Vysoká škola ekonomická v Praze Fakulta informatiky a statistiky Vyšší odborná škola informačních služeb v Praze

# Datové sklady a možnosti analýzy a reportování dat ve výuce

Autor bakalářské práce:

David Procházka

Vedoucí bakalářské práce:

Ing. Bc. David Klimánek, Ph.D

školní rok 2011/2012

<Zadávací list>

# Prohlášení

Prohlašuji, že jsem bakalářskou práci na téma Datové sklady a možnosti analýzy a reportování dat ve výuce zpracoval samostatně a použil pouze zdrojů, které cituji a uvádím v seznamu použité literatury.

V Praze, dne 18. 5. 2012

podpis autora

# Poděkování:

Děkuji vedoucímu mé bakalářské práce Ing. Bc. Davidu Klimánkovi, Ph.D. za trpělivost při společných konzultacích a cenné rady a připomínky, bez kterých by tato práce nemohla vzniknout.

# Abstrakt

Tato bakalářská práce se zabývá návrhem sady úloh zaměřené na analýzu a reportování dat, která jsou uložena ve školní cvičné databázi. Teoretická část obsahuje vysvětlení pojmů: Business Intelligence, datové sklady, OLAP databáze. Dále se zde nachází zamyšlení nad významem Business Intelligence pro malý a střední podnik. V praktické části lze nalézt podrobný postup pro vypracování jednotlivých úloh. Tyto úlohy jsou zaměřeny na rozšíření znalostí získaných v předmětech Databázové systémy 1 a Databázové systémy 2.

# Klíčová slova

Business Intelligence, datový sklad, OLAP kostka, analýza dat, reporting

# Abstract

This bachelor thesis deals with a set of tasks aimed at analyzing and reporting data that is stored in the school's training database. The theoretical part contains explanations of terms: Business Intelligence, data warehousing, OLAP database. You can also find here a reflection on the importance of Business Intelligence for small and medium enterprise. The practical part is based on a detailed procedure for the preparation of individual tasks. These tasks are focused on expanding the knowledge acquired in the course Database Systems 1 and Database Systems 2.

# Key words

Business Intelligence, Data Warehouses, OLAP cube, data analysis, reporting

# Obsah

Úvod	9
I. Teoretická část	
1 Business Intelligence	
1.1 Definice BI	10
1.2 Historie BI	11
1.3 Nástroje a aplikace BI	11
2 Datové sklady	
2.1 Definice datového skladu	13
2.2 Důležité pojmy v teorii datových skladů	14
2.3 Architektura datového skladu	15
2.3.1 Dvouvrstvá architektura	15
2.3.2 Třívrstvá architektura	16
2.3.3 Přírůstkový přístup	17
3 OLAP databáze	
3.1Definice OLAP databáze	
3.2 Uložení dat v OLAP databázích	
3.3 Struktura OLAP databáze	
3.2.1 STAR schéma	20
3.2.2 SNOWFLAKE schéma	20
3.3 OLAP datová kostka	21
3.3.1 Operace spojené s OLAP kostkou	23
3.4 Reporting	23
4 Význam Business Intelligence pro malý a střední	í podnik25
II. Praktická část	27
5 Nástroje pro tvorbu cvičných úloh	27
5.1 SQL Server Management Studio	27
5.1.1 Database Engine	
5.1.2 Integration Services	

5.1.3 Analysis Services	28
5.1.3 Reporting Services	28
5.2 SQL Server Business Intelligence Development Studio	29
5.3 Microsoft Excel 2010	
6 Školní databáze	31
7 Návrh cvičných úloh	32
7.1 Úloha č. 1: OLAP transformace dat	32
7.1.1 Formulace problému	32
7.1.2 Postup řešení	32
7.1.3 Shrnutí	35
7.2 Úloha č. 2: Tvorba datové kostky	
7.2.1 Formulace problému	36
7.2.2 Postup řešení	36
7.2.3 Shrnutí	47
7.3 Úloha č. 3: Návrh cvičného reportu	48
7.3.1 Formulace problému	48
7.3.2 Postup řešení	48
7.3.3 Shrnutí	55
7.4 Úloha č. 4: Analýza dat pomocí MS Excel 2010	56
7.4.1 Formulace problému	56
7.4.2 Postup řešení	56
7.4.3 Shrnutí	65
Závěr	66
Seznam použitých zdrojů	67

# Úvod

Pro zpracování své bakalářské práce jsem si zvolil téma Datové sklady a možnosti analýzy a reportování dat ve výuce. Toto mé rozhodnutí bylo podpořeno zejména tím, že se o danou problematikou zajímám, ale také proto, že s ní pravidelně přicházím do styku v rámci odborné praxe (v bankovním sektoru), která já povinnou součástí pro absolvování bakalářského studia na VOŠIS. Sám jsem se již přesvědčil, jak důležité je budování a údržba datového skladu, na jehož provozu jsou přímo závislá mnohá firemní oddělení a bez kterého se v dnešním tvrdém konkurenčním prostředí může firma této velikosti jen stěží obejít. Své prakticky nabyté znalosti bych proto rád využil i v této práci.

V teoretické části se budu zabývat termínem "Business Intelligence", který zastřešuje celou problematiku mé práce a jehož hlavním stavebním pilířem jsou právě datové sklady. Problematiku týkající se datových skladů dále podrobněji popíšu a vysvětlím jejich význam a souvislost s multidimenzionálními databázemi v rámci celopodnikového zpracování dat. Na základě představení těchto pojmů se následně pokusím shrnout přínos Business Intelligence pro malý a střední podnik.

Cílem praktické části a tedy i celé práce bude vytvoření cvičné sady úloh pro OLAP analýzu dat a jejich reportování nad školní cvičnou databází, která je využívána především ve výukových kurzech Databázové systémy 1 a 2. Tyto předměty slouží k seznámení se s principy relačních databází a procvičení jednoduchých i složitějších dotazů v jazyku SQL a jeho proprietárním rozšíření T-SQL. Účelem této sady úloh bude navržení několika scénářů použití a jejich řešení, které zachytí problematiku analýzy dat při využití OLAP kostek a reportování dat pomocí služeb Microsoft SQL Server 2008 R2. Nebude zde však chybět ani využití nástroje Microsoft Excel 2010, který je více dostupný běžnému uživateli.

Na základě navržené sady úloh by tak měli mít studenti možnost naučit se data z databáze dále zpracovávat a analyzovat. Díky tomu by pro ně mělo být také jednodušší si uvědomit, že hlavním účelem databází není pouze data uchovávat, ale také je uložit takovým způsobem, aby mohla být opětovně využívána pro potřeby firemního řízení a podporu manažerského rozhodování.

# I. Teoretická část

# **1** Business Intelligence

V dnešní moderní době existuje mnoho systémů, které slouží k podpoře nejrůznějších firemních aktivit a zájmů. Velká část současné populace již jistě slyšela něco o ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) případně SCM (Supply Chain Management). K těmto pojmům v relativně nedávné době přibyl termín "*Business Intelligence*".

Cílem této kapitoly je vymezení pojmu Business Intelligence a představení jeho jednotlivých komponent, jejichž význam bude často rozebírán i v kapitolách následujících.

# **1.1 Definice BI**

Popis termínu Business Intelligence se v různých zdrojích liší a existuje celá řada definic, z nichž pro lepší představu uvádím následující:

"Business Intelligence je množina konceptů a metodik, které zlepšují rozhodovací proces za použití metrik, nebo systémů založených na metrikách. Účelem procesu je konvertovat velké objemy dat na poznatky, které jsou potřebné pro koncové uživatele. Tyto poznatky potom můžeme efektivně použít například v procesu rozhodování a mohou tvořit velmi významnou konkurenční výhodu." [9, s. 14]

"BI je výraz pro procesy, znalosti, aplikace, platformy, nástroje, technologie, které podporují porozumění datům, jejich vztahům a trendům." [1, s. 90]

"Sada procesů, aplikací a technologií, jejichž cílem je účinně a účelně podporovat rozhodovací procesy ve firmě. Podporují analytické a plánovací činnosti podniku a organizací a jsou postaveny na principech multidimenzionálních pohledů na podniková data." [12, s. 19]

"Souhrnný pojem pro procesy, technologie a nástroje potřebné k přetvoření dat do informací, informací do znalostí a znalostí do plánů, které umožní provést akce podporující splnění primárních cílů organizace." [2]

Z výše uvedených definic se dá usoudit, že se jedná o velice důležitý pojem, který hraje klíčovou roli ve zkvalitnění strategického řízení firmy.

# **1.2 Historie BI**

Koncem sedmdesátých let minulého století se začala v USA objevovat první řešení za účelem podpory analytických úloh pro podnikové řízení. Protože se tato řešení brzy ukázala jako výhodná pro chod podniku, objevily se na trhu také první produkty založené na multidimenzionálním uložení a zpracování dát, označované jako EIS (Executive Information System). Stejně tak se v USA na přelomu osmdesátých a devadesátých let začal z důvodu neustále narůstajících objemů dat rozmáhat trend zavádění datových skladů. Jako reakci na tento vývoj vymezil v roce 1989 Howard J. Dresner, zaměstnanec konzultační společnosti Gartner Group, pojem Business Intelligence, který se těmito trendy zabývá. Definoval ho jako *"sadu konceptů a metod pro zlepšení rozhodování firem."* [12, s. 18]

Za více jak dvacet let od zavedení tohoto termínu došlo k jeho obrovskému rozmachu. V současnosti se již jedná o celosvětově rozšířený pojem, jehož technologie jsou využívány v menší či větší míře ve většině úspěšných firem.

# 1.3 Nástroje a aplikace BI

Do nástrojů a aplikací Business Intelligence se zahrnují [12, s. 19]:

- Zdrojové systémy
- Dočasná úložiště dat (DSA Data Staging Area)
- Transformační nástroje (ETL Extraction Transformation Loading)
- Integrační nástroje (EAI)
- Datové sklady (DWH Data Warehouses)
- Datová tržiště (DMA Data Marts)
- OLAP (On Line Analytical Processing)
- Reporting
- Manažerské aplikace (EIS Executive Information Systems)
- Dolování dat (Data Mining)
- Nástroje pro zajištění kvality dat

Některé z těchto nástrojů budou z hlediska své důležitosti pro architekturu datových skladů více rozebrány v kapitole 2.2.

Pro lepší názornost ještě uvádím schéma, které tyto pojmy znázorní pravděpodobně lépe, než složité popisování jejich provázanosti vlastními slovy:



Obrázek č. 1: Komponenty Business Intelligence, převzato z [4]

Jak je vidno z předchozího obrázku, tak architektura Business Intelligence je v zásadě tvořena čtyřmi na sebe navazujícími vrstvami. Každá vrstva představuje jeden typ procesu, na jehož základě jsou data zpracovávána. Pořadí těchto procesů je pevně dané a v praxi by se tedy nemělo stát, aby došlo ke zpracování dat v navazující vrstvě, pokud ještě úspěšně neprošla vrstvami předcházejícími.

V první fázi prochází data ze zdrojových systémů vrstvou datové transformace. Zdrojové systémy jsou obvykle zastoupeny podnikovými informačními systémy a dílčími databázemi, kde jsou shromažďována data jednotlivými firemními celky. Jsou následně extrahována a transformována, aby odpovídala požadovanému formátu v místě jejich nového úložiště, kam jsou následně nahrána. Poté, co byla data transformována a uložena do tohoto sjednocujícího úložiště, následuje navázání na vrstvu určenou pro analýzu dat. Tato vrstva, jak už název napovídá, slouží pro zisk strategicky důležitých informací, na jejichž základě se vedení firmy dále rozhoduje, jak bude dále postupovat. Výsledky analýzy dat jsou v konečné fázi předány vrstvě prezentační, kde jsou různými způsoby distribuovány koncovým uživatelům.

# 2 Datové sklady

Datové sklady (Data Warehouses) představují hlavní datovou základnu pro aplikace BI. Cílem kapitoly je seznámení se s významem a základními principy jejich architektury, ale i s jedním z možných pohledů, dle kterého je na datové sklady v této práci nahlíženo.

V souvislosti s prudkým rozvojem informačních technologií představují podniková data pro většinu společností cenný zdroj informací. Tato data jsou často roztříštěna po různých podnikových systémech, jejichž funkcionalita se většinou liší natolik, že i data v nich obsažená mají odlišnou strukturu a existuje u nich vzájemná nekompatibilita. S narůstajícím objemem dat se však tento problém začal stále více prohlubovat, protože získání požadovaných dat napříč celým podnikem bylo čím dál více složitější.

# 2.1 Definice datového skladu

Termín *"datový sklad"* poprvé definoval v roce 1991 William Harvey Inmon, který je často nazýván "otcem datových skladů". Jeho definice zní následovně:

"Datový sklad je integrovaný, subjektově orientovaný, stálý a časově rozlišený souhrn dat, uspořádaný pro podporu potřeb managementu." [7, s. 26]

Pojmy z Inmonovy definice se dají interpretovat takto [12, s. 32]:

- Subjektová orientace data jsou rozdělována podle jejich typu, ne podle aplikací, ve kterých vznikla.
- Integrovanost data jsou ukládána v rámci celého podniku, a ne pouze v rámci jednotlivých oddělení.
- Stálost datové sklady jsou koncipovány jako "Read Only", což znamená, že zde žádná data nevznikají ručním pořízením, a nelze je ani žádnými uživatelskými nástroji měnit.
- Časová rozlišenost aby bylo možné provádět analýzy za určitá období, je nutné, aby byla do datového skladu uložena i historie dat. Načítaná data s sebou tedy musí nést i informaci o dimenzi času.

Obecně může existovat více pohledů, dle kterých lze na datové sklady nahlížet. Jejich hlavní rozdíl spočívá v rozsahu komponent a procesů, které jsou vnímány jako dílčí součást skladu. Na celkové koncepci se však prakticky nic nemění, protože počet a provázanost komponent samotných zůstává nezměněn. Jeden z možných pohledů je do značné míry vystižen přímo v obrázku č. 1 na straně 12. Tento pohled vnímá datový sklad jako hlavní datové úložiště ve vrstvě databázových komponent. V poněkud širším vymezení je tento pojem chápán v druhém pohledu, kde jsou zahrnuty procesy od ETL zpracování dat až po jejich postupné převedení do jednotlivých datových tržišť. Datový sklad je tak v tomto případě tvořen prvními dvěma vrstvami architektury BI.

Osobně jsem si již zvyknul při své praxi v bance datový sklad vnímat tímto druhým pohledem, jelikož právě takto je tam na něj nahlíženo. Konkrétně na mém pracovišti jsou jako součást datového skladu označeni také zaměstnanci, kteří se starají o jeho provoz, protože bez jejich každodenní práce a údržby celého skladu by zkrátka nemohl fungovat a přinášet firmě užitek. I nadále bych si proto dovolil nahlížet na tuto problematiku tímto způsobem.

