



## ZADÁNÍ DIPLOMOVÉ PRÁCE

|                          |  |
|--------------------------|--|
| <b>Název:</b>            | Analýza datového skladu a aplikace pro reportování |
| <b>Student:</b>          | Bc. Jakub Trnka                                    |
| <b>Vedoucí:</b>          | Mgr. Matyáš Bittner                                |
| <b>Studijní program:</b> | Informatika  |
| <b>Studijní obor:</b>    | Webové a softwarové inženýrství                    |
| <b>Katedra:</b>          | Katedra softwarového inženýrství                   |
| <b>Platnost zadání:</b>  | Do konce zimního semestru 2018/19                  |

### Pokyny pro vypracování

1. Nastudujte problematiku datových skladů vzhledem k rozsahu a možnostem jejich použití ve společnosti MEDICON Pharm s.r.o.
2. Analyzujte a navrhněte úpravy datového skladu ve výše zmíněné společnosti.
3. Navrhněte aplikaci pro reportování nad daným datovým skladem.
4. Implementujte prototyp této aplikace.
5. Otestujte aplikaci, vyhodnoťte její přínos a na základě výsledků připravte plán dalšího vývoje.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
děkan

V Praze dne 21. února 2017



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

## **Analýza datového skladu řetězce lékáren a aplikace pro reportování**

*Bc. Jakub Trčka*

Vedoucí práce: Mgr. Matyáš Bittner

3. května 2017



---

## Poděkování

Chtěl bych poděkovat všem lidem, kteří se pohybují kolem společnosti Medicon Pharm s.r.o., v čele s vedoucím této práce Matyášem Bittnerem.

Bez odpovědí na nekonečné počty mých otázek by nevznikl ani jediný odstavec.

Zároveň dlužím obrovské poděkování svým blízkým za podporu a nekončící trpělivost.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 3. května 2017

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2017 Jakub Trčka. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Trčka, Jakub. *Analýza datového skladu řetězce lékáren a aplikace pro reportování*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Práce se zaměřuje na proces zavedení nástroje pro datovou analytiku v malém řetězci lékáren. Proces začíná analýzou podniku a jeho datových zdrojů. Část práce je věnována teoretickému základu v oblasti datových skladů.

S využitím teoretického základu a poznatků z provedené analýzy je navrženo a částečně realizováno řešení datového skladu dle R. Kimballa.

Výstupem práce je návrh webové reportovací aplikace. Aplikace umožňuje vytvářet uživatelské reporty nad datovou kostkou, která je sestavena nad daty z datového skladu.

Součástí práce je implementace prototypu této webové aplikace a nastavení rozhraní datového skladu s využitím Internet Information Services a doplňku Analysis Services.

Na závěr jsou shrnuty poznatky z prezentace prototypu vedení společnosti a je naznačen plán dalšího vývoje.

**Klíčová slova** Datové sklady, SQL Server, Business Intelligence, datová kostka, MDX jazyk, multidimenzionální modelování, reportování.

# Abstract

The thesis focuses on the process of implementation of the data analysis tool in the retail chain of pharmacies. The process starts with the analysis of a company and its data sources.

The part of the work deals with the teoretical basis in the field of data warehousing.

As the result of data analysis, the former solution was transformed into the instance of the data warehouse by R. Kimball's design.

The work output is the architecture of the reporting web application. The application allows user reporting over the data cube which is set up on the data warehouse.

Another component of the work is the prototype implementation of the web application and setting of data warehouse interface, using Internet Information Services and Analysis Services module.

Finally, the feedback from the prototype presentation for the company management are summarized and a plan for further development is outlined.

**Keywords** Data warehouse, SQL Server, Business Intelligence, data cube, MDX language, multidimensional modeling, reporting.

---

# Obsah

|   |          |
|---|----------|
| Odkaz na tuto práci . . . . .                         | viii     |
| <b>Úvod</b>   | <b>1</b> |
| <b>1 Popis problémové domény</b>                      | <b>3</b> |
| 1.1 O společnosti . . . . .                           | 3        |
| 1.2 Používané technologie . . . . .                   | 3        |
| 1.2.1 Vybavení uživatelských stanic . . . . .         | 3        |
| 1.2.2 Propojení sítě . . . . .                        | 4        |
| 1.2.3 Interní firemní systémy . . . . .               | 4        |
| 1.3 Požadavky managementu . . . . .                   | 4        |
| <b>2 Datové sklady</b>                                | <b>7</b> |
| 2.1 Definice a architektura datových skladů . . . . . | 8        |
| 2.1.1 Datové sklady podle W. H. Inmona . . . . .      | 8        |
| 2.1.2 Datové sklady podle R. Kimballa . . . . .       | 11       |
| 2.2 Struktura a návrh datových skladů . . . . .       | 11       |
| 2.2.1 Multidimenzionální datový model . . . . .       | 12       |
| 2.2.2 Multidimenzionální modelování . . . . .         | 17       |
| 2.2.3 Dimenze a změny . . . . .                       | 18       |
| 2.3 Datové pumpy ETL . . . . .                        | 19       |
| 2.3.1 Extrakce . . . . .                              | 19       |
| 2.3.2 Transformace . . . . .                          | 19       |
| 2.3.3 Loading . . . . .                               | 22       |
| 2.4 OLTP a OLAP . . . . .                             | 22       |
| 2.4.1 Datové kostky . . . . .                         | 22       |
| 2.4.2 OLAP operace . . . . .                          | 24       |
| 2.5 Dotazovací jazyk MDX . . . . .                    | 24       |
| 2.5.1 Práce s množinou . . . . .                      | 25       |
| 2.5.2 Počítané členy a funkce CurrentMember . . . . . | 26       |

|          |  |           |
|----------|--|-----------|
| <b>3</b> | <b>Analýza současného řešení</b>             | <b>29</b> |
| 3.1      | Produkční prostředí . . . . .                | 29        |
| 3.1.1    | Instance systému FaRMIS . . . . .            | 30        |
| 3.1.2    | MS SQL Server . . . . .                      | 30        |
| 3.2      | Datový model databáze . . . . .              | 30        |
| 3.3      | Datové pumpy, nástroje ETL . . . . .         | 32        |
| 3.3.1    | Greyfox . . . . .                            | 32        |
| 3.3.2    | MIS . . . . .                                | 32        |
| 3.3.3    | CDB – centrální databáze . . . . .           | 32        |
| 3.3.4    | CRK – centrální registr kont . . . . .       | 34        |
| 3.3.5    | FaRMIS, uniexport . . . . .                  | 34        |
| <b>4</b> | <b>Transformace datového skladu</b>          | <b>41</b> |
| 4.1      | Změna produkčního prostředí . . . . .        | 41        |
| 4.2      | Architektura datového skladu . . . . .       | 42        |
| 4.3      | Nový datový model . . . . .                  | 42        |
| 4.3.1    | Volba business procesů . . . . .             | 43        |
| 4.3.2    | Definice zrnitosti – granularita . . . . .   | 43        |
| 4.3.3    | Volba dimenzí . . . . .                      | 44        |
| 4.3.4    | Identifikace faktů . . . . .                 | 46        |
| 4.4      | Druhý modul ETL . . . . .                    | 48        |
| 4.4.1    | Dimenze ATC . . . . .                        | 48        |
| 4.4.2    | Dimenze Dodavatel . . . . .                  | 50        |
| 4.4.3    | Dimenze Kategorie . . . . .                  | 50        |
| 4.4.4    | Dimenze Podskupina . . . . .                 | 51        |
| 4.4.5    | Dimenze Produkt . . . . .                    | 51        |
| 4.4.6    | Dimenze Provozovna . . . . .                 | 52        |
| 4.4.7    | Dimenze TypDokladu . . . . .                 | 53        |
| 4.4.8    | Dimenze TypPlatby . . . . .                  | 54        |
| 4.4.9    | Dimenze Uživatel . . . . .                   | 54        |
| 4.4.10   | Dimenze Vyrobcce . . . . .                   | 55        |
| 4.4.11   | Dimenze Zakaznik . . . . .                   | 55        |
| 4.5      | Implementace změn . . . . .                  | 56        |
| 4.5.1    | Nastavení SQL serveru . . . . .              | 56        |
| 4.5.2    | Vytvoření databáze . . . . .                 | 57        |
| 4.5.3    | Testovací data . . . . .                     | 57        |
| <b>5</b> | <b>Návrh webové aplikace pro reportování</b> | <b>61</b> |
| 5.1      | Existující aplikace . . . . .                | 61        |
| 5.1.1    | WebManager společnosti Apatyka . . . . .     | 62        |
| 5.1.2    | Google Analytics . . . . .                   | 63        |
| 5.1.3    | Výstupy . . . . .                            | 65        |
| 5.2      | Vymezení požadavků na aplikaci . . . . .     | 65        |
| 5.2.1    | Požadavky na produkční prostředí . . . . .   | 65        |

|          |  |            |
|----------|--|------------|
| 5.2.2    | Požadavky na funkcionalitu aplikace . . . . .      | 66         |
| 5.3      | Modelování případů užití (Use cases) . . . . .     | 67         |
| 5.4      | Návrh UI . . . . .                                 | 75         |
| 5.4.1    | Modelování funkcí uživatelského rozhraní . . . . . | 75         |
| 5.4.2    | Wireframy uživatelského rozhraní . . . . .         | 81         |
| 5.5      | Návrh architektury . . . . .                       | 82         |
| 5.5.1    | Komponenty aplikace . . . . .                      | 82         |
| 5.5.2    | Přístup k datům . . . . .                          | 83         |
| 5.6      | Technologie . . . . .                              | 85         |
| 5.6.1    | Jádro aplikace a Nette framework . . . . .         | 85         |
| 5.6.2    | Zpracování a zobrazení dat – Javascript . . . . .  | 85         |
| 5.6.3    | Interní MySQL, zdrojový SQL Server . . . . .       | 86         |
| <b>6</b> | <b>Prototyp webové aplikace pro reportování</b>    | <b>87</b>  |
| 6.1      | Rozhraní datového skladu . . . . .                 | 87         |
| 6.1.1    | Vytvoření datové kostky . . . . .                  | 88         |
| 6.1.2    | HTTP/HTTPS rozhraní s pomocí IIS . . . . .         | 88         |
| 6.2      | Uživatelské prostředí . . . . .                    | 90         |
| 6.2.1    | Grafické zpracování . . . . .                      | 90         |
| 6.2.2    | Šablonovací systém . . . . .                       | 91         |
| 6.3      | Datová vrstva . . . . .                            | 92         |
| 6.4      | Workflow a integrace komponent . . . . .           | 93         |
| 6.5      | Připravené reporty . . . . .                       | 95         |
| 6.5.1    | Report výsledků po provozovnách . . . . .          | 95         |
| 6.5.2    | Report výkonu společnosti . . . . .                | 97         |
| <b>7</b> | <b>Plán dalšího rozvoje BI řešení</b>              | <b>99</b>  |
| 7.1      | Budování datového skladu . . . . .                 | 99         |
| 7.2      | Vývoj reportovací aplikace . . . . .               | 100        |
| 7.2.1    | Poznatky z prezentace . . . . .                    | 100        |
| 7.2.2    | Proces vývoje . . . . .                            | 100        |
|          | <b>Závěr</b>                                       | <b>103</b> |
|          | <b>Literatura</b>                                  | <b>105</b> |
|          | <b>A Seznam použitých zkratk</b>                   | <b>109</b> |
|          | <b>B Obsah příloženého CD</b>                      | <b>111</b> |



---

## Seznam obrázků

|      |   |    |
|------|---|----|
| 2.1  | Požadavek na stálost datového skladu . . . . .                    | 8  |
| 2.2  | Požadavek na integraci datového skladu . . . . .                  | 9  |
| 2.3  | Architektura datového skladu podle Inmona . . . . .               | 10 |
| 2.4  | Architektura datového skladu podle Kimballa . . . . .             | 12 |
| 2.5  | Porovnání dimenzionální tabulky a tabulky faktů . . . . .         | 13 |
| 2.6  | Ukázka generování reportu z hvězdicového schématu . . . . .       | 14 |
| 2.7  | Schéma datového skladu - star . . . . .                           | 15 |
| 2.8  | Schéma datového skladu - snowflake . . . . .                      | 16 |
| 2.9  | Schéma datového skladu - konstalace faktů . . . . .               | 16 |
| 2.10 | Ukázka datové kostky pro 3 dimenze. . . . .                       | 23 |
|      |   |    |
| 3.1  | Schéma IT systémů . . . . .                                       | 31 |
| 3.2  | Datový model současného řešení, Bowbeen . . . . .                 | 33 |
| 3.3  | Přehled Java ETL aplikací . . . . .                               | 34 |
| 3.4  | Popis tříd ETL uzlu pro zpracování uniexportu z FaRMISu . . . . . | 35 |
|      |   |    |
| 4.1  | Schéma IT systémů . . . . .                                       | 43 |
| 4.2  | Dimenzionální datový model tržiště Bowbeen DMart. . . . .         | 47 |
| 4.3  | Hlavní proces 2. ETL uzlu. . . . .                                | 48 |
| 4.4  | ETL metoda pro dimenzi ATC. . . . .                               | 50 |
| 4.5  | ETL metoda pro dimenzi Dodavatel. . . . .                         | 50 |
| 4.6  | ETL metoda pro dimenzi Kategorie. . . . .                         | 51 |
| 4.7  | ETL metoda pro dimenzi Podskupina . . . . .                       | 51 |
| 4.8  | ETL metoda pro dimenzi Produkt. . . . .                           | 52 |
| 4.9  | ETL metoda pro dimenzi Provozovna. . . . .                        | 53 |
| 4.10 | ETL metoda pro dimenzi TypDokladu. . . . .                        | 54 |
| 4.11 | ETL metoda pro dimenzi TypPlatby. . . . .                         | 54 |
| 4.12 | ETL metoda pro dimenzi Uživatel. . . . .                          | 54 |
| 4.13 | ETL metoda pro dimenzi Výrobce. . . . .                           | 55 |
| 4.14 | ETL metoda pro dimenzi Zakaznik. . . . .                          | 56 |

|      |   |    |
|------|---|----|
| 4.15 | Přehled sql skriptů pro datové tržiště . . . . .                      | 57 |
| 5.1  | Uživatelské prostředí WebManageru . . . . .                           | 63 |
| 5.2  | Uživatelské prostředí Google Analytics . . . . .                      | 64 |
| 5.3  | Diagram případů užití webové aplikace . . . . .                       | 68 |
| 5.4  | Graf popisující hlavní kostru aplikace. . . . .                       | 76 |
| 5.5  | Graf popisující část aplikace pro specifikaci segmentu. . . . .       | 78 |
| 5.6  | Graf popisující část aplikace pro výběr metrik. . . . .               | 79 |
| 5.7  | Graf popisující část aplikace pro výběr dimenzí. . . . .              | 80 |
| 5.8  | Graf popisující část aplikace pro výběr časové dimenze. . . . .       | 80 |
| 5.9  | Graf popisující část aplikace pro změnu zobrazení dat. . . . .        | 80 |
| 5.10 | Wireframe model pro stránku s reportem. . . . .                       | 81 |
| 5.11 | Model architektury aplikace. . . . .                                  | 83 |
| 5.12 | Základní datový model interní aplikační databáze. . . . .             | 84 |
| 6.1  | Vytvoření datové pumpy v aplikačním rozhraní IIS. . . . .             | 89 |
| 6.2  | Soubory modulu MSMDPUMP.dll . . . . .                                 | 90 |
| 6.3  | Rozhraní datového tržiště . . . . .                                   | 91 |
| 6.4  | Přehled komponent prototypu . . . . .                                 | 94 |
| 6.5  | Přehled modulů prototypu aplikace pro reportování . . . . .           | 94 |
| 6.6  | Bowbeen DMart, obrazovka s reportem výsledků po provozovnách. . . . . | 95 |
| 6.7  | Bowbeen DMart, obrazovka s reportem výkonu společnosti . . . . .      | 96 |



---

## Seznam tabulek

|      |   |    |
|------|---|----|
| 2.1  | ETL transformace formátu data. . . . .                          | 20 |
| 2.2  | ETL odstranění duplicitních záznamů . . . . .                   | 20 |
| 2.3  | ETL rozdělení položek . . . . .                                 | 21 |
| 3.1  | Popis uniexportu . . . . .                                      | 36 |
| 4.1  | Atributy produktové dimenze. . . . .                            | 46 |
| 4.2  | Přehled parametrů tabulky faktů dimenzionálního modelu. . . . . | 49 |
| 4.3  | Přehled parametrů generátoru testovacích dat. . . . .           | 59 |
| 5.1  | Případ užití, UC0: EditUsers . . . . .                          | 69 |
| 5.2  | Případ užití, UC0.1: EditUser . . . . .                         | 69 |
| 5.3  | Případ užití, UC4: Login . . . . .                              | 70 |
| 5.4  | Případ užití, UC1: RunCustomAnalysis . . . . .                  | 70 |
| 5.5  | Případ užití, UC1.1: SaveReport . . . . .                       | 71 |
| 5.6  | Případ užití, UC1.2: ChooseSegment . . . . .                    | 72 |
| 5.7  | Případ užití, UC1.3: ChooseDatePeriod . . . . .                 | 72 |
| 5.8  | Případ užití, UC1.4: ChooseMetrics . . . . .                    | 73 |
| 5.9  | Případ užití, UC1.5: ChooseDimensions . . . . .                 | 73 |
| 5.10 | Případ užití, UC1.6: ChangeViewStyle . . . . .                  | 74 |
| 5.11 | Případ užití, UC1.10: ExportData . . . . .                      | 75 |



---

# Úvod

Zdravotnictví se dlouhou dobu příliš neúčastnilo technologické revoluce, která se rozvinula v oblasti internetu, nových typů inteligentních zařízení a s oběma tématy spojené datové analytiky. O to větší je v posledních letech tlak na rychlou implementaci všech moderních řešení, která by mohla přinést konkurenční výhodu.

Cílem této práce je pokusit se implementovat jednu z těchto technologií do společnosti, která provozuje menší řetězec lékáren. Konkrétně se jedná o vybudování zázemí pro datový sklad a návrh a implementaci prototypu nástroje pro reportování nad tímto datovým skladem.

V úvodní části bude představena problémová doména. Čtenář bude seznámen se společností Medicon Pharm s.r.o., která je provozovatelem lékárny, a s očekávanými managementu společnosti směrem k projektu datové analytiky.

Následovat bude teoretická část, která popíše ty nejdůležitější poznatky v oblasti datových skladů a multidimenzionálního modelování. Nebudou chybět dva základní pohledy na architekturu firemních datových uložišť a metody jejich návrhu a provozu.

Součástí projektu bude také analýza současné datové situace v lékárnách společnosti, včetně rozboru aktuálních datových modelů.

Navazující kapitola navrhne, jaké změny by společnost měla učinit směrem k využití potenciálu své datové základny v oblasti Business Intelligence. K těmto změnám budou využity metody a poznatky z teoretického základu této práce.

Jakmile bude připravena datová základna pro analytické úlohy, přijde na řadu návrh reportovací aplikace. Aplikace bude navržena jako průnik nejběžnějších analytických nástrojů, požadavků managementu společnosti a specifík, které s sebou přináší daná business doména.

Práce bude ukončena přípravou prototypu reportovací aplikace, na které budou managementu společnosti představeny možnosti datové analytiky. Součástí prezentace prototypu je zpětná vazba od vedení společnosti a příprava na další fázi projektu, která by měla přinést plnohodnotný produkt.



---

# Popis problémové domény

Tato práce má za cíl zmapovat požadavky managementu skupiny společností ze segmentu zdravotnictví a to v oblasti Business Intelligence. Přípravě tohoto projektu předcházela debata s vedením společnosti, ze které vyvstaly požadavky na řízení založeném na skutečných datech.

## 1.1 O společnosti

Společnost Medicon Pharm s.r.o. je součástí skupiny Medicon a.s., která sdružuje několik společností napříč různými oblastmi zdravotní péče. Společnost Medicon Pharm s.r.o. jako taková se konkrétně zaměřuje na provoz kamenných lékáren, obchodů se zdravou výživou a v neposlední řadě doplňuje své aktivity dvěma internetovými obchody. Nově od minulého roku je portfolio společnosti rozšířeno o nutriční poradnu.

Konkrétně jsou pod hlavičkou společnosti čtyři kamenné lékárny v Praze různých velikostí - od malé lékárny v obchodním centru po největší kamennou lékárnu u nás - Pharmacentrum Budějovická.

Internetové obchody provozuje společnost od roku 2014, kdy vznikla nutri-vyziva.cz. Od druhé poloviny roku 2015 se přidal obchod mojetabletka.cz, nabízející volně prodejné léčivé přípravky, doplňky stravy a kosmetiku.

## 1.2 Používané technologie

### 1.2.1 Vybavení uživatelských stanic

Uživatelské stanice jsou vybaveny operačním systémem Windows 7 a nebo Windows 10. Mezi nejpoužívanější aplikace patří běžné kancelářské programy jako MS Excel, Word a Internet Explorer. Někteří uživatelé používají také internetový prohlížeč Google Chrome.

### 1.2.2 Propojení sítě

Pro účely tohoto projektu jsou důležitá převážně pracoviště společnosti v budově polikliniky a lékárny na ulici Antala Staška v Praze. V těchto prostorách sídlí vedení společnosti, a tím pádem i všichni uživatelé chystaných aplikací. V rámci těchto pracovišť je potřeba rozlišit historicky tři společnosti:

- Medicon a.s.,
- Medicon Pharm s.r.o.,
- DTC clinic s.r.o. (již neexistující).

Společnost Medicon a společnost DTC clinic, respektive historické oddělení těchto společností, spadají v rámci počítačové sítě každá do své vlastní domény<sup>1</sup>. Společnost Medicon Pharm s.r.o. je v plánu přiřadit do domény MDCN (Medicon a.s.), ale v současné době je spojena pouze v rámci pracovní skupiny.

### 1.2.3 Interní firemní systémy

Pro správu sortimentu, skladových zásob a prodejů lékáren slouží farmaceutický informační systém FaRMIS<sup>2</sup>. Společnost dále používá systém Helios<sup>3</sup> a pro vlastní potřebu byl vyvinut webový informační systém MIS, který slouží pro správu interních požadavků a docházky. Polikliniky a další zdravotnická zařízení ve skupině poté využívají nemocniční informační systém GreyFox (od roku 2008 součást společnosti Stapro s.r.o.<sup>4</sup>). V současné době se připravuje náhrada nemocničního informačního systému, a proto není zpracování dat ze systému GreyFox součástí této práce.

Společnost má aktuálně připravený jednoduchý datový sklad na databázi MS Server 2016 ve verzi Express. Toto řešení je prozatím využito pouze pro e-mailové reportování výsledků za aktuální měsíc vzhledem k datům za minulý měsíc a stejný měsíc minulého roku. Vzhledem k limitům verze Express není možné udržovat více historických dat a také nejsou k dispozici pokročilé analytické nástroje služby MS Analytics Services.

## 1.3 Požadavky managementu

Vzhledem k poměrně intenzivnímu nárůstu konkurenčního prostředí v oblasti farmacie a příchodu nových řetězců lékáren začal být ze strany managementu kladen důraz na optimalizaci nákladů a procesů tak, aby byla zakazníkům

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Windows\\_domain](https://en.wikipedia.org/wiki/Windows_domain)

<sup>2</sup><https://www.farmis.cz>

<sup>3</sup><http://www.helios.eu/>

<sup>4</sup><http://www.stapro.cz/>

dodána plnohodnotná služba při celkovém snížení vstupních zdrojů. Součástí těchto změn je i požadavek na pokročilejší možnosti zpracování prodejních a dalších provozních dat.

V počátcích těchto změn byla jediná možnost analýzy dat postavená na exportech ze systému FaRMIS a následném zpracování těchto výstupů v programu MS Excel. Data byla vyhodnocována pouze v omezeném časovém horizontu a proto bez širších historických souvislostí. Navíc se jednalo o předem nadefinované exporty, bez možnosti uživatelských změn.

Dalším krokem na cestě k data driven rozhodování začalo být tedy logicky zvažováno řešení pomocí datového skladu a následně využití reportovacích nástrojů. Ze strany managementu společnosti Medicon Pharm s.r.o. byly požadavky následující:

- Automatické emailové reporty,
- jednoduchý online nástroj pro analýzu dat,
- maximální využití již dostupných prostředků,
- postupný vývoj dle aktuálních požadavků,
- minimální počáteční náklady.





---

## Datové sklady

Jako datový sklad označujeme zvláštní formu databáze určenou převážně pro analýzu dat v rámci Business Intelligence. Datové sklady shromažďují agregovaná data ze všech částí organizace.[1]

U zrodu datových skladů na počátku devadesátých let stáli William H. Inmon<sup>5</sup> a Ralph Kimball<sup>6</sup>. Už od raných 70. let se Inmon zabýval zpracováním a uložením dat a údajně už z této doby pochází termín datový sklad. Nicméně konkrétní obrysy začal obor dostávat až v roce 1992, kdy vychází publikace Building the Data Warehouse od W. H. Inmona, která poskytla ucelený teoretický základ pro problematiku datových skladů. [2]

V rozvíjejícím se oboru vychází v roce 1996 kniha The Data Warehouse Toolkit od R. Kimbala. Tato kniha je více prakticky orientovaná publikace, která přináší ukázky konkrétních použití a příklady OLAP<sup>7</sup> modelování.

Spolu se dvěma jmény v oblasti datových skladů přichází i dva pohledy na jejich architekturu a na přístup v jejich budování.

V této kapitole budou nejprve porovnány dvě zmíněné architektury návrhu datových skladů. Následně bude popsán dimenzionální model, který je základním stavebním kamenem návrhu, a bude upřesněno, jakým způsobem probíhá dimenzionální modelování.

V návaznosti na toto téma bude přiblížena metoda, jakou jsou data nahrávána do datového skladu s pomocí datových pump, nebo také ETL metod.

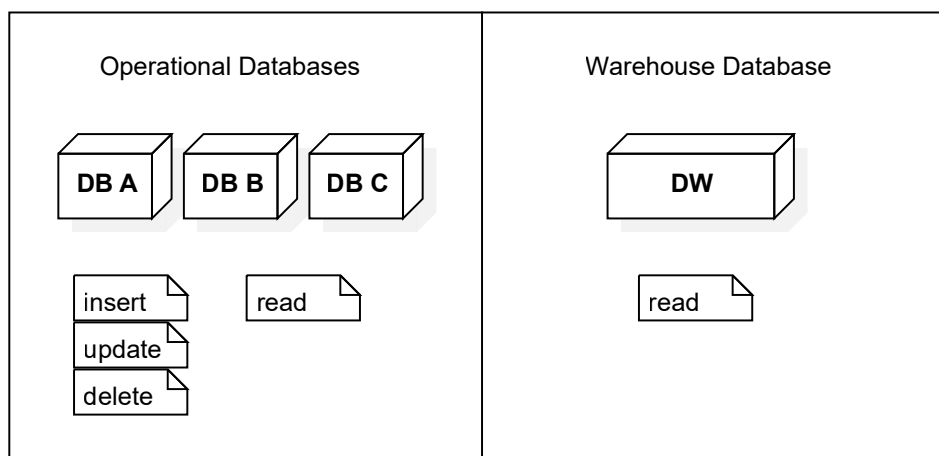
Dále bude vysvětleno co jsou datové kostky, jakým způsobem se s nimi pracuje a jaký je rozdíl mezi OLTP a OLAP přístupem. V závěru kapitoly je krátký úvod do jazyka MDX.

---

<sup>5</sup>[https://en.wikipedia.org/wiki/Bill\\_Inmon](https://en.wikipedia.org/wiki/Bill_Inmon)

<sup>6</sup>[https://en.wikipedia.org/wiki/Ralph\\_Kimball](https://en.wikipedia.org/wiki/Ralph_Kimball)

<sup>7</sup>Online Analytical Processing, podrobněji je popsáno v kapitole 2.4



Obrázek 2.1: Požadavek na stálost datového skladu. V prostředí datového skladu je dostupná pouze operace read.

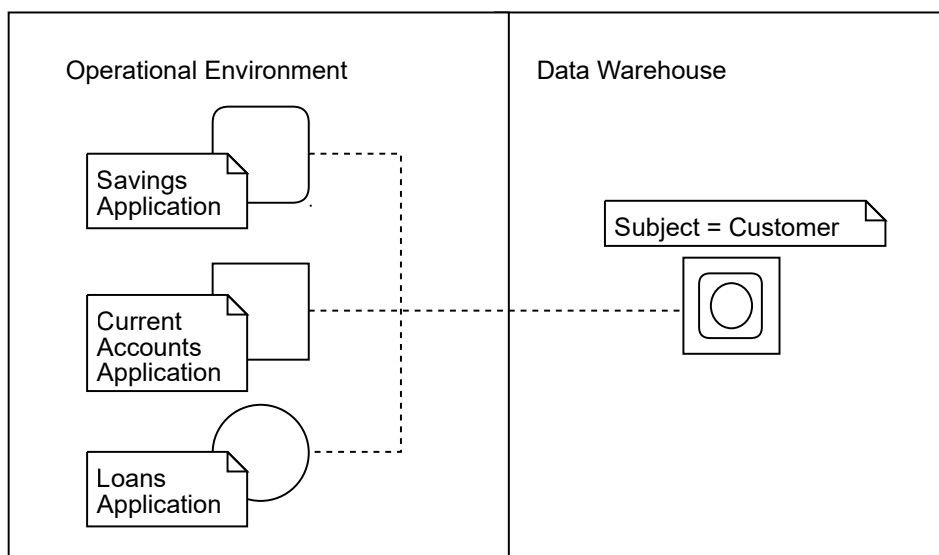
## 2.1 Definice a architektura datových skladů

### 2.1.1 Datové sklady podle W. H. Inmona

W. H. Inmon popisuje datový sklad jako subjektově orientovanou, integrovanou, časově proměnnou a stálou kolekci dat pro podporu rozhodování managementu [3].

Podle Inmona jsou tedy důležité tyto charakteristiky:

- Subjektivní orientaci můžeme vysvětlit jako organizaci kolem hlavních subjektů sledované domény, jako zákazník, produkt, prodej apod. Data jsou organizována dle subjektu spíše, než dle aplikace. Příkladem může být pojišťovna, která by měla v rámci datového skladu organizovaná data podle zákazníků, bonusů a nároků na pojistné spíše, než podle druhu produktů.
- Integrace v této definici odkazuje na sjednocení více různorodých zdrojů dat v rámci organizace – znázorněno na obrázku 2.2. Může se jednat o relační databáze, soubory, transakce apod. Pracuje se s technikami čištění a integrace dat. Příkladem může být sjednocení pojmenování atributů, sjednocení jednotek, nebo formátu zápisu.
- Časový horizont, se kterým pracuje datový sklad je výrazně delší, než u operačních databází. V běžných operačních databázích pracujeme s aktuálními daty, naproti tomu datové sklady se zaměřují na informace



Obrázek 2.2: Sjednocení dat z více zdrojů a seskupení kolem důležitých subjektů.

z historické perspektivy (např. posledních 5 až 10 let). Každá informace v datovém skladu by měla být pojena s časovým elementem.

