

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky



Tvorba OLAP kostky pro reportování obchodních výsledků

BAKALÁŘSKÁ PRÁCE

Studijní program: Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Autor: Mike Holeksa

Vedoucí bakalářské práce: Ing. et Ing. Soňa Karkošková, Ph.D.

Praha, květen 2019

Prohlášení

Prohlašuji, že jsem bakalářskou práci Tvorba OLAP kostky pro reportování obchodních výsledků vypracoval samostatně za použití v práci uvedených pramenů a literatury.

V Praze dne 6. května 2019

.....

Mike Holeksa

Poděkování

Tímto bych chtěl poděkovat své vedoucí práce, Ing. et Ing. Soně Karkoškové, Ph.D., za odborné vedení a připomínky při zpracování této práce.

Abstrakt

Tato bakalářská práce má za cíl seznámit čtenáře s postupem tvorby OLAP kostky pomocí technologie Microsoft SQL Server Analysis Services pro obchodní oddělení společnosti a následným reportingem z této kostky pomocí SQL Server Reporting Services. Součástí práce je také vytvoření datového modelu z dat, která společnost aktuálně používá. Tato data jsou transformována do podoby, která je vhodná pro tvorbu OLAP kostky. Současně čtenáře tato práce seznámí obecně s oblastí business intelligence, datovými sklady a samozřejmě OLAP kostkami v teoretické rovině.

Klíčová slova

Business intelligence, OLAP, OLAP kostka, reporting, SQL Server Reporting Services, SQL Server Analysis Services, datový sklad, MDX

Abstract

The aim of this bachelor thesis is to acquaint reader with the process of creating OLAP cube via Microsoft SQL Server Analysis Services for the sales departement and reporting from this cube with SQL Server Reporting Services. Part of the bachelor thesis also includes creating a data model from data which company currently use. This data are transformed into a form suitable for OLAP cube. This bachelor thesis will also introduce readers to the areas of business intelligence, data warehouses and of course OLAP cubes in the theortical point of view.

Keywords

Business intelligence, OLAP, OLAP cube, reporting, SQL Server Reporting Services, SQL Server Analysis Services, data warehouse, MDX

Obsah

Úvod	11
Cíle práce	11
Rešerše podobných prací	11
1 Business intelligence	13
1.1 Kdy by měl podnik požadovat kvalitní BI řešení	13
2 Architektura a jednotlivé komponenty BI systému	15
2.1 Datový sklad	15
2.1.1 Dimenze	16
2.1.2 Faktové tabulky	16
2.1.3 Granularita dat v datovém skladu	16
2.1.4 Schémata datového skladu	16
2.2 Extract Transform Load (ETL)	18
2.2.1 Získávání dat (Extracting)	18
2.2.2 Transformace dat (Transform)	18
2.2.3 Načítání dat (Load)	19
3 OLAP	20
3.1 Definice OLAP kostky	20
3.2 Jak lze přistupovat k OLAP kostce?	21
3.2.1 Jazyk MDX	21
3.3 Operace v OLAP kostce	22
3.3.1 Roll-up	22
3.3.2 Drill-down	22
3.3.3 Slice and dice	23
3.3.4 Pivot	23
3.4 Schéma pro OLAP	23
3.5 Rozdíl mezi OLTP a OLAP	24
3.6 Typy OLAP kostek	24
3.6.1 MOLAP	24
3.6.2 ROLAP	25
3.6.3 HOLAP	26
3.6.4 Ostatní typy kostek	26
3.6.5 Shrnutí typů kostek	26
3.7 Datové typy v kostce	26

3.8 SQL dotazy při procesování OLAP kostky	27
3.8.1 Procesování dimenzí.....	27
3.8.2 Procesování sloučených dimenzí	27
3.8.3 Procesování faktových dimenzí.....	27
3.9 OLAP a více datových zdrojů	27
3.10 Výhody OLAP kostky.....	28
3.11 Nevýhody OLAP kostky	29
4 Prostředí a zadání pro vytvoření OLAP kostky	30
4.1 Prostředí.....	30
4.2 Zadání pro OLAP kostku.....	30
4.2.1 Definované metriky	30
4.2.2 Definované pohledy na metriky	31
5 Příprava dat	33
5.1 Data o smlouvách	33
5.1.1 Popis faktové tabulky Fct_Agreements	33
5.1.2 Popis dimenze Dim_Agreement_Type	35
5.1.3 Popis dimenze Dim_Sales_Source.....	36
5.1.4 Popis dimenze Dim_Product.....	37
5.1.5 Popis dimenze Dim_Structure	38
5.1.6 Popis dimenze Dim_Date.....	39
5.2 Data o zákaznících	39
5.2.1 Popis faktové tabulky Fct_Customers	40
5.2.2 Popis dimenze Dim_Customer_Attributes.....	41
5.3 Finální datový model v Enterprise Architect	42
6 Tvorba OLAP kostky.....	44
6.1 Vytvoření projektu ve Visual Studio.....	44
6.2 Definování datového zdroje	45
6.3 Nastavení Data Source View	46
6.4 Vytvoření kostky	48
6.4.1 Datový model OLAP kostky	51
6.4.2 Ověření funkčnosti kostky	52
6.4.3 Nasazení kostky na Analysis Services Server.....	52
6.5 První pohled na data z kostky	54
6.5.1 Ověření dat v kostce přes Visual Studio	54
6.5.2 Ověření dat v kostce přes Excel.....	55

6.5.3 Ověření dat v kostce přes výraz MDX	56
6.6 Tvorba hierarchie – časová dimenze	57
6.7 Tvorba hierarchie – obchodní struktura.....	59
6.8 Úprava dimenzí.....	60
6.9 Vytvoření Calculations.....	61
6.10 Nastavení KPI – pro poměr výběrů a očekávaných splátek.....	62
7 Tvorba reportu z OLAP kostky	64
8 Zpětná vazba od uživatelů	68
Závěr	69
Použitá literatura.....	70

Seznam obrázků

Obr. 1 Architektura a komponenty BI, převzato z: (Rad, 2014)	15
Obr. 2 Ukázka star schématu, převzato z: (Rad, 2014)	17
Obr. 3 Ukázka snow flake schématu, převzato z: (Rad, 2014)	18
Obr. 4 OLAP kostka a její osy (vlastní tvorba)	21
Obr. 5 Model faktové tabulky smluv (Enterprise Architect, vlastní tvorba)	33
Obr. 6 Model dimenze typu smlouvy (Enterprise Architect, vlastní tvorba)	35
Obr. 7 Model dimenze zdroje prodeje (Enterprise Architect, vlastní tvorba)	36
Obr. 8 Model dimenze produktů (Enterprise Architect, vlastní tvorba)	37
Obr. 9 Model dimenze obchodní struktury (Enterprise Architect, vlastní tvorba)	38
Obr. 10 Model časové dimenze (Enterprise Architect, vlastní tvorba)	39
Obr. 11 Model faktové tabulky zákazníků (Enterprise Architect, vlastní tvorba)	40
Obr. 12 Model dimenze vlastnosti zákazníků (Enterprise Architect, vlastní tvorba)	41
Obr. 13 Celkový datový model vytvořený pro účely OLAP (Enterprise Architect, vlastní tvorba)	43
Obr. 14 Zakládání projektu ve Visual Studio (vlastní snímek obrazovky)	44
Obr. 15 Ukázka pracovního prostředí Visual Studio (vlastní snímek obrazovky)	45
Obr. 16 Nastavení připojení k datovému zdroji (vlastní snímek obrazovky)	46
Obr. 17 Vytváření Data Source View přes průvodce (vlastní snímek obrazovky)	46
Obr. 18 Datový model Data Source View (vygenerováno ve Visual Studio)	47
Obr. 19 Ukázka okna se seznamem tabulek v Data Source View (vlastní snímek obrazovky)	48
Obr. 20 Definování nové vazby v Data Source View (vlastní snímek obrazovky)	48
Obr. 21 Průvodce vytvářením kostky - volba tabulek (vlastní snímek obrazovky)	49
Obr. 22 Průvodce vytvářením kostky - volba tabulek obsahujících metriky (vlastní snímek obrazovky)	49
Obr. 23 Průvodce vytvářením kostky - volba jednotlivých metrik z tabulek (vlastní snímek obrazovky)	50
Obr. 24 Průvodce vytvářením kostky - výběr dimenzí (vlastní snímek obrazovky)	51
Obr. 25 Průvodce vytvářením kostky - pojmenování kostky a shrnutí (vlastní snímek obrazovky)	51
Obr. 26 Vygenerovaný datový model OLAP kostky (vygenerováno při vytváření kostky ve Visual Studio)	52
Obr. 27 Nastavení umístění pro nasazení kostky (vlastní snímek obrazovky)	53
Obr. 28 Nastavení procesování kostky (vlastní snímek obrazovky)	53
Obr. 29 Okno průběhu procesování kostky (vlastní snímek obrazovky)	54
Obr. 30 Prohlížení dat v kostce ve Visual Studio (vlastní snímek obrazovky)	55
Obr. 31 Vytváření připojení ke kostce přes Excel - volba serveru (vlastní snímek obrazovky)	55
Obr. 32 Vytváření připojení ke kostce přes Excel - volba databáze a kostky (vlastní snímek obrazovky)	56
Obr. 33 Ukázka kostky jako kontingenční tabulky v Excelu (vlastní snímek obrazovky)	56
Obr. 34 Jednoduchý dotaz v jazyce MDX pro ověření funkčnosti kostky v SSMS (vlastní snímek obrazovky)	57
Obr. 35 Vytváření hierarchie v časové dimenzi (vlastní snímek obrazovky)	58

Obr. 36 Vygenerované vztahy v hierarchii (vlastní snímek obrazovky).....	58
Obr. 37 Správně nastavené vztahy v hierarchii (vlastní snímek obrazovky).....	59
Obr. 38 Zobrazení hierarchie při výběru v kontingenční tabulce – Excel (vlastní snímek obrazovky)	59
Obr. 39 Správně nastavené vztahy v hierarchii pro obchodní strukturu (vlastní snímek obrazovky)	59
Obr. 40 Ukázka hierarchie obchodní struktury v kontingenční tabulce (vlastní snímek obrazovky)	60
Obr. 41 Úprava dimenzí - přesun pole pro zobrazení uživatelům (vlastní snímek obrazovky)	61
Obr. 42 Vlastnosti pole v dimenzi - skrytí pole pro uživatele (vlastní snímek obrazovky) ..	61
Obr. 43 Vytváření počítaného pole (vlastní snímek obrazovky)	62
Obr. 44 Ukázka funkčnosti počítaného pole (vlastní snímek obrazovky).....	62
Obr. 45 Nastavení KPI - klíčového ukazatele (vlastní snímek obrazovky).....	63
Obr. 46 Ukázka funkčnosti klíčového ukazatele v Excelu (vlastní snímek obrazovky)	63
Obr. 47 Nastavení připojení k serveru analytických služeb v Reporting Services (vlastní snímek obrazovky).....	64
Obr. 48 Vytváření dotazu MDX přes Query Designer v Reporting Services (vlastní snímek obrazovky)	65
Obr. 49 MDX dotaz v datovém setu vytvořený přes Query Designer (vlastní snímek obrazovky)	66
Obr. 50 Nastavení parametrů v Reporting Services (vlastní snímek obrazovky)	66
Obr. 51 Návrh reportu v Report Builderu (vlastní snímek obrazovky)	67
Obr. 52 Výsledná tabulka z vytvořeného reportu (vlastní snímek obrazovky)	67

Úvod

V dnešní době jsou data, která podniky získávají při vykonávání své podnikatelské činnosti, nejcennějšími informacemi, jež podniky mají. K tomu, aby mohly podniky tato získaná data naplno využívat pro svůj další rozvoj a rozhodování, je nutné, aby byla data kvalitní a maximálně přizpůsobená pro analytické činnosti. Data z podnikových informačních systémů jsou ukládána v propracovaných transakčních systémech, které označujeme jako databáze OLTP (Online Transaction Processing). Z toho důvodu podniky spolupracují s business intelligence profesionály, kteří nastavují procesy získávání kvalitních a konsolidovaných dat, nad kterými lze snadno vytvářet analýzy. Součástí takového řešení je samozřejmě tvorba datového skladu, ze kterého poté vznikají reporty a zároveň OLAP kostka nebo i více kostek, které poskytují možnost analyzovat data analytikům jednotlivých oddělení společnosti, jelikož práce s daty v OLAP kostce je velice snadná i pro uživatele neznalého dotazovacích jazyků. Data v OLAP kostce jsou připravená na míru danému podniku a přístup k takto připraveným datům je rychlejší a snazší než k datům přímo z datového skladu. Dobře zpracovaná OLAP kostka umožní vytvářet velkou část analýz nad kostkou, bez potřeby přípravy ad-hoc nebo i pravidelných reportů z datového skladu.

Cíle práce

Hlavním cílem této práce je vytvoření OLAP kostky skrze technologii Microsoft SQL Server Analysis Services, která bude sloužit obchodnímu oddělení společnosti Provident Financial s.r.o., dále jen Provident. Vzhledem k oboru - nebankovních spotřebitelských úvěrů, ve kterém společnost Provident působí, sleduje obchodní oddělení společnosti zejména ukazatele spojené s objemy a počty prodejů úvěrů spotřebitelům, ale také další ukazatele, se kterými se seznámíme později a zahrneme je současně do vytvářené OLAP kostky.

Součástí práce je také seznámení se s oblastí business intelligence a jejími základními pojmy, jako datový sklad, ETL procesy a samozřejmě OLAP kostkami v teoretické rovině. Budou popsány operace, které lze v OLAP kostkách vykonávat, srovnány jednotlivé typy OLAP kostek a následně i shrnuty výhody a nevýhody řešení OLAP.

Rešerše podobných prací

Pro porovnání byly zvoleny tři práce na podobná témata, která se týkají business intelligence a OLAP kostek.

První z nich je bakalářská práce pod názvem Využití MS EXCEL pro analýzu dat z OLAP databází od Tomáše Majzlíka, kterou zpracoval v roce 2010. Tomáš Majzlík se ve své práci věnuje také tématu OLAP, ovšem z jiného úhlu pohledu. Jelikož se věnuje spíše analýze dat z OLAP přes Microsoft Excel, načež tato práce se věnuje přímo tvorbě samotné OLAP kostky a přípravě dat pro analýzu. Excel je využíván pouze k ukázce funkčnosti OLAP kostky.

Další prací, která byla zvolena pro srovnání, je práce na téma Datové sklady a OLAP v prostředí MS SQL Serveru od Pavla Rudola z roku 2010. Ve své práci se autor věnuje více datovým skladům a prostředí MS SQL Serveru. V této práci je zaměření zejména na OLAP kostky. V teoretické části se v mnohem práce shodují, jelikož i v této práci je část věnována popisu datových skladů a kostek. Co se týče praktické části práce, pak se autor věnuje tvorbě datového skladu, včetně transformace dat. V této bakalářské práci je pozornost věnována pouze přípravě dat pro kostku, nejedná se však o datový sklad.

Posledním srovnáním je práce od Zdeňka Laštůvky s názvem Importy dat z relační databáze do OLAP datových kostek, kterou zpracoval v roce 2012. Autor se ve své práci věnuje importu dat z relační databáze do balíku analytických nástrojů Infor PM. V praktické části se tedy nezabývá tvorbou OLAP kostky, jako je tomu v případě této bakalářské práce. Pokud se podíváme na teoretickou část práce, pak se obě práce věnují popisu datového skladu, transformace dat – ETL a zejména OLAP kostkám.

1 Business intelligence

V této části se seznámíme s oblastí business intelligence a jejími základními pojmy, jako datový sklad, ETL, reporting a následně přejdeme k OLAP kostce.

Business intelligence je zastřešující termín, který se vztahuje ke znalostem, procesům, technologiím, aplikacím a postupům, které usnadňují podnikové rozhodování (1). Z této definice vyplývá, že se jedná o velice komplexní oblast.

Hlavním účelem BI systému je pomáhat osobám s rozhodovací pravomocí dělat správná rozhodnutí na základě výsledků analýzy dat poskytnutých BI systémem (2).

V dnešní době jsou podnikové systémy hojně využívány v téměř každé podnikové sféře. Podniky tyto systémy využívají ke zjednodušení svých procesů, jejich standardizaci a automatizaci. To přímo souvisí se snižováním nákladů na pracovníky, optimalizací výroby a samozřejmě zvyšováním příjmů. Každý z těchto systémů může mít vlastní databázi, do které vstupují veškerá data ze systému. V našem konkrétním případě se bude jednat o Microsoft SQL Server. Podnikové systémy jsou užitečné při každodenních činnostech podniků, např. zaznamenávání všech prodejů, finančních toků, atd. Tato data jsou pro podnik velice důležitá zejména pro analýzu a další plánovanou strategii.

Se zvyšujícím se počtem podnikových informačních systémů přichází potřeba podniku tato data sjednotit. Členové vedení podniku rovněž potřebují tato sjednocená data dále analyzovat. Získaná data jsou podkladem pro naprostou většinu rozhodnutí o dalším fungování podniku, jako je další expanze, optimalizace prodejní sítě, atd.

Cílem business intelligence je zajistit tuto integraci dat z několika podnikových informačních systémů, které společnost využívá, a následně vytvořit konsolidované zprávy pro vedení společnosti ve formě pravidelných reportů a přehledů, které budou sloužit k činění správných rozhodnutí pro další rozvoj podniku.

1.1 Kdy by měl podnik požadovat kvalitní BI řešení

Podnik by měl zvážit BI řešení, jestliže:

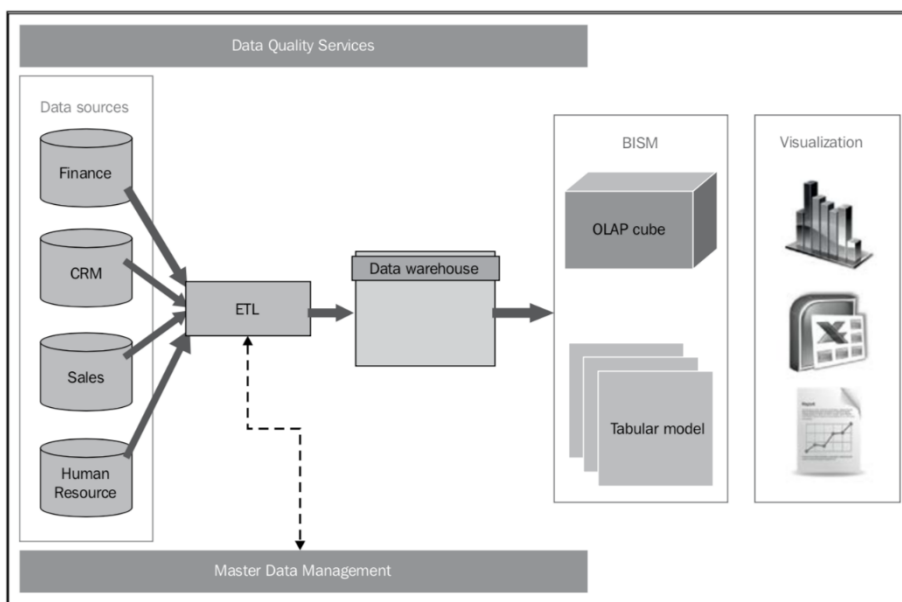
- nejcennější, co podnik využívá pro svá další rozhodnutí a hodnocení, jsou získaná data,
- podnik požaduje z dat získat informace, na které se může spolehnout při vykonávání důležitých rozhodnutí,
- potřebuje efektivně reagovat na podmínky na trhu.

To, co společnosti v dnešní době potřebují, je nabídnout koncovým business uživatelům přístup k informacím přes uživatelsky přívětivé prostředí, kde by si mohli uživatelé vytvářet vlastní analytické reporty (3). Analytické reporty dávají business uživatelům schopnost sledovat trendy, dívat se na data přes různé agregační úrovně, následně procházet data do

detailu pomocí kliknutí myši. Ve většině případů to obnáší tvorbu datového skladu, zejména v případě, pokud chceme informace využívat v několika reportech. Důvodem vytvoření datového skladu je zejména možnost kombinace několika různých zdrojů do jedné struktury (3). Taková struktura je poté využívána pro reportování a analýzy.

2 Architektura a jednotlivé komponenty BI systému

Pro komplexní pochopení systému BI je potřeba poznat jednotlivé součásti a pochopit jak tyto komponenty navzájem spolupracují. Následující schéma ukazuje hlavní součásti systému business intelligence.



