Mendelova univerzita v Brně
Provozně ekonomická fakulta

Možnosti analytických nástrojů v prostředí MS SQL Serveru

Diplomová práce

Vedoucí práce:
Ing. Jan Přichystal, PhD.

Bc. Markéta Mazáčová

Brno 2017

Na tomto místě bych ráda poděkovala svému vedoucímu práce, panu Ing. Janu Přichystalovi, Ph.D., za odborné vedení, cenné rady a ochotu při tvorbě práce. Dále chci poděkovat svojí rodině a přátelům za podporu během celého studia.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Možnosti analytických nástrojů v prostředí MS SQL Serveru**
vypracovala samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*
Jsem si vědoma, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.
Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 21. května 2017 .....

Abstract

Mazáčová, M. *MS SQL Server analysis tools options.* Brno, 2017. Master thesis. Mendel University in Brno, Faculty of Business and Economics.

The thesis compares the development time and performance of two types of Microsoft analytical models created in Visual Studio that are deployed on the local analytics servers and the Azure cloud server. Theoretical research will be summarizing business intelligence and data warehouses components, Microsoft tools for business intelligence realization and features using in-memory database technologies. The practical part introduces the AdventureWorks demonstration database, describes the implementation of both analytical models and compares both models with respect to the time-consuming development, performance and suitability of deployment.

Keywords

business intelligence, OLAP, MS SQL Server, Azure Analysis Services, SQL Server Analysis Services, multidimensional model, tabular model, in-memory

Abstrakt

Mazáčová, M. *Možnosti analytických nástrojů v prostředí MS SQL Serveru.* Brno, 2017. Diplomová práce. Mendelova univerzita v Brně, Provozně ekonomická fakulta.

Práce porovnává časovou náročnost vývoje a výkon u dvou typů analytických modelů od firmy Microsoft vytvořených v nástroji Visual Studio, nasazených na lokální analytické servery a na cloudový server Azure. V rámci teoretické rešerše jsou stručně popsány hlavní komponenty business intelligence a datových skladů. Dále jsou shrnuty nástroje firmy Microsoft pro oblast business intelligence a jejich prvky využívající in-memory databázové technologie. Praktická část seznamuje s demonstrační databází AdventureWorks, popisuje realizaci obou analytických modelů a srovnává oba modely z hlediska časové náročnosti vývoje, výkonnosti a vhodnosti nasazení.

Klíčová slova

business intelligence, OLAP, MS SQL Server, Azure Analysis Services, SQL Server Analysis Services, multidimenzionální model, tabulární model, in-memory

Obsah

Seznam obrázků

Seznam tabulek

Seznam zdrojových kódů

1 Úvod a cíl práce

1.1 Úvod

První myšlenka využívání informací v podnikání se objevila v první polovině 19. století. Autor Richard Devense ve svém díle *Cyclopaedia of Commercial and Business Anecdotes* popisoval techniku úspěšného bankéře Henryho Furnese (Heinze, 2014). Ten ve svých obchodech získával konkurenční výhodu díky detailnímu poznání nestabilit trhu a politických otázek. Tato myšlenka zůstala ve své omezené podobě až do 20. století, kdy její realizace začala být spojována s rozvojem počítačových technologií a informačních systémů.

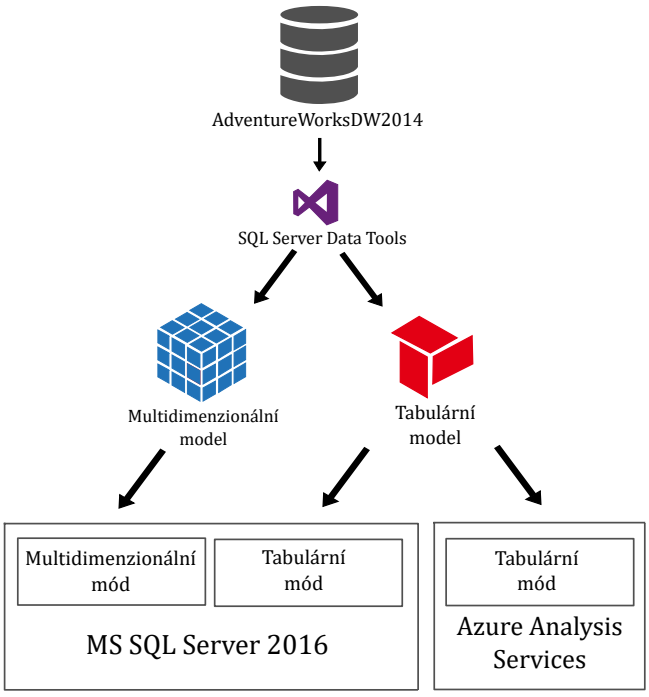
Informačním systémem označujeme soubor hardwaru, softwaru, lidí, procesů a činností sloužící pro sběr, zpracovávání a šíření dat. Data jsou neorganizované a surové údaje ve formě písmen, čísel a symbolů. Jsou vytvářena zkoumáním, měřením a zaznamenáním vlastností určitého objektu nebo jevu. Údaje jsou shromažďovány v mnoha organizacích a institucích, například v podnicích (data o příjmech, prodejích, cenách akcií nebo jednotlivých ziscích), ve vládních organizacích (kriminalita, nezaměstnanost, gramotnost) a nevládních organizacích (údaje o počtu bezdomovců v neziskových společnostech). Jejich uložení je zajištěno pomocí relačních databázových systémů nezávisle na použitých softwarových prostředcích. Databáze ukládají data do formy tabulek a umožňují mezi tabulkami vytvářet vztahy (relace), čímž je možné realizovat schémata na základě reálných vlastností původních objektů či zkoumaných jevů. Tabulka obsahuje řádky (záznamy) a sloupce (atributy), na jejichž průniku jsou obsaženy konkrétní hodnoty dat. Každý záznam je opatřen jedinečným identifikátorem (primárním klíčem). Pokud navrhujeme relační databázové schéma do kterého jsou data ukládána, je třeba dodržovat pravidla normálních forem. Ty odstraňují nadbytečná (redundantní) data a zvyšují kvalitu práce s databází při aplikaci logiky organizace.

Ukládání dat umožňuje jejich převod na informace mající svůj účel a smysl. Kvalitní informace jsou důležitou součástí organizací a institucí. Slouží jako podpora pro rozhodování a efektivní, rychlou reakci na požadavky zákazníků a změn trhu. Musejí být dostupné pro řídící pracovníky v určitý čas, na konkrétním místě a ve správné formě. Tyto požadavky jsou zajištěny pomocí metod business intelligence (BI).

Business intelligence je soubor technologií, aplikací a praktik umožňující sběr, sjednocení, analýzu a prezentaci informací. Předkládá uživateli historický, současný a předpokládaný budoucí stav podnikových operací. Hlavním úkolem BI je tvorba systému zajišťující podnikové zpravodajství a online analytické zpracování dat. Dále pak zajišťuje objevování nových znalostí pomocí metod dolování z dat a textu, měření výkonnosti podniku a časové porovnávání.

1.2 Cíl práce

Cílem práce je provést rozbor analytických možností MS SQL Serveru ve verzi 2016. Prakticky budou realizovány dva typy analytických modelů: multidimenzionální a tabulární, na základě datového skladu demonstrační databáze AdventureWorks, jenž se nasadí na instance MS SQL Serveru a Azure Analysis Services. Obrázek č. 1 zobrazuje schéma jejich nasazení.



Obrázek 1: Schéma nasazení analytických modelů na servery

Vytvořené modely budou srovnány z hlediska výkonu, doby vývoje a vhodnosti nasazení. Srovnání bude sloužit pro zhodnocení obou modelů a vyvození příslušných závěrů. Pro splnění celkového cíle je třeba realizovat následující dílčí kroky vycházející ze zadání:

- Seznámit se s problematikou tvorby datových skladů a business intelligence s důrazem na in-memory technologie v prostředí MS SQL Server.
- Seznámit se se strukturou dat v demonstrační databázi AdventureWorks.
- Realizovat analytickou vrstvu datového skladu s využitím multidimenzionálního a tabulárního přístupu v nástroji SQL Server Analysis Services.
- Srovnat efektivitu obou řešení s důrazem na vhodnost nasazení, časovou náročnost vývoje a výkonnostní rozdíly.
- Sumarizovat a diskutovat výsledky. Vyvodit příslušné závěry a doporučení.

2 Metodika práce

Práce bude rozdělena na teoretickou a praktickou část. V teoretické části bude vypracována rešerše zabývající se tématem business intelligence, jejími principy a komponenty. Dále budou popsány produkty SQL Server 2016 a Azure Analysis Services firmy Microsoft, vyvinuté pro oblast business intelligence se zaměřením na online analytické modely a jejich in-memory technologie. Poslední část rešerše bude seznamovat s demonstrační databází AdventureWorks a výběrem tabulek, ze kterých bude praktická část vycházet.

Ve druhé části bude realizován tabulární a multidimenzionální model na základě zdrojového datového skladu v aplikaci Visual Studio. Tyto modely budou nasazeny na instance SQL Serveru a na server Azure. Další část se bude zabývat srovnáváním modelů z následujících tří hledisek:

1. Výkonnostní rozdíly
2. Časová náročnost vývoje
3. Vhodnost nasazení

2.1 Výkonnostní rozdíly

Pro hodnocení výkonnostních rozdílů bude využita metodika představená Marcem Russo na přednášce TechEd v roce 2012 (Russo a Ferrari, 2012). Výkon analytických modelů je zde zhodnocen časem zpracování individuálních dotazů zaslaných na model. Metodika se skládá z následujících kroků:

- Rozšíření klíčové zdrojové tabulky z databáze datového skladu
- Vytvoření dotazů, které budou databázím předkládány
- Opakované spouštění dotazů, prokládané čistěním mezipaměti
- Zapsání celkového času zpracování
- Shrnutí výsledků výkonnostních rozdílů

2.2 Časová náročnost vývoje

Pro měření časové náročnosti vývoje bude využita některá z metod měření práce. Ta se snaží určit čas, který daná operace spotřebovává. V praxi jsou rozlišovány metody nepřímé a přímé. Jednou z nepřímých metod je i hrubý odhad času. Tato metoda se využívá, pokud má odhadující dostatek zkušeností s vývojem, nebo má k dispozici zkušenou osobu v této oblasti. Nepřímý odhad může být proveden i na základě historických údajů nebo pomocí metody *Methods-Time Measurement* (MTM). Ta rozděluje práci na jednotlivé pohyby a měří jejich časovou spotřebu. Vyžaduje ale detailní znalost všech možných operací, které lze provádět. Tato metodika je však

svým obsahem velmi rozsáhlá. Mezi metody přímého měření patří časová studie, založena na chronometráži neboli měření dílčích úseků práce pomocí stopek. Tím se stanoví délka určitého pracovního děje a zaznamená se do předem připraveného formuláře nebo tabulky (Dlabač, 2015).

Jelikož nejsou k dispozici historické záznamy o době trvání tvorby modelu a nemáme k dispozici osobu s dostatečnými zkušenostmi umožňující odhad časové náročností vývoje, bude využita metoda časové studie. Tato metoda slouží zejména pro odhad časové náročnosti výroby, proto bude v této práci mírně upravena, aby odpovídala potřebám analytických modelů. Měření času se bude skládat z následujících kroků (Magagnotti a Spinelli, 2012):

- Zvolení cíle měření
- Studium modelovacích technik
- Návrh analytického modelu pro měření
- Sběr časových údajů

2.3 Vhodnost nasazení

Vhodnost nasazení bude posouzena formou praktických poznatků získaných během modelování a podle dostupných zdrojů zabývající se touto problematikou. Hodnoceny budou následující prvky:

- **Zdroje serveru** – zejména doporučený hardware
- **Funkce modelů** – shrnutí poznatků z dostupných materiálů
- **Objem zpracování dat** – použití modelu v závislosti na množství dat
- **Aplikace obchodní logiky** – shrnutí možností pro aplikaci obchodní logiky s poznatky získanými při praktickém modelování

3 Business Intelligence a datové sklady

Business Intelligence je část informatiky, která primárně podporuje analytické, rozhodovací a plánovací operace podniků. Je tvořena souborem nástrojů, aplikací a metodik, které umožňují využívat vlastní data vznikající při běžných transakcích (Pour, Maryška a Novotný, 2012). Dalším důležitým pojmem jsou datové sklady (*Data Warehouses*). Zjednodušeně se jedná o uložiště sjednocených dat z více zdrojů, poskytující celistvý obraz o podnikání v určitém čase.

Úspěšné společnosti investují do řešení business intelligence a datových skladů nemalé finanční prostředky. To má do podniku přinést aktuální informace o jejich zásobování, produkci a zákaznících, které jim mohou v budoucnu zajistit větší úspěch.

Podnikem kolují dva typy informací: operativní a analytické. Operativní informace vznikají v transakčních aplikacích podporujících firemní procesy. Může se jednat o sběr objednávek, registrace nových zákazníků nebo monitorování stávajících procesů. Transakční aplikace vytvářejí nová data a případně je aktualizují. Dále musejí zajistit rychlý přístup k jejich detailům a vykonat všechny realizované operace. Analytické informace naproti tomu vznikají v aplikacích business intelligence. Ty žádná nová data nevytvářejí, ale využívají ta z databází transakčních systémů (Pour, Maryška a Novotný, 2012).

3.1 Historie Business Intelligence

V 70. a 80. letech 20. století se prodával drahý hardware ve formě sálových počítačů s omezenou výpočetní silou. Některé podniky si nicméně počítače zakoupily a začaly používat první aplikace pro ukládání dat. Tehdejší podpora rozhodování byla zajištěna převážně v listinné podobě. Pokud chtěly využít systémy pro podnikové zpravodajství, bylo třeba najmout programátora. První takové systémy byly velmi neflexibilní a označovaly se jako systémy pro podporu rozhodování (*Decision Support Systems*).

V 80. letech se začaly využívat intuitivnější relační databáze, které umožňovaly koncovým uživatelům vytvářet jednoduché zprávy pomocí dotazů. Ty byly značně neefektivní a musely se spouštět mimo pracovní dobu, aby neovlivnily běh transakcí. Na konci 80. let začaly podniky využívat osobní počítače s kancelářskými aplikacemi typu Word, Excel nebo Access. Vývoj umožnil koncovým uživatelům tvorbu vlastních reportů v tabulkové formě bez pomoci IT pracovníků. Firemní data ale pro jejich potřeby nebyla přímo přístupná.

Od roku 1990 začalo, díky dohodám o volném obchodu, docházet ke globalizaci výpočetní techniky a sítí. Začala vznikat konkurence a došlo k rozvoji datových skladů. Obecný termín business intelligence, zastřešující veškeré metodiky, pojmy a přístupy, byl definován v roce 1989 analytikem Howardem J. Dresnerem ze společnosti Gartner (Heinze, 2014).

3.2 Dimenzionální modelování

Business intelligence ukládá data do analytických databází využívajících dimenzionální modelování. Jedná se o způsob zjednodušení databáze a prezentace analytických dat. Model obsahuje stejná data jako normalizovaná relační databáze v transakčních systémech nebo v datových skladech, pouze jsou uložena v jiném schématu. Dimenzionální model rozlišuje dva typy tabulek, jedná se o tabulky faktů a dimenzí. Tyto tabulky jsou odvozeny od způsobu nahlížení na podniková data (Kimball a Ross, 2013).

Tabulky faktů ukládají měrné jednotky podniku z obchodování (neboli metriky) zaznamenané v určitém čase. Tato data jsou často uložena v největších a nejširších tabulkách databáze. Příkladem metrik mohou být bankovní operace, objemy prodejů, objednávek, dodávek nebo návštěv jednotlivých webových stránek. Tabulka obsahuje cizí klíče do tabulek dimenzí a hodnoty jednotlivých metrik.

Dimenzionální tabulky jsou nedílnou součástí tabulek faktových. Obsahují textové údaje, které jsou spojeny s metrikami. Dimenzionální tabulky mají často více sloupců nebo atributů, ale mívají méně řádků než tabulky faktové. Každá dimenze je definována primárním klíčem, který slouží k zachování referenční integrity s propojenými faktovými tabulkami. Atributy dimenzí jsou používány jako primární zdroj dotazů, seskupování a popisů reportů. Pokud by uživatel chtěl zjistit objemy prodejů určité značky produktu, musí být značka atributem jedné z dimenzí. Nejčastějšími dimenzemi jsou zákazník, produkt, datum nebo geografie. Dimenze mohou obsahovat hierarchie dat. Typickým příkladem je geografická dimenze, která se rozděluje do úrovní podle konkrétního geografického členění lokace. Na nejvyšší úrovni je kontinent, dále následuje konkrétní země, pak město a případně ulice. Podrobnosti detailů úrovní nazýváme granularitou. Tyto úrovně umožňují libovolné vnořování (*drill down*) a vynořování (*drill up*) (Kimball a Ross, 2013).

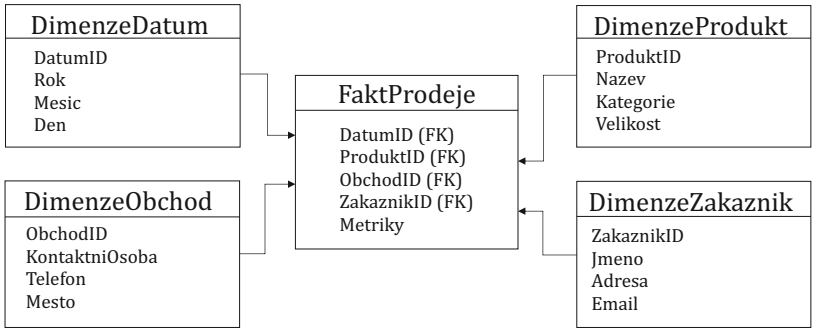
Dimenzionální model můžeme realizovat v relačním databázovém systému nebo v prostředí multidimenzionální databáze (Kimball a Ross, 2013).

Dimenzionální model v relační databázi

Topologie modelů realizovaných v relačním databázovém systému jsou označovány jako hvězdicové schéma *(star schema)* nebo vločkové schéma *(snowflake schema)* (Kimball a Ross, 2013).