- Poslední požadavek na stálost odkazuje na fyzické oddělení dat od operačního prostředí. Z principu datových skladů a z požadavků na poskytování historických souvislostí vychází požadavek na stálost dat, jakmile jsou jednou nahrána. Datový sklad vyžaduje pouze dvě operace v přístupu k datům, a to prvotní nahrání dat a následný přístup k datům, viz obrázek 2.1. Díky fyzickému oddělení dat se není třeba starat o systémy zpracování transakcí, obnovy, kontroly konkurenčního přístupu atd.

[1, 4]

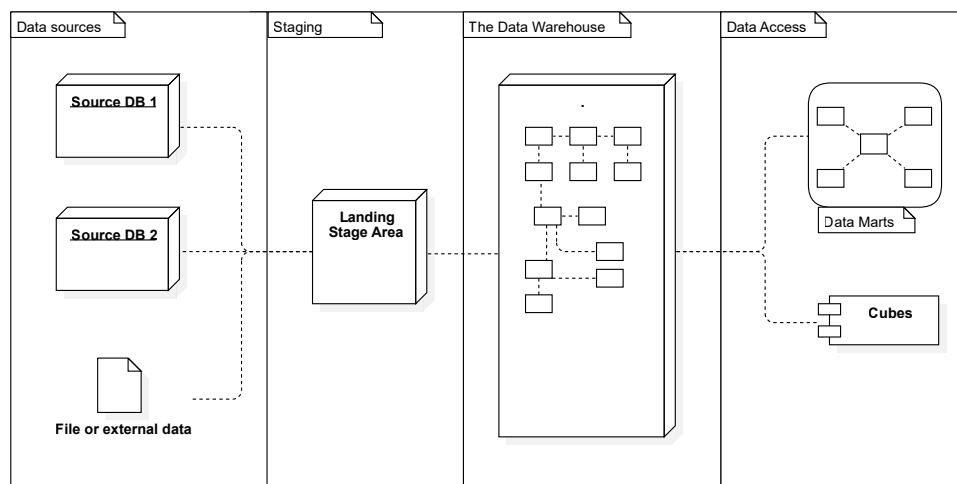
Architektura datového skladu dle W. H. Inmona staví na celopodnikovém uložení dat, kde je možné sledovat veškeré souvislosti bez omezení na konkrétní vnitropodniková oddělení. Nejprve je tedy třeba analyzovat všechny datové zdroje a všechna data dané organizace a vytvořit komplexní model, ze kterého vychází datový sklad – top down přístup.

Takto vytvořený datový sklad agreguje v jedné databázi veškerá podniková data, která by měla být atomická (nízka granularita) a udržována v 3NF.

Architektura datového skladu je v tomto případě čtyřvrstvá. Rozvržení je vidět na obrázku 2.3.

První vrstva se skládá ze všech datových zdrojů. Jedná se převážně o provozní databáze, externí soubory, exporty atd.

## 2. DATOVÉ SKLADY



Obrázek 2.3: Návrh architektury datového skladu podle Inmona. V části The Data Warehouse je možné vidět jednu komplexní skupinu tabulek, která reprezentuje jednotný velký datový sklad.

Druhá vrstva je dočasné uložení dat, neboli stage area, spolu s fyzickým datovým skladem. Nejprve přichází na řadu extrakce dat, která je součástí ETL procesů<sup>8</sup>. V rámci procesu extrakce jsou data nahrána ze zdrojových systémů právě do stage area, kde je možné data dále čistit a transformovat – například sjednocením formátu atributů z různých systémů. Následně jsou očištěná a transformovaná data pomocí ETL metody load nahrána ze stage area do datového skladu.

Jakmile je centrální datový sklad připravený, je možné vytvářet datová tržiště, neboli tzv. „data marts“. Datová tržiště jsou menší databáze, které obsahují podmnožinu dat určenou pro konkrétní oddělení, nebo business proces. Jedná se o třetí vrstvu modelu.

Poslední vrstva už jsou konkrétní aplikace pro koncové uživatele. Může se jednat o reportovací nástroje, například doplněk Power Pivot pro MS Excell, nebo automatické emailové reporty apod.

Mezi hlavní výhody tohoto konceptu patří:[5, 6]

- design je komplexní, a proto jednodušší na údržbu,
- všechna data jsou pohromadě, a je tedy možné hledat souvislosti na úrovni celého podniku,
- jakmile je datový sklad připravený, je relativně jednoduché vytvářet libovolná datová tržiště pro konkrétní potřeby podniku

<sup>8</sup>Problematika ETL je podrobněji popsána v kapitole 2.3

### 2.1.2 Datové sklady podle R. Kimballa

R. Kimball definoval datový sklad jako kopii dat z transakčních systémů, jejíž struktura je uzpůsobena pro dotazování a analýzu. Zároveň prohlásil, že datový sklad není nic jiného, než sjednocení všech datových tržišť.

Architektura datového skladu podle R. Kimballa staví na tzv. bottom-up principu, tedy že nejprve jsou vybudovány datová tržiště pro jednotlivá oddělení, nebo části podniku a následně později je možné tato tržiště sjednotit do jednoho velkého datového skladu.

Je důležité zmínit, že velký datový sklad – dle definice sjednocení datových tržišť – nemusí být další fyzická databáze. Naopak se jedná o virtuální entitu. Při návrhu jednotlivých datových tržišť je tedy doporučeno udržovat jednotný formát dat a následně i integritu dat napříč záznamy. Toho je docíleno převážně vhodným návrhem tzv. conformed dimensions, které jsou sdílené pro jednotlivá datová tržiště a slouží tak jako propojení mezi jednotlivými datovými modely.

Výhodou tohoto procesu je znovupoužitelnost ETL uzlů, nebo jejich částí. Nevýhodou mohou být poměrně velké nároky na údržbu jednotlivých datových tržišť, jelikož z důvodu výkonu může docházet napříč tržišti k duplicitě dat.

Pokud je architektura správně navržena a je dbáno o zmíněné parametry, je možné jednotlivá datová tržiště propojit s pomocí toho, co Kimball označil jako „data warehouse data bus“ a přistupovat k nim jako k jednotnému komplexnímu datovému skladu. [5, 6]

Architekturu dle Kimballa znázorňuje obrázek 2.4. Oproti pohledu W. Inmona je tento model prezentovaný jako třívrstvá architektura. První vrstva je v tomto případě totožná a jedná se o zdrojové systémy.

Druhá vrstva se skládá jako v prvním případě ze stage area, ale jako druhý modul zde již není komplexní datový sklad. V tomto případě jsou součástí druhé vrstvy již přímo datová tržiště, která jako celek reprezentují virtuální datový sklad.

Třetí vrstva je totožná se čtvrtou předchozího modelu – jedná se o uživatelské aplikace, analytické nástroje a další.

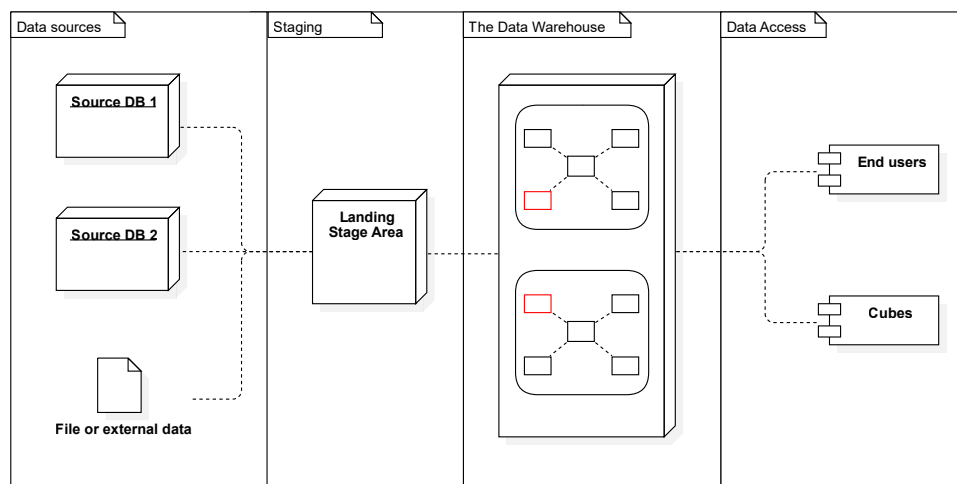
Výhody konceptu podle R. Kimballa jsou:[5, 6]

- poskytuje první přidanou hodnotu relativně brzy a s malým rozpočtem,
- není tolik rigidní a úpravy a změny jsou jednodušší,
- je uživatelský přívětivější, jelikož vychází z požadavků menší skupiny lidí konkrétního oddělení podniku

## 2.2 Struktura a návrh datových skladů

Kolem datových skladů se objevuje několik nových pojmů, kterým je třeba porozumět. V následujících odstavcích budou vysvětleny ty nejzásadnější.

## 2. DATOVÉ SKLADY



Obrázek 2.4: Návrh architektury datového skladu podle Kimbala. V části The Data Warehouse je možné vidět více tzv. datových tržišť, které při správném návrhu dohromady tvoří velký datový sklad. Pro propojení datových tržišť je využito tzv. conformed dimensions (znázorněno červeně), které dohromady tvoří data warehouse bus architecture.

Nejprve bude vysvětlen nový přístup k datovému modelování pomocí dimenzí a faktů. V další části bude přiblíženo jakým způsobem probíhá návrh datového skladu a na závěr této části bude vysvětleno něco málo ke změnám modelu v průběhu času – konkrétně ke změně atributů jednotlivých datových dimenzí.

### 2.2.1 Multidimenzionální datový model

Datové sklady jsou založeny na multidimenzionálním datovém modelu, který je vhodný pro následné složité analytické dotazy nad velkým množstvím dat. Kromě toho je více srozumitelný uživatelům provádějícím reportování nad datovým skladem.

Objekty jsou při dimenzionálním modelování rozděleny na fakta a dimenze. Dimenze obecně popisují objekty a metriky, jejichž konkrétní hodnoty jsou uloženy v tabulkách faktů. Dimenze mohou být například čas, provozovna, značka, produkt apod. Metriky v tabulkách faktů potom bývají často prodeje v nějaké měně, nebo počty prodaných kusů zboží. Porovnání dimensionální tabulky s faktovou tabulkou je na obrázku 2.5.

Na obrázku 2.6 je možné vidět jednoduché hvězdicové schéma a jeho promítnutí do základního reportu. V reportu je využito produktové dimenze Product Dimension pro pohled na metriky dle jednotlivých značek (Brand

| Product Dimension Table |                          |
|-------------------------|--------------------------|
| PK                      | Product Key              |
|                         | Product Description      |
|                         | SKU Number NK            |
|                         | Brand Description        |
|                         | Category Description     |
|                         | Department Description   |
|                         | Package Type Description |
|                         | Fat Content Description  |
|                         | Weight                   |
|                         | Weight Units of Measure  |
|                         | Storage Type             |
|                         | Shelf Life Type          |
|                         | Shelf Width              |
|                         | Shelf Height             |
|                         | Shelf Depth              |
|                         | ...                      |

| Daily Sales Fact Table |               |      |
|------------------------|---------------|------|
| FK                     | Date Key      |      |
| FK                     | Product Key   |      |
| FK                     | Store Key     |      |
|                        | Quantity Sold | Fact |
|                        | Sales Amount  | Fact |

Obrázek 2.5: Porovnání dimenzionální tabulky (nalevo) a tabulky faktů (na-pravo).

Description) a dimenze lokality *Store Dimension* pro zobrazení metrik rozdělených dle lokalit, ve kterých se nachází prodejní místa.

V reportu byly následně zvoleny metriky z tabulky faktů *Daily Sales Facts*. Konkrétně jsou zobrazeny součty prodejů v dolarech (*Dollar Sales Amount*) a počty prodaných položek (*Quantity Sold*).

### Tabulka faktů [7]

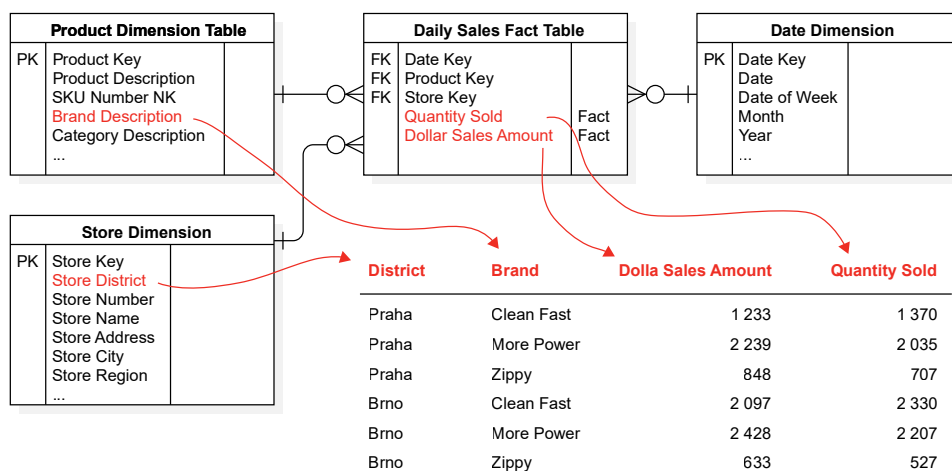
Faktová tabulka je hlavní tabulka dimenzionálního modelu. Každý řádek v této tabulce představuje numerickou metriku, která reprezentuje specifický aspekt daného podnikání, procesů a aktivit. Při vytváření jednotlivých metrik je vhodné udržovat co nejnižší úroveň detailu – granularitu, což umožní sestavit libovolné analytické dotazy při následném vyhodnocování a procházení dat.

Tabulky mohou obsahovat řádově i desítky cizích klíčů, odkazujících na tabulky dimenzí. Počet samotných metrik nemusí být velký, obecně ale obsahují faktové tabulky velké množství záznamů, a proto mají majoritní podíl na potřebném paměťovém prostoru datového skladu.

### Tabulka dimenze [7, 8]

Tabulky dimenzí jsou tabulky, které znázorňují určitý pohled na metriky uložené v tabulce faktů a doplňují další informace o těchto metrikách. Tabulky mívají hodně sloupců, nebo atributů a většina je převážně textová. Mají definovaný jeden primární klíč. Porovnání tabulky dimenze s tabulkou faktů je na příkladu znázorněno na obrázku 2.5.

## 2. DATOVÉ SKLADY



Obrázek 2.6: Ukázka generování reportu z hvězdicového schématu. Přepřacováno, zdroj:[7].

Jsou to právě tabulky dimenzí, s jejichž pomocí jsou vytvářeny analytické dotazy.

Dimenzionální tabulky jsou obecně velmi denormalizované, což vede k větší paměťové náročnosti takového modelu. Tento problém se dá řešit s využitím schématu sněžové vločky 2.2.1

V rámci dimenzionálních tabulek je možné atributy rozdělit do dvou skupin:

- dimenzionální atributy – jedná se o atributy agregační úrovně. Úroveň je dána množinou instancí. Mezi atributy agregační úrovně existuje vztah 1:N, nebo M:N. Na základě těchto atributů jsou dále budovány dimenzionální hierarchie.
- Nedimenzionální atributy – slouží pro popis vlastností prvků. Blíže charakterizují jednotlivé úrovně.

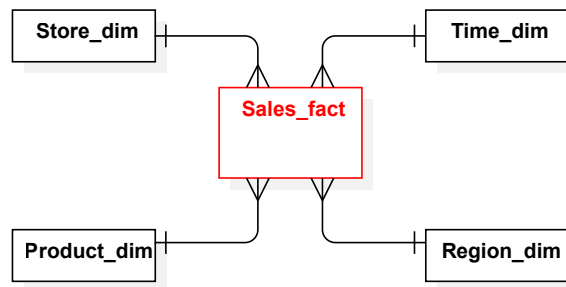
### Náhradní klíč (Surrogate Key) [9, 7, 8]

Náhradní klíč je při modelování datového skladu velmi důležitý, jelikož pomáhá udržovat integritu dat a zároveň podporuje zachování kompletní historie změn.

Náhradní klíč je číselná hodnota, která nemá žádný obchodní význam a je sekvenčně přiřazena při nahrávání dat do tabulky dimenzí. Slouží ke spojení tabulky dimenzí a tabulky faktů. Dotazy a externí aplikace by neměly obsahovat žádnou logiku založenou na úvodních primárních klíčích.

Mezi výhody používání náhradních klíčů patří: [8]





Obrázek 2.7: Schéma datového skladu - star. Jedna tabulka faktů a jednoduché členění dimenzí. Přepracováno, zdroj:[10].

- menší nároky na paměť,
- při integraci z více zdrojů pomáhá omezit nekonzistenci,
- chrání před změnami ve zdrojových transakčních systémech
- u dimenze data umožňuje přímočaré rozdělení tabulek na více partitions.

### Základní schémata propojení tabulek [10]

V datovém modelu jsou přítomny jedna nebo více tabulek faktů, které je třeba propojit s tabulkami dimenzí. Jakým způsobem tabulky propojíme a do jaké míry rozpadneme informace o konkrétních dimenzích, to záleží na zvoleném schématu.

- Star schéma je nejjednodušší schéma pro datové sklady. Schéma je znázorněno na obrázku 2.7. V centru schématu se nachází tabulka faktů obsahující primární informace. Tabulka faktů je obklopena menšími dimenzionálními tabulkami, které definují a upřesňují konkrétní záznamy. Primární klíč každé dimenzionální tabulky odkazuje na cizí klíč v tabulce faktů.

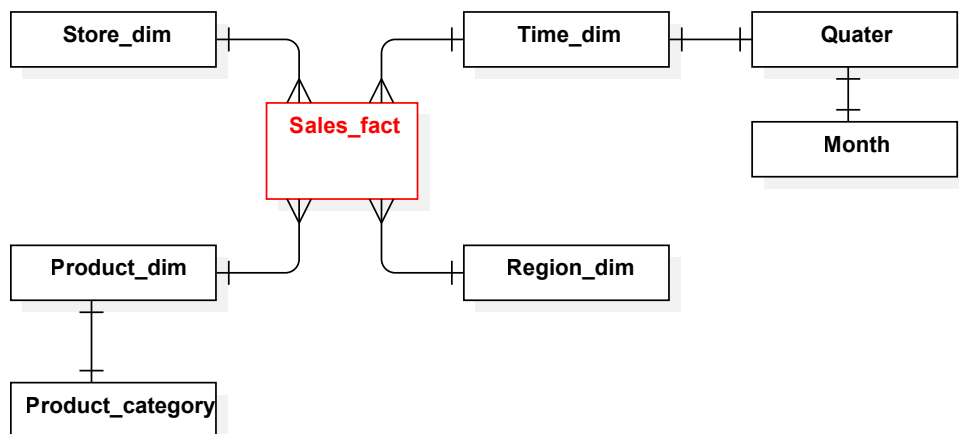
Na obrázku 2.8 je vidět například rozpad časové dimenze na **Quarter** pro čtvrtletí a **Month** pro měsíce.

Výhodou tohoto modelu jsou nižší nároky na diskovou kapacitu a spojování menších tabulek při dotazování. Nevýhodou může být údržba většího množství tabulek a vazeb a v neposlední řadě brání účinnému využití bitmapového indexování. [7]

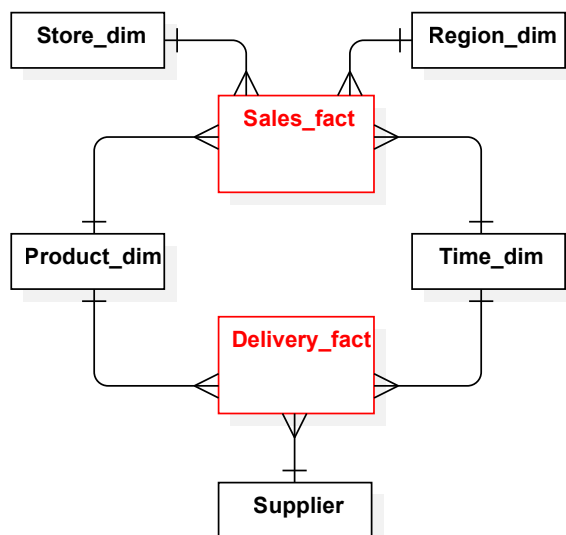
- Konstalace faktů je schéma, které obsahuje více tabulek faktů, které mohou sdílet stejné tabulky dimenzí. Schéma je znázorněno na obrázku 2.9.

## 2. DATOVÉ SKLADY

---



Obrázek 2.8: Schéma datového skladu - snowflake. Jedna tabulka faktů. Tabulky dimenzí jsou rozděleny na další tabulky, což snižuje redundanci. Přepřacováno, zdroj:[10].



Obrázek 2.9: Schéma datového skladu - konstalace faktů. Může obsahovat více tabulek faktů, které mohou sdílet některé tabulky dimenzí. Přepřacováno, zdroj:[10].

### 2.2.2 Multidimenzionální modelování

V publikaci [7] je doporučeno postupovat při dimenzionálním modelování podle následujících čtyř bodů:

1. Volba business procesu – je třeba identifikovat proces, který bude nosný prvek datového modelu. Příkladem může být objednání zboží, inventurizace, fakturace apod.
2. Definovat zrnitost – zvolit granularitu v popisování vybraného procesu. V případě evidence letů leteckou společností, může být základní položka jedna vystavená letenka.
3. Volba dimenzí – dalším krokem je volba dimenzí, které popisují zvolený proces – pomocí kterých je možné vytvářet otázky k danému procesu. Například jaká letenka byla zakoupena? (produkt), Kdo si letenku koupil? (zákazník), Jak za letenku bylo zapláceno? (způsob platby) apod.
4. Identifikování faktů – co se bude u daného procesu měřit? Odpověď na tuto otázku definuje jednotlivá fakta.

Při modelování se můžeme setkat s některými speciálními případy, kdy je potřeba přistoupit k danému problému s kompromisním řešením, které vyváží požadavky mezi čistotou návrhu, potřebným výkonem pro zpracování dotazů a využitým diskovým prostorem.

#### Degenerované dimenze [8]

Jedná se o dimenzi, která nemá vlastní tabulku a je součástí tabulky faktů. Důvodem je, že takový dimenze je charakterizovaná jednou úrovní podrobnosti a neobsahuje vlastnosti atributů. Většinou se jedná o primární a alternativní klíče z transakčních systémů – například číslo faktury apod.

Degenerované dimenze se většinou podílí na tvorbě primárního klíče faktové tabulky.

#### Denormalizace tabulky faktů [7]

Po dokončení prvního návrhu dimenzionálního modelu se může stát, že model obsahuje příliš mnoho dimenzí. Dle [7] by měla jít většina business procesů popsat pomocí méně než 15 dimenzí. Pokud má model 25, nebo i více dimenzí, měl by se designer zamyslet nad možností spojení některých korelovaných dimenzí do jedné kombinované dimenze.

Pro velmi dobře korelované atributy – například ty, které tvoří jednu hierarchii – platí, že by měly tvořit jednu dimenzi.

### 2.2.3 Dimenze a změny

[8, 11]

Při návrhu datového skladu jsou definovány tabulky dimenzí spolu s konkrétními atributy. V průběhu času se ale tyto atributy mohou měnit. Například zákazníci mění svá jména, mají děti, stěhují se apod. Podle toho, jestli změny probíhají rychle, nebo pomalu se rozlišují:

- pomalu se měnící dimenze (většina),
- rychle se měnící dimenze.

Z důvodu možných změn je třeba již při návrhu stanovit strategii pro vyjádření změn. Součástí toho je analýza, do které je zapojený management společnosti a jejíž výsledek by měl stanovit:

- jaké změny hodnot atributů jsou možné,
- jaký výstup bude management s ohledem na tyto změny požadovat.

**Pomalu se měnící dimenze** Takové dimenze se blíží konstantním dimenzím – jejich atributy se mění v čase pomalu. Možnosti pro vyjádření změn jsou následující:

1. přepsat hodnotu – stará hodnota atributu je nahrazena novou. Toto řešení je rychlé a jednoduché, ale vede ke ztrátě historie a může působit neočekávané změny v agregovaných datech. Například při změně bydliště zákazníka se všechny jeho historické nákupy budou vázat k novému místu bydliště.
2. přidat dimenzi řádek – původní řádek zůstává v tabulce, ale je přidán nový řádek, který vyjadřuje změnu. Toto řešení ovšem vyžaduje nový primární klíč pro vytvořený záznam, případně další sloupce pro identifikování aktuálního záznamu apod. Výhodou je zachování historických souvislostí.
3. přidat dimenzi sloupec – zachování historických souvislostí je docíleno přidáním nového sloupce, který definuje starou hodnotu. Toto řešení se už na první pohled nehodí na častější změny a tím pádem ztrácíme možnost sledovat vývoj změn.

V praxi se nejčastěji užívá druhá varianta. Kromě těchto tří možností existují dále hybridní techniky, které je kombinují.

**Rychle se měnící dimenze** U rychle se měnících dimenzí probíhají změny atributů s denní i častější periodicitou. Řešení potom závisí na velikosti dimenzi a povaze rychle se měnících atributů.

Pro malé dimenze se dle [8] nejčastěji používá druhý typ řešení pro pomalu se měnící dimenze. Pro rozsáhlé dimenze je vhodné oddělit rychle se měnící atributy do jiné tabulky dimenzí. Výhodou takového řešení je možnost sledování změn.

## 2.3 Datové pumpy ETL

[12, 13]

Datové pumpy slouží v kontextu datových skladů k přenosu dat mezi zdrojovými transakčními systémy a samotným datovým skladem. ETL nástroje je možné rozdělit do tří kategorií:

1. Extrakce – ze zdrojových transakčních systémů jsou extrahována relevantní data,
2. Transformace – získaná data se transformují do požadované podoby (očistění dat, sjednocení formátu z více systémů apod.),
3. Loading – finalní očistěná data se načtou do datového skladu.

### 2.3.1 Extrakce

První fáze ETL je poměrně přímočará, nicméně může být problematická na straně zdrojových systémů. Zdrojové systémy podniku mohou využívat různé technologie i platformy. Navíc, ne všechny systémy jsou datově otevřené.

Zdrojová data mohou být v různých formátech. Je možné se setkat se zdroji dat v nějakém databázovém systému, souborech různých formátů (\*.csv, \*.xml apod.). V dnešní době nejsou vyloučeny ani zdroje v podobě naskenovaných dokumentů, nebo zvukových záznamů, pro jejichž použití je třeba využít dalších nástrojů.

### 2.3.2 Transformace

Součástí transformační části je očistění dat, sjednocení datových formátů a další akce zajišťující kvalitu a konzistenci dat. Cílem transformačních aktivit je zajistit:

- úplnost – identifikace chybějících dat,
- soulad – ošetření dat v nestandardních formátech,
- konzistenci – náprava záznamů, které reprezentují konfliktní informace,
- přesnost – ošetření nepřesných, nebo zastaralých dat,

## 2. DATOVÉ SKLADY

---

- unikátnost – odstranění duplicitních dat,
- integritu – zajištění vazeb mezi důležitými datovými zdroji.

V rámci procesu transformace dat jsou používány následující typy změn.

**Změna formátu** V rámci datového skladu budou data spojena do jednotlivých tabulek faktů a dimenzí. Z tohoto důvodu je potřeba sjednotit formáty jednotlivých atributů zdrojových systémů. Příkladem může být datum a čas – znázorněno v tabulce 2.1.

Tabulka 2.1: ETL transformace formátu data.

| Formát zdroje A | Formát zdroje B | Sjednocený formát |
|-----------------|-----------------|-------------------|
| 10.3.2015       | 03-10-2015      | 10.3.2015         |

**Odstranění duplicitních záznamů** Při zpracování dat z různých systémů dochází často k duplicitám. Příkladem můžou být dvě tabulky obsahující data o zákaznících. Jedna pochází ze systému využívaného marketingovým oddělením a druhá ze systému oddělení obchodního.

V takovém případě je třeba taková data sjednotit a při tom rozhodnout, která informace je pravdivá v případě, že se neshodují.

Rozdílné hodnoty daného atributu pro jednu entitu (například zákazníka) mohou být způsobeny i lidskou chybou. Například v případě překlepu může být mnohdy nemožné strojově rozhodnout, která informace je pravdivá.

Řešením může být stanovení jednoho ze zdrojů jako referenčního a v případě nejasností vybrat hodnotu obsaženou v referenčním zdroji dat. Druhá možnost je problémová místa označit a následně vyhodnotit ručně.

V příkladu uvedeném v tabulce 2.2 by bylo možné využít databázi měst a obcí jako referenční množinu a na základě toho vybrat správný, jednotný název.

Tabulka 2.2: ETL odstranění duplicitních záznamů

| Různé zápisy názvu města | Sjednoceno       |
|--------------------------|------------------|
| Moravská Třebová         | Moravská Třebová |
| M. Třebová               | Moravská Třebová |
| MT                       | Moravská Třebová |
| Mor. Třebová             | Moravská Třebová |
| Třebová, Moravská        | Moravská Třebová |

**Nahrazení chybějících hodnot** V případě chybějících hodnot je nejprve vhodné pokusit se data doplnit z jiných zdrojů. Pokud to není možné, nezbyvá,

než danou hodnotu nahradit vhodným příznakem značícím, že hodnota nebyla uvedena.

**Rozdělení položek** Občas se může stát že jsou některé atributy ve zdrojových systémech uvedeny jako jeden textový řetězec. Typickým příkladem může být adresa, viz ukázka v tabulce 2.3. Občas je naopak vhodné spojit více záznamů ze zdrojových systémů do jednoho.

Tabulka 2.3: ETL rozdělení položek

| Zdroj: adresa             | DW: Město  | DW: Ulice  | DW: číslo p. |
|---------------------------|------------|------------|--------------|
| Linhartice, neznámkova 13 | Linhartice | Neznámkova | 13           |

**Odstranění nejednoznačnosti a nesrozumitelnosti údajů** Součástí zdrojových dat mohou být nejrůznější typy příznaků, zkratek a generovaných hodnot. Pokud jsou problémové atributy zásadní a důležité pro problémovou doménu, je třeba je nahradit takovým způsobem, aby byly konečnému uživateli srozumitelné.

**Zajištění referenční integrity** V rámci části věnované dimenzionálnímu modelování bylo popsáno jakým způsobem se vypořádat se změnou dimenzí 2.2.3. Tyto změny je ale třeba řešit už v nástrojích ETL.

**Doplnění chybějícího časového údaje** V úvodní části této kapitoly 2.1 bylo zdůrazněno, že architektura datového skladu vyžaduje, aby každý jeho záznam byl spojen s časovým údajem. Některé zdrojové systémy ale nemusí časový údaj obsahovat. V takovém případě je potřeba časové údaje doplnit v rámci transformační fáze ETL procesů.

**Doplnění agregovaných hodnot** Existují dva přístupy k vytváření agregovaných dat. Buď jsou do datového skladu nahrána atomická data a agregace jsou následně vypočítány přímo v datovém skladu, nebo je možné některé agregace předpočítat a nahrát je do datového skladu přímo.

