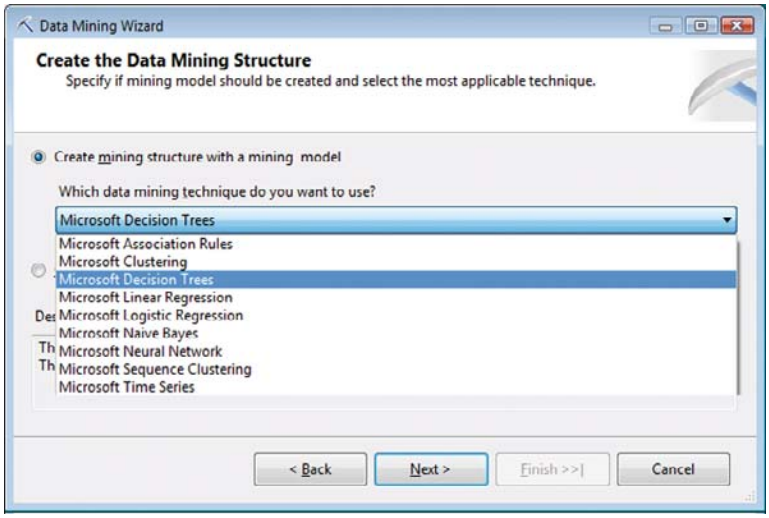
Návrh data miningového modelu

Po prípravných fázach definovania zdroja údajov pristúpime k vytvoreniu data miningového modelu. V priečinku Mining Sctructures aktivujeme sprievodcu Data Mining Wizard. Tento sprievodca umožní vytvoriť data miningový model buď z relačnej databázy, alebo z OLAP kocky. V tomto cvičnom príklade budeme model vytvárať z relačnej databázy. V dialógu pre výber štruktúry údajov, ktorú plánujeme podrobiť data miningu označíme voľbu From existing relational database or data warehouse.



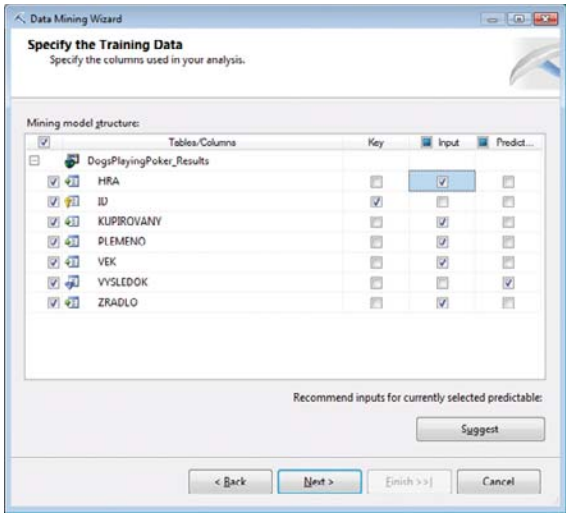
Data miningový model môžeme vytvárať z existujúcej databázy alebo údajového skladu, prípadne z existujúcej OLAP kocky

Nasleduje výber algoritmu. Vyskúšame nevyvážené rozhodovacie stromy (Microsoft Decision Tree).



Výber algoritmu pre data mining

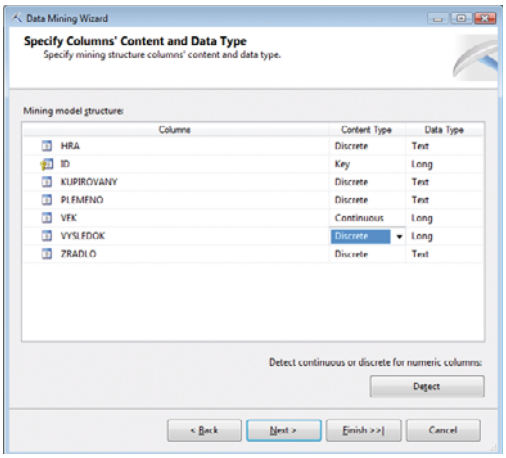
Skôr než začneme upresňovať údaje pre data miningový model, je potrebné určiť tabuľku s ktorou bude model pracovať, alebo laicky povedané – na ktorej sa bude učiť. V našom prípade to bude tabuľka DogsPlayingPoker. Pre identifikáciu prípadov potrebujeme definovať stĺpec (povedané databázovou terminológiou primárny kľúč), pomocou ktorého bude jednoznačne identifikovaná každá entita. V našom prípade je to jednoznačne ID, nakoľko tento stĺpec je de facto aj primárnym kľúčom zdrojovej tabuľky. Nasleduje špecifikácia vstupných stĺpcov a stĺpca ktorého hodnotu chceme predikovať. Predmetom nášho záujmu je, či ten ktorý pes vyhrá príslušnú hru, takže ako predikovateľný stĺpec označíme VYSLEDOK. Ostatné stĺpce označíme ako vstupné.



Výber predikovaného atribútu a atribútov na základe ktorých budeme predikovať

Pri kvalifikovanom výbere vstupných stĺpcov nám môže pomôcť odporučenie sprievodcu, ktoré je dostupné tlačidlom Suggest.

Po výbere vstupných stĺpcov ešte môžeme špecifikovať, či obsahujú diskrétne, alebo spojité hodnoty. Niektoré hodnoty sú typicky spojité, napríklad vek (čas plynie spojito), sumy obeživa a podobne, iné stĺpce obsahujú diskrétne údaje. Typickou diskrétnou hodnotou je napríklad počet detí.



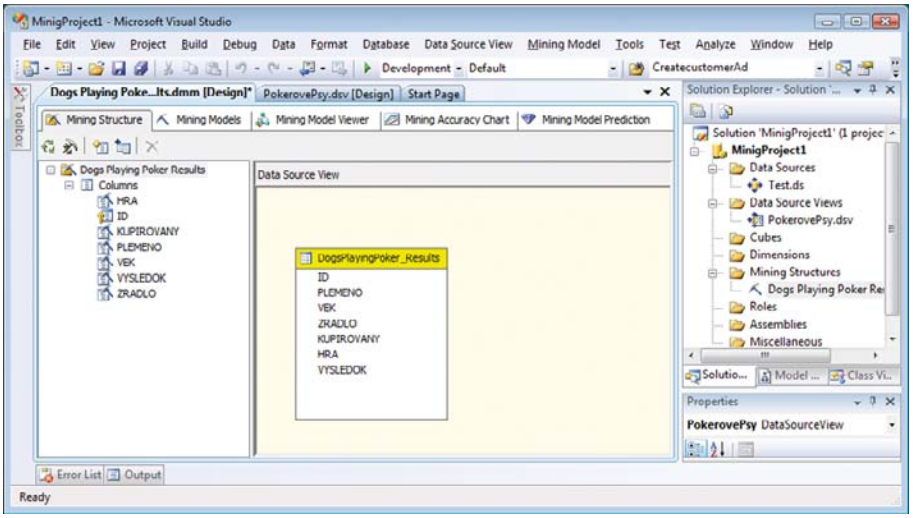
Detekcia a nastavenie diskrétnych a spojitých atribútov

Údaje sa náhodne rozdelia na tréningovú a testovaciu množinu. Pri rozsiahlejších údajoch môžeme nastaviť veľkosť testovacej množiny, alebo maximálny počet údajov na ktorých sa bude model „trénovať". V našom prípade nastavíme 50 percent, bez obmedzenia počtu veď máme len necelých 6000 záznamov.

Nastavenie veľkosti testovacej množiny údajov v percentách

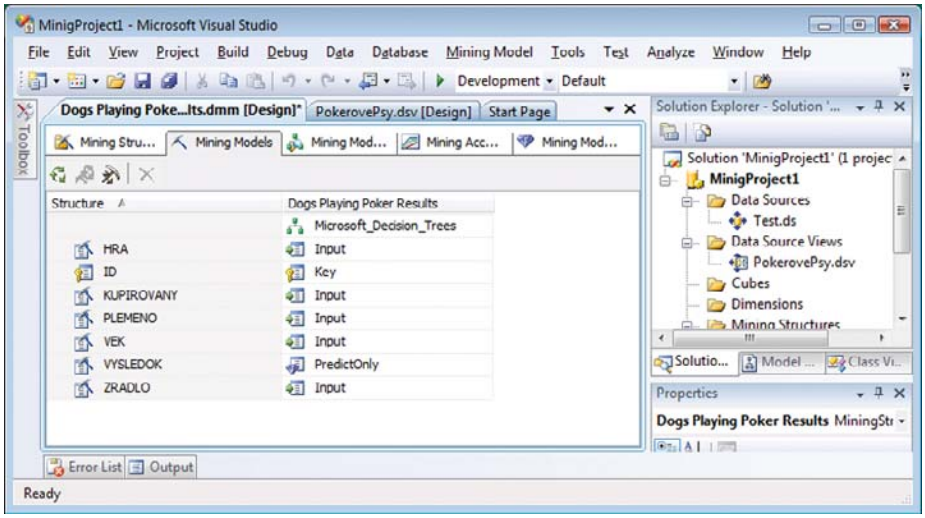
Všimnime si rozloženie pracovnej obrazovky po vytvorení modelu. Hlavné okno je rozložené na záložky:
- Mining Structure,
- Mining Models,
- Mining Model Viewer,
- Mining Accuracy Chart,
- Mining Model Prediction.

Záložka **Mining Structure** obsahuje zoznam stĺpcov.



Data miningový model v prostredí vývojového prostredia (záložka Mining Structure)

Momentálne nás bude najviac zaujímať záložka **Mining Models**, kde sú vybrané vstupné stĺpce, predikovaný a kľúčový stĺpec pre každý data miningový model.

