

# **Analýza a návrh řešení datového skladu a reportingu prodejů nad maloobchodní databází**

Ondřej Zálešák

---

Bakalářská práce  
2015



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2014/2015

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ondřej Zálešák**  
Osobní číslo: **A12602**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Informační a řídicí technologie**  
Forma studia: **kombinovaná**

Téma práce: **Analýza a návrh řešení datového skladu a reportingu prodeje nad maloobchodní databází**

Téma anglicky: **The Analysis and Design of Data Warehouse Solutions and Retail Sales Database Reporting**

Zásady pro vypracování:

- 1. Vytvořte literární rešerši na téma řešení datového skladu.**
- 2. Analyzujte aktuální řešení datového skladu.**
- 3. Navrhněte optimalizované řešení datového skladu.**
- 4. Pomocí libovolného nástroje navržené řešení realizujte.**
- 5. Zhodnoťte řešení, přínos a možný budoucí vývoj.**

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

1. LABERGE, Robert. Datové sklady: agilní metody a business intelligence. 1. vyd. Brno: Computer Press, 2012, 350 s. ISBN 978-80-251-3729-1.
2. NOVOTNÝ, Ota, Jan POUR a David SLÁNSKÝ. Business intelligence: jak využít bohatství ve vašich datech. 1. vyd. Praha: Grada, 2005, 254 s. ISBN 8024710943.
3. LACKO, L'uboslav. Mistrovství v SQL Server 2012: [kompletní průvodce databázového experta]. 1. vyd. Brno: Computer Press, 2013, 640 s. ISBN 978-80-251-3773-4.
4. KNIGHT, Brian. Professional SQL server 2008 integration services. Indianapolis, IN: Wiley Pub., c2008, xxxiii, 972 p. ISBN 0470247959.
5. LARSON, Brian. Microsoft SQL Server 2012 reporting services. New York: McGraw-Hill, c2012, xx, 762 p. ISBN 0071760474.
6. HESTER, Matthew a Chris HENLEY. Microsoft windows server 2012 administration: instant reference. Indianapolis, Ind.: Sybex, c2013, 1 online zdroj (xviii, 574 p.).
7. KROENKE, David a David J AUER. Databáze. 1. vyd. Brno: Computer Press, 2015, 496 s. ISBN 978-80-251-4352-0.

Vedoucí bakalářské práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

6. března 2015

Termín odevzdání bakalářské práce:

22. května 2015

Ve Zlíně dne 6. března 2015

doc. Mgr. Milan Adámek, Ph.D.  
děkan



prof. Ing. Vladimír Vašek, CSc.  
ředitel ústavu

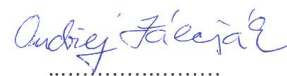
### Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal, jsou uvedeny v seznamu použité literatury.

Souhlasím s prezenčním zpřístupněním své bakalářské práce v univerzitní knihovně.

Ve Zlíně dne 18.5.2015

Ondřej Zálešák



.....

## **ABSTRAKT**

Tato práce se zabývá analýzou aktuálního řešení reportingu a návrhem optimalizovaného řešení nad obchodní databází firmy XY.

Klíčová slova: Datový sklad (DWH), Business Intelligence (BI), OLAP (Online Analytical Processing), OLTP (Online Transaction Processing), Dimenze, Fakta, MS SQL Server, SSIS (SQL Server Integrované Služby), SSAS (SQL Server Analytické Služby), SSRS (SQL Server Reportovací Služby)

## **ABSTRACT**

This thesis deals with the issue of current XY company reporting solution and provides a proposal of the optimized solution.

Keywords: Data Warehouse (DWH), Business Intelligence (BI), OLAP (Online Analytical Processing), OLTP (Online Transaction Processing), Dimension, Fact, MS SQL Server, SSIS (SQL Server Integration Services), SSAS (SQL Server Analyses Services), SSRS (SQL Server Reporting Services)

## **PODĚKOVÁNÍ**

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Petru Šilhavému, Ph.D. za odborné vedení, konstruktivní připomínky a návrhy na zlepšení práce.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

## OBSAH

ÚVOD.....	9
I. TEORETICKÁ ČÁST .....	10
1 BUSINESS INTELLIGENCE.....	11
1.1 CÍL BI.....	11
1.2 KLÍČOVÉ INDIKÁTORY VÝKONU BI.....	11
1.3 KVALITA SYSTÉMU BI.....	12
1.4 ZÁKLADNÍ PRINCIPY MULTIDIMENZIONÁLNÍCH DATABÁZÍ .....	12
1.4.1 STRUKTURA MULTIDIMENZIONÁLNÍ DATABÁZE .....	13
2 DATOVÝ SKLAD.....	16
2.1 DEFINICE DATOVÉHO SKLADU .....	16
2.2 SYSTÉM DATOVÉHO SKLADU.....	16
2.3 ARCHITEKTURA DATOVÉHO SKLADU .....	18
2.3.1 TERMINOLOGIE TOKU DAT.....	19
2.4 ÚČEL DATOVÉHO SKLADU .....	19
3 OLAP DATABÁZE.....	21
4 DATOVÉ TRŽIŠTĚ – DMA (DATA MART).....	22
5 ETL .....	23
5.1 ARCHITEKTURA ETL .....	24
5.2 POŘIZOVÁNÍ DAT.....	24
5.3 DISTRIBUCE DAT.....	25
6 MS SQL SERVER.....	27
6.1 HISTORIE A VÝVOJ SQL SERVERU .....	27
6.2 SSIS.....	28
6.3 SSAS .....	29
6.4 SSRS .....	29
II. PRAKTICKÁ ČÁST.....	30
7 ÚVOD DO PROBLEMATIKY .....	31
7.1 POPIS NEDOSTATKŮ .....	31
7.2 NÁVRH ŘEŠENÍ .....	32
7.3 STRUKTURA OBCHODNÍ DATABÁZE .....	32
7.4 SCHÉMA OBCHODNÍ DATABÁZE .....	34
8 TVORBA DATOVÉHO SKLADU .....	35
8.1 STRATEGIE.....	35
8.2 DEFINICE .....	35
8.3 ANALÝZA .....	36
8.3.1 STAGE .....	36
8.3.2 CORE.....	36
8.3.3 MART .....	36
8.3.4 DIMENZE A FAKTA.....	36
8.4 NÁVRH .....	37

8.4.1	ARTIKLY .....	37
8.4.2	PRODEJNY .....	37
8.4.3	KALENDÁŘ.....	37
8.4.4	ČASOVÉ OBDOBÍ.....	38
8.4.5	HLAVIČKY PRODEJŮ .....	38
<b>8.5</b>	<b>TVORBA STRUKTUR PRO DATOVÝ SKLAD .....</b>	<b>38</b>
8.5.1	DWH_STAGE .....	38
8.5.2	DWH_CORE.....	39
8.5.3	DWH_MART .....	42
<b>8.6</b>	<b>ETL.....</b>	<b>43</b>
<b>9</b>	<b>REPORTING.....</b>	<b>44</b>
<b>9.1</b>	<b>REPORT PRODEJŮ V AKTUÁLNÍM ROCE.....</b>	<b>44</b>
<b>9.2</b>	<b>REPORT PRODEJŮ ZA MINULÝ A AKTUÁLNÍ ROK.....</b>	<b>46</b>
<b>9.3</b>	<b>MĚŘENÍ VÝKONU PŘI GENEROVÁNÍ REPORTU .....</b>	<b>47</b>
	<b>ZÁVĚR .....</b>	<b>49</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>51</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>52</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>54</b>
	<b>SEZNAM TABULEK.....</b>	<b>55</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>56</b>

## ÚVOD

Díky digitalizaci dat je možné s nimi efektivně pracovat, vyhodnocovat je a upravovat svou budoucí obchodní strategii na základě dosažených výsledků.

Stejně tak, jako je třeba správně třídit data například v kartotéce, tak i uchování dat v elektronické podobě má svá specifika a pravidla, díky kterým je práce s nimi efektivní a rychlá.

Již od 60. let 20. století začal vývoj jednotného jazyka pro práci s databázemi, který vedl k vývoji první relační databáze v roce 1970.

Jak s postupem času množství dat v databázi přibývá, vyhodnocení začíná být časově zdlouhavé a náročné na výpočetní složitost. Data, například z prodejních transakcí, jsou uloženy do struktury OLTP, kdy každá transakce představuje jeden řádek. Toto je výhodné pro ukládání do databáze, ale ne pro následné vyhodnocení a analytickou práci s daty. Proto se na základě požadovaných reportů data transformují do struktury OLAP, kde jsou logicky slučována tak, aby poskytla co nejvyšší míru efektivity práce.

Firma XY je na trhu již více než 15 let a vyvíjí obchodní databáze pro maloobchod. Aby data měla přidanou hodnotu a poskytla uživatelům možnost na jejich základě rozhodovat, je třeba mít možnost data vyhodnocovat a reportovat z nich. Mezi klíčové metriky reportingu se řadí hodnota prodeje, počet prodejů, počet zákazníků, hodnota slev, hodnota DPH a hlavní dimenze jsou místo prodeje (prodejna), datum a čas, zákazník, druh slev, sazba DPH. Tato data jsou momentálně ukládána v zastaralém systému, kde každý týden dat představuje jednu tabulku a při generování reportů s rozsáhlejším časovým úsekem musí uživatel čekat i několik hodin. Cílem této práce je návrh řešení, které bude poskytovat stejnou kvalitu dat jako předchozí řešení, ale generování reportu bude trvat kratší dobu.

## **I. TEORETICKÁ ČÁST**

## 1 BUSINESS INTELLIGENCE

*Business Intelligence (BI)* je obecný pojem, který se vztahuje ke znalostem, procesům, technologiím, aplikacím a postupům usnadňujícím podnikové rozhodování. Technologie Business Intelligence využívá historicky použitá data v požadovaných souvislostech a na jeho základě je možné řídit a rozhodovat otázky ohledně firemní strategie a budoucího směru vývoje firmy. [1]

### 1.1 Cíl BI

Technologie Business Intelligence se zaměřuje primárně na vnitropodnikové informace o provozních pohledech, které se ať přímo, či nepřímo dotýkají životně důležitého plánování. Informace jsou zpravidla určitým způsobem řazeny a přímo nebo nepřímo se odvozují z aktuálních vnitropodnikových procesů. Výstup základních dat se využívá nejčastěji v interní analýze, nicméně je možné jej kombinovat i s vnější analýzou, jako je například srovnávací analýza SWOT a PEST. [1]

### 1.2 Klíčové indikátory výkonu BI

Požadavky na technologii Business Intelligence jsou určení podnikových cílů na základě základních klíčových ukazatelů výkonu KPI (= Key Performance Indicator). Uvedené klíčové indikátory výkonu se zpravidla dělí na tzv. metriky, jinak řečeno fakta, jež jsou bezprostředně odvozeny z produkčních systémů dané organizace, v jiném případě mohou být z uvedených bazálních produkčních metrik slučována. Fakta bývají zpravidla numerická a také kvantifikovatelná, což znamená, že je možné je počítat. Standardní průběh analýzy v systému Business Intelligence většinou probíhá více na agregované úrovni než na úrovni instancí. Společnost má potřebu analyzovat, jaký počet klientů se vyskytuje, bydlí nebo pravidelně kupuje zboží v dané oblasti, ne však přesné místo, kde daný klient žije a jaké množství zboží nakoupil. Pokud nastane varianta číslo dvě, je totiž možné požadovaný výsledek zjistit, nebo vyhodnotit z aktuálního produkčního systému. Tytéž informace je možné také získat ze systému BI, pokud předpokládáme, že systém má informace uloženy na námi požadované úrovni granularity. Existují i organizace, jejichž produkční systémy neudržují bazální informace ohledně toho, jaké množství zboží zakoupil sám jednotlivý klient. Proto se používá jako hlavní úložiště dat o provozu a událostech systém Business Intelligence. Jsou i podniky, které preferují tvorbu tzv. úložišť provozních dat (ODS – operational data store), které obsahují informace na vyšší úrovni granularity. Tento typ dato-

vých úložišť ovšem mohou mít limitované možnosti pro ukládání historických dat, tudíž jako nejběžnější bývá využíváno prostředí Business Intelligence. [1]