**Detekce a oprava nejasných číselníků** Při zavádění podnikových číselníků do datového skladu se mohou objevit nejasnosti ve vazbách některých entit na příslušné číselníky. Například entita zboží může být spojena s větším množstvím kódů z nahraných dat, což může vést k nepřesnostem při reportování nad datovým skladem. Součástí transformační části by mělo být odstranění těchto nepřesností a sjednocení nahrávaných dat.

### 2.3.3 Loading

Na závěr ETL procesu je třeba očištěná a transformovaná data nahrát do datového skladu. Některá řešení využívají ještě mezivrstvu dočasného uložení. Celý loading proces, stejně jako předchozí dvě fáze ETL, by měly být maximálně automatizovány a většinou probíhají v časech mimo nejnávětší části dne, kdy na datový sklad nejsou kladeny požadavky ze strany uživatelů.

V rámci nahrání dat do datového skladu je doporučeno vypnout okamžitou indexaci. Následně probíhá generování náhradních klíčů pro nové záznamy, případné výpočty agregací a na závěr reindexace celého datového skladu pro zlepšení dotazovacích výkonů.

Speciální případ je prvotní nahrání historických dat, které je časově velmi náročné. Další aktualizace již probíhají většinou na denní bázi a jsou doplněny pouze nejaktuálnější záznamy.

## 2.4 OLTP a OLAP

[14]

V přístupu k datům se objevují dvě základní technologie uložení a zpracování dat. První z nich je OLTP<sup>9</sup>, který je charakteristický velkým množstvím poměrně malých transakcí (insert, update, delete). OLTP systém je přizpůsoben pro rychlé vyřízení transakcí a údržbu integrity dat. Zároveň je kladen důraz na snížení redundance dat. Model databáze je většinou udržován v 3NF<sup>10</sup>.

Měřítkem je v tomto případě počet transakcí za jednotku času. Typickým zástupcem jsou provozní databáze, které se starají o day-to-day operace jako nákupy, zásoby, bankovníctví apod.

Datové sklady a další součásti Business Intelligence na druhé straně využívají přístup OLAP<sup>11</sup>, který je typický pro relativně malé množství transakcí. Dotazy jsou ale mnohem komplexnější.

Měřítkem pro tento model je čas odezvy na dané komplexní dotazy. Data typicky nemusí splňovat podmínky 3NF, naopak je běžná redundance. Databáze jsou obvykle založeny na multidimenzionálních schématech a obsahují velké množství agregovaných historických dat.

### 2.4.1 Datové kostky

[14]

Na data uložená v databázi založené na multidimenzionálním modelu bývá nahlíženo s pomocí jejich zobrazení do datové kostky. Základní datová kostka

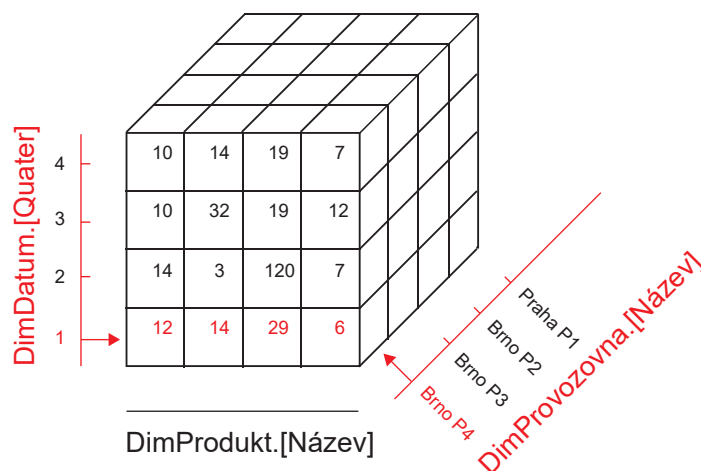
---

<sup>9</sup>On-line Transaction Processing

<sup>10</sup>Třetí normální forma: [https://en.wikipedia.org/wiki/Third\\_normal\\_form](https://en.wikipedia.org/wiki/Third_normal_form)

<sup>11</sup>On-line Analytical Processing





Obrázek 2.10: Ukázka datové kostky pro 3 dimenze. Červeně je naznačen výsledek dotazu, který zmenší datový výběr upřesněním dvou dimenzí.

náležící k databázi má tolik rozměrů (dimenzí), kolik má samotná databáze dimenzionálních tabulek<sup>12</sup>.

Ukázku datové kostky je možné vidět na obrázku 2.10. Dimenzionální datový model obsahuje nejméně tři tabulky dimenzí: DimDatum, DimProdukt a DimProvozovna. Hodnoty v jednotlivých buňkách jsou metriky – v tomto případě se může jednat o počty prodaných kusů daných produktů.

V oblasti datových kostek se můžeme dále setkat s těmito vybranými pojmy:

- **Base cuboid** je maximální datová kostka pro daný model. Obsahuje nejdetailnější informace v rámci všech dostupných dimenzí.
- **Apex cuboid** je nejmenší datová kostka, která obsahuje maximálně sumarizovanou hodnotu. Jedná se vlastně o součet všech hodnot dané metriky napříč všemi dimenzemi.
- **Agregace kostky** jsou sumarizované metriky napříč vybranými dimenzemi.

Volbou dimenzí je možné procházet data a prohlížet si jejich vybrané podmnožiny. Zároveň vidíme, že pomocí tohoto zobrazení je možné poměrně rychle a efektivně hledat souvislosti.

<sup>12</sup>Podrobněji popsáno v kapitole 2.2.1

### 2.4.2 OLAP operace

Multidimenzionální uložení dat 2.2.1 a jejich reprezentace pomocí datové kostky vede k definici nového typu operací nad těmito daty. Operace nad datovou kostkou nám umožňují vybírat její podmnožiny napříč dostupnými dimenzemi.

Mezi základní operace patří:[15][16][14]

- drill-down, umožňující navigaci v rámci hierarchie vybrané dimenze směrem k větší granularitě (například pro přesun od kvartálů k měsícům v dimezi času), nebo také přidání dimenze,
- roll-up, která je opačnou operací k předchozí. Umožňuje snížení granularity (například od měst k okresům), a to pomocí redukce počtu dimenzí,
- slice, výběrem konkrétní hodnoty z jedné dimenze poskytne novou podkostku, která obsahuje podmnožinu dat pouze pro vybranou hodnotu ze zvolené dimenze,
- dice, zvolením dvou, nebo více dimenzí je vytvořena nová datová kostka (opět podmnožina původní) – příklad je na obrázku 2.10,
- pivot (rotate), slouží pro rotaci datové kostky, jedná se o prohození jednotlivých datových os.

## 2.5 Dotazovací jazyk MDX

Operace zmíněné v 2.4.2 jsou součástí většího celku operací, který zapouzdřuje dotazovací jazyk MDX<sup>13</sup>. [16]

MDX dotazovací jazyk je syntakticky podobný SQL. Základní kostra MDX dotazu vypadá takto:

**Select**

```
{[Dimenze1], [Dimenze2], ... } ON COLUMNS,  
{[Metrika1], [Metrika2], [Metrika3], ... } ON ROWS
```

**From** [DatovaKostka]

**Where** /\* slicer \*/

---

<sup>13</sup>MultiDimensional Expressions je jazyk originálně vydaný společností Microsoft. Dnes se jedná v kombinaci se standardem XMLA (XML for Analytics) o nejpoužívanější technologii pro OLAP operace nad multidimenzionálními daty [17].

**Select** specifikuje dimenze a jejich rozdělení na řádky a sloupce pro výsledný report.

Řádky a sloupce jsou osy v rámci kterých se zobrazují dotazovaná data. Sloupce jsou také označovány jako osa 0 (nebo klíčovým slovem `COLUMNS`), řádky potom jako osa 1 (nebo `ROWS`). Jazyk MDX podporuje až 127 různých os.

**From** specifikuje datové kostky. Může jich být i víc, ale v takovém případě musí jednotlivé kostky sdílet dimenze.

**Where** slouží pro nastavení tzv. sliceru, který upřesňuje data, která požadujeme. Specifikují se zde metriky.[16]

Syntakticky se můžeme v dotazech setkat s třemi druhy závorek. Hranaté závorky `[]` slouží pro zapisování speciálních znaků, mezer apod. v názvech všech parametrů.

Složené závorky `{ }` umožňují zapisovat množiny. Množina je skupina buněk se stejnou dimenzionalitou.[18] Na závěr je možné používat kulaté závorky `()`, které slouží pro všechny ostatní případy (zápis souřadnic v tuplu<sup>14</sup>, funkce, matematické operace apod.).

Při výběru dat v části **Select** existují dvě možnosti reference členů. Je možné volit přímo jméno a nebo klíč. [18] doporučuje volat s pomocí klíče a pro složitější dotazy využívat tzv. defaultních memberů.

Příklad volání přes klíč: `[dimproduct].[Category].&[1]`

Příklad volání přes jméno: `[dimproduct].[Category].[Bikes]`

### 2.5.1 Práce s množinou

Množinou v sémantice jazyka MDX rozumíme skupinu tuplů stejné dimenzionality. Při výběru množin v rámci statementu `select` daného dotazu je možné použít volání přes klíč. To ale v případě rozsáhlých dimenzí není vhodné řešení. Lepším řešením je využití hierarchií. [19]

V rámci jazyka MDX a dotazování nad datovou kostkou rozlišujeme dva typy hierarchií.

#### Atributová hierarchie

Je hierarchie tvořená pouze atributem. Má pouze dvě úrovně (`all`, také `All-member` a jednotlivé položky). Příkladem může být časová dimenze, ze které je vybraná množina Rok a položky 2012, 2013, 2014 a 2015.

<sup>14</sup>Tuple popisuje buňku v prostoru.

### Vlastní hierarchie

Je víceúrovňová hierarchie, definovaná uživatelem. Například u dimenze čas to může být hierarchie *Rok* → *Kvartál* → *Měsíc* → *Den*.

Pro výběr všech prvků v rámci vybrané dimenze tedy není třeba vypisovat pomocí klíčů všechny jednotlivé prvky, ale je možné využít hierarchie a funkcí **Members** a **Allmembers**.

Příklad použití na ukázkové OLAP kostce 2.10. Výsledek následujícího dotazu bude počet prodaných produktů rozdělených po řádcích dle jednotlivých provozoven.

#### Select

```
{ [Measures] . [Mnozstvi] } ON COLUMNS,  
{ [DimProvozovna] . [Nazev] . members } ON ROWS
```

**From** SalesCube

Pokud navíc dává smysl zobrazit sumární hodnotu pro danou dimenzi, je možné použít právě funkci **Allmembers**. Výsledek v ukázkovém příkladu bude součet prodaných kusů zboží ve všech provozovnách dohromady – v tomto případě dává smysl tuto hodnotu uživateli zobrazit.

### 2.5.2 Počítané členy a funkce **CurrentMember**

#### Počítané členy

V rámci dotazu je mnohdy vhodné doplnit nějakou hodnotu, která přímo není součástí datové kostky, ale jsou k dispozici všechny údaje pro její výpočet. V takovém případě je možné použití počítaného členu.

Počítané členy je možné používat přímo při MDX dotazování, ale i je předpřipravit v datové kostce a zpřístupnit je tak jejím uživatelům.

Následující ukáзка použití počítaného členu znázorňuje vytvoření nové metriky **Mnozstvi**, která reprezentuje součet dvou zdrojových metrik **Mnozstvi Online** a **Mnozstvi Retail**.

Dotaz je postavený na konstruktu **WITH MEMBER ... AS**. V části **select** je poté nově vytvořená metrika zobrazena na ose **COLUMNS** spolu s těmi zdrojovými.

Navíc je zde použita funkce **FORMAT\_STRING** umožňující naformátovat výstup počítaného členu dle běžných možností.

Před definicí obsahu, který bude zobrazen na ose **ROWS** je možné si všimnout klíčového slova **NON EMPTY**. Tato vlastnost slouží pro filtraci prázdných null hodnot z výsledku dotazu. [20]

```

WITH MEMBER [Mnozstvi] AS
    [Measures].[Mnozstvi Online]
    +
    [Measures].[Mnozstvi Retail]
,FORMAT_STRING = "0"
SELECT
    {
        [Measures].[Mnozstvi Online]
        ,[Measures].[Mnozstvi Retail]
        ,[Mnozstvi]
    }
ON COLUMNS,
NON EMPTY
    {[DimProvozovna].[Nazev].members}
ON ROWS
FROM [SalesCube]

```

### CurrentMember

Dotazování nad datovou kostkou je z velké části postaveno na hierarchiích. Pokud je funkce pro definici počítaných členů obohacena o klauzuli `CurrentMember`, je možné při průchodu hierarchií získat cenné informace o aktuálně zpracovávaném prvku.[21]

Pomocí konstruktů `[Mnozina].currentmember.vlastnost` je možné se nahrazením části `vlastnost` dotázat na `name` (jméno), `unique_name`, nebo `member_key`.

Metoda `CurrentMember` navíc umožňuje navigaci v rámci hierarchie o úroveň výš. K tomu slouží speciální vlastnost `parent`.<sup>15</sup>

Konkrétní případy použití a pokročilejší konstrukce jsou popsány v praktické části této práce 6, v rámci vytvoření prototypu reportovací aplikace nad datovou kostkou prodeje.

<sup>15</sup>Podrobnější informace, včetně příkladů použití jsou k dispozici na [21].



---

## Analýza současného řešení

Současné řešení není datovým skladem v pravém slova smyslu. Jedná se o relační databázi, která do značné míry kopíruje datový model hlavního systému, ze kterého pochází zpracovávaná data. Databáze dostala pracovní název Bowbeen.<sup>16</sup>

Nicméně i tak se toto řešení ukázalo být velmi užitečné. Mezi hlavní výhody patří přímý přístup k datům, jelikož systém FaRMIS, ze kterého jsou data čerpána, je datově uzavřený a neposkytuje žádné rozhraní. Další výhodou, která je vlastně součástí transformační části budoucí ETL<sup>17</sup> metody, je možnost očistit data od dále nepotřebných parametrů.

### 3.1 Produkční prostředí

Nejdůležitějším zdrojem dat je lékárenský systém FaRMIS, který poskytuje veškerá prodejní data. Další zapojené systémy, nebo systémy, které plánuje vedení společnosti vytěžovat, jsou systém Grayfox s lékařskými daty pacientů a interní systém MIS, který poskytuje data o zaměstnancích.

Zároveň jsou zde důležité datové zdroje CDB<sup>18</sup>, který poskytuje informace o kategorizaci sortimentu a dodavatelském řetězci a je aktualizovaný nepravidelně podle potřeby, a CRK<sup>19</sup>, který obsahuje informace o klientských kartách a registrovaných zákaznících. Pro data z CRK se do budoucna počítá se širším využitím. Kompletní prostředí je znázorněno na obrázku 3.1.

---

<sup>16</sup>Jedná se o vtip projektového týmu. Při výběru jména byl požadavek na tradiční český název. Zároveň bylo ale vhodné, aby zněl název světově. Jméno Bowbeen odkazuje na Boubínský prales na Šumavě.

<sup>17</sup>Popsáno v kapitole 2.3

<sup>18</sup>Centrální databáze

<sup>19</sup>Centrální Registr Kont

#### 3.1.1 Instance systému FaRMIS

Systém FaRMIS má pro každou kamennou lékárnou vlastní instanci a zároveň je uvnitř systému logicky oddělený provoz obou eshopů – rozlišení na úrovni typu pohybů v rámci skladu.

Eshop Nutri-vyziva.cz využívá pro správu skladu a logistiky kompletně služby poskytované systémem FaRMIS. Eshop Mojetabletka.cz využívá z tohoto systému pouze skladové hospodářství. Výdej a logistika zboží jsou řešeny přímo v systému eshopu.

#### 3.1.2 MS SQL Server

Současný datový sklad je umístěný na samostatném serveru se systémem Windows Server 2012 R2. Na serveru je nainstalovaná distribuce MS Server 2016 ve verzi Express.

Na počátku projektu datového skladu nebyly jasné cíle a i rozpočet byl poměrně omezený. Z toho důvodu byla zvolena varianta s MSSQL ve verzi Express, která měla obstarat první část projektu a ukázat možnosti datového skladu a jejich přínos pro skupinu.

### 3.2 Datový model databáze

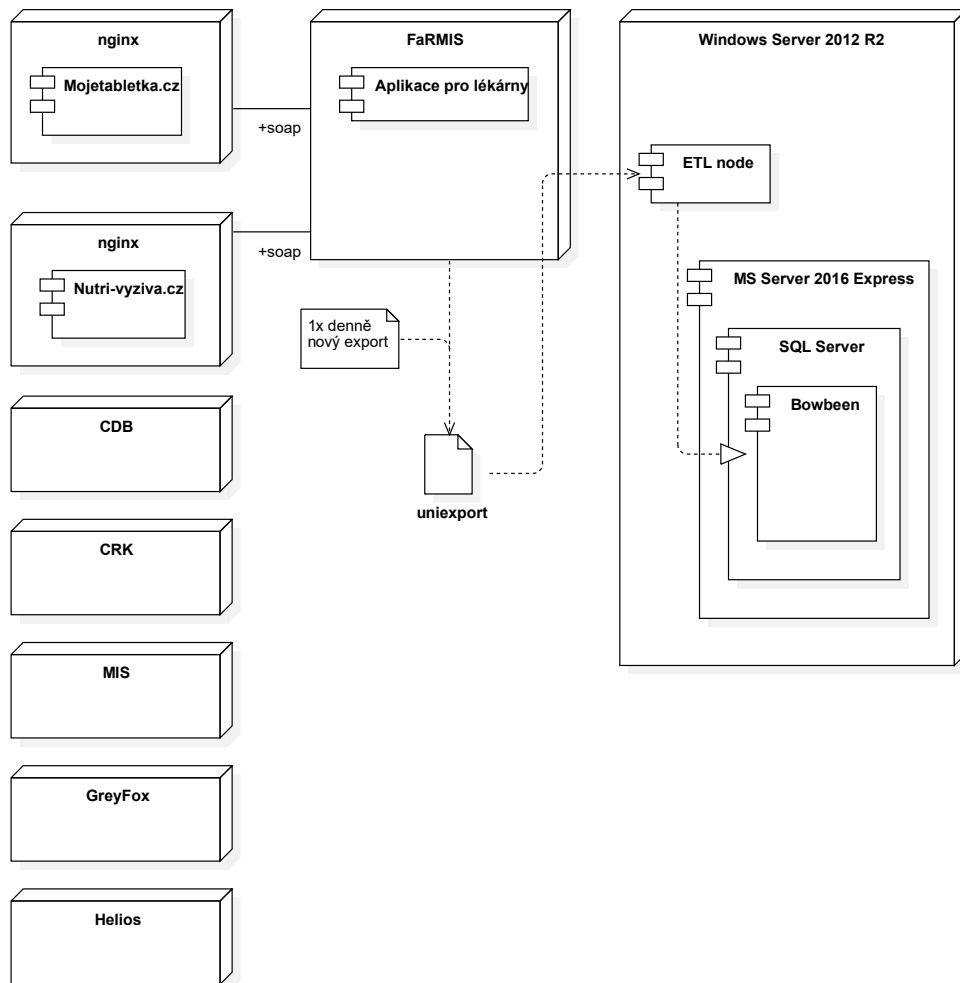
Aktuální datový model je možné vidět na obrázku 3.2. Aktuální rozvržení vychází z velké části z modelu databáze systému FaRMIS.

Hlavním a základním prvkem modelu je **Pohyb**, který reprezentuje skladovou akci provedenou na jednom druhu zboží. Tohoto zboží může být libovolný počet kusů. Produkty ale mohou být naskladněny v různých šaržích – v takovém případě je skupina produktů stejného typu rozdělena do jednotlivých pohybů podle šarží.

Důležitým parametrem je **Typ\_pohybu**, který určuje, jestli se jedná o přesun mezi sklady, nebo přímo prodej apod. Mezi další parametry patří:

- **Zbozi** – reprezentuje konkrétní typ zboží,
- **Typ\_platby** – označuje typ platby, v rámci systému jsou podporovány platby převodem, hotovost, platební karty, platby poukázkami a benefitními kartami,
- **Lekar** – odkaz na lékaře, který vydal recept, pokud jde o volně neprodejné zboží. Pomocná tabulka **Global\_lekar** slouží pro sjednocení číselníku lékařů ze systému FaRMIS a Greyfox,
- **Sklad** – identifikátor konkrétního skladu, kde se zboží nachází
- **Dodavatel** – dodavatel zboží,





Obrázek 3.1: Schéma IT systémů společnosti vzhledem k datovému skladu

- **Sazbadph** – sazba dph pro dané zboží (10 %, 15 % nebo 21 %),
- **Provozovna** – každý pohyb je vázaný na konkrétní provozovnu. Provozovna je vázána na konkrétní instanci systému FaRMIS.

Pohyby se sdružují do **Transakcí**. Transakce představuje skupinu pohybů, které provedl v rámci konkrétního připojení k systému jeden uživatel na konkrétní stanici. Může se jednat například o **pohyb** pro přesun zboží mezi sklady a následně **pohyb** pro výdej zboží zákazníkovi.

Jednotlivé transakce jsou sdruženy do **Expedičního případu**, který reprezentuje jednotlivé nákupní košíky.

## 3.3 Datové pumpy, nástroje ETL

Proces extrakce dat ze zdrojových systémů, jejich transformace a nahrání do databáze Bowbeen je postavený na několika modulech napsaných v jazyce Java – přehled na obrázku 3.3.

Moduly jsou spouštěny s pomocí Windows Plánovače úloh. Moduly buď načítají data ze souborů, nebo se připojují k databázím pomocí JDBC<sup>20</sup>.

### 3.3.1 Greyfox

Systém Greyfox bohužel neposkytuje žádné veřejné rozhraní, a tak jsou data v tuto chvíli přenášena ručně přímo z databáze. Pro transformaci a nahrání dat do Bowbeenu slouží jednoduchý program **Recepty**.

Program **Recepty** zpracovává ruční csv export ze systému Greyfox. Součástí exportu jsou zatím pouze data o receptech vystavených lékaři z poliklinik skupiny Medicon a.s.

Jak bylo již uvedeno, skupina Medicon a.s. aktuálně stojí u změny nemocničního informačního systému a proto nebude tento zdroj dat dále rozebírán.

### 3.3.2 MIS

Interní systém MIS byl spuštěn teprve v lednu roku 2017 a stále se vyvíjí, proto zatím nebylo připraveno žádné automatizované rozhraní. Potřebná zaměstnanecká data byla ručně importována a jsou aktualizovaná nadále ručně dle potřeby.

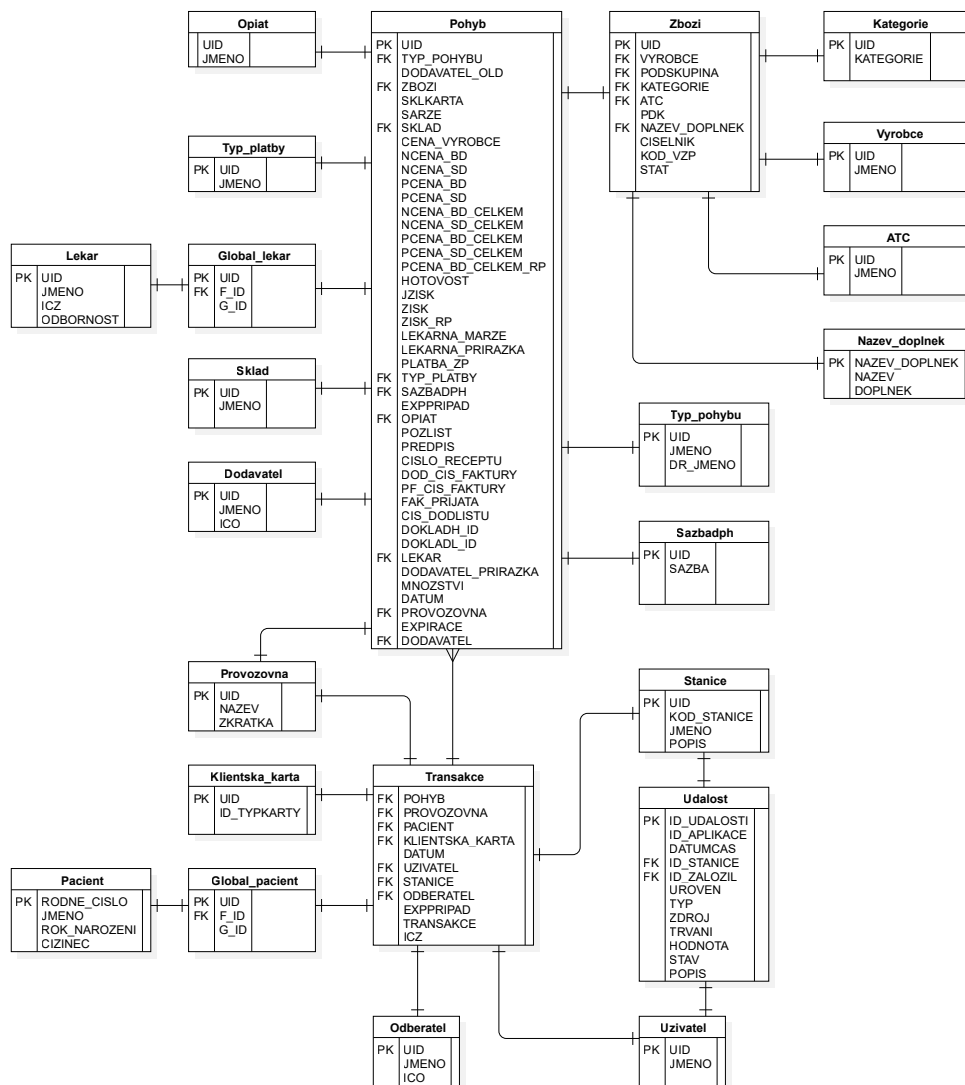
### 3.3.3 CDB – centrální databáze

Číselníky kategorií, podkategorií, dodavatelů, výrobců a další jsou v Bowbeenu aktualizovány dle potřeby pomocí dvou datových pump.

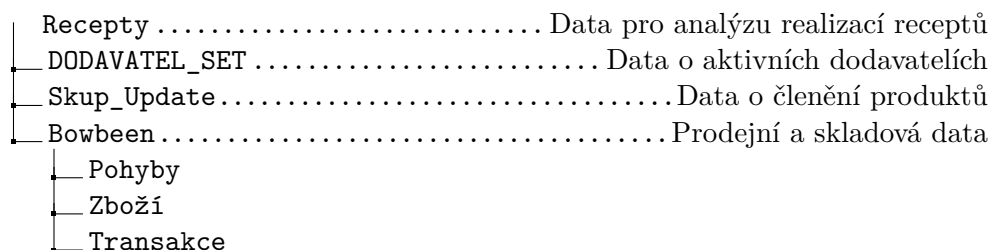
---

<sup>20</sup>Java Database Connectivity – [https://en.wikipedia.org/wiki/Java\\_Database\\_Connectivity](https://en.wikipedia.org/wiki/Java_Database_Connectivity)

### 3.3. Datové pumpy, nástroje ETL



Obrázek 3.2: Datový model současného řešení, Bowbeen. Podoba datového skladu před transformací.



Obrázek 3.3: Přehled Java ETL aplikací

**Dodavatelé** Informace o dodavatelích jsou získávány aplikací DODAVATEL\_SET. Aplikace se připojuje pomocí JDBC přímo do centrální databáze. Data jsou získávána pro každou lékárnu zvlášť – proběhnou tedy celkem čtyři iterace. Jediná změna v datech je v tomto případě přejmenování entity `Partner` na `Dodavatel`. Ihned po přejmenování jsou data nahrána do Bowbeenu. Tyto data o dodavatelích slouží pro zpětné nastavení tzv. preferovaných dodavatelů, a není tedy důležité pro další cíle této práce.

**Informace o produktech** Číselníky k produktům jsou zpracovány aplikací `Skup_Update`. V rámci zpracování je opět provedené přímé načtení z databáze CDB pomocí JDBC a následně přímé nahrání dat do MS SQL Serveru Bowbeen. Bez transformací jsou takto získána data o skupinách, podskupinách a kategoriích produktů.

#### 3.3.4 CRK – centrální registr kont

Centrální registr kont slouží pro uchování informací o klientech, kteří obdrželi zákaznickou kartu. Tato data jsou v současnosti exportována ručně a využívána pro segmentaci pro marketingové kampaně v externích systémech.

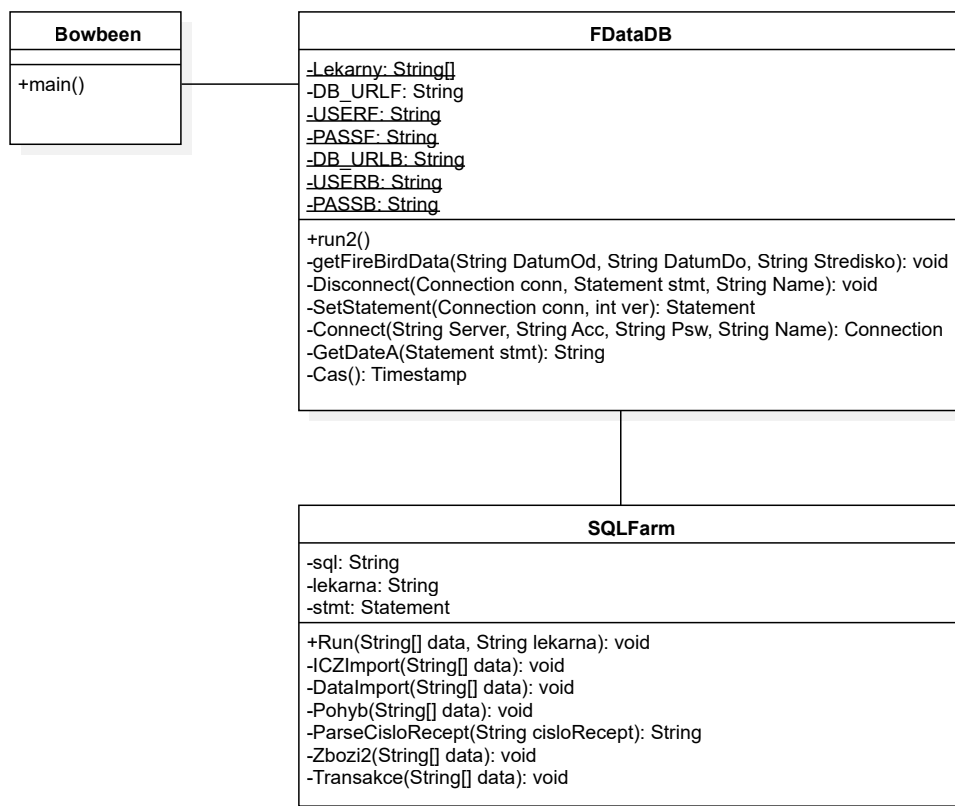
V plánu je využití těchto dat pro systém příprodejů zboží v rámci kamenných prodejen.

#### 3.3.5 FaRMIS, uniexport

Hlavní část dat současného Bowbeenu pochází ze systému FaRMIS. Data jsou s pomocí ETL uzlu zpracována každý den v jednu hodinu ráno.

Pro účely exportu dat ze systému FaRMIS byla vytvořena v databázi systému zvláštní tabulka `uniexport`, do které byl umožněn přístup zvenčí pomocí skriptů. V tabulce 3.1 je možné vidět kompletní popis uniexportu, včetně popisu důležitých atributů.