## 2.2 Důležité pojmy v teorii datových skladů

Dříve, než se pustím do popisu samotné architektury datového skladu, bude třeba blíže představit některé komponenty a procesy, které jsou součástí dané problematiky. Tyto pojmy byly již zmíněny v kapitole 1.3 a setkáme se s nimi minimálně ještě v následující kapitole v rámci jednotlivých architektonických přístupů k zavádění datových skladů. Jedná se o následující výrazy:

**ETL** (Extaction Transformation Loading) – tzv. *"datová pumpa"*. Jedná se o sled komplexních algoritmů, které slouží k požadované úpravě a převedení dat ze zdrojových systémů do datového skladu. Pro tento účel jsou často využívány specializované ETL nástroje (např. MS SQL Server Integration Services, Informatica).

**Dočasné úložiště** (Data Staging Area) – jedná se speciální typ relační databáze, jejíž význam spočívá v rychlé extrakci (extrakční fáze ETL) dat ze zdrojových systémů, aby mohla být následně vyčištěna a transformována (transformační fáze ETL) bez zbytečného zatěžování těchto systémů, které jsou již tak dost vytíženy běžným provozem. Data jsou zde uložena pouze krátkodobě, obvykle v rozmezí jednoho dne až měsíce a jsou tak obrazem dat ve zdrojových systémech za tento časový interval.

**Datové tržiště** (Data Mart) – je podmnožinou datového skladu, která se typicky orientuje na jednu konkrétní část podnikového zaměření. Každé datové tržiště obsahuje data vztahující se k dané části a je tedy primárně určeno pro potřeby odpovídající skupině uživatelů, kteří jsou na okruhu těchto dat závislí.

Podle Marka Humpriese [6, s. 35] by se dala definice datového tržiště z předchozí strany shrnout i následovně: "Data pro datová tržiště jsou vybírána s cílem vyhovět specifickým požadavkům částem organizace. Není neobvyklé najít datová tržiště vyvinutá a implementovaná pro oddělení, divizi či geografickou lokaci."

# 2.3 Architektura datového skladu

Existují rozdílné přístupy, koncepce k tvorbě datových skladů. Přístupy se liší zvolenou architekturou, složitostí a nákladností implementace, možnostmi a složitostí škálovatelnosti. Obvykle se můžeme setkat se třemi přístupy k řešení [12, s. 45]:

- Postupné budování datových tržišť, založené na architektuře nezávislých datových tržišť tzv. "dvouvrstvá architektura"
- Jednorázové vybudování celkového řešení, založené na architektuře konsolidovaného datového skladu – tzv. "*třívrstvá architektura*"
- Přírůstkový přístup založený na architektuře konsolidovaného datového skladu

# 2.3.1 Dvouvrstvá architektura

Tato architektura byla navržena Ralphem Kimballem a její princip je založen na konceptu vzájemně nezávislých datových tržišť. Datový sklad se buduje postupně po jednotlivých tržištích a nejen výsledky, ale i finanční prostředky na vývoj jsou rozloženy v čase. Tento přístup je volen především tehdy, pokud je třeba upřednostnit konkrétní oddělení či pobočku a dodat první výstupy z datového skladu v co nejkratším možném čase. [4]



Obrázek č. 2: Postupné budování datových tržišť, převzato z [8]

Dvouvrstvá architektura však spolu nese i určitá rizika. Při postupné integraci nových datových tržišť může dojít k jejich částečnému překrývání, kdy se jeden atribut vyskytuje na několika místech a jakákoli změna musí být provedena ve všech instancích tohoto atributu, což sebou nese i vyšší náklady na údržbu datového skladu.

Tento přístup je vhodné používat v následujících situacích [12, s. 47]:

- Není technologicky možné nebo není potřebné budovat celopodnikové řešení založené na třívrstvé architektuře.
- Je potřeba vybudovat rychle řešení pro několik vzájemně nezávislých oddělení, přičemž se neočekává do budoucna potřeba celkové integrace řešení.
- Zadavatel nemá nebo není ochoten vynakládat finanční prostředky na počáteční integrační činnosti spojené s architekturou konsolidovaného datového skladu.

# 2.3.2 Třívrstvá architektura

Zakladatelem třívrstvé architektury je již výše zmíněný William Harvey Inmon. Její podstatou je jednorázové vybudování celého databázového řešení, které pokryje všechny požadované analytické potřeby firmy. Tato koncepce s sebou nese vyšší počáteční náklady a ve většině případů také značně dlouhou dobu na kompletní realizaci.



Obrázek č. 3: Architektura konsolidovaného datového skladu, převzato z [8]

Oproti předchozímu má tento přístup následující výhody [12, s. 48]:

• Architektura je dostatečně flexibilní a integrovaná pro podporu náročných analytických úloh požadujících nejen agregovaná, ale i detailní data.

- Vzhledem ke komplexnosti uložených dat lze jednoduše budovat téměř neomezené množství datových tržišť pro potřeby různých uživatelů.
- Díky udržování normalizovaných transakčních dat na úrovni datového skladu je možno tato data použít i pro jiné typy analytických úloh, než pro jaké byly původně určeny.
- Datový sklad přímo podporuje tvorbu specializovaných datových úložišť i v jiné než dimenzionální formě (např. pro dolování dat).

# 2.3.3 Přírůstkový přístup

Tento typ je stejně jako předcházející přístup založen na principu třívrstvé architektury a jedná se o nejmladšího zástupce z již uvedených architektur, který se snaží skloubit výhody obou předchozích přístupů. Jeho asi vůbec největší výhodou je dle [12, s. 49] fakt, že jednotlivá řešení jsou dodávána postupně a proto i finanční náklady jsou více rozprostřeny v čase, což umožňuje pružnější sledování návratnosti těchto investic. Neméně důležitým kladem je i to, že je zde díky principu postupných přírůstků více prostoru pro případné změny v konceptu. Není tedy divu, že se v současnosti jedná o nejpoužívanější typ řešení.



Obrázek č. 4: Sklad založený na přírůstkovém principu, převzato z [8]

Tento přístup je vhodné používat například v následujících situacích [12, s. 50]:

- Zadavatel chce vybudovat konsolidované řešení, díky čemuž bude mít možnost zajišťovat případné změny v koncepci datového skladu, neboť očekává signifikantní rozvoj v uživatelských požadavcích.
- Zadavatel je ochoten investovat počáteční časové i finanční prostředky k vytvoření celkové strategie a následně budovat malá řešení přinášející okamžitý užitek

# 3 OLAP databáze

V této kapitole se dostáváme ke konečnému stádiu zpracování dat, které má v praxi pravý význam pro firemní plánování a stanovování nových strategií. Dosud jsem se zabýval hlavně problematikou, která se zabývá integrací dat do celopodnikového řešení v podobě budování datových skladů. Tato část se týkala především tzv. "transakčního" zpracování dat OLTP (On-Line Transactional Processing), které je primárně uzpůsobeno pro relační databázovou základnu, nad kterou běží klíčové aplikace a systémy podporující každodenní chod firmy. Od této chvíle se však situace mění a nadále se budu zabývat technologií OLAP (On-Line Analytical Processing), na jejíž podstatě jsou z velké části založeny aplikace Business Intelligence.

# **3.1Definice OLAP databáze**

Na rozdíl od transakčních databází jsou OLAP databáze určeny pro analýzu dat, která jsou za tímto účelem uložena v multidimenzionální podobě. Tato struktura zahrnuje již předzpracované agregace dat.

Data v datovém skladu jsou sice čistá, konsolidovaná, ale často velmi objemná. Abychom byli schopni tato data efektivně analyzovat, často se ukládají do speciálních datových struktur typu OLAP. Technologie OLAP nám dává možnost prohlížet sumarizované údaje o nákupech, prodejích, apod. v rozpadu podle libovolné dimenze či kombinace více dimenzí (např. časová nebo produktová dimenze) a na libovolném stupni agregace (např. stát – region – okres – město – pobočka) a to v reálném čase bez nutnosti čekat na zdlouhavé procházení celé historie datového skladu. Tímto dostáváme do rukou výkonný analytický a reportovací nástroj oblíbený zejména u manažerů. [4]

## 3.2 Uložení dat v OLAP databázích

Existuje několik způsobů, jakými lze data v OLAP databázích ukládat. Liší se hlavně velikostí výsledné databáze a způsobem, jakým mohou být tato data využívána. Jedná se o následující varianty:

> MOLAP (Multidimensional OLAP) – asi vůbec nejrozšířenější způsob uložení dat, který nabízí nejvyšší dotazovací výkon ze všech zde uvedených metod. Data jsou uložena v optimalizované multidimenzionální databázi, kde se nachází všechny potřebné agregace.

- ROLAP (Relational OLAP) poskytuje uživatelům multidimenzionální zobrazení dat, která však zůstávají uložena v původní relační databázi, což poskytuje vyšší úroveň škálovatelnosti a rychlejší dobu odezvy.
- HOLAP (Hybrid OLAP) hybridní uložení dat, které je kombinací předchozích dvou variant a snaží se maximalizovat jejich výhody. Data jsou ponechána v původních relačních tabulkách a agregace jsou uloženy v multidimenzionální podobě. Toto řešení tedy poskytuje propojení mezi velkými objemy dat v relačních tabulkách a zároveň nabízí výhodu rychlejšího zpracování multidimenzionálních agregací.
- DOLAP (Desktop OLAP) nejmladší technologie, která umožňuje uživateli stáhnout si požadovanou podmnožinu kostky z OLAP databáze na lokální disk a provádět nad ní analytické operace

# 3.3 Struktura OLAP databáze

Tabulky vytvořené v OLAP databázích jsou oproti transakčním databázím uloženy v denormalizované podobě, nesplňují tedy podmínky třetí normální formy. Tyto tabulky lze rozdělit na dva základní druhy:

**Tabulky faktů** - jedná se o nejobjemnější tabulky v databázi. Pod pojmem fakta (measures) si můžeme představit číselná vyjádření měrných jednotek obchodování. Jako příklad lze uvést počet prodaných kusů daného zboží, případně zisk z tohoto prodeje. Kromě měrných jednotek obsahují tyto tabulky ještě cizí klíče tabulek dimenzí, pomocí kterých jsou k nim dimenze napojené.

**Tabulky dimenzí** – obsahují logicky nebo organizačně uspořádané údaje, které popisují různé aspekty obchodování. Prakticky jde o jakousi obdobu podnikových číselníků, protože jejich účel i obsah jsou velmi podobné.

"Tabulky dimenzí vysvětlují všechna "proč" a "jak", pokud jde o obchodování a transakce prvků. Zatímco dimenze obecně obsahují relativně stabilní data, dimenze zákazníků se aktualizují častěji. Nejčastěji se používají časové, produktové a geografické dimenze." [9, s. 388]

Tabulky faktů a dimenzí mohou tvořit různá topologická uspořádání, která se liší především rozdílností implementace hierarchií v jednotlivých dimenzích. Více viz. následující podkapitoly.

### 3.2.1 STAR schéma

Jedná se o tzv. "*hvězdicové schéma"*, které se skládá z centrální tabulky faktů a okolních dimenzí. Každá dimenze je vždy tvořena právě jednou tabulkou dimenzí a mezi těmito tabulkami nenajdeme žádné vzájemné relační propojení. Dimenze v tomto schématu nejsou normalizovány, což často logicky vede k redundantnímu uložení dat a větší zátěži dimenzí samotných. Tato denormalizace je zapříčiněna především výskytem hierarchického uspořádání dat v jednotlivých dimenzích.



Obrázek č. 5: STAR schéma, vlastní zpracování

Toto schéma má následující výhody a nevýhody [9, s. 390]:

- Výhody vysoký dotazovací výkon
- Nevýhody v důsledku nenormalizovaných dimenzí je vytvoření takového modelu relativně pomalé

### 3.2.2 SNOWFLAKE schéma

Toto schéma také označováno jako tzv. "*sněhová vločka*". Na rozdíl od hvězdy má o něco složitější strukturu, protože každá dimenze je obvykle tvořena více vzájemně propojenými dimenzionálními tabulkami, které jsou normalizovány za účelem sníženi redundance dat v jednotlivých dimenzích. Výsledný nárůst velikosti databáze není tedy tak rapidní jako v případě hvězdicového schématu.



Schéma sněhové vločky lze znázornit následujícím příkladem:

Obrázek č. 6: SNOWFLAKE schéma, vlastní zpracování

I toto schéma má své následující výhody a nevýhody [9, s. 390]:

- Výhody rychlé zavedení údajů do normalizovaných tabulek
- Nevýhody v důsledku nenormalizovaných dimenzí je vytvoření takového modelu relativně pomalé

# 3.3 OLAP datová kostka

Pojem OLAP kostka (OLAP cube) vychází přímo z koncepce multidimenzionální databáze. Prakticky se dá říct, že jedna taková databáze může být základem pro více datových kostek (kostka je tedy podmnožinou OLAP databáze) a záleží pouze na pohledu, ze kterého na data v ní obsažená nahlížíme. Toto pojmenování vychází ze skutečnosti, že se jedná o vícerozměrnou tabulku, jejíž struktura je tvořena daty, která pocházejí z jedné nebo více tabulek faktů a informacemi prezentovanými formou dimenzí.

Podle L. Lacka [9, s. 390] je definice OLAP kostky následující: "Vícerozměrná kostka jako soubor skupin měřítek je prostředkem pro kombinace faktů s různou granularitou. Z hlediska interpretace se jedná o jakousi virtuální strukturu či kontejner pro skupinu měřítek, dimenzí, politik, nebo jinak řečeno, je to spíše logická struktura nežli fyzické údaje."

Díky multidimenzionální struktuře datové kostky máme možnost pohlížet na realitu z několika možných úhlů pohledu. Pro lepší názornost nám poslouží následující obrázek:



Obrázek č. 7: Princip datové kostky, převzato z [11]

Datová kostka, znázorněná na obrázku č. 7, je pro snadné pochopení této problematiky tvořena třemi dimenzemi. V praxi však žádné omezení pro počet dimenzí v kostce neexistuje a není tedy neobvyklé, aby byla tvořena třeba i několika desítkami dimenzí. Grafické znázornění takovéto struktury by bylo velmi složité a nepřehledné.

## 3.3.1 Operace spojené s OLAP kostkou

Protože se v případě datové kostky jedná o vícerozměrnou strukturu, musí také existovat způsob, jakým na požadovaná data nahlížet a analyzovat je. Pro tuto potřebu existují speciálně navržené operace, pomocí kterých lze s kostkou pracovat. Jedná se o tyto následující operace:

- Drill-down umožnuje sestoupit na nižší úroveň agregace a dostat se tak k více detailním informacím.
- Roll-up jde o opak operace drill-down, který umožňuje přechod do vyšších agregačních úrovní z úrovně aktuální.
- **Pivoting** umožňuje "otáčet" datovou krychlí, tj. měnit úhel pohledu na data. Díky této operaci lze pohlížet na data z pohledu různých dimenzí.
- Slicing umožňuje provádět tzv. "řezy" kostkou, tj. nalézt pohled, v němž je jedna dimenze fixována v určité agregační úrovni. Jinými slovy, tato dimenze aplikuje filtr na instance příslušné agregační úrovně dané dimenze.
- Dicing jedná se o obdobu slicingu, jenž umožňuje nastavit filtr pro více dimenzí.