Obr. 1 Architektura a komponenty BI, převzato z: (Rad, 2014)

2.1 Datový sklad

Základním prvkem BI architektury je datový sklad. Datový sklad je centrální databázi vytvořenou zejména proto, aby podnik měl data ze všech zdrojů použitelná a dostupná (4).

Jedná se databázi, která je zároveň vytvořena pro účely reportingu a analýzy dat. Datový sklad je navržený tak, aby maximálně zkracoval dobu odezvy a zvyšoval výkon dotazů pro reporty a analýzy.

Datový sklad je integrovaný a stálý. Integrita zajišťuje, že jsou data ukládána v rámci celého podniku do datového skladu pouze jednou. Stálost datového skladu označuje, že data do datového skladu nevstupují ručním zásahem, ale pomocí předem nastavených procesů z podnikových informačních systémů.

Data vstupují do datového skladu prostřednictvím ETL procesu a primárním účelem je pokrytí požadavků na reporting a analýzu dat. Z toho plyne, že datový sklad není standardní transakční databázi, do které operační systém zapisuje data.

Mezi základní komponenty datového skladu patří faktové tabulky a dimenze, jimiž je tvořený datový model. Vztahy mezi těmito tabulkami jsou definované přes cizí a primární klíče (primární klíč dimenze je cizím klíčem ve faktové tabulce).

Pro možnosti analýzy za různá časová období je nutné ukládat data v datovém skladu také historicky, tzn., data musí obsahovat časové rozlišení.

2.1.1 Dimenze

Dimenze jsou tabulky, které obsahují popisné informace. Jedná se o tabulky se zákazníky, adresy zákazníků, produkty, atd. Každá dimenze má primární klíč, který nám slouží k propojení dimenze s faktovou tabulkou.

2.1.2 Faktové tabulky

Tabulkám, které obsahují detailní hodnoty pro metriky nebo fakta, říkáme faktové (5). Faktové tabulky zabírají většinu místa v datovém skladu (přibližně 80%). Jedná se o seznam skutečností (např. jednotlivých prodejů). Ve faktových tabulkách se nachází velké množství záznamů, které lze napojit na jednotlivé dimenze.

2.1.3 Granularita dat v datovém skladu

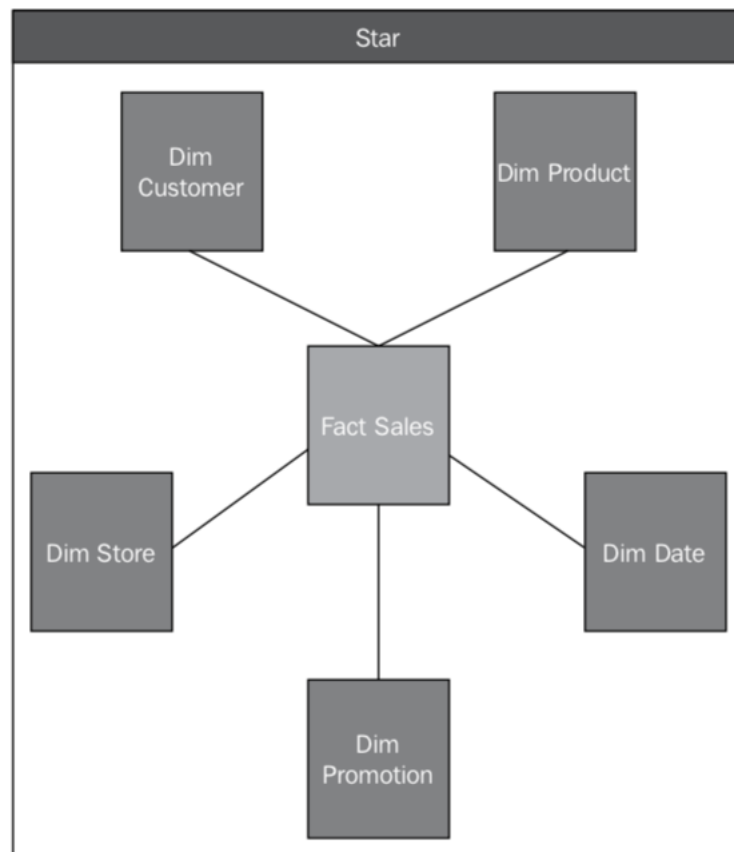
Granularita dat nám označuje, v jakém detailu chceme data používat. V našem případě si můžeme uvést příklad, zda chceme vidět konkrétní prodané půjčky, nebo si vystačíme s počty prodaných půjček po jednotlivých oblastech. Zvolená granularita datového skladu ovlivňuje jaké a kolik dimenzí budeme vytvářet.

2.1.4 Schémata datového skladu

Pro vytvoření vztahů mezi faktovými tabulkami a dimenzemi používáme dvě různá schémata: jedná se o star (hvězdicové) schéma a snow flake schéma, což je v překladu schéma sněhové vločky.

Star schéma

Star schéma se nejčastěji používá pro modelování multidimenzionálních dat. Ve star schématu je faktová tabulka uprostřed mezi dimenzemi a ty jsou propojené s faktovou tabulkou vždy pouze jedním vztahem. Faktová tabulka obsahuje cizí klíče, které jsou ve vztahu s primárními klíči v dimenzích. V ideálním případě, by neměla existovat dimenze, která by měla vztah s faktovou tabulkou pouze přes další dimenzi.



Obr. 2 Ukázka star schématu, převzato z: (Rad, 2014)

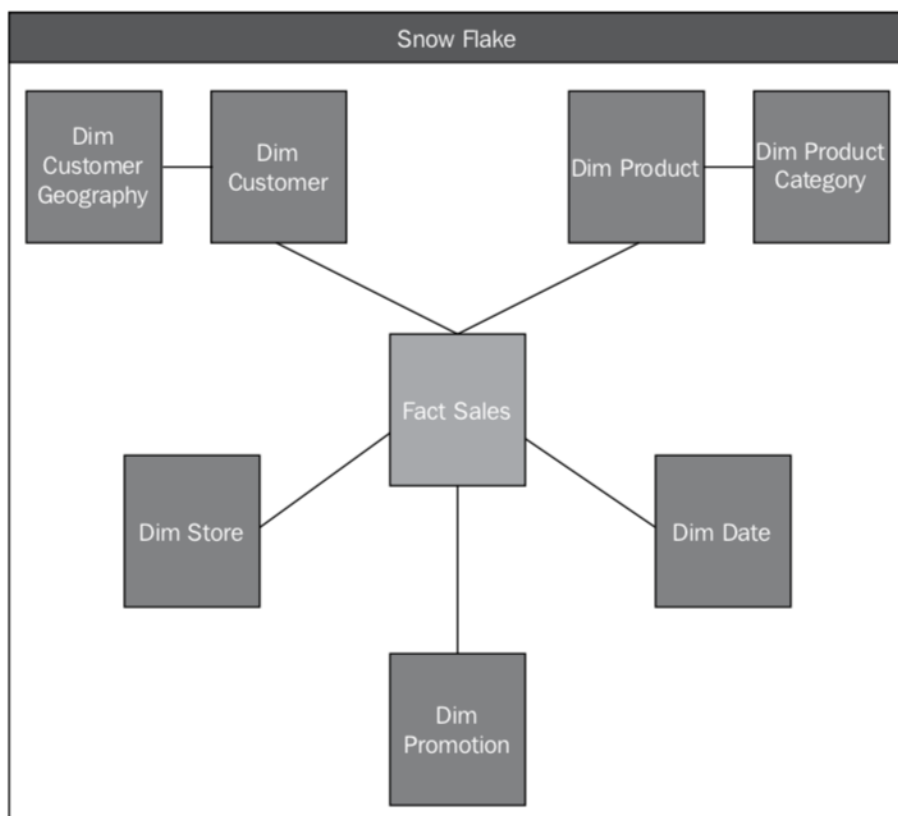
Vzhledem k jednoduchému propojení faktové tabulky s dimenzemi je odezva dotazů rychlejší, jelikož nedochází ke spojování několika dimenzí.

Snow flake schéma

Schéma snow flake není nic jiného, než hvězdicové schéma, které je zkomplikováno přítomností tabulky, která spojuje dimenze mezi sebou (6).

Schéma sněhové vločky se liší od hvězdicového schématu tím, že dochází k propojení dimenzí mezi sebou a teprve následně se propojují na faktovou tabulku. Tabulky dimenzí jsou v tomto schématu normalizované. V praxi to znamená, že schéma sněhové vločky musí spojovat více tabulek a proto je dotazovací výkon slabší než ve schématu hvězdicovém.

Pokud dochází k častým změnám v dimenzích a jejich uspořádání (hierarchií), pak je snow flake schéma výhodnější. Datový model při snow flake schématu je méně přehledný než v případě star schématu.



Obr. 3 Ukázka snow flake schématu, převzato z: (Rad, 2014)

2.2 Extract Transform Load (ETL)

Většinou pro BI systém potřebujeme data z více než jednoho zdroje. Právě pomocí ETL procesu jsme schopni datový sklad plnit daty z několika zdrojů. Jak se můžeme z názvu domnívat, při ETL procesu dochází nejdříve k získání (Extraction) dat z různých zdrojových systémů. Následně probíhá úprava dat do požadované podoby (Transformation), jejich konsolidace a také odstranění chyb v datech. Následně jsou data připravená pro nahrání do datového skladu v požadované struktuře (Load).

Stručně si popíšeme jednotlivé fáze ETL procesu:

2.2.1 Získávání dat (Extracting)

Na začátku procesu ETL dochází k získávání dat ze zdrojových systémů podniku. Tato data jsou získávána bez jakýchkoliv úprav. Data můžeme dočasně ukládat do databázových tabulek nebo do souborů. V této fázi klademe důraz zejména na rychlost procesu a minimální zátěž pro zdrojové systémy. Současně by měla zůstat zachována možnost opakovaného spuštění, v případě, že nastane nějaký problém.

2.2.2 Transformace dat (Transform)

Součástí kroku transformace dat jsou zejména procesy, které upravují zdrojová data tak, aby bylo možné data načíst do datového skladu a zároveň nad daty provádět smysluplné analýzy.

Většinou dochází k selekci sloupců, které budeme používat v datovém skladu. V procesu transformace dat můžeme také vytvářet nové kalkulované sloupce, které budou pro potřeby analýzy v datovém skladu již předem vypočítané.

V tomto kroku ETL procesu se provádí také čištění dat. Jelikož do zdrojových systémů mohou vstupovat i méně kvalitní data, například chybným ručním zásahem, je nutné, pro zvýšení kvality analýzy a podnikových dat obecně, tato chybná data opravit v procesu čištění dat. Dochází zde k opravám dat a odstraňování duplicitních dat.

Vzhledem k tomu, že nám data plynou z několika podnikových informačních systému, může se stát, že ve skutečnosti či pro účely analýzy stejná data, jsou v každém systému označena jinak. V kroku konsolidace dat dochází k jejich sloučení.

Jako příklad si můžeme uvést sladění zadaných adres zákazníků z několika podnikových systémů. Vzhledem k tomu, že některá města lze zadat v různých zkrácených či delších mutacích, je pro potřeby analýzy ze všech možností vytvořit pouze jednu, abychom byli schopni tato data sloučit právě na název konkrétního města.

Součástí transformace dat je také změna některých číselníkových kódů na jejich popis. Například pohlaví může být v zdrojovém systému označeno kódy 1 a 2, my při procesu transformace tyto kódy změníme na hodnoty Ž a M, které následně použijeme při načítání do datového skladu.

2.2.3 Načítání dat (Load)

Po transformaci jsou data připravená k načtení do datového skladu. Při načítání dat do datového skladu může docházet k procesům, které zneplatní již neaktuální záznamy. Tyto záznamy mohou být také mazány, ale v případě, kdy chceme data historizovat, pak dochází právě k zneplatnění předchozího údaje. Data jsou načítána do předem stanoveného schématu, který zjednodušuje práci s datovým skladem a také rychlost dotazů do datového skladu.

3 OLAP

OLAP je zkratka pro online analytické zpracování a můžeme jej označit za poslední díl stavebnice při řešení datového skladu. Tento pojem nás většinou odkazuje na specializované nástroje, které udělají data z datového skladu snadno přístupnými (7).

OLAP provádí multidimenzionální analýzu podnikových dat z různých databázových systémů současně a poskytuje možnost komplexních výpočtů, analýzu trendů a sofistikovaného modelování dat (8). Je tedy základem pro mnoho druhů podnikových procesů:

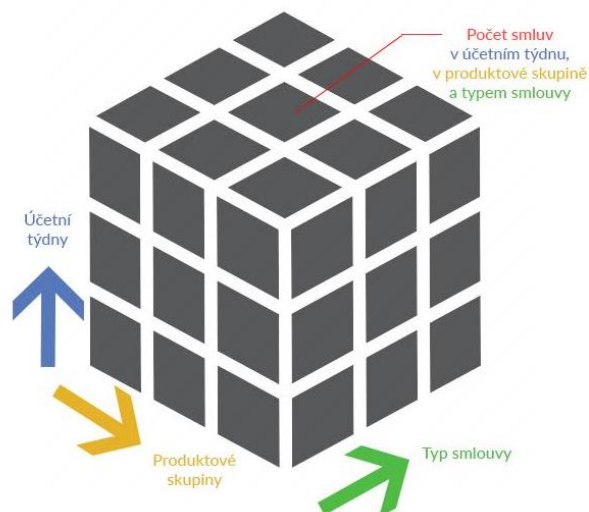
- sledování výkonnosti podniku,
- plánování,
- rozpočtování,
- prognózování,
- finanční výkazy,
- analýzu,
- reportování.

3.1 Definice OLAP kostky

OLAP kostka je specifická metoda, která slouží k ukládání velkého množství dat v prostředí datového skladování (data warehousing). Pojem “kostka” označuje několik předem definovaných dimenzí, faktů (měřítek) a hierarchií, pomocí níž mohou uživatelé přistupovat k uloženým datům, vytvářet si tak různé pohledy na data a z toho následně tvořit analýzy.

OLAP kostka je v nejlepším případě postavená nad datovým skladem, což znamená, že data jsou již sloučena a vyčištěna. Uložená data v OLAP kostce jsou již předem vypočítaná a připravená k okamžitému použití. V praxi to znamená, že pro přístup k datům není potřeba složitých a pomalých dotazů do datového skladu (relační databáze), ale uživatel má data k dispozici téměř ihned.

Tento vícerozměrný přístup k datům umožňuje velice pokročilou podnikovou analýzu, jelikož kombinuje a seskupuje data takovým způsobem, který je přesně přizpůsobený potřebám daného podniku (např. v našem případě OLAP kostka pro obchodní oddělení bude připravená pro sledování obchodních ukazatelů - zejména počty a hodnoty prodejů, které budou sledovat v čase, po jednotlivých regionech, atd.).



Obr. 4 OLAP kostka a její osy (vlastní tvorba)

3.2 Jak lze přistupovat k OLAP kostce?

Do OLAP kostky mohou uživatelé přistupovat nejčastěji přes Microsoft Excel, kde lze snadno používat data jako kontingenční tabulku. Další variantou jsou dotazy pomocí dotazovacího jazyka MDX. Pro dotazování můžeme využít například SQL Server Management Studio. Blíže si popíšeme jazyk MDX níže a později si také ukážeme MDX dotazy do vytvořené kostky pro reportování obchodních výsledků.

3.2.1 Jazyk MDX

MDX (Multidimensional eXpressions) je jazyk vytvořený pro analytické dotazování. Jedná se o doplněk (rozšíření) jazyka SQL, který je přizpůsobený dotazování a přístupu k multidimenzionálním datům. Jazyk MDX je využíván k získávání dat uložených v SSAS kostce, jenž vrací datový set, který se skládá z osy a buněk.

Tento dotazovací jazyk byl původně vyvinut společností Microsoft v devadesátých letech a byl využíván i dalšími dodavateli multidimenzionálních databází.

Téměř vždy vývojář časem čelí novým požadavkům od business uživatelů, se kterými se původně při tvorbě datového skladu nepočítalo. Stává se také, že tyto požadavky nemůžeme vykonávat v datovém skladu, ale musíme si s tím poradit přímo v kostce.

V takovém případě můžeme vytvořit kalkulovaná pole. Řešení analytických služeb SSAS od Microsoftu takovou možnost nabízí právě pomocí výrazového jazyku MDX. Všechny dotazy z aplikace klienta jsou odesílány na SSAS server jako dotaz v jazyce MDX a výsledky dotazu jsou navraceny v multidimenzionálním formátu (2).

Příkladem kalkulovaného pole může být například výpočet marže, kdy odečteme kupní cenu od ceny prodejní, přičemž tuto hodnotu nemusíme udržovat nikde v datovém skladu.

Základní syntaxe MDX

Základním dotazem je stejně jako v jazyce SQL klauzule SELECT, FROM a WHERE.

Výpis skriptu 1 - Základní syntaxe MDX (vlastní tvorba)

```
SELECT [<axis_specification>  
      [, <axis_specification>...]]  
FROM [<cube_specification>]  
[WHERE [<licer_specification>]]
```

V jazyce MDX lze napsat dotaz, který využívá stovky os, je však omezený klientem, např. SSMS, což v praxi znamená, že pomocí MDX jsme schopni data zobrazit pouze ve dvou osách.

Pomocí SELECT vybíráme osy, na nichž chceme zobrazit požadovaná data. Za příkazem FROM se nachází specifikace kostky, tedy hodnot, které chceme sledovat. Pomocí příkazu WHERE můžeme zvolit jednu hodnotu z dimenze, která nás zajímá pro danou analýzu (např. konkrétní rok).

3.3 Operace v OLAP kostce

Při práci s OLAP kostkou můžeme provádět různé operace tak, abychom dosáhli požadovaného výsledku (pohledu na data). Procházení přednastavenými hierarchiemi, vytváření vlastních hierarchií, filtrování či volba, co uvidíme v řádku a co ve sloupci, je zcela na uživateli. K tomu slouží následující základní operace:

3.3.1 Roll-up

Roll-up můžeme také nazývat jako konsolidaci nebo agregaci dat na určitou vyšší úroveň. Tato operace je prováděna procházením hierarchií směrem vzhůru. (9) V našem případě si můžeme tuto operaci představit následovně:

Když budeme mít objem prodeje po jednotlivých sekcích a postupně budeme kostku procházet směrem vzhůru, přes oblasti, které jsou nadřazené sekcím, až na úroveň celé České republiky. Data se postupně agregují na vyšší územní celky.

3.3.2 Drill-down

Při drill-downu dochází k rozdělení dat na menší části. Drill-down je opakem k předchozí operaci roll-up, jelikož z agregovaných dat postupujeme k detailnějším datům.

Velice dobrým příkladem operace drill-down je případ, kdy máme data agregovaná na úroveň jednoho roku, a následně jdeme více do detailu, přes čtvrtletí až na jednotlivé měsíce, či dokonce dny v roce.

3.3.3 Slice and dice

Při operaci slice (řezání) si vybereme jednu hodnotu z některé dimenze a pomocí ní vytvoříme podmnožinu kostky. (9) Při řezání tak dochází ke snížení počtu dimenzí, které používáme. Příkladem může být, když chceme sledovat počty prodaných půjček v regionálních oblastech po jednotlivých produktových skupinách, ale pouze v jednom vybraném roce. Tímto vybraným rokem tzv. kostku řízeme. Dimenzi sice použijeme, ale pouze jako filtr našeho výběru dat.

Pokud provádíme operaci dice, pak vybíráme specifické hodnoty z několika dimenzí a tím vytvoříme novou vrstvu pod kostkou. Jestliže máme opět počty prodaných půjček v jednotlivých letech, ale zajímají nás pouze specifické produktové skupiny a vybereme si například produktové skupiny A, B, C, pak dochází k operaci dice. (provádíme výběr jen několika hodnot z dimenze)

Rozdíl mezi operaci slice a dice je ten, že při operaci slice dochází k vytvoření podmnožiny s nižším počtem dimenzí, tzn., dochází k výběru hodnoty pouze z jedné dimenze. Operace dice nám umožňuje vybírat specifické hodnoty z několika dimenzí. Při používání kostky lze rozdíl mezi slice a dice shrnout velice jednoduše. Při operaci slice může uživatel vybrat pouze jednu hodnotu (např. konkrétní rok nebo produktovou skupinu), naopak při operaci dice uživatel vybírá hodnot více (několik let a několik produktových skupin).