ETL uzel je skupina metod v jazyce Java. Hlavní část je popsána diagramem tříd na obrázku 3.4. Celý proces se skládá z několika částí. Nejprve je spuštěna metoda `run2()`, v rámci které je vytvořeno připojení k databázi Bowbeen pomocí JDBC.



Obrázek 3.4: Popis tříd ETL uzlu pro zpracování uniexportu z FaRMISu

Ve stejné metodě následně proběhnou 4 iterace – pro každou lékárnu (instanci systému FaRMIS) zvlášť. V rámci každé iterace je vytvořeno připojení k příslušné tabulce uniexportu v systému FaRMIS a jsou získána data z předchozího dne pomocí metody `getFireBirdData()`.

Data jsou uložena do stringu `sql`, který je následně rozparsován na jednotlivé záznamy a předaný metodě `Run()` třídy `SQLFarm`. V další fázi probíhá transformace dat. Jelikož je databáze FaRMIS hlavní zdroj dat, je brána jako referenční a transformace jsou pouze minoritní – změny názvů atributů apod.

Poslední krok je nahrání přípravečích dat do databáze Bowbeen. K tomu slouží sada metod `Pohyb()`, pro nahrání dat o pohybech, `Zbozi2()`, pro aktualizaci dat o nabízených produktech a `Transakce()`, pro nahrání dat o provedených transakcích – tedy sérii pohybů.

### 3. ANALÝZA SOUČASNÉHO ŘEŠENÍ

Tabulka 3.1: FaRMIS uniexport

| Sloupec         | Typ       | Význam   |  |
|-----------------|-----------|--|--|
| PROVOZOVNA      | integer   | Provozovna v tomto kontextu odpovídá kamenné lékárně, nebo jednomu ze dvou eshopů. |  |
| ID_POHYB        | integer   | Časové údaje prodeje.  |  |
| DATUMCAS        | timestamp |  |  |
| DATUM           | date      |  |  |
| ROK             | integer   |  |  |
| MESIC           | integer   |  |  |
| DEN             | integer   |  |  |
| HODINA          | integer   |  |  |
| MINUTA          | integer   |  |  |
| ROK_MESIC       | integer   |  |  |
| DEN_TYDNE       | integer   |  |  |
| ID_TYPPOHYB     | varchar   |  | Identifikace typu pohybu. Mezi typy pohybu patří veškeré změny, které jsou se zbožím prováděny: fyzické přesuny, výdej zboží, ale i virtuální přesuny mezi sklady. |
| TYPPOHYB        | varchar   |  |  |
| DOD_CIS_FAKTURY | integer   |  |  |
| PF_CIS_FAKTURY  |           |  |  |
| ID_FAK_PRIJATA  |           |  |  |
| CIS_DODLISTU    |           | varchar  |  |
| ID_DODAVATEL    |           | integer  |  |
| DODAVATEL       |           | varchar  |  |
| ICO_DODAVATEL   | integer   | Dodavatel konkrétní šarže daného produktu.   |  |
| ID_VYROBCE      | integer   |  |  |
| VYROBCE         | varchar   |  |  |
| ID_ODBERATEL    | integer   |  |  |
| ODBERATEL       | varchar   |  |  |
| ICO_ODBERATEL   | integer   |  | V případě prodeje na žádanku <sup>21</sup> je evidován odběratel.  |
| STREDISKO       | integer   |  |  |
| ID_PODSKUPINA   | integer   | Středisko je účetní jednotka, která v tomto případě odpovídá provozovně.           |  |

*Pokračování na další stránce*

<sup>21</sup>Prodej na žádanku je uskutečněný nejčastěji v případě objednání zboží nějakým zdravotnickým zařízením.

Tabulka 3.1 – Pokračování z předchozí stránky

| Sloupec   | Typ                           | Význam  |
|---|-------------------------------|---|
| PODSKUPINA                                      | varchar                       | Podskupina, do které spadá prodané zboží.   |
| ID_KATEGORIEZB<br>KATEGORIE<br>ZBOZI_TYP<br>ATC | integer<br>varchar<br>varchar | Kategorie prodaného zboží.<br><br>ATC skupiny slouží pro státem definované dělení léčiv podle účinných látek. <sup>22</sup> |
| UCINNA_LATKA                                    |                               | Definuje účinnou látku léčiva – na základě tohoto je léčivo zařazeno do ATC skupiny.  |
| PDK   |                               | Jedná se o speciální identifikační kód léčiv <sup>23</sup> .  |
| EAN<br>KOD_APA<br>NAZEV<br>DOPLNEK              | varchar<br>varchar            | Název produktu.<br>Doplňkový text k názvu produktu, například typ balení, dárek zdarma apod.                                |
| NAZEV_DOPLNEK                                   | varchar                       | Název produktu a doplňkový text.  |
| ID_CISELNIK<br>KOD_VZP                          | integer                       | Kód léčiva přidělený zdravotní pojišťovnou.   |
| DOKLADH_ID                                      | integer                       | Číslo dokladu, nebo receptu.  |
| DOKLADL_ID                                      | integer                       | Číslo řádku na daném dokladu.   |
| ID_ZBOZI  | integer                       |   |
| ID_SKLKARTA                                     | integer                       |   |
| SARZE   | varchar                       | Šarže konkrétního produktu.   |
| EXPIRACE  |                               |   |
| ID_SKLAD  | integer                       |   |
| SKLAD   | varchar                       | Název skladu pro lepší orientaci.   |
| MNOZSTVI  | integer                       | Množství zboží v rámci jednoho pohybu. Jedná se o konkrétní typ produktu náležící do stejné šarže při stejné prodejní ceně. |

Pokračování na další stránce

<sup>22</sup>Více o ATC skupinách na [http://www.sukl.cz/modules/medication/atc\\_tree.php](http://www.sukl.cz/modules/medication/atc_tree.php)<sup>23</sup>Více na [http://www.pharmdata.cz/ciselnik\\_pdk.htm](http://www.pharmdata.cz/ciselnik_pdk.htm)

### 3. ANALÝZA SOUČASNÉHO ŘEŠENÍ

Tabulka 3.1 – Pokračování z předchozí stránky

| Sloupec            | Typ     | Význam   |
|--------------------|---------|--|
| CENA_VYROBCE       | real    | Některá léčiva mají zákonem regulovanou cenu. V takovém případě je známá prodejní cena na straně výrobce a zároveň je stanoven limit pro koncovou prodejní cenu. |
| NCENA_BD           | real    | Nákupní cena bez DPH.  |
| NCENA_SD           | real    | Nákupní cena s DPH.  |
| PCENA_BD           | real    | Prodejní cena bez DPH.   |
| PCENA_SD           | real    | Prodejní cena s DPH.   |
| NCENA_BD_CELKEM    | real    | Nákupní cena bez DPH celkem za celé množství   |
| NCENA_SD_CELKEM    | real    | Nákupní cena s DPH celkem za celé množství.  |
| PCENA_BD_CELKEM    | real    | Prodejní cena bez DPH celkem za celé množství.   |
| PCENA_SD_CELKEM    | real    | Prodejní cena s DPH celkem za celé množství.   |
| PCENA_BD_CELKEM_RP | real    | Historická pomocná metrika pro evidenci regulačních poplatků.  |
| HOTOVOST           | real    |  |
| JZISK              | real    | Zisk za jednu položku v rámci pohybu.  |
| ZISK               | real    | Zisk za celé množství.   |
| ZISK_RP            | real    | Historická pomocná hodnota pro evidenci regulačních poplatků.  |
| LEKARNA_MARZE      | real    | Obchodní marže lékárny na daném produktu v procentech.   |
| LEKARNA_PRIRAZKA   | real    | Obchodní přírážka lékárny na daném produktu v procentech.  |
| DODAVATEL_PRIRAZKA | real    | Obchodní přírážka, kterou má na zboží nasazenou dodavatel v procentech. Je relevantní u státem cenově regulovaných produktů.                                     |
| PLATBA_ZP          |         |  |
| TYP_PLATBY         |         | Typ platby za zboží.   |
| POPIS_K            | varchar |  |
| ID_SAZBADPH        | integer |  |
| SAZBA_DPH          | real    | Příslušná sazba DPH, do které položka spadá.   |
| ID_UZIVATEL        | integer |  |

*Pokračování na další stránce*



Tabulka 3.1 – Pokračování z předchozí stránky

| Sloupec        | Typ     | Význam   |
|----------------|---------|--|
| UZIVATEL       | varchar | Zaměstnanec, který provedl pohyb.  |
| ID_STANICE     | integer | Identifikátor konkrétní táry <sup>24</sup> , na které byl prodej uskutečněn.           |
| KOD_STANICE    | varchar |  |
| STANICE        | varchar | Název stanice, na které byl zadán pohyb.   |
| ID_TRANSAKCE   | integer | Identifikátor transakce, která obsahuje daný pohyb.                                    |
| ID_EXPPRIPAD   | integer | Identifikátor nákupního košíku, ke kterému se vztahují data.                           |
| RODNECISLO     | integer | V případě výdeje na recept je evidován pacient, na kterého byl recept vydán.           |
| PACIENT        |         | V případě výdeje na recept je evidován pacient, na kterého byl recept vydán.           |
| ICZ_LEKAR      |         | V případě výdeje na recept je evidován lékař, který recept vystavil.                   |
| LEKAR          |         | V případě výdeje na recept je evidován lékař, který recept vystavil.                   |
| ID_OPIAT       | integer | Označuje, jestli a do jaké skupiny opiátů dané léčivo spadá.                           |
| OPIAT          | varchar |  |
| PREDPIS        |         | Definuje, jestli byl prodán produkt v době prodeje na pozitivním listu <sup>25</sup> . |
| CERTIFIKAT     |         |  |
| POZLIST        | varchar |  |
| ID_TYPKARTY    | integer | Rozlišení mezi starším a novějším typem klientské karty.                               |
| KLIENSKA_KARTA |         | ID klientské karty, pokud byla k nákupu použita.                                       |
| EXTERNI_DOKLAD |         |  |

<sup>24</sup>Tára je označení výdejního pultu v lékárně

<sup>25</sup>Pozitivní list slouží pro označení produktů, u kterých je prodej preferovaný před dalšími alternativami ve stejné skupině.



## Transformace datového skladu

První část projektu jejíž výsledek je databáze Bowbeen byla experimentem, jehož hlavním přínosem je extrakce dat ze systému používaných v provozu lékáren a poliklinik. Na výsledné databázi byla poté pomocí SQL dotazování analyzována nákupní data a vzniklo několik zajímavých jednorázových reportů.

Zároveň bylo vytvořeno pravidelné emailové reportování, které mělo za cíl porovnávat několik základních KPI<sup>26</sup> a jejich vývoj oproti minulému měsíci a stejném měsíci v minulém roce.

Vedení společnosti bylo s výsledky poměrně spokojeno, a proto přišla na řadu další část projektu. Součástí této fáze je vytvoření plnohodnotného datového skladu pro obchodní a marketingové oddělení, a to včetně dalších analytických nástrojů a přípravy několika aplikací navázaných na novou datovou základnu.

### 4.1 Změna produkčního prostředí

První krok v rámci transformace bylo rozšíření produkčního prostředí. Vzhledem k omezením MS SQL Serveru 2016, které přinášela verze Express bylo nutné přejít na variantu Standard. Společnost Medicon Pharm s.r.o., a i další společnosti ve skupině Medicon a.s., využívají převážně technologie společnosti Microsoft, což vedlo k volbě MS SQL Serveru 2016.

Zásadní důvody k přechodu na verzi Standard byly:<sup>27</sup>

- omezení na maximální velikost databáze (10 GB u verze Express, 524 PB u verze Standard),
- paměťové omezení (1410 MB u verze Express, 128 GB u verze Standard),

<sup>26</sup>Key performance indicator – klíčový ukazatel výkonnosti

<sup>27</sup><https://www.microsoft.com/en-us/sql-server/sql-server-2016-editions>

- omezená podpora více jader (4 jádra u verze Express, až 24 jader u verze Standard),
- verze Express neobsahuje modul MS Analysis Services,
- další omezení v oblasti Business Intelligence

Načítání dat bylo navrženo s využitím původní databáze Bowbeen. ETL uzel pro načtení dat z uniexportu systému FaRMIS a dalších podpůrných systémů zůstal zachovaný a byl vytvořený nový ETL uzel, který transformuje data z databáze Bowbeen a načítá je do datové skladu. Toto řešení umožňuje zachovat všechny současné aplikace napojené na starší databázi, recyklovat již připravené řešení ETL a rozložit zátěž při aktualizaci dat na více kroků.

## 4.2 Architektura datového skladu

Při volbě architektury datového skladu hrály velkou roli požadavky managementu společnosti. Byl kladen velký důraz na snížení vstupních nákladů při dodání alespoň minimální hodnoty v co nejkratším čase.

Důležitým faktorem bylo také, že při vzniku této práce a při zadávání požadavků na nový datový sklad, probíhalo výběrové řízení na nový nemocniční informační systém – nebyly známy specifikace nového systému a nebylo by tedy možné navrhnout komplexní řešení, jako je v případě návrhu postaveném na top-down 2.1.1 přístupu vyžadováno.

Dalším výběrovým kritériem byly požadavky na data, která by měla být analyzována. Většina datové základny, o kterou měl management společnosti zájem pro první fázi projektu byla v oblasti prodeje.

Právě z těchto důvodů byla zvolena realizace datového skladu dle R. Kimballa 2.1.2 a jako první vytvoření datového tržiště pro obchodní oddělení.

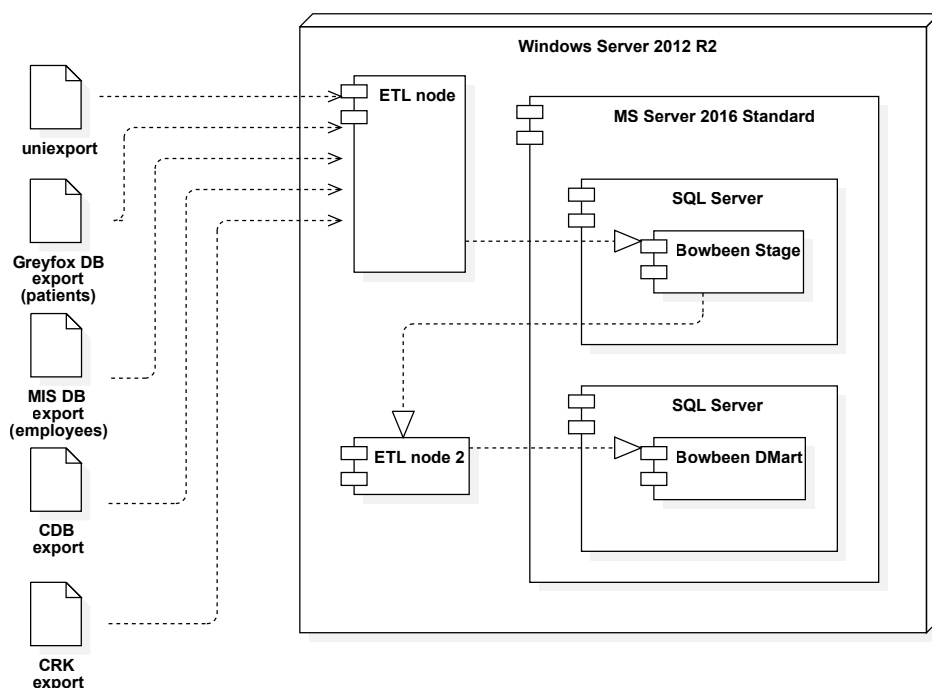
Na schématu 4.1 je znázorněno nasazení nových bloků oproti původnímu stavu 3.1. V novém řešení se počítá s MS SQL serverem 2016 ve verzi Standard. Původní databáze Bowbeen bude nově sloužit jako odkládací stage area.

Nově vzniká datové tržiště **Bowbeen DMart**, které je postavené na modifikovaném dimenzionálním datovém modelu 4.3. Data jsou ze stage area nahrávána pomocí druhého ETL uzlu 4.4.

Zdrojové systémy zůstávají stejné jako v prvním případě. O budoucím řešení nového nemocničního informačního systému společnosti zatím nebylo rozhodnuto.

## 4.3 Nový datový model

Současná databáze Bowbeen, datový model je znázorněný na obrázku 3.2, je v podstatě kopií hlavního zdrojového systému, což vede k velkému množství



Obrázek 4.1: Schéma IT systémů společnosti vzhledem k datovému skladu

nepotřebných dat a parametrů, které jsou ukládány. Databáze je navíc navržena v klasickém relačním stylu, a proto není vhodná pro komplikovanější a komplexnější analytické dotazování.

V první fázi projektu datového skladu bylo na základě požadavku rozhodnuto realizovat datové tržiště pro potřeby obchodního oddělení. Pro návrh datového modelu byla zvolena metodika popsaná v části 2.2.2.

#### 4.3.1 Volba business procesů

Hlavní proces, který bude předmětem modelování, je pro obchodní oddělení prodej. V rámci zdrojového systému je prodej definovaný jako expediční případ a reprezentuje každý nákupní košík.

Každý expediční případ se může skládat z několika transakcí, v rámci kterých je proveden jeden a více pohybů zboží. Pro potřeby modelování prodeje nebude nutné dělení na pohyby sledovat, jelikož se jedná spíše o provozní data.

#### 4.3.2 Definice zrnitosti – granularita

Granularita navazuje na předchozí bod. Při modelování prodejů může být základní položkou nákupní košík (expediční případ). Lepší volbou bude ale

soustředit se na jednotlivé položky nákupního košíku. V tomto případě tedy jeden prodaný produkt. Tím bude splněna podmínka na atomicitu dat v datovém skladu. V tomto případě odpovídá položka nákupního košíku jednomu řádku na účtence při prodeji zboží.[22][23]

Pro konečnou volbu je ale důležité brát v potaz i velikost konečné tabulky faktů. V lékárnách se poměrně často některé produkty prodávají v objemech po více produktech. V takovém případě by bylo v tabulce udržováno pro takový prodej i několik v podstatě totožných řádků. Z toho důvodu je konečnou volbou pro základní jednotku libovolný počet produktů stejné šarže prodaných v rámci jednoho expedičního případu a se stejnou prodejní cenou.

Z pohledu managementu bude důležité rozlišovat data na úrovni jednotlivých dnů, konkrétních zákazníků, prodejců a samozřejmě je důležité, jaký konkrétní produkt se prodal.

### 4.3.3 Volba dimenzí

[23]

Jedná se o první projekt v oblasti Business Intelligence ve společnosti, a proto bylo poměrně náročné identifikovat vhodné dimenze. V rámci analýzy požadavků managementu byl kladen důraz na maximální zjednodušení problémové domény omezení atributů a parametrů, které nejsou zásadní pro následné reportování výsledků pro obchodní oddělení.

Pro návrh prvního prototypu byla zvolena následující sada dimenzí, kterou dle domluvy bude možné mírně modifikovat po ukončení testovacího provozu.

**DimDatum** Základní časová dimenze, která je kostrou každého datového skladu. Dle [23] bylo rozhodnuto pro implementaci vlastní časové dimenze, bez využití vnitřních nástrojů databázového serveru.

Výhodou je, že se vyhneme závislosti na konkrétní technologii a budeme mít možnost zaznamenat významné dny jako svátky apod. Navíc získáme lepší podporu pro indexování.

Základní jednotkou časové dimenze bude jeden den. Zároveň bude vytvořena následující časová hierarchie:

$$Den \rightarrow Týden \rightarrow Měsíc \rightarrow Čtvrtletí \rightarrow Rok$$

**DimTypDokladu** Poměrně důležitá z hlediska managementu je dimenze pro rozlišení jednotlivých typů dokladů. Typ dokladu v tomto případě reprezentuje typ prodeje z hlediska vztahu zdravotní pojišťovny k danému expedičnímu případu.

V současné době jsou rozlišovány následující typy dokladů:

- **Recepty** jsou prodeje na základě receptu vystaveného lékařem.

- **Poukazy** reprezentují speciální prodej zdravotnických pomůcek. Poukazy jsou vystaveny lékařem.
- **Žádanky** zastupují prodej zdravotnického materiálu a léčiv na základě žádosti zdravotnického zařízení (nemocnice, polikliniky, nebo samostatné ordinace).
- **Volný prodej** zastupuje všechny ostatní expediční případy.

**DimTypPlatby** Důležitá dimenze pro rozlišení jednotlivých typů plateb. Rozlišuje se:

- hotovost,
- platební karty,
- poukázky,
- benefitní karty,
- platební příkazy.

**DimZakaznik** Zatím okrajová dimenze, která ale nejspíše získá na důležitosti spolu s integrací dalších systémů. V současné době je zákazník identifikován dvojitým způsobem:

1. Jako zákazník s klientskou kartou,
2. jako pacient s platným receptem.

**DimProvozovna** Důležitá dimenze pro segmentaci na úrovni jednotlivých kamenných lékáren (případně eshopů). V současné době jsou k dispozici následující provozovny:

- Lékárna Zelený pruh,
- Lékárna Budějovická,
- Lékárna DBK,
- Lékárna Vysočany,
- Eshop Nutri-vyziva.cz,
- Eshop Mojetabletka.cz.

**DimUživatel** Uživatel v tomto kontextu označuje prodejce, který se váže k danému prodeji. Tato dimenze bude sloužit převážně pro výpočet bonusů a kontrolu výkonu jednotlivých lékárníků.

Vzhledem k možnému přesunu pracovníků mezi jednotlivými provozovny v průběhu času a vzhledem ke zřízeným tzv. plovoucím účtům, není možné tuto dimenzi spojit s dimenzí DimProvozovna a je třeba ji řešit zvlášť.

**DimProdukt** Nejkomplexnější dimenze modelu popisuje předmět prodeje jako takového – produkt. Produktová dimenze je hierarchicky rozšířená dle návrhového vzoru schématu sněhové vločky 2.8 o další upřesňující tabulky.

Hlavní dimenzionální tabulka je popsána v 4.1. Na tuto tabulku navazují doplňující tabulky:

- DimPodskupina – označuje podskupinu, do které je výrobek zařazený<sup>28</sup>,
- DimVyrobcce – výrobce daného výrobku,
- DimATC – označení skupiny dle účinné látky léčiva,
- DimKategorie – kategorie zařazení výrobku,
- DimDodavatel – dodavatel dané položky.

Tabulka 4.1: Atributy produktové dimenze.

| Parametr     | Význam   |
|--------------|--|
| UID          | PK   |
| SazbaDPH     | Sazba DPH (10 %, 15 %, nebo 21 %)  |
| PDK          | Označení léčiva  |
| NazevDoplnek | Název produktu + text doplňkové informace (jedná se o upřesnění balení, nebo způsobu podání léčiva). |
| KodVZP       | Kód přidělený léku zdravotní pojišťovnou.  |
| NID          | Identifikátor ve zdrojovém systému.  |

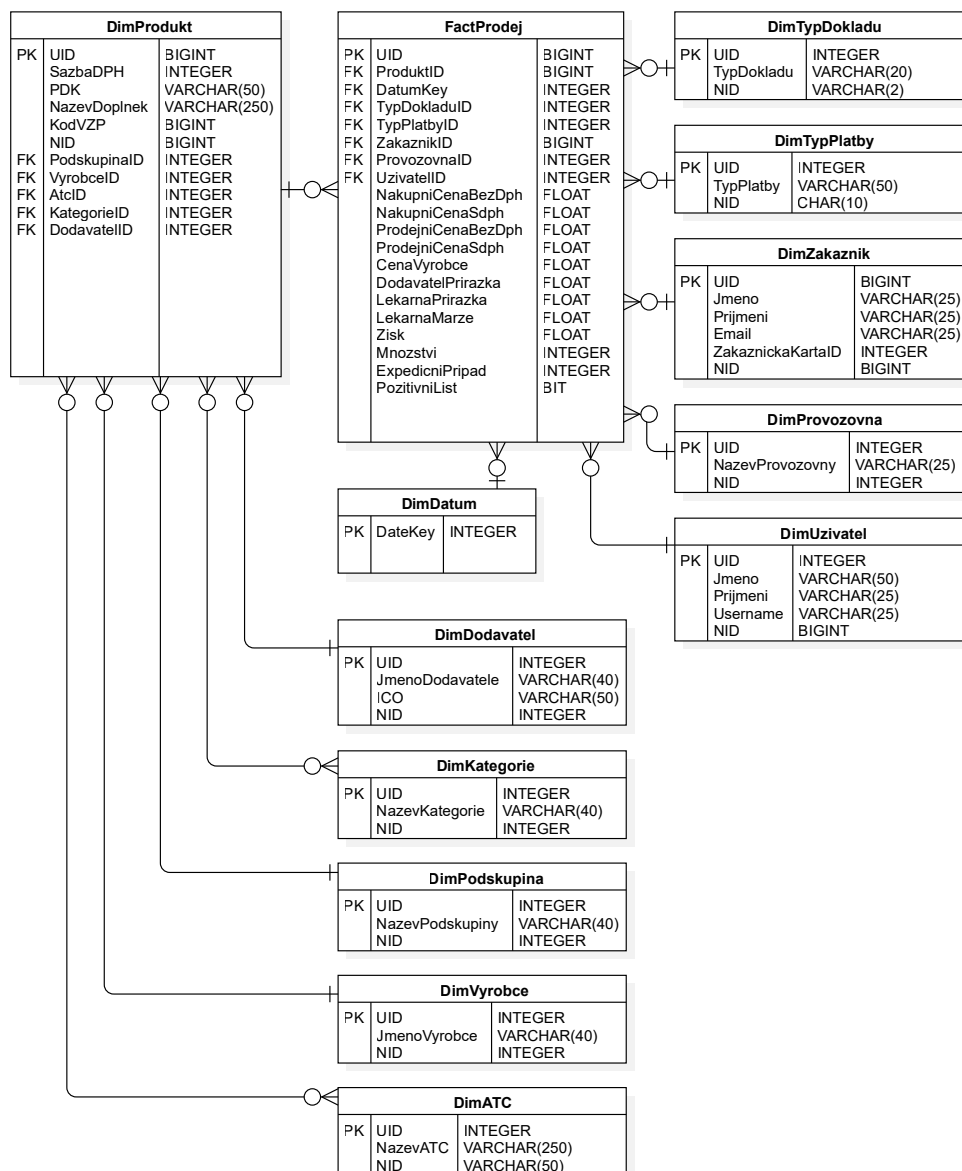
#### 4.3.4 Identifikace faktů

Při identifikaci faktů je třeba se zeptat, jaké údaje budeme u procesu měřit. V tomto případě bylo pracováno se dvěma základními vstupy. Nejprve byla provedena analýza dostupných údajů a poté byly na základě požadavků managementu vybrány ty podstatné.

<sup>28</sup>Z historických důvodů jsou produkty členěny zvlášť do kategorií a zvlášť do podskupin. V případě produktů zdravé výživy a zdravotnických pomůcek je hierarchie Podskupina → Kategorie. V případě volného prodeje jsou podskupiny a kategorie dva rozdílné přístupy na stejné úrovni hierarchie.



### 4.3. Nový datový model



Obrázek 4.2: Dimenzionální datový model tržiště Bowbeen DMart.

Kompletní přehled zvolených metrik je popsán v tabulce 4.2. Některé nové metriky bylo třeba dopočítat ze známých informací. Konkrétně se jedná o následující údaje:

##### Zisk

$$Zisk = ProdejniCenaBezDph / NakupniCenaBezDph$$

##### LekarnaPrirazka

$$LekarnaPrirazka = Zisk / ProdejniCenaBezDph$$

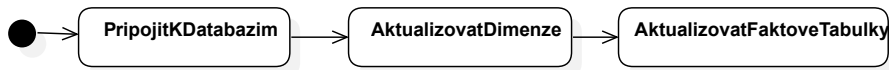
##### LekarnaMarze

$$LekarnaMarze = Zisk / NakupniCenaBezDph$$

## 4.4 Druhý modul ETL

Druhý ETL modul slouží pro přenos a transformaci dat ze stage area (data-báze Bowbeen) do nového datového tržiště Bowbeen DMart. V rámci tohoto procesu je potřeba očistit data od přebytečných parametrů a z velké části transformovat strukturu datového modelu. Díky tomu, že data v Bowbeen DB již prošla prvním ETL uzlem, není třeba v tomto kroku provádět žádné složitější transformace na úrovni jednotlivých atributů.

Rozsah dat a vlastně i počet změn není velký. Z toho důvodu bylo rozhodnuto nevyužít Microsoft Integration Services, ale provést potřebné procesy s pomocí jednoduchých komponent v jazyce Java. Výhodou tohoto řešení je navíc možnost znovupoužití částí již hotových ETL metod z prvního ETL uzlu.



Obrázek 4.3: Hlavní proces 2. ETL uzlu.

### 4.4.1 Dimenze ATC

ATC skupiny je možné pouze přejmenovat a okopírovat. V rámci této dimenze není třeba řešit problém měnících se dimenzí 2.2.3, jelikož ATC skupiny jsou dány a z podstaty věci se nemění, maximálně vznikají nové.

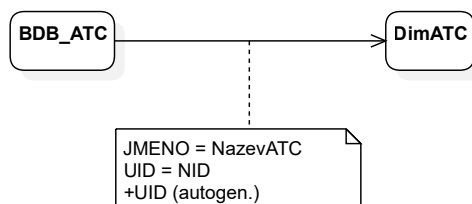
Tabulka 4.2: Přehled parametrů tabulky faktů dimenzionálního modelu.

| Parametr           | Význam  |
|--------------------|---|
| UID                | Unikátní identifikátor v rámci datového skladu  |
| ProduktID          | FK pro dimenzi Produkt  |
| DatumKey           | FK pro dimenzi Datum  |
| TypDokladuID       | FK pro dimenzi TypDokladu   |
| TypPlatbyID        | FK pro dimenzi TypPlatby  |
| ZakaznikID         | FK pro dimenzi Zakaznik   |
| ProvozovnaID       | FK pro dimenzi Provozovna   |
| UzivatelID         | FK pro dimenzi Uzivatel   |
| NakupniCenaBezDph  | Nákupní cena bez DPH  |
| NakupniCenaSdph    | Nákupní cena s DPH  |
| ProdejniCenaBezDph | Prodejní cena bez DPH   |
| ProdejniCenaSdph   | Prodejní cena s DPH   |
| CenaVyrobce        | Cena, za kterou výrobek poskytuje jeho výrobce.*  |
| DodavatelPrirazka  | Přirážka, kterou na daný produkt nastavil daný dodavatel.**                                       |
| LekarnaPrirazka    | Přirážka stanovená lékárnou. Výpočet viz 4.3.4.   |
| LekarnaMarze       | Marže stanovená lékárnou. Výpočet viz 4.3.4.  |
| Zisk               | Zisk lékárny. Výpočet viz 4.3.4.  |
| Mnozstvi           | Množství stejných produktů v rámci konkrétní šarže prodaných v rámci jednoho expedičního případu. |
| ExpedicniPripad    | Číslo expedičního případu, neboli označení nákupního košíku                                       |
| PozitivniList      | Značí, jestli byl v tomto případě prodán produkt z pozitivního listu.***                          |

\*Některá léčiva jsou regulovaná státem na úrovni prodejní ceny v jednotlivých fázích distribuce k zákazníkovi.