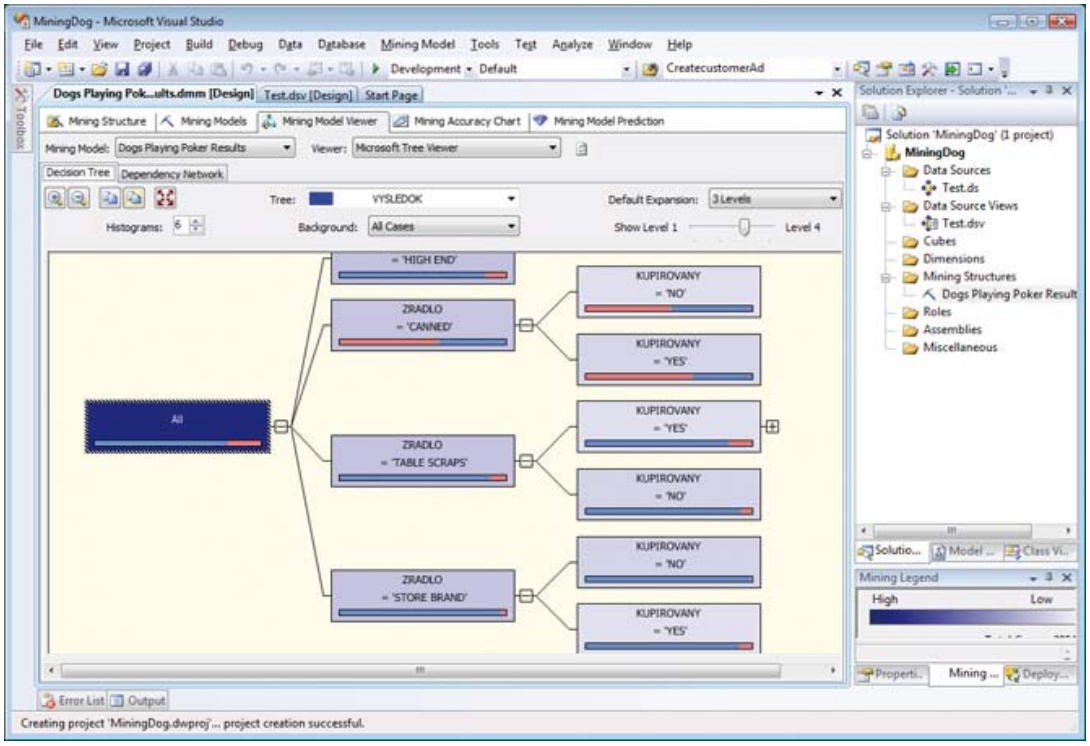


Data miningový model v prostredí vývojového prostredia (záložka Mining Models)

Pomaly, ale isto sa blížime k cieľu. Po nastavení všetkých parametrov môžeme data miningový model zostaviť a zaviesť na server (menu Build | Deploy Solution). Postup zostavovania môžeme sledovať vo výstupnom okne.

Mining Model Viewer

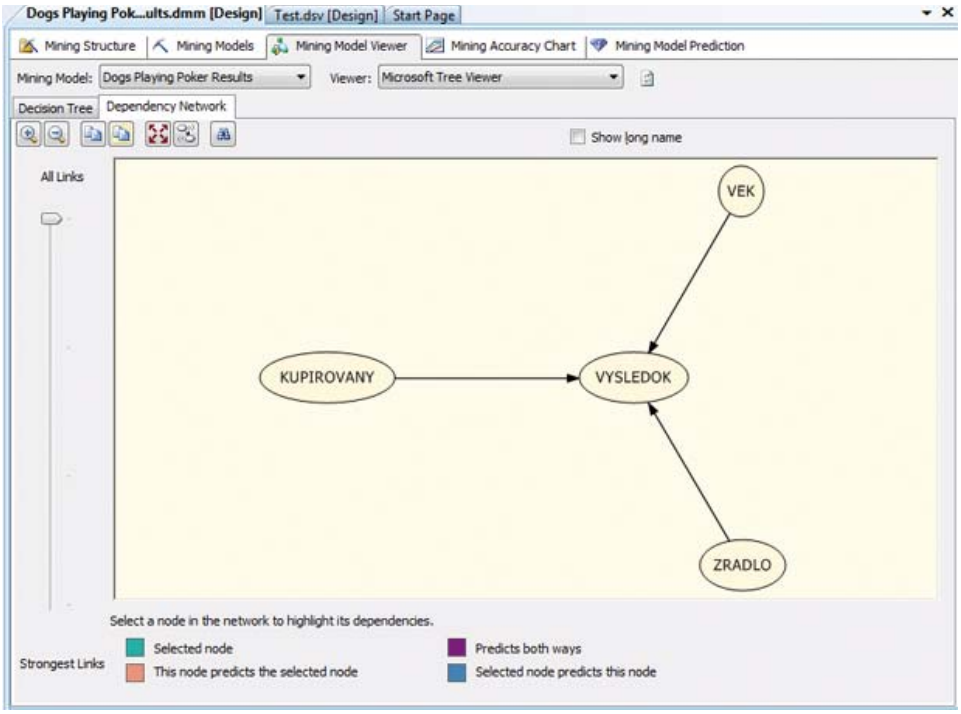
Táto záložka je riešená univerzálne pre prezeranie výsledkov činnosti všetkých algoritmov. V našom prípade sme navrhli model pre dva algoritmy, takže aj v okne Model Viewer si môžeme prezerať Po zostavení modelu sa dej prenesie do záložky Mining Model Viewer. Na zobrazenom diagrame nevyváženého rozpadového stromu môžeme analyzovať hĺbku jednotlivých závislostí a vzťahy vo vnútri nájdenej hierarchie. Čím je farba políčka tmavšia, tým je pravdepodobnosť výskytu tejto vlastnosti vyššia. Každý obdĺžnik diagramu stromovej štruktúry obsahuje aj pomerový indikátor.
Pomocou symbolov + a – môžeme rozbaľovať, prípadne zbaľovať ďalšie úrovne hierarchie.



Decision Tree

Pomer závislosti výskytu predikovanej veličiny vieme približne určiť z pomerového stĺpcového diagramu. Kliknutím na konkrétny uzol zobrazíme aj presné percentuálne podiely v okne Node Legend.

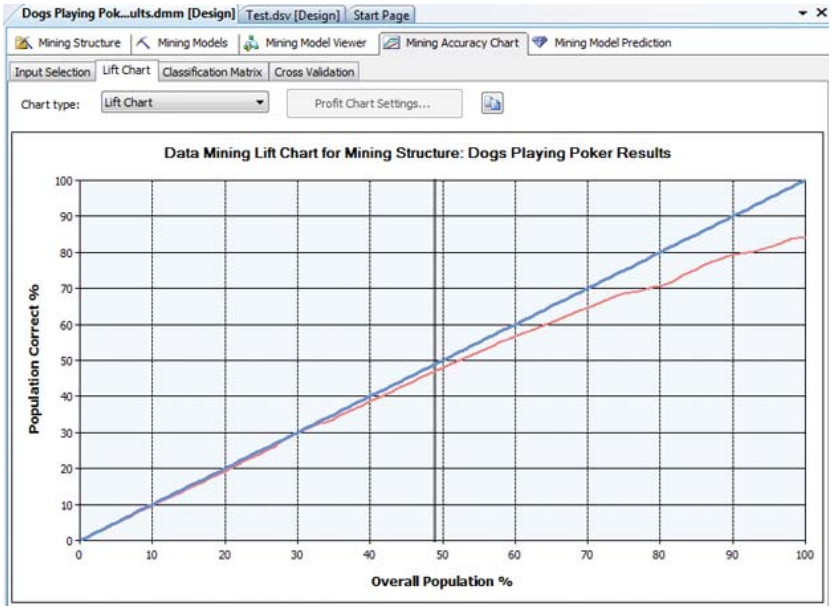
Práca s grafom závislostí (Dependency Network) je úplne intuitívna. Vo forme prehľadného diagramu zobrazí od čoho a v akej miere závisí predikovateľná veličina. Posúvaním „potenciometra" v ľavej časti pracovnej obrazovky môžeme slabšie väzby postupne eliminovať, takže na konci nám zostanú len najsilnejšie závislosti.



Dependency Network

Mining Accuracy Chart

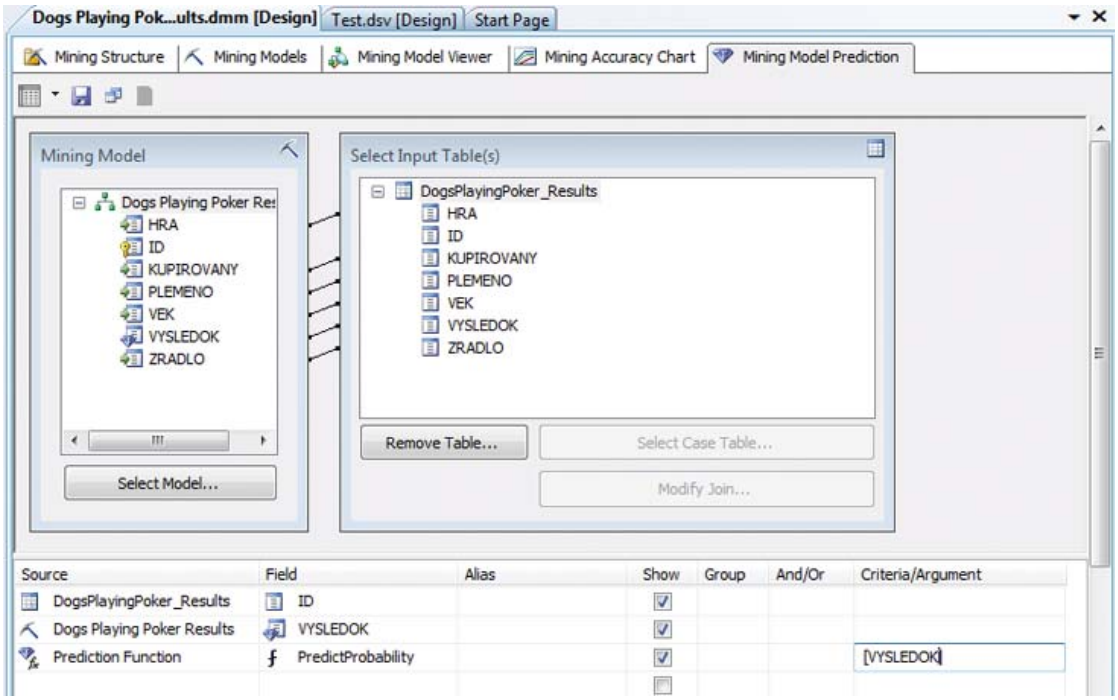
Táto záložka poskytuje možnosť zobrazovania prehľadných Lift Chart grafov pre testovanie data miningových modelov. Princíp grafu je jednoduchý. Najskôr zoradíme výsledky podľa klesajúcej pravdepodobnosti predpovedí a porovnáme ich s náhodným výberom (najhorší možný prípad) a s najlepším teoreticky možným výberom. Krivka pre správne navrhnutý model by samozrejme mala byť čo najbližšie ideálnej krivke a jej sklon by sa mal plynulo znižovať. V opačnom prípade máme nesprávne zvolený model.



Lifting Chart

Predikcia

Návrh dopytu pre predikciu sa bude odohrávať v záložke Mining Model Prediction. Najskôr zvolíme typ algoritmu, ktorý pre predikciu plánujeme použiť. Ak použijeme viac algoritmov, a to, ktorý z nich je lepší, nám prezradí Lift chart v záložke Mining Accuracy Chart.
Ako zdroj údajov pre predikciu použijeme rovnakú tabuľku ako pre trénovanie modelu. Na obrazovke sa zobrazia dve tabuľky, v jednej sú stĺpce z modelu a v druhej stĺpce z predikčnej tabuľky. Pre návrh predikčného dopytu slúži dolná časť obrazovky, kam môžeme presúvať polia z modelu, prípadne predikčnej tabuľky.
Z predikčnej tabuľky ako prvú hodnotu presunieme kľúčový stĺpec, ktorý popisuje jednotlivé prípady, teda ID.
Z modelu použijeme stĺpec VYSLEDOK, hodnotu ktorého chceme predikovať. Najskôr budeme predikovať pravdepodobnosť každého prípadu. V treťom riadku nastavíme Prediction Function, field PredictProbability. Do stĺpca Kriteria /Argument zadáme hodnotu VYSLEDOK Situáciu názorne popisuje obrázok.