Tyto dvě operace jsou při práci s OLAP kostkami velmi často využívány, jelikož umožňují uživatelům přistupovat k pohledům na konkrétní data, která potřebují pro danou analýzu.

3.3.4 Pivot

Tato operace je spíše otázkou správné či lepší prezentace dat, jelikož při operaci pivot dochází pouze k záměně osy x za osu y. Při práci v Excelu operace pivot znamená záměnu řádků za sloupce a zase naopak sloupců za řádky. Pro lepší prezentaci je vhodné dimenzi, která obsahuje více hodnot použít na řádcích a do sloupců vložit dimenzi, která má hodnot méně. Pro uživatele je přívětivější posun nahoru a dolů, než pohyb do stran.

3.4 Schéma pro OLAP

OLAP kostka by měla být vytvářena spíše nad hvězdicovým schématem, dost to následně usnadní práci. Pokud se však kostka vytváří nad datovým skladem, který obsahuje snow flake schéma, pak je vhodné provést úpravu, která bude data prezentovat ve schématu hvězdicovém.

Můžeme využít pohledů (views), abychom transformovali podkladové snow flake schéma na star schéma. Spojením tabulek v pohledech můžeme docílit skrytí snow flake schématu, jelikož víme, jak by měly být tabulky ve snow flake schématu spojené, můžeme do Analysis Services prezentovat již čisté hvězdicové schéma (6).

3.5 Rozdíl mezi OLTP a OLAP

Když hovoříme o databázích, pak se často setkáváme s operačními databázemi a analytickými databázemi.

Mnoho z principů, které jsou používány v tradičních relačních databázích, jsou v případě multidimenzionálních databází praktikovány zcela opačně (10).

V případě operačních databází se jedná o databáze, které jsou primárně používané pro online zpracování transakcí (online transaction processing - OLTP). Jsou důležitou součástí většiny firem, jelikož jejich úkolem je sběr, změna a správa data pro každodenní fungování podniku. Tato data jsou v téměř neustálém toku a dochází k jejich častým změnám - vždy jsou aktuálním odrazem skutečnosti. Jako příklad si můžeme uvést například databázi CRM systému, kdy při kontaktu se zákazníkem musí mít operátor zcela aktuální data o stavu zákazníka, aby mohl úspěšně řešit všechny jeho požadavky.

V databázích pro online analytické zpracování (online analytical processing - OLAP) je jejich hlavním úkolem poskytovat kvalitní data pro analýzu. Analytická databáze je vhodná, pokud chceme sledovat vývoj v čase (trend) nebo zobrazovat data za delší časové období. V těchto databázích se nacházejí většinou statická data, jelikož jsou to ve velké míře data historická. Analytické databáze využívají všechny společnosti, které provádějí rozsáhlejší analýzy a předpovědi vývoje podniku.

3.6 Typy OLAP kostek

OLAP kostky můžeme rozdělit na několik typů podle toho, jak přistupují k podkladovým datům, zda si kostka drží data v sobě, nebo zda je kostka jen prostředníkem k přístupu do relační databáze datového skladu. Jednotlivé typy OLAP kostek si dále představíme.

3.6.1 MOLAP

Multidimenzionální OLAP je spíše tradičnější způsob analýzy. V případě MOLAP jsou data uložena ve vícerozměrné kostce. Data tedy nejsou uložena v relační databázi, ale v proprietárních formátech.

Výhody MOLAP

Hlavní výhodou MOLAP je naprosto vynikající výkon. Kostka je konstruována tak, aby data byla rychle načtena. Zároveň je kostka již připravená pro řezací (slice a dice) operace. Další výhodou je provádění složitějších výpočtů, jelikož všechny výpočty jsou již předem vypočítány při vytváření kostky. Což uživateli nejenže tyto výpočty umožňuje, ale také výsledky výpočtů jsou ihned k dispozici. MOLAP má menší velikost dat ve srovnání s daty uloženými v relační databázi a to díky kompresním technikám.

Nevýhody MOLAP

Jednou z nevýhod MOLAP kostek je omezené množství dat, které můžeme zpracovat. Vzhledem k tomu, že všechny výpočty jsou prováděny při sestavování kostky, není možné do samotné kostky dát velké množství kalkulovaných dat. To ovšem neznamená, že MOLAP kostka neumí s velkým množstvím dat pracovat. Je však nutné tato data sumarizovat na rozumně definovanou úroveň.

Zpracování trvá poměrně dlouho zejména při větších objemech dat. To se obvykle řeší tak, že se zpracovávají pouze jednotlivé části kostky, tedy data, u nichž došlo ke změně (obvykle nová data), namísto celé sady dat.

Další nevýhodou jsou další investice do kostky, jelikož MOLAP kostky jsou proprietární a často znalost kostky není přímo ve společnosti, ale je dodávána externě. Z toho plynou nemalé výdaje na úpravy kostky, které provádí dodavatel řešení.

Procesování kostky trvá mnohem déle, jelikož dochází k přípravě kalkulovaných dat.

3.6.2 ROLAP

Relational OLAP nevyžaduje předchozí vypočítání dat, jelikož pracuje s daty uloženými přímo v relační databázi, díky níž nabízí vzhled a tradiční funkce jako slice and dice pro data uložená v databázi. V případě, že využíváme akci slice and dice, ve skutečnosti přidáváme podmínky do klauzule WHERE v dotazu do relační databáze. ROLAP představuje standardní dotaz do relační databáze, využívající tabulky této databáze pro získání odpovědi. Vzhledem k tomu, že pracujeme přímo s podkladovou databází, pak nejsme omezení obsahem kostky a můžeme zadávat jakékoliv dotazy i na nejnížší úroveň granularity, kterou máme v databázi.

Pokud plánujeme využívat ROLAP, pak musíme mít také vhodně navrženou relační databázi.

Výhody ROLAP

Největším rozdílem oproti MOLAP a zároveň výhodou je možnost zpracování téměř neomezeného množství dat - omezením ROLAP je pouze velikost podkladové relační databáze, sám o sobě žádným způsobem ROLAP množství dat neomezuje.

Jako další výhodu můžeme uvést využití funkcionality relačních databází. Vzhledem k tomu, že se ROLAP nachází na samotném vrcholu relační databáze, může tyto funkce využívat.

Při použití ROLAP můžeme využít kontroly autorizace přes nastavení databáze, jako je například zabezpečení na úrovni řádků.

Nevýhody ROLAP

Každý ROLAP dotaz je ve své podstatě dotazem SQL (nebo více SQL dotazy) do relační databáze. Vzhledem k tomu, může být délka trvání dotazu dlouhá, a to zejména v případě, kdy je velké množství podkladových dat. U ROLAP často narazíme na omezené funkce SQL,

kteře neodpovídají všem analytickým potřebám (např. obtížné provádění složitých výpočtů pomocí SQL). Dochází tedy k omezení na možnosti jazyka SQL.

3.6.3 HOLAP

Hybrid OLAP se snaží spojit výhody MOLAP a ROLAP. Pro souhrnné informace využívá HOLAP technologii pro rychlejší výkon, ale v případě, že potřebujeme získat detailní data, pak může kostka sahat přímo do podkladových relačních dat.

3.6.4 Ostatní typy kostek

WOLAP - Web OLAP

Stručně řečeno, jedná se o web OLAP kostku, která je dostupná prostřednictvím webového prohlížeče. Na rozdíl od standardní OLAP kostky se v případě WOLAP jedná o tříúrovňovou architekturu, která se skládá z komponent: klient, middleware a databázového serveru.

DOLAP - Desktop OLAP

Desktop OLAP je založený na myšlence, že si uživatel část dat stáhne z databáze či jiného zdroje a bude s těmito daty pracovat lokálně na své ploše. DOLAP je nejjednodušším řešením OLAP a jedná se zároveň o nejméně nákladné řešení. Ve srovnání s ostatními typy je však velice DOLAP velice omezený svou funkcionalitou.

3.6.5 Shrnutí typů kostek

Dva hlavní typy kostek, tedy MOLAP a ROLAP patří mezi nejčastěji využívané typy. Osobně bych si dovolil tvrdit, že MOLAP je častěji využívaným typem. MOLAP přináší nespornou výhodu zejména ve své rychlosti načítání požadovaných dat. Na druhou stranu vyžaduje důkladnou analýzu ještě před samotnou tvorbou kostky, aby bylo možné přesně definovat požadované metriky, které budou pomocí kostky sledovány. Naopak ROLAP je pomalejší, ale lze přes něj sledovat data skrz relační databáze a tak uživatel nepřichází o možnost pohledu na veškerá data v databázi, která mu mohou občas posloužit.

3.7 Datové typy v kostce

Když jsou připravována data pro kostku, je nutné volit vhodné datové typy pro jednotlivé sloupce. Pokud jsou zvoleny správné datové typy, významně se tím urychlí dotazování kostky, stejně tak i její procesování. Cílem je volit co nejmenší datové typy. Zde je příklad, jaké by se měly volit datové typy:

Tabulka 1 - Příklad datových typů pro data v kostce (převzato z Expert Cube Development with SSAS Multidimensional Models)

Faktová tabulka - typ sloupce	Nejvhodnější datový typ
-------------------------------	-------------------------

Klíče tabulky (primární, cizí)	tinyint, smallint, int nebo bigint
Datum - klíč	int ve formátu rrrrmmdd
Celočíselné metriky	tinyint, smallint, int nebo bigint
Číselné metriky	smallmoney, money, real nebo float (musíme brát v úvahu, že decimal je náročnější na procesování než money nebo float)

3.8 SQL dotazy při procesování OLAP kostky

Jakmile se spustí procesování OLAP kostky, pak se jedná vlastně o zasílání dotazů SQL do relační databáze, z níž dostáváme potřebné informace. Často se nejedná pouze o obyčejné SELECTy, ale bývají generovány i složitější dotazy.

Pokud je to možné, pak tyto dotazy běží současně, aby se zkrátil čas procesování.

3.8.1 Procesování dimenzí

Když je procesována dimenze, pak probíhá několik dotazů. Pro každou hodnotu z dimenze se při procesování vytváří dotaz ve formát `SELECT DISTINCT název sloupce`, kde název sloupce je sloupcem, který obsahuje požadovanou hodnotu.

3.8.2 Procesování sloučených dimenzí

Jestliže má dimenze hodnoty, které si přebírá z jiné dimenze, pak je generován JOIN. To se stává, jestliže pracujeme se snow flake schématem. Tento dotaz je vykonáván přímo SQL Serverem.

3.8.3 Procesování faktových dimenzí

Procesování dimenzí, které jsou přímo spojené s faktovou tabulkou se generují také dotazem `SELECT DISTINCT`.

3.9 OLAP a více datových zdrojů

V případě, že by měla být OLAP kostka postavena nad více databázemi, pak lze narazit na problém s výkonem.

Všechny dimenze a faktové tabulky by měly existovat ve stejné relační databázi. Pokud by bylo použito více zdrojů, pak lze zjistit, že jeden z nich je označován jako primární zdroj a

všechny další zdroje jsou připojovány přes primární zdroj pomocí funkce SQL Serveru OPENROWSET, což může vést k problémům s výkonem při procesování kostky (6).

3.10 Výhody OLAP kostky

Rychlost

Při procesování OLAP kostky se předem vypočítají a fyzicky uloží agregovaná data. (11) Z toho důvodu, když business uživatel spustí dotaz do kostky, odpověď je rychlejší, než při klasickém SQL dotazu. Další výhodou je ukládání dotazů do mezipaměti, což znamená, že když je stejný dotaz spuštěn vícekrát, pokusí se kostka získat výsledek právě z mezipaměti, čímž dosáhne vyšší rychlosti.

Multidimenzionální analýza

Skrze OLAP kostku mohou uživatelé vytvářet velice snadno analýzy, jelikož pomocí operací, které s kostkou můžeme provádět, docílíme požadovaných výsledků za pomoci pár kliknutí. V případě Excelu lze tato data snadno zobrazit v grafech a dále s nimi pracovat. Pomocí operaci drill-down se lze snadno dostat na požadovaný detail zobrazení dat, případně pomocí operace slice snadno data vyfiltrovat.

Hierarchie

Při sestavování OLAP kostky si lze předem definovat hierarchie dat, které poté mohou uživatelé používat, aniž by museli volit jednotlivé položky hierarchie. Uživatelé si tak ihned zobrazí data za jednotlivé roky, čtvrtletí, měsíce a týdny, aniž by do kontingenční tabulky museli vybírat všechna tato časová hlediska jednotlivě.

Předem nastavené výpočty

V OLAP kostce lze předem nastavit složité výpočty, které by pomocí SQL bylo možné získat zdlouhavým skriptem. Jedná se například o year-to-date výpočty, klouzavé průměry a porovnání aktuálních dat k datům z minulých let.

Snadný přístup pro uživatele

Uživatelé mohou ke kostce snadno přistupovat přes Excel. Kostka se zobrazí jako kontingenční tabulka. Uživatelé tak mohou pro standardní analýzy využívat právě OLAP kostek, namísto zadávání požadavku na vytvoření reportu.

Nastavení zabezpečení kostky

Při nastavení zabezpečení kostky lze nastavit koncovým business uživatelům různá oprávnění, tedy přístup pouze k určitým částem (dimenzím a metrikám) kostky (11). Uživatelé tak přistupují pouze k oblastem, které jsou pro ně zajímavé, případně pouze k částem kostky, které mohou v rámci ochrany dat vidět.

3.11 Nevýhody OLAP kostky

Vysoké náklady

Ve většině případů tvorba komplexní OLAP kostky levným řešením. Z toho důvodu nevyužívají tuto technologii všechny podniky, které by uvítaly možnost self-service analýzy.

Nutnost přípravy dat

Pro využití dat je potřeba tato data do kostky připravit tak, aby s nimi bylo možné v kostce pracovat. Nevýhodou je, že není možné rychle doplňovat další oblasti do OLAP kostky bez předchozí analýzy a přípravy těchto dat.

4 Prostředí a zadání pro vytvoření OLAP kostky

4.1 Prostředí

Kostka bude vytvářena na Microsoft SQL serveru, verze 2016. K vytváření bude použito SQL Server Management Studio ve verzi 17 a pro práci s kostkou Visual Studio ve verzi 2015. Následně pro vytvoření jednoduchého reportu nad vytvořenou OLAP kostkou se použije Reporting Services ve verzi 2016.

4.2 Zadání pro OLAP kostku

Nyní bude pozornost věnována tvorbě samotné OLAP kostky pro reportování obchodních výsledků ve společnosti Provident Financial s. r. o., která poskytuje nebankovní úvěry. Veškerá data budou sledována od počátku roku 2017. Ukázky reálných dat budou částečně anonymizovány tak, aby nebylo možné identifikovat skutečné prodeje.

4.2.1 Definované metriky

Tyto metriky byly definovány obchodním oddělením společnosti a současně tyto metriky by měla být vytvářena kostka schopna zobrazit.

Tabulka 2 - Popis definovaných metrik v zadání pro OLAP kostku (vlastní tvorba na základě informací od obchodního oddělení)

CZ název	EN název	Popis
Objem prodeje	Issue volume	Počet všech poskytnutých půjček v účetním týdnu.
Hodnota prodeje	Issue value	Hodnota v Kč všech poskytnutých půjček v účetním týdnu.
Průměrná výše půjčky	Average issue value	Průměrná hodnota poskytnutých půjček (celková hodnota / počet kusů celkem)
Počet zrušených smluv	Cancellation volume	Počet půjček, které byly zrušeny ze strany zákazníka – odstoupení do 14 dní.
Počet doplacených smluv	Paid up volume	Počet doplacených půjček v účetním týdnu.
Počet odepsaných smluv	Write off volume	Počet půjček v účetním týdnu, které jsme byli nuceni odepsat z důvodu neplacení zákazníkem.

Počet aktivních zákazníků	Active customers	Počet aktivních zákazníků, což znamená, že mají minimálně jednu aktivní půjčku.
Počet doplacených zákazníků	Paid customers up	Počet zákazníků, které svou půjčku již doplatili.
Počet odepsaných zákazníků	Written off customers	Počet zákazníků, kteří zůstali odepsáni.
Hodnota výběrů splátek	Collection value	Hodnota výběrů splátek v Kč, které byly u zákazníků vybrány.
Hodnota očekávaných splátek	Debit value	Hodnota očekávaných výběrů v Kč, které by měli zákazníci zaplatit.
Poměr vybraných splátek vůči očekávaných splátkám	Collection performance	Dělení hodnoty výběrů a hodnoty očekávaných splátek. Výsledek by měl být zobrazený v procentech.

4.2.2 Definované pohledy na metriky

Obchodním oddělením byly dodány následující možnosti pohledů, přes které by se uživatelé kostky mohli na metriky výše dívat.

Tabulka 3 - Popis definovaných pohledů na metriky v zadání pro OLAP kostku (vlastní tvorba na základě informací od obchodního oddělení)

CZ název	EN název	Popis
Datum	Date	Časový pohled, dle Provident účetního kalendáře, kdy týden začíná středou a končí úterým.
Obchodní struktura	Operational structure	Pohled na metriky po organizační struktuře společnosti.
Produktová skupina	Product group	Pohled na vytvořené smlouvy přes definované produktové skupiny.
Typy půjček	Agreement types	Rozdělení půjček, zda se jedná o nového či stávajícího zákazníka.
Prodejní kanál	Sales source	Pohled na smlouvy dle prodejního kanálu. Zda obchodní zástupce tuto smlouvu vytvořil na základě leadu od společnosti nebo ji získal sám bez pomoci.

Věk zákazníka	Customer age	Pohled přes věkové skupiny zákazníků.
Pohlaví	Gender	Pohled na metriky skrze pohlaví zákazníků.
Příjmová skupina	Income group	Pohled přes příjmové skupiny zákazníků.

5 Příprava dat

Vzhledem k tomu, že pro systém, nad kterým bude tvořena OLAP kostka, neexistuje datový sklad, budou data připravována formou tabulek. Tyto tabulky pak budou sloužit speciálně pro následné vytvoření OLAP kostky.

Tabulky budou vytvářeny na serveru ADS, v databázi s názvem MI_INFO, kde vznikne schéma s názvem mike, které bude sloužit pouze pro účely OLAP kostky.

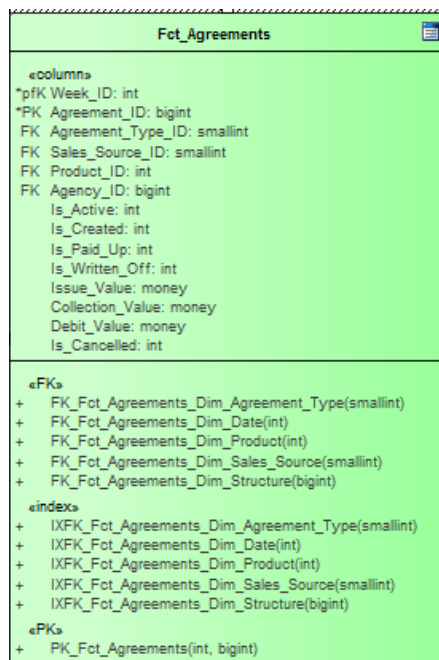
Metriky a pohledy na metriky budou rozděleny do faktových tabulek a dimenzí se snahou o zachování hvězdicového schématu. Datový model bude vytvořen v programu Enterprise Architect, včetně popisů jednotlivých tabulek a části tohoto datového modelu.

Dalším krokem bude zajištění naplnění dat do těchto tabulek.

5.1 Data o smlouvách

Nyní se podíváme na faktovou tabulku smluv, která je propojená s dalšími dimenzemi tak, abychom z takového modelu byli schopni čerpat potřebné informace dle zadání pro vytvoření OLAP kostky.

5.1.1 Popis faktové tabulky Fct_Agreements



Obr. 5 Model faktové tabulky smluv (Enterprise Architect, vlastní tvorba)

Datový model pro tuto část kostky obsahuje jednu faktovou tabulku, ve které se nacházejí jednotlivé smlouvy (půjčky). Smlouvy jsou zobrazovány v každém účetním týdnu tak, že

mohou nabývat v těchto týdnech různých stavů. Smlouva může být aktivní, tzn., byla vytvořena v minulosti (nebo v daném týdnu, na který se právě díváme), pak nabývá hodnoty `Is_Active = 1`. Další události smlouvy je její vytvoření, informace o vytvoření smlouvy je držena ve sloupci `Is_Created`, pokud je účetní týden, na který se právě díváme zároveň týdnem vytvoření smlouvy, pak tato hodnota `Is_Created = 1`. Následně může být smlouva ukončena třemi způsoby:

- řádným doplacením – k tomu slouží sloupec `Is_Paid_Up`, který v týdnu doplacení smlouvy nabývá hodnoty 1,
- odepsáním půjčky, k čemuž dochází v případě, kdy zákazník neplatí své závazky – pro tyto účely nám slouží sloupec `Is_Written_Off`, který v případě odepsání smlouvy nabývá hodnoty 1,
- posledním způsobem, jak lze smlouvu ukončit, je odstoupení od smlouvy do 14 dní – v tomto případě je využito sloupce `Is_Cancelled`, který v týdnu, kdy bylo podáno odstoupení od smlouvy, nabývá hodnoty 1.