# 3.4 Reporting

Reporting se dá popsat jako činnost, která slouží k získávání a interpretaci informací, jež jsou vyžadovány firmou pro podporu rozhodování. Nosným základem reportingu obvykle není nic jiného, než dotazování se do databáze pomocí jejího standardního rozhraní. Přestože je často reporting prováděn i nad transakčními databázemi, rozhodl jsem se jej zařadit do této kapitoly, protože se přeci jen jedná o analytický typ činnosti a má tak s aktuálně probíranou tematikou mnoho společného.

Ať už chceme znát stav a vývoj kritických aspektů ve firmě nebo efektivitu podřízených a výroby produktů, nabízí se nám rozdílné situace z pohledu množství potřebných dat, rychlosti rozhodovacího procesu a potřeby plánování. Co je však v těchto i ostatních situacích společné je to, že se potřebujeme rychle a správně rozhodnout, aniž bychom znali celý proces Business Intelligence. Důležitý je pro nás až validní výsledek. Reportingem tak lze nazvat právě tento poslední stupínek celého procesu. Prostřednictvím reportingu získá oprávněná osoba informace ve formě, která je

mu graficky příjemná, funkcionálně dostatečná a se snadnou obslužností tak, aby nebyl uživatel zdržován a zahlcován úkony, které nejsou bezpodmínečně nutné. [5]

Existuje několik typů reportingu, které se liší způsobem vytvoření, využitím a svým přínosem [5]:

### Statický reporting

Je vhodný zejména pro vizualizaci informací standardní struktury a vzhledu s takřka neměnnými vstupními parametry. Hodí se dobře pro finanční výkaznictví, přehledy o prodejích produktů nebo automaticky pravidelně zasílané reporty e-mailem. Výhodou tohoto typu je, že spotřebitel informací získá informace "jedním klikem".

### Dynamický reporting

Jeho charakter je podobný statickému reportingu, s tím rozdílem, že uživatel může ovlivňovat obsah a formu reportu zadáváním vstupních parametrů. Tento typ je vhodný pro přehledy vztahující se k předem neznámým časovým obdobím, kategoriím produktů a zákazníků, v případě potřeby lze částečně ovlivnit i design a formu samotného reportu. Výhodou dynamického reportu je přizpůsobení reportu potřebám konkrétního uživatele.

### Ad hoc reporting

Jestliže si uživatel nevystačí s jedním z výše zmíněných typů reportů, má možnost si vytvořit report dle vlastních požadavků. Ad hoc reporty se hodí v situacích, kdy je těžko dopředu určit, jaký obsah a formu má daný report splňovat, případně tyto informace ještě vůbec nejsou známy. Výhodou ad hoc reportů je jejich nezávislost na vývojářích reportovacích systémů (uživatel si může report vytvořit sám) a možnost vytvořit si report až ve chvíli, kdy jeho potřeba nastane a budou známy všechny informace potřebné k definici reportu.

# 4 Význam Business Intelligence pro malý a střední podnik

Nyní se dostávám k závěrečné kapitole teoretické části. Doposud jsem se zabýval vysvětlením pojmu Business Inteligence a podrobným popisem jeho jednotlivých komponent a procesů, které díky své provázanosti tvoří jeho vrstvy. Na základě těchto teoretických poznatků bych rád nastínil možnosti jeho využití v podmínkách malých a středních podniků, protože za svou poměrně krátkou historii existence se tento pojem stal výsadou především velkých společností. Není se ani čemu divit, protože právě tyto společnosti většinou disponují velkými objemy dat, pro jejichž analýzu a následné přetvoření ve smysluplné informace hraje BI klíčovou roli. Správné zavedení technologie BI totiž přináší většině velkých podniků i přes nezanedbatelné počáteční náklady do její implementace zvýšení celkového obratu a ziskovosti. Zda se však zavedení technologie BI vyplatí i pro podniky menšího charakteru je předmětem této kapitoly.

Jako první předpoklad pro implementaci BI by si měl podnik uvědomit, zda dokáže nashromáždit dostatečné množství dat, aby mělo smysl je dále analyzovat a hledat v nich skryté závislosti [3]. Jestliže je objem podnikových dat v tak malém rozmezí, že je možné z nich vyvozovat dostatečně přesné závěry bez nutnosti využití pomocných nástrojů k tomuto účelu navržených, nebude zde patrně důvod, aby se podnik touto otázkou nadále zabýval.

Dalším důležitým aspektem jsou omezené finanční prostředky, které jsou tyto firmy obvykle schopny vynaložit na realizaci takovýchto řešení. To ovšem automaticky neznamená, že by se neměly pokoušet alespoň určitou část BI technologií implementovat. Nakonec není přeci ani nutné, aby se vydávaly směrem komplexního budování datového skladu, jehož plný potenciál by vzhledem k jejich potřebám pravděpodobně zůstal nevyužitý (nemluvě o velké finanční a časové náročnosti takového projektu). Tento problém interpretuje ve své dokumentaci firma SAP následovně:

"Organizace středních velikostí mají omezené zdroje. Řešení BI by mělo nabídnout nízké náklady na vlastnictví prostřednictvím standardní integrace. Dále by mělo poskytnout šablony a nástroje, které zjednodušují a urychlují vytváření vlastních výkazů, dotazů a dashboardů, a eliminují tak potřebu najímat drahé externí konzultanty. Řešení BI by mělo zahrnovat důležité komponenty, aby nebylo nutné nakupovat další software." [13] Výběr vhodného řešení tak závisí na důkladné analýze těch potřeb, které jsou pro konkrétní podnik nejzásadnější. Na jejich základě je pak třeba formulovat požadavky pro aplikaci odpovídajícího řešení, které může být v budoucnu rozšířeno o další prvky. Různé komerční varianty, zaměřené pro potřeby malých a středních podniků, dnes nabízí již většina hlavních dodavatelů těchto služeb, jako např. Microsoft, Oracle, IBM, SAP, SAS a další.

V současnosti již existují i poskytovatelé několika open source BI řešení (např. JasperSoft, Pentaho), která se možná nemohou svou komplexností zcela rovnat jejich výše zmíněným konkurenční produktům, pro potřeby menších podniků však mohou být zcela dostačující.

Musím přiznat, že já sám jsem byl ještě poměrně nedávno skeptický vůči významu BI pro menší podniky, aniž bych se nad tím nějak do hloubky zamýšlel. Byl jsem přesvědčen, že přínos BI technologií bude u těchto podniků zanedbatelný v porovnání s investovaným časem a kapitálem. Tato investice je však úměrná rozsahu konkrétního řešení a logicky tak bude pro potřeby menšího podniku nižší nebo téměř nulová. To ostatně potvrzuje již výše zmíněná nabídka komerčních i open-source řešení, která odpovídá požadavkům této cílové skupiny.

Osobně si tedy myslím, že určitý potenciál Business Intelligence pro malé a střední firmy zde je a záleží jen na nich, jakým směrem se vydají.

# II. Praktická část

# 5 Nástroje pro tvorbu cvičných úloh

Než přejdu k samotnému návrhu úloh, pokusím se v krátkosti představit všechny prostředky, pomocí kterých budou tyto úlohy vznikat. Jak už jsem zmínil v úvodu práce, rozhodl jsem se využít nástroje, které nabízí platforma Microsoft SQL Server 2008 R2 Enterprise. V současné době je již k dispozici i nejnovější verze s označením Microsoft SQL Server 2012. Ta však byla uvedena až v průběhu psaní této práce, proto se budu i nadále držet předchozí verze, se kterou mám já i ostatní studenti větší zkušenosti.

Pomocí výše zmíněné platformy lze realizovat i ta nejkomplexnější řešení zahrnující činnosti od vybudování datového skladu, jeho postupného plnění novými daty až po jejich využití pro analýzu a firemní reporting, které jsou hlavním předmětem cvičných úloh zaměřených na Business Intelligence. Tuto platformu jsem zvolil především z toho důvodu, protože je ve školní výuce využívána její odlehčená verze Microsoft SQL Server 2008 Express, která slouží pro cvičné dotazování nad školní databází prostřednictvím jazyka SQL. Logicky se tak jedná o nejvhodnější volbu, protože studenti jsou s touto platformou již obeznámeni přímo z výuky a budou se tedy moci soustředit výhradně na rozšiřující funkce, se kterými se doposud v Express verzi nesetkali.

Dále bych chtěl pro svou práci využít možností, které nabízí ve své poslední verzi aplikace Microsoft Excel 2010. Z hlediska dostupnosti je MS Excel jedním z nejrozšířenějších nástrojů pro analýzu dat a proto by jeho využití nemělo chybět ani zde. Nespornou výhodou tohoto nástroje je i fakt, že se společně s platformou MS SQL Server jedná o produkty stejné společnosti, což má celkový dopad i na jejich vzájemnou kompatibilitu.

# 5.1 SQL Server Management Studio

Jedná se o komplexní sadu nástrojů určených pro správu databázového serveru, která umožňuje přístup k OLTP i OLAP databázím. Prostřednictvím těchto nástrojů lze programovat nové databáze v jazyce SQL (případně T-SQL), prohlížet jejich obsah a provádět různé úkony spojené s administrací, jako např. nastavování přístupových práv

a přidělování oprávnění pro práci s datovými strukturami. Jednotlivé nástroje jsou koncipovány jako služby, které běží na pozadí.

### **5.1.1 Database Engine**

Database Engine (tzv. "databázový stroj") tvoří základ celé platformy, protože umožňuje spravovat obrovské objemy dat, která jsou často integrována ze všech částí podniku do datového skladu, kde mohou být následně reportována nebo podrobována nejrůznějším analýzám.

Z hlediska cvičných úloh bude mít tato služba význam především pro připojení ke školní databázi a naprogramování skriptu, pomocí kterého budou na základě již existujících tabulek vytvořeny tabulky nové. Hlavním účelem skriptu bude vytvoření předpřipravené struktury, která bude odpovídat multidimenzionálnímu uložení dat.

#### 5.1.2 Integration Services

Tato služba představuje na platformě MS SQL Server 2008 základ pro ETL zpracování dat. Slouží k integraci dat ze zdrojových systémů do datového skladu a celkově se jedná o poměrně komplexní nástroj, jehož důkladný rozbor by vydal na samostatnou práci. Tato práce se však zabývá problematikou analýzy a reportování dat, která jsou již v cvičné databázi uložena. Proto není třeba tuto službu nadále rozebírat, jelikož nebude mít význam pro návrh jednotlivých úloh.

#### 5.1.3 Analysis Services

Služba SQL Server Analysis Services poskytuje integrované zobrazení obchodních dat pro účely správy vytvořených sestav, analýzy OLAP, přehledů klíčových ukazatelů výkonu (KPI) a dolování dat.

Pomocí této služby si v krátkosti na jedné z úloh ukážeme, jaké jsou možnosti analýzy dat z cvičné databáze a jak pracovat s vytvořenou OLAP datovou kostkou.

### 5.1.3 Reporting Services

Již z názvu lze snadno usoudit, že úlohou této služby je zabezpečení požadavků týkajících se reportování dat nad databázemi. Hlavním účelem služby je generování výstupních sestav (v elektronické nebo papírové podobě), což není obvykle nic jiného, než výsledek konkrétního SQL dotazu. Samotným návrh dotazu a zprostředkování jeho výsledku však často nestačí, jelikož nedílnou součástí této problematiky je i vizuální

styl, jakým je daný report prezentován. Grafická podoba reportu je totiž v některých případech (např. reporty určené pro vrcholové vedení podniku) stejně důležitá jako samotný výsledek dotazu.

Jedna z cvičných úloh bude zaměřena právě na tento nástroj a na možnosti, které nabízí. Bude v ní obsažen kompletní návrh reportu (ve vývojovém prostředí BIDS) včetně zajištění jeho odpovídající logické stavby a vizuálního vzhledu.

Serverture	
Server name: Authentication: Login:	Database Engine Analysis Services Reporting Services SQL Server Compact Integration Services
Password:	

Obrázek č. 8: SQL Server Management Studio – nabídka služeb

## 5.2 SQL Server Business Intelligence Development Studio

Jedná se o vývojový nástroj, který je založen na funkcionalitě oblíbené aplikace Microsoft Visual Studio .NET, na rozdíl od ní však umožňuje vytvářet projekty zaměřené na integrační, analytické a reportovací služby. Jeho hlavní silnou stránkou je návrh a testování integračních algoritmů a vizuální modelování datových kostek a reportů.

Ze všech již zmíněných nástrojů bude v cvičných úlohách zastoupeno využití BI Development Studia největší měrou, jelikož bude sloužit jako pomůcka pro vytvoření OLAP datové kostky, ale také pro návrh pokusného reportu včetně jeho adekvátní grafické úpravy.

# 5.3 Microsoft Excel 2010

Asi není třeba tento nástroj, který je součástí kancelářského balíčku Microsoft Office, příliš představovat. Tento tabulkový kalkulátor je jedním z nejrozšířenějších nástrojů zabývajících se zpracováním velkých objemů dat, kontingenčními tabulkami, grafy a dalšími užitečnými funkcemi. Především ve svých posledních verzích doznal výrazných změn v možnostech analýzy a vizualizace dat, které naleznou uplatnění v této práci.

Jako příklad nových funkcí můžu uvést možnost přímého připojení k databázovému serveru a následné analýzy dat bez nutnosti využití jakéhokoliv dalšího nástroje pro sestavení potřebného dotazu, což bude také náplní jedné z cvičných úloh.

# 6 Školní databáze

Cílem šesté kapitoly je představení školní cvičné databáze, jelikož právě ona je hlavním zdrojem, z jehož základů budou vycházet jednotlivé úlohy určené pro analýzu a reportování dat. Jedná se o standardní relační OLTP databázi, jejíž koncepce je založena na poměrně běžném obchodním modelu, který představuje firemní prodej určitých výrobků svým cílovým zákazníkům.



Obrázek č. 9: ER diagram cvičné databáze, vlastní úprava

Jak je vidno z ER (Entity-Relationship) diagramu, databáze je tvořena několika entitami, které jsou navzájem provázány vazbami, jež jsou zastoupeny nejběžnějšími typy relací N:1 a M:N.