### 1.3 Kvalita systému BI

Kvalitní systémy Business Intelligence mají jako hlavní znaky přesnost a aktuálnost, obsahují vysokou hodnotu a dávají manažerovi možnost dalšího rozhodování: *Přesnost* znamená to, že se můžeme na uložená data plně spolehnout; *aktuálnost* značí to, že data jsou dostupná na základě určitého časového plánu; *vysoká hodnota* je úzce spojena s efektivní prací standardního podnikového uživatele a *možnost rozhodování* má přímou souvislost s využitím získané informace v dalším průběhu rozhodovacího procesu. Informace ohledně toho, že daná společnost má 100 židlí může z určitého pohledu být užitečná, otázkou ovšem je, zda je možné tuto informaci dále uplatnit v rozhodovacím procesu? V případě, že je odpověď kladná, potom je samozřejmě logické tato data uchovávat. V opačném případě nemá smysl plýtvat prostředky, pokud společnost tato data nezužítkuje a nemá možnost na základě jejich vyhodnocení dělat další rozhodnutí. [1]

### 1.4 Základní principy multidimenzionálních databází

V běžné praxi pracují informační systémy s informacemi, které mohou být pro pořádek rozděleny do dvou hlavních skupin. Jsou to informace **operativní a analytické**.

První z uvedených typů, tzv. operativní informace, která je vhodná pro ukládání obchodních a jiných transakcí, jež jsou v dané firmě prováděny. Běžně jsou tyto transakce uloženy v relační databázi a mohou ukazovat momentální stav firmy. Obvykle se děje to, že jsou data v relační databázi během dne i vícekrát změněny. Jako příklad takového systému můžeme uvést například účetnictví. Systémy, které ukládají data transakcí v reálném čase, se nazývají **OLTP** (On Line Transaction Processing) systémy. S ohledem na analytické aplikace se tato data uložená v OLTP systémech považují za **primární, zdrojová nebo také produkční**. [2]

Druhou skupinou jsou systémy, které využívají data uložená v **analytické** podobě. Tyto systémy potom využívají strukturu OLTP. [2]

Díky druhu ukládání a práci s daty nazýváme analytické systémy **OLAP – On Line Analytical Processing**. Tento název se začal běžně používat během 80. let 20. století. Pojem Business Intelligence má do určité míry spojitost také se strukturou OLAP, ale zároveň je

spojován s vývojem sady nástrojů a procesů, které podporují analytickou činnost. Díky tomu je význam OLAP o něco užší. [2]

Pokud bychom chtěli popsat OLAP po stránce čistě technologické, stala by se „informační technologií, která je založena zpravidla na koncepci multidimenzionálních databází“. Hlavním jádrem OLAP technologie je multidimenzionální tabulka, která nám dovoluje efektivně a rychle upravovat dimenze a díky tomu upravovat náhled uživatelů na představovanou ekonomickou realitu. [2]

Velkým rozdílem v charakteristice analytických systémů od OLTP systémů jsou hlavně tyto **parametry**:

- Výstupní informace jsou dodávány díky vstupům z produkčních dat.
- Data OLAP systémů jsou uložena v databázích, které mají mnohadimenzionální strukturu.
- OLAP systémy mají rozličné hladiny slučování informací, které záleží na hierarchické struktuře zadaných dimenzí.
- Důležitým prvkem je uchování faktoru času, který dovoluje provádět porovnání na základě času, předpovídat očekávaný trend požadovaných metrik, atd. [2]

Abychom mohli lépe porozumět základním principům práce analytického OLAP systému a viděli rozdíly, který jeho struktury mají vůči transakčním OLTP systémům, budeme dále v této práci pracovat s faktem, že OLAP systém bude zastřešovat standardní požadavky na vykazování, což je v současnosti ten nejběžnější a nejčastěji využívaný způsob použití analytických aplikací. [2]

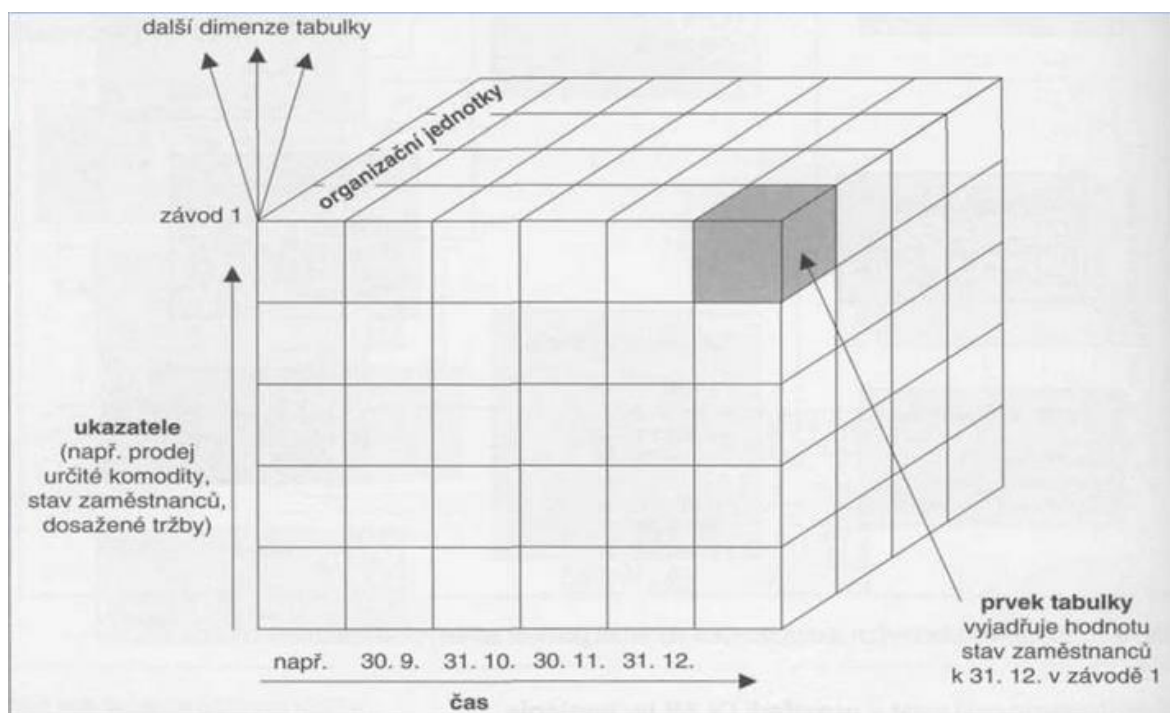
#### **1.4.1 Struktura multidimenzionální databáze**

V případě použití dat, která jsou analytického typu, není vhodné, abychom je ukládali do databází relačních, která jsou, jak je uvedeno výše, ve třetí normální formě, jež je standardem pro transakční systémy. Aby systémy OLAP měly možnost nabídnout rozličné druhy analýz a přehledů, které jsou nezbytné pro klíčové rozhodování společnosti, je třeba mít možnost pohlížet na data z různých pohledů zároveň. Základním požadavkem je mít možnost tvorby tzv. mnoharozměrných, neboli mnohadimenzionálních pohledů, které jsou v případě dat v 3NF složitě dostupná. Uživatel systému by měl být schopen využít takové nástroje, které mu dají možnost analyzovat zdrojová data takovým způsobem, jak na první pohled nemusí být vůbec jasné. Dalším problémem je nezbytnost listovat velkým objemem

informací, počítat jejich agregaci, jež v databázích, které jsou navrženy ve 3NF nejsou běžně provedeny, efektivně obměňovat datové náhledy a také tato data velmi rychle a pokud možno automaticky zapisovat do logicky navržených tabulek a grafů. [2]

**Mnohadimenzionální databáze** mají jako jednu z klíčových vlastností právě připravenost pro zapisování a efektivní práci s mnohadimenzionálními daty. Obrovským pozitivem mnoharozměrnosti, neboli využití OLAP technologie je právě vysoká efektivita, se kterou nám dovoluje analyzovat data uložená v OLAP strukturách. Mezi tyto technologie patří například *drilling* a *slice and dice*. [2]

Základním **principem**, na kterém jsou systémy BI postaveny, je **multidimenzionální tabulka**, která dovoluje velice efektivně obměňovat **své dimenze**, díky tomu nabízí uživateli různé pohledy na modelovanou ekonomickou realitu. Pro představu se tak jedná o „n-dimenzionální Rubikovu kostku“, která obsahuje ta nejvíce důležitá data společnosti – viz Obrázek 1. [2]



Obrázek 1 Princip multidimenzionální databáze. [2]

Podle nákresu na obrázku je evidentní, že jako dva **základní rozměry** zde vystupují **ukazatele (ekonomické proměnné)** a samozřejmě **čas**. Další rozměry neboli dimenze, jsou navrhovány dle aktuální potřeby. Mezi tyto dimenze mohou patřit například typ organizační jednotky, prodávaná komodita, seznam dodavatelů, lokace, atd. Obsah jednotlivých rozměrů je sestaven pomocí **prvků dimenzí**, což jsou například určité prodejny, pokladny,

nebo klienti, případně skupiny zboží, zboží a podobně. Díky sloučení všech dimenzí do jednoho konkrétního bodu můžeme vytvořit prvek mnohazměrné databáze. Jakýkoliv z těchto prvků potom může obsahovat jak data, tak algoritmy pro změnu formy dat. [2]

**Prvky dimenzí** bývají zpravidla seřazeny v **hierarchické struktuře**, znamená to, že jsou děleny na samostatné skupiny prvků, podskupiny až na jednotlivé prvky. Systémy BI potom umožňují plně automatizované agregace dat, kterými mohou být například prodejní data. Tyto agregace dat probíhají na základě logického řazení dimenzí. [2]

## 2 DATOVÝ SKLAD

Pojem BI je zařazen před téma datových skladů s ohledem k tomu, že většina systémů datových skladů vzniká hlavně proto, aby nabídla určitou podnikovou hodnotu. Tuto hodnotu obvykle představuje Business Intelligence. V následující kapitole bude blíže rozebrána definice datového skladu. [1]

### 2.1 Definice datového skladu

Datový sklad (Data Warehouse / DWH) je systém, jenž umožňuje shromažďovat, organizovat, uchovávat a sdílet historická podniková data. Obsahuje „již použitá“ data pocházející z provozních podnikových systémů, které data generují a používají tak, jak plyne z jejich funkce. Je pravdou, že existují také jiné systémy, nebo zdroje, ale zpravidla se v projektu datového skladu využívá termín „provozní systém“, anglicky „Operational system“. DWH obsahuje běžně větší množství zdrojových systémů. DWH se obvykle koncipují pro podnik jako celek, ale v mnoha případech není výjimkou, že jsou zaměřeny na určitý obor činnosti, jako jsou například finance nebo i marketing. [1]

Pojem „datový sklad“ často značí systém datového skladu a v jiných případech se může odkazovat na úložiště datového skladu. V této práci budeme termínem DWH označovat vždy komplexní systém.