\*\*Počítá se v případě produktů se státem regulovanou cenou.

\*\*\*Pozitivní list je skupina produktů, které jsou v prodeji preferované v rámci dané skupiny.

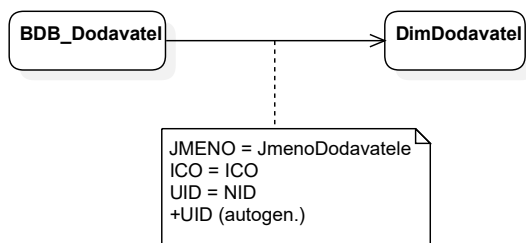


Obrázek 4.4: ETL metoda pro dimenzi ATC.

#### 4.4.2 Dimenze Dodavatel

Společnost Medicon Pharm s.r.o. využívá jednotky různých dodavatelů. Za sledovanou historii u těchto společností nenastala žádná změna ve sledovaných parametrech.

Pokud by taková změna nastala (například změna názvu společnosti, nebo identifikačního čísla), management společnosti nepotřebuje rozlišovat historii změn těchto parametrů.

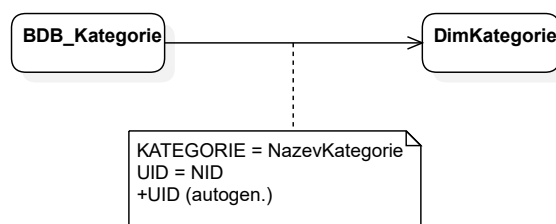


Obrázek 4.5: ETL metoda pro dimenzi Dodavatel.

#### 4.4.3 Dimenze Kategorie

Kategorie produktů je spolu s dimenzí Podskupin hlavní způsob členění produktů. Z databáze Bowbeen se pouze kopírují názvy kategorií a identifikátory.

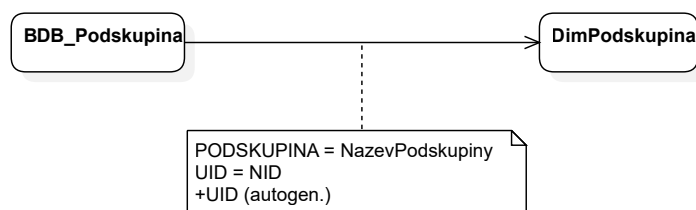
Kategorie produktů je dimenze, která se s časem může měnit. Změny nicméně nepotřebují historickou evidenci. V každém okamžiku bude uživatel pracovat s aktuálním členěním kategorií.



Obrázek 4.6: ETL metoda pro dimenzi Kategorie.

#### 4.4.4 Dimenze Podskupina

Podskupiny jsou řešeny stejným způsobem jako dimenze Kategorie. I přes možné změny není třeba tyto změny evidovat a je možné je řešit 1. variantou řešení pro pomalu se měnící dimenze 2.2.3.



Obrázek 4.7: ETL metoda pro dimenzi Podskupina

#### 4.4.5 Dimenze Produkt

Dimenze produkt je nejkomplexnější – zdrojem jsou tři různé tabulky. Konkrétní transformace je znázorněna na diagramu 4.8.

V rámci atributů produktové dimenze je potencionálně největší možnost různých změn. Vzhledem k rozšiřujícím tabulkám je navíc potřeba řešit i kaskádově změny v těchto tabulkách.

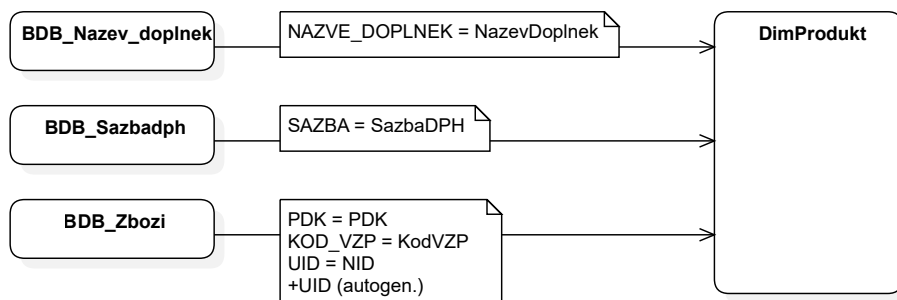
Je potřeba řešit následující situace:

- SazbaDPH – v případě, že dojde ke změně sazby dph, bude se tato změna řešit 3. variantou pro vyjádření změn pomalu měnících se dimenzí 2.2.3.
- PDK – neměnný kód.
- NazevDoplnek – základní název produktu se mění z různých důvodů, které jsou důležité i pro přístup k takové změně. Jednotlivé případy budou řešeny individuálně.
- KodVZP – neměnný kód.

#### 4. TRANSFORMACE DATOVÉHO SKLADU

---

- DimPodskupina - není třeba řešit změny.
- DimVyrobcce - není třeba řešit změny.
- DimATC - není třeba řešit změny.
- DimKategorie - není třeba řešit změny.
- DimDodavatel - v případě změny dodavatele pro daný produkt bude změna zaznamenána 3. variantou pro vyjádření změn pomalu měnících se dimenzí 2.2.3.



Obrázek 4.8: ETL metoda pro dimenzi Produkt.

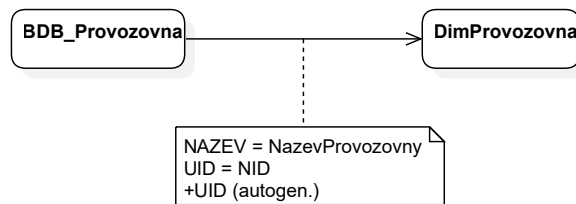
#### 4.4.6 Dimenze Provozovna

Dimenze pro provozovny v současné době odpovídá čtyřem kamenným lékárnám. Po zkušebním provozu budou doplněny dvě provozovny odpovídající eshopům.

Z hlediska sledování změn se jedná o pomalu se měnící dimenzi. Mohou nastat situace, kdy se objeví nová instance provozovny, nebo je možné, že bude některá provozovna rozpadena na více menších provozoven. Poslední zmíněná situace je velmi reálná u lékárny na Budějovické, kde je zároveň prodejna zdravé výživy. Tato prodejna by mohla v budoucnu nově vytvořit samostatnou provozovnu. Stejný osud by mohl potkat nově vzniklou prodejnu zdravé výživy v lékárně na Vysočanské.

Z tohoto pohledu bude v tomto bodě důležité rozhodnout, jakým způsobem vyřešit rozpadnutí metrik z jedné provozovny na více – toto totiž povede ke snížení metrik v původních provozovnách.

Pravděpodobnější bude vytvoření podrobnější hierarchie dělení prodejen – toto prozatím nebylo rozhodnuto.



Obrázek 4.9: ETL metoda pro dimenzi Provozovna.

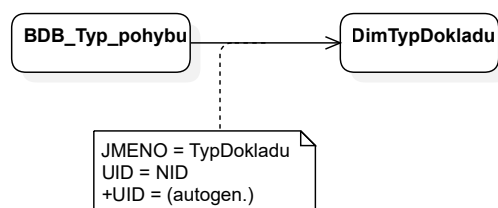
#### 4.4.7 Dimenze TypDokladu

V případě koncového pohybu zboží, kdy se uzavírá expediční případ a dochází k samotnému prodeji, je možné dle typu pohybu rozlišit jednotlivé typy dokladů tak, jak byly uvedeny v 4.3.3.

V současnosti budou rozlišovány čtyři základní skupiny dokladů, které budou ze zdrojové tabulky pohybu seskupeny následujícím způsobem:

- Recepty
  - Výdej receptem (VR)
- Poukazy
  - Výdej poukazem (UID=VP)
- Volný prodej
  - Výdej volným prodejem (UID=VV)
- Žádanky
  - Výdej žádankou (UID=VZ)
  - Výdej nemocniční žádankou (UID=VN)
  - Výdej obecnou žádankou/objednávkou (UID=VO)
  - Výdej žádankou statim (VS)

Ze samotné podstaty této dimenze se nepočítá s žádnými závažnými změnami. Pokud bude nějaký typ dokladu přejmenován, není potřeba uchovávat záznam o této změně. Kdyby došlo k rozštěpení jednoho typu dokladu na více podskupin, bude se tento problém řešit v závislosti na dopadu této změny – nejspíše s pomocí 2. varianty pro pomalu se měnící dimenze 2.2.3.

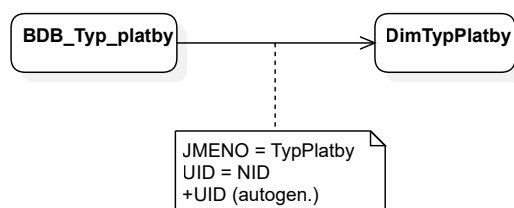


Obrázek 4.10: ETL metoda pro dimenzi TypDokladu.

#### 4.4.8 Dimenze TypPlatby

Transformace typu plateb je přímočará a pro lepší orientaci je opět pouze přejmenován hlavní atribut.

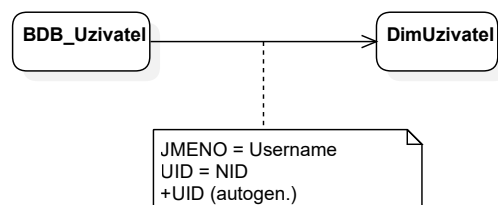
V rámci změn je možné, že přibudou nové možnosti platby – například benefiční poukázky apod.



Obrázek 4.11: ETL metoda pro dimenzi TypPlatby.

#### 4.4.9 Dimenze Uzivatel

Dimenze uživatelů bude sloužit pro hodnocení výkonu jednotlivých lékárníků a lékárníků. Z pohledu měnících se dimenzí je možné všechny změny řešit první variantou pro pomalu se měnící dimenze 2.2.3.



Obrázek 4.12: ETL metoda pro dimenzi Uzivatel.

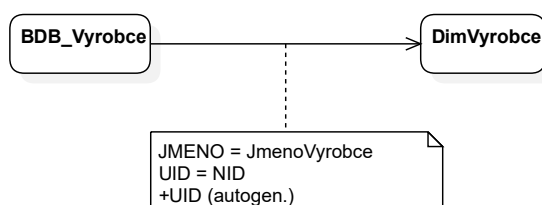


#### 4.4.10 Dimenze Vyrobcce

Výrobci jsou historicky poměrně komplikovaná dimenze. Vzhledem k několika různým dodavatelům a různým značením výrobců. V rámci lékáren navíc vznikala vlastní označení v průběhu času, a tak bylo potřeba zavést sjednocující číselník výrobců, které seskupují různé názvy pod reálné výrobce.

V datovém tržišti je použitý právě tento jednotný číselník, díky tomu se nepředpokládají žádné další změny. Pokud přece jen změny nastanou, bude zavedený nový výrobce v externím systému a sjednocení dat se vyřeší v rámci jmenovaného číselníku.

Pro sledování historie nejsou kladeny požadavky na uchování historie těchto změn.



Obrázek 4.13: ETL metoda pro dimenzi Vyrobcce.

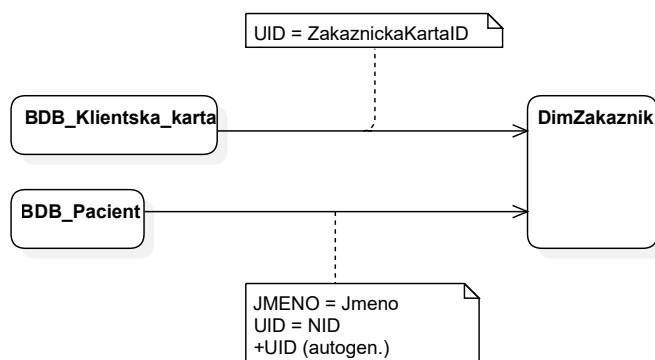
#### 4.4.11 Dimenze Zakaznik

Zákaznická dimenze je kombinací dat ze dvou zdrojových tabulek:

- Pacient – data jsou relevantní v případě, že zákazník přišel s receptem vydaným jednou z poliklinik společnosti.
- Klientska\_karta – informace související se zákaznickým účtem a příslušné klubové kartě. Data jsou dostupná pouze v případě, že zákazník na pokladně předložil klubovou kartu.

Data jsou čistě orientační. Do budoucna se počítá se sjednocením zákaznické datové základny a rozšířenými možnostmi filtrování a porovnávání prodejních dat dle zákaznických segmentů.

Jedná se pouze o experimentální dimenzi, a proto nebudou změny v rámci této dimenze řešeny.



Obrázek 4.14: ETL metoda pro dimenzi Zakaznik.

## 4.5 Implementace změn

### 4.5.1 Nastavení SQL serveru

Pro přípravu nového prostředí bylo potřeba nainstalovat dvě základní komponenty:

- MS SQL Server 2016 Developer <sup>29</sup>
- MS SQL Server Management Studio (SSMS) <sup>30</sup>

Instalace SQL serveru probíhá pomocí instalačního centra, které je staženo v rámci instalačního souboru pro MS SQL Server 2016. V nabídce v levém panelu je třeba zvolit položku Installation a následně vybrat první možnost – New SQL Server stand-alone installation.

Spuští se instalační proces, v rámci kterého je možné modifikovat nastavení serveru a zvolit možné doplňky. Pro naše použití je důležité:

- v části Feature Selection zvolit **Analysis Services**,
- v části Database Engine Configuration a záložce Data Directories nastavit adresáře pro data a zálohy – v našem případě zvolíme pevný disk E, který je fyzicky na SSD disku,
- v části Analysis Services Configuration necháme nastavený Multidimenzionální mód serveru a přidáme do služby uživatele s přístupem,
- v záložce Data Directories opět nastavíme adresáře dle potřeby.

<sup>29</sup><https://www.microsoft.com/en-us/sql-server/application-development>

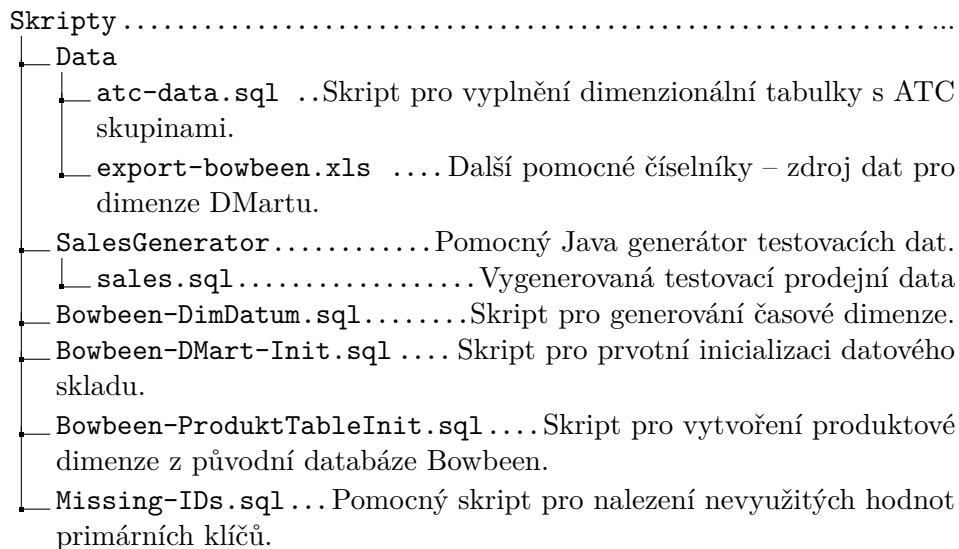
<sup>30</sup><https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms>

### 4.5.2 Vytvoření databáze

Pro vytvoření databáze z dimenzionálního datového modelu bylo připraveno několik sql skriptů, které jsou dostupné jako příloha k této práci 4.5.2.

Pro vygenerování časové dimenze byla použita varianta skriptu [24] upraveného pro české prostředí.

Konkrétně bylo potřeba upravit číslování dnů v týdnu, následně způsob výběru víkendových dnů a na závěr přepracovat výběr svátků a prázdnin pro české prostředí.



Obrázek 4.15: Přehled sql skriptů pro datové tržiště

### 4.5.3 Testovací data

Přípravu ETL metod bude mít na starosti jiná společnost. Jelikož nebylo možné připravit tyto nové metody v termínu pro prezentaci chystaného řešení, bylo rozhodnuto připravit testovací databázi.

Databáze vychází z reálného prostředí, kdy byly využity číselníky následujících zdrojů:

- Bowbeen/číselník dodavatelů
- Bowbeen/číselník kategorií
- Bowbeen/číselník podskupin
- Bowbeen/číselník atc skupin
- Bowbeen/číselník výrobců

- FaRMIS/číselník typů dokladů
- FaRMIS/číselník typů plateb
- FaRMIS/číselník provozoven
- FaRMIS/číselník uživatelů
- Bowbeen/Produktová tabulka

Datová dimenze byla připravena v rozsahu let 2015 – 2030.

Na závěr byl připravený generátor testovacích prodejních dat pro tabulku faktů `FactProdej`. Generátor je možné přizpůsobit několika parametry, které jsou popsány v tabulce 4.3.

Výpočet obchodních metrik odpovídá aktuálním zákoným normám. Zde je ukázka části funkce pro výpočet těchto metrik `getFinancialsValues()`, jejíž část bude možné použít i v rámci budoucího 2. ETL uzlu.

```
String getFinancialsValues(int amount) {
    String line = ""; double dph;

    double cenaVyrobce = amount*randomDoubleInRange(1,
        CENA_VYROBCE_MAX+1);
    double dodavatelPrirazka = randomIntInRange(1,
        DODAVATEL_PRIRAZKA_MAX+1);

    double ncbd =
        (dodavatelPrirazka/100*cenaVyrobce + cenaVyrobce);

    switch (randomIntInRange(1,3)) {
        case 1:dph = 1.10;break;
        case 2:dph = 1.15;break;
        default:dph = 1.21;break;
    }

    ncsd = (dph*ncbd);
    double lekarnaPrirazka = randomIntInRange
        (LEKARNA_PRIRAZKA_MIN,LEKARNA_PRIRAZKA_MAX);

    double pcbd = ncsd*(1+lekarnaPrirazka/100);
    double pcsd = dph*pcbd;
    double lekarnaMarze = 100*((pcbd - ncbd)/pcbd);
    double zisk = pcbd - ncbd;

    ... return line; }
```

Tabulka 4.3: Přehled parametrů generátoru testovacích dat.

| Parametr               | Význam  |
|------------------------|---|
| PRODUKT_NUM            | Parametr označuje počet produktů v databázi. Slouží pro hranici generování cizích klíčů do tabulky produktů.          |
| SELL_YEAR_START        | Parametr označuje počáteční rok, pro který se generují prodeje.   |
| SELL_YEAR_END          | Parametr označuje koncový rok, pro který se generují prodeje.   |
| TYP_DOKLADU_NUM        | Parametr označuje počet typů dokladů v databázi. Slouží pro hranici generování cizích klíčů do tabulky typů dokladů.  |
| TYP_PLATBY_NUM         | Parametr označuje počet typů plateb v databázi. Slouží pro hranici generování cizích klíčů do tabulky typů plateb.    |
| ZAKAZNIK_NUM           | Parametr označuje počet zákazníků v databázi. Slouží pro hranici generování cizích klíčů do tabulky zákazníků.        |
| PROVOZOVNA_NUM         | Parametr označuje počet provozoven v databázi. Slouží pro hranici generování cizích klíčů do tabulky zákazníků.       |
| UZIVATEL_NUM           | Parametr označuje počet uživatelů v databázi. Slouží pro hranici generování cizích klíčů do tabulky uživatelů.        |
| MNOZSTVI_MAX           | Počet prodaných ks zboží. V 80% je prodán 1ks zboží. Ve zbylých 20% případů je prodáno od 1 do MNOZSTVI_MAX ks zboží. |
| CENA_VYROBCE_MAX       | Vstupní cena produktu je náhodná hodnota mezi 1 a CENA_VYROBCE_MAX.   |
| DODAVATEL_PRIRAZKA_MAX | Maximální možná přírážka na straně dodavatele zboží.  |
| LEKARNA_PRIRAZKA_MIN   | Minimální možná přírážka na straně lékární.   |
| LEKARNA_PRIRAZKA_MAX   | Maximální možná přírážka na straně lékární.   |



# Návrh webové aplikace pro reportování

V první části návrhu byla realizována analýza trhu podobných řešení. V návaznosti na analýze dostupných alternativ byl v několika iteracích provedený sběr požadavků managementu společnosti Medicon Pharm s.r.o.

Na základě požadavků na aplikaci byl připravený návrh, včetně zpracování uživatelského prostředí, volby architektury a byl připravený přehled komponent, ze kterých by se budoucí reportovací aplikace měla skládat.

## 5.1 Existující aplikace

V současné době existuje velké množství Business Intelligence řešení. Ne všechna dostupná řešení jsou ale vhodná pro využití v lékárenském prostředí, které má z důvodů státní regulace celou řadu specifík.

Při snaze najít vhodná hotová řešení v oblasti Business Intelligence v lékárenství na českém trhu bylo zjištěno, že většina lékáren spadá do dvou základních skupin:

### Lékárna je součástí většího řetězce

V tomto případě využívá technologie vyvinuté na míru danému řetězci. Konkrétně lékárny z řetězců Dr.Max<sup>31</sup> a Magistra<sup>32</sup> spolupracují se společností CEOS Data s.r.o.<sup>33</sup>

V tomto případě si řetězce mohou dovolit mnohem větší investice do vývoje takového řešení a pro skupinu lékáren společnosti Medicon Pharm s.r.o. toto řešení není dostupné.

---

<sup>31</sup><https://www.drmax.cz>

<sup>32</sup><http://www.magistra.cz>

<sup>33</sup><http://www.ceosdata.com>

### Lékárna není součástí řetězce

Případ, kdy je lékárna samostatná, nebo se jedná pouze o jednotky lékáren v rámci daných společností je stále ještě relativně rozšířený. Často se jedná o lékárny přidružené ke zdravotnickým zařízením apod.

Takovéto lékárny nemají prostředky na drahé Business Intelligence řešení, a proto zatím žádná vlastní ani cizí řešení nevyužívají. Několik společností využívá nějaké obecné řešení bez přizpůsobení konkrétní business doméně lékárenství, nebo vyvíjí vlastní řešení.

Z těchto zjištění vychází potřeba vytvořit relativně levné řešení na míru dané business doméně. Pokud by se podařilo vyvinout dané řešení modulárně a přizpůsobitelně, bylo by možné v dalších fázích projektů financovat další vývoj i díky možnému nasazení mimo skupinu Medicon Pharm s.r.o.

Z důvodu nedostatku dostupných řešení v rámci dané domény a také z důvodů požadavků managementu, který od počátku požaduje vyvinout řešení na míru, byl provedený průzkum nejznámějších řešení v oblasti datové analýzy pro potřeby upřesnění funkčních požadavků na vlastní aplikaci.

#### 5.1.1 WebManager společnosti Apatyka

Jeden ze vstupních modelů budoucího řešení je i současný interní nástroj využívaný ve společnosti Pronatal s.r.o., která byla připojena ke skupině Medicon a.s. na počátku roku 2017.

Lékárny společnosti Pronatal využívají lékárenský systém Mediox, který poskytuje analytickou nádstavbu. Webová nádstavba umožňuje připojení odkudkoliv a následně generování jednoduchých reportů a prohlížení provozních dat.

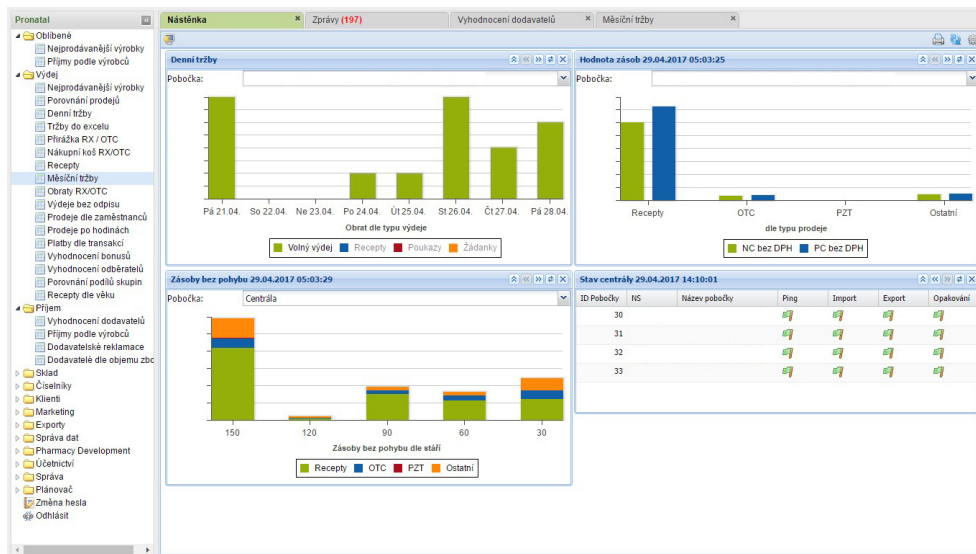
Aplikace se skládá z nástěnky (dashboardu), která zobrazuje nadefinované základní grafické přehledy a KPI. U přehledů je umožněné základní filtrování zobrazovaných dat. Ukázka uživatelského prostředí je vidět na obrázku 5.1.

Aplikace dále poskytuje v levém menu přehled všech dostupných datových oblastí, nad kterými je poté pomocí formuláře spustit nadefinovaný filtr. Výsledkem je tabulkový report.

#### Výhody:

- Aplikace je plně přizpůsobená business doméně lékárny,
- aplikace umožňuje přístup ke všem datům z provozních databází,
- k aplikaci je možné díky zvolené platformě přistupovat odkudkoliv,
- aplikace udržuje vytvořené reporty ve formě záložek (tabů),
- aplikace umožňuje export dat ve formátu pro xls.





Obrázek 5.1: Uživatelské prostředí WebManageru – nástěnka.

### Nevýhody:

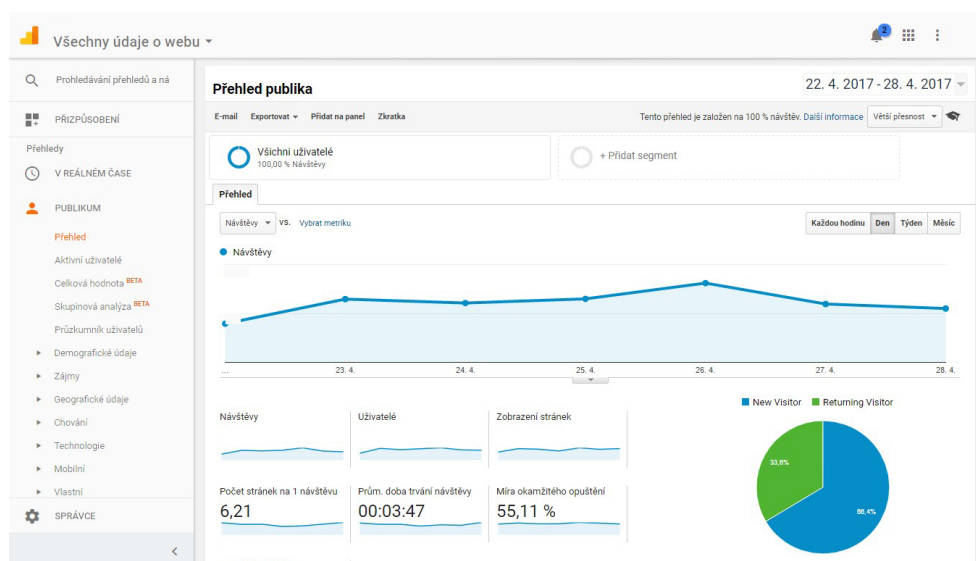
- Aplikace neumožňuje měnit parametry filtru interaktivně, je vždy třeba vytvořit nový report,
- aplikace je poměrně pomalá i při zpracování menšího objemu dat,
- aplikace neumožňuje porovnávat mezi sebou data z různých časových období,
- aplikace neumožňuje zobrazit data graficky,
- aplikace neumožňuje porovnání dat z různých zdrojů.
- nástěnku (dashboard) není možné přizpůsobit

### 5.1.2 Google Analytics

Nejznámější volně dostupné řešení v oblasti datové analýzy jsou bezesporu Google Analytics<sup>34</sup>. Nejedná se sice o nástroj pro analýzu prodejů jako takových, ale obecné zpracování a způsob pohledu na data bude velmi podobný ve všech datových doménách. Navíc rozšíření pro internetové obchody umožňuje vyhodnocení dat i na úrovni prodejů.

<sup>34</sup><https://analytics.google.com>

## 5. NÁVRH WEBOVÉ APLIKACE PRO REPORTOVÁNÍ



Obrázek 5.2: Uživatelské prostředí Google analytics – před.

Uživatelské prostředí je vidět na obrázku 5.2. Jak je u produktů společnosti Google zvykem, uživatelské prostředí je minimalistické a zaměřené převážně na funkční prvky.

Stejně jako u předešlého produktu 5.1.1 je zde hlavní menu na levé straně. S pomocí hlavního menu je možné přistupovat k různým pohledům na data. Je to způsob, jakým rozdělit rozsáhlou datovou základnu pro koncové uživatele do menších souvisejících částí.

Zobrazení dat v Google Analytics kombinuje tabulkovou i grafickou reprezentaci. Rozhraní poskytuje všechny nejpoužívanější druhy grafů. Tabulková reprezentace na druhé straně využívá funkčního datagridu se spoustou možností filtrace a řazení dat.

Aplikace umožňuje vytvářet libovolné reporty a měnit je v uživatelském prostředí i v průběhu analýzy. Navíc je umožněno reporty ukládat v rámci aplikace a exportovat v různých formátech.

Google Analytics umožňují vytvářet různé sestavy a porovnávat je mezi sebou. Pro porovnání výkonu metrik v průběhu času je zde možné zvolit dvě časová období.

### Výhody:

- K aplikaci je možné díky zvolené platformě přistupovat odkudkoliv,
- aplikace umožňuje export dat v různých formátech, včetně xls,
- aplikace umožňuje měnit nastavení filtrů interaktivně i po zobrazení dat,

- aplikace umožňuje porovnání dat ze dvou časových období,
- aplikace umožňuje porovnání dat z více zdrojů,
- aplikace umožňuje data zobrazit tabulkově, ale i graficky,
- uživatelské prostředí a zobrazovaná data jsou přizpůsobitelná pro daného uživatele,
- aplikace umožňuje nahrát vlastní dataset.

### **Nevýhody:**

- Aplikace není plně přizpůsobená business doméně lékárny,
- aplikace je dostupná pouze jako služba,
- aplikaci není možné jakkoli modifikovat.

### **5.1.3 Výstupy**

Zmíněné aplikace nelze úplně porovnávat, protože vznikly za jiných okolností pro úplně jiný segment trhu. Na základě zjištěných dat a otestování obou uvedených řešení byly ale zjištěny cenné poznatky a best practises v oblasti datové analýzy.