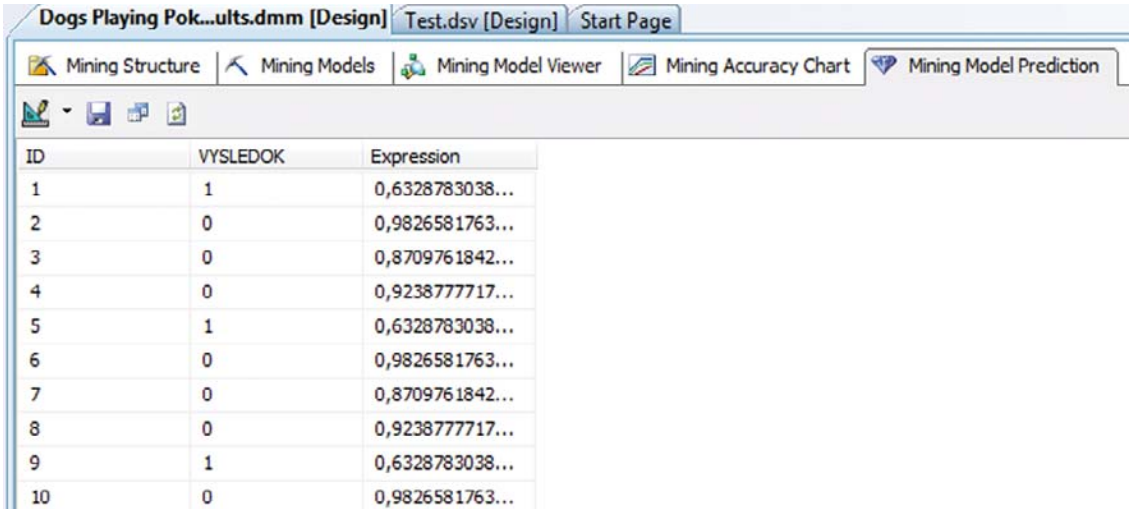


Výber tabuľky a priradenie stĺpcov pre predikciu

O tom koľko práce sme vlastne v spolupráci s wizardom pre návrh predikčného dotazu vykonali interaktívnym vyplnením troch riadkov tabuľky sa môžeme presvedčiť ak si necháme novovytvorený predikčný dopyt zobraziť.

```
SELECT
  t.[ID],
  [Dogs Playing Poker Results].[VYSLEDOK],
  PredictProbability([VYSLEDOK])
From
  [Dogs Playing Poker Results]
PREDICTION JOIN
  OPENQUERY([Test],
    'SELECT
      [ID],
      [PLEMENO],
      [VEK],
      [ZRADLO],
      [KUPIROVANY],
      [HRA],
      [VYSLEDOK]
    FROM
      [dbo].[DogsPlayingPoker Results]
    ') AS t
ON
  [Dogs Playing Poker Results].[PLEMENO] = t.[PLEMENO] AND
  [Dogs Playing Poker Results].[VEK] = t.[VEK] AND
  [Dogs Playing Poker Results].[ZRADLO] = t.[ZRADLO] AND
  [Dogs Playing Poker Results].[KUPIROVANY] = t.[KUPIROVANY] AND
  [Dogs Playing Poker Results].[HRA] = t.[HRA] AND
  [Dogs Playing Poker Results].[VYSLEDOK] = t.[VYSLEDOK]
```

Po prepnutí hlavného okna do režimu Results si môžeme prezrieť výsledky predikcie. Pre každý konkrétny prípad je v tabuľke jednak výsledok predikcie a taktiež aj koeficient zhody, čím si môžeme overiť hodnovernosť predikcie každého konkrétneho prípadu.



ID	VYSLEDOK	Expression
1	1	0,6328783038...
2	0	0,9826581763...
3	0	0,8709761842...
4	0	0,9238777717...
5	1	0,6328783038...
6	0	0,9826581763...
7	0	0,8709761842...
8	0	0,9238777717...
9	1	0,6328783038...
10	0	0,9826581763...

Výsledky predikcie

Keď nahradíme funkciu „Predict Propability" funkciou „Predict" uvidíme predikované údaje pre každý prípad.

Data mining (dolovanie z dát) je veda a umenie extrahovania skrytých hodnotných informácií z veľkých objemov dát, ktoré sa použijú pri tvorbe efektívnych rozhodnutí. Data mining je spôsob učenia sa z minulosti tak, aby sa v budúcnosti prijímali lepšie rozhodnutia. Data mining umožňuje premeniť reaktívnu organizáciu na proaktívnu. Firmy, ktoré nevyužívajú svoje najhodnotnejšie aktívum – údaje a v nich ukryté informácie, budú porazené konkurenciou používajúcou stratégie vyvinuté na základe extrahovania informácií z ich dát.

Kapitola 6: Reportovacie služby

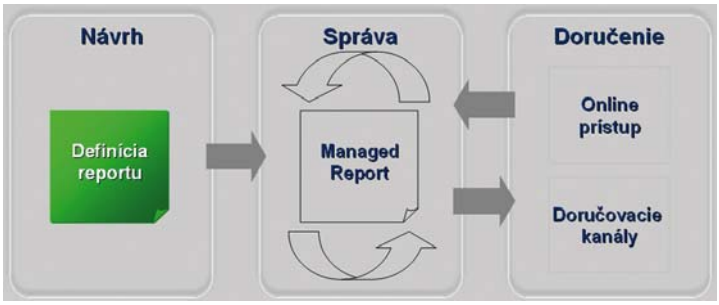
Prevažná väčšina bežných používateľov BI služieb vyžaduje informácie spracované vo forme reportov. Reporty by mali slúžiť na podporu rozhodovania na všetkých stupňoch organizačnej infraštruktúry.

Vo verzii 2008 bola odstránená závislosť na Internet Information Services. Konfigurácia Reporting Services sa tak zjednotila na jedno miesto a nevyžaduje inštaláciu webovej infraštruktúry. Reportovacie služby v jedinom procese umožňujú kontrolovať spotrebu pamäti a jej prideľovanie jednotlivým komponentám. „Veľké" reporty tak už nevytláčajú z pamäte „malé". Výsledkom je väčšia škálovateľnosť a robustnosť s lepšou dobou odozvy. Reportovacie služby získavajú komunikačný protokol priamo z „http.sys" z kernelu operačného systému. Vrstva ASP.NET je hostovaná vo vnútri procesu Reporting Services s vlastným nastavením nezávislým na IIS. IIS a Reporting Services môžu pritom na jednom serveri bezproblémovo spolunažívať aj na rovnakom porte, pretože sa rozlišujú na úrovni rezervovania URL adries.

Životný cyklus reportu začína jeho návrhom, teda vytvorením predpisu, ktorý definuje k akým údajom a akým spôsobom sa bude pristupovať. Až po otestovaní a nasadení bude report generovať požadované údaje jednak na základe informácií, ktoré si nesie v svojom kóde a jednak na základe požiadaviek používateľa. Ani v tejto etape nemôžeme ponechať proces bez „údržby". Azda najvýstižnejšou analógiou je administrácia databázového servera. V životnom cykle reportu už zostáva len jedna maličkosť. Doručiť report vhodným spôsobom v požadovanom čase a požadovanom objeme klientovi. Ak by sme zhrnuli predchádzajúci odsek, dospeli by sme k trom fázam životného cyklu reportu:

- Návrh,
- Správa,
- Doručenie.

Medzi poslednými dvomi etapami, teda správou reportu a procesom jeho doručovania je obojsmerná interakcia.

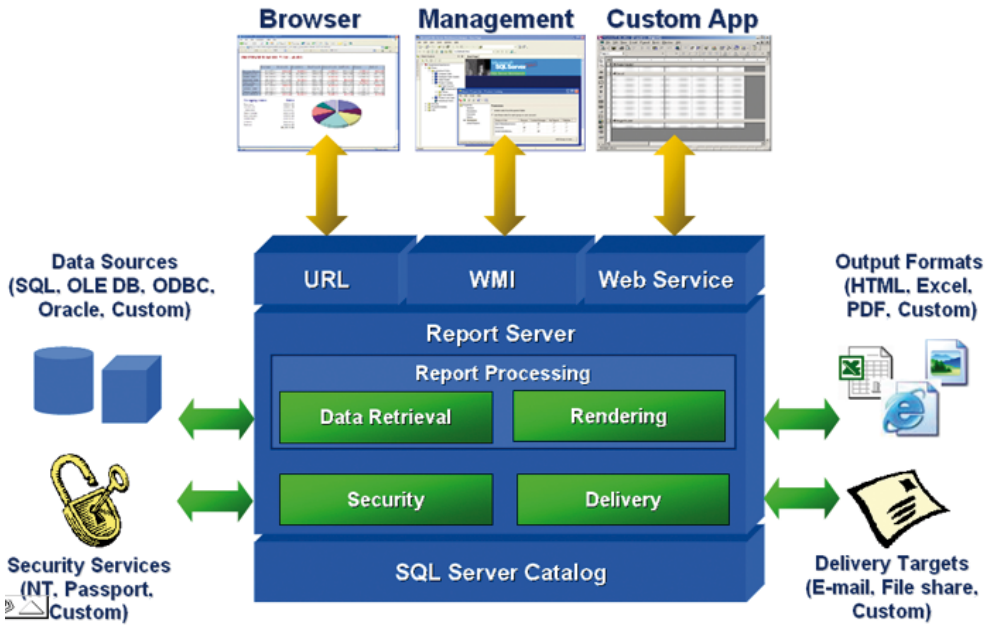


Životný cyklus reportovania

V etape návrhu vývojári navrhujú reporty, ktoré budú následne publikované prostredníctvom servera Report Server. Reporty sa definujú v jazyku **RDL** (Report Definition Language). Kód v jazyku RDL sa zapisuje vo forme XML dokumentu.