Podnikoví uživatelé využívají datový sklad většinou k podpoře rozhodování. Podpora rozhodování je v této souvislosti synonymem k technologii Business Intelligence, která je založena na principu využívání dat a způsobu jejich shromažďování, ukládání a prezentace v DWH. [1]

### 2.2 Systém datového skladu

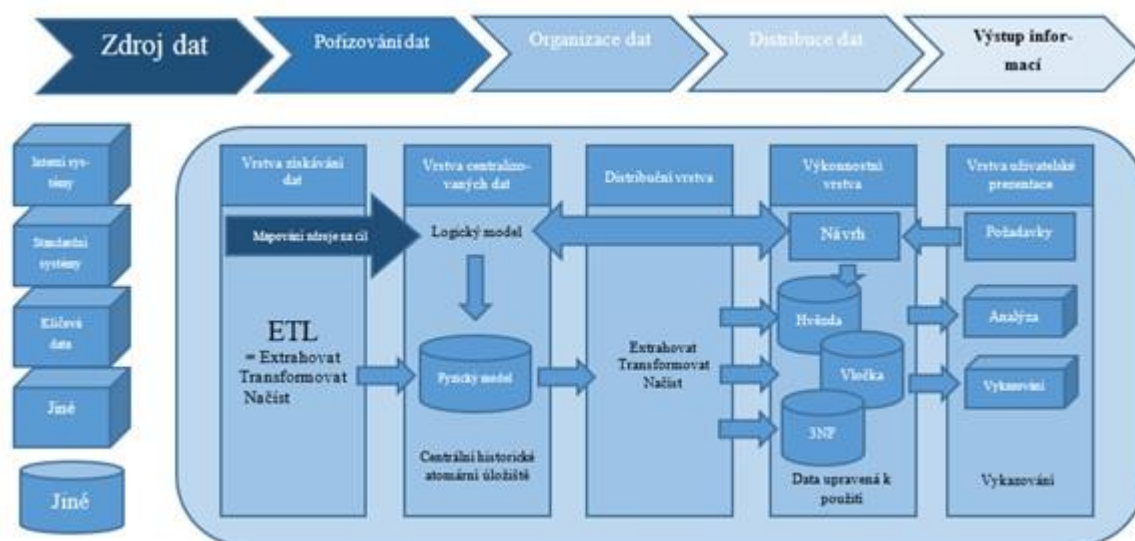
Tak jako jsou u jiných systémů klíčové komponenty vstup, zpracování, výstup a zpětná vazba, i u datových skladů pracujeme se stejnými pojmy. Vstup v systému datového skladu je úzce spojen s identifikací a zápisem získaných dat. V této fázi má nejvyšší význam vysoká kvalita dat. Jakákoli nekvalitní data totiž zákonitě povedou k nepřesným výstupům, protože se následně přenášejí do procesů, které jsou na nich závislé, subsystémů a následných vnitropodnikových analýz a rozhodnutí. Transformace a načítání dat do centrálního prostředí také může zahrnovat i fázi vstupu i zpracování. [1]

Ve středu celého systému se zpravidla nachází jedna velká databáze, ale rovněž se může jednat o celou sadu databází, které mohou být umístěny v odlehlých lokalitách a na různých serverech. Bez ohledu na nastavení by však měly být pokud možno řízeny centrálně. Uvedená centrální část tvoří výkonnou složku při transformaci a uchování dat strukturovaným a logickým způsobem. Struktura dat je založena na datové architektuře, přesněji vzato na datovém modelu. Logický datový model nám dovoluje navrhnout a popsat základní, popisné a doplňující charakteristiky dat, a tedy i celého podniku. Fyzický datový model slouží k optimalizaci dat pro jejich použití v databázi a souvisejícím prostředí. Výstupní část systému zajišťuje přenos dat uživatelům, kteří je využívají. Data lze rozdělit několika různými způsoby. Běžně se toto označuje jako kombinace datových trhů, jež mohou v závislosti na své definici přesahovat výkonnostní vrstvu a oblast nebo vrstvu uživatelské prezentace. Datové trhy jsou menší částí centrálního prostředí, které se řídí logickými nebo fyzickými centrálními datovými modely, ale jsou také optimalizovány s ohledem na specifické koncové využití. V tomto kontextu se datové trhy označují jako *výkonnostní vrstva* (performance layer). Může zahrnovat materializované dotazové tabulky vytvořené systémem, specializované přístupové cesty, virtuální pohledy atd. Oblast uživatelské prezentace popisuje část datového skladu, se kterou podnikoví uživatelé skutečně pracují. Obvykle k tomu používají některý z nástrojů pro vykazování, jako je Cognos, Microstrategy, Business Objects, Crystal Reports nebo SAS. K jiné skupině nástrojů patří Microsoft Access, ruční příkazy SQL, Excel nebo libovolná další kombinace těchto a jiných možností. Oblast uživatelské prezentace je hierarchicky umístěna nad konkrétními datovými trhy, jejichž pozice je níže. [1]

Určitá část zpětné vazby systému je založena na vstupu a výstupu. Výsledky odvození nebo případně agregace dat můžeme v budoucnu potřebovat. V tomto případě může být praktické uložit agregaci pro budoucí využití, které bude třeba například při výpočtech. Tato výstupní data jsou proto nyní požadována jako cyklický vstup systému datového skladu. Tato nově odvozená metrika nyní pochází ze samotného datového skladu a může sloužit jako základ pro budoucí dotazy. Uvedený přístup se zpětnou vazbou je samozřejmě možné optimalizovat. Příslušné postupy jsou veskrze závislé na tom, ve které fázi výstupního procesu jsou vytvořeny. [1]

## 2.3 Architektura datového skladu

Architektura datového skladu obecně popisuje návrh systému datového skladu. Architekturu bychom mohli přirovnat například k plánům stavby. Pro znázornění architektury datových skladů se nejčastěji používají diagramy toku dat, jak je patrné níže z Obrázku 2. [1]



Obrázek 2 Architektura datového skladu. [1]

Výše uvedený způsob zobrazení se stal oblíbeným proto, že nabízí dobrou představu o základních součástech. Systém datového skladu je komplexní a sofistikovaný systém. Může se na první pohled zdát, že se jedná o jednoduchý projekt, kdy stačí pouze vzít zdrojová data, umístit je do elegantně strukturované databáze a potom spustit nějaké sestavy. V reálném prostředí je však tento úkol velmi složitý a také časově náročný. [1]

Po levé straně obrázku jsou zobrazeny vstupní zdrojové systémy, což bývají obvykle provozní systémy uvnitř v podniku. Vstup také může zahrnovat zdroje zpětné vazby téměř z libovolné jiné vrstvy. Dále přicházejí různé procesy, které zaznamenávají, ukládají a organizují získaná data. Ve finále se na pravé straně nachází distribuce dat a využití nebo výstup informací určený pro podnikovou komunitu. [1]

Jak ve fázi návrhu, sestavení tak i implementace projektu je vždy třeba každou z těchto vrstev rozšířit, ale nejprve je samozřejmě nezbytné sesbírat požadavky. Klíčový bod je stanovení *účelu* projektu DWH. V případě, že podnik má vyhrazen na vývoj datového skladu zvláštní rozpočet, znamená to, že přiděluje tomuto systému nebo produktu finanční hodnotu. To je důvod, proč by měla být jeho existence – stejně tak jako u každého jiného

podnikového projektu – zdůvodněna reálným hmatatelným využitím, u kterého je možné odhadnout návratnost investic. [1]

Jsou odborníci, kteří tvrdí, že samotné strukturování dat poskytne organizaci nemalý přínos samo o sobě. S tím nelze než souhlasit, ale bez jasně daného podnikového využití není možné určit, jestli se vyplatí například věnovat 14 dní na jeden určitý parametr dat, a nikoli na jiný. Z obecného hlediska rozhodně není vhodné stavět čistě datové základy s představou, že se o ně podnik v některé další fázi začne sám zajímat. Při projektu je třeba vycházet z konkrétního a hmatatelného cíle navýšení hodnoty podniku. Základna organizovaných a strukturovaných dat by měla být vedlejším produktem přístupu a řešení, které je založeno na podnikové hodnotě. [1]

### **2.3.1 Terminologie toku dat**

Pokud se pohybujeme v oblasti datových skladů, často se používají termíny jako „shora dolů“ (top-down) a „zdola nahoru“ (bottom-up). Uvedené výrazy vycházejí z architektury toku dat. Malou pomůckou může být toto: pokud pootočíme obrázek 2 o 90 stupňů doprava, zobrazí se část se vstupem dat na „horní“ straně architektury datového skladu a část výstupu dat (tj. část podnikového použití) se dostane na „dolní“ stranu architektury. Datový tok lze rovněž pojmout s ohledem na to, zda data do systému vstupují, nebo z něj vystupují. Metodika Billa Inmona přistupuje k datovým skladům primárně shora dolů, což znamená z pohledu dat, ale je zde uveden také podnikový účel. Metodika Ralpha Kimballa naopak směřuje zdola nahoru. Tato metodika tedy řadí na první místo podnikový účel a data dle něj slouží k jeho podpoře. Hybridní přístup potom zvyšuje efektivitu projektu a jeho realizace. Obě uvedené metody ve skutečnosti směřují ke stejnému cílovému bodu, tzn. k tomu, aby podnik mohl na základě strukturování a organizace základních dat přijímat informovaná rozhodnutí. [1]

## **2.4 Účel datového skladu**

Prostředí datového skladu má takovou strukturu, aby mohlo organizovaným způsobem uchovávat historická data, která jsou integrována z více zdrojových systémů. Provozní systémy jsou navrženy pro konkrétní funkci. Jedná se například o řízení pokladních terminálů, fakturační systémy, skladové hospodářství atd. Některé z těchto systémů nepatří do podnikové třídy a nejsou vytvořeny s ohledem na analýzu nebo dolování dat. Je proto nut-

né vyvinou nové prostředí, které sloučí data z těchto systémů do jedné centrální oblasti zvané systém datového skladu, kde budou k dispozici pro celopodnikové použití. [1]



Obrázek 3 *Architektura datového skladu: Použití informací.* [1]

Potíže s kvalitou se mohou vyskytnout i v jediném systému, ale při slučování nezávislých systémů je kvalita dat zásadně důležitá. [1]

Vzhledem ke slučování dat z více systémů musí správa datového skladu věnovat pozornost modelování (tj. strukturování nebo organizování) dat, aby byl zajištěn společný slovník a pružný návrh. Systém A například označuje mužské pohlaví kódem 0 a ženské 1. Systém B pracuje s kódy pohlaví M, Ž a N (neznámé). Při kombinaci dat z těchto systémů je nutné zajistit přiřazení termínů a datových hodnot. Příslušná položka může být označena jako Pohlaví a její přípustné hodnoty budou definovány jako M (mužské), Ž (ženské) a N (neznámé). Uveden příklad je poměrně prostý, ale zeptejte se několika uživatelů v různých odděleních své organizace, jak si představují demografickou kategorii. Nepochybně dostanete mnoho různých odpovědí. [1]