Databáze slouží jako výuková pomůcka v předmětech Databázové systémy 1 a Databázové systémy 2. Studenti si zde mohou procvičit své znalosti jazyka SQL při psaní jednoduchých i složitých dotazů, vytváření vlastních tabulek a pohledů nebo programování procedur a triggerů pomocí procedurálního rozšíření T-SCRIPT.

# 7 Návrh cvičných úloh

Protože jsme si již představili cvičnou databázi i jednotlivé nástroje, které budou využity pro tvorbu cvičné sady úloh, můžeme přistoupit k formulaci a řešení úloh samotných. Jednotlivé úlohy budou koncipovány jako návrh možného postupu, kterým lze dosáhnout jejich úspěšného vyřešení. Vždy se tak bude jednat o jakýsi "modelový" příklad. V konečném důsledku však není požadováno (ba právě naopak), aby byl tento postup striktně dodržován. Spíše by měl studentům sloužit jako inspirace pro jejich vlastní řešení.

# 7.1 Úloha č. 1: OLAP transformace dat

Cílem první úlohy je předvést, jak lze transformovat data z existující OLTP databáze do takové podoby, která bude odpovídat multidimenzionálnímu uložení dat.

### 7.1.1 Formulace problému

Jak již bylo řečeno, cvičná databáze má standardní OLTP strukturu, která je primárně určena pro transakční zpracování dat. To však není zcela ideální v situacích, kdy budeme chtít tato data podrobněji analyzovat. Proto se pokusíme data v ní uložená transformovat do OLAP podoby. Pro tento účel bude vytvořen SQL skript, pomocí kterého se vytvoří a naplní nová tabulka faktů a jestliže to bude nadále nutné, tak bude vytvořen i potřebný počet tabulek dimenzionálních. Nově vytvořené tabulky se stanou základem pro další operace v následující úloze.

### 7.1.2 Postup řešení

Existuje více možností, jak se dá tato úloha řešit. Se vzrůstající velikostí databáze a počtem jejích tabulek roste i počet možností, jakým na data v nich uložená nahlížet. Podle těchto skutečností se nadále odvíjí rozhodnutí, jaké zaměření bude mít výsledná tabulka faktů a jaké měrné jednotky (fakta) si zvolíme pro účel analýzy. Pokud máme již rozmyšleno, co by měla tabulka faktů obsahovat, můžeme přejít k volbě dimenzí, dle kterých si budeme moci fakta filtrovat.

### Volba potřebných tabulek

Vzhledem k zaměření cvičné databáze na obchodování mezi firmou a jejími zákazníky bude patrně pro vedení firmy stěžejní analýza počtu prodaných výrobků a objemu tržeb. Vytvoříme si tedy tabulku s názvem *Prodeje*, která bude pro naše potřeby představovat ústřední tabulku faktů. Dále bude následovat volba požadovaných

dimenzí. Pro své řešení jsem zvolil již existující tabulky *Zamestnanci*, *Zakaznik* a *Produkty*. Tyto tabulky bude možno využít přímo v následující úloze jako základ pro jednotlivé dimenze.

V příkladu by však neměla chybět ani dimenze časová, protože pohled na prodeje je pro podnik z hlediska plynoucího času velice důležitý. V cvičné databázi se čas objevuje v tabulce *Objednavky*, kde je reprezentován pomocí atributů *datum\_prijeti* a *datum\_odeslani*. Datum přijetí objednávky nemusí ještě nutně znamenat, že k samotnému prodeji položek na objednávce dojde (zákazník může objednávku např. zrušit, položka na objednávce nemusí být dostupná skladem atd.), proto bude pro časovou dimenzi využit atribut představující datum odeslání, které je přeci jen více závazné.

### Skript pro vytvoření a naplnění tabulek

Nyní již můžeme přistoupit k samotnému vytvoření SQL skriptu. V Microsoft SQL Management Studio se připojíme ke školnímu serveru pomocí služby Database Engine, vybereme si databázi Cviceni\_create a můžeme začít psát. V první řadě si přednostně vytvoříme tabulku *Cas*, protože se na ni bude odkazovat faktová tabulka *Prodeje* pomocí cizího klíče.

```
CREATE TABLE Cas
(
datum DATE PRIMARY KEY,
rok SMALLINT NOT NULL,
kvartal TINYINT NOT NULL,
mesic VARCHAR(10) NOT NULL,
den VARCHAR(10) NOT NULL
);
```

#### Obrázek č. 10: Vytvoření časové dimenzionální tabulky

Skriptem, uvedeným na obrázku č. 10, si vytvoříme prázdnou časovou tabulku, kterou naplníme následujícím způsobem:

```
INSERT INTO Cas

SELECT DISTINCT datum_odeslani AS datum,

YEAR(datum_odeslani) AS rok,

Datename(q, datum_odeslani) AS kvartal,

Datename(mm, datum_odeslani) AS mesic,

Datename(dw, datum_odeslani) AS den

FROM objednavky
```



Obsah naplněné časové tabulky bude díky skriptu z obrázku č. 11 tvořen daty, která vzniknou rozpadem původního atributu (*datum\_odeslani*) na časovou hierarchii ve formátu *rok – kvartál – měsíc – den*. Výhody této hierarchie si ukážeme v následující úloze, až budeme mít vytvořenou OLAP datovou kostku.

Protože jsme si již definovali, jaké tabulky použijeme pro tvorbu jednotlivých dimenzí, můžeme přistoupit k návrhu tabulky faktů. Její sloupce budou tvořeny jednak měrnými jednotkami, ale také cizími klíči, které se budou odkazovat do příslušných tabulek dimenzí. Návrh faktové tabulky vypadá následovně:

#### CREATE TABLE Prodeje

1

	id_prodeje	INT PRIMARY KEY IDENTITY,
	datum	DATE REFERENCES cas(datum) ON DELETE SET NULL,
	id produkt	INT REFERENCES produkty(id_produkt)ON DELETE SET NULL,
	id_zakaznika	INT REFERENCES zakaznik(id_zakaznika)ON DELETE SET NULL,
	id zamestnance	INT REFERENCES zamestnanci(id zamestnance)ON DELETE SET NULL,
	kusy	INT,
	trzba	FLOAT
1 .		

#### Obrázek č. 12: Vytvoření faktové tabulky

Vytvořenou faktovou tabulku například naplníme takto:

```
INSERT INTO Prodeje
SELECT o.datum odeslani,
       p.id produkt,
       o.id zakaznika,
       o.id zamestnance,
       SUM(od.mnozstvi),
       SUM(od.mnozstvi * p.cena)
FROM
       objednavky o
       INNER JOIN objednavka_detail od
         ON o.id objednavky = od.id objednavky
       INNER JOIN produkty p
         ON od.id produkt = p.id produkt
GROUP BY o.datum odeslani,
          p.id produkt,
          o.id zakaznika,
          o.id zamestnance
```

#### Obrázek č. 13: Naplnění faktové tabulky

Ze skriptu na obrázku č. 13 lze vyčíst, že nová faktová tabulka je vytvořena na základě tabulek *Objednavky*, *Objednavka\_detail* a *Produkty*. Protože se ale v této úloze zajímáme o celkové počty prodaných kusů zboží a jim odpovídající tržby, nebude pro nás až tak podstatné, co bylo obsahem jednotlivých objednávek. Další tabulky již

vytvářet nemusíme, protože základ, který jsme si připravili, bude pro naše cvičné zaměření zcela dostačující.

# 7.1.3 Shrnutí

V této úloze jsme si ukázali, jak se dají data poměrně jednoduchým a názorným způsobem transformovat do multidimenzionální podoby, která bude sloužit jako výchozí stav pro zpracování úlohy následující.

# 7.2 Úloha č. 2: Tvorba datové kostky

Druhá úloha bude zaměřena na OLAP datovou kostku, pro jejíž vytvoření nám poslouží tabulky, které jsme si navrhnuli již v úloze č. 1. Pro tyto účely využijeme Microsoft SQL Server Business Intelligence Studio (dále jen BIDS), kde si ukážeme, jak kostku nejen vytvořit, ale také umístit pod správu analytických služeb (Analysis Services). Co do rozsahu se bude jednat o nejobsáhlejší úlohu, protože samotný proces návrhu datové kostky se skládá z několika dílčích operací, které bude třeba popsat a předvést na názorném příkladu.

## 7.2.1 Formulace problému

Představme si situaci, kdy bude chtít jistá firma pro určitou oblast svého zaměření urychlit a usnadnit proces zisku informací ze své OLTP databáze. Standardně u ní tento proces probíhá tak, že při každém požadavku musí být sestaven nový dotaz, což není časově příliš efektivní. Na základě parametrů jednotlivých dotazů je vždy proveden individuální výpočet, který zprostředkuje výsledek. Východiskem tohoto problému může být zavedení OLAP databáze (v našem případě máme již díky úloze č. 1 základ OLAP databáze vytvořený). Následně může být na základě této databáze vytvořena jedna nebo i více datových kostek, které díky předem agregovaným datům znají odpověď na tyto otázky ještě dříve, než jsou položeny.

# 7.2.2 Postup řešení

Protože jsme si v předchozí úloze již ujasnili, jaké tabulky budeme chtít využít pro návrh datové kostky, můžeme pomocí vývojového nástroje BIDS přistoupit k založení nového analytického projektu.

New Project		2
Project types:	Templates:	.NET Framework 3.5 🔹 🖬
Business Intelligence Projects	Visual Studio installed templates	
Other Project Types	Analysis Services Project	🞲 Import Analysis Services Database
	Integration Services Connections Proje	🚯 Integration Services Project
	Report Server Project Wizard	Report Model Project

#### Obrázek č. 14: Výběr typu projektu v BIDS

Vybraný projekt si pojmenujeme a potvrdíme jeho vytvoření, čímž se přepneme do prostředí, ve kterém následně můžeme začít na přípravných pracích pro modelování datové kostky. Jednotlivými kroky návrhu nás provede okno *Solution Explorer*  (klávesová zkratka Ctrl+Alt+L), kde pro nás budou v rámci této úlohy podstatné záložky, které jsou vyznačené na následujícím obrázku.



Obrázek č. 15: Okno Solution Explorer

## Volba datového zdroje

Záložka *Data Sources* je určena k výběru zdroje, ze kterého budou vybrána potřebná data pro tvorbu datové kostky. Logicky se tedy budeme chtít připojit ke školnímu databázovému serveru, na němž je umístěna cvičná databáze *Cviceni\_create*, ve které máme připraveny všechny potřebné tabulky.

	S			A10804 CI
tion	klimanek.sks.cz		•	Refres
ſ	Log on to the ser Use Windows Use SQL Serve User name: Password:	ver s Authentication er Authentication student		
		Save my password		
	Connect to a data	abase		
	Select or ente	er a database name:		

Obrázek č. 16: Výběr zdroje dat pro OLAP kostku

### Vytvoření datového pohledu

Jestliže máme nadefinovaný datový zdroj, můžeme přejít k další záložce s názvem Data Source Views, která slouží k výběru požadované podmnožiny dat z tohoto zdroje. V praxi totiž existují databáze, které řádově čítají i stovky tabulek, jejichž podstatná část nemusí mít pro návrh konkrétní datové kostky význam. Proto je vhodné vytvořit datový pohled, díky němuž omezíme počet tabulek jen na ty, které skutečně využijeme.

Pomocí záložky *Data Source Views* tedy přejdeme k tvorbě nového pohledu. Nejprve potvrdíme, že chceme využít datový zdroj, který jsme si pro tento účel definovali v předchozím kroku. Poté se nám zobrazí okno obsahující seznam všech dostupných tabulek v tomto zdroji, z nichž si vybereme pouze ty, které budeme chtít použít pro vytvoření datové kostky. Pro výběr zvolíme nejprve tabulku *Prodeje*, protože ta nám v našem schématu poslouží jako centrální tabulka faktů. Následně vybereme tabulky, které budou tvořit jednotlivé dimenze: *Cas, Produkty, Zakaznik, Zamestnanci*. Tento výběr by nám teoreticky mohl stačit, abychom se však mohli na data v budoucí kostce dívat co nejdetailněji, přidáme si do našeho výběru ještě tabulky *Oddeleni* a *Kategorie*, díky nimž nám bude umožněn hierarchický pohled na data v dimenzích *Zamestnanci* a *Produkty*.



#### Obrázek č. 17: Výběr tabulek pro datový pohled

Výběr tabulek potvrdíme a dle vlastního uvážení si pohled pojmenujeme, čímž dojde k jeho automatickému vytvoření a vizuálnímu znázornění.



Obrázek č. 18: Vizualizace datového pohledu

Výsledné schéma má v této názorné ukázce díky přidání tabulek *Oddeleni* a *Kategorie* podobu sněhové vločky. Záleží však na samotných studentech, jaké tabulky si vyberou pro vypracování svých vlastních řešení. Jejich schéma tak může být případně i hvězdicové.

### Definice dimenzí

Dalším krokem pro tvorbu datové kostky je specifikace nových dimenzí pomocí záložky *Dimensions* v okně *Solution Explorer*. Tato volba je užitečná především v situacích, kdy bychom chtěli ke svému návrhu připojit tabulku z jiného datového zdroje a následně z ní vytvořit dimenzi. V tomto konkrétním případě máme již rozmyšleno, které tabulky budou součástí jednotlivých dimenzí a navíc nemáme k dispozici více datových zdrojů, které by se daly využít. Záložka *Dimensions* však poskytuje i poměrně komfortní nástroj zaměřený na tvorbu časových dimenzí, což si mohou studenti vyzkoušet jako alternativu k vytvoření časové tabulky pomocí skriptu v úloze č. 1.

### Návrh datové kostky

Protože jsme v našem případě neměli potřebu specifikovat nové dimenze, můžeme přejít k samotnému návrhu datové kostky pomocí záložky *Cubes*. Zde vybereme možnost využití již vytvořených tabulek a přejdeme k výběru faktové tabulky.

Cviceni Create pohled	
Measure group tables:	Suggest
🛛 🏢 Prodeje	
Zamestnanci	
Zakaznik	
Cas	
Produkty	
🔲 🔲 kategorie	
Oddeleni	

#### Obrázek č. 19: Výběr faktové tabulky

Po určení faktové tabulky následuje výběr měrných jednotek. Kromě námi navržených atributů *Kusy* a *Trzba* nám aplikace nabízí vytvoření agregované měrné jednotky *Prodeje Count*, která počítá uskutečněné prodeje, což může být užitečné. Označíme tedy všechny tři nabízené možnosti a můžeme pokračovat dále.