Hvězdicové schéma (obrázek č. 2) je tvořeno menším počtem dimenzí, které často obsahují vlastní hierarchie. Pokud v tabulkách dochází k časté aktualizaci dat, je lepší dimenze normalizovat do schématu sněhové vločky (obrázek č. 3). Tím dochází k rozdělení dimenzí podle hierarchií do více tabulek. Data v tabulkách tak nejsou redundantní.

V databázi datového skladu se používá kombinace obou schémat. Metriky logicky rozdělujeme do faktových tabulek, mezi nimiž jsou používány stejné sdílené dimenze (Pour, Maryška a Novotný, 2012).



Obrázek 2: Datový model hvězdicové schéma (Pour, Maryška a Novotný, 2012).

Dimenzionální model v multidimenzionální databázi

Prostředí multidimenzionální databáze je navrhnuto pro realizaci dimenzionálních modelů. Databázi zobrazujeme jako vícerozměrnou krychli (kostku), která odpovídá tabulce v relační databázi. Každá hrana kostky přestavuje jednu dimenzi. Jednotlivé záznamy se nacházejí na jejich průsečících. Může se stát, že průsečík neobsahuje data žádná a pak hovoříme o tzv. řídké kostce (Lacko, 2003).

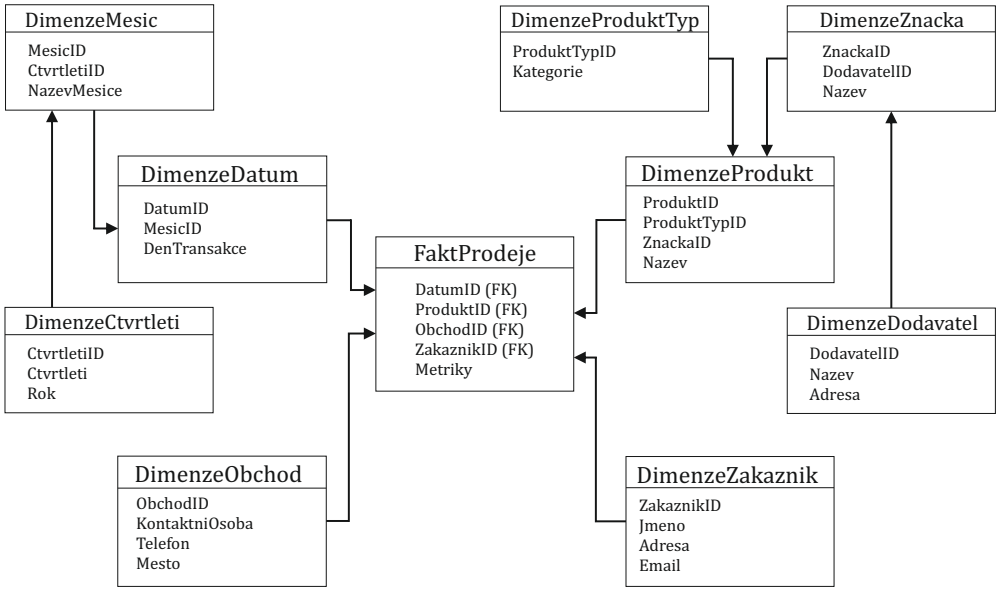
Krychle je realizovaná za pomocí **OLAP** (*Online Analytical Processing*) technologií, můžeme se tedy setkat s označením OLAP kostka. Ta umožňuje online zpracování a analyzování dat díky rychlé změně pohledů (dimenzí) na ekonomická data (Pour, Maryška a Novotný, 2012).

Příkladem může být trojrozměrná kostka obsahující časovou, produktovou a geografickou dimenzi. Při analýze je pak snadné zjistit objemy prodejů určitého produktu za jednotlivé roky a v určitých městech. V praxi se často používá více dimenzí a je obtížné si databázi představovat jako kostku (Lacko, 2003).

Multidimenzionální databáze používají více technologií pro zpracování a komprimaci dat. Jedná se o multidimenzionální OLAP, relační databázový OLAP a hybridní OLAP:

- **Multidimenzionální OLAP (MOLAP)** zpracovává záznamy z datového skladu nebo z primárních systémů. Data jsou ukládána do multidimenzionálních datových struktur společně s jejich předběžnými výpočty, sumarizacemi a agregacemi. Ty jsou prováděny v závislosti na dostupných časových a technologických zdrojích. Díky těmto výpočtům je databáze schopná rychle zobrazovat výsledky nad více dimenzemi. Výhodou MOLAP je tedy rychlá odezva na uživatelské dotazy. Data jsou ale ukládána redundantně v multidimenzionální i v relační databázi a při realizaci modelu s více dimenzemi může dojít k radikálnímu nárůstu objemu dat k uložení (Lacko, 2003).
- **Relační databázový OLAP (ROLAP)** zpracovává data z relační databáze datového skladu. Multidimenzionalita dat je vyřešena pomocí metadat[1], která

[1]Metadata jsou data, která popisují informace o jiných datech.



Obrázek 3: Datový model vločkové schéma (Pour, Maryška a Novotný, 2012)

jsou spolu s daty ukládána do relačních databází. Při uživatelském dotazování metadata sestaví SQL[2] příkaz k získání požadovaných dat. Uložením záznamů do relačních databází nevzniká jako v případě MOLAP problém s redundancí. Jeho nevýhodou jsou velké nároky na správu a výkon databáze (Lacko, 2003).

- **Hybridní OLAP (HOLAP)** využívá oba způsoby uložení dat a tím částečně odstraňuje jejich nevýhody. Detailní data se ukládají do relačních databází a výpočty a agregace nad daty jsou uloženy v multidimenzionálních strukturách. Data jsou při uživatelském dotazování ukládána do multidimenzionální mezipaměti[3] (Lacko, 2003; Pour, Maryška a Novotný, 2012).

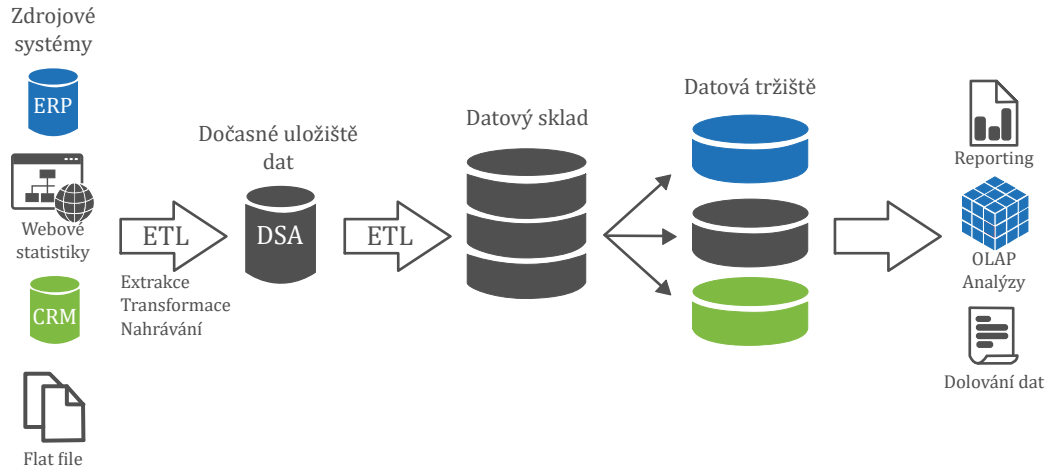
3.3 Architektura business intelligence

Samotné řešení business intelligence se mění v závislosti na podmínkách daného podniku. Obecné uspořádání jeho prvků je zobrazeno na obrázku č. 4. Využití a organizace komponent je velice variabilní a může být realizováno za pomocí různých

[2]SQL (*Structured Query Language*) je standardizovaný dotazovací jazyk, sloužící pro komunikaci s relační databází. Příkazy jsou používány například pro aktualizaci dat nebo získávání informací.

[3]Mezipaměť (*Cache*) je část hardwaru nebo softwaru ukládající data. Tímto principem je zrychlen přístup k daným datům.

technologií. Následující podkapitoly budou stručně popisovat jednotlivé prvky BI architektury.



Obrázek 4: Architektura Business Intelligence (Pour, Maryška a Novotný, 2012)

Zdrojové systémy

Zdrojové systémy zaznamenávají data z podnikových operací. Jsou označovány také jako transakční, OLTP[4], operační nebo primární systémy. Nad jejich kontextem a formátem má realizátor BI řešení pouze minimální kontrolu. Jsou charakteristické svou vysokou dostupností a vysokým výpočetním výkonem, který je nezbytný pro obstarávání velkého množství transakcí. Podniky využívají transakční systémy s cílem automatizovat podnikové činnosti. Může se jednat o správu skladového hospodářství, nákupů, prodejů nebo mezd (Lacko, 2003; Pour, Maryška a Novotný, 2012).

Transakční databáze systémů ukládající normalizovaná data jsou určeny k přímému, jednoduchému dotazování a nejsou obecně vhodné pro tvorbu analýz. Historické záznamy jsou v databázi uloženy pouze omezenou dobu, což by v případě analýz historie nebo predikce představovalo nedostatek dat. Podniky běžně používají více operačních systémů, které mezi sebou nemají jednotný integrovaný systém sdílení dat. Data jsou tak rozptýlena v různých nejednotných OLTP databázích. Tvorba analýzy nad transakční databází by i v případě jednoduchého řešení byla velice výpočetně a časově náročná. Během výpočtu by klesl výkon celého databázového stroje, což by zpomalilo i čas odezvy uživatelům systému (Lacko, 2003). Kvůli nevhodnosti transakčních systémů pro analýzy se využívají datové sklady, které z transakčních databází vycházejí a neomezují jejich primární funkci.

[4]Zkratka OLTP neboli *Online Transaction Processing*, je označení pro systémy ukládající data v databázi umožňující snadné zpracování operací v reálném čase za přístupu více uživatelů.

Extrakce, transformace a nahrávání (ETL)

Nástroje ETL (*Extract, Transform, Load*) patří k nejdůležitějším, nejnáročnějším a nejdražším částem procesu business intelligence. Jsou také označovány souslovím datová pumpa. Proces ETL by měl zajišťovat přenos dat mezi libovolnými databázemi nebo soubory (prosté databázové soubory, CSV soubory, Excel soubory atd.). Jeho kvalita ovlivňuje kvalitu celého business intelligence řešení. Datová pumpa je složena ze tří fází – extrakce, transformace a nahrávání (Pour, Maryška a Novotný, 2012).

Ve fázi extrakce je třeba detailní znalost datového zdroje a dat potřebných pro podnikové rozhodování a analýzy. Tato data jsou zkopírována a zpracovávána v dalších fázích ETL.

Transformace převádí data do jednotného formátu, opravuje kolize, zpracovává chybějící údaje, odstraňuje multiplicitní data a zajišťuje datovou kvalitu (Kimball a Ross, 2013).

Po transformaci dochází k nahrávání dat do struktur datového skladu. Převod probíhá v jednotlivých dávkách po určitém časovém intervalu (Pour, Maryška a Novotný, 2012).

Dočasné uložiště dat (DSA)

Dočasné uložiště dat (*Data Staging Area*) slouží k uložení extrahovaných dat ze zdrojových systémů. Jedná se o nejednotná data, která jsou zde, před vstupem do datového skladu, očištěna a sjednocena. Tím, že jsou transformace prováděny v komponentě dočasného uložiště, nedochází k omezování výkonu primárních systémů.

Úložiště obsahuje vždy jen aktuální nová data, ta jsou po transformaci převedena do datového skladu a z úložiště odstraněna (Pour, Maryška a Novotný, 2012).

Datový sklad

Datový sklad (*Data Warehouse*) byl definován jako subjektově orientovaný, integrovaný, časově variabilní a neměnný souhrn dat pro podporu rozhodování v managementu (Inmon, 2002). Pro lepší pochopení datového skladu jsou pojmy dále interpretované (Lacko, 2003; Pour, Maryška a Novotný, 2012):

- **Subjektově orientovaný** – datový sklad klasifikuje data podle předmětu, a ne podle jejich oboru aplikace. Kategoriemi může být zákazník, výrobek, prodejce nebo obchod.
- **Integrovaný** – sjednocuje data z různých zdrojů a útvarů do jednotné podoby.
- **Časově variabilní** – je schopen ukládat historii a data proměnná v čase za pomocí dimenze času (datumu).
- **Neměnný** – v datovém skladu se nevytvářejí nová data, pouze jsou nahrávány přírůstky z operačního prostředí. Data se nemodifikují ani neodstraňují.

Realizace vyžaduje nemalé finanční prostředky v závislosti na návrhu, hardwaru a softwaru. Návrh přidává datovému skladu největší hodnotu. Jeho struktura by měla vycházet z konkrétního podnikového know-how. Hardware zahrnuje výkonné servery umožňující rychlý přístup k velkému objemu dat. Sklad je vytvořen v prostředí databázových systémů a vyžaduje databázový server (Lacko, 2003).

Datová tržiště

Vytvoření datového skladu je časově náročný proces, který můžeme budovat v malých podmnožinách. Podmnožiny pak nazýváme datová tržiště (*Data Mart*). Tento přístup je vhodný pro minimalizaci rizik a minimalizaci nespokojenosti koncových uživatelů. Datová tržiště řeší požadavky pro konkrétní divize v podniku (Pour, Maryška a Novotný, 2012). Příkladem může být marketing, prodeje nebo lidské zdroje. Tyto části se později spojují do jednoho celku – datového skladu (Lacko, 2003).

Reporting

Úlohou podnikového zpravodajství (*reporting*) je automatická přeměna dat v informace, jejich vizualizace a včasné poskytnutí v podobě zpráv (reportů). Reporty monitorují klíčové podnikové ukazatele a kontrolují plnění podnikových cílů. Výstupy jsou tvořeny v elektronické podobě a můžeme je obecně rozdělit do dvou skupin (Lacko, 2009):

- **Statické** – report lze pouze číst.
- **Interaktivní** – report je možné přizpůsobovat díky ovládacím panelům a interaktivním prvkům.

Data jsou získávána z relačních databází nebo z multidimenzionálních databází za pomocí jazyků SQL, MDX[5] nebo DAX[6].

OLAP analýzy

OLAP analýzy jsou prováděny nad jednou nebo více mezi sebou propojenými OLAP databázemi. Ty v sobě komponují obchodní logiku a předzpracované kalkulace a agregace.

Dolování dat

Dolování dat (*Data Mining*) je speciální případ analýz, kdy mezi daty objevujeme nové vztahy a skutečnosti, které nebyly doposud definovány. Z dat můžeme odvo-

[5]MDX (*Multidimensional Expressions*) je dotazovací jazyk pro extrakci a vizualizaci informací z datové kostky.

[6]DAX (*Data Analysis Expressions*) je skriptovací jazyk, který umožňuje uživateli definovat vlastní kalkulace a metriky. Obsahuje některé vzorce, které jsou součástí programu Excel, k nimž přidává vlastní navržené pro práci s relačními daty.

zovat i prediktivní informace. Pro hledání závislostí jsou používány matematické a statistické metody, jako jsou rozhodovací stromy nebo neuronové sítě.

Pomocí dolování znalostí můžeme analyzovat nákupní košíky, segmentovat zákazníky nebo predikovat vývoj akcií (Pour, Maryška a Novotný, 2012).

3.4 Architektura datových skladů

Datový sklad je možné budovat pomocí dvou přístupů. Jedná se o budování individuálních datových tržišť nebo o vytvoření celopodnikového řešení datového skladu.

Architektura individuálních datových tržišť

První koncept architektury datových tržišť byl navrhnut v 80. letech Ralphem Kimballem. Jednalo se o tvorbu kompletních samostatných datových tržišť pro podniková oddělení včetně ostatních komponent (ETL, OLAP kostky atd.). To způsobovalo nekonzistenci v terminologii a v podnikových datech (Kimball a Ross, 2013). Tento koncept byl pak v následujících letech mírně přepracován a zavedl sběrnicovou architekturu, která integruje podniková datová tržiště za pomocí sdílených dimenzí (Pour, Maryška a Novotný, 2012). Data v tržištích jsou ukládána v dimenzionálním modelu.

Realizace řešení se provádí postupně. Nejprve jsou identifikovány nejdůležitější požadavky podniku a dle toho se vytvoří datové tržiště. To pak slouží pro určení prvních sdílených dimenzí. Další tržiště řešící jiné požadavky již obsahuje kromě vlastních dimenzí i dimenze sdílené (Pour, Maryška a Novotný, 2012).

Budování datových tržišť se sebou přináší výhodu v rychlém uspokojení hlavních požadavků podniku, v jednoduché organizaci a tvorbě dotazů. Nevýhodou může být obtížnější integrace a duplicita obsahu v datových tržištích nebo v ostatních komponentách (Pour, Maryška a Novotný, 2012).

Architektura celopodnikového datového skladu

Druhá architektura nejprve buduje celopodnikový datový sklad a poté jsou realizována příslušná datová tržiště. Stejně jako v případě architektury individuálních datových tržišť lze koncept realizovat buď přírůstkovou metodou, nebo jednorázově.

Při realizaci je nejprve provedena analýza a dokumentace všech požadavků podniku. Vytvoří se orientační časový plán a navrhne datový sklad. Po jeho celkové implementaci pokračuje realizační tým v tvorbě navazujících datových tržišť.

Hlavní výhodou tohoto řešení je velká integrovanost a jednodušší tvorba dílčích tržišť, neobsahujících duplicitní obsah. V celopodnikovém datovém skladu ukládáme data v normalizované podobě, což umožňuje jeho použití i pro jiné typy úloh než jen analytické. Nevýhodou je dlouhá doba vývoje. Podnik může analyzovat svá data až po vybudování celého datového skladu. Tím se požadovaný efekt oddálí a je obtížné sledovat návratnost vynaložených finančních prostředků (Pour, Maryška a Novotný, 2012).