Hlavní důvod pro existenci datového skladu spočívá v tom, že odstraňuje „špagetový“ zmatek, jaký ve všech organizacích vzniká kvůli kombinaci odlišných systémů, která vznikla historickým vývojem, slučováním organizací a z datových hodnot. [1]

### 3 OLAP DATABÁZE

**OLAP databáze** je ve skutečnosti sada propojených OLAP kostek, můžeme si ji představit například jako kontingenční tabulku v Excelu. OLAP kostky zpravidla obsahují již předpřipravené agregace, které jsou určeny podle hierarchické struktury dimenzí a kombinací dimenzí. [2]

Systém ve struktuře OLAP se v praxi tvoří v několika různých modifikacích, kterými jsou MOLAP, ROLAP, HOLAP a DOLAP: [2]

- **MOLAP** (Multidimensional OLAP) jedná se o základní specifické uspořádání uložených dat v multidimenzionálních OLAP kostkách, které můžeme nazývat jako binární.
- **ROLAP** (Relational OLAP) je datový model, který dokáže vyřešit multidimenzionalitu systému za pomoci uložení dat do struktur relační databáze.
- **HOLAP** (Hybrid OLAP) jedná se o určitý hybrid mezi přístupem MOLAP a ROLAP. Struktura ukládání dat je taková, že transakce jsou uloženy ve struktuře ROLAP a agregované hodnoty naopak v MOLAP kostkách.
- **DOLAP** (Desktop OLAP) je nejnovější architektura OLAP databází. Světlo světa spatřila na konci 90. let 20. století. DOLAP architektura dovoluje napojení na hlavní úložiště OLAP a dále stažení potřebné datové kostky přímo na pracovní stanici. Další práce s datovou kostkou jsou následně prováděny už přímo z počítače uživatele a není třeba vytěžovat server. Toto řešení je nejefektivnější zpravidla při použití v mobilních aplikacích. [2]

## 4 DATOVÉ TRŽIŠTĚ – DMA (DATA MART)

Pokud chceme popsat funkci datového tržiště, jedná se o velmi podobný systém, jako jsou datové sklady. Mezi hlavní rozdílné znaky patří to, že datová tržiště (Data Marts) jsou primárně určeny pro určitou skupinu uživatelů, kteří mohou být na úrovni pracovní skupiny, oddělení, divize, pobočky, nebo celé firmy. Hlavním principem jsou potom datové sklady, které je možné plynule integrovat do řešení, které využívá celý podnik. Existují i případy, kdy datové tržiště slouží dále i po vytvoření celopodnikového DWH a jsou určitým mezistupněm v procesu transformace dat z produkční databáze. [2]

Jinými slovy můžeme tedy datové tržiště popsat jako problémově orientovaný DWH, který je primárně zaměřený na řešení přesně určené problematiky dané skupiny uživatelů a který dovoluje pružnou „ad hoc“ analýzu. Ve výsledku je vytváření Data Marts v podstatě zkrácením období návratnosti investic, dále snížení nákladů a v neposlední řadě snížení možného rizika při implementaci. [2]

## 5 ETL

DWH je systém, který má svůj určitý vstup, zpracování a výstup. V následující kapitole se zaměříme přímo na vstupní aspekt systému, který se kvůli nedostatku vhodnějšího označení nazývá ETL (extract, transform, load). První proces ETL v DWH směřuje od zdroje do oblasti přípravy a ukládání. Nazývá se *pořizování dat* (data population). Druhá část zpracování ETL v DWH postupuje směrem od oblasti uchovávání do datových trhů (Data Marts) a jeho název je *distribuce dat*. Oblastí uchovávání může v této souvislosti v závislosti na použité architektuře být centrální úložiště nebo také realizační prostředí. [1]

Datový tok ETL představuje na obecné úrovni jeden z nejdůležitějších faktorů celkové architektury DWH. [1]

Data jsou aktivy každé společnosti, a její vedení proto musí zajistit nejvhodnější správu těchto podnikových aktiv. Znamená to, že data mají pro organizaci obrovskou hodnotu a v mnoha případech jsou nezbytně nutná pro její efektivní fungování. V případě špatně udržovaných nebo nekvalitních dat budou s nejvyšší pravděpodobností nesprávná i podniková rozhodování založená primárně na těchto informacích, což jsou data v souvislostech. Vedení podniku bude mít chybné podklady, a podnik proto nemusí být schopen zachovat si svůj správný strategický směr. To je jeden z důvodů proč je nezbytně důležité zajistit efektivní správu dat na co nejvyšší úrovni transparentnosti, dostupnosti a kvality. [1]

Z důvodu dosažení uvedeného cíle organizace budují systémy DWH, které dovolují používat analytické funkce BI. Základní prvek strategie datového skladu je v centralizaci podnikových dat, která je zpravidla zpočátku založena na návrzích logických datových modelů. Vlastní implementace datového skladu je zpravidla centralizovaná také, ale lze ji rovněž distribuovat po celé organizaci. V tomto případě je nutné fyzické a logické návrhy administrovat centrálně, aby byla zajištěna konzistence s terminologií a základními strukturami návrhu v organizaci jako celku. [1]

Zpracování ETL se začne uplatňovat ve fázi fyzické implementace. Ve fázi návrhu představuje ETL datový tok směrem od zdroje k cíli a v rámci tohoto toku dochází k jisté transformaci dat. ETL se následně změní na mapování dat mezi zdrojovým a cílovým systémem a dále na specifikace skutečných programů, které nakonec programátorům ETL umožní vybudovat příslušné řešení. Klíčem k tvorbě subsystému ETL je porozumění použitým zdrojovým a cílovým systémům a jejich vzájemným vztahům a rovněž pochopení časové dostupnosti dat pro zpracování. [1]

ETL zpravidla bývá nejnáročnější částí projektu DWH a ověřená poučka říká, že právě tato fáze ve výsledku představuje asi sedmdesát procent z celého úsilí na tvorbu DWH. Během posledních pár let ale velký kus práce převzali tzv. datoví architekti, kteří mají na starosti hlavně mapování a identifikaci relací mezi zdrojem a cílem. [1]

## 5.1 Architektura ETL

Z obecného hlediska je možné ETL popsat jako extrakci dat ze vstupního zdroje, transformaci extrahovaných dat do požadovaného tvaru vhodného pro načítání a finálně načítání dat do cílové databáze. Tento relativně snadný princip se však vyznačuje nemalými komplikacemi. Dále si projdeme všechny fáze z pohledu jejich architektury a následně i na detailní úrovni. [1]

Výraz „transformace“ pod sebou zahrnuje spoustu dílčích procesů a kroků, které jsou závislé na vlastních požadavcích transformace dat. Může se jednat o některé z uvedených činností: čištění, slučování, třídění, definování jedinečných identifikátorů, zajištění pořizovacích časových razítek, zajištění datových razítek období platnosti, zpracování delta, vytváření dat, ověřování dat, zajištění referenční integrity, agregování, sumarizace a profilování dat. V subsystému ETL je možné dělat jakékoliv množství uvedených kroků, což je závislé na použité architektuře a na požadavcích řešení. [1]

Transformaci dat je možné provádět na neformátovaných souborech, v databázovém systému nebo i v jejich vzájemné kombinaci. Záleží také na použitých nástrojích, objemu dat, času potřebném ke zpracování, možnostem okna zpracování (noční dávkové okno) a v neposlední řadě na závislostech dat. [1]

## 5.2 Pořizování dat

V praxi rozdělujeme vrstvu pořizování na dva hlavní aspekty:

- Aspekt získávání dat přímo ze zdroje po realizační prostředí.
- Aspekt pořizování dat od realizačního prostředí po centrální datovou vrstvu. [1]

Z obecného pohledu klade pořizování dat velké nároky na zpracování s akcentem na požadovaná data, technické prostředí, rozdíly mezi zdrojem (zpravidla provozním systémem OLTP) a datovým skladem, kterým je systém OLAP. OLTP označuje v podstatě statický návrh, jež zůstává stejný tak dlouho, dokud se nezmění konkrétní proces. Systém OLAP je mnohem dynamičtější, obzvlášť v oblasti ETL. Dodatečné analytické požadavky konco-

vých uživatelů mohou vynutit další importy ze zdrojových systémů, změny vlastních zdrojových systémů, optimalizace požadavků procesů a navýšení objemu pravidelných importů. Role architekta ETL z toho důvodu musí zajistit to, že bude k dispozici flexibilní, škálovatelné a v neposlední řadě dokumentovatelné prostředí, jež bude fungovat kontrolovaným způsobem. [1]

Velmi jednoduše řečeno, prostředí pořizování dat představuje vstupní část DWH a zpravidla vyžaduje značnou transformaci dat z velkého množství zdrojových systémů. Samotná realizační část v rámci prostředí pořizování dat může skladovat data po delší dobu, aby je následně bylo možné korektně připravit pro načtení do centrálního datového úložiště. [1]

### 5.3 Distribuce dat

Vrstva distribuce dat je dalším prostředím ETL, které slouží k extrakci dat z oblasti uchování, ať už se jedná o realizační oblast nebo centrální úložiště, převod dat do konkrétních formátů datových trhů a načtení těchto dat do jednoho nebo mnoha datových trhů. V mnoha případech bývají datové trhy navrženy ve hvězdicovém schématu, zatímco centrální úložiště se ukládá do třetí normální formy. To v praxi znamená, že transformace souvisí hlavně s generováním klíčů a převodem z historie na úrovni sloupců do historie na úrovni řádků. [1]

Teoreticky je možné datové trhy kompletně odstranit a rekonstruovat je potom z úložiště datového skladu. U datových trhů větších rozměrů se to ale stává jen velmi výjimečně, neboť by to bylo dost drahé. Data se zpravidla importují do datových trhů, a pokud již starší řádky nejsou potřebné, je možné je dle potřeby odstranit v závislosti na servisní smlouvě s podnikovými uživateli, která se týká daného datového trhu. [1]

Po zpřístupnění datových trhů podnikovým uživatelům se určitou dobu využívají, dokud se s nimi podnikoví uživatelé blíže neseznámí. V mnoha případech jsou trhy rozšířeny o další detaily, které umožní hlubší analýzu. Jakmile se podnikoví uživatelé naučí s daty lépe pracovat, začnou vyžadovat podrobnější úrovně granularity a další možnosti agregace. Uživatelé totiž hledají jiné způsoby, jak provést analýzu činnosti podniku. Vrstva distribuce dat se proto stává relativně dynamickou a k jejím změnám dochází mnohem častěji než u systémů OLTP. [1]

Architekt ETL musí velmi dobře znát podnikové požadavky na data a je jeho povinností zajistit, že bude k dispozici prostředí škálovatelné z více aspektů: objemu dat, možnosti

databázového systému, flexibility rutin ETL, délky spouštění a vybavení nástroji. Architektura ETL se musí umět přizpůsobit měnícím se datům a technologickým požadavkům, aniž by bylo nutné kompletně měnit návrh prostředí. [1]

## 6 MS SQL SERVER

Microsoft SQL Server představuje relační databázový systém poskytovaný společností Microsoft.