#### Obrázek č. 20: Volba měrných jednotek

Nyní už jen zbývá zvolit jednotlivé dimenze a první část vytváření datové kostky je téměř hotova. Zde jsem se však setkal s problémem, jehož příčinu se mi nepodařilo odhalit. I když měly být dle mého návrhu vytvořeny celkem čtyři dimenze, z toho dvě pomocí jedné tabulky a zbylé dvě pomocí tabulek dvou, průvodce se mi snažil vnutit dimenzí celkem pět. Kromě tabulek *Cas*, *Produkty*, *Zakaznik* a *Zamestnanci*, které jsem v podobě dimenzí ve výsledné datové kostce skutečně chtěl, byla jako samostatná dimenze vyhodnocena také tabulka *Kategorie*, přestože není na faktovou tabulku přímo

napojena. Místo toho však měla být napojena na tabulku *Produkty* a být tak rozšiřující tabulkou produktové dimenze. To ale pro nás není zas až tak velká překážka, protože si budeme moci tuto tabulku o něco později připojit k produktové dimenzi ručně, takže ji nyní ve výběru dimenzí necháme neoznačenou (viz. následující obrázek).



#### Obrázek č. 21: Volba dimenzí

Poté, co jsme se postarali o všechny náležitosti spojené se specifikací dimenzí, faktové tabulky a jejích měrných jednotek, můžeme datovou kostku pojmenovat a potvrdit tyto změny. To ale ještě není zdaleka konečná fáze, protože i nadále se jedná o pouhou koncepci a kostka tak stále nemá fyzickou podobu. To však zatím ani není naším cílem, protože máme před sebou ještě několik dalších úprav, které bude nutno provést.

Jestliže jsme úspěšně prošli celým procesem definice datové kostky v záložce *Cubes*, pak se nám zároveň v záložce *Dimensions* zobrazily jednotlivé dimenze.



Obrázek č. 22: Solution Explorer - Rekapitulace dosavadní tvorby

### Definice hierarchií

Nyní můžeme přistoupit k úpravám vytvořených dimenzí. Tyto úpravy budou spojené především s definováním hierarchií, pomocí kterých budeme schopni pohodlněji provádět operace *roll-up* a *drill-down*.

Jako první si otevřeme dimenzi *Produkty*, ke které si navíc připojíme již zmíněnou tabulku *Kategorie*, jelikož se nám to nepovedlo pomocí průvodce při návrhu kostky samotné. Po otevření této dimenze se nám zobrazí v záložce *Dimension Structure* tři okna s názvy *Attributes* (atributy), *Hierarchies* (hierarchie) a *Data Source View* (datový pohled). V okně *Data Source View* si naklikneme pravým tlačítkem myši tabulku *Produkty* a z nabídky zvolíme možnost *"Show Related Tables"*, čímž si zobrazíme ostatní tabulky, které s ní jsou v datovém pohledu spojeny. Tím se nám zpřístupní tabulka *Kategorie*.

Defaultně jsou mezi atributy vzniklé dimenze zastoupeny pouze primární a cizí klíče jejich tabulek. To nám samo o sobě příliš nepomůže, proto si dle uvážení tahem myši přemístíme potřebné atributy z tabulek v okně *Data Source View* k defaultním atributům do okna *Attributes*. Tím, že přetáhneme alespoň jeden atribut z tabulky *Kategorie*, docílíme automatického začlenění této tabulky do produktové dimenze. Případné duplicitní názvy atributů (v našem případě se jedná o shodný atribut *Nazev* v tabulkách *Produkty* a *Kategorie*) si můžeme v okně *Attributes* přejmenovat dle vlastního uvážení tak, aby nás jejich název nemýlil (v tomto případě přejmenováno na *kategorie* a *produkt* – viz. následující obrázek).



Obrázek č. 23: Připojení tabulky do produktové dimenze

Pokud máme vybrány všechny atributy, které budeme chtít mít v produktové dimenzi zastoupeny, pak bychom měli ještě zvážit, zda mezi některými z těchto atributů existuje jistý stupeň hierarchie. V tomto příkladu spadá každý produkt do některé

z kategorií. Na základě této skutečnosti si vytvoříme příslušnou hierarchii přetažením daných atributů z okna *Attributes* do okna *Hierarchies* následovně:



Obrázek č. 24: Definice produktové hierarchie

Relace mezi atributy této hierarchie si následně ještě upravíme v záložce *Attribute Relationships*:



Obrázek č. 25: Úprava relací mezi atributy produktové hierarchie

Obdobně, jako tomu bylo u dimenze *Produkty*, přetáhneme potřebné atributy a nadefinujeme vhodné hierarchie i u dimenzí zbývajících.

## Záložky Calculations a KPIs

Posledním krokem před vlastním vytvořením datové kostky bude definice nových vypočítaných členů a klíčových ukazatelů výkonnosti v záložkách *Calculations* a *KPIs* (Key Performance Indicators), které se nám zobrazí poté, co si otevřeme návrh kostky v okně *Solution Explorer*.

Záložka *Calculations* slouží k vypočtení nových hodnot odvozených z kombinace agregovaných měrných jednotek nebo dat v dimenzích. Tyto hodnoty fungují na principu měrných jednotek, nejsou však definovány přímo v tabulce faktů, protože jsou vypočítány až na základě výsledků, které jsou poskytnuty datovou kostkou. Jako příklad lze uvést požadavek, aby se dala pomocí kostky zobrazit průměrná tržba (vypočítá se z měrných jednotek jako *Trzba / Prodeje Count*), případně daň z přidané hodnoty, která

bude odvedena státu (při DPH = 20%: *Trzba* \* 0.2). Tyto nově vypočítané členy nadefinujeme takto:

Kostka Prodeje.cube [Design] Start Page			
🔍 Cube Structure 🎒 Dimension Usage 🆓 Calculation	ns 📑 KPIs 🔃 Actions 🆓 Par	titions 📳 Aggregations 📷 Perspe	ectives 🞑 Translations 🔍 Browser
🖻 🕄   嬰 🐸   1 🖬 (.) 🐒   × 🔹   🔳 🗉	] 🚰   💞 A 🔳 澤 🚝 🕯	=   🕭 🖩 📕 🕈 🖉 🚮	
Script Organizer	Name:		
두 🗗 Command	prumerna_trzba		
1 🧊 CALCULATE			
2 📋 prumerna_trzba			
3 🛅 DPH	Parent hierarchy:	Measures	<u> </u>
	Parent member:		Change
	★ Expression		
	[Measures].[Trzba]/	[Measures].[Prodeje Coun	t]
Calculation Tools	A Additional Boson Pro-		
Metadata 🗣 Functions 🛄 Templates	Additional Properties	1.000 C	
Measure Group:	Format string:	"0"	<b>-</b>
<all></all>	Visible:	True	<b></b> ]
🮯 Kostka Prodeje	Non-empty behavior:		•
Measures	Associated measure group:	(Undefined)	•
E Le Cas	Display folder:		
🗄 🙋 Zakaznik			
🕀 😥 Zamestnanci			

Obrázek č. 26: Definice vypočítaných členů

Případně lze nové vypočítané členy nadefinovat i pomocí skriptu následovně (předchozí způsob je však méně pracný a tudíž výhodnější):

```
CALCULATE;

CREATE MEMBER CURRENTCUBE.[Measures].prumerna_trzba

AS [Measures].[Trzba]/[Measures].[Prodeje Count],

FORMAT_STRING = "0",

VISIBLE = 1 ;

CREATE MEMBER CURRENTCUBE.[Measures].DPH

AS [Measures].[Trzba]*0.2,

FORMAT_STRING = "0",

VISIBLE = 1 ;
```

#### Obrázek č. 27: Skript pro vytvoření vypočítaných členů

Záložka *KPIs* je určena pro stanovení určitých cílů, které by měly být splněny. Díky tomu se dá jednoduše kontrolovat, jak vypadá současný vývoj dané hodnoty v porovnání s naším očekáváním. Podle toho, jak jsou tyto cíle plněny, se následně může vedení podniku rozhodovat, zda je potřeba provést potřebné změny pro zlepšení daných výsledků.

Řekněme, že v naší úloze budeme chtít sledovat průměrný počet kusů výrobků, které byly součástí jednoho prodeje, přičemž bude naším cílem, aby si u nás zákazník nakoupil v průměru alespoň tři výrobky během jednoho nákupu (v opačném případě

může být např. rozhodnuto o zavedení množstevních slev, které podpoří nákup zboží ve větším množství). Takovýto ukazatel lze vytvořit následovně:

Name:				
Pocet pre	odanych kusu na je	den prodej		
Associated r	measure group:			
Prodeje			•	
☆ Value Ex	pression			
[Meas	ures].[Kusy	]/[Measures].[Prodeje C	ount]	
∫ ☆ Goal Exp	ression			3
[Meas	ures].[Kusy	]/[Measures].[Prodeje C	ount] >= 3	
★ Status				0
Status in	dicator:	G Road signs	•	
Status e:	xpression:			
CASE WHEN ( WHEN ( A WHEN ( END	[Measures]. [Measures]. ND [Measures]. [Measures].	[Kusy]/[Measures].[Prod [Kusy]/[Measures].[Prod s].[Kusy]/[Measures].[F [Kusy]/[Measures].[Prod	eje Count] >= 3) THEN 1 eje Count] >= 2 rodeje Count] < 3) THEN 0 eje Count] < 2) THEN -1	

Obrázek č. 28: Definice KPI

Vyhodnocení námi definovaného cíle lze zobrazit tlačítkem *Browser View*. Tento výsledek vypadá v případě cvičných dat takto:

Kostka Prodeje.cube [[	Design] Start Page			
🔍 Cube Structure 🛛 💇	Dimension Usage 🛛 🍓 Calculations 🗧	F KPIs 🔀 Actions 阁 Partitions	]Σ Aggregations	🗊 Perspectives 🚺
😼 🖧   🛃 🔤   📰	× + + 🗉 🛃 🖤 👘			
Dimension	Hierarchy	Operator	Filter Exp	ession
<select dimension=""></select>				
Display Structure		Value	Goal	Status
Focet prodanych kusu	na jeden prodej	3.53123521060104	ſrue	6

### Obrázek č. 29: Výsledný ukazatel v záložce KPI

Jak je vidět, náš cíl byl splněn. Průměrný nákup obsahuje přibližně 3.5 kusů výrobků, což vyhovuje stanoveným podmínkám. V případě, že by cíl splněn nebyl (výsledná hodnota by byla menší než 3.0), pak by byl status tohoto výsledku indikován červenou šipkou otočenou směrem dolů.

### Validace a vytvoření datové kostky

Nyní můžeme konečně přejít k finálnímu vytvoření datové kostky. Nejprve bude třeba zkontrolovat, zda je kostka navržena správně. To si zjistíme pomocí tlačítka *Process*, které nám pomůže s validací návrhu kostky. Jestliže se v průběhu validace vyskytne vážnější problém, pak se nám zobrazí chybová hláška, na jejímž základě je třeba chybu opravit.

Command			
	ng Cube 'Kostka Prodeje' completed		
	time: 6.5.2012 19:51:48: End time: 6.5	5,2012,19:51:48: Duration: 0:00:00	
	essing Measure Group 'Prodeie' com	npleted.	
Gr (and	tart time: 6.5.2012 19:51:48: End time	e: 6.5.2012 19:51:48: Duration: 0:00:00	
F 🤗 I	Processing Partition 'Prodeie' comple	eted. 4226 rows have been read.	
	Start time: 6.5.2012 19:51:48: End t	time: 6.5.2012 19:51:48: Duration: 0:00:00	
<b>A</b>	SOL queries 1		

Obrázek č. 30: Validace návrhu datové kostky

Pokud validace proběhla úspěšně, můžeme kostku vytvořit a odeslat pod správu analytického serveru. Toho docílíme pomocí tlačítka *Deploy* (záložka *Build* v hlavním menu):



Obrázek č. 31: Vytvoření datové kostky a odeslání pod správu serveru

#### Prohlížení datové kostky

V této fázi se konečně můžeme podívat na výsledek naší práce. Prostřednictvím MS SQL Management Studio se pomocí služby Analysis Services připojíme k serveru, pod jehož správu jsme datovou kostku odeslali. Zde si můžeme dle aktuální potřeby sestavit požadovanou kontingenční tabulku, která je jakýmsi dvojrozměrným pohledem na data v této kostce obsažená.

Measure Group:	Dimensio	n		Hierarch	ıy	Op	erator	Filter Express	on	
<all></all>	Drop Filte	r Fields He	ere							
🮯 Kostka Prodeje 🛛 🔺					kategorie 🔻					
🖹 📷 Measures	1				IT HW	I Média	(∓) Mobil	(∓) Pivo	I Tvrdý	Grand Total
🖃 🍋 Prodeje	Rok	<ul> <li>Kvart</li> </ul>	al Mesic	Den	Trzba	Trzba	Trzba	Trzba	Trzba	Trzba
Kusv	E 2009	<b>H</b> 1			63273.2	33896	7938.5	1146	17398	123651.7
Prodeie Count		<b>±</b> 2			65092.8	33109.6	5977.1	2116.7	27683.9	133980.1
I Traba		⊟ 3	August		72631	32197.9	7326.4	1480.5	18001.5	131637.3
11204			🕀 July		17140.6	14463.4	2282.7	386.8	10179.4	44452.9
DPH			September	Monday	2997	2348.7	299.4		43.8	5688.9
prumerna_trzba			2227	Tuesday	902	10238.2	728.1	164.2	8005.9	20038.4
🖃 🙋 Cas 👘				Wednesday	2698.2	4905.3	1322.5	165.9		9091.9
🗄 Datum				Thursday	21826.8	10017.4	1742	748.4	2566.5	36901.1
🕀 📕 Den				Friday	6306.2	12596.6	1074.4	394	2598	22969.2
E Kvartal				Saturday		1998	179.4	13.8		2191.2
Mesic				Sunday	9893	5004.3	251.1	208	999	16355.4
H Mesic				Total	44623.2	47108.5	5596.9	1694.3	14213.2	113236.1
н н кок			Total		134394.8	93769.8	15206	3561.6	42394.1	289326.3
Hierarchy		<b>H</b> 4			57724.6	24956.2	5811.3	791.6	9102.6	98386.3
🕀 🌄 Members 💷		Total			320485.4	185731.6	34932.9	7615.9	96578.6	645344.40000000
🕀 🍷 Rok					450368.2	239837.5	36170.9	10025.8	132611.3	869013.70000002
🕀 🥶 Kvartal					1019367.4	434678	83994.4	22520.8	327443.5	1888004.09999999
🕀 🙏 Mesir	<b>± 2012</b>				84880.2	30974.7	6464.7	1599	15985.2	139903.8
	Grand To	tal			1875101.2	891221.8	161562.9	41761.5	572618.6	3542265.99999997

Obrázek č. 32: Prohlížení datové kostky v MS SQL Analysis Services

Obrázek č. 32 představuje analýzu firemních tržeb, na které je nahlíženo ze dvou úhlů pohledu. Prvním pohled představuje uplynulý čas, druhý pak jednotlivé produktové kategorie a produkty v nich obsažené. Právě v tomto příkladu jsou vidět výhody časové hierarchie *rok – kvartál – měsíc – den*, kterou jsem zmiňoval již na straně č. 34. Díky ní můžeme jednoduše sestoupit na nižší úroveň agregace a dostat se k detailnějším datům (*drill-down*) nebo naopak přejít na vyšší agregační úroveň a dívat se na data z širšího pohledu (*roll-up*). Samozřejmě můžeme libovolně měnit úhly pohledu (*pivoting*), protože máme ještě k dispozici dimenze *Zakaznik* a *Zamestnanci*, jejichž pohled lze také využít.