4 Business Intelligence v MS SQL Serveru

Business intelligence bývá realizováno za použití komerčních aplikačních softwarů. Pro výběr správného softwaru by měl být brán v potaz druh podniku, struktura jeho oddělení a zaměstnanců. Následující seznam jmenuje několik nejrozšířenějších aplikací pro BI:

- IBM Cognos Business Intelligence
- Microsoft Business Intelligence
- MicroStrategy
- Oracle Business Intelligence
- Pentaho BI
- QlikView
- SAP Business Intelligence
- Tableau

Jedním z hlavních leaderů aplikací business intelligence je právě firma Microsoft. Jejím stěžejním produktem v této oblasti je SQL Server. Jedná se o databázový systém s komplexní sadou prostředků pro správu dat a jejich analýzu. Aktuální verze MS SQL Server 2016 obsahuje následující sety programů (Ray, Vance a Guyer, 2016):

- **SQL Server Database Engine** – databázový engine[7] zajišťující ukládání, zpracovávání a zabezpečení dat. Obsahuje nástroje pro replikaci dat, fulltextové vyhledávání a umožňuje tvorbu a správu relačních databází.
- **Analysis Services (SSAS)** – set prvků pro tvorbu a správu online analytického zpracování (OLAP) a aplikací dolování dat.
- **Reporting Services (SSRS)** – serverové a klientské prostředky pro realizaci, správu a nasazování tabulek, matic, grafů a vizualizací ve formě reportů. Může se jednat i o platformu umožňující tvorbu reportovacích aplikací.
- **Integration Services (SSIS)** – set grafických nástrojů a programovatelných objektů pro přenos, kopírování a transformaci dat. Často obsahuje komponentu Data Quality Services (DQS), což je doplněk zajišťující datovou kvalitu. V SSIS je možné realizovat ETL v BI procesu.
- **Master Data Services (MDS)** – produkt spravující kmenová data podniků. Může být nakonfigurován tak, aby řídil libovolné domény (produkty, zákazníci,

[7] Databázový engine je modulem softwaru, který používá databázový systém k vytváření, načítání, aktualizaci a mazání dat z databáze.

účty). Obsahuje hierarchie, zabezpečení, transakce, verzování dat a obchodní pravidla.

- **R Services (In–Database)** – služba podporující funkci analýz v databázi, integruje programovací jazyk R s SQL Serverem.
- **SQL Server Data Tools (SSDT)** – zdarma dostupný vývojový nástroj, který umožňuje vytváření relačních databází SQL Serveru, databází Azure SQL, balíčků Integration Services, datových modelů Analysis Services a reportů v Reporting Services.
- **SQL Server Management Studio (SSMS)** – prostředí pro přístup, konfigurování, administraci a vývoj komponent SQL Serveru.

Produkty SQL Serveru obsahují software nezbytný pro tvorbu, nasazení a správu systémů datového skladu nebo business intelligence. Microsoft dále nabízí i několik významných nástrojů, které nejsou součástí SQL Serveru a jsou navržené pro koncové uživatele (Mundy, Thornthwaite a Kimball, 2011):

- **Excel** – jeden z nejběžnějších nástrojů pro přístup k databázím Analysis Services. Ve známém excelovém prostředí se mohou uživatelé přímo připojit k SSAS kostkám nebo k relačnímu datovému skladu.
- **SharePoint** – produkt sloužící jako business intelligence portál, který poskytuje integrované místo pro reporty, dotazovací nástroje, online vzdělávání a dokumentaci.
- **PowerPivot** – funkce kombinující možnosti Analysis Services s programem Excel. Zajišťuje zpracovávání dat v operační paměti (*In-Memory*). Jedná se o populární, analyticky výkonný nástroj poskytující BI aplikační platformu při integraci s SharePointem.
- **PowerPivot pro SharePoint** – rozšiřuje možnosti a práci s PowerPivot. Koncoví uživatelé mohou jednoduše sdílet Excel sešity používající PowerPivot v SharePointu.
- **Master Data Services** – kromě SQL Serveru může být konfigurován a integrován do SharePointu.
- **Power BI** – aplikace obsahující set nástrojů pro tvorbu datových modelů, interaktivních repotů a dashboardů.

Mezi produkty pro business intelligence od firmy Microsoft patří i služba Azure.

4.1 Analysis Services

Analytické databáze OLAP jsou vytvářeny v nástroji SQL Server Analysis Services. Aplikace obsahuje prvky pro vytvoření sémantické vrstvy (vrstvy metadat) nad

relačním datovým skladem. Tato vrstva je optimalizována pro dotazování a kalkulace z oblasti business intelligence. Poskytuje analytické údaje klientským aplikacím pro vizualizaci a analýzu dat, a tím podporuje podnikové rozhodování (Duncan a Guyer, 2016).

Analytické databáze jsou tvořeny ve formě modelů, které obsahují metadata o způsobu měření a agregacích propojených faktových a dimenzionálních tabulek. Nástroje SSDT podporují tvorbu hierarchií a kalkulací pro koncové uživatele. Modely v sobě mají zakomponovánu logiku daného podniku, která snižuje riziko uživatelských chyb při dotazování. Tím jsou zobrazovány pouze smysluplné výstupy. Databázová vrstva zahrnuje centralizované údaje, které jsou jednoduše přístupné všem potřebným pracovníkům.

Použití analytických databází OLAP zjednodušuje práci IT oddělení, které nemusí samostatně vytvářet složité reporty definované požadavky pracovníků s rozhodovací pravomocí. Jednoduché ovládání modelu poskytuje managementu nástroje pro vlastní tvorbu vizualizací a reportů bez nutnosti znalostí technických detailů (Russo, Ferrari a Webb, 2012).

Analysis Services podporuje tvorbu dvou typů datových modelů, multidimenzionálního nebo tabulárního. Proces realizace analytické vrstvy začíná vytvořením jednoho ze dvou modelů a jeho nasazení na instanci SQL Serveru Analysis Services nebo na Azure Analysis Services. Dále je třeba nastavit opakované zpracování dat a definovat uživatelské role a zabezpečení služby. Poté již můžeme modely zpřístupnit koncovým uživatelům a aplikacím, které je podporují (Duncan a Guyer, 2016).

Instalace Analysis Services umožňuje při konfiguraci výběr módu serverové instance, kde každá z instancí nabízí jiné služby pro konkrétní analytické řešení. Jedná se o následující módy:

- **Tabularní mód** – používá relační in-memory prvky (modely, tabulky, sloupce, metriky a hierarchie).
- **Multidimenzionální a data miningový mód** – používá prvky OLAP modelování (kostky, dimenze, metriky).
- **PowerPivot mód** – používá datové modely Excelu a PowerPivot ve službě SharePoint.

Každá instance může být nakonfigurována pouze v jednom módu, který nelze změnit. Nicméně je možné instalovat více instancí v jiných módech na stejném serveru.

Nástroj SQL Server Data Tools umožňuje výběr šablon pro tabulární, multidimenzionální nebo data miningový model. Šablony obsahují složky všech potřebných objektů modelu. Po nasazení databázového modelu se využívá aplikace SQL Server Management Studio, která se používá ke konfiguraci datového zpracování a monitorování serveru a databází (Duncan a Guyer, 2016).

Vývoj Analysis Services

Společnost Microsoft zahájila svůj vstup do oblasti business intelligence v roce 1996. Toho roku zakoupila technologii OLAP od kanadské společnosti Panorama Software (Microsoft: News Center, 1996). První verze Analysis Services byla vydána v roce 1998 pod názvem OLAP Services jako součást SQL Serveru 7. Služba podporovala multidimenzionální architektury MOLAP, ROLAP a HOLAP. Pro dotazování používala jazyk MDX (Microsoft: TechNet, 2001).

V roce 2000 byla zveřejněna nová verze produktu již pod názvem Analysis Services. Obsahovala nové nástroje pro data mining, propracovanější tvorbu dimenzí a podporu unárních operátorů (Sharma, 2002). S tímto produktem se Microsoft stal jednou z hlavních firem ovlivňující trh s business intelligence (Russo, Ferrari a Webb, 2012).

Verze Analysis Services 2005 představila Unifikovaný Dimenzionální Model (*Unified Dimensinal Model – UDM*), který vytváří most mezi klientskými aplikacemi a datovými zdroji. UDM je vytvořeno na jednom nebo více datových zdrojích a koncový uživatel používá nástroje klienta pro dotazování (Melomed, 2007). Tato verze se stala jednou z nejprodávanějších mezi BI nástroji.

V roce 2008 byla vydána verze Analysis Services 2008, která se soustředila na zlepšení výkonu a rychlejší dotazování. Tato verze byla variabilnější v rámci rozdílných uživatelských řešení a představila první koncept tabulárního modelu jako součást doplňku PowerPivot – Analysis Services do programu Excel. PowerPivot používal analytické jádro VertiPaq, které komprimovalo tabulky in-memory, tedy do operační paměti počítače.

O čtyři roky později došlo ve verzi Analysis Services 2012 k přejmenování technologie VertiPaq na xVelocity engine. Toto nové jádro bylo ve verzi xVelocity In-memory analytics engine zakomponováno do nově vzniklého tabulárního modelu a do tabulek PowerPivot.

Aktuální verzí je SQL Server 2016. Analytické služby se dočkaly optimalizovanějšího výkonu, snadnější tvorby řešení, automatizované správy, vylepšení relací pomocí obousměrného křížového filtrování, paralelního zpracování oddílů (*partitions*) a dalších vylepšení (Duncan a Guyer, 2016).

Tabulární model

Tabulární model je novější typ analytických databází kombinující technologii MOLAP a relačních databází. Relační databáze ukládají data do tabulek a jejich nevýhodou je nemožnost modelování klíčových podnikových ukazatelů (*Key performance indicator – KPI*) nebo podnikových metrik. Toto modelování poskytuje právě MOLAP, ale jeho problémem je pomalý přístup k datům. Jejich kombinace je v modelu efektivní a jednoduchá.

Modely mohou využívat rozdílné enginy pro ukládání dat, jedná se o In-memory nebo DirectQuery. Nejvyšším objektem modelu je právě databáze obsahující tabulky

dat, založená na relačním principu. Tabulární databáze jsou vytvářeny nebo upravovány pomocí generovaných příkazů aplikace SSDT (Gandhi, 2016). Jedna instance analytického serveru může obsahovat více databází, a každá databáze obsahuje vlastní skupiny objektů řešící konkrétní podnikové požadavky. Tabulka má pevný počet sloupců definovaných při návrhu modelu a libovolný počet řádků závisející na množství načtených záznamů. Každý sloupec tabulky má jeden fixní datový typ, takže obsahuje například pouze číselné hodnoty do určitého rozsahu nebo pouze textové hodnoty. Data jsou načítána z jedné tabulky v relačním zdroji, která vznikla jako výsledek SQL dotazu. Ve fázi zpracování neboli anglicky *processing* dochází k nahrávání dat do analytické databáze (Russo, 2016).

Relace v tabulárním modelu propojují data mezi dvěma tabulkami a určují, jaký vztah mezi sebou mají. Pokud do modelu importujeme tabulky z datového zdroje, mohou být automaticky vytvořeny i relace mezi nimi. Vztahy mezi tabulkami můžeme vytvořit i manuálně. Po jejich nadefinování lze vyhledávat hodnoty v propojených tabulkách a filtrovat data za použití propojených sloupců. Nástroje modelu podporují pouze relace 1:1 nebo 1:M. Pro vytvoření vztahu M:N je třeba použít DAX funkce nebo **obousměrné křížové filtry** (*bi-directional cross filters*), které jsou novou vestavěnou funkcí v SQL Serveru 2016. Eliminují potřebu psaní DAX výrazů rozšiřujících obsah filtrů pro relace mezi tabulkami. Křížové filtrování je schopnost nastavení kontextového filtru pro tabulku na základě hodnot v související tabulce. Pojem obousměrné znamená, že převádějí filtr na druhou související tabulku na opačné straně relace. Tyto filtry umožňují procházení dat v obou směrech relace a rozšiřují kontext filtru pro dotazování na nadmnožinu dat. Existence více relací mezi dvěma tabulkami může vést k nejednoznačným závislostem. Proto je mezi každou dvojicí pouze jedna aktivní relace. Počet neaktivních relací není omezen a jejich použití je dále specifikováno klientem při dotazování. Zdrojový sloupec nemůže být použit pro více relací. Pokud už byl použit pro propojení k jinému souvisejícímu sloupci v jiné tabulce, musí být pro definování nové relace vytvořena kopie sloupce. V tabulárním modelu není možné používat složené klíče, vždy musí existovat sloupec s jedinečným identifikátorem. Tabulární model nepodporuje rekurzivní vztahy tabulek.

Definování dotazů a kalkulací modelu je prováděno v jazyku DAX, který vznikl právě pro tabulární modely, PowerPivot tabulky a pro Power BI. Klientské aplikace procházejí modely za použití jazyku MDX a DAX. K tabulárnímu modelu tak mohou přistupovat i aplikace vzniklé dříve pouze pro multidimenzionální model. Kalkulace jsou odvozovány již z existujících sloupců. V modelu přestavují sloupce nové tzv. *calculated columns*. Kromě odvozených sloupců je možné vytvořit i odvozené tabulky, *calculated tables*, do kterých jsou na základě DAX dotazu nahrána data z jiných tabulek stejné databáze. Modely dále podporují tvorbu metrik (*measures*). Stejně jako v případě kalkulací, jsou definovány výrazy jazyka DAX. Jedná se o seskupené hodnoty dat jednoho nebo více sloupců tabulky. Příkladem může být součet všech hodnot v daném sloupci (Russo, 2016). Klíčové podnikové ukazatele jsou v tabulárním modelu používány pro měření výkonu podnikových hodnot, který

je definovaný základními metrikami. Úkolem těchto ukazatelů je zjistit, zda se daná hodnota blíží k požadované cílové hodnotě, která je buď také definována metrikami, nebo absolutní hodnotou.

Nástroje umožňují tvorbu hierarchií. V tomto případě se jedná o metadata, která určují vztahy mezi dvěma nebo více sloupci v tabulce. V uživatelských reportech se mohou objevovat zvlášť od ostatních sloupců, což usnadňuje navigaci drill down a drill up pro koncové uživatele.

Další funkcí modelu je rozdělení tabulky na logické části neboli *partitions*. Každý oddíl pak může být aktualizován nezávisle na ostatních částech. Partitions se v SSDT během vytváření aplikují na modelovanou databázi. Po nasazení modelu jsou duplikovány a také nasazeny do databáze. Části nasazené databáze je dále možné vytvářet a spravovat pomocí dialogového okna Partitions v aplikaci SSMS.

Databáze tabulárního modelu mohou být pro koncové uživatele velice komplexní. Jeden model může zahrnovat kontext celého datového skladu obsažený v mnoha tabulkách, metrikách a dimenzích. To je pro uživatele vyžadující pouze malou část modelu matoucí. Pro tyto případy jsou k dispozici perspektivy neboli viditelné podmnožiny modelu, které specifikují části pro určitou aplikaci nebo podnik. V perspektivách jsou tabulky, sloupce, metriky a KPI definovány jako objekty. Pro každou perspektivu lze vybrat její viditelnou oblast. Příkladem může být model obsahující produktové, prodejní, finanční, zaměstnanecké a geografické údaje. Jedna perspektiva pro obchodní oddělení může obsahovat pouze údaje o produktech, prodejích, propagacích a geografii. Naopak lidské zdroje mohou využívat pouze údaje o zaměstnancích a prodejích v jiné perspektivě. Když se uživatelé připojí k modelu s perspektivami, zvolí si tu, kterou chtějí použít. Perspektivy nejsou určeny jako bezpečnostní mechanismus, ale pouze jako nástroj pro zlepšení práce koncových uživatelů. Bezpečnost dat v perspektivě vychází ze základního modelu.

Přístupy uživatelů k datům jsou zajištěny pomocí rolí. Ty definují oprávnění uživatelů modelu. Členové rolí mohou provádět akce podle jejich povolení. Role, které jsou oprávněné pouze ke čtení dat, mohou být pomocí filtrů specifikovány až na úroveň řádků. V Analysis Services existují dva typy rolí (Duncan a Guyer, 2016):

- **Serverová role** – pevná role, která poskytuje administrátorovi přístup k serverové instanci Analysis Services.
- **Databázová role** – role definovaná autory modelu a administrátory. Slouží pro kontrolování přístupů uživatelů k modelu databáze a datům.

Role definované v tabulárním modelu jsou databázové role. Obsahují členy mající specifická oprávnění, umožňující provádět dané operace. Databázová role je definována jako samostatný objekt vztahující se pouze na databázi, ve které byla vytvořena. Členy můžeme definovat jako jednotlivé uživatele nebo skupiny uživatelů. Jejich definici provádí autor modelu, který má ve výchozím nastavení administrátorské oprávnění. Řádkové filtry jsou vytvářeny pomocí DAX formulí. Definují, na které řádky a v jakých směrech se mohou uživatelé dotazovat. Filtry mohou být použity

pouze pro role čtení nebo čtení a zpracování. Výchozí nastavení nového projektu tabulárního modelu neobsahuje role žádné.

Při práci s modely se můžeme setkat s pojmem level kompatibility. Ten odkazuje na specifikace vydání Analysis Services. V praxi dochází k rozdílné implementaci DirectQuery, objektů a metadat modelu na základě jejich levelu kompatibility. Levely 1200 a 1400 (dostupný od verze SQL Server 2017) jsou aktuálně jediné podporovány serverem Azure Analysis Services (Duncan a Guyer, 2016).

Multidimenzionální model

Multidimenzionální model využívá kostkovou strukturu pro analyzování dat skrze více dimenzí. Jedná se o výchozí mód, jehož modely jsou kompatibilní s řadou BI nástrojů jako je Excel, Reporting Services nebo aplikacemi třetích stran. Modely ukládají data do MOLAP, ROLAP nebo HOLAP a podporují dotazování a kalkulace (Russo, Ferrari a Webb, 2012). Jejich výhoda je v rychlém výkonu pro Ad-Hoc[8] dotazy.