### 6.1 Historie a vývoj SQL Serveru

Historie produktu SQL Server se začala psát v roce 1988, kdy tento produkt ještě neměl nic společného se společností Microsoft. Dodávala ho společnost Sybase pro operační systém OS/2. V roce 1993 byla firmou Sybase uvedena verze SQL Server 4.2, což byla klasická desktopová databáze pro kanceláře a malé firmy určená pro operační systém Windows. V roce 1994 koupil tento produkt Microsoft a začal ho vyvíjet ve své režii. [3]

První verzí plně v režii Microsoftu byl v roce 1995 SQL Server 6.05, primárně určený jako databázový produkt do segmentu small business. Narostl výkon a tuto verzi už bylo možné využívat i u internetových aplikací. Verze SQL Server 6.5, která byla uvedena v roce 1996, byla určena pro platformu Windows NT. Verzi SQL Server 7.0, která přišla na trh v roce 1998, bylo možné označit přívlastkem „webová databáze“. Jádro bylo kompletně přepsáno a optimalizováno. Tento produkt už začínal nesměle konkurovat databázím Oracle a IBM DB2, hlavně svou velmi příznivou cenou. Verze SQL Server 2000 přinesla už i podporu Business Intelligence. SQL Server 2005 představoval významnou inovaci, ať už v oblasti Business Intelligence nebo XML jako nativního datového typu. [3]

SQL Server 2008 a jeho vylepšená reedice 2008 R2 podporuje kompresi údajů a záloh, umožňuje vynucení politik přes Declarative Management Framework, správu zdrojů prostřednictvím funkcionality Resource Governor a prediktivní systém optimalizace výkonu. Na serveru je možné přidat nejen paměť, ale i dodatečnou procesorovou kapacitu (Hot Add CPU) bez nutnosti vypnutí databázového serveru. Stabilita a předvídatelnost zpracování dotazů se zvyšuje i díky možnosti uzamknout plány dotazů během náhrady hardwarového serveru, inovace serveru a nasazování do provozu. [3]

V takových situacích se využívají stabilní plány dotazů. Rozšířená správa událostí zachytává, filtruje a třídí události generované serverovými procesy. To umožňuje rychlou a efektivní diagnostiku problémů za běhu serveru, monitorování zásobníků, volání procedur procedurálního jazyka T-SQL. Významnou novinkou v této verzi je i vylepšená technologie Spatial pro práci s geometrickými a geografickými údaji. Transparentní šifrování údajů umožňuje povolit zašifrování celé databáze, datových souborů, nebo souborů protokolu

bez nutnosti změny aplikací. Auditování zjednodušuje dosažení souladu s legislativními požadavky. [3]

Verze 2008 přináší i nové datové typy pro práci s datem a časem (Date, Time, DateTimeOffset, DateTime2, SmallDateTime). Datový typ Table-Valued umožňuje ukládat tabulková data. Významnou novinkou je i datový typ FileStream, který podporuje uložení velkých binárních objektů do souborového systému NTFS, zatímco tyto objekty zůstávají nedělitelnou součástí databáze. Díky integrovanému fulltextovému vyhledávání je přechod mezi prohledáváním textu a relačních dat transparentní a uživatelé mohou na rychlé prohledávání dlouhých textových sloupců využívat textové indexy. [3]

Funkce zachytávání změn dat zajišťují zaznamenávání všech změn do tabulek změn. Zaznamenává se vždy celý obsah změny, přičemž se udržuje konzistence mezi tabulkami. Tato funkce správně funguje i v případě změn schématu. Organizace tedy mohou do datového skladu integrovat ty nejaktuálnější informace. Zavedení příkazu MERGE do jazyka SQL umožňuje vývojářům efektivnější zpracování běžných situací v datových skladech, například ověření, zda určitý řádek existuje, a následné vložení či aktualizaci dat. [3]

## 6.2 SSIS

S informačními systémy úzce souvisí nejen ukládání dat, ale i jejich sběr, přesuny, import a export. Tuto problematiku je potřeba řešit téměř u každého databázového systému. Fáze zavádění údajů je také neodmyslitelnou součástí každého datového skladu. [3]

V praxi se import a export dat nejčastěji řeší jako součást komplexnějších scénářů:

- Instalace informačního systému nebo jeho databázové součásti.
- Migrace na jiný informační systém.
- Výměna databázového serveru.
- Zavádění údajů do datového skladu. [3]

Hlavní rozdíl mezi importem údajů do informačních systémů a zaváděním údajů do datového skladu spočívá v tom, že import je záležitostí jednorázovou, naproti tomu zavádění údajů do datového skladu se odehrává periodicky v určitých, například 24hodinových intervalech. I u těchto dvou scénářů je začátek velmi podobný: import údajů do informačního systému a prvotní naplnění datového skladu. A aby to nebylo tak jednoduché, nelze nezmínit archivní data obsahující historické údaje. [3]

K systémovému řešení výše uvedené problematiky slouží SQL Server Integration Services (SSIS)

### **6.3 SSAS**

SQL Server Analyses Services (SSAS) slouží ke tvorbě OLAP kostek.

### **6.4 SSRS**

Hlavní úlohou reportovacích služeb je poskytovat ve vhodné formě a včas podklady pro podporu rozhodování na všech stupních organizační infrastruktury. Hlavní důvod k nasazování reportovacích služeb je generování výstupů v elektronické nebo papírové podobě. Ty umožní zaměstnancům na všech stupních efektivní přístup k údajům a tím je podpoří v jejich činnosti. V případě manažerů jim poskytnou ve vhodné formě důležité informace, které budou potřebovat v procesu rozhodování. [3]

Ke tvorbě reportů slouží v prostředí MS SQL tzv. Reporting Services (SSRS).

## **II. PRAKTICKÁ ČÁST**

## 7 ÚVOD DO PROBLEMATIKY

Firma XY, jejíž část databáze je v práci využita, je firmou s cca 50 zaměstnanci zabývající se automatizací maloobchodu se spotřebním zbožím, speciálně se zaměřující na módu a oblast „fashion & clothes“, kde je schopna nabídnout unikátní možnosti řešení skladového hospodářství a práci se samotnými variantami zboží.

V současné době své obchodní aktivity provozuje jak na českém a slovenském trhu, tak i v řadě zemí jak v Evropě, tak v zámorí. ERP aplikace, kterou si firma sama vyvinula je připravena k napojení na e-commerce, na jiný ERP, nebo CRM systém, případně na jakýkoliv další kanál.

Data obchodní aplikace je ukládána do databáze ve 3. normální formě, která je vhodná pro ukládání transakcí do struktury OLTP. Díky této struktuře je aplikace schopna počítat nákupní ceny a hodnotu skladu velmi efektivně.

Databáze obsahuje mimo jiné tabulky pro uložení artiklů, dokladů, položek dokladů, skladové tabulky, které udržují nákupní ceny počítané metodou FIFO, zákaznickou databázi a další atributy. Pro potřeby této práce budou ve schématu zobrazeny pouze tabulky, které se přímo týkají datového skladu, nebo jsou nezbytně nutné pro vysvětlení souvislostí.

Podporované databáze jsou *MS SQL 2008* a novější, toto je hlavně z důvodu využívání datových typů *date*, které ve starších verzích nejsou podporovány (viz. Kapitola 6.1).

### 7.1 Popis nedostatků

Pro reporting je využíván samostatný systém a sekundární databáze, která obsahuje prodejní data ve formě OLTP. Každá tabulka představuje jeden týden, což je neefektivní jak při zápisu, tak při čtení, případně aktualizaci záznamů.

Systém se snaží o ukládání dat online, což není pro účely plánování přímo vyžadováno a navíc tato metoda vede, obzvláště při větším objemu transakcí, k nárůstu fronty a vyšší zátěži systému. Při generování reportů se nejprve data vyčtou z tabulek, poté se agregují a zobrazují, tento proces trvá i při generování toho nejjednoduššího reportu enormně dlouho (v řádu desítek vteřin, až jednotek minut).

Dalším problémem je možná nekonzistence dat v případě, že dojde k úpravě v produkčních datech a změní se datum transakce. Toto může narušit aktuální strukturu dat.

Databáze reportovacího systému je ve formě, se kterou není možné nativně pracovat v případě, že by bylo třeba využít standardní BI nástroje.

Souhrnně řečeno nedostatky aktuálního řešení jsou tyto:

1. Dlouhá odezva při generování reportu.
2. Vytížení systému při generování reportu.
3. Neexistuje kontrola na konzistenci dat.
4. Data jsou uložena ve formě, která znesnadňuje práci s nimi.

## 7.2 Návrh řešení

Jako řešení navrhuji vytvořit datový sklad, který bude aktualizován inkrementálně na denní bázi a data v něm budou optimalizována pro nejčastěji využívané reporty s možností dle potřeby jednoduše rozšiřovat o nové reporty.

Souhrnně řečeno mezi hlavní body návrhu řešení patří tyto:

1. Tvorba datového skladu, který bude obsahovat aktuální data v optimalizované formě.
2. Struktura dat bude v takové podobě, aby bylo možné s nimi dále pracovat i pomocí běžných BI nástrojů.

## 7.3 Struktura obchodní databáze

Hlavní data dokladů jsou uloženy v tabulkách posHeader a hlpHeader, data ohledně zákazníků jsou v tabulce posLogistical a data s položkami dokladů v tabulce posItem. Artikly jsou uloženy v tabulce fGTINVART.

Vazby mezi tabulkami jsou:

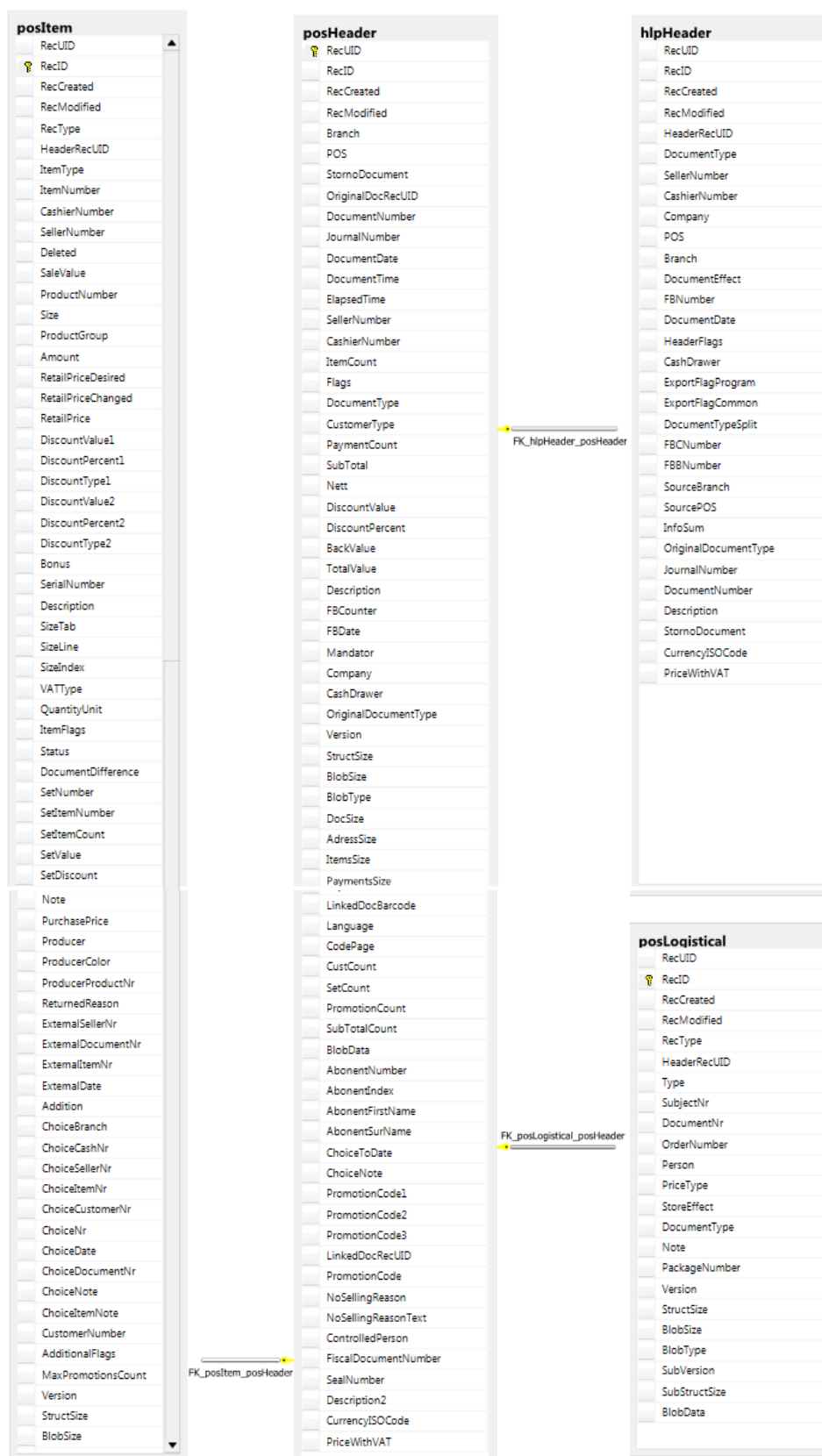
- posHeader.RecUID = hlpHeader.HeaderRecUID (1:1)
- posHeader.RecUID = posItem.HeaderRecUID (1:N)
- posHeader.RecUID = posLogistical.HeaderRecUID (1:1)