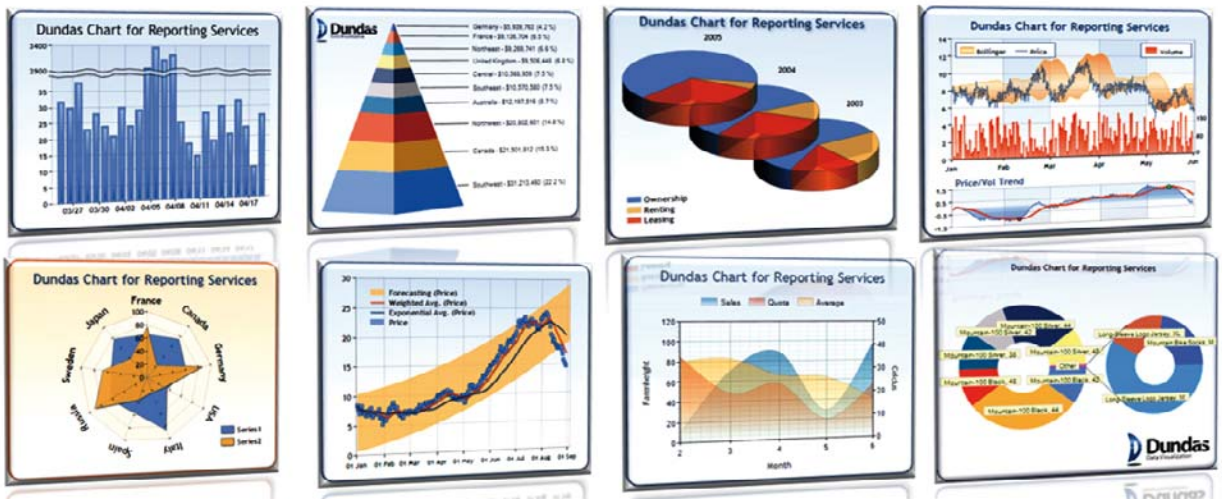
Architektúra reportovacích služieb

Architektúra reportovacích služieb je niekoľkovrstvová. Základnú vrstvu tvorí SQL Server Catalog, čo je databáza pod správou servera SQL Server. Údaje z tejto databázy využíva Report Server pre ukladanie metaúdajov, snapshotov, definícií reportov, priečinkov, údajov pre zabezpečenie a podobne. Dôležitou informáciou je to, že reportovacie služby samotné sú (podobne ako napríklad HTML) bezstavové. Nad vrstvou Report server je vrstva primárnych aplikačných rozhraní URL, WMI a rozhranie pre webové služby. Rozhranie WMI slúži na správu reportovacích služieb. Vo vrstve Report Server prebiehajú rôzne procesy, ktorých cieľom je na základe návrhu reportu a údajov vyrenderovať report v požadovanom výstupnom formáte. Na primárne aplikačné rozhrania sú naviazané aplikácie. Na URL rozhranie pristupujeme pomocou prehliadača webového obsahu, napríklad aplikácie Internet Explorer. Cez WMI rozhranie pristupujeme z administrátorskej konzoly. Bežné aplikácie využívajú rozhranie webovej služby.



Komplexný diagram architektúry reportovacích služieb a ich najbližšieho IT okolia

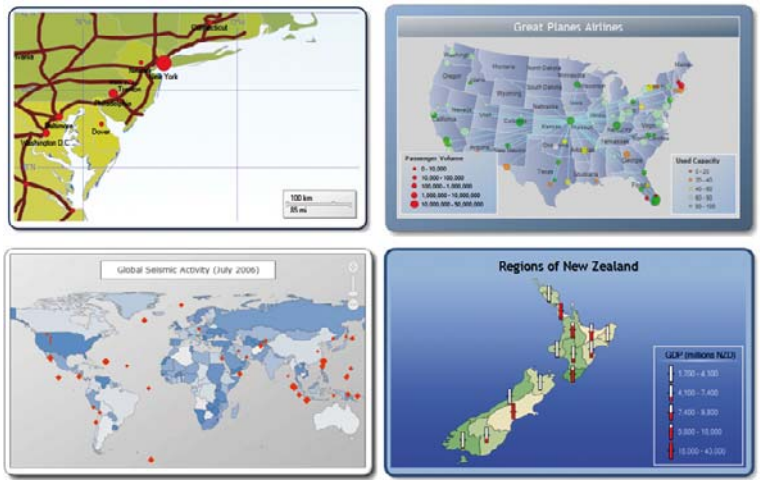
Novinky servera SQL Server 2008 v oblasti reportovacích služieb
Okrem pozmenenej filozofie návrhu nás na prvý pohľad upútajú nové podstatne rozšírené možnosti grafickej prezentácie údajov. Nové typy grafických ukazovateľov získal Microsoft akvizíciou spoločnosti Dundas.



Reportovacie služby podporujú nové typy grafov získaných akvizíciou firmy Dundas

Vzhľadom na globalizáciu biznisu sú zaujímavé možnosti geografickej prezentácie údajov.

Prezentácia geografických údajov

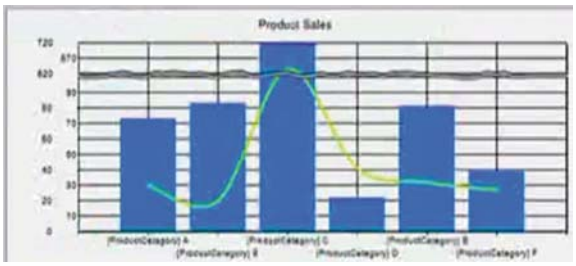


Prezentácia geografických údajov§§

Novinkou je napríklad aj škálovateľná mierka grafov Na obrázku sú ukázané problémy bežných grafov. Niektoré hodnoty sú malé vzhľadom k maximu, podľa ktorého sa vytvorí metrika grafov. Podobne môže byť problém so zobrazovaním dvoch veličín v jednom grafe, najmä ak sú hodnoty jednej z nich nepomerne menšie ako pri druhej veličine.

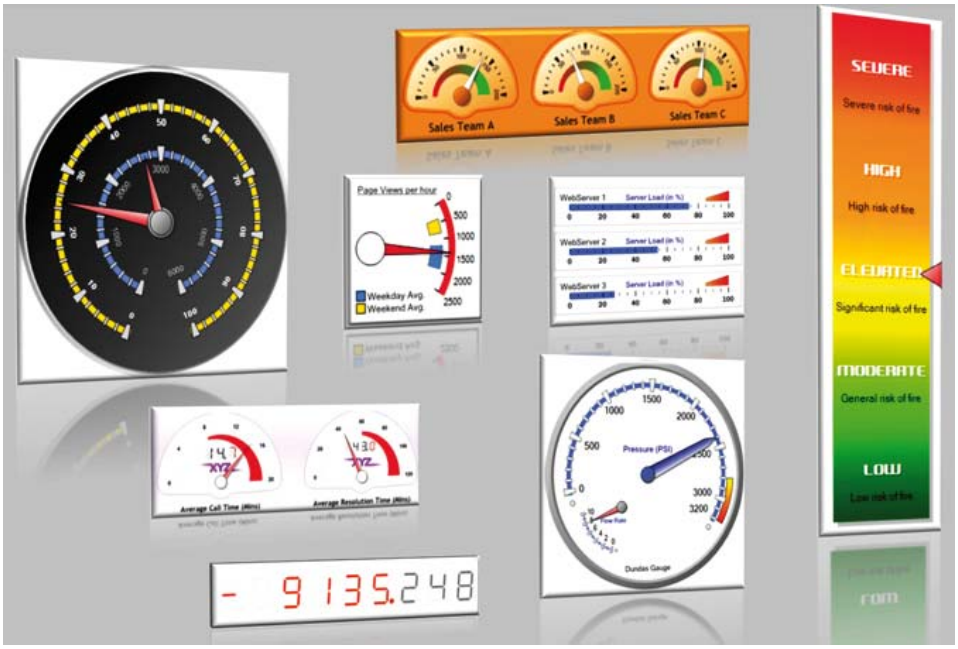
Problémy bežných grafov

Škálovateľná mierka umožní zobraziť aj takéto údaje.

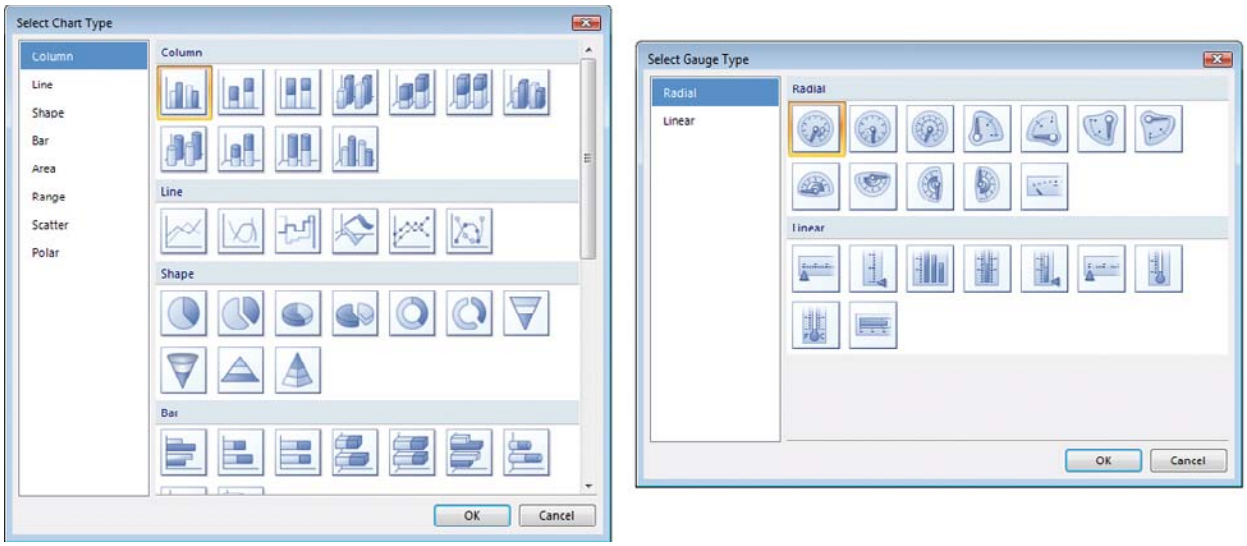


„Scale break"

U manažérov sa obzvlášť veľkej obľube tešia ukazovatele typu „gauges", teda v takej podobe ako ich poznáme z prístrojových dosiek automobilov. Spomeňte si napríklad na otáčkomer. Aj napriek tomu, že mnohí vodiči nerozumejú, aké majú byť presné hodnoty otáčok, aj laik pochopí rozdelenie stupnice na zelené a červené polia a na základe takéhoto jednoduchého ukazovateľa dokáže ustrážiť otáčky v povolenom rozsahu aby nepoškodil motor. Tieto ukazovatele môžu byť lineárne, napríklad vodorovný percentuálny ukazovateľ priebehu, prípadne zvislý ukazovateľ typu „teplomer", alebo radiálne ako na prístrojových doskách automobilov. Pomocou vhodných typov ukazovateľov môžeme zostaviť virtuálny „velín" príslušného procesu.



Vizualizácia údajov a trendov ich zmien pomocou grafických ukazovateľov



Ponuka typov grafov a ukazovateľov priebehu pri návrhu reportu

TABLIX

Veľmi zaujímavou črtou je TABLIX. TABLIX je akronymom zložený z časti slov TABLe a matrIX. Táto nová forma prezentovania výsledkov môže byť odpoveďou na otázku ako najvýhodnejšie prezentovať údaje v tabuľkovej forme. TABLIX spája výhody bežnej a kontingenčnej tabuľky (matice hodnôt).