Databáze multidimenzionálního modelu jsou tvořeny z následujících setů objektů:

- Datové zdroje (*Data Sources*)
- Pohledy na datové zdroje (*Data Source Views*)
- Kostky (*Cubes*)
- Dimenze (*Dimensions*)
- Struktury dolování dat (*Mining Structures*)
- Role (*Roles*)
- Sestavy (*Assemblies*)
- Smíšené objekty (*Miscellaneous*)

Všechna data, která jsou načítána do modelu, pocházejí z datových zdrojů. Typickým zdrojem je datový sklad, ale může to být i relační databáze. Multidimenzionální model obsahuje objekt datového zdroje, který v Analysis Services specifikuje připojení na externí datový zdroj. Obsahuje informace o pověření, fyzickém umístění dat a jeho poskytovateli. Pohledy na datové zdroje obsahují relační schémata oddělená od zdroje dat, která jsou základem pro tvorbu kostek a dimenzí. Jedná se o prostor pro manipulaci s datovou strukturou projektu nezávisle na podkladových datech. Dimenze jsou skupiny příbuzných atributů poskytující informace o faktových tabulkách v kostkách. Uživatelé uspořádávají dimenze do manuálně definovaných hierar-

[8]Dotaz Ad-Hoc se vytváří pouze v případě potřeby. Skládá se z dynamických SQL a je obvykle konstruován pro jednorázové účely. Je to protiklad jakýchkoli dotazů, jenž jsou předem definovány a rutinně prováděny.

chií. Kostky představují vícerozměrné struktury obsahující informace pro analytické účely. Hlavními jejími prvky jsou dimenze a metriky. Dimenze definují strukturu kostky, umožňují její procházení a agregování. Logická struktura kostky určuje, jak bude klient data prohlížet. Agregované hodnoty jsou uložené v buňkách kostky na průsečících dimenzí. Jedna databáze může obsahovat více kostek používajících ostatní objekty modelu v jiném kontextu. Struktura rolí poskytuje, podobně jako v případě tabulárního modelu, autorizaci uživatelů, ověření přístupů k operacím, objektům a datům.

Multidimenzionální model rovněž podporuje tvorbu metrik a jejich skupin, kalkulací, partitions, perspektiv, KPI a jazykových překladů.

4.2 Microsoft Azure

Microsoft Azure je cloudové řešení pro tvorbu, nasazování a správu aplikací přes globální síť datových center. Azure nabízí širokou nabídku produktů a služeb pro firmy. Jedná se například o Office 365, Dynamics CRM Online, nebo SharePoint Online. Jednou z oblastí Azure jsou i data. Nabízí nástroje pro business intelligence, datové sklady a datovou analýzu. Azure obsahuje následující služby související s business intelligence (Microsoft Azure, 2017):

- **Power BI Embedded** – produkt integruje Power BI reporty do webových nebo mobilních aplikací.
- **Azure SQL Data Warehouse** – cloudový datový sklad se škálovatelnými výpočetními prostředky. Používá architekturu masivního paralelního zpracování (*MPP – Massively Parallerl Processing*)[9].
- **Azure Analysis Services** (Azure AS) – cloudová verze Analysis Services podporující tabulární modely.

Azure Analysis Services

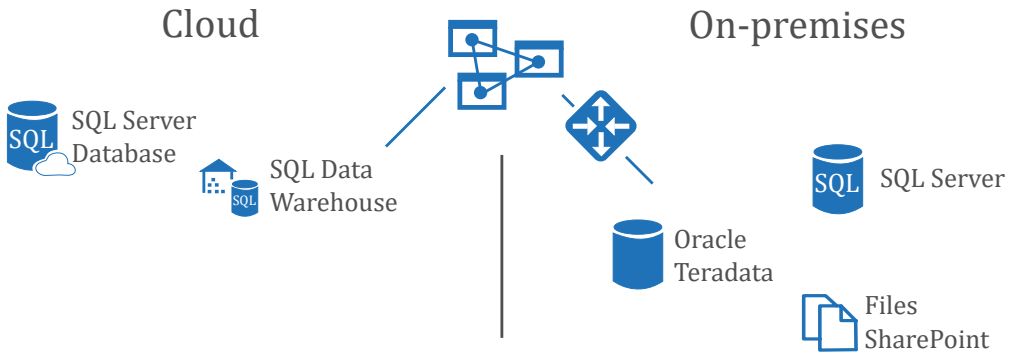
Jedná se o službu Analysis Services běžící v rámci cloudového řešení Azure, která je dostupná v plné verzi od 19. dubna 2017. Umožnuje vytvářet instanci Azure Analysis Services (Azure AS) a rychle ji nasazovat. Uživatelé platí pouze za dobu, kdy je daná instance aktivní. Při pozastavení služby nejsou účtovány žádné poplatky, ani když jsou do ní stále načítaná data (Russo, 2016).

Azure Analysis Services podporuje tabulární modely na levelu kompatibility 1200 a výše. Modely vytvářené pro Azure AS používají stejné nástroje jako SSAS. Jejich vytvoření a nasazení je prováděno v aplikaci SQL Server Data Tools nebo za

[9]Více informací o MPP lze nalézt v publikaci *Introducing Microsoft SQL Serve 2016* v kapitole Hyperscale Cloud (Varga, Cherry a D'antoni, 2016).

použití šablon Azure PowerShell[10] a Azure Resource Manager[11] v SQL Server Management Studio (Duncan a Wheeler, 2017). Jelikož služba běží na cloudu, umožňuje připojení odkudkoliv. Stejně tak podporuje připojení pro klientské aplikace, Power BI nebo Excel.

Datové modely jsou nasazovány na servery podporující dva typy datových zdrojů. Jedná se o zdroje přímo v podniku (neboli on-premises) nebo v cloudu. Oba zdroje můžeme zkombinovat a vytvořit tak hybridní BI řešení podobné tomu, které je zobrazeno na obrázku č. 5. Připojení k datovému zdroji v cloudu by mělo být bez komplikací, ale připojení k on-premises zdrojům vyžaduje *on-premises data gateway* chovající se jako speciální most, instalovaný na počítače v síti. Zajišťuje rychlost a bezpečnost spojení s cloudovou verzí Analysis Services.



Obrázek 5: Způsoby hybridního připojení na analytický server Azure (Duncan a Wheeler, 2017)

Zabezpečení přístupů analytických databází v Azure je vyřešeno pomocí služby Azure Active Directory (AAD). Uživatelé se do služby přihlašují pomocí podnikových účtů. Služba si zkontroluje, zda daný uživatel pokoušející se o přístup k databázím má požadovaná oprávnění. Podnikové účty musejí být členy AAD pro předplatné sdílené analytické databáze. Uživatelé mohou získat přístup propojením adresářů AAD s lokální Active Directory podniku. Přihlášení do služby je zprostředkováno hlavním uživatelským jménem (UPN) a heslem. Pokud dochází k synchronizaci s lokálním podnikovým serverem Active Directory, bývá přihlašovacím jménem podniková emailová adresa uživatele. Oprávnění pro správu všech zdrojů Azure AS jsou přidělena pomocí rolí v předplatném. Administrátorské role mají jako výchozí nastavení veškeré oprávnění vlastníka serveru. Přidávání dalších uživatelů může být provedeno pomocí Azure Resource Manager.

Služba pro opakované ukládání využívá uložiště Azure Blob[12]. Jeho datové soubory jsou zašifrovány pomocí šifrování Azure Blob Server Side Encryption (SSE).

[10] Azure PowerShell nabízí sadu odlehčených příkazů, které používají model Azure Resource Manager pro správu Azure zdrojů.

[11] Azure Resource Manager poskytuje nástroje pro skupinovou správu zdrojů daného řešení.

[12] Azure Blob zajišťuje uchovávání dat v nestrukturované podobě na cloudu ve formě objektů nebo objektů blob. Produkt podporuje uložení libovolného textu nebo dat v binární struktuře jako jsou dokumenty nebo mediální soubory.

Při použití DirectQuery jsou ukládána pouze metadata.

Zabezpečený přístup k datům uloženým v podniku zajišťuje instalace a konfigurace výše uvedené on-premises data gateway. Jedná se o bránu poskytující přístup pro DirectQuery a In-memory enginy. Když se cloudové analytické modely Azure připojí na on-premises zdroj dat, zároveň se vygeneruje dotaz se zašifrovanými pověřeními pro datový zdroj. Služba cloudové brány analyzuje dotaz a zašle ho jako požadavek na sběrnici Azure. On-premises data gateway zkontroluje nevyřízené požadavky na sběrnici Azure. Obdrží dotaz, dešifruje pověření, připojí se k datovému zdroji a realizuje dotaz. Výsledky jsou ze zdroje odeslány zpět do brány a pak do analytické databáze Azure (Duncan a Wheeler, 2017).

Cenové úrovně

Pro použití služby je třeba mít aktivní předplatné. Microsoft nabízí jeho bezplatnou zkušební verzi Azure poskytující novým zákazníkům kredit ve výši € 170. Ten lze po dobu jednoho měsíce využít na libovolné služby v rámci Azure.

Microsoft definuje jednotku nazývanou QPU, která označuje sílu zpracování instance Azure Analysis Services. Tato jednotka zjednodušuje porovnávání jednotlivých úrovní služby, ale složitěji se porovnává s výkonem hardwaru. Podle prohlášení Microsoftu odpovídá 20 jednotek QPU zhruba jednomu jádru.

Celková cena cloudového analytického serveru se odvíjí od zvolené úrovně a instance produktu. Aktuálně jsou k dispozici tři úrovně: Developer, Basic a Standard. Verze Developer a Standard nabízí následující funkce:

- Perspektivy
- Oddíly
- DirectQuery
- Překlady
- Výpočty DAX
- Zabezpečení na úrovni řádku
- Ukládání dat do paměti
- Zálohování a obnovení

Úroveň Basic také podporuje výše uvedené funkce, kromě perspektiv, oddílů a DirectQuery. Jednotlivé úrovně mohou nabízet více instancí lišících se výpočetním výkonem, velikostí paměti a jednotkami QPU. Ceny uvedené v následující tabulce č. 1 nabývají své platnosti od 1. června 2017.

Úroveň Developer je doporučena pro účely zhodnocování, vývoj a testování. Nabízí nižší počet jednotek QPU a menší velikosti paměti.

Tabulka 1: Cenové úrovně Azure Analysis Services pro Západní Evropu

Instance	**Jednotky QPU**	**Paměť (GB)**	**Cena/měsíc**
Developer	20	3	€ 82,82
Basic 1	40	10	€ 269,79
Basic 2	80	20	€ 539,58
Standard 0	40	10	€ 759,18
Standard 1	100	25	€ 1 273,66
Standard 2	200	50	€ 2 547,31
Standard 4	400	100	€ 5 088,34

Úroveň Basic lze využít pro malá produkční řešení s jednodušším datovým modelem, nižším souběžným přístupem více uživatelů a s nižšími nároky na obnovování dat.

Úroveň Standard nabízí flexibilní přístup více uživatelů k jednomu serveru, podporuje složitější datové modely s rychlým nárůstem nových dat, obsahuje detailnější a pokročilejší nastavení obnovy dat umožňující téměř okamžitý přístup k aktuálním datům. Úroveň je obecně doporučena pro složitější klientské aplikace (Microsoft Azure, 2017).

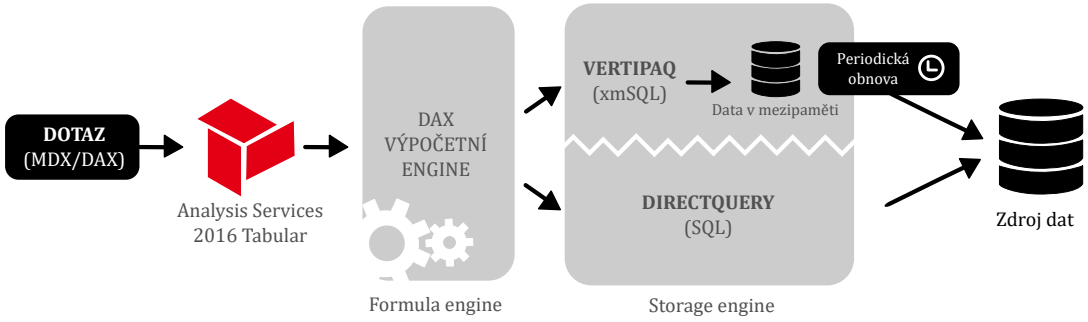
5 In-memory technologie

In-memory technologie jsou typem databázových systémů, které ukládají veškerá data přímo v hlavní paměti. To vede k mnohonásobnému zrychlení oproti systémům ukládajícím data na disky. Se zrychlením souvisí také snížení provozních nákladů, což obecně zvyšuje zájem o in-memory. Přidaná hodnota dat se postupem času snižuje, čím rychleji je máme k dispozici, tím jsou cennější. Proto je důležitá rychlost zpracovávání umožňující okamžité použití informací v reálném čase. Databáze in-memory umožňují OLAP analýzu v reálném čase na základě dat z online zpracování transakcí (Lake a Crowther, 2013).

Jedním z programů používající in-memory technologie je i SQL Server Analaysis Services a Azure Analysis Services od společnosti Microsoft.

5.1 Výpočetní enginy tabulárního modelu

Každý dotaz na tabulární databází Analysis Services ve verzi 2016 je zpracován dvěma vrstvami výpočetních enginů. Analysis Services parsuje DAX a MDX dotazy, transformuje je v dotazovací plán zpracovaný tzv. formula enginem, který dokáže pracovat s jakoukoliv funkcí nebo operací těchto dvou jazyků. Pro získání surových dat nebo provedení kalkulací musí formula engine provést zavolání na storage engine (ukládací engine). Tabulární model rozlišuje dva typy storage enginů. Jedná se o in-memory analytický engine (dále označovaný původním názvem VertiPaq), nebo o DirectQuery engine. Při tvorbě modelu je možné definovat, jaký engine se bude používat. Jak ukazuje obrázek č. 6, VertiPaq obsahuje kopii dat, která jsou nahrávána z datového zdroje při každém obnovení datového modelu. Své požadavky na data pak obdrží v interní binární struktuře, která je externě popsána ve formátu xmSQL. DirectQuery oproti tomu komunikuje s externí zdrojovou relační databází, získává z ní data a snižuje časovou prodlevu mezi aktualizacemi zdroje a dostupností aktualizovaných dat v Analysis Services. DirectQuery požaduje data za použití jazyka SQL, který je datovými zdroji podporován (Russo, 2016).



Obrázek 6: Formula engine a storage engine v tabulárním modelu (Russo, 2016)

VertiPaq

Jedná se o in-memory sloupcovou databázi. In-memory ukládá veškerá data, se kterými model manipuluje, do paměti RAM. Sloupcová databáze organizuje data do samostatných sloupcových struktur, optimalizovaných pro vertikální (sloupcové) prohledávání. V případě horizontálního (řádkového) prohledávání napříč všemi sloupci vyžadují sloupcové databáze naopak více výpočetních prostředků. VertiPaq nepodporuje další struktury pro optimalizace, jako jsou například indexy relačních databází. Data jsou v paměti komprimovaná algoritmy, které umožňují rychlé prohledávací operace, snižují dobu přístupu k datům a velikost dat v paměti.

VertiPaq engine je částí zpracování dotazu, kde jsou poskytovány výsledky na DAX a MDX výrazy. Jedná se o jediný storage engine, který má fyzický přístup ke komprimovaným datům a provádí jednoduché agregace, filtrování a spojení mezi tabulkami. Složitější kalkulace vyjádřené v DAX nebo MDX jsou zpracovávány formula enginem, který okamžitě obdrží výsledky od storage enginu (VertiPaq nebo DirectQuery) a provede další kroky pro dokončení kalkulace. Formula engine je částí, která při použití VertiPaqu zpomaluje dotazy. To je zapříčiněno tím, že engine zpracovává dotaz v jediném vlákně (v případě potřeby zpracovává požadavky od různých uživatelů paralelně). Oproti tomu VertiPaq může používat více vláken, pokud je databáze dostatečně velká (často vyžaduje minimálně 16 milionů řádků v tabulce, ale konkrétní číslo záleží na velikosti segmentů použitých během processingu).

Ukládání dat je ve VertiPaqu založeno na algoritmech hash kódování, kódování hodnot a komprimaci RLE. Každá hodnota v sloupci je mapována do 32bitové celočíselné hodnoty. Kódování hodnot používá dynamické aritmetické výpočty pro převedení reálné hodnoty na celočíselné a naopak. Hash kódování naopak vkládá nové hodnoty do hashovací tabulky. Celočíselná 32bitová hodnota je před uložením do sloupců komprimovaná. Algoritmus RLE nejprve seřadí data tak, aby následující řádky měly stejnou hodnotu ve sloupci, tím dosáhne lepšího kompresního poměru. Po komprimaci je pak místo stále stejných hodnot uložen pouze počet jejich opakování (Russo, 2016).

DirectQuery

In-memory analytický engine (VertiPaq) je při vytvoření nového tabulárního modelu ve výchozím nastavení. Nicméně existuje i druhá možnost DirectQuery, která neukládá data do mezipaměti. Jedná se o rozdílný přístup, který převádí dotazy provedené nad tabulárním modelem na SQL dotazy dále zasílané na datový zdroj.

Hlavní výhodou DirectQuery je, že se vždy dotazujeme na aktuální data. Protože v tomto případě Analysis Services neukládá kopii dat do mezipaměti, může být velikost databáze větší než velikost paměti serveru. Výkon DirectQuery závisí na výkonu a optimalizaci relační databáze používané jako datový zdroj. Nicméně původní relační databáze nikdy neposkytuje lepší výkon než správně optimalizovaný analytický server pro tabulární model. Tento engine je vhodné používat pro malé

databáze, které jsou častěji aktualizované nebo pro velké databáze, které nemohou být uchovávány v paměti.

DirectQuery podporuje jako zdroje pouze následující relační databáze:

- Microsoft SQL Server (od verze 2008),
- Microsoft Azure SQL Database,
- Microsoft Azure SQL Data Warehouse,
- Microsoft Analytics Platform Systém (APS),
- Oracle (od verze 9i),
- Teradata (od verze V2R6).

Ostatní jsou podporované pouze částečně a Microsoft pracuje na jejich větší kompatibilitě.

Další nevýhodou enginu může být nedostatečná podpora všech možností tabulárního modelu nebo příkazů MDX či DAX, například nepodporuje *calculated columns*. Z pohledu jazyka DAX může dojít k použití funkcí, které mají po převedení na SQL výraz jiný význam, a tím dochází ke špatné interpretaci původního dotazu. Používání jazyka MDX má řadu omezení, která ovlivňují pouze styl kódování. Není možné používat relativní názvy nebo n-tice s členy z různých úrovní MDX výběru. Nicméně existuje i omezení při návrhu datového modelu. V MDX dotazu není možné vytvářet reference uživatelsky definovaných hierarchií, a to ovlivňuje jeho použití v programu Excel (Russo, 2016).