Klíče tabulek jsou:

- posHeader.RecUID (PK, uniqueidentifier)
- hlpHeader.RecID (PK, bigint)
- hlpHeader.HeaderRecUID (FK, uniqueidentifier)
- posLogistical.RecID (PK, bigint)

- posLogistical.HeaderRecUID (FK, uniqueidentifier)
- posItem.RecID (PK, bigint)
- posItem.HeaderRecUID (FK, uniqueidentifier)

## 7.4 Schéma obchodní databáze



Obrázek 4 Struktura databáze.

## 8 TVORBA DATOVÉHO SKLADU

### 8.1 Strategie

Datový sklad, který je součástí této práce má za cíl poskytnout rychlejší odezvu při generování reportů, které využívá vedení firem ke každodenním strategickým rozhodováním a vykazování. Datový sklad bude využívat inkrementální metodu importu dat na denní bázi.

Pro denní export bude sloužit úloha, kterou bude pravidelně spouštět SQL Server Agent.

Důležitým aspektem je rozsah datového skladu. Pro tento projekt bude připravena taková struktura datového skladu, která je dostatečně robustní, ale umožňuje v budoucnu pohodlné rozšíření podle aktuálních požadavků uživatelů.

Orientace datového skladu je primárně na manažery, kteří vyhodnocují informace dostupné z reportů. Sekundárně ale bude vznikat potřeba běžných uživatelů na dostupnost provozních reportů, jako je například běžný přehled prodejů.

### 8.2 Definice

Pro potřeby reportingu budou využity tabulky, které obsahují prodeje. Jedná se o tabulky uvedené v kapitole 7.1.

Artikly v DWH budou obsahovat pouze nezbytně nutné informace. V obchodní databázi je tabulka s artikly velice obsáhlá a není třeba používat všechny informace, které poskytuje. Pro potřeby datového skladu budou dostačující informace ohledně čísla zboží, velikosti a ceny.

Z exportu budou vyloučeny prodeje, které byly přerušeny, nebo stornovány, toto bude provedeno na základě flagů, které označují storno a přerušný doklad. Stejně tak budou z exportu vyloučeny položky dokladů, které byly v průběhu prodeje odstraněny z dokladu. Tyto informace nejsou předmětem tohoto projektu.

Hlavní reporty, které budou v rámci této práce vytvořeny:

- Report prodejů za posledních 7 dní.
- Report prodejů za posledních 14 dní.
- Report prodejů za posledních 30 dní.
- Report prodejů za posledních 7 dní a odpovídajících 7 dní v minulém roce.
- Report prodejů za posledních 14 dní a odpovídajících 14 dní v minulém roce.

- Report prodeje za posledních 30 dní s odpovídajícími 30 dny v minulém roce.

Každý z reportů bude mít možnost výběru prodejny a skupiny zboží.

Ve výsledku bude na stejných datech a infrastruktuře vygenerován report v původním reportovacím systému a v novém systému datového skladu. Tyto výsledky budou porovnány a na jejich základě bude vyhodnocena práce. Pro přesnost měření bude každý report spuštěn vždy 3x a čas zprůměrován.

### **8.3 Analýza**

Pro potřeby datového skladu budou vytvořeny celkem 3 databáze, stage, core a mart. To z důvodu přehlednosti architektury a také možností škálovatelnosti řešení. Kvalita dat bude zaručena pomocí kontroly pouze platných dokladů.

#### **8.3.1 Stage**

Databáze, do které budou načteny „hrubá“ data ze zdroje, a to pouze ta, která se budou používat ve výsledných reportech. Bude se jednat o data ohledně artiklů, hlaviček prodeje a položek.

#### **8.3.2 Core**

Centrální databáze datového skladu, která bude obsahovat požadované dimenze a fakta a data v nich logicky uspořádána.

#### **8.3.3 Mart**

Datový trh, který bude zobrazovat data z databáze Core formou pohledů (view). Nebude se tedy jednat o redundantní data v tabulkách.

#### **8.3.4 Dimenze a fakta**

Jako dimenze budou v projektu určeny tyto položky:

- Artikly.
- Prodejny.
- Kalendář.
- Časové období.

Fakta budou tato:

- Hlavičky prodejmů.
- Položky prodejmů.

## 8.4 Návrh

### 8.4.1 Artikly

Pro export artiklů budou použity tyto tabulky z obchodní databáze:

- setArticleGroup (skupiny artiklů)
- fGTINVRT (artikly)
- fGTINAttributeValue (atributy artiklů)
- fGTINAttributeCross (vazba na atributy)
- fskuDescriptionTexts (texty artiklů)

### 8.4.2 Prodejny

Data ohledně prodejen jsou uložena v tabulce:

- setShop

### 8.4.3 Kalendář

Tabulka kalendáře slouží k uložení informací ohledně dne, týdne, měsíce, kvartálu a roku a jejich atributy. Tato data pomáhají rychlosti generování reportů, které obsahují časové údaje. Jedná se o tato data:

- Datum.
- Den.
- První den týdne.
- Číslo dne týdne.
- Číslo dne v měsíci.
- Číslo dne v roce.
- Číslo týdne v roce.
- Číslo týdne v měsíci.
- Číslo měsíce.
- Název měsíce.

- Kvartál.
- Název kvartálu.
- Rok.
- Standardní datový formát.

#### 8.4.4 Časové období

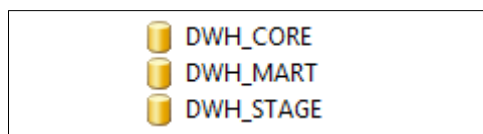
Tabulka s časovými období slouží k vyhodnocování sezónnosti artiklů. Obsahuje data ohledně začátků a konců sezón.

#### 8.4.5 Hlavičky prodejů

Hlavičky a položky prodejů jsou uvedeny v kapitole 7.1.

### 8.5 Tvorba struktur pro datový sklad

Nejprve jsou vytvořeny databáze pro datový sklad. Standardní *collation*, který používá i obchodní databáze je: *Latin1\_General\_CI\_AS*, proto použijeme stejný typ:



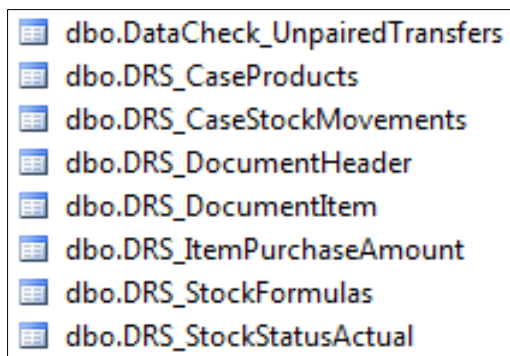
Obrázek 5 Databáze datového skladu.

#### 8.5.1 DWH\_STAGE

Tabulky v této databázi jsou v podstatě vybraná data z produkční databáze, které obsahují

Tabulky:

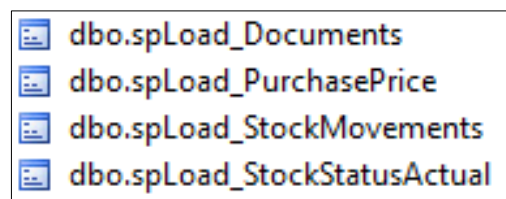
- |                                   |                                    |
|-----------------------------------|------------------------------------|
| • dbo.DataCheck_UnpairedTransfers | = nespárované transakce            |
| • dbo.DRS_CaseProducts            | = produkty ve skladových tabulkách |
| • dbo.DRS_CaseStockMovements      | = pohyby ve skladových tabulkách   |
| • dbo.DRS_DocumentHeader          | = hlavičky dokladů                 |
| • dbo.DRS_DocumentItem            | = položky dokladů                  |
| • dbo.DRS_ItemPurchaseAmount      | = nákupní cena položek             |
| • dbo.DRS_StockFormulas           | = skladové vzorce                  |
| • dbo.DRS_StockStatusActual       | = aktuální stav skladu             |



Obrázek 6 Tabulky DWH\_Stage.

#### Stored Procedures:

- `dbo.spLoad_Documents` = procedura pro načtení dokladů do Stage
- `dbo.spLoad_PurchasePrice` = procedura pro načtení nákupní ceny do Stage
- `dbo.spLoad_StockMovements` = procedura pro načtení skladových pohybů
- `dbo.spLoad_StockStatusActual` = procedura pro načtení aktuálního stavu skladu



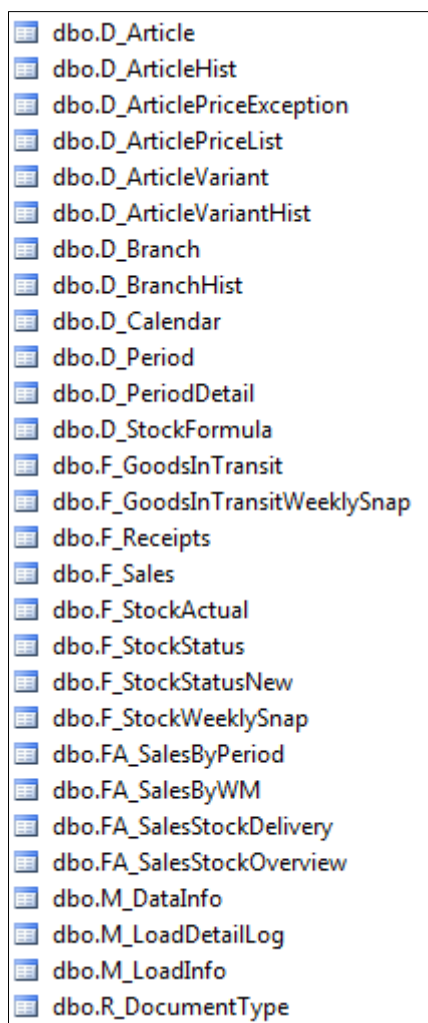
Obrázek 7 Procedury DWH\_Stage.