Table + Matrix

Customer	Growth
Retail	
Acme	19%
Nadir, Inc.	322%
Wholesale	
ABC Corp.	19%
XYZ, Ltd.	322%
Grand Total	56%

		2001	2002	Total
Retail	Acme	1,115	1,331	2,446
	Nadir, Inc.	152	642	794
Wholesale	ABC Corp.	11,156	13,312	24,468
	XYZ, Ltd.	1,523	6,421	7,944
Grand Total		13,946	21,706	35,653

Kombinácia tabuľky a matice (kontingenčnej tabuľky)

Tablix

Customer	2001	2002	Total	Growth
Retail	1,267	1,973	3,230	56%
Acme	1,115	1,331	2,446	19%
Nadir, Inc.	152	642	794	322%
Wholesale	12,679	19,733	32,412	57%
ABC Corp.	11,156	13,312	24,468	19%
XYZ, Ltd.	1,523	6,421	7,944	322%
Grand Total	13,946	21,706	35,653	56%

TABLIX = TABLE + MATRIX

Hierarchické riadky a dynamické umožňujú oveľa prehľadnejšiu prezentáciu rozsiahlejšej štruktúry údajov, pretože sa zobrazujú len tie polia, ktoré obsahujú agregácie.

Pôvodné

		2005	2006
Washington	Total	80	100
	Seattle	50	60
	Spokane	30	40
Oregon	Total	60	80
	Portland	40	50
	Eugene	20	30

Hierarchické riadky

	2005	2006
Washington	80	100
Seattle	50	60
Spokane	30	40
Oregon	60	80
Portland	40	50
Eugene	20	30

Tabuľka obsahujúca hierarchicky usporiadané údaje v riadkoch s dynamickým záhlavím

Pôvodné tabuľky

		2005	2006
WA	Seattle	50	60
	Spokane	30	40
OR	Portland	40	50
	Eugene	20	30

		Table	Chair
WA	Seattle	20	30
	Spokane	10	20
OR	Portland	10	10
	Eugene	25	5

Dynamické skupiny

		Year		Product	
		2005	2006	Table	Chair
WA	Seattle	50	60	20	30
	Spokane	30	40	10	20
OR	Portland	40	50	10	10
	Eugene	20	30	25	5

Paralelné zobrazenie viacerých skupín údajov v jednej tabuľke

Pôvodná tabuľka

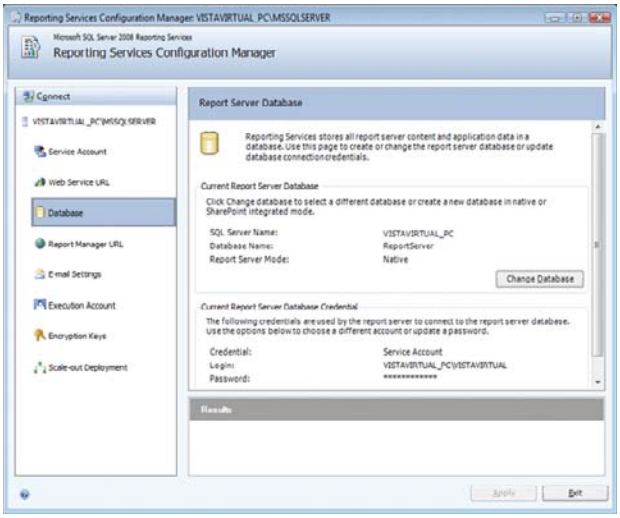
			2005	2006
NY	Joe	Count	1	1
		Sales	50	60
		Avg	50	60
	Sue	Count	1	1
		Sales	80	100
		Avg	80	100
	Total	Count	2	2
		Sales	130	160
		Avg	65	80

Nová tabuľka

			2005	2006
NY	Joe		50	60
	Sue		80	100
	Total	Count	2	2
		Sales	130	160
		Avg	65	80

Zobrazenie hierarchických údajov bez polí, ktoré neobsahujú agregácie

Nástroj na konfiguráciu reportovacích služieb je prístupný cez menu operačného systému *Start – Programs – Microsoft SQL Server – Configuration tools – Report Services Configuration*. Prostredníctvom tohto vizuálneho nástroja dokážeme získať a v prípade potreby aj zmeniť, väčšinu dôležitých parametrov pre prácu reportovacích služieb.



Nástroj na konfiguráciu reportovacích služieb

Príklad reportu

Účelom príkladu je nielen zoznámenie sa vývojovým prostredím v režime návrhu reportu, ale hlavne kompletný postup vytvorenia jednoduchého reportu. Skôr než pristúpime k vytvoreniu reportu, je potrebné venovať sa jeho databázovej časti, t. j. vytvoriť a odladiť SQL dopyt pre výber údajov, ktoré potrebujeme v reporte mať. Príklad je vytvorený nad cvičnou databázou AdventureWorks2008.

Report bude obsahovať atribúty `Category, SubCategory, Sales, Cost` presnejšie – v databázovej terminológii povedané – prvé tri názvy sú aliasy stĺpcov pričom názvy stĺpcov, z databázových tabuliek pre jednotlivé aliasy sú v tabuľke:

Category	ProductCategory.Name
SubCategory	ProductSubCategory.Name

Štvrtý a piaty stĺpec sú výsledkom agregačných funkcií SUM:

Sales	SUM(Product.ListPrice * SalesOrderDetail.OrderQty)
Cost	SUM(Product.StandardCost * SalesOrderDetail.OrderQty)

SQL dopyt pre výber údajov potom bude v tvare:

```
SELECT ProductCategory.Name AS Category,
       ProductSubCategory.Name AS SubCategory,
       SUM(Product.ListPrice * SalesOrderDetail.OrderQty) AS Sales,
       SUM(Product.StandardCost * SalesOrderDetail.OrderQty) AS Cost
FROM Sales.SalesOrderHeader
    INNER JOIN Sales.SalesOrderDetail
      ON SalesOrderHeader.SalesOrderID = SalesOrderDetail.SalesOrderID
    INNER JOIN Production.Product
      ON SalesOrderDetail.ProductID = Product.ProductID
    INNER JOIN Production.ProductSubCategory
      ON Production.Product.ProductSubCategoryID = ProductSubCategory.
ProductSubCategoryID
    INNER JOIN Production.ProductCategory
      ON ProductSubCategory.ProductCategoryID = ProductCategory.ProductCategoryID
    INNER JOIN Sales.Customer
      ON SalesOrderHeader.CustomerID = Customer.CustomerID
GROUP BY ProductCategory.Name, ProductSubCategory.Name
```

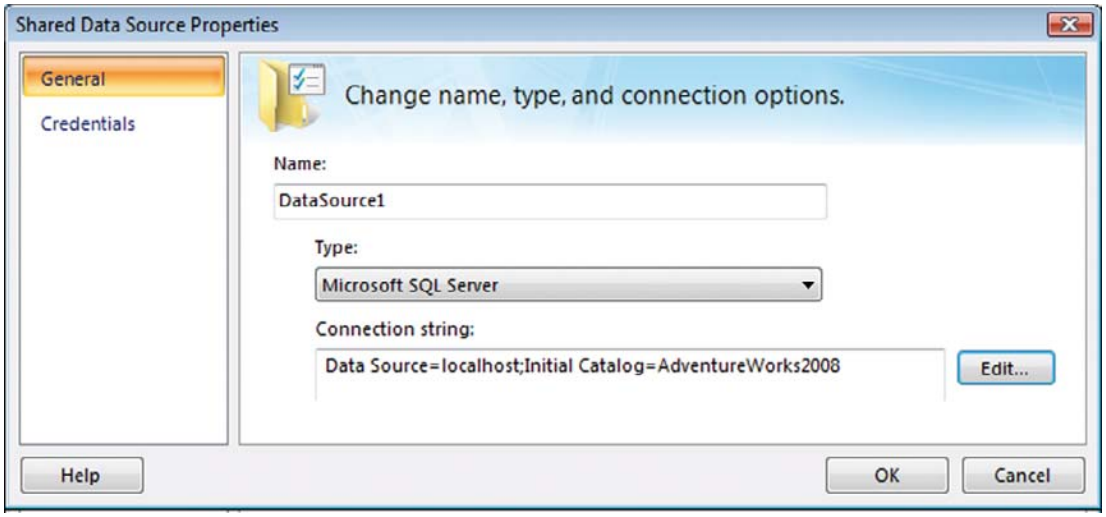
Dopyt vyskúšame, prípadne odladíme pomocou konzolovej aplikácie, napríklad pomocou nástroja SQL Server Management Studio. SQL dopyt môžeme vyskúšať a odladiť aj vo fáze návrhu pomocou nástroja Query Builder.

Návrh reportu

Teraz už môžeme pristúpiť k vytvoreniu projektu. V hlavnom menu nástroja Visual Studio aktivujeme položku menu pre vytvorenie nového projektu. V priečinku typov projektov Business Intelligence Projects máme k dispozícii tri šablóny projektov:

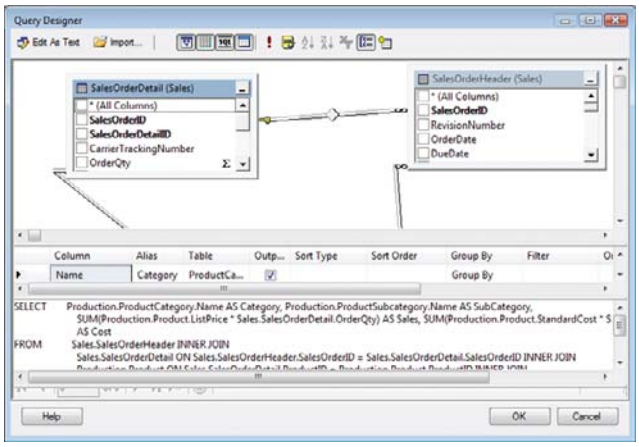
- Report Model Project,
- Report Server Project.

Pre túto aplikáciu využijeme šablónu Report Server Project. Prvým krokom sprievodcu je výber údajového zdroja a nastavenie parametrov pre pripojenie sa k databáze. Údajový zdroj definujeme v záložke Shared Data Sources.