### 7.2.3 Shrnutí

V této úloze jsme si názorně předvedli kompletní proces návrhu datové OLAP kostky, který zahrnoval operace od definování datového zdroje až po vytvoření kostky samotné a její následné odeslání pod správu analytického serveru. Jako základní stavební kámen návrhu bylo zvoleno snowflake schéma o čtyřech dimenzích a jedné faktové tabulce. V závěru jsme se přesvědčili o funkčnosti tohoto návrhu, jehož výsledkem může být přehled o tržbách, počtech prodejů nebo počtech prodaných kusů výrobků v rámci libovolné kombinace pohledů z existujících dimenzí.

# 7.3 Úloha č. 3: Návrh cvičného reportu

Ve třetí úloze si ukážeme, jak lze jednoduše vytvořit report, pro jehož základ budou opět použita data ze školní cvičné databáze. Na tomto místě bych chtěl uvést, že existuje nepřeberný počet možností, jakými lze docílit jeho výsledné podoby. Tyto možnosti se mohou odvíjet jednak od požadavků zadavatele, ale samozřejmě také na individuální tvořivosti jedince, který dostal návrh reportu na starost. Cílem tohoto úkolu tedy nebude podrobné seznámení se všemi funkcemi a nástroji, které se dají pro jeho tvorbu využít. V takovém případě by byl výsledný report zcela jistě "přeplácaný" a tím i nepřehledný, což není naším záměrem. Studenti by se proto měli spíše snažit prozkoumat vizuální možnosti návrhu vlastními silami a následně vytvořit své vlastní originální řešení.

### 7.3.1 Formulace problému

Řekněme, že jsme obdrželi požadavek z vyšších míst na zpracování sestavy, která má obsahovat přehled výkonosti jednotlivých zaměstnanců za rok 2011 se zaměřením na produkty z kategorií *Mobil*, *HW* a *Média*. Jako kritérium k posouzení výkonnosti je požadován výpis počtu uzavřených objednávek, počet kusů prodaného zboží a celkový objem tržeb pro tyto kategorie. Na základě tohoto požadavku se pokusíme navrhnout report, který tyto informace našemu zadavateli zprostředkuje.

### 7.3.2 Postup řešení

Podobně jako u minulé úlohy začneme s vytvořením nového projektu v návrhovém prostředí BIDS. Tentokrát však zvolíme možnost s názvem *Report Server Project* (viz. obrázek č. 14 na straně 36). Poté si projekt pojmenujeme a zvolíme umístění, kde bude uložen.

### Volba datového zdroje

I v této úloze bude nutné zvolit zdroj, který obsahuje potřebná data pro její realizaci. V okně *Solution Explorer* klikneme pravým tlačítkem myši na záložku *Shared Data Sources* a z nabídky vybereme možnost přidání nového datového zdroje (*Add New Data Source*). Dále je postup pro jeho specifikaci podobný jako u předchozí úlohy, není tedy potřeba ho opět rozepisovat. Hlavní je především to, aby byla zvolena cvičná databáze *Cviceni\_create*, jejíž tabulky nám poslouží pro tvorbu reportu.

### Návrh SQL dotazu

Nyní přejdeme k záložce *Reports* a v jejím menu zvolíme možnost přidání nového reportu (*Add -> New Item -> Report*) a dle vlastního uvážení si ho pojmenujeme.

ategories:	Templates:
Report Project	Visual Studio installed templates Report Wizard Data Source My Templates Search Online Templates

Obrázek č. 33: Přidání nového reportu

Poté, co jsme si nový report přidali, se nám zobrazí (prozatím prázdné) návrhové okno. Pro vizuální a logickou strukturu reportu si vystačíme s nástrojem *Toolbox* (Ctrl+Alt+X), který je pro tyto účely navržen. Pomocí tohoto nástroje lze navrhovat jak tabulky pro výstup daného reportu, tak nejrůznější grafy, obrázky a další objekty. Základ našeho reportu bude tvořit jednoduchá tabulka, proto si ji přetáhneme z nabídky nástroje *Toolbox* do návrhového okna. Tím se nám automaticky otevře nové okno, v němž potvrdíme, že chceme využít databázi *Cviceni\_create* jako datový zdroj pro vytvoření SQL dotazu.

DataSet1		
Use a shared dataset.		
Use a dataset embedded in my report.		
Data source:		
Cviceni create		
Query type:		
Query type: <u>Text</u> Table O <u>S</u> tored Procedure		
Query type: Iext Table Stored Procedure Query:		
Query type: <u>Iext</u> Table O Stored Procedure Query:	*	f_x
Query type: <u>Text</u> Table © <u>S</u> tored Procedure <u>Q</u> uery:	*	fs
Query type: <u>Text</u> Table O Stored Procedure Query:	*	f <sub>x</sub>
Query type: <u>Text</u> Table O Stored Procedure Query:	*	fx
Query type: <u>Text</u> Table O <u>S</u> tored Procedure Query:	*	

Obrázek č. 34: Okno pro specifikaci datového zdroje a návrh SQL dotazu

V tomto okně se také nachází kolonka *Query*, která slouží pro samotný návrh dotazu, bylo by však nepraktické dotaz psát přímo zde. Jednodušší bude, pokud si dotaz předem připravíme v MS SQL Management Studiu (obdobně jako skript v úloze č. 1) a následně si jej zkopírujeme sem. Dotaz, který bude splňovat požadavky ze zadání, může vypadat například takto:

```
SELECT zam.Jmeno + ' ' + zam.Prijmeni AS Zamestnanec,
      zam.Mesto
                         AS Mesto,
                                   AS PSC,
      zam.PSC
      Cast (Datum narozeni AS DATE) AS 'Datum narozeni',
                                    AS oddeleni,
      odd.Nazev
                                   AS kategorie,
      k.nazev
      Sum(od.mnozstvi * p.Cena)
                                   AS trzba,
      Sum(od.mnozstvi)
                                   AS kusy,
      Count (o.id objednavky)
                                   AS pocet objednavek
FROM Objednavky o
      INNER JOIN Objednavka detail od
        ON o.ID objednavky = od.ID objednavky
           AND Year(o.datum odeslani) = 2011
      INNER JOIN Produkty p
        ON od.id produkt = p.Id produkt
      INNER JOIN kategorie k
        ON p.ID kategorie = k.ID_kategorie
           AND k.nazev IN ( 'HW', 'Mobil', 'Média' )
      INNER JOIN Zamestnanci zam
        ON o.ID zamestnance = zam.ID Zamestnance
      INNER JOIN Oddeleni odd
        ON zam.ID Oddeleni = odd.ID Oddeleni
GROUP BY zam.Jmeno + ' ' + zam.Prijmeni,
         zam.Mesto,
         zam.PSC,
         Cast (Datum narozeni AS DATE),
         odd.Nazev,
         k.nazev
```

#### Obrázek č. 35: SQL dotaz pro vytvoření reportu

#### Návrh vizuální stránky reportu

Na základě získaných dat, která nám poskytne předpřipravený SQL dotaz, můžeme nyní přikročit k dalším úpravám, abychom dodali našemu reportu odpovídající vzhled. Zde se již může prezentace těchto dat ubírat odlišnými směry a já se pokusím nastínit, jak může jedno z možných řešení vypadat.

V tomto příkladu budou jednotlivé sloupce tabulky tvořeny hodnotami atributů *zamestnanec, kategorie, pocet\_objednavek, kusy* a *trzby*, které jsou výsledkem našeho dotazu.

oolbox	<b>→</b> ‡ X	Vyk	on zamestnancu	2011.rdl [Desig	n]* Start Page			
Report Items				1154) - 55 1111	- Linearconing and			
Pointer			esign Sesign	N				
b Textbox			ĺ		1	<u> </u>		
Table		8	Zamestnanec	kategorie	pocet objednavek	kusy		
Matrix		-	[Zamastaanaa]	[ketegorie]	[noost_objednavaki			
Rectangle			Zamesthanecj	[kategone]	[pocet_objednavek]	[kusy]		Zamestnanec
🛛 List							-	Mesto
Image								DEC
Chart	E							PSC
Subreport							EE.	Datum_narozeni
Line							22°	oddeleni
Gauge								kategorie
🕽 Мар								trzba
Data Bar							<b>EEE</b>	kusy
Z Sparkline							-	pocet_objednavek
Indicator								

Obrázek č. 36: Návrh reportu, 1. část

Pro větší přehlednost budeme chtít zobrazit jméno zaměstnance vždy pouze jednou, zbylé atributy tedy budeme chtít seskupit právě podle něj. Toho docílíme v okně s názvem *Row Groups*, které se nachází hned pod samotným oknem návrhového prostředí. Zde prostřednictvím volby *Add Group -> Parent Group* vybereme atribut *zamestnanec*.

Dále budeme chtít, aby se nám u každého zaměstnance vypisovaly některé základní informace, jako např. jeho adresa nebo datum narození. Dle těchto požadavků si upravíme výšku příslušných řádků a podle potřeby sloučíme na potřebných místech více buněk (merge cells) do buňky jediné. Pro lepší pochopitelnost si ještě následně přejmenujeme nadpisy jednotlivých atributů. Po těchto krocích by mohl návrh tabulky vypadat přibližně takto:

[Zamestnanec]				
	Kategorie	Zpracované objednávky	Prodané kusy	Tržba celkem
	[kategorie]	[pocet objednavek]	[kusy]	[trzba]

Obrázek č. 37: Návrh reportu, 2. část

Kdykoli během návrhu reportu se můžeme přepnout do náhledového zobrazení, kde si můžeme zkontrolovat, jak bude vypadat jeho prezentace v reálu. V této fázi je znázorněn náhled našeho reportu na výkon prvních dvou zaměstnanců na obrázku č. 38 na další straně.

◀ 1 of 2?	▶ N   + ®	😳   🖨 🔲 💷 🔍 -   🛛 10	0%	Fi
Adéla Krásná				
	Kategorie	Zpracované objednávky	Prodané <mark>ku</mark> sy	Tržba celkem
	HW	54	186	164946,4
	Média	112	371	64176,4
	Mobil	65	215	11871,2
Anna Novotná				
	Kategorie	Zpracované objednávky	Prodané <mark>ku</mark> sy	Tržba celkem
	HW	22	97	86319,2
	Média	40	121	19289,8
	Mahil	30	07	4970 0

#### Obrázek č. 38: Náhled rozpracovaného reportu

Nyní přejdeme k již zmíněnému výpisu základních informací o našich zaměstnancích. Pokud chceme zobrazit více hodnot v jedné buňce, využijeme k tomu funkci *Create Placeholder*, která se nám zobrazí v nabídce po nakliknutí buňky pravým tlačítkem myši v místě, kde chceme hodnotu zobrazit. Tímto způsobem si do jedné z větších buněk uložíme atributy *oddeleni*, *datum\_narozeni*, *mesto* a *PSC*.

Zbývající prázdnou buňku využijeme pro grafický prvek, pomocí kterého budeme schopni na první pohled porovnat, jak jsou na tom s výkonností jednotliví zaměstnanci. Pro tento účel zvolíme v nástroji *Toolbox* prvek *Gauge* a poté si vybereme takový typ vizualizace, který nám bude nejvíce vyhovovat.

elect Gauge Type	2	23
	Radial	
Linear		

Obrázek č. 39: Výběr doplňujícího grafického prvku

Dále bude třeba tento prvek provázat s výstupem našeho SQL dotazu tak, aby nás mohl o výkonu zaměstnance skutečně informovat. K tomu bude třeba určit, na základě kterého atributu bude výkon mezi zaměstnanci porovnáván a zobrazen. V našem případě zvolíme hodnotu průměrné tržby (Avg(trzba)) následujícím způsobem:



Obrázek č. 40: Grafický prvek Gauge – nastavení

Pro lepší přehlednost můžeme přidat ke všem vypočítaným hodnotám sumarizované údaje za všechny tři produktové kategorie. To provedeme jednoduše tak, že si označíme příslušné buňky a pomocí pravého tlačítka myší vybereme možnost s názvem *Add Total*, čímž se nám vygeneruje nový řádek s těmito sumarizovanými hodnotami.

Protože máme již všechny potřebné buňky vyplněné, zbývá nám pouze celý report vizuálně doladit. Jedná se především o úpravy stylu písma, ohraničení a barevnou výplň jednotlivých buněk apod. Tyto úpravy jsou poměrně triviální a nebude třeba ukazovat jejich postup, protože si s nimi poradí každý sám dle vlastního uvážení.

### Sestavení reportu

Pokud máme všechny předchozí kroky dokončeny, můžeme přejít k finální kontrole reportu. Kontrolu provedeme pomocí záložky *Debug* v hlavním menu, kde vybereme možnost *Start Debugging*. Jestliže tato kontrola proběhne bez problému, zobrazí se nám výsledná podoba reportu, kterou můžeme rovnou uložit v několika různých formátech (XML, CSV, TIFF, PDF, MHTML, XLS, DOC) a poskytnout zadavateli nebo skupině koncových uživatelů. Ideálnější však bude, pokud report umístíme na server, kde si ho budou moci tito uživatelé zobrazit a dle vlastního uvážení uložit v libovolném ze zmíněných formátů.

Sestavení reportu provedeme pomocí záložky *Build* v hlavním menu a následně se můžeme pokusit jej odeslat pod správu reportovacího serveru (*Deploy*). Nové operační systémy mají z důvodu bezpečnosti omezená některá privilegia při spouštění aplikací ve standardním uživatelském módu. Pro úspěšné nasazení (deployment) je tedy nutné mít spuštěno vývojové prostředí BIDS v administrátorském režimu. Dále je třeba mít nastavenou URL adresu serveru, kam hodláme náš report umístit. Tu zadáme v okně *Solution Explorer* pomocí položky *Properties*, která je ukryta pod názvem našeho projektu. Adresu vyplníme do kolonky s názvem *TargetServerURL*. Následně by nám nemělo nic bránit v úspěšném nasazení reportu.