### 8.5.2 DWH\_CORE

#### Tabulky:

- `dbo.D_Article` = artikly
- `dbo.D_ArticleHist` = historie změn na artiklu
- `dbo.D_ArticlePriceException` = cenové výjimky pro artikly
- `dbo.D_ArticlePriceList` = ceník, ve kterém jsou zařazeny artikly
- `dbo.D_ArticleVariant` = varianty artiklů
- `dbo.D_ArticleVariantHist` = historie změn variant artiklů
- `dbo.D_Branch` = seznam prodejen
- `dbo.D_BranchHist` = historie změn prodejen
- `dbo.D_Calendar` = kalendář pro práci s daty

- `dbo.D_Period` = časové intervaly pro sezóny
- `dbo.D_PeriodDetail` = detail intervalů pro sezóny
- `dbo.D_StockFormula` = skladové vzorce
- `dbo.F_GoodsInTransit` = zboží na cestě
- `dbo.F_GoodsInTransitWeeklySnap` = zboží na cestě po týdnech
- `dbo.F_Receipts` = účtenky
- `dbo.F_Sales` = prodeje
- `dbo.F_StockActual` = aktuální stav skladu
- `dbo.F_StockStatusNew` = nový stav skladu
- `dbo.F_StockWeeklySnap` = stav skladu po týdnech
- `dbo.FA_SalesByPeriod` = prodeje po obdobích
- `dbo.FA_SalesByWM` = prodeje po týdnu a měsíci
- `dbo.FA_SalesStockDelivery` = sklad pro dodání
- `dbo.FA_SalesStockOverview` = přehled skladu



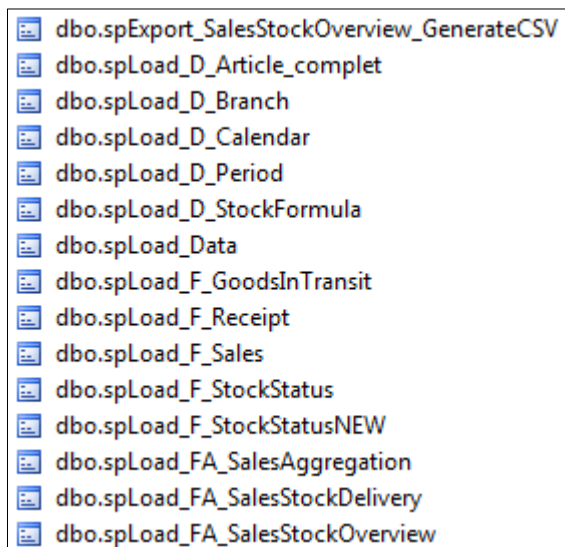
dbo.D_Article
dbo.D_ArticleHist
dbo.D_ArticlePriceException
dbo.D_ArticlePriceList
dbo.D_ArticleVariant
dbo.D_ArticleVariantHist
dbo.D_Branch
dbo.D_BranchHist
dbo.D_Calendar
dbo.D_Period
dbo.D_PeriodDetail
dbo.D_StockFormula
dbo.F_GoodsInTransit
dbo.F_GoodsInTransitWeeklySnap
dbo.F_Receipts
dbo.F_Sales
dbo.F_StockActual
dbo.F_StockStatus
dbo.F_StockStatusNew
dbo.F_StockWeeklySnap
dbo.FA_SalesByPeriod
dbo.FA_SalesByWM
dbo.FA_SalesStockDelivery
dbo.FA_SalesStockOverview
dbo.M_DataInfo
dbo.M_LoadDetailLog
dbo.M_LoadInfo
dbo.R_DocumentType

Obrázek 8 Tabulky DWH\_Core.

Procedury:

- dbo.spExport\_SalesStockOverview\_GenerateCSV = generuje csv se stavem skladu
- dbo.spLoad\_D\_Article\_complet = načítá artikly do CORE
- dbo.spLoad\_D\_Branch = načítá prodejny do CORE
- dbo.spLoad\_D\_Calendar = načítá kalendář do CORE
- dbo.spLoad\_D\_Period = načítá období do CORE
- dbo.spLoad\_D\_StockFormula = načítá skladové vzorce
- dbo.spLoad\_Data = spustí kompletní import dat
- dbo.spLoad\_F\_GoodsInTransit = načítá zboží na cestě
- dbo.spLoad\_F\_Sales = načítá prodeje do CORE
- dbo.spLoad\_F\_StockStatus = načítá stav skladu do CORE
- dob.spLoad\_F\_StockStatusNEW = načítá nový stav skladu

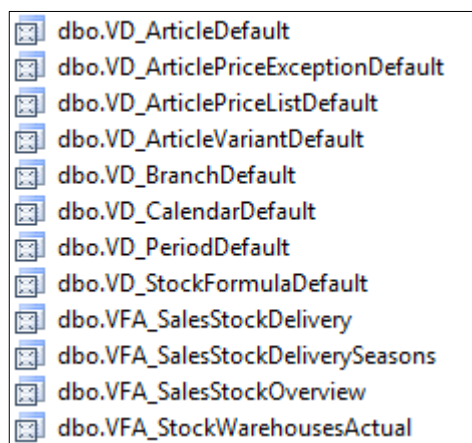
- `dbo.spLoad_FA_SalesAggregation` = načítá agregované prodej
- `dbo.spLoad_FA_SalesStockDelivery` = načítá sklad dodání
- `dbo.spLoad_FA_SalesStockOverview` = načítá přehled skladu

Obrázek 9 *Procedury DWH\_Core.*

### 8.5.3 DWH\_Mart

#### Pohledy (Views):

- `dbo.VD_ArticleDefault` = seznam artiklů
- `dbo.VD_ArticlePriceExceptionDefault` = seznam cenových výjimek artiklů
- `dbo.VD_ArticlePriceListDefault` = seznam ceníků
- `dbo.VD_ArticleVariantDefault` = seznam variant
- `dbo.VD_BranchDefault` = seznam prodejen
- `dbo.VD_CalendarDefault` = kalendář
- `dbo.VD_PeriodDefault` = sezóny
- `dbo.VD_StockFormulaDefault` = skladové vzorce
- `dbo.VFA_SalesStockDelivery` = prodeje
- `dbo.VFA_SalesStockDeliverySeasons` = prodeje po sezonách
- `dbo.VFA_SalesStockOverview` = přehled prodejů a skladů
- `dbo.VFA_StockWarehousesActual` = aktuální stav na hlavním skladu



Obrázek 10 Pohledy DWH\_Mart.

## 8.6 ETL

Data jsou exportována každou noc vždy po zavření prodejen pomocí úlohy v SQL Server Agentovi. Tato úloha spouští hlavní proceduru: **DWH\_Core.dbo.spLoadData**, která postupně spustí další požadované procedury pro import dat do datového skladu.

Jsou využívány časové značky, které určují, která data už byla vyexportována a která je třeba ještě exportovat.

Pro kompletní vyčištění a export dat používají procedury možnost použít parametr „0“, na základě kterého proběhne kompletní reexport dat. Toto řešení se doporučuje provádět pouze na začátku před naplněním datového skladu, případně při nutnosti kompletního reexportu.

## 9 REPORTING

Pomocí nástrojů *SQL Server Reporting Services* je vytvořen report prodeje, který zobrazí prodej artiklů na všech prodejnách za intervaly:

- Report prodeje za posledních 7 dní.
- Report prodeje za posledních 14 dní.
- Report prodeje za posledních 30 dní.
- Report prodeje za posledních 7 dní a odpovídajících 7 dní v minulém roce.
- Report prodeje za posledních 14 dní a odpovídajících 14 dní v minulém roce.
- Report prodeje za posledních 30 dní s odpovídajících 30 dní v minulém roce.

Všechna tato data jsou dostupná pomocí view:

DWH\_Mart.dbo.VFA\_SalesStockOverview.

Ve výsledku bude změřen čas potřebný pro generování reportu a porovnán s časem potřebným pro generování reportu v původním systému. Na základě výsledku bude vyhodnocena efektivita datového skladu.

Reporty nejsou graficky nijak optimalizovány, slouží pouze pro měření výkonu při jejich generování a jejich vzhled je základní vzhled reportů tvořených v Reporting Services.

Reporty jsou vyexportovány do xps formátu, díky čemuž ztrácejí možnost drill-down menu, které je jinak k dispozici.

### 9.1 Report prodeje v aktuálním roce

Report prodeje v aktuálním roce je vyčtení dat připravených v tabulce DWH\_Mart.dbo.VFA\_SalesStockOverview a využívá tento jednoduchý SQL skript:

```
select
    BranchNumber,
    ArticleCodeL2,
    ArticleVariantTxt1,
    ArticleGroupNumber,
    SalesQuantityLast7D,
    SalesQuantityLast14D,
    SalesQuantityLast30D
from [dbo].[VFA_SalesStockOverview]
where BranchNumber between @branchFrom and @branchTo
and ArticleGroup between @articleFrom and @articleTo
```

Kde platí, že @branchFrom je počáteční hodnota interval prodejen, @branchTo je konečná hodnota intervalu prodejen, @articleFrom je počáteční hodnota intervalu skupin zboží a @articleTo je konečná hodnota intervalu skupin zboží.

Jejich hodnoty jsou potom tyto:

- @branchFrom = 1
- @branchTo = 1
- @articleFrom = '0'
- @articleTo = 'zzzzzzzz'

To znamená, že hledáme hodnoty prodejny číslo 1 patřící do všech skupin zboží (které jsou uvnitř uvedeného intervalu 0 - zzzzzzzz).

## Report prodejů

Prodejna: 1

Skupina zboží	Varianta	Číslo zboží	Prodej -7D	Prodej -14D	Prodej -30D
1NBA			1.00000	2.00000	2.00000
1NBP			0.00000	1.00000	4.00000
1OMP			1.00000	1.00000	2.00000
1TMA			3.00000	20.00000	87.00000
1TMP			61.00000	134.00000	273.00000
1TNA			0.00000	3.00000	17.00000
1TNP			0.00000	0.00000	2.00000
1TPA			0.00000	0.00000	3.00000
1TRP			2.00000	9.00000	23.00000
1TTA			0.00000	1.00000	10.00000
1TTP			0.00000	0.00000	1.00000
3TBA			1.00000	2.00000	4.00000
3TBP			0.00000	1.00000	1.00000
3TFA			0.00000	-1.00000	1.00000
3TFP			2.00000	2.00000	3.00000
6TBA			0.00000	1.00000	5.00000
6TBP			2.00000	16.00000	28.00000
6TFI			3.00000	3.00000	7.00000
6TPA			1.00000	7.00000	10.00000
6TPI			2.00000	2.00000	3.00000

Obrázek 11 Report prodejů v aktuálním roce.

## 9.2 Report prodejmů za minulý a aktuální rok

Report prodejmů za minulý a aktuální rok čte data ze stejné tabulky, jako předchozí report. Data jsou taktéž předem připravena, a proto by ve výsledku jeho generování mělo trvat přibližně stejně dlouho. Důvodem, proč se zde tento, na první pohled redundantní, report objevuje je ten, že původní reportovací systém při generování obdobného reportu potřebuje několikanásobně delší čas, než při generování reportu za aktuální rok.