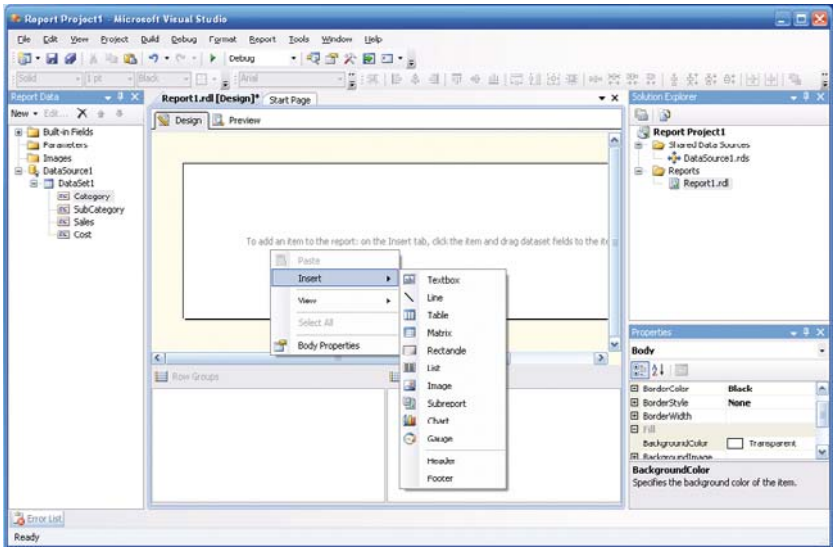
Definovanie databázy AdventureWorks2008 ako zdroja údajov

V priečinku Reports vytvoríme nový report. Kliknutím na odkaz na ploche prázdneho reportu nasleduje pravdepodobne najzaujímavejšia, ale zároveň aj najnáročnejšia časť návrhu reportu – návrh SQL dopytu. Ak máme SQL dopyt vopred navrhnutý a vyskúšaný, čo je aj náš prípad, zadáme jeho textový reťazec v dialógu „Query Designer"



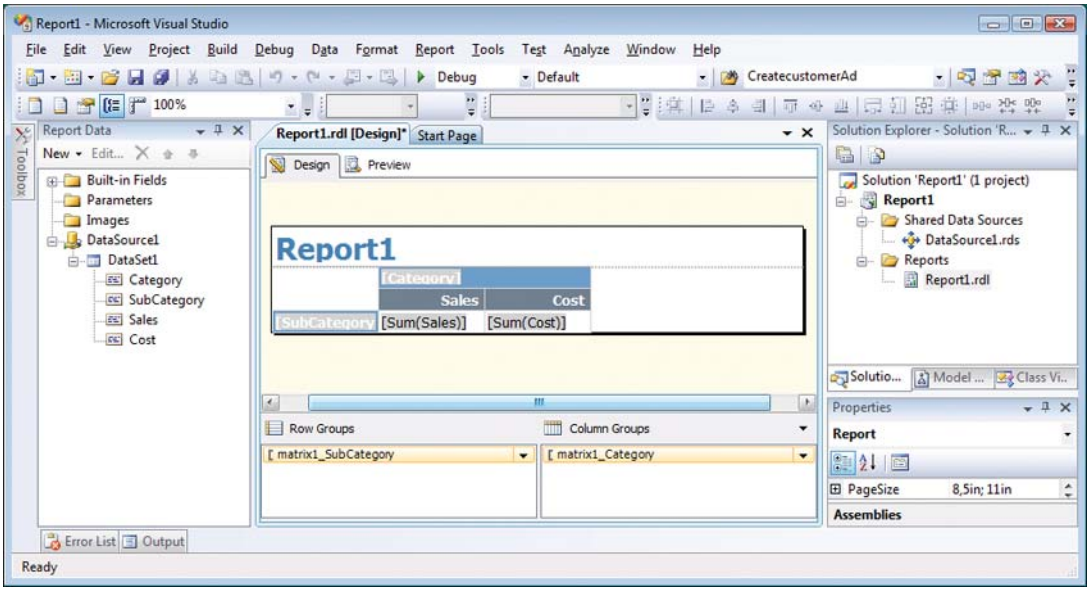
Dialóg pre návrh a otestovanie dopytu slúžiaceho na výber údajov

Nastal čas zoznámiť sa z pracovnou obrazovkou vývojového prostredia v režime Report Designer. Hlavnú pracovnú plochu v strede je možné prepnúť do dvoch režimov:
- **Design** pre návrh formulára reportu,
- **Preview** pre prehliadanie reportu.



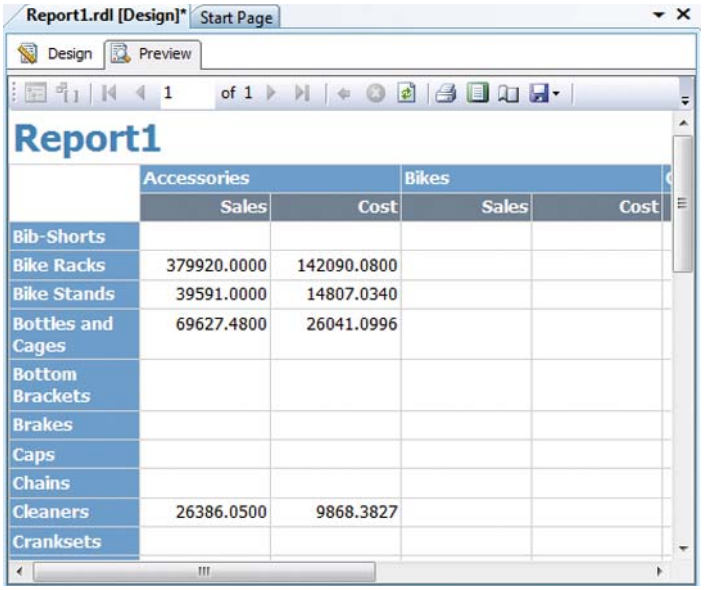
Prostredie pre grafický návrh reportu

Pomocou kontextového menu Insert môžeme vkladať a vhodne na plochu reportu umiestňovať rôzne prvky. My sme využili prvok Matrix, teda kontingenčnú tabuľku do ktorej sme vhodne umiestnili jednotlivé atribúty údajového zdroja.



Príklad návrhu reportu obsahujúceho kontingenčnú tabuľku

Ak sa nám podarí zobraziť náhľad reportu, znamená to, že všetky základné parametre a návrhové prvky sú nastavené a nakonfigurované správne.



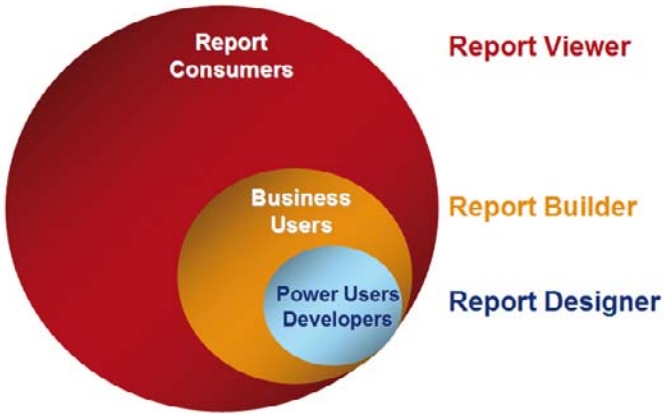
Preview reportu

Pomocou tlačidla so zelenou šípkou môžeme report zostaviť a odoslať na reportovací server. kde bude k dispozícii pre klientov. V okne Output môžeme sledovať protokol tejto akcie aj URL adresu, na ktorej je report prostredníctvom reportovacieho servera prístupný.Túto URL adresu môžeme zadať do prehliadača Internet Explorer a prezrieť si momentálnu podobu reportu, tak ako ju reportovací server poskytuje klientom.

Reportovacie služby servera SQL Server 2008 umožňujú vytvárať aj parametrické reporty. Umožňujú zobraziť len tú podmnožinu údajov, ktorú klient potrebuje. Ak chceme vytvoriť parametrický report, je potrebné principiálne zaviesť do SQL dopytu, ktorý je základom reportu parametre do podmienky.

Report Designer

Popri najpočetnejšej skupine pasívnych konzumentov existujú aj používatelia, ktorí by si chceli pre nich generované reporty prispôsobovať svojim momentálnym požiadavkám, prípadne predmetu skúmania. Do určitej miery im to umožňujú parametrické reporty, v tomto prípade však návrhová štruktúra reportu zostáva nezmenená, pomocou parametrov je možné upraviť len množinu údajov, ktorá bude do reportu zahrnutá. Reportovacie služby servera SQL Server obsahujú aj nástroj Report Builder, pomocou ktorého je možné navrhovať reporty. No je tu jeden veľký rozdiel. Zatiaľ čo ak navrhujeme report v prostredí BI Development Studio, vtedy kvalifikovaný vývojár potrebuje poznať štruktúru údajov. Poznáte to z predchádzajúcich príkladov, kedy bolo potrebné sformulovať SQL dopyt pre ich výber. Od bežného používateľa, napríklad analytika však nemôžeme očakávať, že bude poznať štruktúru údajov, nad ktorými bude report navrhovať. Pre Report Builder preto musia kvalifikovaní návrhári pripraviť akúsi modelovú medzivrstvu – „business model" ktorý bude východiskom pre návrh reportov používateľom. Ten potom potrebuje poznať len štruktúru „business modelu" ale nebude potrebovať poznať štruktúru údajov nad ktorými bol model vytvorený.



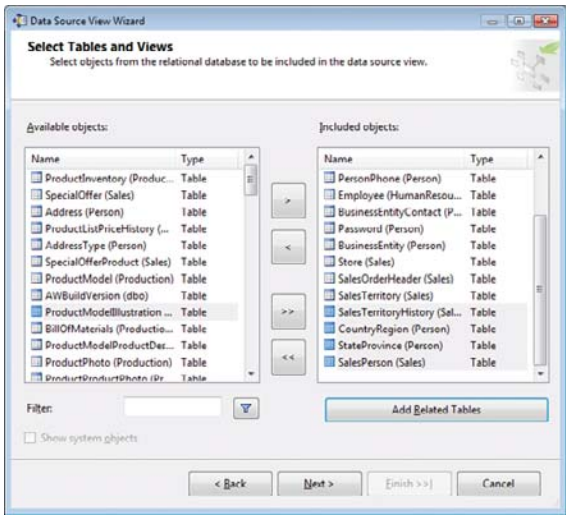
Filozofia a pozícia Report Buildera

Report Designer je klasická Windows aplikácia, ktorá je klientom reportovacích služieb. Po ukončení návrhu bude report umiestnený na server.

Vytvorenie modelu reportu vo vývojovom prostredí BI Dev Studio
V priečinku Business Intelligence Projects vytvoríme nový projekt podľa šablóny Report Model Project. Aj v tejto šablóne je postup vytvárania modelu daný priečinkami v okne Solution Explorer:
- Data Sources (údajové zdroje),
- Data Source Views (pohľady na údajové zdroje),
- Report Models (modely reportov).

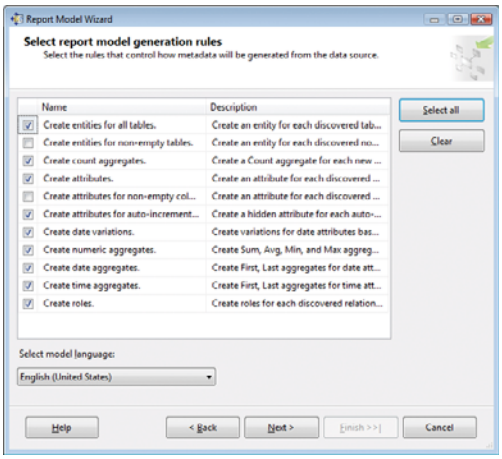
Definovanie zdroja údajov
Ako zdroj údajov definujeme pomocou sprievodcu databázu AdventureWorks2008.