Output	
Show output from: Build	-   Q   4 🔍   🛼   🖬
Build started: Project: Skipping 'Cviceni_create2.rds' Skipping 'Vykon zamestnancu 20 Build complete 0 errors, 0 Deploy started: Project Deploying to <u>http://mosby-pc/P</u> Deploying data source '/Data S Deploying report '/Vykon zames Deploy complete 0 errors, 0 ====== Build: 1 succeeded ===== Deploy: 1 succeeded	Vykon zamestnancu 2011, Configuration: Release . Item is up to date. 11.rdl'. Item is up to date. warnings : Vykon zamestnancu 2011, Configuration: Release <u>eportServer</u> ources/Cviceni_create2'. tnancu 2011/Vykon zamestnancu 2011'. warnings or up-to-date, 0 failed, 0 skipped ===================================
<	
👌 Error List 🔳 Output	
Deploy succeeded	

#### Obrázek č. 41: Průběh sestavení a nasazení reportu

Nyní si můžeme zobrazit konečný výsledek naší práce v okně webového prohlížeče:



Obrázek č. 42: Hlavní menu pro správu reportingových projektů

🥖 Vykon zamestnancu 201	1 - Report Manager - W	/indows Internet Explorer			
😋 🔾 🗢 🙋 http://m	osby-pc/ReportServer/I	Pages/Report.aspx?ItemPath=%2fVyI	kon+zamestnancu+201	1%2fVykon+zamestnar	
🚖 Oblíbené položky 🛛 👔	Vykon zamestnancu	2011 - Report Manager			
Home > Vykon zamestna	incu 2011 > Vykon za	amestnancu 2011	^ 		
II of 2 ?	▶ ▶ 100%	← Find   Ne	oxt 🖳 🛛 🚱 😭	b 🔲	
Výkon	i jednotlivý	ich zaměstnanci	ů v roce 2	011	
Auela Krasna	Datum narození: Město: Praha PSČ: 170 00	14.10.1966	in the		
	Kategorie	Zpracované objednávky	Prodané kusy	Tržba celkem	
	HW	54	186	164946,4	
	Média	112	371	64176,4	
	Mobil	65	215	11871,2	
	SUM	231	772	240994	
Anna Novotná	Oddělení: Ekonom Datum narození: Město: Brno PSČ: 160 00	nické 14.10.1976	the second secon		
	Kategorie	Zpracované objednávky	Prodané kusy	Tržba celkem	
	HW	22	97	86319,2	
	Média	40	121	19289,8	
	Mobil	30	97	4870,9	
	SUM	92	315	110479,9	

Obrázek č. 43:	Výsledná podo	oba nasazeného	reportu
----------------	---------------	----------------	---------

# 7.3.3 Shrnutí

V této úloze jsme si názorně předvedli, jak si lze vlastními silami navrhnout poměrně jednoduchý report. Úloha sama o sobě však obsahuje využití pouhého zlomku funkcí, které nám společnost Microsoft ve své aplikaci nabízí. To ale nikterak nebrání studentům v tom, aby pro vypracování svých úloh použili zcela jiný postup a vytvořili si tak report dle vlastních představ.

# 7.4 Úloha č. 4: Analýza dat pomocí MS Excel 2010

Cílem této úlohy bude předvedení názorné ukázky, jaké možnosti nabízí MS Excel 2010 jako klient pro přístup k databázím a podrobný popis, jakým způsobem je s nimi možné pracovat. Úloha bude obsahovat popis funkcí, které Excel nabízí pro analýzu dat a demonstrovat jejich použití na praktických ukázkách.

# 7.4.1 Formulace problému

Ne každý uživatel má přístup k tak komplexním a licenčně drahým nástrojům, jaké nabízí platforma MS SQL Server, která byla využita pro návrh předchozích úloh. Uvažujme proto situaci, kdy se chce představitel této skupiny uživatelů připojit k požadované databázi a data v ní uložená analyzovat i bez nutnosti použití těchto nástrojů. Právě v takovém případě lze využít analytické funkce, které nabízí běžně rozšířený tabulkový procesor MS Excel. Proto se nyní budeme snažit "vžít" do této situace a pokusíme se tímto způsobem získat z naší cvičné databáze co nejširší spektrum informací.

# 7.4.2 Postup řešení

Nejprve si v krátkosti představíme některé možnosti, které nám MS Excel nabízí v souvislosti s různými typy připojení k databázi.



Obrázek č. 44: Možnosti připojení ke zdroji dat v MS Excel

Z popisků jednotlivých připojení na předchozím obrázku se dá vyčíst, že se můžeme připojit k samotné tabulce (možnost *Ze serveru SQL Server*), ale i k existující

datové kostce (*Ze služby pro analýzu*). Ani jednu z těchto možností však nevyužijeme, protože první varianta v podobě jedné tabulky je pro náš účel nedostačující. Druhá varianta je sice sama o sobě velice užitečná, ale jelikož jsme se problematice datové kostky věnovali v druhé úloze, nebylo by zde mnoho nového k představení. Místo toho zvolíme až úplně poslední variantu – *Z aplikace Microsoft Query*. Právě zde si budeme moci ukázat, jak se dostat k datům z více tabulek a jak z těchto dat získat potřebné informace.

#### Volba datového zdroje

Ani v této úloze se neobejdeme bez specifikace datového zdroje. Aplikace nás automaticky vyzve k výběru zdroje hned poté, co jsme zvolili typ připojení. Z nabídky v záložce *Databáze* vybereme možnost *<Nový zdroj dat>* a vyplníme nově otevřené okno následujícím způsobem:

vytvi	
1.	Klimanek.sks.cz
	Vyberte ovladač pro typ databáze, ke které žádáte přístup:
2.	SQL Server Native Client 10.0
	Klikněte na tlačítko Připojit a zadejte informace požadované ovladačem:
3.	<u>Připojit</u> cviceni_create
	Vyberte výchozí tabulku pro zdroj dat (volitelné):
4.	•
111111	
2	) OK Storno



### Výběr databázových tabulek a jejich atributů

Jestliže máme zvolený zdroj dat, pak nás aplikace vyzve k bližšímu upřesnění, které tabulky budeme chtít pro naše potřeby využít. S tímto výběrem nemusíme být zatím příliš pečliví, můžeme vybrat všechny tabulky a jejich atributy, o kterých si myslíme, že by mohly mít pro naši analýzu určitý význam. Čím širší datovou základnu nyní specifikujeme, tím lépe pro nás, protože v konečném důsledku si budeme moci vyfiltrovat jen ta data, která nás budou aktuálně zajímat. V tomto názorném příkladu tedy vybereme z cvičné databáze (viz. strana 31, obrázek č. 9) všechny tabulky kromě

tabulek *Skoleni* a *ZamestnanciSkoleni*. Dále zvolíme libovolné množství neklíčových atributů (klíčové atributy nemá příliš smysl analyzovat), které budeme chtít použít. Jediný klíčový atribut, který můžeme do našeho výběru zařadit, je *ID\_objednavky*. Jednotlivé objednávky totiž nelze přímo identifikovat jinak, než tímto číslem, kdežto zbylé tabulky mají většinou neklíčový atribut, který představuje název nebo jméno dané entity. Reprezentace entity pomocí názvu je pro nás totiž mnohem lépe stravitelná než pomocí nic neříkajícího čísla.

ID_kategorie ID_nadrizene ⊡ Objednavka_detail ID_objednavky id_produkt ⊡ Ωhiednavky Náhled dat ve vybraném sloupci:	•	U_objednavky datum_odeslani Nazev Nazev Cena prijmeni mesto	
---	---	---	--

Obrázek č. 46: Výběr tabulek a jejich atributů pro sestavení dotazu

Po výběru atributů následuje okno, kde si můžeme omezit jejich rozsah. Tuto možnost však nevyužijeme, protože naším cílem je, abychom měli pro plánovanou analýzu všechna data dostupná. Případná omezení bude výhodnější provádět až při samotné analýze formou filtrace. Z obdobného důvodu přeskočíme i okno pro seřazení dat.

### Úprava dotazu pomocí Microsoft Query

Průvodce se nás dále ptá, zda chceme specifikovaná data načíst do MS Excel, čímž se automaticky vygeneruje SQL dotaz (bez možnosti jeho kontroly nebo úpravy), s jehož výsledkem budeme moci následně pracovat. My však zvolíme druhou variantu, která nám umožní vygenerovaný dotaz upravit. Díky této volbě se přepneme do návrhového prostředí *Microsoft Query*, které je součástí MS Excel. Zde si můžeme prohlédnout náhled, který vznikl na základě naší specifikace výběru dat.

🗳 Microsoft Query					
Soubor Úpravy Zobrazit For	mát Tabulka Kritéria	Záznam Okno N	ápověda		
<b>₽₽₽₽₽</b> ₩ ₩₩₩		• • • •			
Dotaz z local				[	
kategorie Objednavk	a_detail Objednavky	Oddeleni	Produkty	Zakaznik	Zamestnanci
ID_kategorie ID_nadrizene nazev ID_notrizene nazev	avky datum_odeslani t datum_prijeti ID_objednavky ID_zakaznika ID_zamestnance	Datum_zalozeni ID_Oddeleni Nazev Popis	Cena ID_kategorie Id_produkt Nazev	* <b>ID_zakaznika</b> jmeno mesto prijmeni psc	* ^ Datum_narozer datum_nastupu ID_nadrizeny ID_Oddeleni ID_ <b>Zamestnan</b> *
mnozstvi ID obiednav	vkv datum odeslani	Nazev	Nazev	Cena	priime: 🔺
	2010-01-19 00:00:00,00	Technické	Ostravar Premium 12*	6,9000000000000004	Bláha
3 1	2010-01-19 00:00:00,00	Technické	Litovel 12*	11,9	Bláha —
4 1	2010-01-19 00:00:00,00	Technické	Bakalář 12°	14,9	Bláha
1 1	2010-01-19 00:00:00,00	Technické	BlueRAY Memorex	19,89999999999999999	Bláha
3 1	2010-01-19 00:00:00,00	Technické	CELLY / ochranná fólie	29,8999999999999999	Bláha
5 1	2010-01-19 00:00:00,00	Technické	Matrox TripleHead2Go E	902,0	Bláha
IC 1 IIIZáznam:1 ► III 4	2010 01 19 00-00-00 00	Teobnická	ATLE adapts 0000000 1	1100 በ	Diába

Obrázek č. 47: Návrhové prostředí Microsoft Query

Úpravu SQL dotazu budeme moci provést po kliknutí na tlačítko s názvem *SQL* (obrázek č. 47), čímž se nám otevře okno s automaticky vygenerovaným kódem, který budeme moci pozměnit. Tuto změnu provedeme z toho důvodu, protože budeme chtít ve výsledku dotazu zobrazit sloupec s tržbami, který vznikne kombinací atributů *mnozstvi* a *cena*. Tato hodnota bude seskupována pomocí klauzule GROUP BY. Dále si přejmenujeme potřebné sloupce (pomocí přidělení aliasů) tak, aby se mezi nimi nevyskytovaly duplicitní názvy, což by pro nás mohlo být v budoucnu matoucí. Po těchto úpravách může vypadat náš SQL dotaz zhruba takto:



<pokračování skriptu na další straně>

WHERE	<pre>kategorie.ID_kategorie = Produkty.ID_kategorie AND Objednavka_detail.ID_objednavky = Objednavky.ID_objednavky AND Objednavka_detail.id_produkt = Produkty.Id_produkt AND Objednavky.ID_zakaznika = Zakaznik.ID_zakaznika AND Objednavky.ID_zamestnance = Zamestnanci.ID_Zamestnance</pre>						
GROUP	BY Objednavky.ID_objednavky, Objednavky.datum_odeslani, Oddeleni.Nazev, Produkty.Nazev, Zakaznik.prijmeni, Zakaznik.mesto, Zamestnanci.Prijmeni, Zamestnanci.Mesto, Zamestnanci.Pohlavi, kategorie.nazev						

#### Obrázek č. 48: Úprava automaticky vygenerovaného SQL dotazu

V případě, že jsme dotaz upravili a nevyskytla se v něm žádná chyba, můžeme jeho výsledek odeslat pomocí volby *Načíst data do aplikace Microsoft Excel*.

#### Návrh kontingenční tabulky a grafu

Ihned po dokončení práce v *Microsoft Query* se nám zobrazí okno pro import dat v podobě klasické tabulky, kontingenční tabulky nebo kombinace kontingenční tabulky a grafu. V tomto příkladu zvolíme třetí možnost a označíme buňku, ve které bude začínat levý horní roh kontingenční tabulky.

Importovat data	8 3
Vyberte způsob zobrazení dat v sešitu.	
<ul> <li> <u>gestave kontingerich i debaky</u> </li> <li> <u>Sestave kontingerich i debaky</u> </li> <li> <u>Sestave kontingerich i debaky</u> </li> <li> <u>Sestave kontingerich i debaky</u> </li> <li> <u>Pouze vytvořit připojení</u> </li> <li> <u>Umístění dat</u> </li> <li> <u>Existující list:</u> </li> </ul>	kontingenční tabulky
=\$A\$1	<b>S</b> )
🔘 <u>N</u> ový list	
Vlastnosti OK	Storno

Obrázek č. 49: Volba typu výstupu z SQL dotazu

Kontingenční tabulka představuje interaktivní způsob rychlého shrnutí velkých objemů dat. Díky ní lze do maximálních podrobností analyzovat numerická data pomocí filtrování, řazení, seskupování a podmíněného formátování. Tabulku sestavíme přerozdělením potřebných atributů mezi čtyři, pro tento účel navržené, oblasti. Dejme tomu, že budeme chtít zobrazit tržby pro jednotlivé produktové kategorie v uplynulém čase. To můžeme provést například tak, jak je to uvedeno na obrázku č. 50 na další straně.