Pro potřeby reportu z datového trhu stačilo rozšířit původní skript o 3 sloupce:

```
select
    BranchNumber,
    ArticleCodeL2,
    ArticleVariantTxt1,
    ArticleGroupNumber,
    SalesQuantityLast7D,
    SalesQuantityLast7DLY,
    SalesQuantityLast14D,
    SalesQuantityLast14DLY,
    SalesQuantityLast30D
    SalesQuantityLast30DLY
from [dbo].[VFA_SalesStockOverview]
where BranchNumber between @branchFrom and @branchTo
and ArticleGroup between @articleFrom and @articleTo
```

Kdy hodnota „LY“ na konci názvu sloupce značí pojem „Last year“ (= minulý rok).

Vzhled reportu je znázorněn na Obrázku 12.

## Porovnání prodejů za aktuální a minulý rok

Prodejna: 1

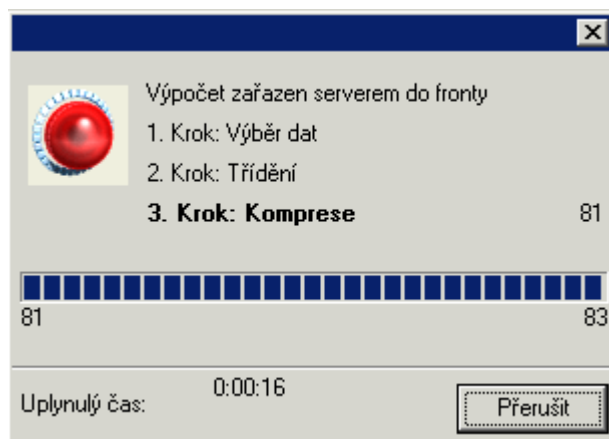
Skupina artiklů	Prodej -7D	Prodej -7D min. rok	Prodej -14D	Prodej -14D min. rok	Prodej -30D
1NBA	1.00000	0.00000	2.00000	0.00000	2.00000
1NBP	0.00000	0.00000	1.00000	0.00000	4.00000
1OMP	1.00000	0.00000	1.00000	0.00000	2.00000
1TMA	3.00000	0.00000	20.00000	4.00000	87.00000
1TMP	61.00000	1.00000	134.00000	6.00000	273.00000
1TNA	0.00000	0.00000	3.00000	0.00000	17.00000
1TNP	0.00000	0.00000	0.00000	0.00000	2.00000
1TPA	0.00000	0.00000	0.00000	0.00000	3.00000
1TRP	2.00000	0.00000	9.00000	0.00000	23.00000
1TTA	0.00000	0.00000	1.00000	0.00000	10.00000
1TTP	0.00000	0.00000	0.00000	0.00000	1.00000
3TBA	1.00000	0.00000	2.00000	0.00000	4.00000
3TBP	0.00000	0.00000	1.00000	0.00000	1.00000
3TFA	0.00000	0.00000	-1.00000	0.00000	1.00000
3TFP	2.00000	0.00000	2.00000	0.00000	3.00000
6TBA	0.00000	0.00000	1.00000	0.00000	5.00000
6TBP	2.00000	0.00000	16.00000	0.00000	28.00000
6TFI	3.00000	0.00000	3.00000	0.00000	7.00000
6TPA	1.00000	0.00000	7.00000	0.00000	10.00000
6TPI	2.00000	0.00000	2.00000	0.00000	3.00000

Obrázek 12 Report prodejů za aktuální a minulý rok.

### 9.3 Měření výkonu při generování reportu

Pro měření času při generování reportů v obou konkurenčních systémech byla po zvážení využita metoda sledování lidským okem a měření pomocí stopek. Tato metoda je dle mého názoru nejspravedlnější. Přesné důvody jsou uvedeny v závěru práce.

Při generování reportů z původního systému bylo nápomocno také okno, které zobrazuje průběh a čas tvorby reportu viz Obrázek 13.



Obrázek 13 *Průběh generování reportů v původním systému.*

Pro eliminaci možných chyb měření a nerovnoměrnému zatížení systému v čase bylo každé měření provedeno 3x a výsledky zprůměrovány.

V následující Tabulce 1 jsou zachyceny výsledky měření. Čas je zaokrouhlen na celá čísla:

Tabulka 1 *Výsledky měření času potřebného pro generování reportu.*

Měření	Původní reportovací systém - aktuální rok	Nové řešení - aktuální rok	Původní reportovací systém - minulý rok	Nové řešení - minulý rok
Měření 1 t [s]	16	<1	29	<1
Měření 2 t [s]	18	<1	42	<1
Měření 3 t [s]	16	<1	25	<1
Průměr měření t [s]	17	<1	32	<1

## ZÁVĚR

Tato práce se zabývala vyhodnocením aktuálního reportovacího systému a návrhem nového řešení datového skladu, které by pomohlo k rychlému přístupu k požadovaným informacím bez nutnosti dlouhého čekání, než se report vygeneruje. Tento problém je zákazníky často připomínán, protože v extrémních případech může zabrat tvorba reportu i několik hodin a během této doby je server výrazně vytížen.

Jako jeden z nejslabších bodů původního systému považuji jeho strukturu, kdy každý týden je uložen v jedné samostatné tabulce. V době tvorby tohoto systému bylo toto řešení pravděpodobně vyhovující, nicméně pro aktuální potřeby už výrazně zaostává. Dalším bodem je online export dat přímo do systému, což opět zatěžuje systém a v některých případech dokonce fronta pro export naroste tak, že trvá i hodiny, než vše proběhne. Dle mého názoru není nutné data držet uložena online přímo v reportovací vrstvě, ale stačí provádět pravidelný, například noční inkrementální export a v případě potřeby čtení z dat, která jsou uložena v období mezi dobou „nyní“ a posledním exportem vytvořit vhodnou strukturu, která data vyčte ze systému a přidá k již připraveným reportům.

Pro vyhodnocení efektivity reportingu jsem použil testovací databázi, která obsahuje historii dokladů cca 3 roky, jedná se o 85.000 prodejních dokladů, celkem je v dokladech použito na 25.000 artiklů a 250.000 variant artiklů, což v tomto případě znamená velikost a barva. Měření ve starém reportovacím systému a nově navrženém datovém skladu probíhalo na stejném stroji, jedná se o starší server s procesorem Intel Xeon X3430 běžící na frekvenci 2.39 GHz a využívající 8 GB RAM. Databáze, na které běží oba systémy, je Microsoft SQL Server 2014 – 12.0.2000.8 (X64).

Při generování reportu jsem nejprve uvažoval měřit a porovnat čas pro generování skriptu v SQL, další myšlenkou bylo kontrolovat čas generování přímo v Reporting Services pomocí kódu v programovacím jazyku Visual Basic, ale po zvážení všech pro a proti jsem se rozhodl pro měření času stopkami, protože není možné jednoduše porovnat proti sobě dva systémy na naprosto odlišné bázi, kdy každý z nich jinou dobu čte data a jinou dobu generuje zobrazení. Myslím si, že využití „uživatelského“ náhledu je v oblasti reportingu zaměřenému právě na uživatele ta nejvhodnější varianta.

Ve výsledku dopadlo v měření dle očekávání mnohem lépe řešení navrženého datového skladu oproti původnímu systému. Důvodem bylo to, že datový sklad má požadovaná data již připravena, zatímco starý systém je nejprve čte, agreguje a řadí, což způsobuje velkou

prodlevu a vytižení systému. Na druhou stranu je třeba podotknout, že řešení původního reportovacího systému dovoluje tvorbu určitých datových náhledů, které je v projektu datového skladu třeba vytvořit.

## SEZNAM POUŽITÉ LITERATURY

- [1] LABERGE, Robert. Datové sklady: agilní metody a business intelligence. 1. vyd. Brno: Computer Press, 2012, 350 s. ISBN 978-80-251-3729-1.
- [2] NOVOTNÝ, Ota, Jan POUR a David SLÁNSKÝ. Business intelligence: jak využít bohatství ve vašich datech. 1. vyd. Praha: Grada, 2005, 254 s. ISBN 8024710943.
- [3] LACKO, Ľuboslav. Mistrovství v SQL Server 2012: [kompletní průvodce databázového experta]. 1. vyd. Brno: Computer Press, 2013, 640 s. ISBN 978-80-251-3773-4.
- [4] KNIGHT, Brian. Professional SQL server 2008 integration services. Indianapolis, IN: Wiley Pub., c2008, xxxiii, 972 p. ISBN 0470247959.
- [5] LARSON, Brian. Microsoft SQL Server 2012 reporting services. New York: McGraw-Hill, c2012, xx, 762 p. ISBN 0071760474.
- [6] HESTER, Matthew a Chris HENLEY. Microsoft windows server 2012 administration: instant reference. Indianapolis, Ind.: Sybex, c2013, 1 online zdroj (xviii, 574 p.).
- [7] KROENKE, David a David J AUER. Databáze. 1. vyd. Brno: Computer Press, 2015, 496 s. ISBN 978-80-251-4352-0.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

BI	Business Intelligence.
DWH	Datový sklad.
OLAP	Online Analytical Processing.
OLTP	Online Transaction Processing.
MS	Microsoft.
SQL	Structured Query Language.
SSIS	SQL Server Integrační Služby.
SSAS	SQL Server Analytické Služby.
SSRS	SQL Server Reportovací Služby.
DPH	Daň z přidané hodnoty.
SWOT	Strengths, Weaknesses, Opportunities, Threats.
PEST	Political, Economic, Social and Technological analysis.
KPI	Key Performance Indicator.
ODS	Operational data store.
tzv.	Takzvaná(ý).
atd.	A tak dále.
3NF	Třetí normální forma.
SAS	Statistical Analysis Systém.
MOLAP	Multidimensional OLAP.
ROLAP	Relational OLAP.
HOLAP	Hybrid OLAP.
DOLAP	Desktop OLAP.
DMA	Data Mart.
ETL	Extract, Transform, Load.

XML	eXtensible Markup Language.
CPU	Central Processing Unit.
NTFS	New Technology File System.
ERP	Enterprise Resource Planning.
CRM	Customer Relationship Management.
FIFO	First in first out.

**SEZNAM OBRÁZKŮ**

Obrázek 1 <i>Princip multidimenzionální databáze. [2]</i> .....	14
Obrázek 2 <i>Architektura datového skladu. [1]</i> .....	18
Obrázek 3 <i>Architektura datového skladu: Použití informací. [1]</i> .....	20
Obrázek 4 <i>Struktura databáze.</i> .....	34
Obrázek 5 <i>Databáze datového skladu.</i> .....	38
Obrázek 6 <i>Tabulky DWH_Stage.</i> .....	39
Obrázek 7 <i>Procedury DWH_Stage.</i> .....	39
Obrázek 8 <i>Tabulky DWH_Core.</i> .....	41
Obrázek 9 <i>Procedury DWH_Core.</i> .....	42
Obrázek 10 <i>Pohledy DWH_Mart.</i> .....	43
Obrázek 11 <i>Report prodejů v aktuálním roce.</i> .....	45
Obrázek 12 <i>Report prodejů za aktuální a minulý rok.</i> .....	47
Obrázek 13 <i>Průběh generování reportů v původním systému.</i> .....	48

**SEZNAM TABULEK**

Tabulka 1 <i>Výsledky měření času potřebného pro generování reportu. ....</i>	48
---	----

## SEZNAM PŘÍLOH

Příloha P1: DWH\_CORE.sql

Příloha P2: DWH\_MART.sql

Příloha P3: DWH\_STAGE.sql