Všechny tyto hodnoty jsou typu integer, aby byla zachována možnost tyto hodnoty sčítat v kostce. Pro potřeby zobrazení hodnoty vytvořených půjček slouží sloupec `Issue_Value`, který obsahuje částku, na kterou byla daná smlouva sjednána. Tento sloupec obsahuje hodnotu pouze v týdnu vytvoření smlouvy, pro zobrazení požadovaných dat ze zadání, tedy hodnoty prodejších. Další dva sloupce `Collection_Value` a `Debit_Value` slouží k podání informací o výběrech a očekávaných splátkách na smlouvách. Vzhledem k návrhu datového modelu, pokryjeme oblast týkající se smluv a jejich transakcí v jedné faktové tabulce, čímž se datový model značně zjednoduší. Poměr vybraných splátek oproti očekávaným splátkám bude vytvořen jako napočítané pole již přímo v OLAP kostce.

Kolem této faktové tabulky bylo potřeba také vytvořit 5 dimenzí, které jsou spojeny s faktovou tabulkou přes cizí klíče. Vždy existuje pouze jedna vazba mezi dimenzí a faktovou tabulkou. Tyto dimenze umožňují dívat se na data z různých pohledů.

Primárním klíčem tabulky je unikátní identifikátor každé půjčky a týden, na který se právě díváme. Ve faktové tabulce smluv se nachází celkem pět cizích klíčů, které odkazují do dimenzí v hvězdicovém schématu okolo této tabulky. Jsou drženy cizí klíče do každé z dimenzí:

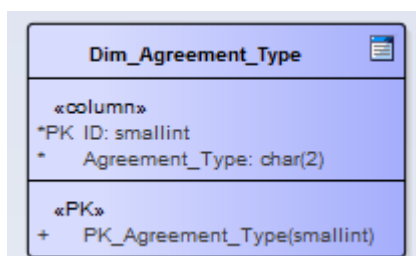
- Cizí klíč `Agreement_Type_ID`, který odkazuje do dimenze `Dim_Agreement_Type`, označuje typ poskytnuté půjčky (zda se jedná o půjčku novému či stávajícímu zákazníkovi).
- Cizí klíč `Sales_Source_ID`, odkazující do dimenze `Dim_Sales_Source`, poskytuje informaci, zda byla smlouva poskytnuta na základě leadu z call centra či affiliate partnera, nebo zda došlo k získání tohoto zákazníka činností obchodního zástupce.
- Cizí klíč `Product_ID` odkazuje do dimenze `Dim_Product` a podává informaci o produktu, jaký byl při poskytnutí této půjčky zvolený a odsouhlasený zákazníkem.
- Cizí klíč `Agency_ID` odkazuje do dimenze `Dim_Structure`. Jelikož je agentura nejnižší úrovní obchodní struktury, která je sledována, proto bude celá obchodní struktura získávána právě na základě tohoto klíče.

Poslední dimenzi, která je ve vztahu s faktovou tabulkou smluv je dimenze Dim_Date. Tato dimenze dává pohled na data v čase.

Výpis skriptu 2 - Vytvoření tabulky Fct_Agreements (vlastní tvorba)

```
CREATE TABLE mike.Fct_Agreements(
    Week_Id INT FOREIGN KEY REFERENCES mike.Dim_Date(Week_ID)
    ,Agreement_Id BIGINT
    ,Agreement_Type_ID SMALLINT FOREIGN KEY REFERENCES mike.Dim_Agreement_Type(ID)
    ,Sales_Source_ID SMALLINT FOREIGN KEY REFERENCES mike.Dim_Sales_Source(ID)
    ,Product_ID INT FOREIGN KEY REFERENCES mike.Dim_Product(ID)
    ,Agency_ID BIGINT FOREIGN KEY REFERENCES mike.Dim_Structure(Agency_ID)
    ,Is_Active INT
    ,Is_Created INT
    ,Is_Paid_Up INT
    ,Is_Written_Off INT
    ,Is_Cancelled INT
    ,Issue_Value MONEY
    ,Collection_Value MONEY
    ,Debit_Value MONEY
    PRIMARY KEY (Week_Id, Agreement_Id)
)
```

5.1.2 Popis dimenze Dim_Agreement_Type



Obr. 6 Model dimenze typu smlouvy (Enterprise Architect, vlastní tvorba)

Dimenze Dim_Agreement_Type obsahuje informaci o typu smlouvy. Rozlišují se smlouvy NC – pro nové zákazníky a smlouvy typu RS – pro stávající zákazníky. Současně tato dimenze obsahuje hodnotu s ID = 3, která slouží pro případy, kdy by smlouva z faktové tabulky neměla vyplněné Agreement_Type_ID.

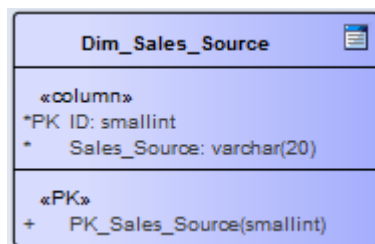
Tabulka 4 - Ukázka dat v Dim_Agreement_Type (vlastní tvorba)

ID	Agreement_Type
1	NC
2	RS
3	NA

Výpis skriptu 3 - Vytvoření tabulky Dim_Agreement_Type (vlastní tvorba)

```
CREATE TABLE mike.Dim_Agreement_Type (  
    ID SMALLINT PRIMARY KEY  
    ,Agreement_Type CHAR(2)  
)
```

5.1.3 Popis dimenze Dim_Sales_Source



Obr. 7 Model dimenze zdroje prodeje (Enterprise Architect, vlastní tvorba)

V dimenzi prodejního kanálu je rozlišován zdroj, kterým obchodní zástupce získal tuto smlouvu. Existují dva typy prodejních kanálů: Lead – smlouva byla vytvořena na základě aktivity společnosti a Own – kdy smlouvu získal obchodní zástupce vlastní iniciativou. Pokud u smlouvy není vyplněný prodejní kanál, pak tento číselník obsahuje také hodnotu „NA“, kam se odkazují právě tyto smlouvy.

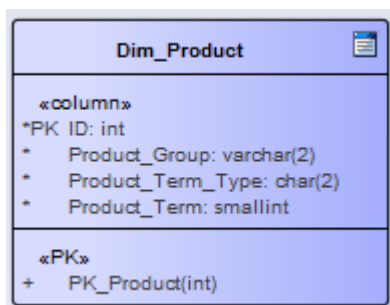
Tabulka 5 - Ukázka dat v Dim_Sales_Source (vlastní tvorba)

ID	Sales_Source
1	Lead
2	Own
3	NA

Výpis skriptu 4 - Vytvoření tabulky Dim_Sales_Source (vlastní tvorba)

```
CREATE TABLE mike.Dim_Sales_Source (  
    ID SMALLINT PRIMARY KEY  
    ,Sales_Source VARCHAR(20)  
)
```

5.1.4 Popis dimenze Dim_Product



Obr. 8 Model dimenze produktů (Enterprise Architect, vlastní tvorba)

Dimenze produktů poskytuje informaci o produktu, se kterým byla smlouva poskytnuta. Sloupec `Product_Group` je informace o produktové skupině, dle rizikovosti klienta (A až D, kdy A je nejméně rizikový klient). Dále v této dimenzi je informace o frekvenci splátek (rozlišuje se týdenní a měsíční splácení) a jejich počtu. Pokud by se v systému z nějakého důvodu dostala do systému smlouva, která nemá vyplněný identifikátor produktu, pak jsou takové smlouvy zařazeny do speciální skupiny „NA“.

Tento číselník produktů obsahuje i další data, pro ukázkou však postačí pouze pár řádků.

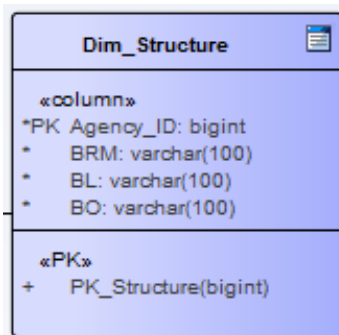
Tabulka 6 - Ukázka dat v Dim_Product (vlastní tvorba)

ID	Product_Group	Product_Term_Type	Product_Term
9999	NA	NA	0
623	A	MO	36
610	D	WE	100

Výpis skriptu 5 - Vytvoření tabulky Dim_Product (vlastní tvorba)

```
CREATE TABLE mike.Dim_Product (  
    ID INT PRIMARY KEY  
    ,Product_Group VARCHAR(2)  
    ,Product_Term_Type CHAR(2)  
    ,Product_Term SMALLINT  
)
```

5.1.5 Popis dimenze Dim_Structure



Obr. 9 Model dimenze obchodní struktury (Enterprise Architect, vlastní tvorba)

Předposlední dimenzí kolem faktové tabulky smluv je dimenze obchodní struktury. Tato dimenze umožňuje získat informaci o zařazení smluv a později také zákazníků do správných oblastí. Nejnižší úroveň je agentura, kterou spravuje právě jeden obchodní zástupce. Agentury spadají pod BRM – business relationship managera, který je podřízený BL – business leader. Nejvyšší úroveň jsou BO – business owners, kteří jsou pro celou zemi jen dva. V případě, že se objeví smlouva nebo zákazník, který není přiřazený v daném týdnu do žádné agentury, pak pro tyto případy opět existuje speciální skupina „NA“.

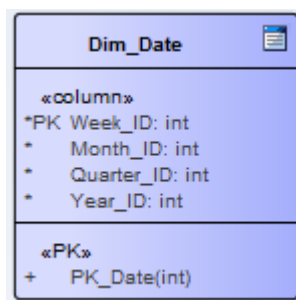
Tabulka 7 - Ukázka dat v Dim_Structure (vlastní tvorba)

Agency_ID	BRM	BL	BO
1023000001890	Brno BRM 31E	Brno BL 31	BO 2
1019000008946	Olomouc BRM 31G	Olomouc BL 31	BO 2
99999999	NA	NA	NA

Výpis skriptu 6 - Vytvoření tabulky Dim_Structure (vlastní tvorba)

```
CREATE TABLE mike.Dim_Structure (  
    Agency_ID BIGINT PRIMARY KEY  
    ,BRM VARCHAR(100)  
    ,BL VARCHAR(100)  
    ,BO VARCHAR(100)  
)
```

5.1.6 Popis dimenze Dim_Date



Obr. 10 Model časové dimenze (Enterprise Architect, vlastní tvorba)

Je žádoucí, aby všechna data bylo možné časově zařadit, k čemuž bude sloužit dimenze Dim_Date. Nejnížší časovou jednotkou je v případě Providentu účetní týden, který je specifický tím, že začíná ve středu a končí v úterý. Veškerá data jsou sledována právě na bázi tohoto speciálního účetního týdne. Měsíce, čtvrtletí i roky se také řídí pomocí těchto účetních týdnů.

Tabulka 8 - Ukázka dat v Dim_Date (vlastní tvorba)

Week_ID	Month_ID	Quarter_ID	Year_ID
201701	201701	201701	2017
201702	201701	201701	2017
201703	201701	201701	2017

Výpis skriptu 7 - Vytvoření tabulky Dim_Date (vlastní tvorba)

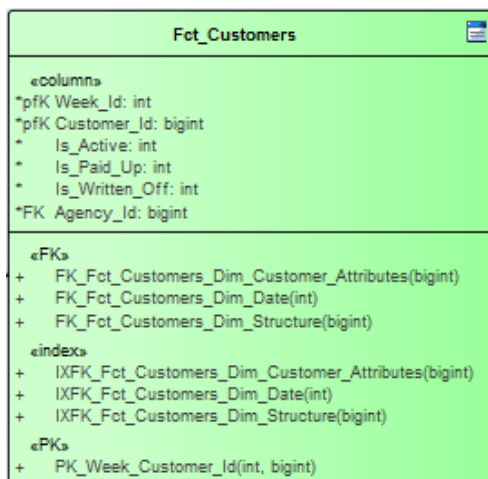
```
CREATE TABLE mike.Dim_Date (  
    Week_ID INT PRIMARY KEY  
    ,Month_ID INT  
    ,Quarter_ID INT  
    ,Year_ID INT  
)
```

5.2 Data o zákaznících

Pro sledování metrik spojených s počty zákazníků, je připravená faktová tabulka, která bude obsahovat týdenní záznamy jednotlivých zákazníků. Jelikož taková tabulka v tuto chvíli v systémové databázi neexistuje, musí být teprve vytvořena. Pro tyto účely se tedy vytvoří faktová tabulka s názvem Fct_Customers, kde lze vidět v každém týdnu informace o každém ze zákazníků. Současně je potřeba tuto faktovou tabulku propojit s dimenzemi tak, aby bylo možné sledovat data přes požadované pohledy dle zadání. Bude tedy vytvořena dimenze s názvem Dim_Customer_Attributes, kde budou uloženy informace o zákaznících. Následně

se využije již vytvořených dimenzí Dim_Date a Dim_Structure, které byly původně pro data o smlouvách, a napojí se k nově vytvořené faktové tabulce se zákazníky.

5.2.1 Popis faktové tabulky Fct_Customers



Obr. 11 Model faktové tabulky zákazníků (Enterprise Architect, vlastní tvorba)

Faktová tabulka Fct_Customers obsahuje 6 sloupců, z nichž dva jsou složeným primárním klíčem a zároveň cizími klíči do dimenzí Dim_Date a Dim_Customer_Attributes, který je tvořený sloupci Week_Id a Customer_Id. Sloupec Week_Id poskytuje informaci o účetním týdnu, ve kterém je na daného zákazníka pohlíženo, a sloupec Customer_Id je jednoznačný identifikátor zákazníka. Další sloupce ukazují, co se se zákazníkem v daném týdnu stalo. Pokud měl zákazník aktivní smlouvu, pak má ve sloupci Is_Active hodnotu 1. Jestliže zákazník v daném týdnu všechny své smlouvy doplatil nebo byl odepsán, pak nabývá hodnoty 1 ve sloupci Is_Paid_Up nebo Is_Written_Off a současně již není aktivním zákazníkem, tudíž sloupec Is_Active v daném týdnu nabývá hodnoty 0. Tyto hodnoty mají datový typ integer, aby bylo možné s těmito hodnotami počítat v kostce.

Tabulka 9 - Ukázka dat v Fct_Customers (vlastní tvorba)

Week_Id	Customer_Id	Agency_Id	Is_Active	Is_Paid_Up	Is_Written_Off
201701	6000001988	6000005573	1	0	0
201701	6000005232	6000001926	0	1	0
201701	1002000522282	1002000004180	0	0	1

Výpis skriptu 8 - Vytvoření tabulky Fct_Customers (vlastní tvorba)

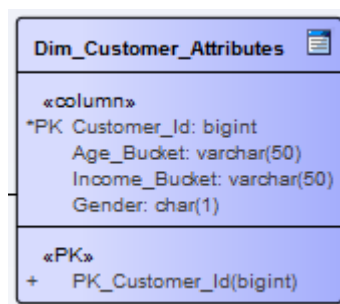
```
CREATE TABLE mike.Fct_Customers (
    Week_Id INT FOREIGN KEY REFERENCES mike.Dim_Date(Week_Id)
    ,Customer_Id BIGINT FOREIGN KEY REFERENCES
mike.Dim_Customer_Attributes(Customer_Id)
```

```

,Agency_Id BIGINT FOREIGN KEY REFERENCES mike.Dim_Structure(Agency_Id)
,Is_Active INT
,Is_Paid_Up INT
,Is_Written_Off INT
PRIMARY KEY (Week_Id, Customer_Id)
)

```

5.2.2 Popis dimenze Dim_Customer_Attributes



Obr. 12 Model dimenze vlastnosti zákazníků (Enterprise Architect, vlastní tvorba)

Pomocí této dimenze lze tvořit pohledy na data o zákaznících. Jedná se o rozdělení zákazníků podle věku, jejich příjmu a také pohlaví. Jelikož je zbytečné rozdělovat portfolio zákazníků podle každého věku, který databáze obsahuje, lze přistoupit k zařazení zákazníků do věkových skupin (18-30, 31-40, 41-50, atd.). Stejně tak lze rozdělit zákazníky podle jejich prokazatelného příjmu do příjmových skupin. Posledním sloupcem v této dimenzi je Gender, tedy pohlaví, který bude nabývat pouze hodnot M (male) a F (female).

Tabulka 10 - Ukázka dat v Dim_Customer Attributes

Customer_Id	Age_Bucket	Income_Bucket	Gender
6000001128	41-50	NA	M
6000001027	51-60	15001 - 20000	F
6000001375	51-60	> 50000	M

Výpis skriptu 9 - Vytvoření tabulky Dim_Customer_Attributes

```

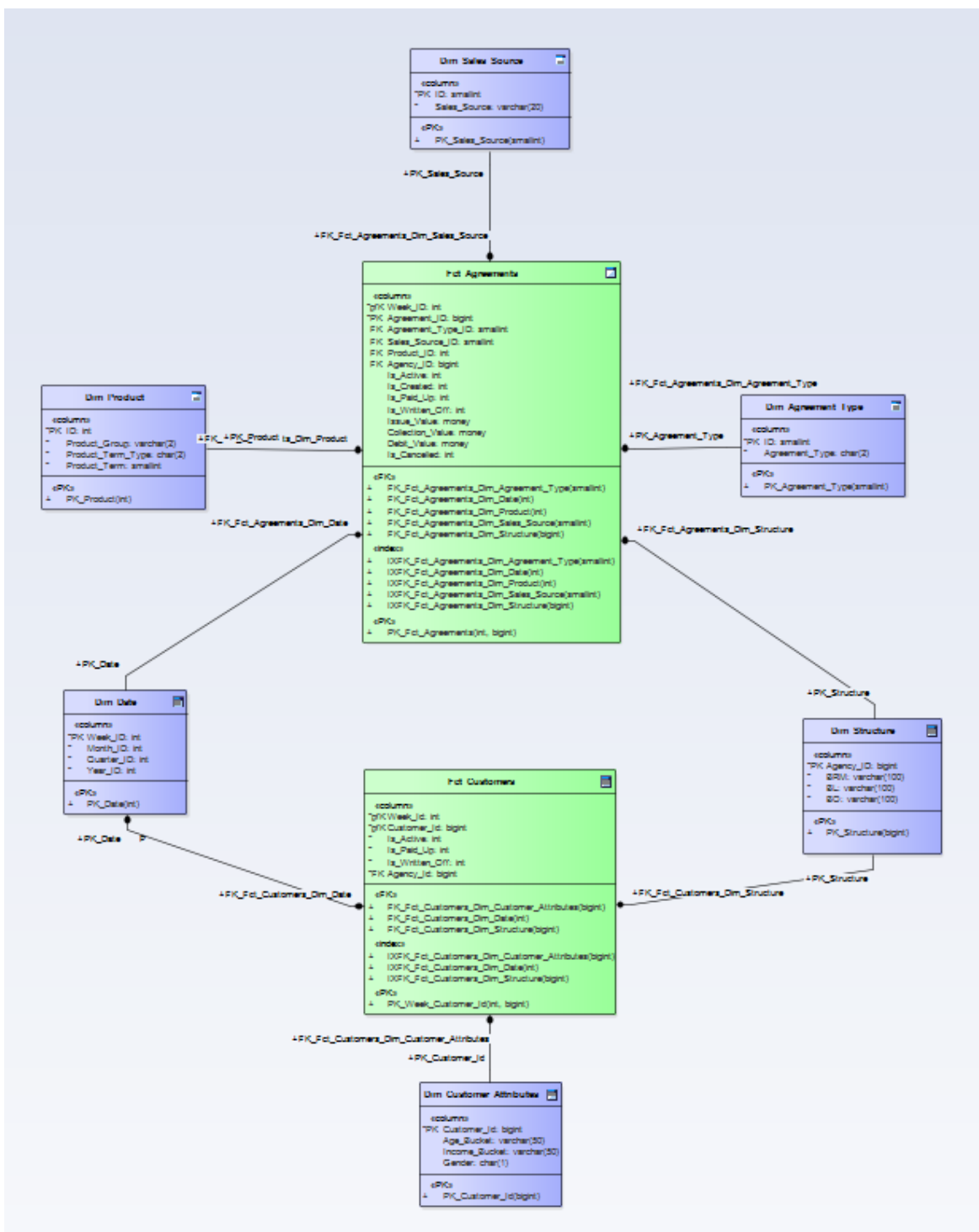
CREATE TABLE mike.Dim_Customer_Attributes (
    Customer_Id BIGINT PRIMARY KEY
    ,Age_Bucket VARCHAR(50)
    ,Income_Bucket VARCHAR(50)
    ,Gender CHAR(1)
)

```

5.3 Finální datový model v Enterprise Architect

Celý datový model se skládá ze dvou faktových tabulek. I přesto, že původním záměrem bylo rozdělení části dat o smlouvách a části o výběrech do dvou faktových tabulek, nakonec bylo zvoleno řešení, kdy došlo k vytvoření týdenních záznamů (snapshotů) každé smlouvy a k nim pak doplnění informací, zda byla v tom daném týdnu očekávána splátka a kolik bylo skutečně na dané smlouvě vybráno peněz.