6 Databáze AdventureWorks

AdventureWorks je demonstrační databáze od Microsoftu vytvořena z dat fiktivní firmy Adventure Works Cycle. Jedná se o velkou mezinárodní společnost zabývající se výrobou a prodejem cyklistických součástek a jízdních kol. Jejich prodeje jsou uskutečňovány v Severní Americe, v Evropě a v Asii. Základna firmy se nachází ve městě Washington s 290 zaměstnanci, dále má několik regionálních prodejních týmů. V roce 2000 společnost zakoupila malou výrobní pobočku Importadores Neptuno v Mexiku. Zde jsou vyráběny některé dílčí součástky pro produkty firmy, které jsou montovány v Bothellu. Od roku 2001 je pobočka v Mexiku jediným výrobcem a distributorem skupiny turistických jízdních kol.

Data společnosti jsou dostupná ve třech verzích databáze. Jedná se o OLTP neboli zdrojovou databázi, datový sklad a online analytické databáze. Pro realizaci analytických modelů a porovnávání jejich vývoje využijeme databázi datového skladu (*AdventureWorksDW*) ve verzi 2014. Ta obsahuje jak podmnožinu tabulek ze zdrojové databáze, tak nové finanční informace získané z jiného datového zdroje. Databáze odpovídá vločkovému schématu a je tvořena z 16 dimenzionálních tabulek a 12 faktových tabulek. Detailní informace o všech tabulkách a schématu jsou k dispozici v dokumentaci Elasoft (Database reference, 2010). Databáze se stahuje ve formě zálohovaného souboru (Adventure Works, 2015), ten je třeba vložit do adresáře, kde je nainstalován SQL Server. Instalování je pak provedeno pomocí nástrojů SQL management studia nebo pomocí obnovovacího skriptu.

Data lze rozdělit na dvě předmětové skupiny – finance a prodeje. Finance zahrnují dvě následující podskupiny:

- **Finance** – finanční data v původní měně pro Adventure Works a jeho dceřiné společnosti.
- **Měnová sazba** – údaje o konverzích měn, denní průměrné sazby a kurzy amerického dolaru k jednotlivým měnám.

Prodeje jsou dále rozděleny na čtyři následující schémata:

- **Prodeje od obchodníků** – základem schématu jsou tabulky obsahující prodeje od regionálních obchodníků a odeslané objednávky. Číselné údaje v tabulkách jsou převedeny na americké dolary, ale zaznamenávají původní měnu.
- **Internetové prodeje** – tabulky obsahují podrobná data o prodejích na internetu a o individuálních zákaznících. Stejně jako tabulky prodejů od obchodníků i internetové prodeje zaznamenávají pouze odeslané objednávky a obsahují údaje v amerických dolarech se sledováním původní měny.
- **Přehledy prodejů** – souhrnné přehledy o prodejích obchodníků a internetových prodejích. Snižují počet dimenzí pro srovnání všech prodejů.

- **Prodejní kvóty** – data zahrnují prodejní kvóty pro obchodníky, které jsou odvozovány od marketingových plánů firmy.

Pro praktické porovnávání budou realizovány analytické databáze na základě internetových prodejů a prodejů od obchodníků. Tato schémata byla vybrána z důvodu velkého počtu dostupných zdrojů, snadnějšímu porozumění dat a možnému budoucímu využití pro scénáře dolování dat. Analytické modely budou vytvořeny z tabulek, jejichž název a kontext je zobrazen v tabulce č. 2.

Tabulka 2: Seznam využívaných tabulek a jejich obsah

Název tabulky	**Příklad obsahu tabulky**
DimCurrency	jméno měny, kód měny
DimCustomer	detailní data zákazníků, příjmy, vzdělání, povolání
DimDate	kalendářní údaje od roku 2005 do roku 2014
DimEmployee	jména zaměstnanců, pracovní pozice, oddělení, mzda
DimGeography	města, provincie, regiony, země
DimProduct	jméno produktu, kód, barva, velikost
DimProductCategory	vícejazyčné kategorie produktů
DimProductSubcategory	vícejazyčné podkategorie produktů
DimPromotion	název zvýhodnění, typ zvýhodnění, procentuální sleva
DimReseller	obchodníci, telefonní čísla, počty objednávek
FactInternetSales	číslo objednávky, objemy internetových prodejů
FactResellerSales	číslo objednávky, objemy prodejů obchodníků

7 Realizace analytických modelů

Před realizací datových modelů bylo třeba nainstalovat a zprovoznit analytické servery. Celkem se jedná o tři samostatné instance serverů:

- MS SQL Server 2016 – multidimenzionální mód
- MS SQL Server 2016 – tabulární mód
- Azure Analysis Services

První dva typy MS SQL Analysis Services serverů byly naistalovány na lokální počítač za využití studentské licence programu Microsoft Imagine. Jejich instalace byla provedena samostatně s rozdílným nastavením módů. Vytvoření cloudového serveru Azure popisuje podkapitola Nasazení na analytický server Azure, která je součástí realizace tabulárního modelu.

7.1 Multidimenzionální model

Tvorba multidimenzionálního modelu (příloha A1) začíná vytvořením nového projektu v aplikaci SQL Server Data Tools. Zde je třeba vybrat projekt typu business intelligence a šablonu Analysis Services. Nový projekt byl pojmenován a uložen na požadované místo. Aplikace vytvoří základní strukturu projektu složeného z dílčích objektů popsaných v podkapitole věnované multidimenzionálnímu modelu.

Definování datového zdroje

Prvním krokem je definování datového zdroje, ze kterého data vycházejí. V tomto případě se jednalo o databázi datového skladu AdventureWorksDW2014 uloženou na lokálním MS SQL Serveru. Připojení se realizovalo pomocí průvodce, do kterého bylo potřeba zadat název serveru a jméno požadované databáze i s přihlašovacími údaji. Dále bylo možné změnit název nového datového zdroje a průvodce následně zobrazil vygenerovaný propojovací řetězec. V dalším kroku již došlo k vytvoření datového zdroje a jeho zobrazení do příslušné složky *Data Sources*.

Definování pohledů na datové zdroje

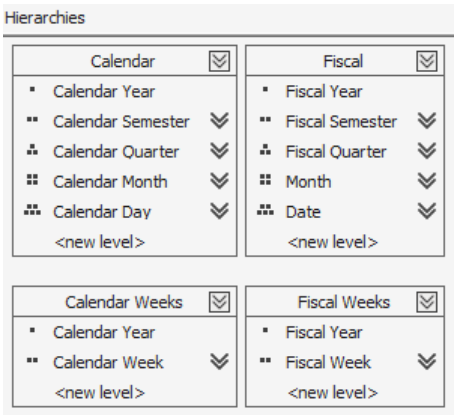
Pohledy na datové zdroje jsou taktéž realizovány pomocí průvodce. Po otevření kontextového okna se vybere dostupný datový zdroj a definují se zdrojové tabulky zmíněné v kapitole Databáze AdventureWorks. Ty slouží jako základ dalších objektů modelu. Nový datový pohled byl pojmenován a zkontrolován v jednoduchém náhledu. Vytvořený pohled je součástí složky *Data Source Views*.

Tvorba dimenzí

Dimenze jsou vytvořeny již z existujících tabulek z *Data Source Views*. Stejně jako v předchozím případě i pro jejich tvorbu byl použít průvodce. Ten nejprve umožňoval specifikaci datového pohledu obsahujícího požadované výchozí tabulky a následně zobrazil jejich seznam, ze kterého byla vybrána hlavní tabulka pro dimenzi. Průvodce dále automaticky označil klíčový sloupec na základě primárního klíče v původní tabulce. K primárnímu klíči bylo možné zvolit také jmenný sloupec, který nahradil hodnoty klíčového sloupce za smysluplnější popisné hodnoty ze sloupce jiného. Dále byl specifikován typ atributů, jejich zastoupení v dimenzi a název nové dimenze. Stejným postupem byly vytvořeny i další dimenze.

Kontextové okno pracující s dimenzí je rozděleno na tři svislá okna. První z nich obsahuje seznam atributů, které jsou obsaženy v dimenzi. U každého atributu je možné upravovat jeho vlastnosti. Informace o dostupných nastaveních lze najít v referenční příručce Microsoftu (Duncan a Guyer, 2016). Druhé okno obsahuje vytvořené dimenze a třetí okno je pak náhled na tabulku z *Data Source View*. Při tvorbě dimenzí bylo třeba provést následující úpravy:

- **Přidání a konfigurace atributu dimenzí** – nové atributy byly do dimenze přidány během tvorby dimenzí za pomocí průvodce nebo přetažením atributu z podokna *Data Source View*. Atributy bylo vhodné přejmenovat pro lepší pochopení jejich obsahu koncovými uživateli. Dalším nastavením je definování typu atributu, to ho pomáhá klasifikovat z hlediska obchodní funkce. Některé typy atributů mají pro analytické služby specifický význam, pomáhají identifikovat ty, které obsahují časové údaje, adresy, obrázek, osobu a další.
- **Definice a nastavení hierarchií** – uživatelsky definované hierarchie jsou vytvořeny přetáhnutím atributů, které jsou v dimenzi již obsaženy. Jejich levely lze uspořádat rovněž přetažením. Hierarchiím lze nastavit jejich viditelnost a nové jméno. Atributy, které jsou v hierarchiích zastoupeny, je vhodné nastavit jako neviditelné. Pro správné zpracování hierarchie bylo dále třeba upravit vztahy mezi atributy (Torrez, 2016). Ukázka vytvořených hierarchií je zobrazena na obrázku č. 7.
- **Definice a konfigurace vztahů mezi atributy** – pro každou tabulku zahrnutou v dimenzi existují vztahy mezi klíčovým atributem a atributem v této tabulce. Tyto vztahy jsou vytvořeny při definování dimenze. Výhodou vztahů je snížení potřebné paměti při zpracování dimenzí, zvýšení výkonu pro dotazování a agregace.
- **Zpracování dimenzí** – aby bylo možno danou dimenzi vidět a procházet, bylo třeba ji po každé změně znovu zpracovat.



Obrázek 7: Ukázka hierarchií dimenze datumu

Tvorba kostky

V průvodci tvorby kostky byly použity existující tabulky z *Data Source View*. Zde se nejprve vybraly tabulky, které jsou základem pro metriky. V dalším kroku průvodce rozpoznal sloupce s číselnými hodnotami, ze kterých je vhodné vytvořit metriky. Ostatní sloupce obsahující například cizí klíče jsou ignorovány. Tyto sloupce může uživatel sám vybrat nebo odebrat. Po definici metrik byly specifikovány dimenze kostky, zde vybíráme buď již z dříve vytvořených, nebo lze definovat nové. Na závěr byla kostka pojmenována a vytvořena do složky *Cubes*.

Kalkulace

V dialogovém okně kostky se nachází panel metrik (*Measures*), ve kterém byly definovány některé nové metriky. Bylo třeba specifikovat zdrojovou tabulku, sloupec a typ agregace (součet, průměr, počet…). Toto řešení dovoluje výběr agregací pouze z předem daného seznamu. Pro složitější metriky se vytvoří nový kalkulovaný člen (*New Calculated Member*) v okně Kalkulace, které je součástí návrhu kostky. Nové kalkulace mohou být specifikovány pomocí následujících polí a vlastností:

- **Jméno** – název nové kalkulace, který bude viditelný při procházení kostky.
- **Nadřazená hierarchie** – výběr nadřazené hierarchie, která bude zahrnuta v nové kalkulaci.
- **Nadřazený člen** – výběr nadřazeného členu z nadřazené hierarchie.
- **Výraz** – vytvoření výrazu, který vypočítává hodnoty kalkulace. Výraz se zaznamenává v jazyce MDX a používá ostatní komponenty kostky, může používat aritmetické operátory, čísla a funkce.
- **Dodatečné vlastnosti** – mezi dodatečné vlastnosti patří nastavení formátu kalkulace, který ovlivňuje, jak budou hodnoty zobrazeny při procházení. Dalším nastavením je viditelnost. Neviditelné kalkulace se používají jako součásti výra-

zů nových navazujících kalkulací. Kalkulacím můžeme nastavit barvu a písmo založené na hodnotě členu.

Klíčové ukazatele výkonnosti

Klíčové ukazatele výkonosti jsou definovány jako součást návrhu kostky v dialogovém okně *KPIs*. Pro každý nový prvek je třeba zadat jméno KPI a vybrat skupinu metrik, se kterou souvisí. Dále je třeba definovat čtyři následující elementy v jazyce MDX (Sheldon, 2010):

- **Hodnota** (*Value*) – představuje číselné vyjádření skutečné hodnoty KPI.
- **Cíl** (*Goal*) – výpočet vrací cílovou hodnotu KPI, které chce podnik dosáhnout za určité období.
- **Status** – představuje grafický indikátor hodnoty k cíli. MDX výraz by měl vracet číslo mezi -1 a 1. Záporné hodnoty jsou interpretované jako špatné, hodnoty poblíž nuly jako přijatelné a hodnoty blížící se k číslu 1 jako dobré.
- **Trend** – určuje vývoj KPI v čase. Trendem může být jakékoliv časové kritérium, které je součástí obchodní logiky. Tento element umožňuje uživatelům zjistit, zda se hodnota KPI časem zvyšuje nebo naopak snižuje. V grafickém vyjádření bývá reprezentován šipkami.

Perspektivy

Perspektivy jsou vytvořeny za pomocí dialogového podokna *Perspectives* v návrhu kostky. Výběr objektů obsažených v perspektivě je zajištěn jednoduchým zaškrtávacím políčkem. Mohou být schovány skupiny metrik a metriky, dimenze, hierarchie, jmenné sety, KPI, akce a kalkulace. Model praktické části obsahuje dvě perspektivy oddělující internetové prodeje a prodeje obchodníků.

Nasazení na lokální server

Po dokončení multidimenzionálního modelu bylo třeba nasadit analytickou databázi na server služby Analysis Services. Za tímto účelem byl použit průvodce nasazením, který je součástí analytických nástrojů v SSDT. Ten využije výstupní XMLA[13] soubory, které jsou z projektu vygenerovány a nasadí metadata modelu na cílový server. Skript XMLA je možné také uložit a později spustit v aplikaci SQL Server Management Studio.

Při nasazování databází na server bylo třeba specifikovat jméno databáze a připojovací řetězec analytického serveru. V tomto případě byl multidimenzionální model nasazen na SQL Server 2016 Analysis Services běžící v multidimenzionálním a data miningovém módu.

[13]XML (nebo XMLA) je standard pro přístup k analytickým systémům typu OLAP nebo ke strukturám dolování dat.

7.2 Tabulární model

Tabulární model (příloha A2) se vytváří obdobně jako multidimenzionální s tím rozdílem, že na počátku specifikujeme kompatibilní level jako **SQL Server 2016 RTM (1200)**. To nám umožní používat nástroje SQL Serveru 2016 a nasadit model na Azure Analysis Services. Prostředí tabulárního projektu obsahuje následující nejpoužívanější okna:

- **Okno návrhu** – prostředí, které poskytuje vizuální reprezentaci modelu a obsahuje nástroje pro práci s tabulkami. Nabízí dva druhy pohledů, ve kterých probíhá veškerá definice a manipulace s modelem:
  - **Pohled na data** – pracovní okno je tvořeno mřížkou horizontálně rozdělenou na dvě části. Horní část mřížky zobrazuje náhled dat v tabulce. Spodní část mřížky je určena pro tvorbu a zobrazení metrik. Manipulační a výpočetní operace modelu jsou provedeny za použití formulí jazyka DAX.
  - **Pohled na diagram** – grafické zobrazení tabulek a jejich vztahů ve formě diagramu. Pohled obsahuje nástroje pro tvorbu hierarchií, výběr perspektiv a tvorbu nových relací.
- **Průzkumník řešení** (*Solution Explorer*) – kontejner tabulárního projektu zobrazující objekty. Obsahuje pouze odkazy a hlavní soubor *Model.bim*, který se otevírá v okně návrhu.
- **Okno vlastností** – jedná se o hlavní okno definující nastavení pro prvky v modelu. Může obsahovat nastavení vlastností tabulek, sloupců, metrik nebo celého souboru s modelem.

Když vytváříme nový projekt tabulárního modelu, je třeba zvolit jeden ze dvou pracovních prostorů. **Integrovaný pracovní prostor** využívá vlastní interní instanci Analysis Services v SSDT. **Pracovní prostor na serveru** vytváří pracovní databázi na instanci Analysis Services lokálního počítače nebo počítačů ve stejné síti. Praktická část využívala pracovní prostor na serveru, kde byla pracovní databáze viditelná pouze při práci na modelu v aplikaci SSDT.

Import dat z datového zdroje

Průvodce pro importování dat je dostupný v menu souboru *Model.bim*. Zde se definuje připojení na datový sklad a vyberou se požadované zdrojové tabulky nebo pohledy. Pro tabulární model byly použity stejné tabulky jako pro multidimenzionální. Veškeré importované objekty se automaticky zkopírují do modelu a jsou viditelné v návrhovém okně.

Importem do tabulárního projektu používajícího in-memory technologie dochází k zaznamenání obrazu dat z datového zdroje. Data jsou aktualizována manuálně v SSDT za pomocí nabídky *Process*. Jakmile je model zavedený na analytickém

serveru, je možné jeho aktualizace provádět z aplikace SQL Server Management Studio nebo pomocí automatizačního skriptu.

Manipulace s datovým modelem

Každá nová tabulka pocházející z datového skladu se automaticky vytvoří i v modelu a je zpřístupněna v datovém i diagramovém pohledu. Pohled na data umožňuje provádět libovolné transformace s dostupnými daty. Jedná se například o přejmenování tabulek a sloupců, definování sloupců nových, nastavení viditelnosti pro uživatele nebo tvorba metrik. Příkladem manipulace může být vytvoření nového sloupce za pomocí podmínky ve zdrojovém kódu č. 1, který přepisuje hodnoty 1 a 0 ze sloupce HouseOwnerFlag.