Pro vznikající aplikaci budou proto důležité možnosti filtrování a přístupu k datům, které reprezentuje aplikace 5.1.1, která reflektuje business doménu lékárenství.

Z analýzy aplikace 5.1.2 vyplývají potřeby na interaktivní práci z daty, vizualizaci výstupu pomocí grafů, uživatelské přizpůsobení prostředí a v neposlední řadě možnosti porovnávání různých zdrojů v průběhu času.

## **5.2 Vymezení požadavků na aplikaci**

Požadavky na aplikaci jsou soubor funkčních i nefunkčních požadavků, které vznikly sjednocením požadavků managementu společnosti a řešerši známých řešení v oblasti datové analýzy.

V rámci analýzy byly stanoveny tyto požadavky:

### **5.2.1 Požadavky na produkční prostředí**

1. Aplikace pracuje s prodejními daty společnosti ze systému FaRMIS,
2. aplikace využívá aktualizované číselníky z CDB,
3. serverové části aplikace a její data jsou uloženy na vnitřní síti společnosti,
4. klientská část aplikace je dostupná online,

## 5. NÁVRH WEBOVÉ APLIKACE PRO REPORTOVÁNÍ

---

5. klientská část aplikace nevyžaduje žádné speciální programové vybavení,
6. klientská část aplikace je dostupná pouze pro uživatele s uživatelským účtem,
7. uživatelský účet do aplikace může vytvořit administrátor aplikace,
8. s aplikací může pracovat více uživatelů současně (řádově jednotky uživatelů).

### 5.2.2 Požadavky na funkcionalitu aplikace

1. Aplikace umožňuje uživateli nastavit pravidelné zasílání vybraných reportů emailem,
2. uživatel má možnost exportovat data ve formátu pro MS Excel,
3. aplikace umožňuje procházet a analyzovat dostupná data (OLAP dotazování nad datovým skladem)
  - a) uživatel může zvolit segment dat, se kterým bude pracovat,
    - i. segment dat může být vytvořený kombinací libovolných parametrů, které jsou dostupné
    - ii. v aplikaci jsou dostupné přednastavené segmenty dat
    - iii. připravený segment dat je možné uložit pro příští použití
    - iv. uživatel může zvolit více segmentů dat pro porovnání
    - v. každý nový segment dat bude odlišený jinou barvou
  - b) uživatel může zvolit časový rozsah dat, se kterými bude pracovat,
    - i. časový rozsah je určený počátečním a koncovým datem
    - ii. časový rozsah je možné zadat ručně, nebo pomocí výběrových časových úseků
    - iii. výběrové časové úseky jsou den, týden, měsíc, čtvrtletí, rok
    - iv. uživatel může zvolit druhý časový rozsah (jiný, než první) pro porovnání
  - c) uživatel může zvolit libovolný počet metrik, které chce zobrazit
    - i. základní metriky jsou definovány v externím souboru a jsou načteny při spuštění aplikace
    - ii. metriky odpovídají sloupcům
  - d) uživatel může zvolit dimenzi, v rámci které budou seskupeny vybrané metriky
    - i. uživatel může zvolit hlavní dimenzi
    - ii. uživatel může zvolit sekundární dimenzi, podle té se poté rozpadnou hlavní dimenze a metriky

- iii. dimenze odpovídají řádkům
- iv. v případě více dimenzí je řádek s názvem položky hlavní dimenze použitý jako součtový řádek
- e) uživatel může volit různé typy zobrazení dat
  - i. uživatel může zvolit zobrazení dat jako tabulky
  - ii. uživatel může zvolit zobrazení dat jako grafu
- f) uživatel může zvolit způsob řazení dat při tabulkovém zobrazení
  - i. uživatel může zvolit řazení řádků v tabulce dle zvolené metriky sestupně, nebo vzestupně
- g) uživatel může omezit počet zobrazených záznamů a zobrazit stránkování
- h) uživatel má možnost uložit nadefinovaný report do knihovny Moje reporty

## 5.3 Modelování případů užití (Use cases)

S pomocí jazyka UML byl připravený model případů užití, který pomáhá popsat interakci jednotlivých uživatelů se systémem a zároveň umožňuje zachytit pokrytí jednotlivých požadavků.

Hlavní diagram případů užití je znázorněný na obrázku 5.3. Jednotlivé případy jsou potom popsány v samostatných tabulkách.

### UC0: EditUsers

Případ užití UC0: EditUsers je dostupný pouze uživateli s administrátorskými právy. Hlavní část případu užití je popsána v tabulce 5.1. Pokud jsou splněny vstupní podmínky, je uživateli zobrazen přehled uživatelů. Následně je možné pokračovat jedním z dalších dostupných případů užití.

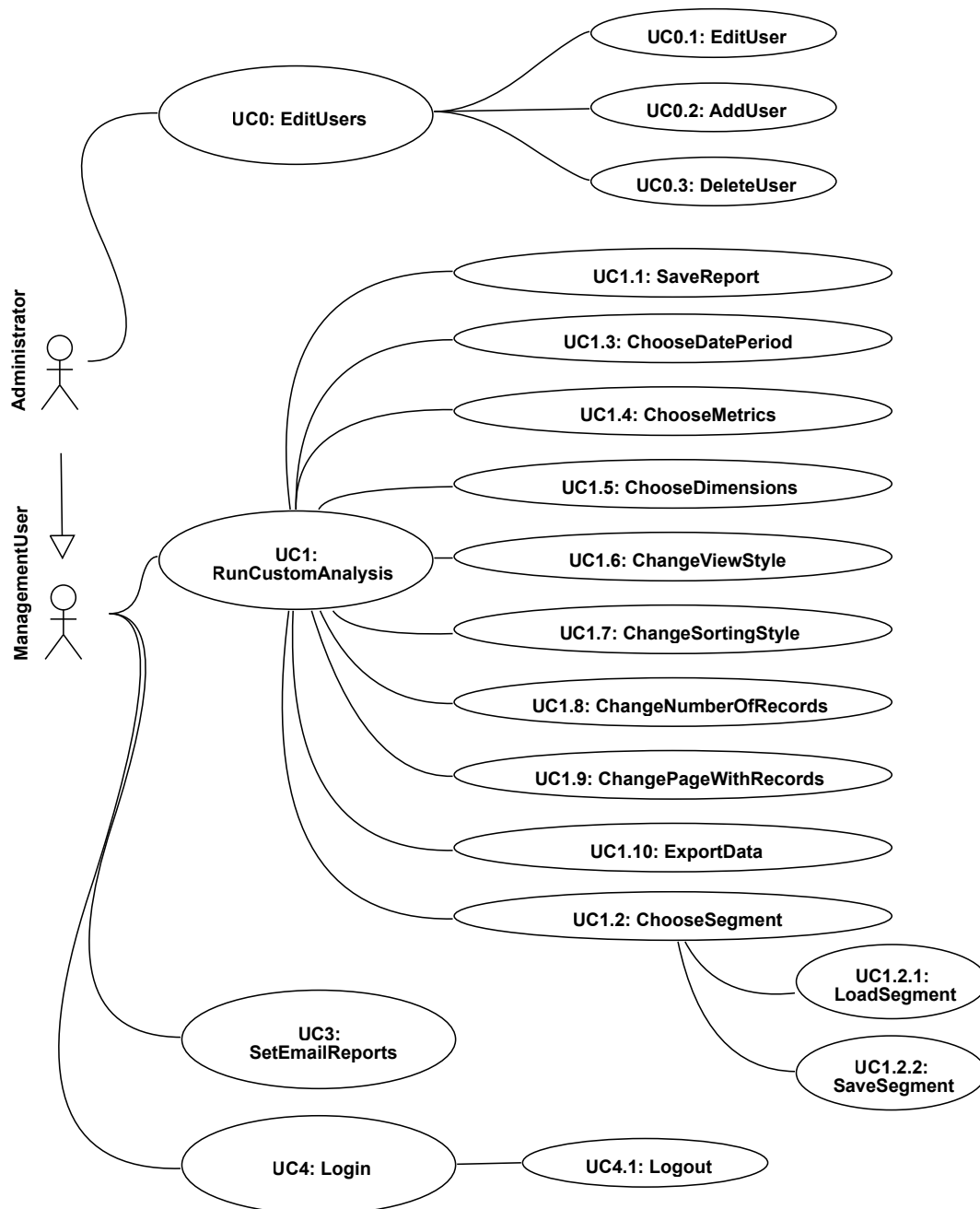
Dostupné jsou UC0.1: EditUser, pro editaci jednoho konkrétního uživatele, UC0.2: AddUser pro přidání nového uživatele do systému a UC0.3: DeleteUser pro smazání uživatele.

### UC0.1: EditUser

V rámci případu užití UC0.1: EditUser je umožněno měnit údaje všech uživatelů v systému, nastavovat jejich práva a preference. Podrobně je daný případ popsán v tabulce 5.2.

### UC0.2: AddUser

Přidání uživatele UC0.2: AddUser má podobný průběh, jako editace již existujícího uživatele s tím rozdílem, že po kliknutí na tlačítko pro přidání uživatele



Obrázek 5.3: Diagram případů užití webové aplikace.

Tabulka 5.1: UC0: EditUsers

| <b>Případ užití, UC0: EditUsers</b> |   |
|-------------------------------------|---|
| Vstupní podmínky                    | Uživatel je administrátor   |
| Výstupní podmínky                   | Je zobrazený seznam uživatelů   |
| Krok                                | Akce  |
| 1.                                  | Aplikace ověří, že je uživatel přihlášený. Pokud není, pokračuje UC4: Login. Pokud je přihlášený, zobrazí seznam všech uživatelů. |
| 2.                                  | Uživateli je zobrazen list všech uživatelů.   |
| 3.                                  | Uživatel může pokračovat UC0.1: EditUser, UC0.2: AddUser, nebo UC0.3: DeleteUser.   |

Tabulka 5.2: UC0.1: EditUser

| <b>Případ užití, UC0.1: EditUser</b> |   |
|--------------------------------------|---|
| Vstupní podmínky                     | Uživatel je administrátor   |
| Výstupní podmínky                    | Byly uloženy úpravy vybraného uživatele   |
| Krok                                 | Akce  |
| 1.                                   | Aplikace ověří, že je uživatel přihlášený. Pokud není, pokračuje UC4: Login. Pokud je přihlášený, zobrazí seznam všech uživatelů. |
| 2.                                   | Uživateli je zobrazen profil vybraného uživatele.   |
| 3.                                   | Uživatel může měnit všechny dostupné údaje.   |
| 4.                                   | Uživatel může uložit změny a pokračuje na 1.krok tohoto případu užití, nebo může změny neuložit a pokračuje na UC0:EditUsers.     |

na stránce s přehledem uživatelů, je otevřený prázdný profil. Profil nového uživatele je možné uložit až po vyplnění všech povinných polí.

### UC0.3: DeleteUser

Smazání uživatele dle případu UC0.3: DeleteUser je možné s kterýmkoliv uživatelem, vyjma uživatele samotného. Tím, že je tato funkce přístupná pouze uživateli v roli administrátor je vyloučeno zablokování systému smazáním všech uživatelů.

V případě, že má být smazán uživatel, který má uložené nějaké vlastní nastavení v rámci systému (například nastavení reportování apod.), které by mohlo být hodnotné pro další uživatele, je administrátorovi umožněno toto nastavení přenést na jiného uživatele.

### UC4: Login a UC4.1: Logout

Případ užití dostupný všem uživatelům je UC4: Login, který je popsán v tabulce 5.3 a umožňuje uživateli přístup do systému. Na tento případ je navázaný

## 5. NÁVRH WEBOVÉ APLIKACE PRO REPORTOVÁNÍ

---

případ UC4.1: Logout, který naopak umožní uživateli opustit prostředí webové aplikace a odhlásit se.

Případy užití UC:0 EditUsers, UC:4 Login a na ně navázané případy naplňují v rámci kategorie 5.2.1 požadavky 6, 7 a 8.

Tabulka 5.3: UC4: Login

| <b>Případ užití, UC4: Login</b> |   |
|---------------------------------|---|
| Výstupní podmínky               | Uživatel je přihlášený a je mu zobrazen Dashboard   |
| Krok                            | Akce  |
| 1.                              | Uživateli je zobrazena přihlašovací obrazovka aplikace.   |
| 2.                              | Uživatel zadá přihlašovací údaje a klikne na tlačítko pro přihlášení do aplikace.   |
| 3.                              | Pokud jsou přihlašovací údaje správné, uživatel je přihlášený a je mu zobrazen Dashboard. Pokud nebyly přihlašovací údaje správné, je na to uživatel upozorněn a je mu umožněno si zaslat zapomenuté heslo emailem. |

### UC1: RunCustomAnalysis

Hlavní účel aplikace, reportování a datová analýza, je pokrytý sjednocujícím případem užití UC1: RunCustomAnalysis, který je definovaný tabulkou 5.4.

Z analýzy 5.1.2 byla využita myšlenka rozdělení datového zdroje na několik skupin sdružujících související data. Proto by aplikace měla obsahovat základní přehledové reporty, které budou sloužit jako vstupní brána pro další interaktivní dotazování. U těchto přednastavených výběrů dat půjde hlavně o volbu metrik a dimenzí, které jsou budou pro daný report v základu nastavené.

Další upřesnění, nebo případné změny reportů, je možné realizovat s pomocí případů užití UC1.x.

Tabulka 5.4: UC1: RunCustomAnalysis

| <b>Případ užití, UC1: RunCustomAnalysis</b> |   |
|---|---|
| Vstupní podmínky                            | Uživatel je přihlášený  |
| Výstupní podmínky                           | Uživateli je vygenerován report   |
| Krok  | Akce  |
| 1.  | Uživatel si v hlavním menu v části reporty zvolí kategorii reportů                          |
| 2.  | Systém načte data pro vybraný report  |
| 3.  | Uživateli je zobrazena přehledová stránka s reportem a vybranými daty.                      |
| 4.  | Uživatel může na přehledové stránce pokračovat libovolným navazujícím případem užití UC1.x. |



**UC1.1: SaveReport**

Případ užití pro uložení reportu UC1.1: SaveReport je popsán v tabulce 5.5. Tato funkce by měla uživatelům umožnit uložit reporty definované pomocí případů užití UC1.2, UC1.4, UC1.5 a UC1.6. Následně je možné uložené reporty najít v kategorii Moje reporty a kdykoliv si je zobrazit bez nutnosti znovu nastavovat filtrování dat.

Případ užití UC1.1: SaveReport naplňuje v rámci bodu 3 kategorie 5.2.2 požadavek h).

Tabulka 5.5: UC1.1: SaveReport

| <b>Případ užití, UC1.1: SaveReport</b> |   |
|--|---|
| Vstupní podmínky                       | Uživatel je přihlášený a nachází se na stránce reportu  |
| Výstupní podmínky                      | V levém menu se v kategorii Moje reporty objeví nový uložený report                                   |
| Krok                                   | Akce  |
| 1.                                     | Uživatel klikne na tlačítko uložit report.  |
| 2.                                     | Systém vyzve uživatele, aby zadal název uloženého reportu.  |
| 3.                                     | Uživatel zadá název reportu a potvrdí tlačítkem Uložit.   |
| 4.                                     | Nový report je uložený do systému pro další použití a uživatel zůstane na obrazovce s tímto reportem. |

**UC1.2: ChooseSegment**

Při základním nastavení aplikace jsou do reportů načítána data z celé datové základny. Někdy může být ale užitečné načíst omezená vstupní data. Příkladem může být zobrazení dat pouze pro jeden prvek zvolené dimenze, nebo i kombinace více podobných omezení a výběrů.

Takové výběry v rámci aplikace jsou reprezentovány segmenty. Výběr segmentu je popsán případem užití UC1.2: ChooseSegment, podrobněji definovaným tabulkou 5.6.

V aplikaci je umožněna volba více různých segmentů a následně porovnání jejich metrik. Různé segmenty jsou barevně odlišeny. Segmenty je možné ukládat, případ užití UC1.2.2: SaveSegment, a později zase načíst, případ užití UC1.2.1: LoadSegment.

Případ užití UC1.2: ChooseSegment naplňuje v rámci bodu 3 kategorie 5.2.2 požadavky a).

**UC1.3: ChooseDatePeriod**

Podobným způsobem jako segmenty v případě užití 5.3 je umožněno nastavení až dvou časových období pro daný report. Případ užití UC1.3: ChooseDatePeriod popisující volbu časového období je definován v tabulce 5.7

## 5. NÁVRH WEBOVÉ APLIKACE PRO REPORTOVÁNÍ

Tabulka 5.6: UC1.2: ChooseSegment

| <b>Případ užití, UC1.2: ChooseSegment</b> |   |
|---|---|
| Vstupní podmínky                          | Uživatel je přihlášený a nachází se na stránce reportu  |
| Výstupní podmínky                         | Uživateli je zobrazena obrazovka s vybraným reportem pro daný segment   |
| Krok                                      | Akce  |
| 1.  | Uživatel zvolí v části pro výběr segmentu možnost pro nový segment.   |
| 2.  | Systém zobrazí uživateli průvodce nastavením segmentu.  |
| 3.  | Uživatel pomocí filtrů definuje příslušný segment vstupních dat pro daný report, nebo zvolí jeden z předdefinovaných, nebo uložených segmentů. Kliknutím na tlačítko se uloží nastavení segmentu. |
| 4.  | Systém načte data pro definované segmenty a zobrazí příslušný report.   |
| 5.  | Uživatel má možnost smazat jeden z definovaných segmentů, nebo definovat další segment. Ve druhém případě pokračuje tento případ od bodu 1.   |

Případ užití UC1.3: ChooseDatePeriod naplňuje v rámci bodu 3 kategorie 5.2.2 požadavky b).

Tabulka 5.7: UC1.3: ChooseDatePeriod

| <b>Případ užití, UC1.3: ChooseDatePeriod</b> |  |
|--|--|
| Vstupní podmínky                             | Uživatel je přihlášený a nachází se na stránce reportu   |
| Výstupní podmínky                            | Uživateli jsou zobrazena přepočítaná data z vybraných časových období.   |
| Krok   | Akce   |
| 1.   | Uživatel otevře panel pro výběr časové dimenze.  |
| 2.   | Uživatel zvolí časové období. Může využít zadání počátečního a koncového bodu ručně, nebo výběrem v kalendáři.                                   |
| 3.   | Uživatel má možnost zakliknout, že má zájem přidaat druhé časové období pro porovnání s prvním, v takovém případě zopakuje bod 2 tohoto případu. |
| 4.   | Uživatel potvrdí vybrané časové rozsahy.   |
| 5.   | Systém uživateli zobrazí report pro vybrané časové období.   |

### UC1.4: ChooseMetrics

Hlavní předmět analýz jsou samotné metriky. V rámci předdefinovaných reportů je vždy zvolena skupina metrik, které spolu souvisí a představují nejobvyklejší ekonomické ukazatele. Občas ale může vyvstat požadavek na zobrazení některých dalších metrik, nebo metriky seřadit v jiném pořadí apod.

Pro výběr metrik k zobrazení a pro jejich následné seřazení bude sloužit samostatné výběrové pole. Případ užití UC1.4: ChooseMetrics, který tento proces definuje je podrobněji popsán tabulkou 5.8.

Případ užití UC1.4: ChooseMetrics naplňuje v rámci bodu 3 kategorie 5.2.2 požadavky c).

Tabulka 5.8: UC1.4: ChooseMetrics

| <b>Případ užití, UC1.4: ChooseMetrics</b> |   |
|---|---|
| Vstupní podmínky                          | Uživatel je přihlášený a nachází se na stránce reportu                  |
| Výstupní podmínky                         | Uživateli je zobrazen report s vybranými metrikami                      |
| Krok                                      | Akce  |
| 1.  | Uživatel pomocí výběrového pole nastaví požadované metricky k zobrazení |
| 2.  | Systém připraví a zobrazí nově definovaný report                        |

### UC1.5: ChooseDimensions

Pohled na data v datové kostce je realizován pomocí volby vhodných dimenzí. Pro základní reportování je vždy zvolena hlavní dimenze. Jednotlivé dimenze je možné použít pro filtrování dat v rámci definice segmentů 5.3.

Pokud ale uživatel požaduje rozpad agregovaných dat dle jednotlivých členů zvolené dimenze, je třeba tuto dimenzi nastavit jako hlavní, případně jako sekundární pro daný report. Jakým způsobem to provést popisuje případ užití UC1.5: ChooseDimensions, popsán tabulkou 5.9.

Případ užití UC1.5: ChooseDimensions naplňuje v rámci bodu 3 kategorie 5.2.2 požadavky d).

Tabulka 5.9: UC1.5: ChooseDimensions

| <b>Případ užití, UC1.5: ChooseDimensions</b> |  |
|--|--|
| Vstupní podmínky                             | Uživatel je přihlášený a nachází se na stránce reportu                         |
| Výstupní podmínky                            | Uživateli je zobrazen report rozpadený dle zvolených dimenzí                   |
| Krok   | Akce   |
| 1.   | Uživatel může změnit hlavní dimenzi, nebo přidat sekundární dimenzi zobrazení. |
| 2.   | Systém přepočítá data a zobrazí je nově hierarchicky dle zvolených dimenzí.    |
| 3.   | Uživatel může odebrat sekundární dimenzi zobrazení.                            |

### UC1.6: ChangeVisualStyle

Pro lepší orientaci ve velkém množství dat a pro jejich snazší analýzu je umožněno zobrazit data jak formou tabulky, tak vizuálně formou několika typů grafů.

Popis případu užití UC1.6: ChangeVisualStyle, který definuje proces změny zobrazení je shrnutý v tabulce 5.10.

Případ užití UC1.6: ChangeVisualStyle naplňuje v rámci bodu 3 kategorie 5.2.2 požadavky e).

Tabulka 5.10: UC1.6: ChangeVisualStyle

| <b>Případ užití, UC1.6: ChangeVisualStyle</b> |   |
|---|---|
| Vstupní podmínky                              | Uživatel je přihlášený a nachází se na stránce reportu  |
| Výstupní podmínky                             | Uživateli je zobrazen report formou zvoleného typu reprezentace   |
| Krok  | Akce  |
| 1.  | Uživatel má možnost aktivovat vizuální zobrazení dat formou grafu.  |
| 2.  | Systém vybraná tabulková data zobrazí zvoleným grafickým zobrazením.  |
| 3.  | Uživatel má možnost vypnout grafické zobrazení dat.   |
| 4.  | Uživatel má možnost volit mezi různými variantami grafické reprezentace dat. Jakmile je provedena volba, případ pokračuje krokem 2. |

### **UC1.7: ChangeSortingStyle, UC1.8: ChangeNumberOfRecords a UC1.9: ChangePageWithRecords**

Předpokladem je zobrazení tabulkové části reportu pomocí vhodného datagridu. Součástí této komponenty by měla být možnost řazení dle jednotlivých sloupců v obou směrech. Tento proces odpovídá případu užití UC1.7: ChangeSortingStyle, který naplňuje v rámci bodu 3 kategorie 5.2.2 požadavek f).

Uživatelské rozhraní by v rámci zobrazení reportu v tabulkovém módu mělo umožňovat stránkovat zobrazené výsledky. Vhodné je také umožnit uživateli zvolit počet záznamů na jednu stránku. Volba počtu záznamů na jednu stránku odpovídá případu užití UC1.9: ChangePageWithRecords. Změna aktuálně zobrazené stránky je reprezentována případem užití UC1.8: ChangeNumberOfRecords. Dohromady oba případy užití naplňují v rámci bodu 3 kategorie 5.2.2 požadavek g).

### **UC1.10: ExportData**

Aplikace by měla umět export zobrazovaných reportů v různých formátech. Základní předpoklad je formát pro Microsoft Excel a rozumná varianta exportu dat jako pdf.

Součástí exportu dat by mělo být základní nastavení pro specifikaci formátu exportovaných dat. Proces exportu UC1.10: ExportData je popsán tabulkou 5.11.

Případ užití UC1.10: ExportData naplňuje v rámci bodu 3 kategorie 5.2.2 požadavky g).

Tabulka 5.11: UC1.10: ExportData

| <b>UC1.10: ExportData</b> |   |
|---------------------------|---|
| Vstupní podmínky          | Uživatel je přihlášený a nachází se na stránce reportu  |
| Výstupní podmínky         | Uživateli je nabídnuto uložení souboru s exportem reportu   |
| Krok                      | Akce  |
| 1.                        | Uživatel zvolí na stránce reportu možnost exportovat.   |
| 2.                        | Systém uživateli nabídne možnosti pro nastavení exportu. Jedná se o rozsah dat, typ souboru apod. |
| 3.                        | Uživatel nastaví požadovaný formát exportu a potvrdí výběr tlačítkem.                             |
| 4.                        | Systém pro uživatele připraví soubor s exportem.  |

## 5.4 Návrh UI

Při návrhu uživatelského rozhraní je kladený důraz na minimalistický vzhled a maximální pozornost je věnována funkčnosti. Hlavní prvek uživatelského rozhraní aplikace jsou samotná data, která jsou zobrazena buď formou tabulky, nebo případně vizuálně různými variantami grafů.

V návaznosti na zobrazená data je potřeba přidat sadu ovládacích prvků, pomocí kterých bude možné data procházet, filtrovat a porovnávat.

Součástí aplikace budou dvě menu:

- Uživatelské menu – slouží aktuálně přihlášenému uživateli pro individuální nastavení a pro odhlášení,
- Hlavní menu – přizpůsobené pro orientaci v rámci aplikace. Obsahuje odkaz na dashboard, na stránky jednotlivých reportů atd.

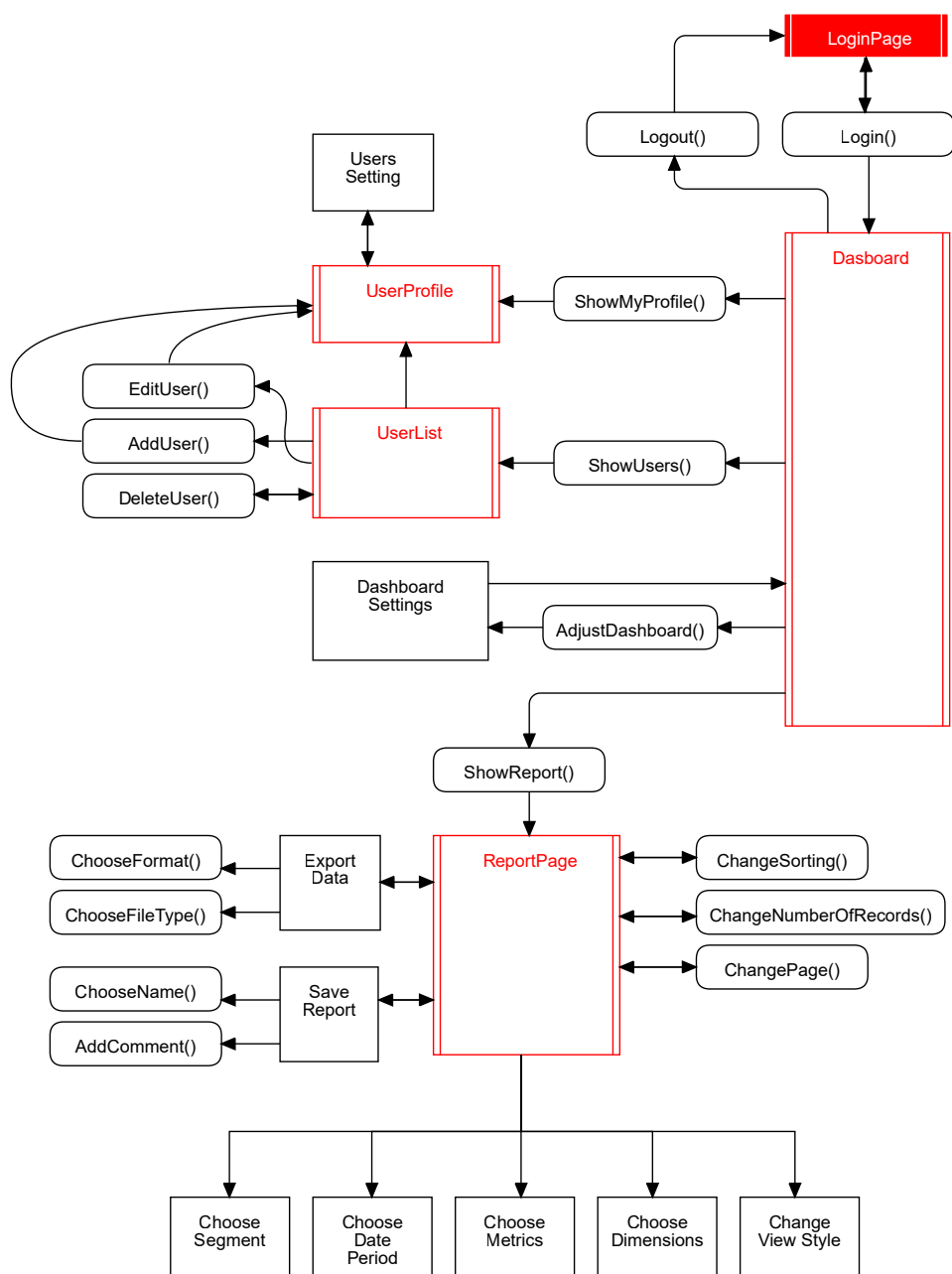
### 5.4.1 Modelování funkcí uživatelského rozhraní

Pro znázornění funkcionalit uživatelského rozhraní napříč aplikací bylo připraveno několik následujících grafů, popisujících možné průchody aplikací a v rámci jednotlivých obrazovek shrnujících dostupné funkce.

#### 5.4.1.1 Úvodní obrazovka a nastavení uživatelů

Přihlašovací stránka `LoginPage` je počáteční obrazovka průchodu aplikací, jejíž popis začíná task grafem 5.4. Při správném zadání hesla je uživateli zobrazen `Dashboard`. `Dashboard` nabízí základní přehledy tak, jak si je uživatel nastavil. Pomocí funkce `AdjustDashboard()` může uživatel tato nastavení měnit.

## 5. NÁVRH WEBOVÉ APLIKACE PRO REPORTOVÁNÍ



Obrázek 5.4: Graf popisující hlavní kostru aplikace.

Přihlášený uživatel má dále možnost měnit svá osobní nastavení pomocí funkce `ShowMyProfile()`. Běžnému uživateli je umožněno měnit své jméno, příjmení, email a heslo.

Pokud má uživatel administrátorská práva, může si zobrazit přehled všech uživatelů v systému a měnit jejich osobní nastavení, případně jim přidělit administrátorská práva.

Administrátor může dále vytvářet nové uživatele a mazat ty stávající.