Model je tedy vytvořený v hvězdicovém schématu (lze říci schématu souhvězdí, kdy faktové tabulky spolu sdílejí některé dimenze), pro snazší práci s tímto modelem při vytváření kostky. Samozřejmě se také nabízelo řešení snow flake schématu, které bylo nakonec zavrženo. V takovém schématu by pak z faktové tabulky Fct_Agreements byla vazba do dimenze s názvem například Dim_Agreement_Attributes, kde by byla informace (identifikátor) o produktu, prodejním kanálu a typu smlouvy, a následně až z této dimenze by bylo možné se odkázat do jednotlivých číselníků.

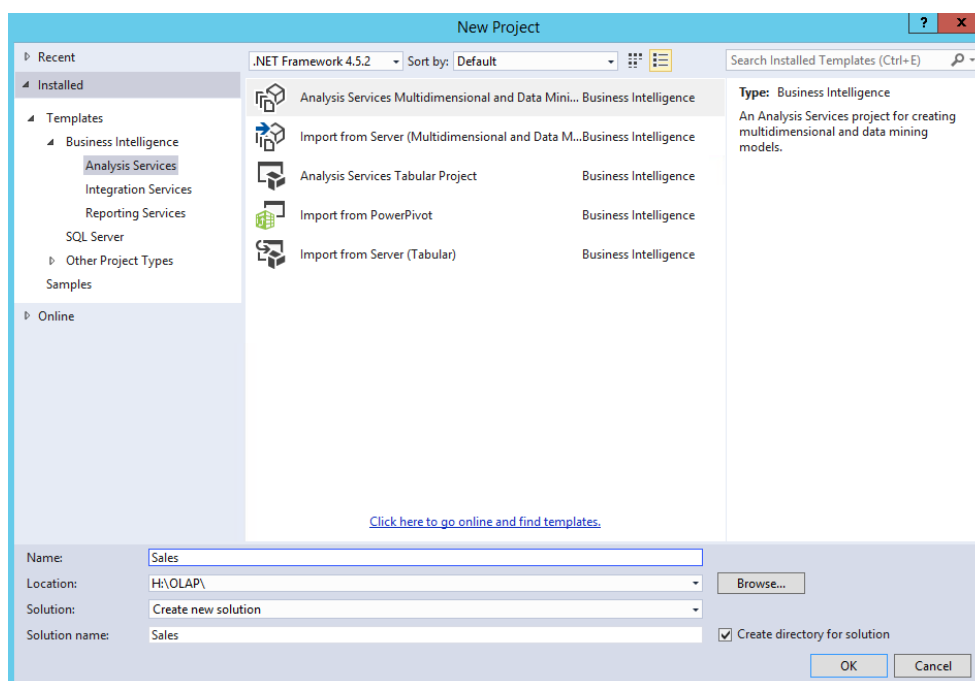


Obr. 13 Celkový datový model vytvořený pro účely OLAP (Enterprise Architect, vlastní tvorba)

6 Tvorba OLAP kostky

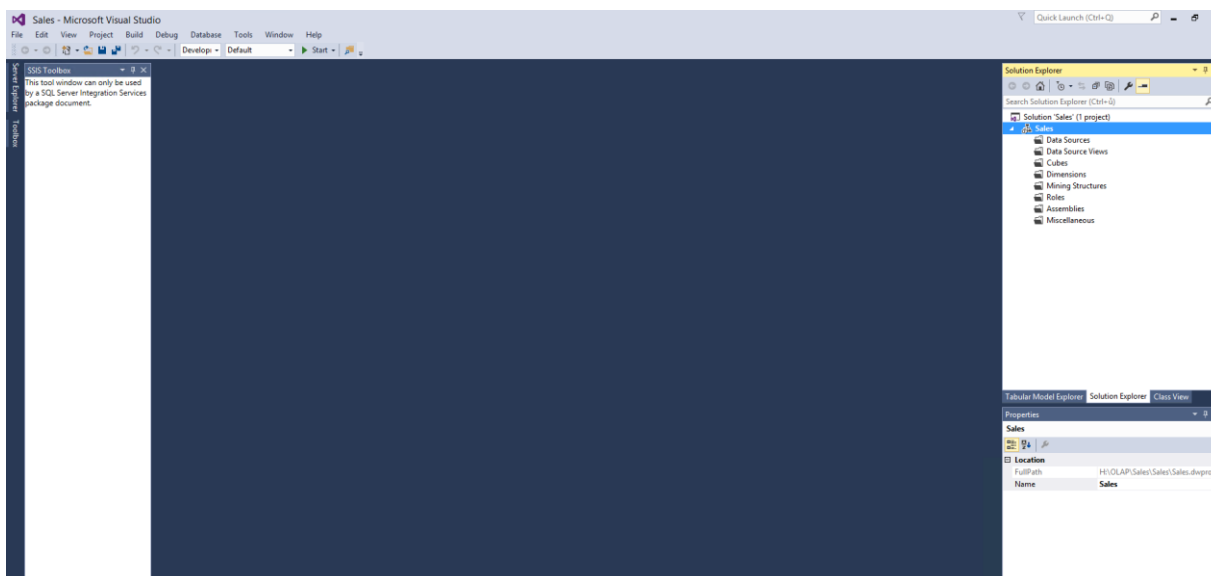
6.1 Vytvoření projektu ve Visual Studio

Prvním krokem k samotnému vytváření OLAP kostky je založení projektu ve Visual Studio. Po otevření Visual Studia klikneme na File, následně New a poté zvolíme Project. Poté si musíme zvolit, o jaký projekt se nám jedná. V tomto případě vybereme z nabídky Business Intelligence, Analysis Services a zde volíme Analysis Services Multidimensional and Data Mining. Projekt bude pojmenován jako Sales, dojde k výběru umístění a následnému potvrzení vytvoření projektu stiskem tlačítka OK.



Obr. 14 Zakládání projektu ve Visual Studio (vlastní snímek obrazovky)

Po úspěšném vytvoření projektu se dostaneme do pracovního prostředí Visual Studia. V tuto chvíli je nejdůležitější částí pracovního okna Solution Explorer vpravo nahoře, kde budeme postupně připojovat datový zdroj, vytvářet datové pohledy (model pro kostku), poté vytvoříme také kostku a budeme řešit nastavení dimenzí.



Obr. 15 Ukázka pracovního prostředí Visual Studio (vlastní snímek obrazovky)

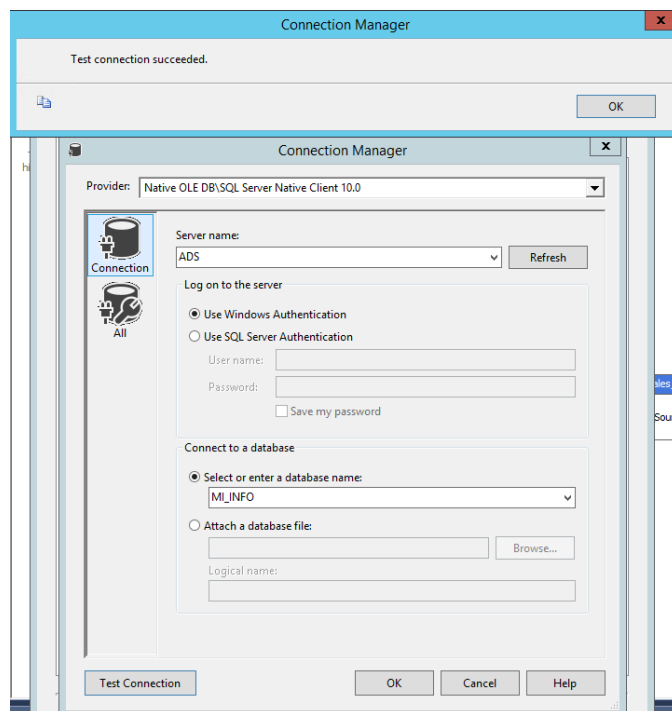
6.2 Definování datového zdroje

V dalším kroku dochází k definování připojení datového zdroje, který budeme používat pro kostku. Klikneme tedy pravým tlačítkem myši na Data Sources a zvolíme New Data Source a otevře se nám průvodce vytvářením datového zdroje.

Vyplníme výše zmíněný název serveru ADS, zvolíme autentizaci Windows a vybereme z nabídky databází MI_INFO. Důležité je změnit volbu připojení (Provider) na SQL Server Native Client 10.0, aby nám při procesování kostky na náš server vše proběhlo v pořádku. Následně ověříme funkčnost datového zdroje tlačítkem Test Connection. Test proběhl úspěšně, což znamená, že můžeme tento datový zdroj potvrdit. Vytvoří se nám následující connection string:

Výpis skriptu 10 - Connection string pro připojení k datovému zdroji (vygenerováno přes průvodce připojením ve Visual Studio)

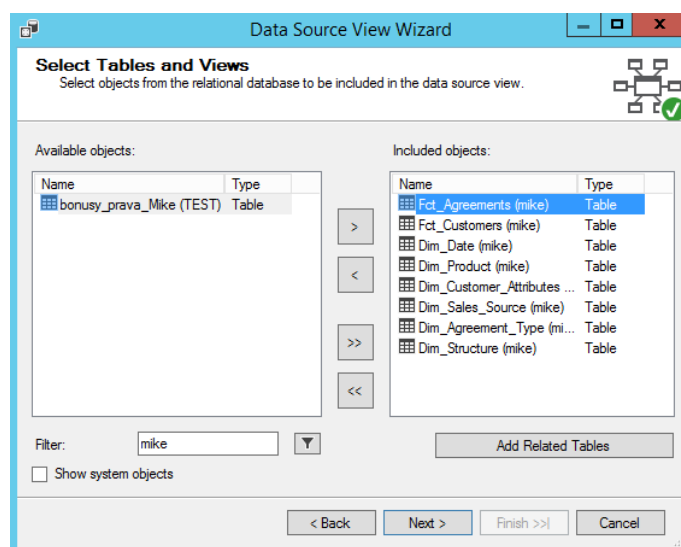
```
Provider=SQLNCLI10.1;Data Source=ADS;Integrated Security=SSPI;Initial Catalog=MI_INFO
```



Obr. 16 Nastavení připojení k datovému zdroji (vlastní snímek obrazovky)

6.3 Nastavení Data Source View

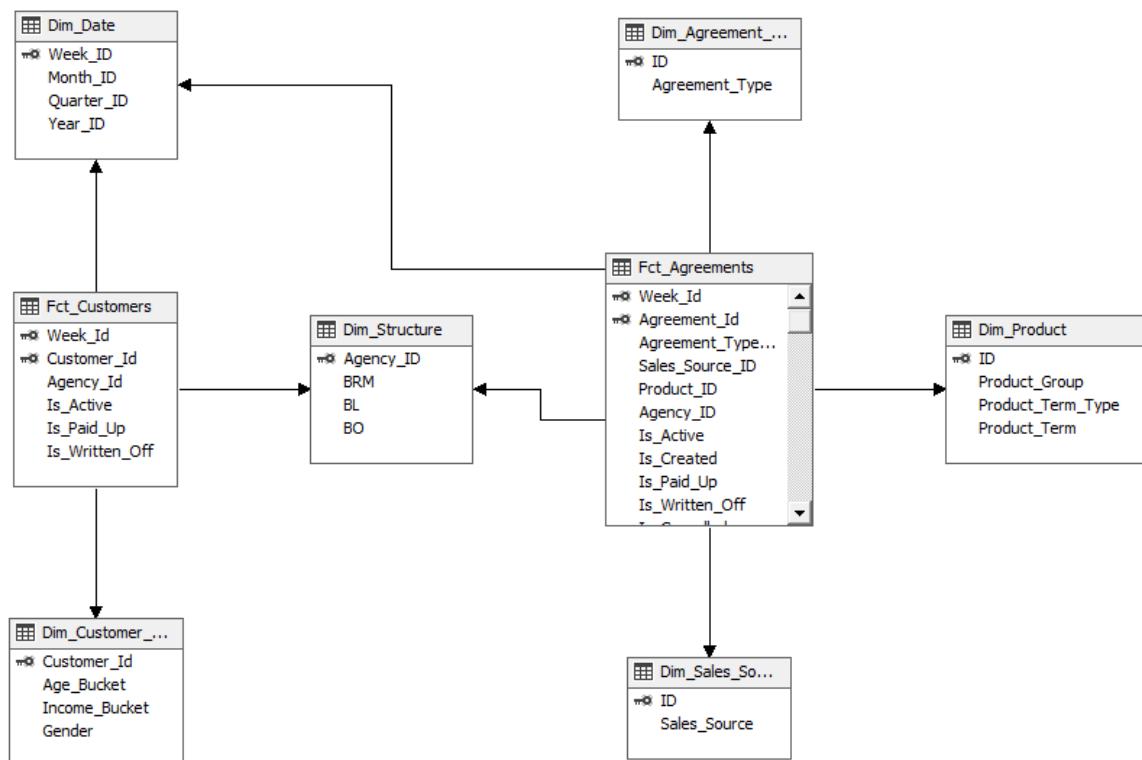
Data Source View je vlastně logický datový model vybraných tabulek z datového zdroje. Pomocí filtru můžeme omezit zobrazované tabulky. Vyfiltrujeme si tedy schéma mike a následně je potřeba přesunout všechny tabulky, které budeme používat v kostce, do pravého sloupce.



Obr. 17 Vytváření Data Source View přes průvodce (vlastní snímek obrazovky)

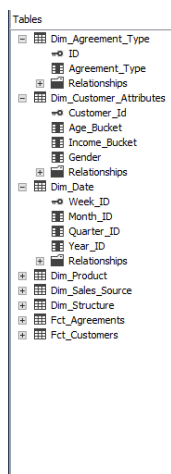
Poté, co dokončíme průvodce vytvořením Data Source View, se nám vygeneruje datový model námi zvolených objektů. V případě, že máme na úrovni databáze správně nastavené

primární i cizí klíče, pak se nám zároveň automaticky vygenerují také vazby mezi jednotlivými tabulkami. Je potřeba ale definovat nejen primární klíče, ale i cizí klíče, stejně tak, jako to bylo provedeno při vytváření tabulek pro kostku (viz. skripty CREATE TABLE výše).



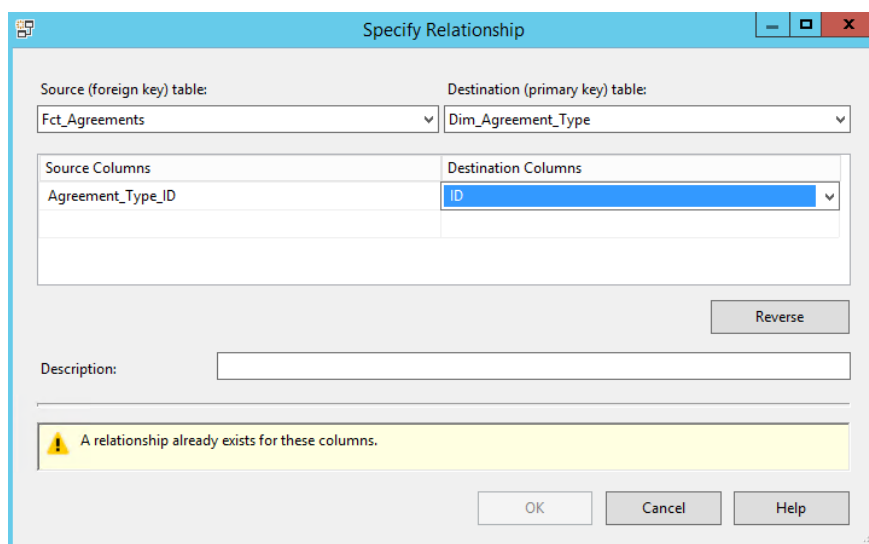
Obr. 18 Datový model Data Source View (vygenerováno ve Visual Studio)

Tabulky můžeme do Data Source View také přidávat dodatečně, případně lze některé špatně zvolené/nepotřebné tabulky z tohoto modelu odstranit. To vše lze provádět v okně v levé části s názvem Tables.



Obr. 19 Ukázka okna se seznamem tabulek v Data Source View (vlastní snímek obrazovky)

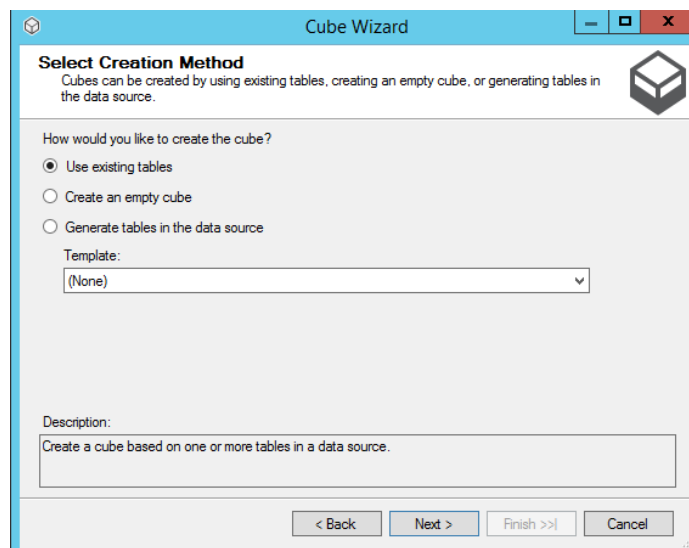
V případě, že bychom tyto klíče neměli definované přímo v databázi, lze vytvořit vazby mezi tabulkami přímo v Data Source View. Když stiskneme pravé tlačítko myši kdekoli v prázdném prostoru datového modelu a zvolíme volbu New Relationship, pak se dostaneme do nabídky Specify Relationship, kde lze vazbu mezi tabulkami definovat.



Obr. 20 Definování nové vazby v Data Source View (vlastní snímek obrazovky)

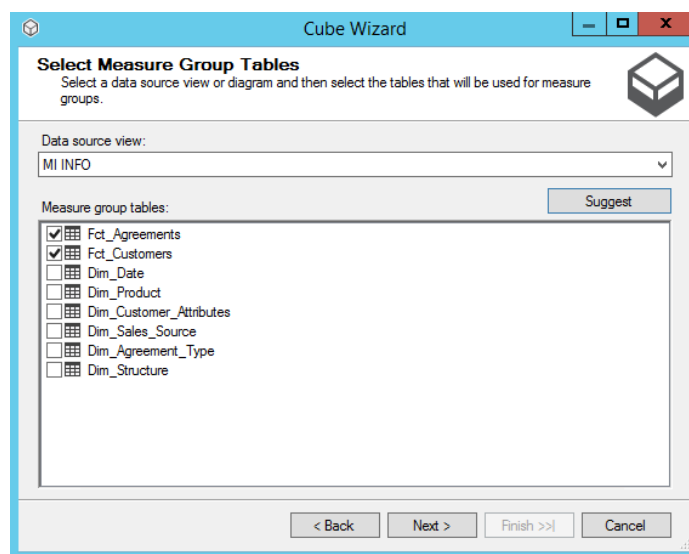
6.4 Vytvoření kostky

Dalším krokem je již samotné vytvoření OLAP kostky. Pro spuštění průvodce vytvářením kostky musíme kliknout pravým tlačítkem myši na Cubes v Solution Exploreru a vybereme New Cube. Otevře se nám průvodce, který se nás ptá, z čeho chceme kostku vytvořit. V našem případě vyberme volbu Use existing tables, jelikož chceme kostku vytvořit z tabulek, které jsme přidali do Data Source View.