Zdrojový kód č. 1: DAX výraz nahrazující hodnoty pomocí podmínky

```
=IF(Customer[HouseOwnerFlag]="1";"Yes";"No")
```

Obdobným způsobem jsou provedeny veškeré drobné úpravy v ostatních tabulkách. Pokud analytický model obsahuje tabulku s datovými údaji, je vhodné ji v programu tak označit. To umožní použití časových a datových funkcí jazyka DAX.

Pohled, ve kterém je zobrazen diagram datového modelu, umožňuje měnit vazby mezi tabulkami. V případě samostatně importované tabulky je třeba vztahy vytvořit přetažením jednoho atributu na druhý. V nastavení vazby specifikujeme sloupce tabulek, typ kardinality a směr filtru. Právě zde je možno nastavit obousměrný křížový filtr, který řeší relace M:N.

Definice hierarchií

Hierarchie se vytvářejí v diagramu modelu přímo v tabulce. Sloupce můžeme přetahovat do již vytvořené hierarchie nebo s nimi vytvořit hierarchii novou. Ta může obsahovat pouze sloupce ze stejné tabulky. Tudíž pokud chceme vytvořit kaskádu s použitím sloupců jiných tabulek, je třeba jejich hodnoty nakopírovat do cílové tabulky obsahující hierarchii.

Zvláštním případem je tvorba hierarchie se vztahem rodič, potomek (Russo, 2014). V datech AdventureWorks se jednalo o tabulku zaměstnanců v jejichž sloupcích se popisuje jednoliniová organizační struktura, kdy každý zaměstnanec má jednoho nadřízeného. Hierarchie byla vytvořena ze sloupců EmployeeKey a ParentEmployeeKey. Nejprve bylo třeba vytvořit nový sloupec obsahující cestu organizační struktury za použití formule ve zdrojovém kódu č. 2.

Zdrojový kód č. 2: DAX výraz tvořící cestu organizační struktury

```
= PATH('Employee'[EmployeeKey]; Employee[ParentEmployeeKey])
```

Z vytvořené cesty byly dále vyhledány konkrétní hodnoty pro jednotlivé organizační levely za pomocí výrazu ve zdrojovém kódu č. 3. Tento výraz byl použit pro vytvoření čtyř sloupců, které obsahují jména zaměstnanců na určitých levelech podle cesty organizačního schématu (obrázek č. 8).

Zdrojový kód č. 3: DAX výraz vyhledávající jména zaměstnanců

```
= LOOKUPVALUE ( Employee[Employee]; Employee[EmployeeKey];
                PATHITEM ( Employee[Path]; 1; INTEGER ))
```

Employ...	ParentEmploye...	Path	Level1	Level2	Level3	Level4
1	18	112|23|18|1	Ken Sánchez	Peter Krebs	Jo Brown	Guy Gilbert
10	189	112|23|189|10	Ken Sánchez	Peter Krebs	Andrew Hill	Ruth Ellerbrock
12	189	112|23|189|12	Ken Sánchez	Peter Krebs	Andrew Hill	Barry Johnson
15	189	112|23|189|15	Ken Sánchez	Peter Krebs	Andrew Hill	Sidney Higa
17	189	112|23|189|17	Ken Sánchez	Peter Krebs	Andrew Hill	Jeffrey Ford
19	189	112|23|189|19	Ken Sánchez	Peter Krebs	Andrew Hill	Doris Hartwig
21	189	112|23|189|21	Ken Sánchez	Peter Krebs	Andrew Hill	Diane Glimp
22	177	112|23|177|22	Ken Sánchez	Peter Krebs	Michael Ray	Steven Selikoff
24	201	112|23|201|24	Ken Sánchez	Peter Krebs	Lori Kane	Stuart Munson

Obrázek 8: Tvorba hierarchie organizační struktury

Definice metrik

Metriky mohou využívat standardní agregační funkce jako je průměr, počet nebo součet. Jsou vytvořené na určitém sloupci se zvoleným typem agregace. Zobrazují se v mřížce pod tabulkou a v databázi vytváří vlastní skupinu složek. Složitější výpočty mohou být vytvořeny v jazyce DAX, a stejně jako jednoduché agregace jsou umístěny v mřížce metrik.

Klíčové ukazatele výkonnosti

Tvorba klíčových ukazatelů výkonosti používá v tabulárním modelu tři prvky (Duncan a Guyer, 2016):

- **Základní hodnota** – hodnota definovaná metrikou. Může to být například agregace prodejů nebo vypočítaný zisk za určité období.
- **Cílová hodnota** – taktéž definovaná metrikou. Jedná se o hodnotu, které chce podnik za určité období dosáhnout.
- **Prahová hodnota stavu** – jedná se o definovaný rozsah prahových hranic stavů, které určují, jak si základní hodnota vede v porovnání s cílovou. Je zobrazována grafickým indikátorem.

Prakticky je třeba nejprve vytvořit metriku zastupující základní hodnotu. Ta sčítá veškeré prodejní sumy na internetu nebo od obchodníků. Následně je třeba definovat metriku cílové hodnoty, která bude násobit součet prodejů z minulého roku předpokládaným koeficientem růstu, v tomto případě číslem 1,5 (zdrojový kód č. 4).

Zdrojový kód č. 4: DAX funkce počítající cílovou hodnotu pro internetové prodeje

```
=(CALCULATE( InternetSales[Internet Sales Amount];
        SAMEPERIODLASTYEAR('Date'[Date])))*1,5
```

Obě tyto metriky tvoří podklad pro prahovou hodnotu stavu, která se vytváří přímo na základní hodnotě. Zde se po zvolení nabídky *Create KPI* otevře nové dialogové okno. To umožňuje procentuální nebo absolutní nastavení prahových hodnot KPI a výběr grafické ikony reprezentující úspěšnost základní hodnoty. Výsledkem použití klíčového podnikového ukazatele může být tabulka zobrazená na obrázku č. 9 sestavená v programu Excel.

Popisky řádků	Internetové prodeje	Předešlý rok	Cílové internetové prodeje	Stav
⊞2010	$43 421,04			
⊞2011	$7 179 723,46	$43 421,04	$65 131,55	
⊞2012	$5 911 706,08	$7 179 723,46	$10 769 585,18	
⊞2013	$16 351 550,34	$5 911 706,08	$8 867 559,11	
⊞2014	$45 694,72	$16 351 550,34	$24 527 325,51	
⊞2015		$45 694,72	$68 542,08	
Celkový součet	**$29 532 095,63**	**$29 532 095,63**	**$44 298 143,44**	

Obrázek 9: Zobrazení klíčových podnikových ukazatelů v programu Excel

Perspektivy

Perspektivy jsou v tabulárním modelu vytvořeny pomocí dialogového okna *Perspectives* v návrháři modelu. Pro každou novou perspektivu se stejně jako v případě multidimenzionálního modelu používají zaškrtávací boxy. V modelu byla definována perspektiva pro internetové prodeje a pro prodeje obchodníků.

Nasazení na analytický server Azure

Pro použití analytických databází v cloudu je třeba nejprve vytvořit nový server v Azure portálu (Microsoft Azure Portal, 2017), kde je nutné mít vytvořený účet s aktivním předplatným. Pro účely této práce byla využita bezplatná zkušební verze (*Azure Free Trial*). Po přihlášení do služby se dostanete na hlavní řídící panel služby, který je složen z přizpůsobitelných dlaždic. V levé části prostředí se nachází hlavní nabídka umožňující výběr a definici konkrétních služeb. Vytvoření nového serveru je provedeno pomocí symbolu plus (*Nový*). Zde se ve složce Intelligence + analytics nachází aplikace Analysis Services. V okně definujícím nový analytický server je třeba vyplnit následující povinné položky:

- **Jméno serveru** – jedinečný název serveru napsaný malými písmeny bez diakritiky.
- **Předplatné** – výběr předplatného, které chce uživatel pro vytvoření serveru využít.
- **Skupinu zdrojů** – jedná se o kontejnery, které spravují sbírky Azure zdrojů. Při definování nového analytického serveru lze vytvořit skupinu novou.
- **Lokalizaci** – umístění hostovacích serverů datacentra Azure. Je doporučováno vybrat lokaci nejblíže největší uživatelské základny organizace. Pro účely práce bylo vybráno datacentrum v západní Evropě.
- **Cenovou úroveň** – definice úrovně a tím příslušné ceny služby. Pro testovací účely byla zvolena úroveň developer.

Nastavení služby ze strany Microsoftu trvá maximálně minutu. Nový server je pak k dispozici buď na řídícím panelu, nebo v navigačním sloupci v levé části obrazovky.

Na vytvořený server je okamžitě možné nasazovat nové tabulární modely. To je zprostředkováno v aplikaci SQL Server Data Tools, kde je již vytvořený tabulární model v příslušné kompatibilní verzi, nebo pomocí aplikace SQL Server Management Studio, kde nasazujeme již existující analytickou databázi pomocí vygenerovaného skriptu. Nejprve je třeba získat název cloudového analytického serveru nacházejícího se na Azure portálu v záložce *Přehled*. Tento název se zkopíruje a vloží do vlastností projektu definujících cílový server. Před samotným nasazením je nutné nejprve nainstalovat on-premises data gateway, kterou lze stáhnout na oficiálních stránkách dokumentujících Microsoft Azure (Duncan a Wheeler, 2017). Samotné nasazení je provedeno pomocí funkce *Build* a možnosti *Deploy solution*.

Databáze je na analytickém serveru k dispozici ihned po jejím nasazení. Samotné připojení je zprostředkováno pomocí programů SSMS, SSDT nebo klientských aplikací jako je Excel, či Power BI. Na Azure portálu naleznete pouze připojovací řetězce, přístup k databázi odtud není možný. Připojení používají HTTPS a vyžadují aktuální knihovny AMO, ADOMD.NET a OLEDB. Pro aplikace Microsoftu komunikující s analytickým serverem jsou aktuální knihovny zahrnuty v jejich nových měsíčních aktualizacích. Při vytvoření analytického serveru bylo třeba specifikovat jedinečný připojovací řetězec skládající se ze jména serveru a regionu, kde byl vytvořen. Řetězec je tvořen na základě následujícího schématu:

<protokol>://<region>/<jméno serveru>

Protokolem je řetězec *asazure*, region je jednotný identifikátor, kde je server vytvořen (v tomto případě *westeurope.asazure.windows.net*) a jméno serveru je jedinečné jméno v rámci regionu. Toto schéma jména serveru používané i jako řetězec pro nasazení je k dispozici v Azure portálu na záložce *Přehled*. Připojení je pak realizováno vložením řetězce do příslušné aplikace. Podrobný popis vytvoření analytického serveru a jeho nasazení je popsán v elektronické příloze A3.

8 Srovnání efektivity modelů

Modely na analytických serverech budou srovnávány z hlediska výkonnosti, která je zastoupena časem zpracovávání uživatelských dotazů. Dále bude srovnávána doba vývoje obou typů modelu a jejich dílčích částí. V neposlední řadě budou modely na serverech posouzeny ohledně vhodnosti nasazení.

8.1 Výkonnostní srovnání modelů

Jak již bylo dříve zmíněno, tabulární model přistupuje k datům přímo z vyrovnávací paměti a zajišťuje rychlé dotazování pomocí sloupcových indexů. Multidimenzionální model oproti tomu přistupuje k předem agregovaným nebo atomickým datům z disku. Následkem toho je i neoptimalizovaný tabulární model výkonnější, což nemusí vždy znamenat rychlejší zpracování uživatelských dotazů. Multidimenzionální model by měl ve skutečnosti poskytovat vyšší výkon pro každou agregaci, jelikož ukládá výsledky dotazů do vyrovnávací paměti. Čím více dotazů tedy nad kostkou provedeme, tím lepšího výkonu dosahuje. Výsledky DAX dotazů zasílané na tabulární model nejsou do vyrovnávací paměti uloženy vůbec, a proto mají dotazy při opakovaném spouštění přibližně stejnou dobu zpracování. Tabulární model je nicméně rychlejší při práci s daty nejnižší úrovně granularity (Duncan a Guyer, 2015).

Výkonnostním srovnáním obou modelů se zabývala i diplomová práce *Využití potenciálu BI sémantického modelu v MS SQL Server 2012* (Zelený J., 2015), ve které autor využil metodiku Marca Russo a Alberta Ferrari (Russo a Ferrari, 2012) a porovnával oba modely na instancích SQL Serveru. Tato práce bude mapovat stejný postup s rozšířením o analytický server Azure.

Testování samotné je prováděno na základě trojice dotazů v jazycích DAX a MDX, které jsou opakovaně spouštěny na analytických serverech. Po každém zpracování dotazu je třeba vyčistit mezipaměť. Výsledky jsou zaznamenány do jednotlivých tabulek v sekundách. Dotazy jsou převzaty z výše zmíněné diplomové práce a upraveny pro analytické modely databáze AdventureWorks. Jejich využití umožní smysluplnější porovnávání výsledků obou prací.

Technické parametry lokálních serverů

Lokální testovací servery jsou vytvořeny na osobním počítači s následujícími technickými parametry:

- Procesor: Intel Core i5-3317U 1,70 GHz
- RAM paměť: 4 GB
- Hard disk: 450 GB, přístupová doba 35,7 ms
- Operační systém: Windows 10

- Databázový systém: SQL Server 216 Enterprise

Tato konfigurace poskytuje rychlou odezvu databází na uživatelské dotazování. Aby bylo možné srovnávat jejich výkon, je třeba rozšířit zdrojovou tabulku internetových prodejů v datovém skladu. V tomto případě je vytvořena nová tabulka FactInternetSalesLarge vycházející z původní tabulky a obsahující nová vygenerovaná data. Ta byla generována pomocí SQL skriptu nacházejícího se v příloze E. Počet záznamů a velikost dat je zobrazen v tabulce č. 3.

Tabulka 3: Parametry původní a rozšířené tabulky s internetovými prodeji

Parametr	**FactInternetSales**	**FactInternetSalesLarge**
Počet záznamů	60 398	1 800 397
Velikost dat [MB]	9,65825	1 772,17

Novou zdrojovou tabulku bylo třeba opět naimportovat do datových modelů a nahradit za tabulku původní. V databázích se pak manuálně provedla aktualizace dat.

Vyprázdnění vyrovnávací paměti

Pro zvýšení výkonů dotazů ukládá služba Analysis Services data do vyrovnávací paměti. V případě multidimenzionálních databází se vyčištěním mezipaměti odstraní vytvořené agregace. Tabulární databáze ukládají do vyrovnávací paměti svá data a výsledky MDX dotazů. Vyprázdnění této paměti má vliv na vytvořené MDX struktury a na mezipaměť enginu VertiPaq ukládající malé sady výpočtů.

Čištění je zajištěno pomocí XMLA skriptu (zdrojový kód č. 5) spustitelném v SQL Server Management Studiu. Oba modely podporují stejný skript, ve kterém se mění pouze jméno databáze (Duncan a Guyer , 2015).

Zdrojový kód č. 5: XMLA dotaz pro vyprázdnění vyrovnávací paměti

```
<ClearCache xmlns="http://schemas.microsoft.com
/analysisservices/2003/engine">
  <Object>
    <DatabaseID> Jmeno databaze</DatabaseID>
  </Object>
</ClearCache>
```

Dotaz č. 1 součet internetových prodejů a počet prodaných výrobků

První dotazy (příloha B) vrací součet internetových prodejů a počet druhů prodaného zboží za jednotlivé prosincové dny před Vánocemi. Dotazy používají agregační

funkci *sum* a operaci *distinct count*. Měření bylo prováděno v pěti iteracích a výsledky jsou zaznamenány v tabulce č. 4.

Tabulka 4: Výkon modelů pro první dotaz

Pokus	**Multidimenzionální mód [s]**	**Tabulární mód [s]**	**Azure AS [s]**
1.	89	66	63
2.	93	66	66
3.	107	68	60
4.	109	66	63
5.	107	69	63
Průměr	101	67	63

Dotaz č. 2 počty zákazníků v jednotlivých městech

Druhý dotaz (příloha C) zjišťuje počty zákazníků v jednotlivých městech. Při použití operace *distinct count* vracející počet jedinečných hodnot jsou využity předem zpracované agregace obou modelů. Výsledky shrnuje tabulka č. 5.

Tabulka 5: Výkon modelů pro druhý dotaz

Pokus	**Multidimenzionální mód [s]**	**Tabulární mód [s]**	**Azure AS [s]**
1.	74	58	58
2.	75	57	58
3.	74	57	60
4.	74	58	61
5.	73	57	58
Průměr	74	57,4	59

Dotaz č. 3 prodeje na základě dojezdových vzdáleností

Poslední dotaz (příloha D) obsahuje agregaci součtu a sčítá prodeje v dojezdových vzdálenostech. Pro každou vzdálenost je také určeno procentuální zastoupení na zisku. Výsledky jsou zaznamenány v tabulce č. 6.

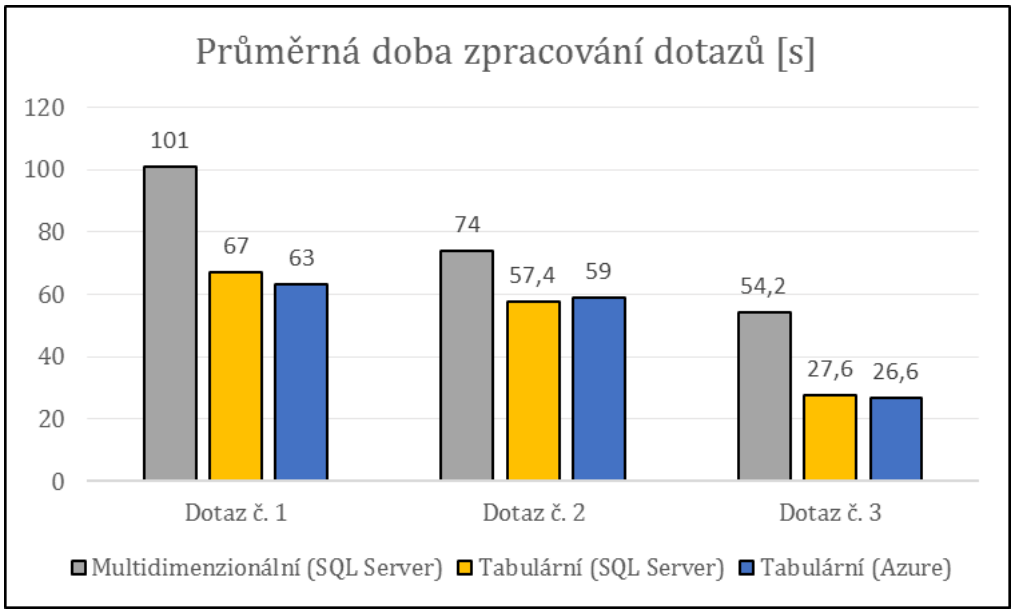
Tabulka 6: Výkon modelů pro třetí dotaz

Pokus	Multidimenzionální mód [s]	Tabulární mód [s]	Azure AS [s]
1.	55	28	26
2.	54	28	26
3.	54	28	27
4.	53	27	26
5.	55	27	28
Průměr	54,2	27,6	26,6

Shrnutí výkonnostního srovnání

Pro každý server bylo provedeno pět měření testujících rychlost modelů v závislosti na předložených dotazech. Rychlost zpracování byla zaznamenána do tabulek a průměrné výsledky jsou zobrazeny na obrázku č. 10.