Zvolte pole, které chcete	přidat do 🛛 📑	-
sestavy:		_
kategorie		*
množství		
oddělení		
pohlaví		
produkt		
<b>V</b> tržba		1
🗾 zákazník		
🗾 zákazník_město		
zaměstnanec 📃		
		-
Přetáhněte pole do jedné	z následujících oblastí:	
Y Filtr sestavy	Pole legendy (rad	y)
	The factor of the second se	
	Kategorie	
Pole osy (kategorie)	Σ Hodnoty	

Obrázek č. 50: Výběr atributů pro kontingenční tabulku a graf

V tomto případě však zobrazení tabulky není zcela ideální, jelikož čas je zobrazen na nejnižším stupni agregace pomocí atributu datum\_odeslani, čímž dostaneme velice detailní (a rozsáhlý) přehled o tržbách za jednotlivé dny. Naštěstí v sobě kontingenční tabulka v aplikaci MS Excel ukrývá chytrou funkci pro seskupení. Stačí kliknout pravým tlačítkem myši kdekoliv do sloupce s časovými hodnotami a z nabídky vybrat možnost *Seskupit*. Pak již pouze stačí vybrat, dle jakých stupňů agregace budou tržby sumarizovány. Každá změna v tabulce se zároveň projeví i ve vykreslení grafu, protože jsou tyto dva prvky spolu vzájemně propojeny. V našem případě seskupíme tabulku následovně:

Součet z tržba	Popisky sloupců 💌							
Popisky řádků 🔻	нw	Média	Mobil	Pivo	Tvrdý	Celkový součet	Seskupit	? <b>X</b>
8.1.2009	1099	4030,5	1127,8	1		6257,3	Automaticky	
11.1.2009	9999	975	749,3	274,6	3996	15993,9	Začátek:	8.1.2009
15.1.2009	6847	510,3	144,5	52,7	6298,4	13852,9	Konec:	12.2.2012
16.1.2009	2378	326,8	456,9	194,5	900,8	4257	Dedle	
21.1.2009	2198	1502,9	146,4	41,4		3888,7	Podle	
24.1.2009			243,6	5		243,6	Sekundy Minuty	*
8.2.2009	1189	3399,5	601,7	27,6		5217,8	Hodiny	
9.2.2009	8700	368,7	75,6	6,9		9151,2	Měsíce	
11.2.2009			720,5	13,8		734,3	Čtvrtletí	
12.2.2009		3996				3996	RORY	
13.2.2009	3297	2865,5	581,1	. 101	1998	8842,6		Počet dnů: 1
14.2.2009	5495	4009,8	1121,7	,		10626,5		
8.3.2009	3001,2	247,4		46,4		3295		OK Storno
9.3.2009		5994	89,7	130,8		6214,5		

Obrázek č. 51: Seskupení hodnot v kontingenční tabulce

Popisky řádků 💌	HW	Média	Mobil	Pivo	Tvrdý	Celkový součet	
<b>=</b> 2009							
⊟Čtv1							
1⊞1	22521	7345,5	2868,5	563,2	11195,2	44493,4	
<b>⊞</b> 2	18681	14639,5	3100,6	149,3	1998	38568,4	
<b>⊞</b> 3	22071,2	11911	1969,4	433,5	4204,8	40589,9	
⊞Čtv2	65092,8	33109,6	5977,1	2116,7	27683,9	133980,1	
⊞Čtv3	134394,8	93769,8	15206	3561,6	42394,1	289326,3	
⊞Čtv4	57724,6	24956,2	5811,3	791,6	9102,6	98386,3	
± 2010	450368,2	239837,5	36170,9	10025,8	132611,3	869013,7	
± 2011	1019367,4	434678	83994,4	22520,8	327443,5	1888004,1	
± 2012	84880,2	30974,7	6464,7	1599	15985,2	139903,8	
Celkový součet	1875101,2	891221,8	161562,9	41761,5	572618,6	3542266	

Po seskupení a naformátování vzhledu vypadá tabulka s rozpadem na roky, čtvrtletí a měsíce takto:

#### Obrázek č. 52: Seskupená kontingenční tabulka

Výsledná tabulka nemusí obsahovat pouze přehled tržeb pro jednotlivé kategorie v čase, proto jsme si také na začátku této úlohy zařadili do SQL dotazu co nejširší okruh atributů. Kdykoliv tak můžeme tabulku změnit tím, že si přetáhneme jiné atributy do oblastí pro vytvoření tabulky, jak již bylo znázorněno na obrázku č. 50 na straně 61. Díky tomu se můžeme dívat na tuto problematiku z různých úhlů pohledu.

Samozřejmě lze vzhled tabulky pomocí dalších nástrojů, které poskytuje aplikace MS Excel, dále upravovat. Za zmínku určitě stojí podmíněné formátování, pomocí něhož můžeme snadno jediným pohledem identifikovat odchylky v určitém rozsahu hodnot. S těmito nástroji jsou však studenti obeznámeni již z ostatních vyučovaných předmětů a proto nebude naším cílem je dále podrobněji rozebírat, jelikož si s nimi studenti jistě poradí sami.

#### Průřezy datovou základnou

Nyní si na vytvořené kontingenční tabulce z obrázku č. 52 ukážeme, jak se dají data přehledně a pohodlně filtrovat za pomoci ovládacího prvku s názvem *Průřez*. Nalezneme jej v záložce *Vložení* a pro jeho aktivaci je nutné mít označenou alespoň jednu buňku v kontingenční tabulce. Abychom mohli docílit co nejdetailnější filtrace, vložíme si do sešitu MS Excel průřezů hned několik. Výběr průřezů pro náš názorný příklad je zobrazen na obrázku č. 53 na následující straně.



Obrázek č. 53: Volba průřezů dat v kontingenční tabulce

Jednotlivé průřezy jsou reprezentovány jako samostatná okna, se kterými můžeme libovolně pohybovat, intuitivně měnit jejich velikost a barevné zobrazení. Pro větší přehlednost si tedy průřezy, které spolu souvisí (např. časové průřezy *Roky* a *Čtvrtletí*), umístíme tak, aby byly vedle sebe a měly stejnou barvu. Po těchto úpravách může zobrazení naší tabulky, grafu a průřezů vypadat přibližně jako na obrázcích č. 54 a 55.



Obrázek č. 54: Kontingenční tabulka s grafem

oddělení	K	zaměstnanec	zaměstnan 🜾				
Ekonomické		Bedrnová	Carvová	Černý	Brno Ostrava		
Ředitelství a	a managem	Koláčková	Krásná	Kraus			
Technické		Malá Nová		Novotná	Praha		
		Novotný	Nový	Peterková	Znojmo		
Roky 📡	Čtvrtletí 🕅	zákazník	zákazník 🕅 🕅				
2009 Čtv1 2010 E Čtv2 E 2011 Čtv3 2012 ↓ Čtv4 ↓		Adámek	Barnabáš	Bartoš	Frýdek Místek		
		Bláha	Bouzek	Burešová	Jihlava Kroměříž		
		Cikryt	Černá	Čiháková			
		Daněk	Diviš	Erbenová	Olomouc		
kategorie	K	Flóra	Galíková	Homola	Ostravice		
HW	Média	Chalupa	Chovanc	Jahoda	Znojmo		
Mobil	Pivo	Košařová	Linda	Morávek			
Tvrdý		Nesrsta	Novotná	Ospálek			
		Pultera	Říha				

Obrázek č. 55: Přidané a naformátované průřezy

Pro porovnání s předchozím obrázkem se nyní pokusíme pomocí průřezů vyfiltrovat data v kontingenční tabulce a grafu tak, aby byly zobrazeny tržby pouze pro určitou podmnožinu polí v těchto průřezech. Uvažujme případ, kdy budeme chtít vědět, jak si vedla zaměstnankyně Koláčková v druhém čtvrtletí roku 2011 s prodejem výrobků z kategorie *Média*. Tyto čtyři filtry si jednoduše aplikujeme (naklikáme) pomocí příslušných průřezů takto:

Součet z tržba	Popisky sloupců ∓		odděle	ní	K	zaměs	stnanec		X	zaměstnan 🜾
Popisky řádků 🚽	Média 7706 1	Celkový součet	Techn	ické		Bedr	Bedrnová	Carvová	Černý	Praha
Celkový součet	7706,1	7706,1	Ekono	Ekonomické		Kolá	čková	Krásná	Kraus	Brno
Součet z tržba			Ředite	Ředitelství a managem			i	Nová	Novotná	Ostrava
					Novo	otný	ý Nový	Peterková	Znojmo	
10000		Roky	¥	Čtvrtletí 🤻	zákazi	ník			zákazník m 🕅	
5000		kategorie 🖓	2009	Â	Čtv1 ^	Bou	zek	Erbenová	Galíková	Frýdek Místek
0		Média	2010	Е	Čtv2 ≣	Chov	vanc	Jahoda	Košařová	Kroměříž
2011			2011		Čtv3	Nov	otná	Adámek	Barnabáš	Olomouc
Roky 🖓 Čtvrtletí	→✔ Měsíce ▼ datum_o	deslani 💌	<8	-	Čtv4 👻	Bart	oš	Bláha	Burešová	Ostravice
2.			katego	rie	<i>K</i>	Cikry	yt	Černá	Čiháková	Jihlava
			HW		Média	Dani	ěk	Diviš	Flóra	Znojmo
			Mobil		Pivo	Hom	nola	Chalupa	Linda	
			Tvrdý			Mor	ávek	Nesrsta	Ospálek	
						Pult	era	Říha		

Obrázek č. 56: Filtrace pomocí průřezů

Jak je vidět z obrázku č. 56, získali nejen požadovaný výsledek ve formě přizpůsobení kontingenční tabulky a grafu těmto filtrům, ale také jsme zároveň "vydolovali" dodatečné informace na základě zvýrazněných buňek ve zbývajících průřezech. Zjistili jsme tak, že paní Koláčková je z Prahy a pracuje v technickém oddělení. Dále jsme se také dozvěděli, díky kterým zákazníkům (a ze kterých měst) se jí podařilo výslednou tržbu pro firmu uskutečnit. Pokud se nyní zamyslíme nad množstvím kombinací, které lze díky filtrování v průřezech provádět, tak zjistíme, že máme k dispozici skutečně šikovný nástroj pro detailní analýzu dat.

### 7.4.3 Shrnutí

V průběhu poslední úlohy jsme si představili, jak lze využít aplikaci Microsoft Excel pro analýzu dat, aniž bychom museli pro přístup k cvičné databázi a sestavení potřebného SQL dotazu použít jakýkoliv jiný nástroj. Je tedy vidět, že základní potřeby "běžného" uživatele by měla tato aplikace dokázat uspokojit bez větších problémů. Jestliže byli studenti úspěšní při vytvoření datové kostky v úloze č. 2, pak bych jim také doporučil, aby se k ní pokusili připojit prostřednictvím MS Excel a vyzkoušeli si, jaké jsou zde rozdíly oproti MS SQL Analysis Services.

# Závěr

V souvislosti s neustálými změnami v globální ekonomice a nástupem informační doby se až nečekaným tempem zvětšuje objem dat, která je třeba uchovávat pro zajištění firemního rozvoje. Stejně tak vzniká potřeba z těchto dat extrahovat potencionálně využitelné informace pro podporu firemního rozhodování. Za tímto účelem dochází k rozvoji v oblasti Business Intelligence, jejíž hlavní jádro tvoří datové sklady, které jsou hlavním prostředkem pro uchování dat a jejich následnou analýzu prostřednictvím nástrojů BI.

Cílem této bakalářské práce byl návrh několika scénářů a jejich řešení, které by zachytily problematiku analýzy a reportování dat v podmínkách školní výuky.

V teoretické části jsem za pomoci odborné literatury a internetových zdrojů nastínil význam termínu Business Intelligence jakožto procesu analýzy dat za účelem zisku strategických informací. Dále jsem rozebral význam a dostupné typy architektur datových skladů, které s problematikou BI úzce souvisí. V závěrečné kapitole teoretické části jsem se následně pokusil shrnout přínos BI pro malý a střední podnik.

V úvodu praktické části jsem v krátkosti popsal jednotlivé aplikace a vývojové nástroje, které byly použity pro její vypracování. Stejně tak jsem pro ucelení této problematiky představil strukturu školní databáze, protože právě na jejím základě bylo následně možno vytvořit sadu čtyř cvičných úloh. V jednotlivých úlohách jsem na mnou připravených modelových příkladech názorně předvedl, jak data z cvičné databáze transformovat do multidimenzionální podoby, jak navrhnout OLAP datovou kostku a jak si vytvořit jednoduchý a přehledný report pomocí nástrojů, které nabízí platforma Microsoft SQL Server 2008 R2. V poslední úloze jsem následně ukázal, jaké možnosti analýzy nabízí běžně rozšířená aplikace Microsoft Excel 2010.

Přínos této práce logicky spočívá v samotném návrhu jednotlivých úloh. Ty mohou být libovolně využity jako možnost rozšíření učiva, které je probíráno v předmětech Databázové systémy 1 a 2. Zároveň si také myslím, že na základě vypracování těchto praktických úloh mohou studenti lépe pochopit problematiku, která je součástí oborově volitelného předmětu Datové sklady. Tento předmět je vyučován bez podpory výpočetní techniky a proto má spíše teoretický charakter.

# Seznam použitých zdrojů

[1] BASL, Josef. *Podnikové informační systémy: podnik v informační společnosti.* 2., výrazně přeprac. a rozš. vyd. Praha: Grada, 2008, 283 s. ISBN 978-80-247-2279-5

 [2] CACH P, HROCH M. Business intelligence staví na datovém skladu. SystemOnline
 [online]. 2007, č. 5 [cit. 2012-04-14]. Dostupné z: http://www.systemonline.cz/business-intelligence/business-intelligence-stavi-nadatovem-skladu.htm

 [3] DESMARAIS, C. Can Small Business Benefit From Business Intelligence Software?. Inc. [online]. 2011 [cit. 2012-04-20]. Dostupné z: http://www.inc.com/articles/201109/business-intelligence-software-for-smallbusiness\_pagen\_2.html

 [4] HANUSEK L, MÁŠA P. ADASTRA. *Technologie Data Warehousingu a Data Miningu* [online]. 2007 [cit. 2012-04-17]. Dostupné z: http://www.adastra.cz/192\_technologie-data-warehousingu-a-data-miningu.aspx

[5] HROCH, M. *Proč potřebujete corporate reporting*. SystemOnline [online]. 2008, č.6 [cit. 2012-04-19]. ISSN 1802-615X. Dostupné z:

http://www.systemonline.cz/business-intelligence/proc-potrebujete-corporate-reporting-1.htm

[6] HUMPHRIES, M. Data warehousing - návrh a implementace Přel. M. Kocan.1.vyd. Praha: Computer Press, 2001, 257 s. ISBN 80-722-6560-1

[7] INMON, William H. Building the data warehouse: podnik v informační společnosti.3rd ed. New York: J. Wiley, c2002, 412 s. ISBN 04-710-8130-2

[8] KAŠÍK, M. Manažerské informační systémy a jejich úloha v řízení podniku. Brno,
2008. Diplomová práce. Masarykova univerzita. Vedoucí práce Jiří Hřebíček

[9] LACKO, L. *1001 tipů a triků pro SQL*. Vyd. 1. Brno: Computer Press, 2011, 416 s.
 ISBN 978-80-251-3010-0

[10] LACKO, L. Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby. Vyd. 1. Brno: Computer Press, 2009, 456 s. ISBN 978-80-251-2887-9

[11] MAŠEK, M. Datové sklady – principy, metody návrhu, nástroje, aplikace, návrh konkrétního řešení. Jindřichův Hradec, 2009. Diplomová práce. Vysoká Škola Ekonomická. Vedoucí práce Jiří Jelínek.

[12] NOVOTNÝ O, POUR J, SLÁNSKÝ D. *Business Intelligence - Jak využít bohatství ve vašich datech*. Praha : Grada Publishing, a.s., 2005. ISBN 80-247-1094-3

[13] Proč má smysl Business Intelligence i pro podniky střední velikosti. SAP Česká
Republika [online]. [cit. 2012-04-20]. Dostupné z:
http://www.sap.com/cz/sme/solutions/pdf/5835\_TL\_50098162\_csCZ.pdf