Obr. 21 Průvodce vytvářením kostky - volba tabulek (vlastní snímek obrazovky)

V dalším kroku si musíme zvolit tabulky, které obsahují metriky. V našem případě tak volíme Fct_Agreements a Fct_Customers. Můžeme si také nechat doporučit, které tabulky by bylo vhodné zvolit jako obsahující metriky pomocí tlačítka Suggest.



Obr. 22 Průvodce vytvářením kostky - volba tabulek obsahujících metriky (vlastní snímek obrazovky)

Následujícím krokem při tvorbě kostky je volba jednotlivých metrik z tabulek, které jsme si vybrali v předchozím kroku. Zároveň můžeme tyto skupiny metrik a metriky přejmenovat do uživatelsky přívětivějšího jazyka. V případě naší obchodní kostky necháme zaškrtnuté metriky, které jsme byly připraveny v datovém modelu. Ve skupině metrik Agreements se jedná o:

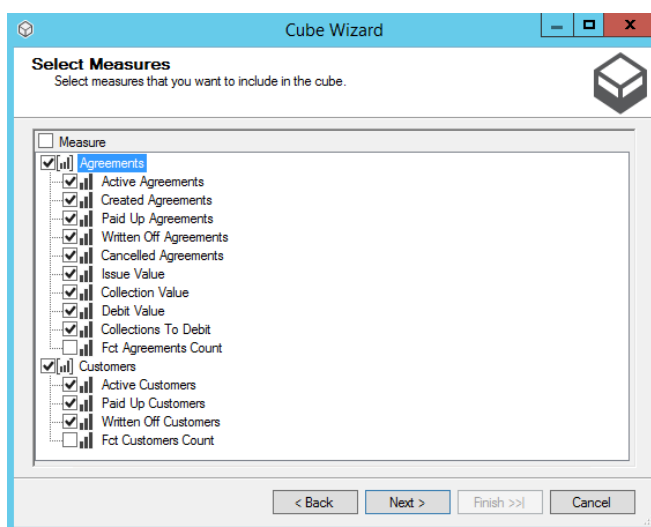
- počet aktivních smluv,
- počet vytvořených smluv,
- počet doplacených smluv,
- počet odepsaných smluv,

- počet zrušených smluv,
- hodnotu vytvořených půjček,
- hodnotu výběrů,
- hodnotu očekávaných splátek,
- poměr výběrů a očekávaných splátek.

Další skupinou metrik jsou Customers, kde budeme, dle zadání, sledovat následující metriky:

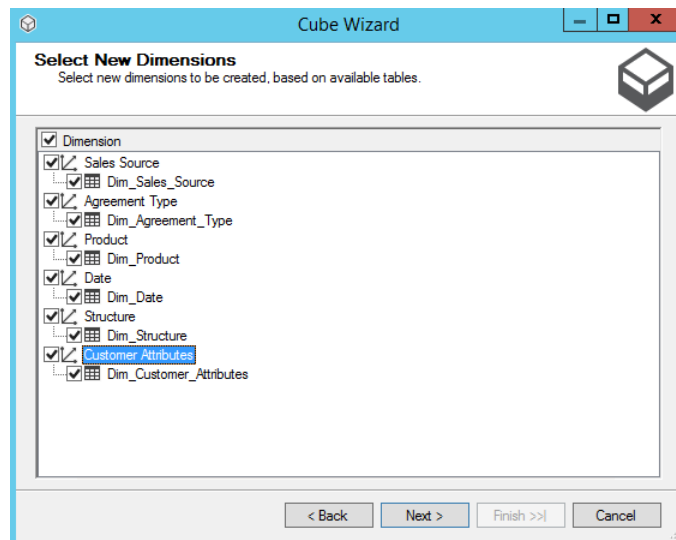
- počet aktivních zákazníků,
- počet doplacených zákazníků,
- počet odepsaných zákazníků.

Nezvolíme pouze celkové počty v jednotlivých skupinách metrik, jelikož tyto údaje nejsou v zadání požadovány.



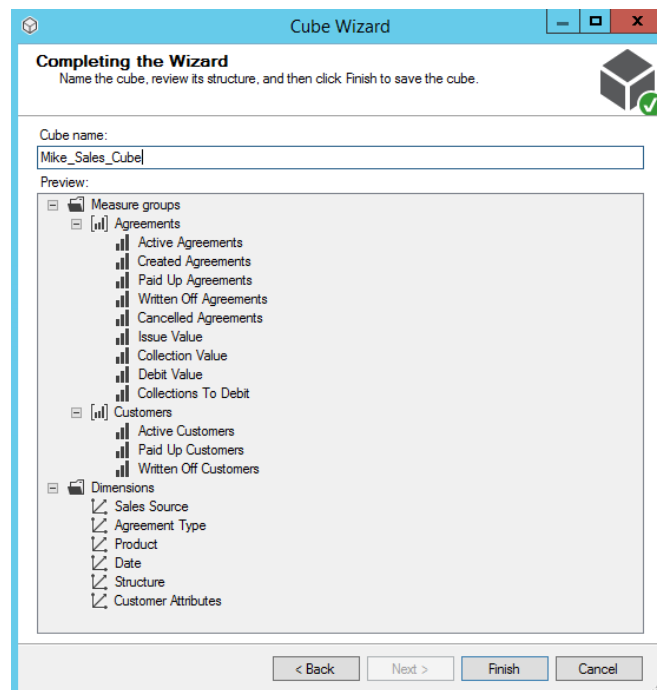
Obr. 23 Průvodce vytvářením kostky - volba jednotlivých metrik z tabulek (vlastní snímek obrazovky)

Posledním krokem k vytvoření OLAP kostky je výběr dimenzí, které chceme v kostce používat. Zvolíme tedy všechny dimenze a přejmenujeme je opět na uživatelsky příjemnější název.



Obr. 24 Průvodce vytvářením kostky - výběr dimenzí (vlastní snímek obrazovky)

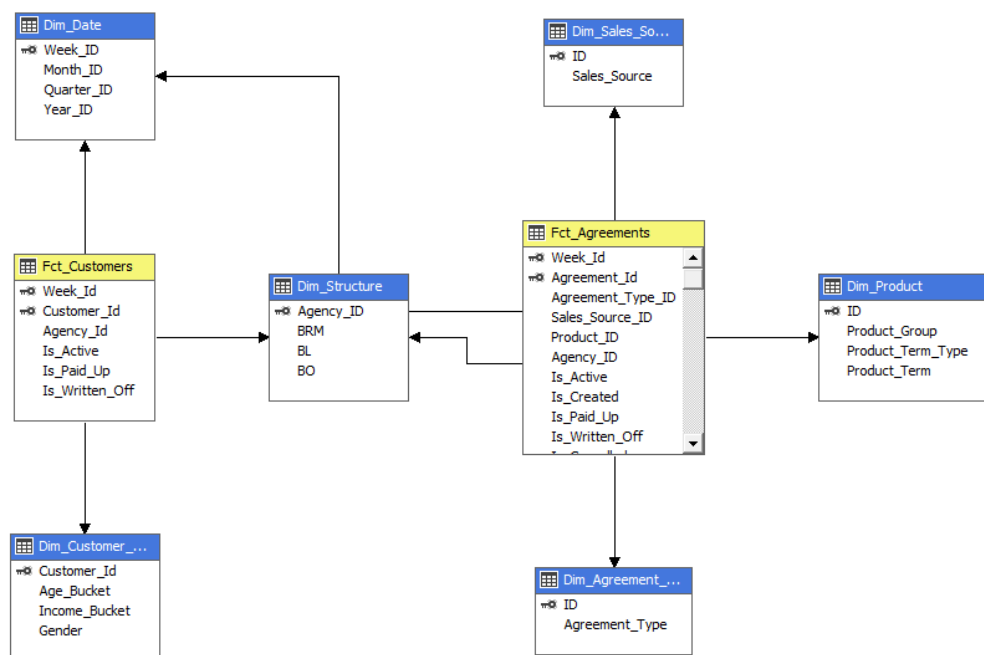
Ted' už nám zbývá pouze kostku pojmenovat (Mike_Sales_Cube) a potvrdit její vytvoření.



Obr. 25 Průvodce vytvářením kostky - pojmenování kostky a shrnutí (vlastní snímek obrazovky)

6.4.1 Datový model OLAP kostky

Jakmile dojde k vytvoření OLAP kostky vygeneruje se zároveň datový model kostky, který rozlišuje faktové tabulky – žlutě a dimenze – modře. V našem případě je to vlastně totožný model s Data Source View modelem, jelikož jsme si zvolili již na počátku tabulky, které chceme využít pro naši kostku. Pokud tvoříme více kostek nad jedním Data Source View, pak se tyto modely mohou od sebe lišit.



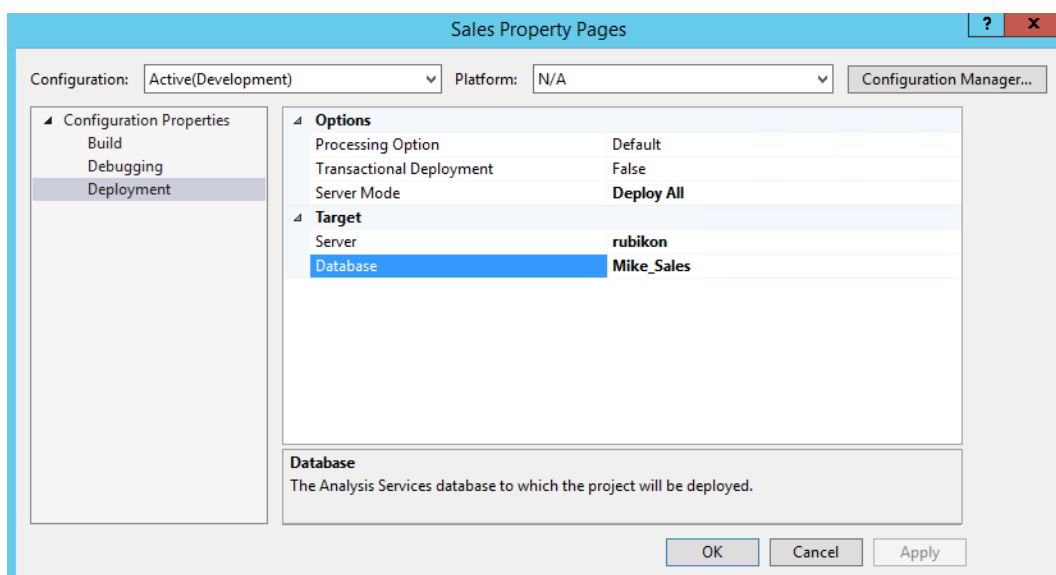
Obr. 26 Vygenerovaný datový model OLAP kostky (vygenerováno při vytváření kostky ve Visual Studio)

6.4.2 Ověření funkčnosti kostky

Poté, co byla OLAP kostka úspěšně vytvořena, je potřeba vyzkoušet, zda je kostka funkční. A teprve následně bude docházet k ladění kostky a provádění dalších nastavení. K tomu, aby bylo možné kostku vyzkoušet, je potřeba nasadit kostku na server. V tomto případě se jedná o Analysis Services Server s názvem rubikon.

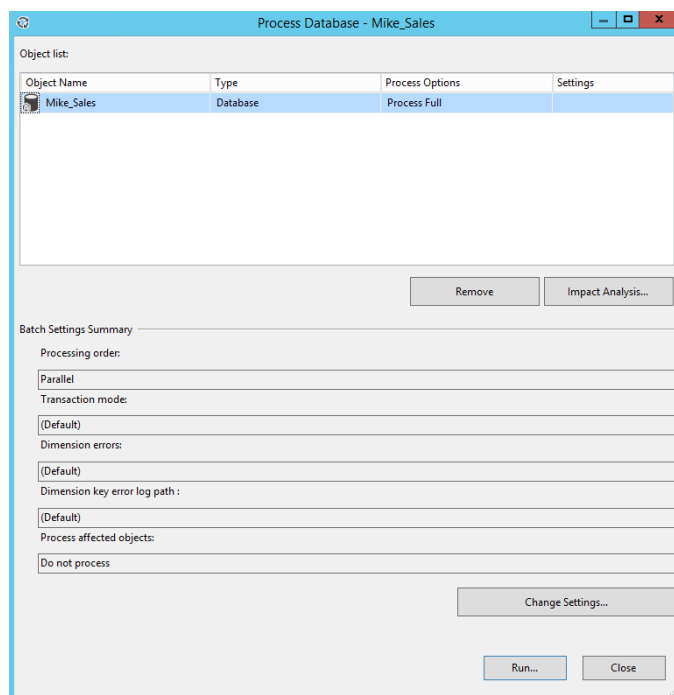
6.4.3 Nasazení kostky na Analysis Services Server

Definice serveru a databáze je prováděna na úrovni celého projektu. Kliknutím pravým tlačítkem myši na název projektu se nám zobrazí nabídka, kde zvolíme volbu Properties. Následně se nám otevře okno s konfigurací projektu. Na záložce Deployment musíme nastavit název našeho serveru: rubikon a zvolit si název databáze, ve které bude projekt uložený na serveru. Vzhledem k prvnímu nasazení kostky, což znamená, že jsme provedli změny ve struktuře dat, musím zvolit Server Mode: Deploy All, abychom docílili úspěšného nasazení kostky na server.



Obr. 27 Nastavení umístění pro nasazení kostky (vlastní snímek obrazovky)

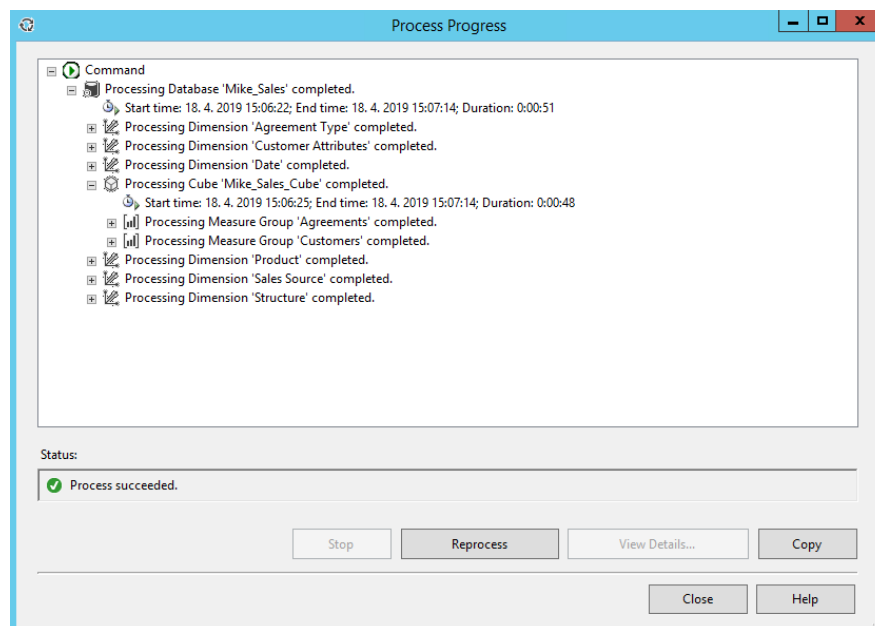
Nyní se dostáváme ke kroku, kdy musíme kostku zpracovat a nasadit na server. K tomu nám stačí kliknout pravým tlačítkem myši na projekt, a zvolit Process. Otevře se nám okno, kde můžeme nastavit jak se má procesování provádět a jak se má chovat pokud se vyskytnou při procesování nějaké chyby (lze některé chyby ignorovat a i přesto docílit úspěšného procesování kostky). Vše ponecháme v původním nastavení a spustíme tlačítkem Run.



Obr. 28 Nastavení procesování kostky (vlastní snímek obrazovky)

Jakmile spustíme procesování, zobrazí se nám okno postupu procesování, kde můžeme sledovat, která část je právě procesována. V našem případě celé procesování trvalo 51 sekund a proběhlo úspěšně, bez chyby a bez jakéhokoliv varování. Často se při procesování

kostky objevují varování, která neovlivní kvalitu dat, ale lze data připravit tak, aby tato varování při procesování vůbec nevznikala.



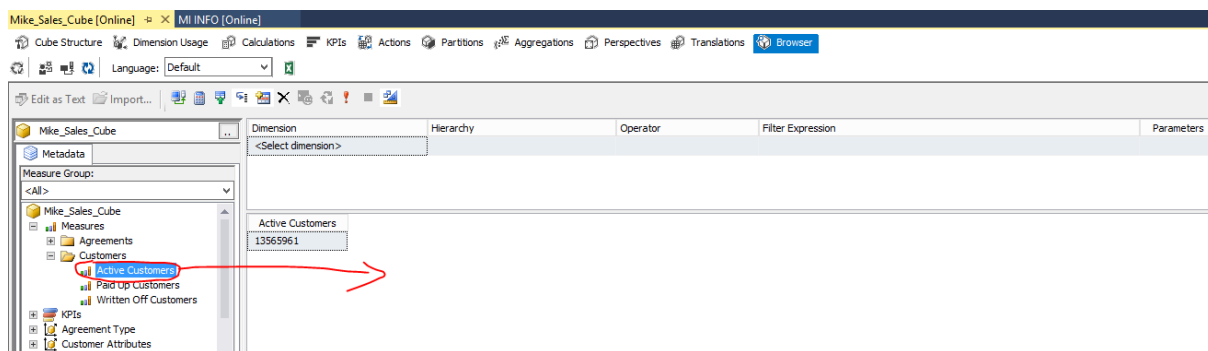
Obr. 29 Okno průběhu procesování kostky (vlastní snímek obrazovky)

6.5 První pohled na data z kostky

Po úspěšném nasazení kostky na server se nyní můžeme podívat, zda nám kostka také ukazuje nějaké hodnoty. Tady se nabízejí tři možnosti, jak rychle ověřit, zda jsou v kostce data. Můžeme také jednoduše ověřit jejich správnost.

6.5.1 Ověření dat v kostce přes Visual Studio

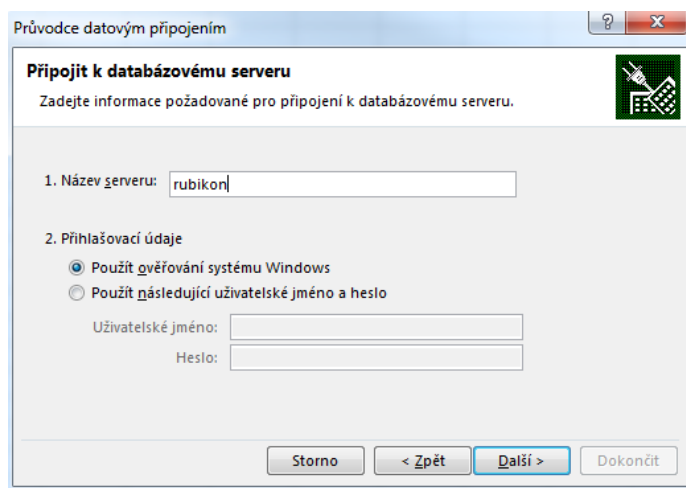
Nejsnazší volbou, jak ověřit funkčnost kostky, je spuštění prohlížeče (Browser) přímo ve Visual Studiu. Když si rozklikneme kostku a vybereme záložku Browser, tak nám postačí přetáhnout si jednu z metrik do volného místa. Pokud se objeví číslo, pak kostka pracuje a obsahuje data.