Pokud se podíváme na výsledné hodnoty prvního dotazu, vidíme, že tabulární model byl mnohem rychlejší než model multidimenzionální, a to nezávisle na použitém serveru. Nízký výkon multidimenzionálního modelu můžeme spojovat s použitím agregace sčítání a operace *distinct count*, která obecně dosahuje lepších výsledků na tabulárním modelu. Rychlost tabulárních analytických serverů byla přibližně stejná s malým výkonnostním náskokem serveru Azure, který má ve verzi developer lehce horší konfiguraci, než je stanice lokálního serveru.



Obrázek 10: Průměrná doba zpracování dotazů

Druhý dotaz byl zaměřený pouze na operaci *distinct count* vracející počty zákazníků z jednotlivých měst. Jeho výsledky jsou mnohem vyrovnanější než v prvním dotazu, hlavně v případě multidimenzionálního modelu. Rozdíly mezi tabulárními servery jsou opět minimální, ale lokální server je v tomto případě výkonnější než cloudový.

Třetí a poslední dotaz je zaměřený pouze na agregaci sčítání a poskytuje nejrychlejší výsledky ze všech měření. Výkonnostní rozdíl mezi řešeními je téměř dvojnásobný. Tabulární model je mírně rychlejší na analytickém serveru Azure.

Pokud se zaměříme na výkon multidimenzionálního modelu, zjišťujeme, že největším problémem je skloubení více operací najednou. Samotná agregační funkce ve třetím dotazu má v porovnání s druhým dotazem mnohem nižší čas zpracování, což potvrzuje problémovost operace *distinct count* pro multidimenzionální model.

Výkon tabulárního modelu je ve všech případech nižší než multidimenzionální. Z výsledků měření můžeme prohlásit, že oba typy serverů dosahují velmi podobných výkonů. Tabulární model nasazený na Azure ve dvou ze tří případů poskytl rychlejší odezvu než lokální SQL Server. Z výkonnostního srovnání můžeme prohlásit, že nejrychlejší reakci na uživatelské dotazování poskytl právě tabulární model nasazený na cloudovém analytickém serveru Azure.

8.2 Časová náročnost vývoje

Porovnávání časové náročnosti vývoje vychází z přímé metody studie času, která se používá v průmyslu pro měření práce. Cílem je porovnat dobu vývoje předem definované databáze pomocí multidimenzionálního a tabulárního modelování. Časové porovnávání umožní určit časově náročná místa vývoje modelů a typ modelu, který je rychleji vytvořen.

Nejprve je třeba nastudovat modelovací techniky vývoje. Ty jsou následně využity při realizaci dvou komplexních modelů použitých pro výkonnostní srovnání. Dále je třeba stanovit rozsah modelované databáze, kterou budou oba modely vytvářet. Vytvořené databáze se budou řídit následujícím předpisem:

- Databáze obsahuje 6 dimenzionálních tabulek a jednu faktovou tabulku.
- Všechny tabulky mají pro uživatele jasné pojmenování.
- V modelu je vytvořeno 5 nových sloupců.
- Dimenzionální tabulky jsou zpracovány do formy dimenzí obsahující 4 hierarchie.
- Z tabulky faktů je vytvořeno 9 metrik s odpovídajícími vlastnostmi (např. měna).
- Databáze obsahuje dvě kalkulace na základě předem daných vzorců.
- Databáze obsahuje předem stanovaný ukazatel výkonosti podniku.

Multidimenzionální i tabulární modelování bylo rozděleno na dílčí podoperace, u kterých se měřila doba vývoje. Modely byly vytvořeny třikrát s časovými rozestupy, aby nedocházelo k přílišné automatizaci úkonů. Nasbírané časové údaje pro multidimenzionální model jsou zaznamenány v tabulce č. 7 a pro tabulární model v tabulce č. 8.

Tabulka 7: Časová náročnost tvorby multidimenzionálního modelu

Název operace	**1** [m]	**2** [m]	**3** [m]	**Průměr** [m]
Vytvoření nového projektu	0,34	0,64	0,47	0,48
Definice datového zdroje	0,75	0,48	0,37	0,53
Definice pohledu na data	1,21	0,81	0,76	0,93
Manipulace s datovým modelem	4,82	5,13	5,45	5,13
Definice dimenzí a hierarchií	17,84	22,22	19,23	19,76
Definice metrik	3,72	3,2	3,06	3,33
Definice kalkulací	3,69	4,46	5,12	4,42
Tvorba KPI	6,65	5,87	4,85	5,79
Nasazení modelu na server	5,74	5,98	4,89	5,54
Celkem	**48,6**	**51,68**	**47**	**49,09**

Tabulka 8: Časová náročnost tvorby tabulárního modelu

Název operace	**1** [m]	**2** [m]	**3** [m]	**Průměr** [m]
Vytvoření nového projektu	1,71	1,44	1,21	1,45
Import dat z datového zdroje	2,06	1,95	2,11	2,04
Manipulace s datovým modelem	6,97	6,85	6,48	6,77
Úprava tabulek a definice hierarchií	9,23	8,87	9,64	9,25
Definice metrik	3,61	3,16	3,35	3,37
Definice kalkulací	3,18	3,27	3,38	3,28
Tvorba KPI	4,88	4,64	4,85	4,79
Nasazení modelu na server	1,23	1,27	1,19	1,23
Celkem	**35,45**	**34,42**	**34,35**	**34,74**

8.3 Vhodnost nasazení

Tabulární model vyšel v obou předchozích typech srovnání jako výhodnější, ale i přesto je třeba porovnávat rovněž vhodnost nasazení modelu. Ta se liší v závislosti na dostupném hardwaru serveru, objemu dat, které chceme zpracovávat a složitosti obchodní logiky aplikované v modelu.

Zdroje serveru

Jednou z nejdůležitějších věcí ovlivňující rychlost obou typů databází jsou hardwarové zdroje serveru. Oba typy modelů využívají jiné prvky hardwaru a jejich konfigurace by měla být rozdílná. Server pro tabulární modely by se měl zaměřit na následující prvky:

- Maximální paměť RAM
- Rychlý více jádrový procesor v co nejnižším počtu soketů
- Mezipaměť (L2)

Nárůst počtu jader na tabulárním serveru zvýší výkon dotazování, ale zároveň se zvyšují licenční poplatky. Z hlediska výnosnosti se vyplatí investovat do interní paměti a její rychlosti. V případě tabuláru není důležitý výkon disku (Sirmon a Steen, 2013).

U serveru pro multidimenzionální databáze by měla být věnována pozornost následujícím komponentám a jejím vlastnostem:

- Maximální paměť na disku
- Rychlý výkon disku
- Dostatek RAM paměti

Na pevné disky se v případě multidimenzionálního řešení ukládají veškerá data v databázi. Proto je tato komponenta pro servery klíčová. Jejich ceny mají za posledních 18 měsíců mírnou tendenci klesat (PC Part Picker, 2017). Paměť RAM by měla být dostatečná pro uložení výsledků z dotazování.

Z hlediska výkonnosti serverů je třeba posoudit hardware, který má firma k dispozici. Pokud by se jednalo o pořízení nového stroje, pak by měla být varianta multidimenzionálního řešení levnější.

Při výběru konfigurace by měl uživatel srovnat celkovou cenu řešení v porovnání se službou Azure Analysis Services. V jejím případě odpadá veškerá starost o hardware nebo o licence. Služba je placena podle času provozu.

Objem dat

Dalším faktorem úspěšného nasazení modelu je i objem dat uložených v datovém skladu. Tato data jsou do modelů importována a jejich množství je důležitým indikátorem, pro který model se rozhodnout. Oba typy modelů používají kompresi, která snižuje původní velikost dat. Velikost kompresního poměru ovšem závisí na charakteristikách dostupných dat a nelze jej předem přesně stanovit. Obvykle se stanovuje odhad, že multidimenzionální databáze bude mít přibližně třetinovou velikost původních dat. U tabulárního modelu se může docílit komprese snižující velikost původních dat na desetinu, a to zejména pokud databáze obsahuje hodně

číselných údajů. U tabulárního modelu platí, že čím více dat máme, tím větší musí být kapacita RAM paměti. Do té jsou kromě dat nahrávány i další struktury, které jsou vytvořeny při načtení databáze do paměti. Požadavky na velikost paměti se během provozu databáze zvyšují, jelikož jsou do ní dále ukládány i výsledky některých dotazů (Duncan a Guyer, 2016).

U projektů s velkými objemy dat se jejich množství stává významným faktorem při výběru modelu. Pokud je třeba zpracovávat terabajty dat, tak tabulární model není tou správnou volbou. Dostupná paměť v takových případech nemůže data zpracovávat. Modely kvůli tomu nabízejí funkci stránkování (neboli *paging*), která zajišťuje výměnu dat v paměti. Pro velké množství dat je lepší používat multidimenzionální řešení.

Funkce modelů

Výběr závisí i na typu funkcí, které modely podporují. V oficiální dokumentaci je shrnut jejich výčet a podpora u jednotlivých modelů (Duncan a Guyer, 2016). Z toho lze vyčíst, že tabulární model nepodporuje tolik funkcí jako multidimenzionální. Proto je třeba při výběru zohledňovat funkce, které chtějí uživatelé při modelování použít.

Aplikace obchodní logiky a dotazovací jazyky

Obchodní logika přidává datovým modelům jejich hlavní hodnotu. Obsahuje různé druhy výpočtů a obchodních pravidel, zjednodušující pozdější analýzy ze strany koncových uživatelů. Pro její aplikaci jsou použity vzorce a jazyky. Multidimenzionální model definuje veškerou logiku za pomocí jazyka MDX. Ten obsahuje propracované možnosti, ale vyžaduje pochopení vícerozměrných konceptů a je celkově složitější na osvojení. Tabulární model využívá jazyk DAX, který je podobný syntaxi formulím programu Excel a díky tomu je i uživatelsky jednodušší. Jeho koncept je založen na relačních databázích a nevyžaduje pochopení multidimenzionálních principů. Tvorba pokročilých výpočtů však může být v jazyce DAX oproti MDX poněkud náročnější.

Multidimenzionální model podporuje tvorbu složitějších MDX skriptů jako je například *Scope*, který umožňuje přepis hodnot v multidimenzionálním prostoru. Další pokročilou funkcí za pomocí MDX je definice jmenných setů (*Named Sets*) vytvářející filtrované sestavy. Tyto prvky jazyk DAX nepodporuje.

Dalším rozdílem při aplikování logiky může být práce s hierarchiemi. Jejich manipulace a kalkulace jsou jednodušeji prováděny za použití jazyka MDX.

Časová logika je v multidimenzionálním modelu aplikována pomocí průvodce, který dimenzi označí a pomůže s návrhem výpočtů. Pro každou kalkulaci se vygeneruje MDX skript. Pokud chceme kalkulace upravovat nebo tvořit nové je třeba znovu spustit průvodce nebo skript definovat ručně. Tabulární model oproti multidimenzionálnímu neobsahuje žádného průvodce a veškeré výpočty jsou tvořeny vzorci a formulemi jazyka DAX.

Konverze měny převádí data z rozdílných zdrojových měn do jedné. Multidimenzionální řešení opět nabízí průvodce, který vytvoří skripty s potřebnými výpočty. Tabulární řešení oproti tomu používá pro výpočty konverzí vzorce DAX.

Veškeré transformace na úrovni řádků se v multidimenzionálním řešení provádějí před načtením dat do modelu. Jsou tvořeny v MDX a jejich provedení se vykoná až při dotazování na model. Tabulární řešení provádí veškeré úpravy přímo v modelu pomocí kalkulovaných sloupců. Transformace jsou provedeny pro každý záznam a uživatel okamžitě vidí výsledek. V případě nedostatečných transformací ve fázi ETL, lze řadu úprav provést velmi jednoduše.

Pro aplikování složité obchodní logiky je výhodnější použít multidimenzionální řešení. Jeho jazyk MDX se sice obtížněji učí, ale pokročilé kalkulace jsou tvořeny jednodušeji než v DAX. Výhodou multidimenzionálního modelu je i možnost použití průvodců. Tabulární řešení je nicméně vhodné pokud máme jednodušší obchodní pravidla nebo je třeba provést řadu dodatečných kalkulací, které nebyly provedeny ve fázi ETL.

8.4 Shrnutí srovnání modelů

Tabulární modelování je nová technologie, pro kterou neustále vznikají nové inovace a počítá se s ní jako s hlavním produktem do budoucna. Pro začínající developery je model jednodušší a rychlejší na naučení a vývoj, jelikož využívá stejného konceptu jako relační databáze. Jeho jazyk DAX definující kalkulace a transformace byl vyvinut z formulí Excelu a dá se jednoduše používat. Co se týká výkonu, tak ve všech případech srovnání poskytl výsledky rychleji než multidimenzionální model. Jeho výhodou je rychlost operace *distinct count*. Pro jeho servery jsou doporučovány obecně dražší konfigurace, tato nevýhoda je však odstraněna podporou cloudového analytického serveru Azure. Model nicméně obtížněji aplikuje složitou obchodní logiku a kvůli omezené kapacitě RAM obtížněji zpracovává velké objemy dat.

Multidimenzionální řešení funguje na trhu už desítku let a jeho funkce jsou velmi propracované, dokáže tak jednodušeji zpracovat složitější obchodní logiku. Hardware jeho serveru bývá často levnější než v případě tabulárního serveru a je vhodný pro zpracovávání velkých objemů dat. Vytvořené multidimenzionální modely jsou často velmi komplexní a jejich vývoj trvá déle. Používají obtížnější jazyk MDX a jejich výkon ani zdaleka neodpovídá tabulárním modelům. Jednou z hlavních nevýhod je, že pro tuto technologii vzniká v současné době jen malé množství inovací.

9 Diskuze a závěr

9.1 Diskuze

V této práci byl realizován tabulární a multidimenzionální model. Modely byly dále srovnávány z několika hledisek.

Ve výkonnostním srovnání byl nejúspěšnější tabulární model nasazený na Azure Analysis Services. Tabulární model je obecně výkonnější hlavně kvůli technologii ukládání dat do sloupcových databází. Pokud porovnáme výsledky výkonu s pracemi jiných autorů (Zelený, 2015; Russo a Ferarri, 2012) docházíme k obdobným závěrům. Ve všech případech tabulární model zpracovává uživatelské dotazy rychleji. Porovnání výkonu tabulárního modelu na serveru Azure Analysis Services je prozatím jediné svého druhu. Výkon služby se odvíjí od zvolené cenové úrovně. V tomto případě byla zvolena úroveň Developer disponující pamětí 3 GB a 20 jednotkami QPU. Výkon 20 jednotek QPU je podle Microsoftu (Duncan a Wheeler, 2017) srovnatelných s výkonem jednoho rychlého jádra. Výsledky byly nicméně srovnané s lokálním SQL Serverem s konfigurací 4 GB RAM a dvěma jádry. Pomalejší odezvu lokálního serveru můžeme přisuzovat dalším aplikacím, jako je například Management Studio, běžících na pozadí. Při použití jiné cenové úrovně by výkon serveru rapidně vzrostl. Součástí zkušebního bezplatného předplatného je kredit ve výši € 170, který byl využit na verzi Developer v ceně € 82,82 za měsíc. Jelikož jsou služby účtovány za jednotlivé dny, mohla přibližně na polovinu měsíce využita verze Basic 1.

Časové srovnání doby vývoje modelu rozděluje modelování do několika pod operací, přičemž u každé stanovuje čas práce. Z průměrné časové náročnosti modelů vidíme, že tabulární model byl vytvořen přibližně o 15 minut rychleji než model multidimenzionální. To je zapříčiněno především časově náročnou definicí dimenzí a zejména hierarchií. V porovnání s tabulárním modelem je i doba nasazení multidimenzionální databáze poněkud zdlouhavá. Tabulární modely byly vytvořeny rychleji a tento proces bylo možné ještě zrychlit. Hlavním zpomalovacím prvkem u tabulárních modelů byla pomalá odezva programu, který musel zpracovávat rovněž náhledy dat.

9.2 Závěr

Jak je zmíněno v knize Tabular Modeling in SQL Server Analysis Services (Russo, 2016) úspěšnost OLAP řešení závisí v 30 až 40 procentech na zvoleném typu modelu. Ptoro bylo hlavním cílem této práce srovnat oba typy analytických modelů služby Analysis Services.

Teoretická část práce se zabývala problematikou business intelligence a datovými sklady. Byly rozebrány koncepty dimenzionálního modelování, které lez využít jak pro návrh datového skladu, tak i pro pochopení principu multidimenzionálních kostek. Součástí rozboru byla i architektura business intelligence, která představovala jeho jednotlivé komponenty. Součástí této práce je rovněž rozbor nástrojů

pro realizaci business intelligence od firmy Microsoft se zaměřením na SQL Server Analysis Services a na Azure Analysis Services. Tato kapitola představuje multidimenzionální a tabulární analytické modely. Tabulární model využívá technologie ukládání dat do RAM paměti, které jsou popsány v kapitole In-memory technologie.