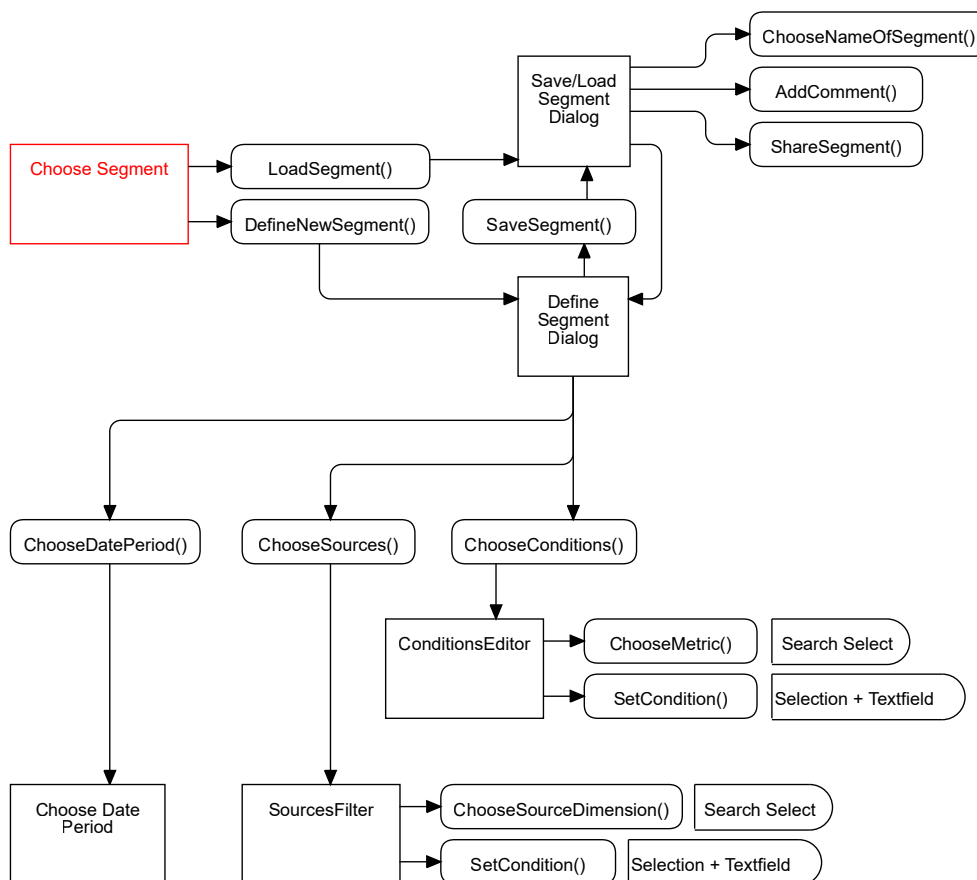
Uživatel může z obrazovky dashboardu pokračovat funkcí `ShowReport()` na libovolný report z nabídky. V nabídce jsou systémové předpřipravené a také vlastní v minulosti uložené reporty.

#### 5.4.1.2 Reporty

Jakmile uživatel zvolí v hlavním menu některý z dostupných reportů, je mu zobrazena obrazovka `ReportPage`, která je sestavena z tabulky, nebo grafu zobrazujícího nadefinovaná data a z ovládacích prvků.

V rámci obrazovky `ReportPage` má uživatel několik dostupných funkcionalit:

- Může upravit řazení tabulkového reportu funkcí `ChangeSorting()`,
- může změnit počet záznamů zobrazených na jednu stránku pomocí funkce `ChangeNumberOfRecords()`,
- má možnost přepínat s využitím `ChangePage()` mezi jednotlivými stránkami se záznamy,
- je mu dostupná možnost exportu dat `ExportData`, v rámci které může upřesnit formát exportu funkcí `ChooseFormat()` a typ souboru s využitím metody `ChooseFileType()`,
- je umožněno uložit aktuální report funkcionalitou `SaveReport`, v rámci které může zvolit jeho název `ChooseName()` a doplnit ho komentářem `AddComment()`,
- aktuální report je možné upravit nastavením segmentů – sada metod `ChooseSegment`, která je dále popsána task grafem 5.5,
- report je možné dále přizpůsobit s pomocí časové dimenze – sada metod `ChooseDatePeriod`, popsána task grafem 5.8,
- vstupní metriky reportu je možné měnit sadou metod `ChooseMetrics`, definovaných task grafem 5.6,
- pohled na data v reportu může být přizpůsoben nastavením dimenzí – sada metod `ChooseDimensions`, shrnutá task grafem 5.7,
- uživatel také může změnit styl zobrazení dat – metoda `ChangeVisualStyle`, které se věnuje task graf 5.9.



Obrázek 5.5: Graf popisující část aplikace pro specifikaci segmentu.

### ChooseSegment

První a nejkompaktnější sada metod obrazovky `ReportPage` se týká volby segmentu `ChooseSegment`. Samotný průchod tímto nastavením je posaný task grafem 5.5.

Volba segmentu odpovídá filtrování dat vstupujících do reportu. Při volbě segmentu má uživatel možnost zvolit z předdefinovaných, nebo dříve uložených segmentů pomocí metody `LoadSegment()`, nebo může nadefinovat úplně nový segment metodou `DefineNewSegment()`.

Při definici vlastního segmentu, nebo v případě upravení existujícího segmentu je v obou případech umožněno takový segment uložit pro budoucí použití metodou `SaveSegment()`. Součástí procesu je zvolení jména metodou `ChooseNameOfSegment()`, přidání komentáře s popisem daného segmentu `AddComment()` a také varianta zaškrtnout možnost sdílení daného nastavení



s ostatními uživateli v systému – sdílené segmenty. Celý proces je zapouzdřený v rámci `Save/LoadSegmentDialog`.

Definování nového, nebo doplnění načteného segmentu je sdruženo do modulu `DefineSegmentDialog`, který poskytuje možnosti filtraci vstupních dat dle času (`ChooseDatePeriod()` – metoda je popsána případem 5.4.1.2), zdroje dat, neboli na úrovni dimenzí (`ChooseSources()`) a nebo nastavením omezení v podobě podmínky na samotné metriky (`ChooseConditions()`).

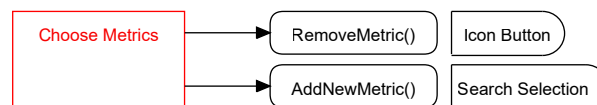
Při omezení vstupních dat na úrovni dimenzí je možné v rámci sady metod `SourcesFilter` zvolit zdrojové dimenze `ChooseSourceDimension()` s pomocí `Search Selectu` a následně nastavit podmínky, které musí prvky (members) dané dimenze splňovat `SetCondition()`. Pro nastavení podmínek využívá uživatelské prostředí elementy `selection` a `textfield`.

Při omezení vstupních dat na úrovni metrik `ConditionsEditor` jsou nejprve vybrány metriky `ChooseMetric()` s pomocí `Search Selectu` a následně zvolena podmínka `SetCondition()`. Podmínka je nastavena u vybraných metrik s pomocí volby logických operátorů v `selectu` a vyplněním parametru v `textfieldu`.

### ChooseMetrics

Všechny reporty mají v základu nastavenou minimálně jednu základní metriku. Uživateli je na obrazovce reportu umožněno přidat libovolné množství nových metrik funkcí `AddNewMetric()`, která je realizována elementem `Search selection`.

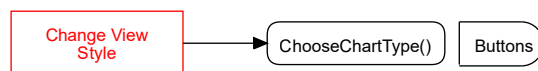
Uživatel může v rámci sady `ChooseMetrics` také libovolně odstranit dříve přidané metriky pomocí metody `RemoveMetric()`, která je realizovaná tlačítkem s ikonou.



Obrázek 5.6: Graf popisující část aplikace pro výběr metrik.

### ChooseDimensions

Každý zobrazený report má nastavenou minimálně hlavní dimenzi. Dimenze se nastavuje metodou `ChooseSecondDimension()` s využitím elementu `Search select`. Uživatel může změnit tuto hlavní dimenzi a nebo přidat sekundární dimenzi metodou `ChooseMainDimension()`.



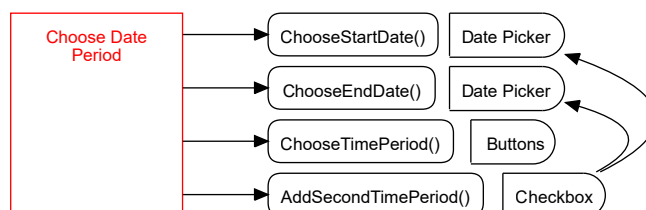
Obrázek 5.7: Graf popisující část aplikace pro výběr dimenzí.

### ChooseDatePeriod

Nastavení časové dimenze je odděleno do samostatného prvku uživatelského prostředí. Zdrojový časový úsek je definovaný počátečním datem (nastaví se metodou `ChooseStartDate()` v datepickeru) a koncovým datem (metoda `ChooseEndDate()` s využitím datepickeru).

Výběr časového období je možné provést pomocí rychlé volby přednastavených časových úseků metodou `ChooseTimePeriod()`. Uživatel má na výběr tlačítka s hodnotami rok, čtvrtletí a měsíc.

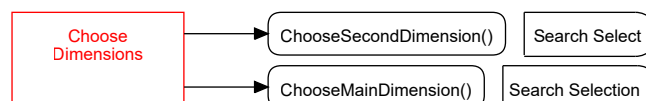
V případě, že chce uživatel porovnat dvě časová období, může s pomocí checkboxu využít metodu `AddSecondTimePeriod()`.



Obrázek 5.8: Graf popisující část aplikace pro výběr časové dimenze.

### ChangeVisualStyle

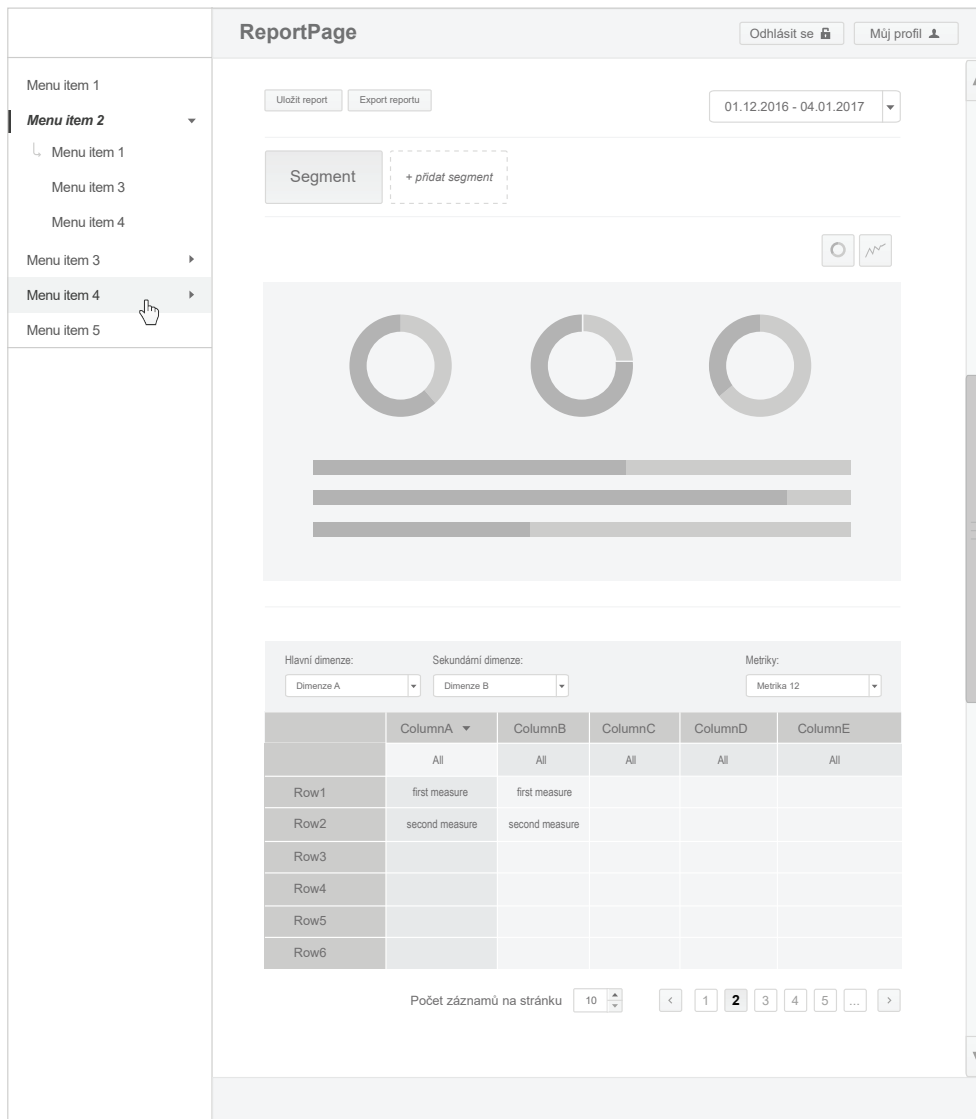
Uživateli je umožněno měnit zobrazení dat metodou `ChooseChartType()`. Pro výběr základních typů vizualizací slouží v rámci uživatelského prostředí tlačítka s ikonami.



Obrázek 5.9: Graf popisující část aplikace pro změnu zobrazení dat.

### 5.4.2 Wireframy uživatelského rozhraní

Na základě návrhu případů užití a následně průchodů aplikací byly připraveny wireframe modely uživatelského prostředí. Pro vytvoření wireframů byly použity sady UX elementů Webpage wireframe stencil<sup>35</sup> a Shades<sup>36</sup>.



Obrázek 5.10: Wireframe model pro stránku s reportem.

Byla zvolena varianta layoutu se dvěma sloupci s hlavním menu po levé

<sup>35</sup><http://www.pixeden.com/web-wireframing/webpage-wireframe-stencil>

<sup>36</sup><http://www.designforfounders.com/freebie/shades/>

straně, jak je vidět na obrázku 5.10. V pravé horní části layoutu se nachází uživatelské menu. Uživatelské prostředí zachovává maximální jednoduchost a soustředí se na funkčnost.

Wireframe 5.10 reprezentuje stránku s nastaveným reportem popsanou v části 5.4.1.2. V rámci wireframu byly zapracovány všechny funkcionality popsané task grafem 5.4.

Další wireframy, včetně 5.10 v plné velikosti, jsou doplněny jako elektronická příloha této práce.

- reportPage.pdf – obsahuje variantu reportu 5.10 v plném zobrazení,
- dashboard.pdf – obsahuje wireframe pro část 5.4.1.1,
- userList.pdf – wireframe realizuje modul pro správu uživatelů,
- userProfile.pdf – reprezentuje sekci aplikace pro správu konkrétního uživatele,
- chooseSegment.pdf – wireframe pro část aplikace 5.4.1.2, v rámci které se nastavují segmenty.

## 5.5 Návrh architektury

Cílem projektu pro podporu Business Intelligence řešení ve společnosti Medicon Pharm s.r.o. a dalších společnostech skupiny Medicon a.s. je reportovací aplikace, která bude sloužit jako analytická nádstavba nad datovým skladem.

Ze seznamu požadavků na produkční prostředí 5.2.1 vyplývá, že se má jednat o online dostupnou aplikaci, a že data mají zůstat uložena v rámci interní sítě společnosti.

Aplikace se tedy bude skládat z webové části, která bude připojena na serverové rozhraní k datové základně – datovému tržišti Bowbeen DMart.

### 5.5.1 Komponenty aplikace

Pro projekt byla připravena architektura zobrazená na obrázku 5.11, která je založená na principech MVC<sup>37</sup>.

Aplikace bude rozdělena na dvě technologicky oddělené části:

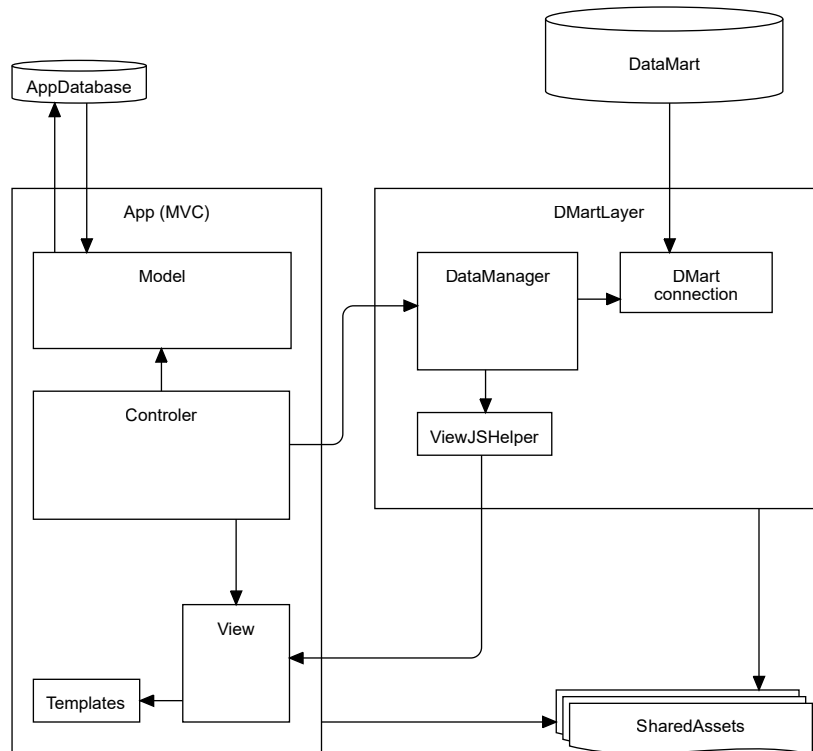
- První část vychází z interní databáze 5.5.2.2 a reprezentuje vnitřní aplikační procesy, uživatelská nastavení a kooperaci uvnitř aplikace.
- Druhá část aplikace bude zaměřená na zpracování MDX dotazů směrem k datovému skladu, získání a zpracování obdržených dat, jejich zobrazení a manipulaci s nimi.

---

<sup>37</sup>Model, view, controller

Pro všechny základní funkcionality jako je pohyb uživatele v rámci aplikace, vykreslení a přizpůsobení uživatelského prostředí, správa uživatelů a správa všech interně ukládaných dat je v aplikaci dostupná vrstva **Modelu**.

**Model** obstarává veškerou interní aplikační logiku na pokyny **Controlleru**, který zároveň přímo komunikuje s částí **View**, pro interakci s uživatelem.



Obrázek 5.11: Model architektury aplikace.

V případě, že bude aplikace vyžadovat práci s daty z datového skladu, předává **Controller** řízení entitě `verbDataManager` v druhé části aplikace, která se stará o zapouzdření a odeslání dotazu na rozhraní datového skladu.

**DataManager** se stará o kompletní komunikaci s datovým skladem, včetně přijmutí odpovědi a rozparsování přijatých dat.

Jakmile jsou data připravená pro další zpracování a zobrazení uživateli, předá **DataManager** řízení pomocné **ViewJSHep** třídě, která je vhodnou formou předá přímo hlavnímu aplikačnímu **View** pro zobrazení uživateli.

### 5.5.2 Přístup k datům

Jak už bylo popsáno výše, aplikace bude pracovat se dvěma oddělenými zdroji dat. Jeden datový zdroj obsahuje interní data, uživatelská nastavení, uložené

reporty atd. Tento datový zdroj bude uložený na stejném serveru, jako samotná aplikace.

Druhý datový zdroj je datové tržiště Bowbeen DMart, které je uloženo ve vnitřní síti společnosti a pro komunikaci s tímto uzlem bude muset aplikace implementovat nějaký druh rozhraní.

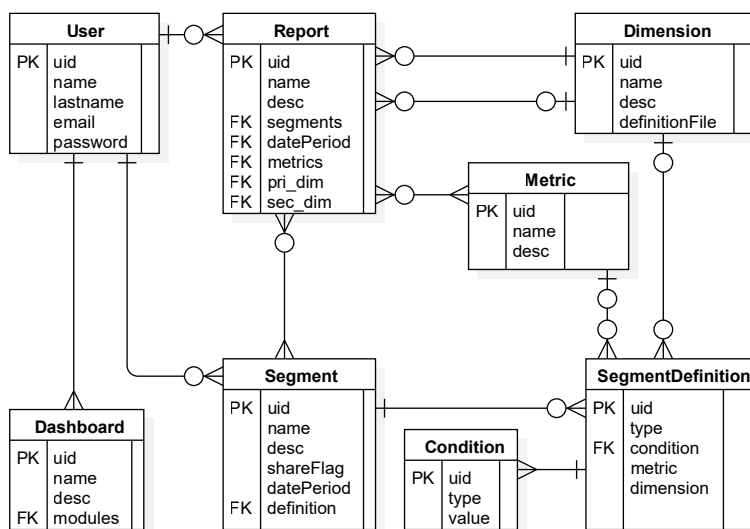
### 5.5.2.1 Externí analytická data

Kompletní informace o nasazení a architektuře datového skladu byly popsány v kapitole 4.

Standardem pro komunikaci s datovými sklady založenými na řešení od společnosti Microsoft je využití Analytické služby databázového serveru.

V tomto případě je vytvořeno rozhraní s pomocí webového serveru Windows IIS<sup>38</sup>. Takové rozhraní představuje `http/https` endpoint pro komunikaci s analytickou službou, kdy je díky použití nativních technologií možné využívat všechny benefity se službou spojené – například Windows ověření uživatelů ve stejné doméně apod.

Aplikace se připojí k webovému serveru a s pomocí jazyka MDX2.5 může odesílat dotazy do datového skladu.<sup>39</sup>



Obrázek 5.12: Základní datový model interní aplikační databáze.

<sup>38</sup>Internet Information Services

<sup>39</sup>Podrobněji popsáno v kapitole 6.1

### 5.5.2.2 Interní aplikační data

Interní databáze bude obsahovat pouze data uživatelů a nastavení základních a uložených reportů, případně segmentů. Základní datový model interní aplikační databáze je na obrázku 5.12.

V modelu je reflektován požadavek na sdílení segmentů mezi uživateli. Zároveň už je dopředu počítáno s možností nabídnout uživatelům více dashboardů, mezi kterými by potom mohli přepínat.

Pro uchování podrobných informací o dimenzích slouží pouze odkaz na externí soubor, který bude obsahovat katalog datové kostky. Konkrétní data k dané dimenzi budou dohledány pomocí parametru name.

## 5.6 Technologie

Na základě navržené architektury aplikace 5.5.1 a s přihlédnutím k její charakteristice – konkrétně velké množství práce nad datovými tabulkami – bylo rozhodnuto oddělit technologicky jádro aplikace od části pro zpracování dat z datového skladu.

### 5.6.1 Jádro aplikace a Nette framework

První část aplikace – jádro – bude postavena na php frameworku Nette. Tato vrstva bude mít na starosti hlavní sestavení aplikace, uživatelská nastavení a inicializaci dalších nástrojů.

Volba Nette frameworku je vhodná vzhledem k široké lokální podpoře. Framework je relativně jednoduchý a nabízí všechny potřebné funkcionality. Se zvoleným frameworkem jsou navíc při tvorbě jiných firemních aplikací velké zkušenosti.

### 5.6.2 Zpracování a zobrazení dat – Javascript

Druhá část aplikace by se měla starat o zpracování dotazů na datový sklad a následně parsování přijatých dat.

Tato vrstva bude postavena na Javascriptu a případných jeho nádstavbách. Už v tuto chvíli je jasné, že součástí finálního řešení budou knihovny:

#### **XMLA4js**

Knihovna, která poskytuje plnohodnotnou podporu pro komunikaci s OLAP servery pomocí standardů XMLA zapouzdřujícího jazyk MDX.<sup>40</sup>

<sup>40</sup><https://github.com/rpbouman/xmla4js>

### **jsGrid**

Komponenta zapouzdřující data do jednoduchých, ale výkonných datagridů. <sup>41</sup>

### **Chartists.js**

Knihovna nabízející širokou škálu modulů pro vizualizaci dat. <sup>42</sup>

### **5.6.3 Interní MySQL, zdrojový SQL Server**

Pro interní datové uložení 5.5.2.2 bude naprosto dostačující řešení s využitím MySQL.

Externí databáze (datový sklad) je podrobně popsán v kapitole 4, nicméně pro aplikaci bude tato entita odstíněna komunikačním rozhraním Analytické služby.

---

<sup>41</sup><http://js-grid.com/>

<sup>42</sup><https://gionkunz.github.io/chartist-js/>



---

# Prototyp webové aplikace pro reportování

Jako výstup první fáze projektu byl připraven prototyp aplikace, který má ověřit realizovatelnost projektu na úrovni databázových a síťových technologií a také prezentovat managementu společnosti, co mohou od datového skladu a dalších přidružených aplikací očekávat.

Přípravě prototypu předcházela příprava změn datového skladu, která se ve výsledku ukázala náročnější, jelikož bylo třeba datovou základnu od základů přepracovat. Tyto změny jsou popsány v kapitole 4.

Souběžně s transformací datového skladu probíhal sběr požadavků na reportovací aplikaci, analýza možných řešení a příprava potřebné dokumentace pro popis její architektury, grafického prostředí apod. Tato část je popsána v kapitole 5.

Pro vytvoření prototypu bylo třeba připravit rozhraní na straně datového skladu, nastavit potřebné moduly pro komunikaci a najít vhodnou komponentu pro uživatelské rozhraní.

## 6.1 Rozhraní datového skladu

Od databáze postavené na multidimenzionálním modelu po reportování nad těmito daty je zde několik mezikroků. Prvním z nich je nastavení připojení k databázi a vytvoření tzv. datového zdroje. Druhým krokem je sestavení datové kostky s použitím metrik a dimenzí z dimenzionálního modelu zdrojové databáze. Posledním krokem je nastavení připojení ke službě SSAS. Jakým způsobem a jaké připojení je třeba nastavit záleží na jeho účelu.

### 6.1.1 Vytvoření datové kostky

Pro vytvoření datového zdroje a následné sestavení datové kostky je možné využít doplněk SSDT<sup>43</sup> pro Visual studio.

Konkrétně byly využity tyto verze:

- Visual Studio Community 2015 <sup>44</sup>,
- SQL Server Data Tools 2015 <sup>45</sup>.

Předpokladem pro vytvoření datové kostky je rovněž instalace těchto produktů:<sup>46</sup>

- MS SQL Server 2016 ve verzi Standard,
- modul MS SQL Server Analysis Services pro MS SQL Server

S pomocí SSDT byl ve Visual studiu připravený projekt pro SSAS. V rámci projektu byl vytvořen datový zdroj. Součástí přípravy datového zdroje je nastavení připojení ke zdrojové databázi, včetně metody autentifikace.

Jakmile je korektně nastavený datový zdroj (Data Source), je možné z něj sestavit datovou kostku. Mezikrokem je vytvoření tzv. Data Source View, které umožňuje upravit datový zdroj přidáním logických klíčů, cachováním metadat, nebo doplněním pomocných výpočtů apod.

Nejdůležitější částí při vytváření pohledu je nastavení vazeb – toho využijeme pro identifikování vazby mezi tabulkou faktů a dimenzionálními tabulkami.

Na vytvořeném pohledu je možné sestavit datovou kostku. Při vytváření zvolíme zdrojovou tabulku s metrikami (tabulku faktů) a doplníme tabulkami dimenzí.

Po vytvoření datové kostky je možné dále upravovat dimenze a vytvářet hierarchie.

Na závěr stačí kostku nasadit (deploy) a následně zprocesovat (process), čímž se kostka stane dostupná pro další analýzu.

### 6.1.2 HTTP/HTTPS rozhraní s pomocí IIS

SSAS umožňuje různé metody připojení k datové kostce, ale všechny využívají TCP jako síťový protokol a XML for Analysis (XMLA) jako komunikační protokol.

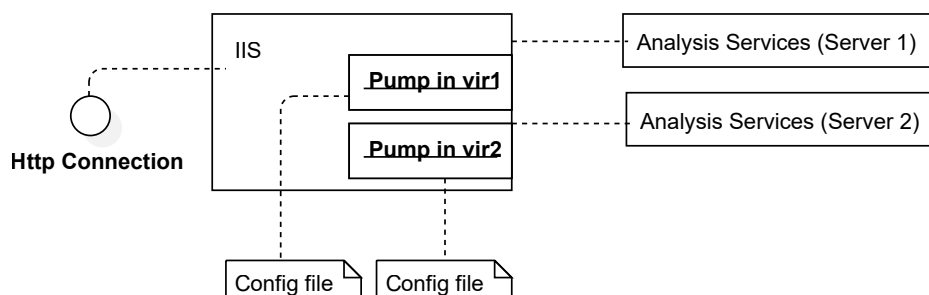
---

<sup>43</sup>SQL Server Data Tools

<sup>44</sup>V průběhu přípravy této práce byla vydána verze 2017. Ke stažení na <https://www.visualstudio.com/cs/downloads/>. Pro tuto verzi zatím bohužel není dostupný doplněk SQL Server Data Tools.

<sup>45</sup>Ke stažení na <https://msdn.microsoft.com/cs-cz/mt186501>

<sup>46</sup>viz 4.5.1



Obrázek 6.1: Vytvoření datové pumpy v aplikačním rozhraní IIS.

Obvyklé aplikace pracují s připojením z aplikace MS Excel, nebo nástavby Power Pivot<sup>47</sup>. Zároveň je možné se k analytické službě připojit z SSMS, což je vhodné pro vývoj a testování. Všechny zmíněné možnosti počítají s připojením v rámci jedné síťové domény.

Pro potřeby připojení z webové aplikace mimo síťovou doménu bude třeba nastavit možnost http/https připojení k SSAS. Pro tyto potřeby využijeme MS IIS.<sup>48</sup> [25].

Webový server IIS nám umožní vytvořit `http/https` endpoint pro připojení k instanci SSAS. K nastavení je třeba nakonfigurovat `MSMDPUMP.dll`<sup>49</sup> a využít modul ISAPI<sup>50</sup>

Pro každý `http/https` endpoint je třeba vytvořit samostatný adresář a samostatnou aplikaci v IIS, jak je znázorněno na obrázku 6.1. K samotnému připojení využívá `MSMDPUMP.dll` poskytovatele pro Analysis Services OLE DB s pomocí TCP/IP. Klientské požadavky mohou pocházet z různých zdrojů i mimo důvěryhodnou doménu. SSAS a IIS se musí ale nacházet v totožné doméně.

Pro samotné nastavení je třeba z adresáře `OLAP/bin/isapi` příslušného MS SQL Serveru zkopírovat adresářovou strukturu 6.2 a umístit ji do adresáře pro budoucí aplikaci rozhraní v rámci IIS v tomto umístění `<drive>:/inetpub/wwwroot/nazev – aplikace/`.

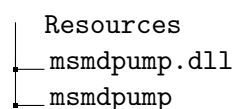
Další krok je vytvoření aplikačního poolu přímo v IIS. Pokud byly správně

<sup>47</sup>Popis doplňku na <https://support.office.com/en-us/article/Power-Pivot-Add-in-a9c2c6e2-cc49-4976-a7d7-40896795d045>

<sup>48</sup>Internet Information Services je softwarový webový server od společnosti Microsoft. Aplikaci je možné rozšířit širokou škálou doplňkových modulů. Více informací na oficiálním webu <https://www.iis.net>

<sup>49</sup>MSMDPUMP je rozšíření ISAPI, které umožňuje přeměrovat http/https dotazy z aplikačního rozhraní přímo na rozhraní SSAS serveru, a to s využitím různých metod autentifikace.

<sup>50</sup>Internet Server Application Program Interface – více informací na oficiálním webu [https://msdn.microsoft.com/en-us/library/ms525172\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/ms525172(v=vs.90).aspx).