Obr. 30 Prohlížení dat v kostce ve Visual Studio (vlastní snímek obrazovky)

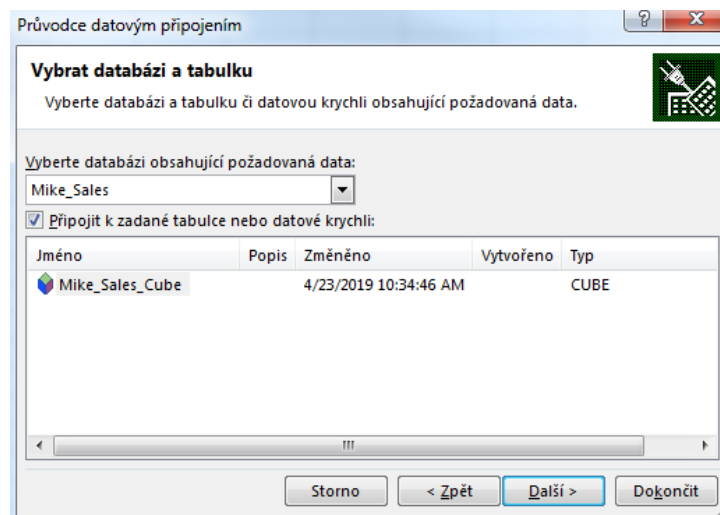
6.5.2 Ověření dat v kostce přes Excel

Další možností, jak otestovat funkčnost kostky, je vytvořit připojení k této kostce přes Microsoft Excel. Po otevření nového sešitu v Excelu klikneme na záložku Data a v sekci Načíst externí data zvolíme možnost Z jiných zdrojů (Ze služby Analysis Services). Poté vyplníme název serveru rubikon, ověření ponecháme přes Windows, a pokračujeme dále stisknutím tlačítka Další.



Obr. 31 Vytváření připojení ke kostce přes Excel - volba serveru (vlastní snímek obrazovky)

V dalším okně musíme v seznamu databází nalézt tu naši – Mike_Sales a následně zvolíme kostku Mike_Sales_Cube a potvrdíme tlačítkem Dokončit.



Obr. 32 Vytváření připojení ke kostce přes Excel - volba databáze a kostky (vlastní snímek obrazovky)

Kostku otevřeme jako kontingenční tabulku.

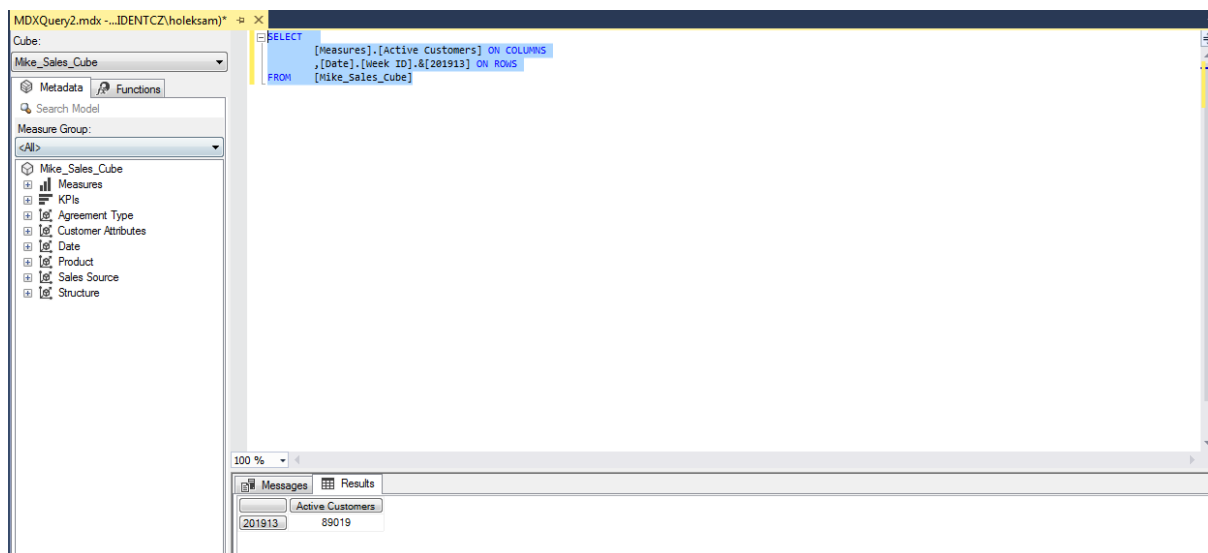


Obr. 33 Ukázka kostky jako kontingenční tabulky v Excelu (vlastní snímek obrazovky)

Pro ověření funkčnosti nám postačí zvolit si jednu z metrik a přetáhnout tuto metriku do hodnot. Abychom ověřili zároveň i správnost dat, můžeme si zvolit filtr podle týdnů, abychom číslo počtu zákazníků mohli porovnat se skutečně reportovanými čísly. Vše vypadá v pořádku.

6.5.3 Ověření dat v kostce přes výraz MDX

Poslední variantou, jak ověřit, zda nám kostka funguje či nikoliv, je napsání jednoduchého MDX dotazu. Opět použijeme naši metriku pro počet aktivních zákazníků, který chceme zobrazit ve sloupci. Pro ověření správnosti dat zvolíme opět aktuální účetní týden z dimenze Date, který zobrazíme na řádcích. Pokud nám tento dotaz proběhne v pořádku, pak je kostka funkční.



Obr. 34 Jednoduchý dotaz v jazyce MDX pro ověření funkčnosti kostky v SSMS (vlastní snímek obrazovky)

6.6 Tvorba hierarchie – časová dimenze

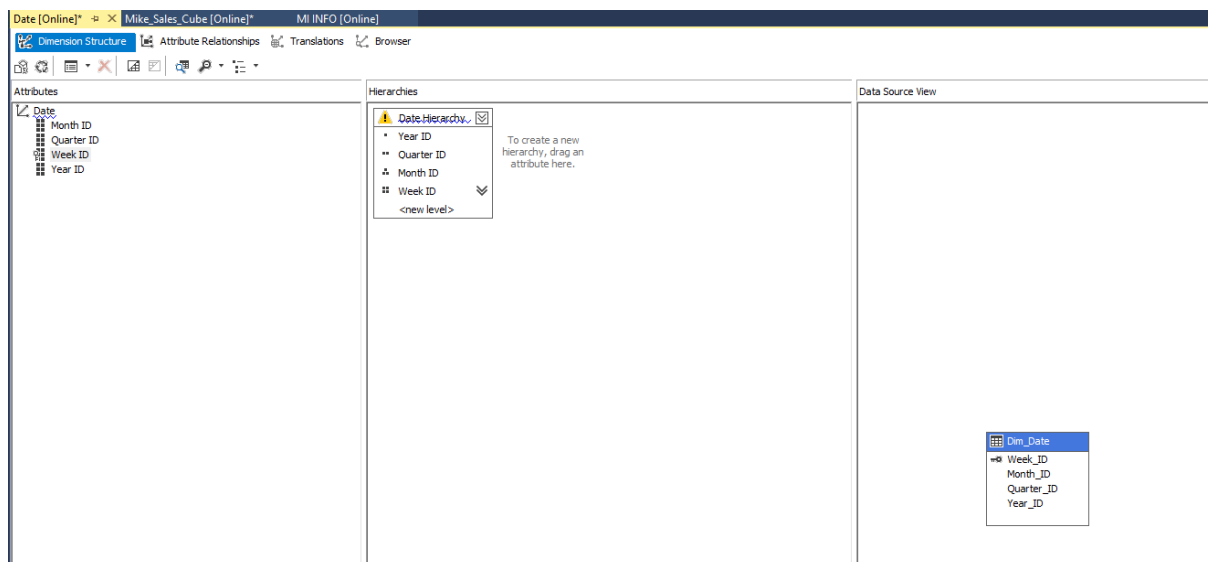
Pro správné zobrazování a časové zařazení všech metrik, je potřeba definovat nad dimenzí Date časovou hierarchii. Musíme kostce říct, že rok se skládá nejdříve ze čtvrtletí, následně měsíců a jako nejnižší úroveň v našem případě použijeme účetní týden. K těmto účelům nám slouží právě definování hierarchie.

Ze všeho nejdříve musíme do atributů dimenze vybrat všechny potřebné sloupce z dimenze Dim_Date, jelikož výchozí atribut v každé dimenzi je pouze primární klíč. Proto musíme z pravé části obrazovky (Data Source View) přetáhnout všechny sloupce, které chceme z naší dimenze používat, do levé části obrazovky (Attributes). V případě této časové dimenze tak přetáhneme sloupce Month_ID, Quarter_ID i Year_ID.

Dalším krokem k vytvoření hierarchie je přetažení atributů do prostřední části obrazovky Hierarchies ve správném pořadí. Nejvýše postavený musí být tedy rok, následně čtvrtletí, měsíc a nakonec účetní týden.

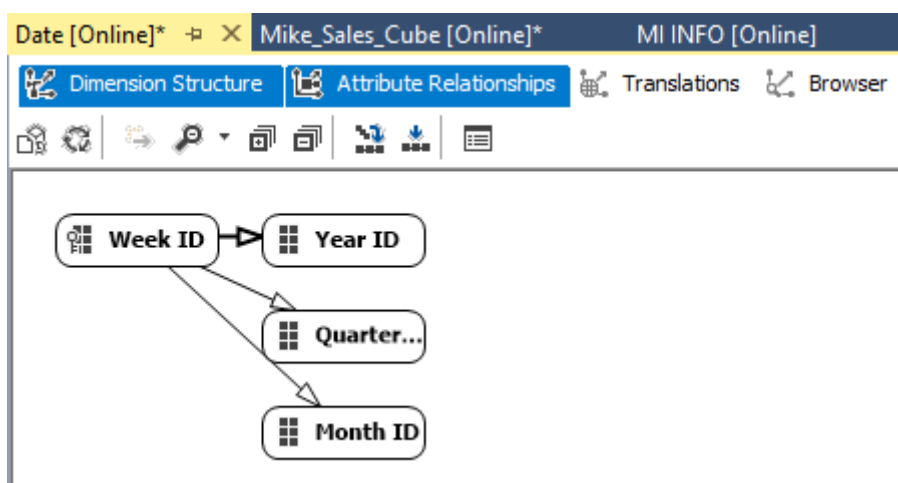
Vytvořenou hierarchii si můžeme také pojmenovat vlastním názvem (Date Hierarchy).

V tuto chvíli se nám zobrazuje u vytvořené hierarchie upozornění (varování), které odstraníme v následujícím kroku. Jedná se o to, že doposud nemáme správně definované pořadí, dle kterých se má tato hierarchie řídit. To lze nastavit na záložce Attribute Relationships.



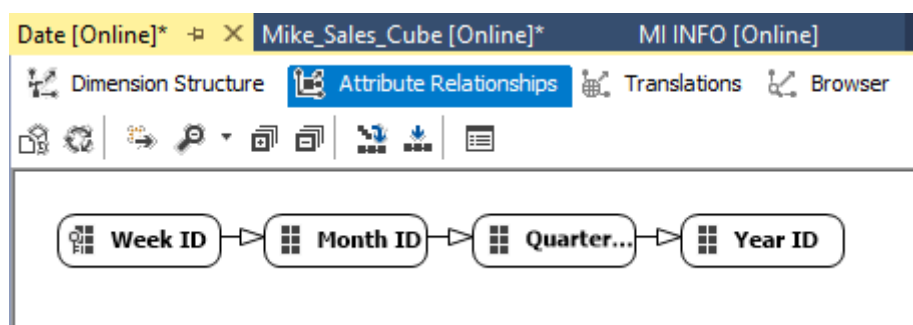
Obr. 35 Vytváření hierarchie v časové dimenzi (vlastní snímek obrazovky)

Jak můžeme vidět, tak vygenerované vztahy mezi atributy nedávají smysl, jelikož nevytváří smysluplnou hierarchii.



Obr. 36 Vygenerované vztahy v hierarchii (vlastní snímek obrazovky)

Přetažením šipek tedy vazby upravíme tak, aby tvořily námi požadovanou hierarchii.



Obr. 37 Správně nastavené vztahy v hierarchii (vlastní snímek obrazovky)

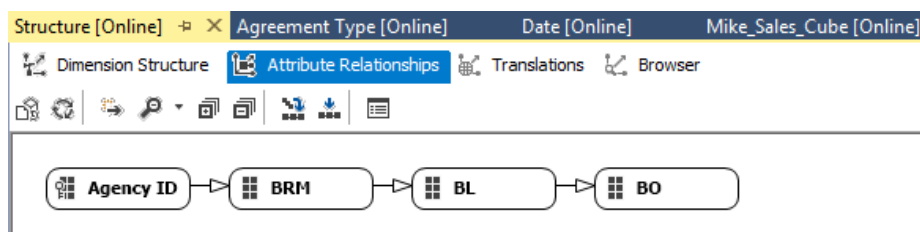
Nyní se podíváme, jak se tato změna projevila, když ke kostce přistoupíme přes Excel. Pod skupinou pohledů (dimenzí) Date, nám přibyla naše hierarchie. Pokud ji vložíme například do řádků, pak vidíme data na úrovni jednotlivých let, ale máme možnost operace drill-down v hierarchii až na úroveň účetních týdnů.



Obr. 38 Zobrazení hierarchie při výběru v kontingenční tabulce – Excel (vlastní snímek obrazovky)

6.7 Tvorba hierarchie – obchodní struktura

Stejně jako časovou hierarchii, musíme vytvořit také hierarchii pro obchodní strukturu. Ta má také určité členění a na jejím vrcholu je pozice BO a následně BL. Každé BL má pod sebou ještě několik oblastních manažerů zvaných BRM. Nejnižší úrovní jsou pak agentury. Tuto hierarchii tedy nastavíme dle úrovní. Postup je totožný s tvorbou časové hierarchie, proto jej nebudu více rozebírat.



Obr. 39 Správně nastavené vztahy v hierarchii pro obchodní strukturu (vlastní snímek obrazovky)

Podíváme se opět na vytvořenou hierarchii do naší kostky přes Excel. Můžeme vidět, že naše hierarchie funguje správně a je možné se pomocí operace drill-down dostat postupně přes BO, BL a BRM až na úroveň jednotlivých agentur.

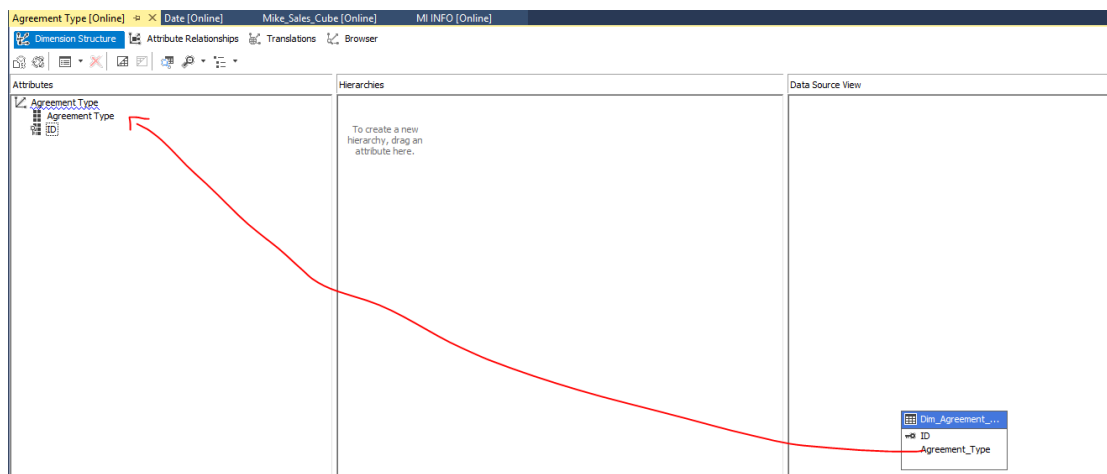
Week ID	
Popisky řádků	Created Agreements
BO 1	1046
Česká Lípa BL 21	16
České Budějovice BL 11	102
Jablonec N. Nisou BL 21	131
Karlovy Vary BL 11	18
Kladno BL 11	128
Kladno BRM 11A	30
1004000001244	3
1004000001338	1
1004000001617	6
1004000001631	7
1004000002577	1
1004000003312	4
1004000003444	2
1004000003855	2
1004000004366	3
1004000004498	1
Kladno BRM 11B	16
Kladno BRM 11C	11
Kladno BRM 11D	28
Kladno BRM 11E	19
Kladno BRM 11F	24
Most BL 11	118
Plzeň BL 11	165
Praha Jih BL 11	106
Praha Sever BL 21	164
Ústí Nad Labem BL 11	98
BO 2	1026
Celkový součet	2072

Obr. 40 Ukázka hierarchie obchodní struktury v kontingenční tabulce (vlastní snímek obrazovky)

6.8 Úprava dimenzí

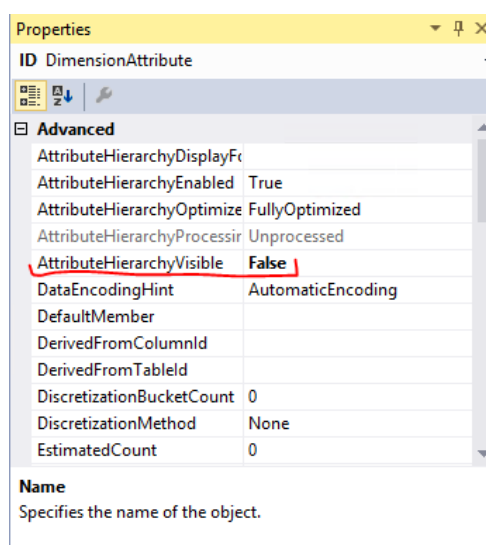
Dalším krokem v úpravě kostky je úprava dimenzí. V dimenzích změníme názvy, které se uživatelům budou při procházení kostky zobrazovat, a zároveň zaměníme zobrazení hodnot dimenzí. Standardně totiž kostka zobrazuje hodnoty ID, které však nejsou pro uživatele příliš směrodatné. Z toho důvodu provedeme změnu, aby se uživatelům zobrazovaly správné popisy daných ID.

To lze provést několika způsoby. Vhodné je ponechat pro práci s kostkou v dimenzích také ID, jelikož při práci s dotazy MDX se můžou hodit. Pro uživatele je však vhodné zobrazit pouze názvy (popisy). Prvním krokem je tedy vložení sloupce s názvem (v případě dimenze je to sloupec Agreement_Type) do atributů dané dimenze.



Obr. 41 Úprava dimenzí - přesun pole pro zobrazení uživatelům (vlastní snímek obrazovky)

Pokud bychom již neprováděli žádné změny, pak by byly pro uživatele viditelné oba sloupce. Naším cílem je však zobrazit uživateli pouze názvy ze sloupce Agreement Type (NC, RS, NA), proto sloupec ID pro uživatele skryjeme. Toho docílíme ve vlastnostech atributu, konkrétně volbu AttributeHierarchyVisible změníme na False. Nyní již při prohlížení kostky vidíme pouze Agreement Type, který nám zobrazuje názvy hodnot (NC, RS, případně NA).



Obr. 42 Vlastnosti pole v dimenzi - skrytí pole pro uživatele (vlastní snímek obrazovky)

Takové nastavení provedeme i pro všechny ostatní dimenze tak, abychom uživateli zobrazovali pouze tyto názvy. Hodnoty IT si ponecháme skryté pro případné využití přes dotazy MDX.

6.9 Vytvoření Calculations

V zadání od obchodního oddělení chceme kromě hodnoty výběrů a hodnoty očekávaných splátek sledovat také jejich poměr. Takový výpočet je ideálním adeptem na vytvoření Calculations v kostce, které bude tento poměr počítat s dat, které již momentálně kostka obsahuje.

Když si otevřeme naši kostku a půjdeme na záložku Calculations, pak můžeme definovat vlastní výpočet, který můžeme také zařadit do nějaké zvolené skupiny metrik. Vyplníme tedy název pro naši novou metriku – Collections To Debit. Nejdůležitější součástí je výraz, pomocí něhož budeme tuto metriku počítat. Jelikož se v našem případě jedná o poměr hodnoty výběrů (Collection_Value) vůči hodnotě očekávaných splátek (Debit_Value), pak do výrazu zadáme toto dělení. Výstupním formátem tohoto výpočtu má být hodnota v procentech, proto vybírám ve Format string možnost Percent. Další volitelný parametr nám umožní zneviditelnit tuto metriku pro uživatele. Pokud chceme tuto metriku zařadit do již stávající skupiny metrik, pak vybereme v Associated measure group vybranou skupinu.

The screenshot shows the configuration interface for a calculated field. The 'Name' field is set to '[Collections To Debit]'. Under 'Parent Properties', the 'Parent hierarchy' is set to 'Measures'. The 'Expression' field contains the formula `[Measures].[Collection Value]/[Measures].[Debit Value]`. Under 'Additional Properties', the 'Format string' is set to 'Percent', 'Visible' is 'True', 'Non-empty behavior' is set to a dropdown, 'Associated measure group' is 'Agreements', and 'Display folder' is empty. There are also checkboxes for 'Color Expressions' and 'Font Expressions'.