Definovanie pohľadu na údaje
V sprievodcovi pre vytvorenie pohľadov na údaje vyberieme tabuľku Person a tabuľky s ňou relačne zviazané. Tie vyberieme pomocou tlačidla „Add Releated Tables". Podobne postupujeme aj pre tabuľku Sales.Customer. V zozname tabuliek relačne zviazaných s tabuľkou Sales.Customer je aj tabuľka Sales.SalesTerritory. V pravom okne ju označíme a vyberieme aj relačne zviazané tabuľky pre túto tabuľku. Nakoniec pridáme tabuľku Sales. SalesOrderDetail ale už bez tabuliek, ktoré sú s ňou relačne zviazané.



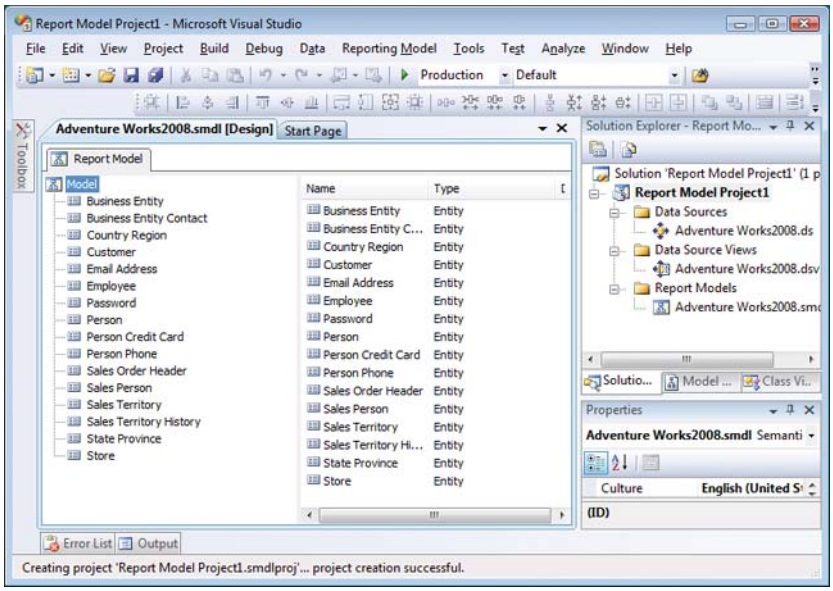
Definovanie pohľadu na údaje

Doteraz vykonávané činnosti sme popísali len heslovito, sú známe z predchádzajúcich projektov. Nasledujúca fáza – vytvorenie modelu je špecifická pre tento druh prípadov, kde sa predpokladá následné použitie nástroja Report Designer. Pri vytváraní modelu preberá taktovku sprievodca Report Model Wizard. Po výbere pohľadu na zdroj údajov sa zobrazí dialóg Select report model generation rules. V ňom ponecháme implicitne označené položky.



Dialóg „Select report model generation rules" sprievodcu „Report Model Wizard"

V dialógu, ktorý bude nasledovať skontrolujeme, či je označená voľba „Update statistics before generating" a nakoniec spustíme zostavenie modelu.

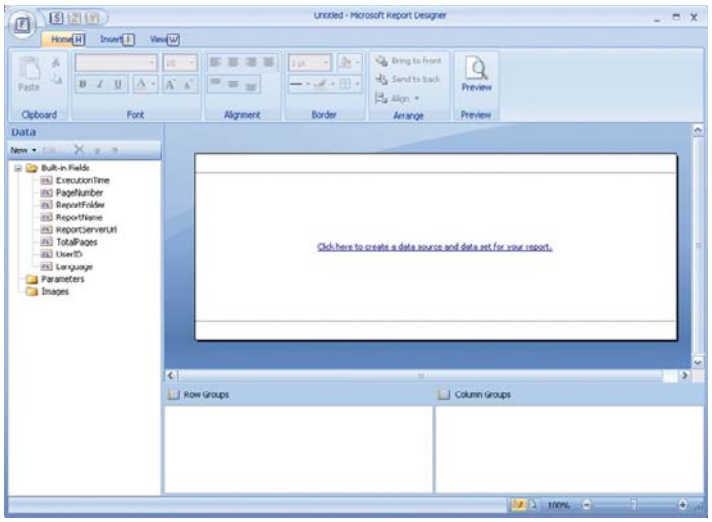


Model reportu zobrazený na pracovnej ploche vývojového prostredia

Projekt zostavíme a uložíme pod správu servera. Ak sa deploy modelu skončil úspešne, to je vlastne finále našej činnosti v nástroji BI Dev Studio. Teraz môžeme pristúpiť k vytváraniu ľubovoľných reportov nad týmto modelom.

Návrh reportu pomocou Report Designer

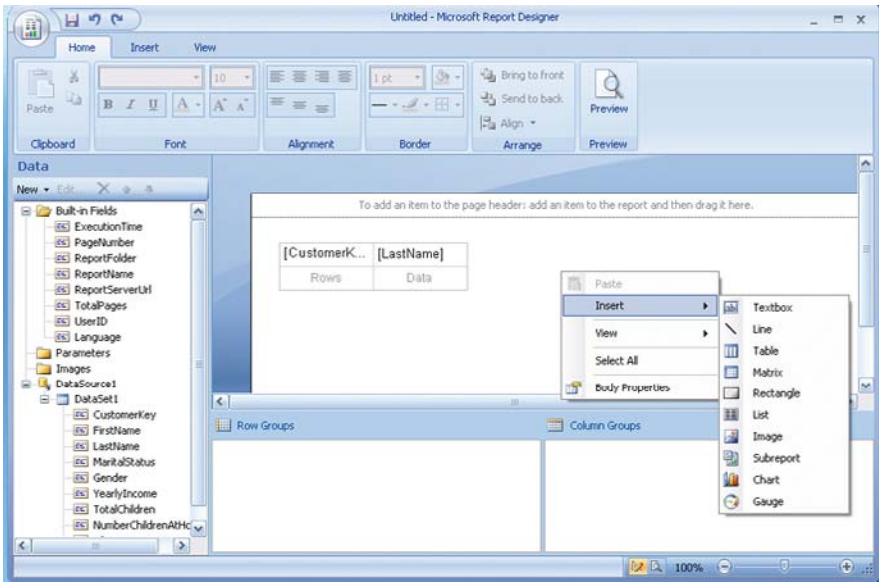
Report môžeme navrhovať buď na základe modelu, alebo priamo definovaním prístupu k údajom.



Report Designer

Využijeme dopyt do databázy AdventureWorksDW2008.

```
SELECT c.[CustomerKey], c.[FirstName], c.[LastName],
              c.[MaritalStatus], c.[Gender], c.[YearlyIncome], c.TotalChildren,
              c.[NumberChildrenAtHome],
              CASE x.[Bikes]
                     WHEN 0 THEN 'No'
                     ELSE 'Yes'
              END AS [BikeBuyer]
       FROM
              [dbo].[DimCustomer] c INNER JOIN (
                     SELECT
                            [CustomerKey] ,[Region],[Age]
                            ,Sum( CASE [EnglishProductCategoryName]
                                          WHEN 'Bikes' THEN 1
                                          ELSE 0
                                   END) AS [Bikes]
                     FROM
                            [dbo].[vDMPrep]
                     GROUP BY
                            [CustomerKey] ,[Region] ,[Age]
                     ) AS [x]
              ON c.[CustomerKey] = x.[CustomerKey]
       join dimgeography g
       on c.geographykey = g.geographykey
```

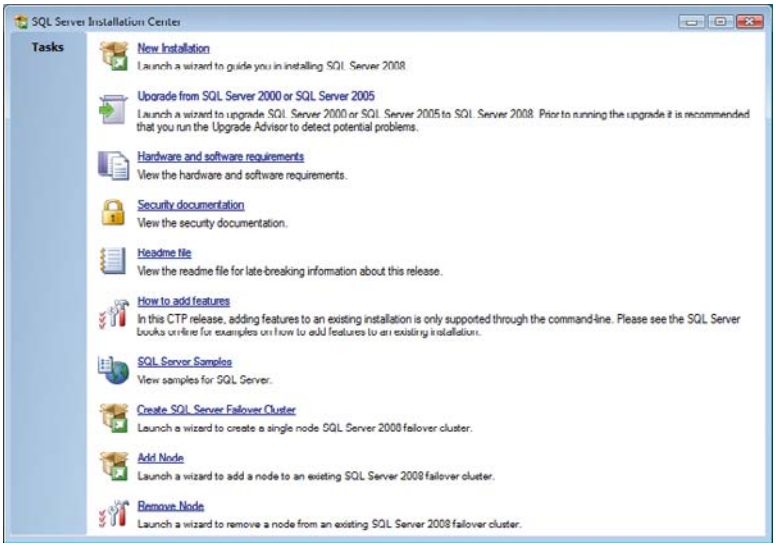


Návrh reportu v prostredí Report Designer

Po výbere zdroja údajov sa zobrazí návrhové prostredie pre vytvorenie reportu. Je tu istá analógia s nástrojom Report Designer vo vývojovom prostredí. Táto aplikácia je však na rozdiel od univerzálneho a všestranne použiteľného vývojového prostredia podstatne jednoduchšia, no je veľmi efektívne vytvorená pre návrh reportu. V ľavej časti sú dva zoznamy – zoznam entít (entities) a polí (fields). Zoznamy obsahujú položky podľa návrhu modelu. Ak sa pozrieme na toolbar, s výnimkou dvoch tlačidiel Design Report (návrh reportu) a Run Report (spustenie reportu), bude nám pole tlačidiel dôverne známe. Obsahuje totiž prvky bežné pre textový editor. Pomocou týchto prvkov môžeme navrhovať dizajn reportu, spôsob zobrazenia reportu, pomocné texty, vysvetlivky a podobne. Keďže report je vlastne sofistikovaný textový dokument, ktorý môže obsahovať prezentačnú logiku, je úplne logické, že návrhové prostredie najviac pripomína textový editor.