V praktické části byla popsána struktura datového skladu AdventureWorks následovaná výběrem tabulek, ze kterých se následně realizoval multidimenzionální a tabulární model. Oba tyto modely byly nasazené na lokální SQL Server 2016 a tabulární model navíc na server platformy Azure Analysis Services. Pro tyto modely byly vytvořeny tři typy dotazů, které zkoumaly jejich výkonost na jednotlivých serverech. Aby byly výsledky srovnatelné, bylo třeba původní databázi rozšířit. Výsledkem výkonnostního srovnání je graf, ve kterém jsou zprůměrovány časy zpracování dotazů. Nejvýkonnějším řešením se jevil tabulární model na Azure Analysis Services. Druhé srovnání se týkalo času vývoje modelů. Pro tyto účely byla navržena menší databáze, kterou bylo možné opakovaně modelovat. Samotné modelování bylo rozděleno na několik kroků a pro každý krok byla změřena časová náročnost. Výsledkem byla celková náročnost modelů a jeho pod operací. Z hlediska doby vývoje je rychlejší tabulární model. Při rozhodování o typu modelu je třeba brát v potaz vhodnost nasazení. Ta byla rozlišena vzhledem k dostupnému hardwaru, objemu dat, funkcím a možnostem aplikace obchodní logiky.

S tímto tématem souvisí i řada rozšíření, které by bylo možné realizovat. Jednou z možností může být testování výkonnosti módů VertiPaq a DirectQuery u tabulárního modelu. Tyto módy můžeme využívat buď samostatně nebo hybridně. Dále by bylo zajímavé srovnávat cenu hardwarové konfigurace tabulárního serveru s náklady na provoz cloudového serveru Azure AS.

Obecným doporučením je využívat tabulární model, kvůli jeho výkonu a jednoduchosti. Nicméně volba závisí na konkrétních požadavcích podniku. Výsledky této práce mohou sloužit jako poklad jejich rozhodování.

10 Reference

[1] Adventure Works 2014: *Sample Database*. CodePlex [online]. 2015 [cit. 2017-05-13]. Dostupné z: https://msftdbprodsamples.codeplex.com/releases/view/125550.

[2] Database reference: *Adventure Works DW*. Elasoft [online]. 2010 [cit. 2017-05-13]. Dostupné z: http://elsasoft.com/samples/analysisserver-amoadventureworks.

[3] Dlabač J. *Analýza a měření práce*. PI: Academy of Productivity and Innovations [online]. 2015 [cit. 2017-05-18]. Dostupné z: http://www.e-api.cz/25784n-analyza-a-mereni-prace.

[4] Duncan 0. a C. Guyer. *Analysis Services* Microsoft Docs [online]. 2015 [cit. 2017-04-20]. Dostupné z: https://docs.microsoft.com/cs-cz/sql/analysis-services/analysis-services.

[5] Duncan O. a S. Wheeler. *What is Azure Analysis Services*. Microsoft Azure [online]. 2017 [cit. 2017-04-27]. Dostupné z: https://docs.microsoft.com/cs-cz/azure/analysis-services/analysis-services-overview.

[6] Gandhi G. *Introduction to tabular model. Sarjen Systems* [online]. 2016 [cit. 2017-04-26]. Dostupné z: http://www.sarjen.com/introduction-to-tabular-model/.

[7] Heinze J. *History of Business Intelligence*. Better Buys [online]. 2014 [cit. 2017-04-05]. Dostupné z: https://www.betterbuys.com/bi/history-of-business-intelligence/.

[8] Inmon B. *Building the Data Warehouse*. 3rd. Indianopolis: John Wiley, 2002. ISBN 0-471-08130-2.

[9] Kimball, R. a M. Ross. *The data warehouse toolkit: the definitive guide to dimensional modeling*. Third edition. Canada: Wiley, 2013. ISBN 978-1-118-53077-1.

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

[11] Lacko L. *Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle*. Brno: Computer Press, 2003. ISBN 80-722-6969-0.

[12] Lake P. a Crowther P. *In-Memory Databases*. Concise Guide to Databases: A Practical Introduction. London: Springer London, 2013, 188-197, DOI 10.1007/978-1-4471-5601-7-8.

[13] Magagnotti N. a Spinelli R. *Operations research and measurement methodologies*. Good practice guidelines for biomass production studies. Siesto Fiorentino: CNR Ivalsa, 2012. ISBN 978-889-0166-044.

[14] Melomed E. *Microsoft SQL Server 2005 analysis services*. Indianapolis: Sams Pub., 2007. ISBN 0-672-32782-1.

[15] Microsoft Azure. *Nástroje pro Business Intelligence* [online]. 2017 [cit. 2017-04-23]. Dostupné z: https://azure.microsoft.com/cs-cz/solutions/business-intelligence/.

[16] Microsoft Azure Portal. [online]. Seattle, 2017 [cit. 2017-05-17]. Dostupné z: https://azure.microsoft.com/cs-cz/features/azure-portal/.

[17] Microsoft: News Center. *Acquisition Of Panorama OLAP Technology* [online]. 1996 [cit. 2017-04-20]. Dostupné z: https://news.microsoft.com/1996/10/29/microsoft-announces-acquisition-of-panorama-online-analytical-processing-olap-technology/.

[18] Microsoft: TechNet. *MS SQL Server 7.0 OLAP Services* [online]. 2001 [cit. 2017-04-20]. Dostupné z: https://technet.microsoft.com/en-us/library/cc966398.aspx.

[19] Mundy J., W. Thornthwaite a R. Kimball. *The Microsoft data Warehouse toolkit: with SQL server 2008 R2 and the Microsoft Business intelligence toolset*. 2nd ed. Indianapolis, IN: Wiley Pub., 2011. ISBN 978-0-470-64038-8.

[20] PC Part Picker: *Price Trends*. How to interpret the price trend graphs [online]. 2017 [cit. 2017-05-21]. Dostupné z: https://pcpartpicker.com/trends/internal-hard-drive/.

[21] Pour J., M. Maryška a Novotný O. *Business intelligence v podnikové praxi*. Praha: Professional Publishing, 2012. ISBN 978-80-7431-065-2.

[22] Ray M., J. Vance a C. Guyer. *Editions and Components of SQL Server 2016*. Microsoft Docs [online]. 2016 [cit. 2017-04-19]. Dostupné z: https://docs.microsoft.com/cs-cz/sql/sql-server/editions-and-components-of-sql-server-2016.

[23] Russo M. *Parent-Child Hiearchies*. DAX Patterns [online]. 2014 [cit. 2017-05-16]. Dostupné z: http://www.daxpatterns.com/parent-child-hierarchies/.

[24] Russo M. *Tabular modeling in microsoft SQL server analysis services*. 2nd edition. ISBN 978-1509302772.

[25] Russo M a A. Ferrari. *Multidimensional vs. Tabular*. SQL BI [online]. 2012 [cit. 2017-05-18]. Dostupné z:

https://www.sqlbi.com/tv/bism-multidimensional-vs-tabular/.

[26] Russo M., A. Ferrari a C. Webb. *Microsoft SQL Server 2012 analysis services: the BISM Tabular Model*. Sebastopol, CA: Microsoft Press, 2012. ISBN 978-0-7356-5818-9.

[27] Sharma R. *Microsoft SQL Server 2000: a guide to enhancements and new features*. Boston: Addison-Wesley, 2002. ISBN 02-017-5283-2.

[29] Sheldon A. *Adding a KPI to an SQL Server Analysis Services Cube*. Simple Talk [online]. 2010 [cit. 2017-05-16]. Dostupné z: https://www.simple-talk.com/sql/reporting-services/adding-a-kpi-to-an-sql-server-analysis-services-cube/.

[28] Sirmon J. a H. Steen. *Hardware Sizing a Tabular Solution* [online]. 2013 [cit. 2017-05-16]. Dostupné z: https://msdn.microsoft.com/en-us/library/jj874401.aspx.

[30] Torrez D. *Creating a Date Dimension*. Simple Talk [online]. 2016 [cit. 2017-05-16]. Dostupné z: https://www.simple-talk.com/sql/bi/creating-a-date-dimension-in-an-analysis-services-ssas-cube/.

[31] Varga S., D. Cherry a J. D'antoni. *Introducing Microsoft SQL Server 2016: Mission-Critical Applications, Deeper Insights, Hyperscale Cloud*. Redmond: Microsoft Press, 2016. ISBN 978-1-5093-0195-9.

[32] Zelený J. *Využití potenciálu BI sémantického modelu v MS SQL Server 2012*. Brno, 2015 . Diplomová práce. Mendelova univerzita v Brně. Vedoucí práce Jan Přichystal.

Přílohy

A Elektronické přílohy

Součástí práce jsou následující elektronické přílohy:

1. **Multidimenzionální model** – projekt Visual Studia obsahující struktury multidimenzionálního modelu.
2. **Tabulární model** – projekt Visual Studia obsahující struktury tabulárního modelu.
3. **Azure Analysis Services** – detailní popis vytváření analytického serveru a způsob nasazení.

B Dotaz č. 1 součet internetových prodejů a počet prodaných výrobků

Zdrojový kód č. 6: DAX dotaz pro součet předvánočních prodejů z roku 2012

```
evaluate(
        filter(
                summarize(
                InternetSales
                , 'Date'[DateKey]
                , "Sum of Sales"
                , sum('InternetSales'[SalesAmount])
                , "Distinct Products"
                , 'Product'[Distinct Count of Products]
                )
                , ('Date'[DateKey]<value(20121224))
        )
)
order by
'Date'[DateKey]
Start at
value(20121201)
```

Zdrojový kód č. 7: MDX dotaz pro součet předvánočních prodejů z roku 2012

```
select
 {
        [Measures].[Product Key Distinct Count],
        [Measures].[Internet Sales Amount]
} on columns,
        [Order Date].[Date].&[2012-12-01T00:00:00]:
        [Order Date].[Date].&[2012-12-24T00:00:00] on rows
from [Sales]
```

C Dotaz č. 2 počet zákazníků v jednotlivých městech

Zdrojový kód č. 8: DAX dotaz počítající jedinečné zákazníky v daných městech

```
evaluate(
        filter(
                summarize(
                        Geography
                        ,'Geography'[City]
                        , "Unique customers"
                        ,'Customer'[Customer Distinct Count]
                )
                , 'Customer'[Customer Distinct Count]>100
        )
)
```

Zdrojový kód č. 9: MDX dotaz počítající jedinečné zákazníky v daných městech

```
select
        nonempty(
                        [Measures].[Customer Key Distinct Count]
                ) on columns,
filter(
                [Geography].[Customers].[City].members,
                [Measures].[Customer Key Distinct Count]>100
) on rows
from
[Sales]
```

D Dotaz č. 3 prodeje na základě dojezdových vzdáleností

Zdrojový kód č. 10: DAX dotaz počítající prodeje a procentuální zisk dle dojezdové vzdálenosti

```
evaluate
(
        filter
        (
                summarize
                (
                        Customer
                        , 'Customer'[Commute Distance]
                        , "Sum of Sales"
                        , 'InternetSales'[Internet Sales Amount]
                        , "Profit Margin %"
                        , ROUND('InternetSales'
                        [Internet Profit Margin]*100,2)
                )
                , 'InternetSales'[Internet Sales Amount])
)
order by
'Customer'[Commute Distance]
```

Zdrojový kód č. 11: MDX dotaz počítající prodeje a procentuální zisk dle dojezdové vzdálenosti

```
select
        nonempty(
                {[Measures].[Internet Sales Amount],
                [Measures].[Internet Profit Margin]}
        ) on columns,
        nonempty(
        [Customer].[Commute Distance].members
        ) on rows
from
[Sales]
```

E SQL kód generující nová data

```
DECLARE @maxProduct INT
DECLARE @minCustomer INT
DECLARE @maxCustomer INT
DECLARE @minPromotion INT
DECLARE @maxPromotion INT
DECLARE @minCurrency INT
DECLARE @maxCurrency INT
DECLARE @minSalesTerritory INT
DECLARE @maxSalesTerritory INT
DECLARE @minExtendedAmount INT
DECLARE @maxExtendedAmount INT
DECLARE @minTotalProductCost INT
DECLARE @maxTotalProductCost INT
DECLARE @minSalesAmount INT
DECLARE @maxSalesAmount INT
DECLARE @minTaxAmount INT
DECLARE @maxTaxAmount INT
DECLARE @minFreight INT
DECLARE @maxFreight INT

DECLARE @counter INT = 0

DECLARE @orderNumber INT
DECLARE @salesOrderNumber NVARCHAR(20)


SELECT @maxproduct = MAX([ProductKey])
FROM [AdventureWorksDW2014].[dbo].[DimProduct];

SELECT @minCustomer = MIN([CustomerKey])
FROM [AdventureWorksDW2014].[dbo].[DimCustomer];

SELECT @maxCustomer = MAX([CustomerKey])
FROM [AdventureWorksDW2014].[dbo].[DimCustomer];

SELECT @minPromotion = MIN([PromotionKey])
FROM [AdventureWorksDW2014].[dbo].[DimPromotion];

SELECT @maxPromotion = MAX([PromotionKey])
FROM [AdventureWorksDW2014].[dbo].[DimPromotion];
```

```
SELECT @minCurrency = MIN([CurrencyKey])
FROM [AdventureWorksDW2014].[dbo].[DimCurrency];

SELECT @maxCurrency = MAX([CurrencyKey])
FROM [AdventureWorksDW2014].[dbo].[DimCurrency];

SELECT @minSalesTerritory = MIN([SalesTerritoryKey])
FROM [AdventureWorksDW2014].[dbo].[DimSalesTerritory];

SELECT @maxSalesTerritory = MAX([SalesTerritoryKey])
FROM [AdventureWorksDW2014].[dbo].[DimSalesTerritory];

SELECT @minExtendedAmount = MIN([ExtendedAmount])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @maxExtendedAmount = MAX([ExtendedAmount])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @minTotalProductCost = MIN([TotalProductCost])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @maxTotalProductCost = MAX([TotalProductCost])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @minSalesAmount = MIN([SalesAmount])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @maxSalesAmount = MAX([SalesAmount])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @minTaxAmount = MIN([TaxAmt])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @maxTaxAmount = MAX([TaxAmt])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @minFreight = MIN([Freight])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];

SELECT @maxFreight = MAX([Freight])
FROM [AdventureWorksDW2014].[dbo].[FactInternetSales];
```

```
SELECT @orderNumber= SUBSTRING(MAX([SalesOrderNumber]), 3, 7)
FROM AdventureWorksDW2014.dbo.FactInternetSales;


WHILE @counter<1200000
BEGIN

SET @orderNumber = @orderNumber+1
SET @counter = @counter+1
SET @salesOrderNumber = CONCAT('ON',CONVERT(NVARCHAR(20), @orderNumber))

INSERT INTO [AdventureWorksDW2014].[dbo].[FactInternetSalesLarge]
 ( [ProductKey]
 , [OrderDateKey]
 , [DueDateKey]
 , [ShipDateKey]
 , [CustomerKey]
 , [PromotionKey]
 , [CurrencyKey]
 , [SalesTerritoryKey]
 , [SalesOrderNumber]
 , [SalesOrderLineNumber]
 , [RevisionNumber]
 , [OrderQuantity]
 , [UnitPrice]
 , [ExtendedAmount]
 , [UnitPriceDiscountPct]
 , [DiscountAmount]
 , [ProductStandardCost]
 , [TotalProductCost]
 , [SalesAmount]
 , [TaxAmt]
 , [Freight])


SELECT
(ABS(CHECKSUM(NEWID())) % @maxProduct + 1) ProductKey

, CONVERT(INT, CONVERT(NVARCHAR(10)
, CAST(DATEADD(dd,ABS(CHECKSUM(NEWID()))%700+1
, CONVERT(DATE,CONVERT(NVARCHAR(10)
, [OrderDateKey])))) AS DATE), 112)) OrderDateKey
```

```
, CONVERT(INT, CONVERT(NVARCHAR(10)
, CAST(DATEADD(dd,ABS(CHECKSUM(NEWID()))%700+1
, CONVERT(DATE,CONVERT(NVARCHAR(10)
, [DueDateKey])))) AS DATE), 112)) DueDateKey

, CONVERT(INT, CONVERT(NVARCHAR(10)
, CAST(DATEADD(dd,ABS(CHECKSUM(NEWID()))%700+1
, CONVERT(DATE,CONVERT(NVARCHAR(10)
, [OrderDateKey])))) AS DATE), 112)) ShipDateKey

, ROUND(RAND()*(@maxCustomer-@minCustomer)+@minCustomer, 0)
 CustomerKey

, ROUND(RAND()*(@maxPromotion-@minPromotion)+@minPromotion, 0)
 PromotionKey

, ROUND(RAND()*(@maxCurrency-@minCurrency)+@minCurrency, 0)
 CurrencyKey

, ROUND(RAND()*(@maxSalesTerritory-@minSalesTerritory)
+@minSalesTerritory, 0) SalesTerritoryKey

, @salesOrderNumber [SalesOrderNumber]

, ROUND(RAND()*(8-1)+1, 0) SalesOrderLineNumber

, '1' RevisionNumber

, '1' OrderQuantity

, dp.StandardCost UnitPrice

, ROUND(RAND()*(@maxExtendedAmount-@minExtendedAmount)+
@minExtendedAmount, 4) ExtendedAmount

, '0' UnitPriceDiscountPct

, '0' DiscountAmount

, dp.StandardCost ProductStandardCost

, ROUND(RAND()*(@maxTotalProductCost-@minTotalProductCost)
+@minTotalProductCost, 4) TotalProductCost
```

```
, ROUND(RAND()*(@maxSalesAmount-@minSalesAmount)+@minSalesAmount, 4)
SalesAmount

, ROUND(RAND()*(@maxTaxAmount-@minTaxAmount)+@minTaxAmount, 4)
 TaxAmt

, ROUND(RAND()*(@maxFreight-@minFreight)+@minFreight, 4)
Freight

FROM [AdventureWorksDW2014].[dbo].[FactInternetSales] fis
JOIN AdventureWorksDW2014.dbo.DimProduct dp
ON fis.ProductKey = dp.ProductKey
END
```