Obr. 43 Vytváření počítaného pole (vlastní snímek obrazovky)

Abychom si ověřili funkčnost tohoto výpočtu, musíme spustit procesování kostky a následně v Excelu ověříme, zda je náš výpočet v pořádku.

Popisky řádků	Collection Value	Debit Value	Collections To Debit
anonymizováno	52684446	117537377	44,82%
	71009315	116008222,4	61,21%

Obr. 44 Ukázka funkčnosti počítaného pole (vlastní snímek obrazovky)

Výpočet se nám podařilo nastavit správně, hodnoty se zobrazují správně i při seskupování dat.

6.10 Nastavení KPI – pro poměr výběrů a očekávaných splátek

Pro snadnou orientaci vývoje tohoto ukazatele můžeme do kostky přidat také klíčový ukazatel výkonu - KPI, díky němuž bude možno snadno identifikovat úspěšnost výběrů splátek.

Pokud je pro uživatele některý z ukazatelů klíčový pro rozhodování, pak je vhodné takové ukazatele nastavit jako klíčové. Tím získáme možnost prezentovat taková data ve formě semaforového ukazatele, kdy uživatel hned na první pohled vidí, jak se ukazatel ve vybraném pohledu pohybuje.

Nastavení provádíme přímo v kostce na záložce KPIs, kde vybereme možnost New KPI v levém horním okně KPI Organizer. Ukazatel si pojmenujeme jako KPI Collections To Debit a vybereme skupinu metrik, ze které tento ukazatel bude vycházet, v případě naší kostky je to skupina Agreements, jelikož obsahuje informace o poměru výběrů a očekávaných splátek. Do pole Value Expression vyplníme hodnotu, na základě níž budeme volit barvy semaforu pro tento ukazatel. Pod záložkou Status si následně zvolíme, v jakém grafickém podání chceme data reprezentovat (lze zvolit jednoduché tvary, které pomocí barev reprezentují vypočtenou hodnotu). Poslední nedílnou součástí nastavení klíčového ukazatele je výraz pro posouzení hodnoty. Hodnota může nabývat různých hodnot, ale pro symboly klíčového ukazatele jsme nuceni tyto hodnoty zařadit do třech skupin tak, abychom ve výsledku dostali hodnoty -1, 0 a 1. Tento výraz napíšeme pomocí jednoduchého CASE výrazu v T-SQL jazyce.

Name:

Associated measure group:

Value Expression

Goal Expression

Status

Status indicator:

Status expression:

```
CASE
WHEN [Measures].[Collections To Debit] < 0.75 THEN -1
WHEN [Measures].[Collections To Debit] >= 0.75 AND [Measures].[Collections To Debit] < 1 THEN 0
WHEN [Measures].[Collections To Debit] >= 1 THEN 1
END
```

Obr. 45 Nastavení KPI - klíčového ukazatele (vlastní snímek obrazovky)

Jak můžeme vidět na ukázce níže, tak tento klíčový ukazatel funguje dle našich představ.

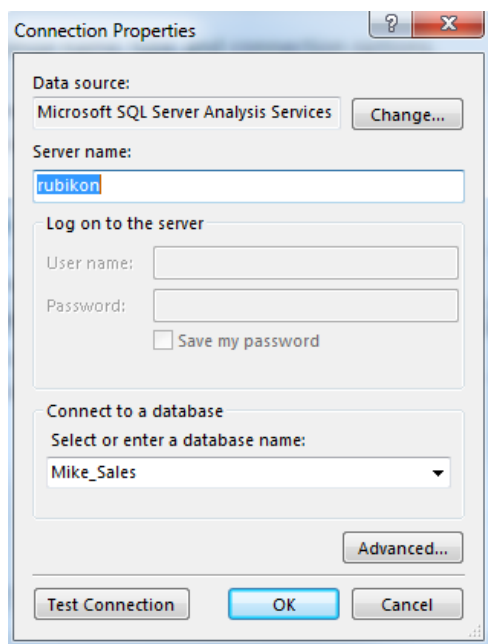
Popisky řádků	Collection Value	Debit Value	KPI Collections To Debit	KPI Collections To Debit – Stav
	52684446	117537377	44,82%	🔴
anonymizováno	71009315	116008222,4	61,21%	🔴
	116173730	115927728,9	100,21%	🟢
	100583890	118493232,8	84,89%	🟡

Obr. 46 Ukázka funkčnosti klíčového ukazatele v Excelu (vlastní snímek obrazovky)

7 Tvorba reportu z OLAP kostky

Z OLAP kostky lze také vytvářet reporty v Reporting Services. K tomu lze využít Report Builder ve verzi 2016, ve kterém bude proveden návrh reportu a uložení reportu na reporting server s názvem ADS.

Prvním krokem k vytvoření reportu je vytvoření datového zdroje, jako připojení k analytické databázi, která obsahuje kostku. V nabídce Report Data proto pravým tlačítkem myši klikneme na Data Sources a zvolíme Add Data Source. Je potřeba datový zdroj pojmenovat, a vybrat, o jaký typ připojení se jedná (v tomto případě Microsoft SQL Server Analysis Services). Tlačítkem Build se následně provede definování připojení k serveru rubikon, z něhož se zvolí databáze Mike_Sales, která obsahuje dříve vytvořenou kostku.



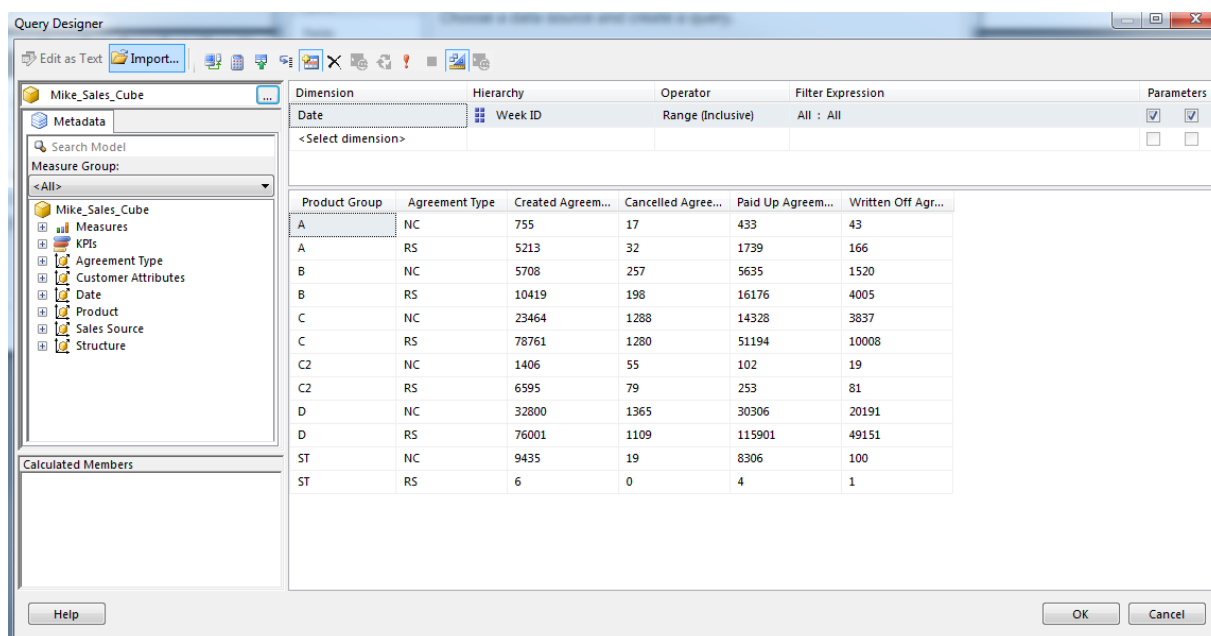
Obr. 47 Nastavení připojení k serveru analytických služeb v Reporting Services (vlastní snímek obrazovky)

Připojení lze otestovat pomocí tlačítka Test Connection.

Dalším krokem pro získání dat je výběr datového setu, se kterým bude report pracovat. Pravým tlačítkem myši tedy klikneme na Dataset a zvolíme Add Dataset. Datový set si pojmenujeme a vybereme datový zdroj pro tento datový set, který jsme si definovali v předchozím kroku. Pokud zvolíme datový zdroj z analytických služeb, pak nemůžeme zapisovat do pole pro dotazy. Z toho důvodu musím přistoupit k variantě Query Designer, přes který můžeme poměrně snadno naklikat požadovaný MDX dotaz.

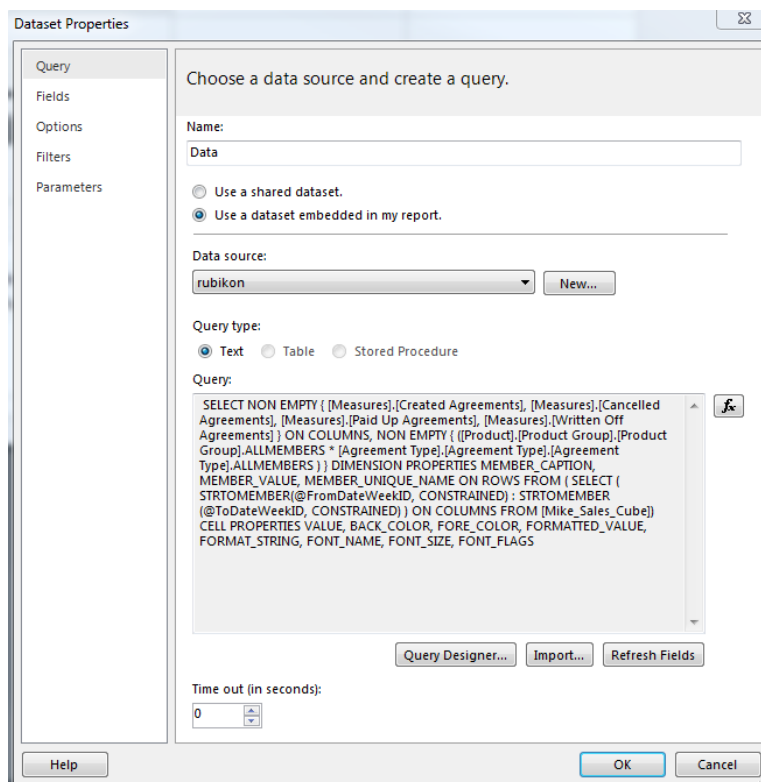
V levé části okna můžeme vidět všechny metriky a dimenze z naší OLAP kostky. Pro naše účely jsem se rozhodl demonstrovat tvorbu reportu na rozdělení portfolia smluv dle produktových skupin a typu smluv za určité časové období, které si bude uživatel reportu volit pomocí parametrů. Proto si vybereme požadované metriky (vytvořené, zrušené,

doplacené a odepsané smlouvy) a k tomu tyto dva pohledy (Product Group a Agreement Type). Abychom uživatelům umožnili volbu časového období, pak musíme do dimenzí (horní část okna) přetáhnout naše účetní týdny z dimenze Date. Jelikož chceme uživatelům nabídnout možnosti časového rozmezí, pak musíme jako Operator zvolit Range. Současně vybereme hodnoty, které se budou uživatelům nabízet, a zaškrtneme možnost zobrazení těchto hodnot v parametrech při otevírání reportu.



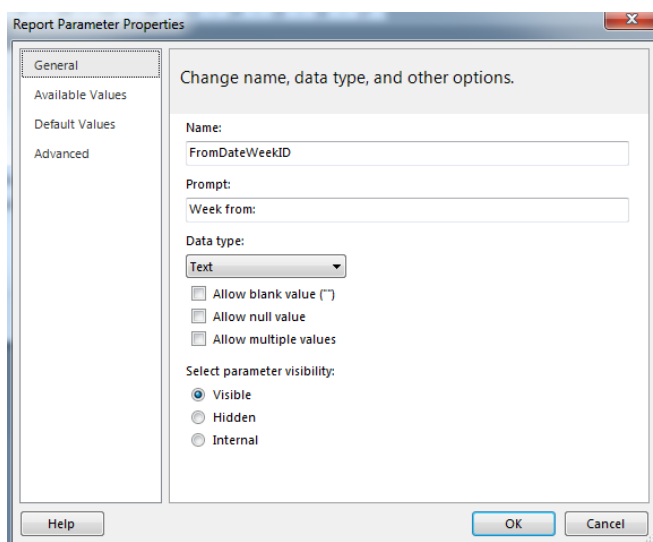
Obr. 48 Vytváření dotazu MDX přes Query Designer v Reporting Services (vlastní snímek obrazovky)

Přes Query Designer se nám automaticky vytvoří MDX dotaz do kostky, pomocí něhož budeme získávat data do reportu.



Obr. 49 MDX dotaz v datovém setu vytvořený přes Query Designer (vlastní snímek obrazovky)

Vzhledem k tomu, že jsme již při vytváření dotazu pracovali s parametry, pak byly tyto parametry automaticky vytvořeny pod záložkou Parameters v reportu, kde je můžeme dále upravovat. Díky tomu, že byly parametry vytvořeny přes Query Designer, jsou ihned připraveny k použití, mají totiž již definované dostupné hodnoty a výchozí hodnoty. Jedinou změnu, kterou provedeme, je změna textu výzvy pro uživatele.



Obr. 50 Nastavení parametrů v Reporting Services (vlastní snímek obrazovky)

Jakmile máme vytvořený datový set a nastavené parametry pro spouštění reportu, můžeme se pustit do prezentace dat samotných. Vytvoříme si tedy tabulku, do které vložíme hodnoty, jenž chceme zobrazovat uživatelům.

Cube Report - Sales

Product Group	Agreement	Agreements			
		Created	Cancelled	Paid Up	Written Off
[Product_Group]	[Agreement_Ty	[Created_Agree	[Cancelled_Agr	[Paid_Up_Agrei	[Written_Off_Ac

[&ExecutionTime]

Obr. 51 Návrh reportu v Report Builderu (vlastní snímek obrazovky)

Abychom data zobrazovali uspořádaně, pak vytvoříme nad typy smluv ještě skupinu produktů, která bude nadřazena typům smluv. Tím docílíme seskupení typů smluv NC a RS do produktových skupin. Níže můžeme vidět, že každá produktová skupina rozlišuje tyto dva typy smluv.

Product Group	Agreement Type	Agreements			
		Created	Cancelled	Paid Up	Written Off
A	NC	755	17	433	43
	RS	5 213	32	1 739	166
B	NC	5 708	257	5 635	1 520
	RS	10 419	198	16 176	4 005
C	NC	23 464	1 288	14 328	3 837
	RS	78 761	1 280	51 194	10 008
C2	NC	1 406	55	102	19
	RS	6 595	79	253	81
D	NC	32 800	1 365	30 306	20 191
	RS	76 001	1 109	115 901	49 151
ST	NC	9 435	19	8 306	100
	RS	6	0	4	1

Obr. 52 Výsledná tabulka z vytvořeného reportu (vlastní snímek obrazovky)

8 Zpětná vazba od uživatelů

Po nasazení kostky byli uživatelé z obchodního oddělení seznámeni s jejím fungováním. Společně jsme provedli připojení ke kostce přes Excel a následně jim byly vysvětleny základní metriky a možnosti pohledů.

Uživatelé následně zkoušeli s kostkou pracovat a ohlasy byly převážně pozitivní. Jelikož kostka zobrazovala požadovaná data a tato data byla označena jako správná, tak ji uživatelé začali využívat pro ad-hoc analýzy zejména pro rozpad smluv po produktech, který doposud neměli nikde k dispozici. Současně za zajímavý pohled označili rozpad portfolia zákazníků po věkových a příjmových skupinách.

Vzhledem k nastavení kostky MOLAP, si uživatele pochvalují také rychlost odezvy.

Již nyní se objevují nové požadavky na úpravy kostky, zejména přidávání dalších metrik a pohledů.

Závěr

Úvodní část práce byla věnována teorii, která z počátku čtenáře seznamuje s oblastí business intelligence v podnicích a následně také veškerému obsahu této oblasti. Následně se teoretická část práce věnuje transformaci dat do datového skladu z jednotlivých podnikových informačních systémů. Ve druhé části teorie pak dochází k seznámení čtenáře s OLAP kostkami. Informace k OLAP kostkám jsou popsány detailněji a dochází k popisu jednotlivých typů kostek a následnému shrnutí výhod a nevýhod tohoto BI řešení.

Praktická část je pak věnována čistě tvorbě OLAP kostky, pro kterou vytvořilo zadání obchodní oddělení společnosti. Jelikož aktuálně ve společnosti neexistuje datový sklad, ze kterého by bylo možné bez jakýchkoliv modifikací čerpat data, tak bylo potřeba pro účely tvorby této kostky vytvořit datový model tak, aby pokryl veškeré požadavky zadání a zároveň byl použitelný pro tvorbu OLAP kostky. Datový model byl detailně popsán, jak technicky tak obchodně. Pro lepší interpretaci datového modelu byl model navržen v Enterprise Architectu a následně došlo k naplnění tohoto modelu daty v navrhované struktuře. Jakmile byla data připravená, přistoupilo se k samotnému vytvoření kostky skrze Visual Studio. V prvním kroku bylo vytvořeno připojení k připravenému datovému zdroji a následně nad tímto zdrojem vytvořena OLAP kostka Mike_Sales_Cube. Jakmile došlo k nasazení kostky na server a ověření její funkčnosti, přistoupilo se k úpravám jednotlivých dimenzí tak, aby uživatelům podávaly požadované informace. V dalším kroku došlo k vytvoření hierarchií pro obchodní strukturu a časovou dimenzi. Současně byla demonstrována tvorba počítaného pole a nastavení klíčového ukazatele, z něhož mohou uživatelé ihned vidět, zda se tento ukazatel vyvíjel pozitivně, negativně či spíše neutrálně. Poslední krok praktické části pak byl věnován vytvoření jednoduchého reportu nad kostkou v Reporting Services.

I přesto, že se vše nepodařilo nastavit hned z počátku korektně, nakonec díky postupnému ladění a opakovanému procesování byla kostka uvedena do provozu a nyní ji lze využít pro základní reporting a ad-hoc analýzy obchodního oddělení.

Použitá literatura

1. LABERGE, Robert, 2012. *Datové sklady: agilní metody a business intelligence*. Brno: Computer Press.
2. RAD, REZA, 2014. *Microsoft SQL Server 2014 business intelligence development beginner's guide*. Birmingham, UK : Packt Pub.
3. LIDBERG, SIMON and MANNARI, YUVRAJ, 2013. *Getting started with SQL server 2012 cube development*. Birmingham : Packt Publishing.
4. WITHEE, Ken, 2010. *Microsoft Business Intelligence for dummies*. Hoboken, NJ: Wiley.
5. CAMERON, Scott, 2009. *Microsoft SQL server 2008 analysis services: step by step*. Redmond, WA: Microsoft Press.
6. WEBB, CHRIS, FERRARI, ALBERTO and RUSSO, MARCO, 2014. *Expert Cube Development with SSAS Multidimensional Models*. 2. Birmingham, UK : Packt Publishing.
7. Differences Between OLAP, ROLAP, MOLAP, and HOLAP - TechNet Articles - United States (English) - TechNet Wiki. *Document Moved* [online]. Copyright © 2015 Microsoft Corporation. All rights reserved. [cit. 20.03.2019]. Dostupné z: <https://social.technet.microsoft.com/wiki/contents/articles/19898.differences-between-olap-rolap-molap-and-holap.aspx>
8. What is the Definition of OLAP? OLAP Definition. *What is OLAP? Online Analytical Processing - Learn about OLAP, Types of OLAP | OLAP.com* [online]. [cit. 20.03.2019]. Dostupné z: <http://olap.com/olap-definition/>
- 9 Data Warehousing OLAP. *Java NIO, PyTorch, SLF4J, Parallax Scrolling, Java Cryptography, YAML, Python Data Science, Java i18n, GitLab, TestRail, VersionOne, DBUtils, Common CLI, Seaborn, Ansible, LOLCODE, Current Affairs 2018, Apache Commons Collections*[online]. Copyright © Copyright 2019. All Rights Reserved. [cit. 22.03.2019]. Dostupné z: https://www.tutorialspoint.com/dwh/dwh_olap.htm
10. GUNDERLOY, Mike and Tim SNEATH, 2001. *SQL Server developers guide to OLAP with analysis services*. San Francisco, CA: SYBEX.
- 11 Why use a SSAS cube? | James Serra's Blog. *James Serra\'s Blog* [online]. Copyright © 2019 [cit. 25.03.2019]. Dostupné z: <https://www.jamesserra.com/archive/2013/08/why-use-a-ssas-cube/>