Príloha 1: Inštalácia databázového servera, analytických, integračných a reportovacích služieb

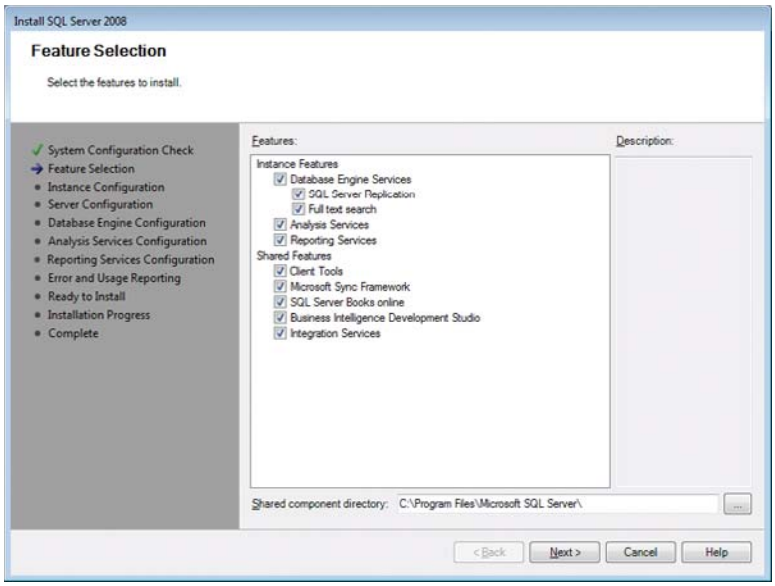
Inštalácia všetkých služieb programov a nástrojov servera SQL Server 2008 je sústredená do utility SQL Server Instalation Center. V prípravnej fáze sa nainštaluje technologická platforma Microsoft .NET Framework 3.5 a podporné súbory (Power Shell, MSI...).



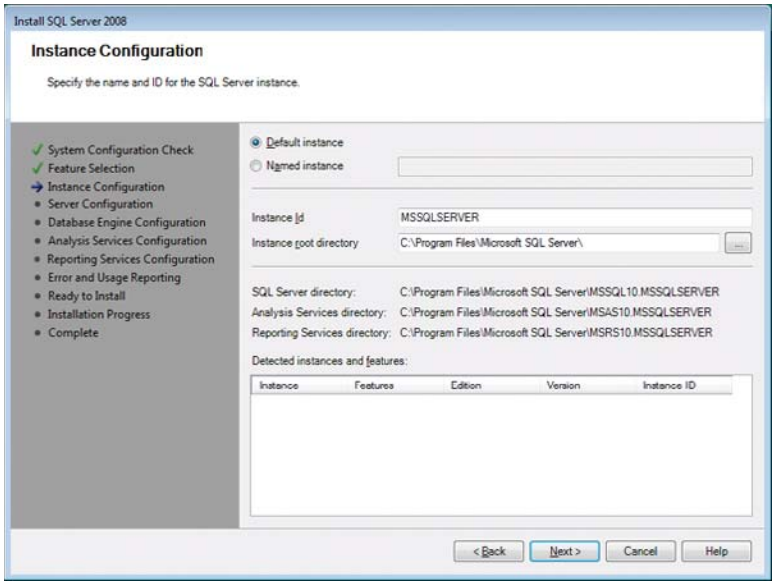
Ponuka možností SQL Server Installation Center

V ďalšej prvej fáze po súhlase s licenčnými podmienkami inštalačný program skontroluje konfiguráciu počítača, či je pre inštaláciu servera SQL Server 2008 vhodný. V prípade nevhodného operačného systému, malej pamäti alebo diskovej kapacity, prípadne pomalého procesora, bude používateľ na túto skutočnosť upozornený.
Po kontrole hardvérovej a softvérovej kompatibility je potrebné špecifikovať, ktoré komponenty chceme nainštalovať. V ponuke sú komponenty:

- databázové služby, v rámci nich môžeme zvoliť, či chceme nainštalovať nástroje pre replikáciu a fulltextové vyhľadávanie,
- analytické služby,
- reportovacie služby,
- integračné služby,
- administrátorské a vývojárske nástroje, Business Intelligence Development Studio, pomocné komponenty a dokumentácia...

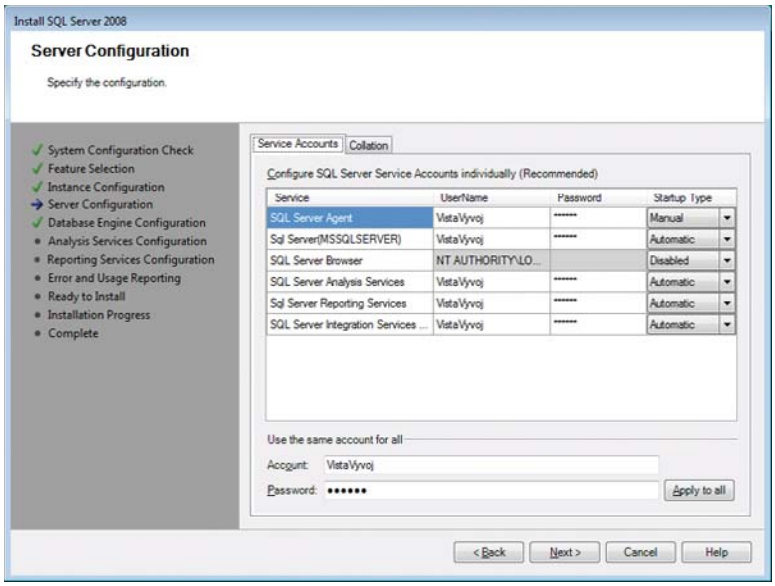


Výber komponentov pre inštaláciu



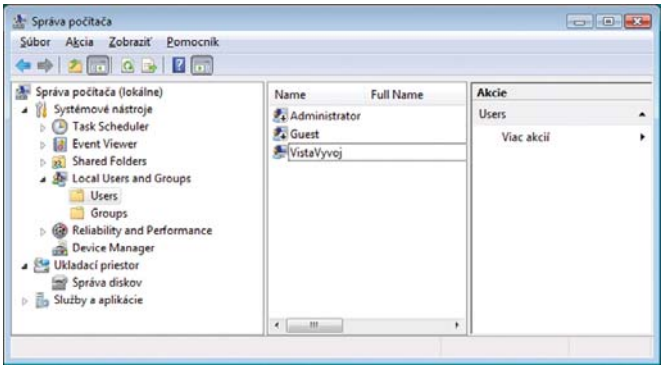
Konfigurácia inštancie

Pokračujeme nastavením prístupových účtov. Vo verzii SQL Server 2008 môžeme nastaviť tieto parametre pre každú požadovanú službu zvlášť. Zároveň v tomto kroku definujeme automatický štart služieb pri spustení operačného systému, prípadne zvolíme možnosť spustiť niektorú službu v prípade potreby ručne.



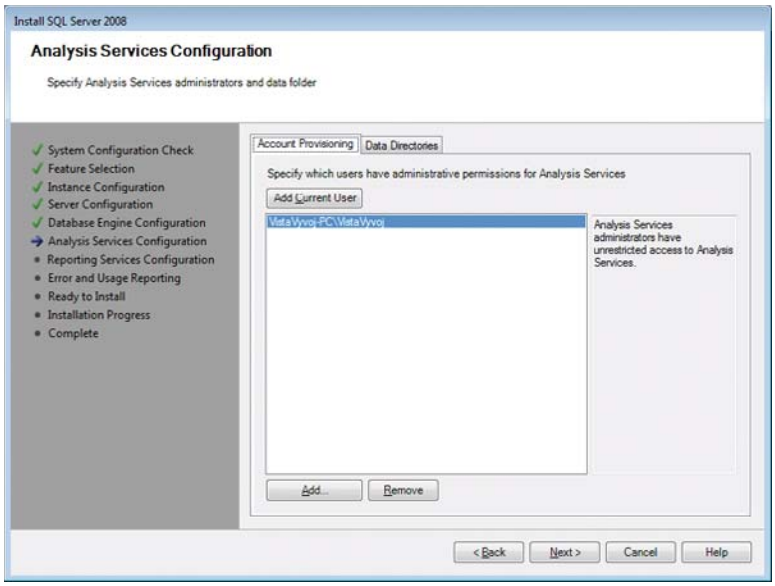
Konfigurácia servera – prístupové účty

Prístupové účty sú spoločné s operačným systémom Windows.

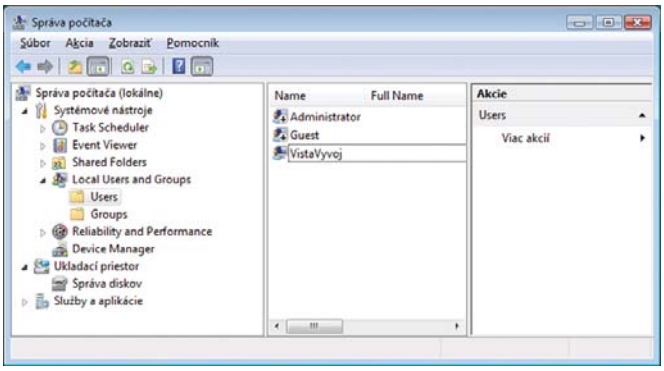


Prístupové účty sú spoločné s operačným systémom Windows

Nasleduje voľba lokálneho alebo doménového prístupu. Môžeme zvoliť možnosť *Windows Authentication Mode* alebo *Mixed Mode*, kedy sa overí autentifikácia operačného systému Windows spolu s prihlasovacím heslom servera SQL Server. Pre tento mód je potrebné zadať heslo, ktoré budeme používať ako administrátor. Používateľské konto musíme špecifikovať aj pre analytické služby.



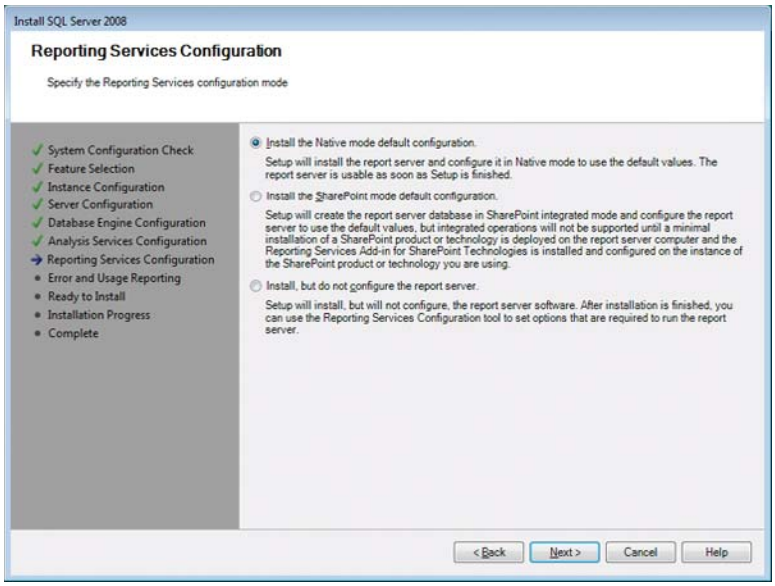
Konfigurácia pre analytické služby



Konfigurácia pre analytické služby – priečinky

C:\Program Files\Microsoft SQL Server\MSAS10.MSSQL2008\OLAP\Data

Posledným krokom je konfigurácia reportovacích služieb. Môžu fungovať buď v natívnom móde alebo v móde kompatibilnom so službou SharePoint.



Konfigurácia reportovacích služieb

Cvičné databázy AdventureWorks2008 a AdventureWorksDW2008 môžeme prevziať na adrese:
http://www.codeplex.com/MSFTDBProdSamples/Release/ProjectReleases.aspx?ReleaseId=10901

www.microsoft.com/slovakia/sqlserver

Microsoft