Obrázek 6.2: Soubory modulu MSMDPUMP.dll

nakopírované potřebné soubory a vytvořený aplikační pool, objeví se v rozhraní IIS složka s názvem aplikace, který byl zvolený jako název složky. Poté stačí z dané složky vytvořit aplikaci výběrem příkazů v kontextovém menu pod pravým kliknutím myši.

Součástí vytvoření aplikace je možnost nastavení autentifikační metody. Pro testování je vhodné využít anonymní autentifikaci, která využívá práva uživatele, který nastavuje danou službu.

V případě nasazení do produkčního prostředí je nejlepší variantou Windows autentifikace, která ale bude dostupná pouze v případě, že se klientská stanice nachází ve stejné doméně.

Pro aplikaci připravovanou v rámci této práce je dostupná kompromisní možnost jednoduché (basic) autentifikace, která umožňuje nastavit přihlašovací údaje a vyžadovat je po uživateli vždy při přihlášení ke službě. V takovém případě se využívá tzv. connection string, v rámci kterého jsou posílány přihlašovací údaje jako prostý text – proto je vyžadováno nastavení připojení s použitím zabezpečeného https.

Na závěr je třeba nastavit mapování požadavků na SSAS přímo na nastavenou MSMDPUMP.dll. V případě, že je na daném serveru dostupno více instancí MS SQL Serveru, je třeba v konfiguračním souboru `msmdpump.ini` nastavit správný server.

Pokud bylo vše v pořádku nastaveno, je rozhraní dostupné na adrese `https://nazev-serveru:8080/nazev-aplikace/msmdpump.dll`. Výsledná situace je znázorněna na obrázku 6.3

## 6.2 Uživatelské prostředí

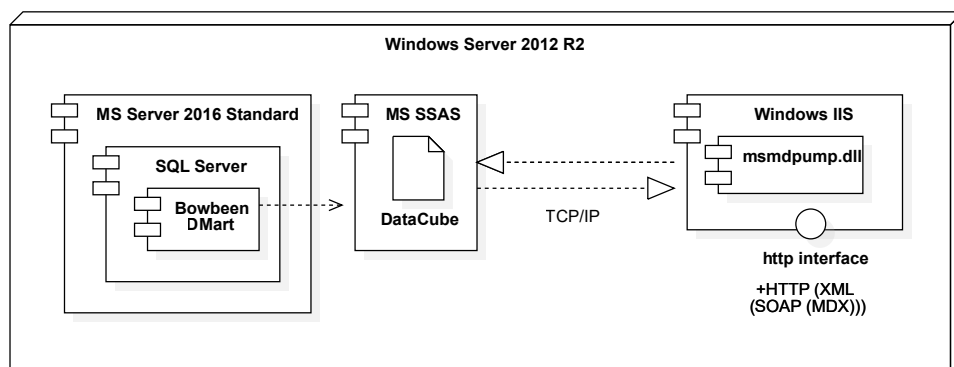
### 6.2.1 Grafické zpracování

Pro realizaci uživatelského prostředí byla použita grafická šablona třetí strany. Konkrétně byla zvolena šablona společnosti Creative Tim<sup>51</sup>. V rámci potřeb sestavení prototypu bylo dostačující volně dostupné demo šablony Light Bootstrap Dashboard. Pro následný další vývoj bude zakoupena plná varianta.

Šablona využívá pokročilé frontendové technologie:

---

<sup>51</sup><https://www.creative-tim.com>



Obrázek 6.3: Rozhraní datového tržiště pro MDX dotazy přes http připojení.

- samozřejmostí je zpracování v HTML5<sup>52</sup> s využitím frameworku Bootstrap.<sup>53</sup>
- Pro správu kaskádových stylů je využitý framework SASS.<sup>54</sup>
- Pro zpracování a zobrazení grafů je využita knihovna Chartist.<sup>55</sup>

### 6.2.2 Šablonovací systém

K oživení grafických podkladů byl vybrán jednoduchý šablonovací systém Foil<sup>56</sup>. Volba tohoto šablonovacího nástroje je postavena převážně na jeho jednoduchosti, což umožnilo snížit náklady na přípravu prototypu. Sytaxe Foilu nevyžaduje zavádění žádných nových prvků a je kompletně postavena na syntaxi php. Instalace Foilu je možná přímočaře s pomocí Composeru. V aktuální verzi funguje Foil i s verzí PHP 5.4 a vyššími.

O řízení šablon se ve Foilu stará hlavní **Engine**, který na začátku převezme jako parametr umístění složky s šablonami. Následně stačí s pomocí příkazu `render()` šablonu vykreslit. Jako druhý parametr je navíc možné do šablony předat libovolná data.

```
echo $engine->render('nazev-sablony', $nejaka-data);
```

<sup>52</sup><https://www.w3.org/TR/html5/>

<sup>53</sup><http://getbootstrap.com/>

<sup>54</sup><http://sass-lang.com/>

<sup>55</sup><https://gionkunz.github.io/chartist-js/>

<sup>56</sup><http://www.foilphp.it>

V rámci frameworku je možné vytvářet aliasy příkazů pro rychlejší vývoj – to může ale ve výsledku kód učinit velmi nepřehledný a nepochopitelný pro ostatní vývojáře.

Co je užitečnější, je možnost práce s adresářovou strukturou. V rámci inicializace `Enginu` je možné nastavit více zdrojových složek, což umožňuje přehlednější organizaci rozsáhlejších aplikací. Adresářovou strukturu je navíc možné měnit i za běhu programu.

Základní stavební kámen šablon je ve Foilu soubor šablony. Takový soubor je jako každý jiný php soubor, což umožňuje využívat veškeré dostupné prostředky, které nabízí php. V rámci šablonového souboru je pod vyhrazenou proměnou `$this` zastoupena hlavní třída šablonovacího systému.

Jednotlivé šablony je mezi sebou možné navzájem zanořovat, což umožňuje jednodušší správu kódu a dělení jednotlivých částí webu do částí (partials). Část se dá později do nadřazené šablony vložit pomocí příkazu `insert()`

```
<?= $this->insert('partials/sidebar') ?>
```

Mezi nejsilnější nástroje Foilu patří dědičnost a využití sekcí v rámci šablon. Tato technika umožňuje vytvářet abstraktní šablony, které definují základní strukturu a následně pomocí šablon potomků vytvářet konkrétní layouty. S pomocí sekcí (sections) je navíc možné vykreslovat pouze některé části takto definovaných šablon a docílit tak poměrně vyváženého řešení.

Pro podrobnější popis všech funkcí a možností šablonovacího frameworku Foil je dostupná online oficiální dokumentace.[26]

### 6.3 Datová vrstva

Prototyp sám o sobě nevyužívá žádnou databázi, jelikož v první části není třeba řešit uživatelské účty, ani ukládat žádná data nastavení, historie, logování apod. Je ale třeba připravit sadu nástrojů, které umožní připojení k připravenému rozhraní datového skladu.

Datová vrstva se stará o připojení k rozhraní, přípravu, zapouzdření a odeslání analytických dotazů a následné zpracování příchozích dat a předání těchto dat do GUI.

Hlavní komponentou datové vrstvy je knihovna `XMLA4js`, která se stará o zapouzdření MDX dotazů a jejich odeslání s pomocí SOAP webové služby. Pro účely komunikace s SSAS byl připravený standard `XMLA`<sup>57</sup>, který definuje dvě metody:

- **Discover** slouží pro získání metadat o datové entitě,

---

<sup>57</sup>XML for Analysis

- **Execute** slouží pro vykonání příkazu, který může být buď MDX, DMX, nebo SQL. Zároveň umožňuje jako druhý parametr doplnit omezení pro daný příkaz – timeout, název katalogu apod. Výsledek dotazu může být multidimenzionální dataset, nebo tabular rowset. [27]

V prostředí prototypu bylo sestaveno několik ukázkových dotazů/reportů, které jsou dostupné uživateli. Vybrané reporty jsou popsány v části 6.5.

## 6.4 Workflow a integrace komponent

Prototyp aplikace je postavený na kombinaci PHP, šablonovacího systému Foil, Javascriptových knihoven a funkcí pro správu dat, komunikaci s datovým skladem a manipulaci s uživatelským prostředím.

Pomocí PHP a šablonovacího systému bylo z šablony vytvořeno několik modulů uživatelského rozhraní.

Uživatelské rozhraní bylo rozloženo do:

- **Složky partials**, do které patří hlavička stránky (`head.php`), navigační lišta (`navbar.php`), hlavní menu (`sidebar.php`), patička stránky (`footer.php`) a hlavní panel (`main-panel.php`).
- **Složku content**, která obsahuje layouty pro jednotlivé reporty.

Řídící prvek prototypu je modul `DataLayer`, který na základě zvoleného reportu vytvoří příslušný MDX dotaz a s pomocí knihovny `XMLA4js` ho odešle na rozhraní serveru. Rozvržení komponent je znázorněno na obrázku 6.4.

Spouštěcí mechanismus je načtení stránky reportu `windows.onload()`, které aktivuje inicializační funkci.

Inicializační funkce dle zvoleného reportu s pomocí metody `mdxFactory()` vytvoří MDX dotaz a s předá ho funkci `Xmla.executeMDX()` pro získání dat z datového skladu.

V případě získání dat je v rámci success callbacku funkce `executeMDX()` předáno řízení a získaná data view funkcí `renderData()`.

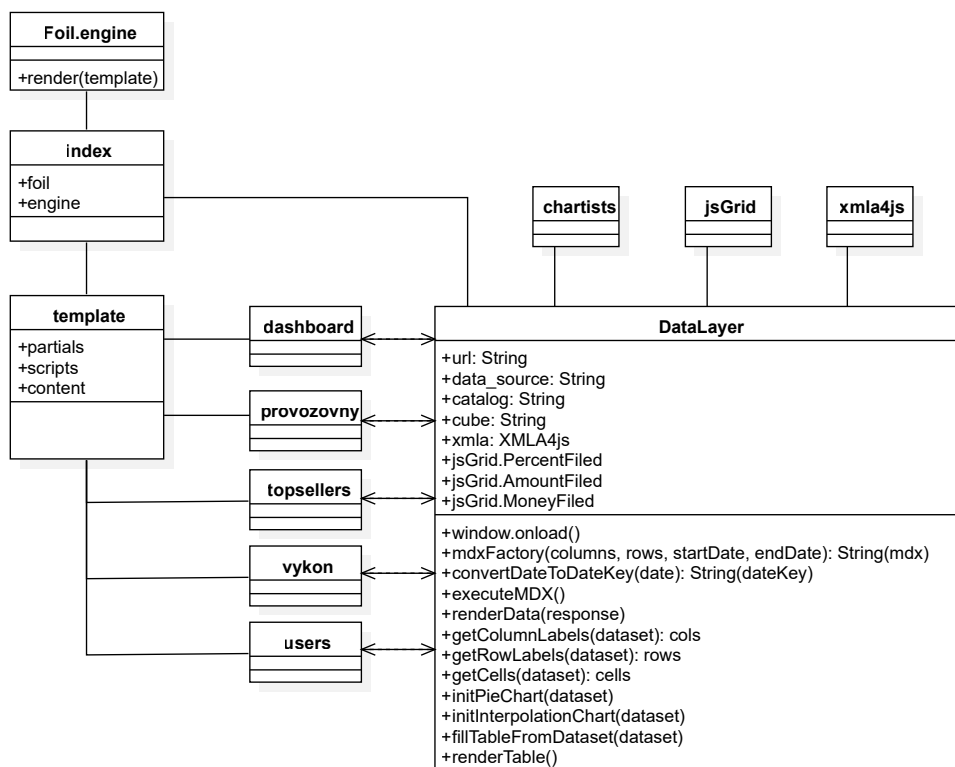
Funkce `renderData()` se stará o vykreslení dat do daného layoutu. Pro vykreslení jsou použity knihovny `jsGrid` pro tabulkový mód a `Chartist` pro vizualizace a grafy.

Aktuálně jsou podporovány dva typy grafů realizované renderovacími funkcemi `initPieChart()` a `initInterpolationChart()`.

Pro zobrazení dat v datagridu bylo vytvořeno několik vlastních formátů pole:

- `jsGrid.AmountFiled` pro zobrazení počtu kusů zboží s jednotkou ks,
- `jsGrid.MoneyFiled` pro finanční metriky s jednotkou Kč

## 6. PROTOTYP WEBOVÉ APLIKACE PRO REPORTOVÁNÍ

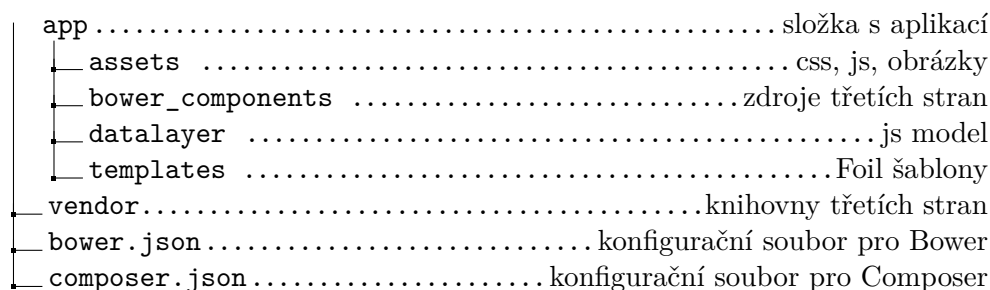


Obrázek 6.4: Přehled komponent prototypu

- `jsGrid.PercentFiled` pro zobrazení symbolu % u procentuálních hodnot.

Součástí modelu je několik parsovacích funkcí `getColumnLabels(dataset)`, `getRowLabels(dataset)` a `getCells(dataset)`, které pomáhají získat z odpovědi serveru metadata a příslušné metriky.

Přehled bloků, ze kterých je sestavený prototyp jsou na schématu 6.4.



Obrázek 6.5: Přehled modulů prototypu aplikace pro reportování



## 6.5. Připravené reporty

The screenshot shows a web application interface for 'BOWBEEN DATAMART'. The main content area is titled 'Reporty/Provozovny' and displays a report titled 'Přehled výkonu jednotlivých provozoven'. The report includes a table with columns for 'MAINDIMENSION', 'MNOZSTVI', 'NAKUPNI CENA BEZ DPH', 'PRODEJNI CENA BEZ DPH', and 'ZISK'. The table lists seven different store locations with their respective metrics. Above the table, there are filters for 'Minulý měsíc', 'Aktuální měsíc', and 'Tento rok'. On the left side, there is a navigation menu with options like 'DASHBOARD', 'TOP SELLERY', 'PROVOZOVNY', 'VÝKON SPOLEČNOSTI', and 'UŽIVATELSKÝ PROFIL'.

| MAINDIMENSION             | MNOZSTVI  | NAKUPNI CENA BEZ DPH | PRODEJNI CENA BEZ DPH | ZISK        |
|---------------------------|-----------|----------------------|-----------------------|-------------|
| E-Shop MTcz               | 325.00 Ks | 255348.60 Kč         | 342292.00 Kč          | 86943.90 Kč |
| E-Shop NVcz               | 286.00 Ks | 207939.90 Kč         | 278255.80 Kč          | 70315.90 Kč |
| Pharmacentrum Budějovická | 297.00 Ks | 229339.20 Kč         | 309164.50 Kč          | 79825.40 Kč |
| Pharmacentrum DBK         | 286.00 Ks | 219641.50 Kč         | 296110.00 Kč          | 76467.70 Kč |
| Pharmacentrum Vysočany    | 359.00 Ks | 260466.70 Kč         | 352079.90 Kč          | 91612.40 Kč |
| Pharmacentrum Zelený pruh | 278.00 Ks | 215204.00 Kč         | 289658.30 Kč          | 74454.00 Kč |

Obrázek 6.6: Bowbeen DMart, Obrazovka s reportem výsledků po provozovnách.

## 6.5 Připravené reporty

Pro účely prezentace možností Business Intelligence a reportování nad datovým skladem bylo v rámci prototypu realizováno několik ukázkových reportů:

- Porovnání výkonu provozoven na dashboardu, včetně grafu,
- report top sellerů,
- report výsledků po provozovnách (minulý měsíc),
- report výsledků po provozovnách (aktuální měsíc, popis v části 6.5.1),
- report výsledků po provozovnách (aktuální rok),
- report výkonu společnosti dle měsíců, včetně grafu (ukázka v části 6.5.2).

### 6.5.1 Report výsledků po provozovnách

První vybraný report je přehled výsledků jednotlivých provozoven za vybrané období.

V následujícím MDX dotazu je vidět první část **SELECT**, v rámci které jsou vybrány metriky Množství, Nákupní cena bez DPH, Prodejní cena bez DPH a Zisk a jsou přiřazeny na osu **COLUMNS**.

**SELECT**

```
{  
  [Measures].[Mnozstvi], [Measures].[Nakupni Cena Bez Dph],  
  [Measures].[Prodejni Cena Bez Dph], [Measures].[Zisk]
```

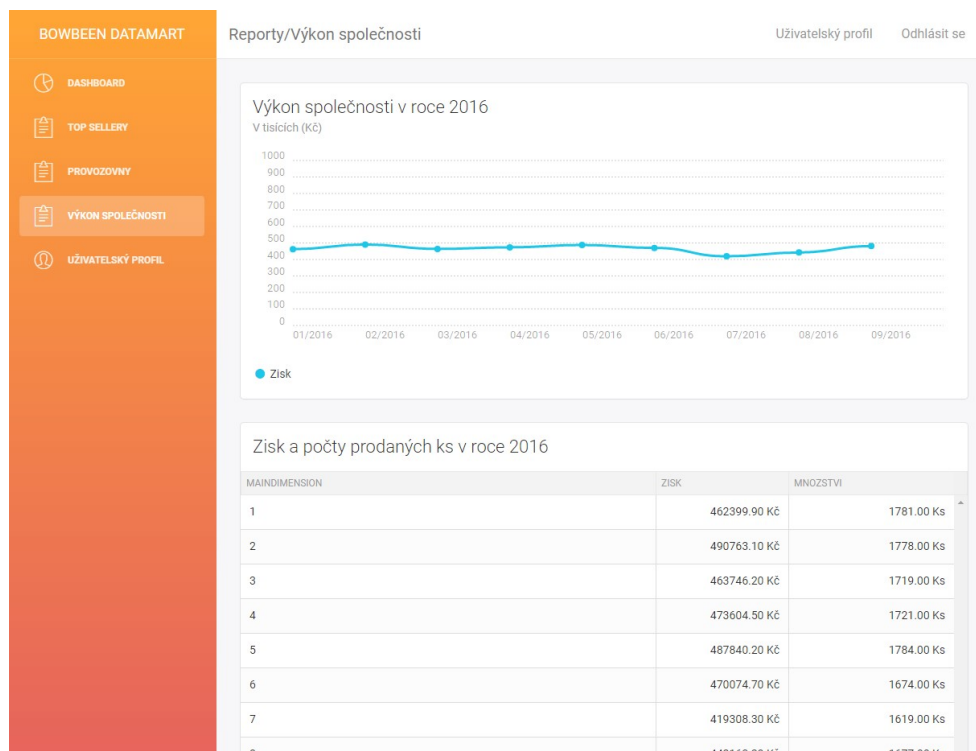
## 6. PROTOTYP WEBOVÉ APLIKACE PRO REPORTOVÁNÍ

```
}  
ON COLUMNS,  
NON EMPTY  
    [Dim Provozovna].[Nazev Provozovny].[Nazev Provozovny]  
ON ROWS  
FROM (  
SELECT [Dim Datum].[Date Key].&[pocatecni_datum] :  
    [Dim Datum].[Date Key].&[koncove_datum]  
ON COLUMNS FROM BowbeenSalesCube ;
```

V druhé části jsou eliminována prázdná pole klauzulí `NON EMPTY` u zdrojové dimenze `Dim Provozovna`, která je umístěna na osu `ROWS`.

Poslední část dotazu upřesňuje časové období, které omezuje zdrojová data.

Výsledek takového dotazu v prostředí prototypu je zachycený na obrázku 6.6.



Obrázek 6.7: Bowbeen DMart, obrazovka s reportem výkonu společnosti, včetně grafu.

### 6.5.2 Report výkonu společnosti

Další ukázkový report se zaměřuje na reprezentaci výkonu společnosti. Je zvoleno jak tabulkové, tak grafické zobrazení dat. Výsledek reportu je vidět na obrázku 6.7.

S pomocí MDX dotazu byla získána surová data. Nejprve je potřeba načíst všechny hodnoty výsledného dotazu z datové struktury. Jakmile jsou data dostupná, je třeba vypočítat výšku grafu (parametr high pro options entitu grafu).

S pomocí získaných hodnot je proveden přepočet hodnot grafu a převod na jednotky tisíců pro přehlednější zobrazení.

Na závěr je datová řada předána entitě grafu spolu s popisky hlavní časové osy a nastavením zobrazení.



## Plán dalšího rozvoje BI řešení

Prezentace prvních výsledků vzbudily u vedení společnosti velká očekávání. Souběžně s přípravou projektu popsaného v této práci probíhala implementace řešení pro podporu prodeje založeného na historických prodejních datech v kombinaci s několika manažerskými parametry. Oba dva projekty mají velký potenciál a obdržely podporu pro další rozvoj.

### 7.1 Budování datového skladu

Na základech databáze Bowbeen vznikl postupně návrh datového skladu na principech stanovených R. Kimballem. V první části bylo rozhodnuto realizovat datové tržiště sdružující prodejní data.

Byla provedena datová analýza a několik iterací dimenzionálního modelování vedlo k návrhu základního modelu tržiště, které je nyní v testovací formě připraveno ve vývojovém prostředí.

Pro následující měsíce je připraveno několik cílů:

- Zakoupení licencí všech potřebných nástrojů a přenesení aplikací z vývojového do produkčního prostředí,
- dopracování všech nových ETL metod, popsaných v části 4.4,
- nastavení pravidelných emailových reportů postavených na datovém tržišti,
- příprava datového tržiště pro potřeby marketingového oddělení,
- datová analýza nového nemocničního informačního systému.

Kromě těchto bodů bude probíhat další testování produkčního prostředí ve snaze maximalizovat výkon datových tržišť. Zároveň je pravděpodobné, že dojde k rozšíření datového tržiště s obchodními daty o nová data vztahující se k nástroji pro příprodeje zboží.

### 7.2 Vývoj reportovací aplikace

Prototyp webové aplikace odhalil potenciál uživatelských reportů. Ve spojení s datovým tržištěm se navíc podařilo data transformovat do lépe pochopitelné formy a odfiltrovat všechny nerelevantní údaje.

#### 7.2.1 Poznatky z prezentace

Při prezentaci první varianty reportovacího nástroje, který zatím umožňuje pouze omezené možnosti, bylo sesbíráno velké množství poznatků a požadavků na nové funkcionality.

Mezi hlavní nedostatky, které vyvstaly z prezentace prototypu a nejsou uvedeny v části 5.2.2, patří:

- **Nedostatečný popis zobrazovaných dat** – Tento problém je z části způsoben neznalostí uživatelů. Nicméně nemalý podíl na něm má i samotný systém. Jako řešení tohoto problému bude připraveno školení uživatelů a v rámci datového skladu bude doplněno větší množství metadat.
- **Omezení v možnostech přímé manipulace s daty** – Uživatelé nejvíce postrádali:
  - funkcionalitu pro práci s grafickou reprezentací dat,
  - funkcionalitu pro změnu pořadí sloupců,
  - funkcionalitu pro práci s jednotlivými řádky v tabulkovém zobrazení,
  - funkcionalitu pro vyhledávání v tabulce.

Uvedené funkcionality byly zařazeny do seznamu požadavků.

#### 7.2.2 Proces vývoje

V následujících měsících bude probíhat vývoj plnohodnotné verze tak, jak byla popsána v kapitole 5. Vzhledem k tomu, že se nové požadavky objevují stále i po ukončení vstupní analýzy, bylo rozhodnuto další vývoj řídit agilně s využitím metodiky SCRUM<sup>58</sup>.

Pro první část projektu bude zvolena velikost sprintu 4 týdny. V nejbližších týdnech bude sestavený backlog, jehož položky budou rozvíjet již připravený prototyp.

Pro první měsíce vývoje jsou připraveny následující úkoly (seřazeno prioritně od nejvyšší):

1. Doprogramovat jádro aplikace s uživatelskými účty,

---

<sup>58</sup><https://www.scrumalliance.org/>

2. připravit sadu nejběžnějších systémových reportů s možností měnit časovou dimenzi,
3. připravit systém definice reportů v externích souborech (pravděpodobně xml),
4. umožnit změnu zobrazených metrik u systémových reportů,
5. doplnit kompletní podporu pro vizualizaci dat pomocí grafů,
6. přidat možnost jednorázově nadefinovat vlastní report (dimenze i metricky),
7. umožnit práci se sekundární dimenzí,
8. přidat podporu pro hierarchie.





---

## Závěr

Cílem práce byla implementace Business Intelligence řešení ve společnosti provozující menší řetězec lékáren. Součástí tohoto řešení byla analýza a návrh datového skladu. V druhé fázi měla být navržena webová reportovací aplikace. V závěru měl být implementován prototyp této aplikace.

Práce popisuje celý proces budování datového skladu od prvotní analýzy až po funkční model. V rámci připraveného prostředí je následně implementováno rozhraní pro dotazování nad datovým skladem s pomocí multidimenzionálního dotazovacího jazyka MDX. Připravené rozhraní je posléze využito prototypem reportovací webové aplikace, která je první fází navrženého plnohodnotného BI nástroje.

Nejprve byla provedena analýza současné situace ve společnosti Medicon Pharm s.r.o. Ukázalo se, že aktuální datové řešení je nedostačující pro analytické úlohy a bylo potřeba provést důkladnou datovou analýzu původních zdrojových databází.

Analýza společnosti v kombinaci s požadavky managementu, omezeným rozpočtem a dalšími kritérii vedla k volbě architektury datového skladu dle R. Kimballa. Při zachování současné databáze Bowbeen jako stage area bylo rozhodnuto jako první realizovat datové tržiště s obchodními daty.

Na základě datové analýzy byl navržen multidimenzionální model nového datového tržiště a byla připravena sada skriptů pro jeho vytvoření a naplnění testovacími daty. Pro dimenzionální tabulky byly využity reálné číselníky ze zdrojových systémů, nebo z původní databáze Bowbeen. Pro finální spuštění datového tržiště je stále třeba zakoupení plné licence SQL Serveru ve verzi Standard a doplnění několika ETL metod.

Po několika iteracích sběru požadavků od vedení společnosti a krátké rešerši dostupných řešení na poli datové analytiky, bylo připraveno zadání pro reportovací aplikaci. Na základě požadavků bylo provedeno modelování případů užití a následně navrženo uživatelské prostředí.

Současně s výše uvedenými byl připraven návrh architektury budoucí aplikace. Návrh je rozdělený na dva funkční celky. První modul zastupuje aplikační

logiku, správu uživatelů a zobrazení ovládacích prvků. Druhý modul obstarává manipulaci z prodejními daty a jejich vizualizaci.

S využitím připravených podkladů bylo implementováno rozhraní datového skladu a následně proběhlo sestavení prototypu reportovací aplikace. Pro vytvoření http rozhraní datového skladu bylo využito webového serveru IIS a komponenty Analysis Services od společnosti Microsoft, která umožňuje jako modul pro SQL Server vytvoření datové kostky a prezentaci dat pro externí programy.

Samotný prototyp reportovací aplikace byl sestavený s využitím PHP a šablonovacího frameworku Foil, které pro druhou část aplikace doplnila sada Javascriptových skriptů pro datovou manipulaci, knihovnen jsGrid a Chartist.js pro zobrazení dat a XMLA4js, která zajišťuje komunikaci s rozhraním datového skladu.

Prototyp aplikace byl prezentován vedení společnosti a schválen pro další vývoj. Současně byl připravený plán na následující měsíce, v rámci kterých by měla být dokončena první verze aplikace připravené k nasazení do produkčního prostředí.

---

## Literatura

- [1] Limited, I.: *Introduction to Information Technology*. Pearson Education, 2005, ISBN 9788177581188.
- [2] A Short History of Data Warehousing. *DATAVERSITY - Data Education for Business and IT Professionals [online]*, srpen 2012, [cit. 07.02.2017]. Dostupné z: <http://www.dataversity.net/a-short-history-of-data-warehousing/>
- [3] Ščuglík, F.: Datové sklady a Technologie OLAP pro dolování dat. [online], [cit. 09.02.2017]. Dostupné z: <http://www.fit.vutbr.cz/study/courses/ZZD/public/seminar0203/DWandOLAP.pdf>
- [4] Ing. Roman Danel, P.: INFORMAČNÍ SYSTÉMY. [online], 2013, [cit. 09.02.2017]. Dostupné z: [http://projekty.fs.vsb.cz/463/edubase/VY\\_01\\_041/Informační systémy/02Textproe-learning/Informační systémy06.pdf](http://projekty.fs.vsb.cz/463/edubase/VY_01_041/Informační systémy/02Textproe-learning/Informační systémy06.pdf)
- [5] Abramson, I.: Data Warehouse: The Choice of Inmon versus Kimball. [online], 2010, [cit. 12.02.2017]. Dostupné z: <https://www.ismll.uni-hildesheim.de/lehre/bi-10s/script/Inmon-vs-Kimball.pdf>
- [6] Austin, B.: Kimball and Inmon DW Models. [online], 2010, [cit. 12.02.2017]. Dostupné z: <https://bennyaustin.wordpress.com/2010/05/02/kimball-and-inmon-dw-models/>
- [7] Abramson, I.: Dimensional modeling. [online], 2010, [cit. 13.02.2017]. Dostupné z: <https://www.ismll.uni-hildesheim.de/lehre/bi-10s/script/bi-00-Lecture6.pdf>
- [8] Zádová, V.: Datový sklad: postavení v rámci IS/ICT, specifika návrhu, modelování. [online], 2014, [cit. 13.02.2017]. Dostupné z: [http://multiedu.tul.cz/~vladimira.zadova/multiedu/Pokrocile\\_databazove\\_systemy/PDS\\_DW\\_navrh\\_2014.pdf](http://multiedu.tul.cz/~vladimira.zadova/multiedu/Pokrocile_databazove_systemy/PDS_DW_navrh_2014.pdf)

- [9] Laberge, R.: *Datové sklady - Agilní metody a business intelligence*. 2016, ISBN 9788025139820.
- [10] Data Warehousing – Schemas. [online], [cit. 14.02.2017]. Dostupné z: <http://www.teradatapoint.com/data-warehousing-schemas>
- [11] Abramson, I.: Dimensional modeling. [online], 2010, [cit. 13.02.2017]. Dostupné z: <https://www.ismll.uni-hildesheim.de/lehre/bi-10s/script/bi-00-Lecture8.pdf>
- [12] Gottwald, T.: Použití SQL Server Integration Services v etapě ETL budování datových skladů. [online], 2017, [cit. 23.02.2017]. Dostupné z: <https://www.vse.cz/vskp/id/7774>
- [13] David, S.; Jan, P.; Ota, N.: *Business Intelligence: Jak využít bohatství ve vašich datech*. 2004, ISBN 9788024766850.
- [14] Abramson, I.: OLAP. [online], 2010, [cit. 23.03.2017]. Dostupné z: <https://www.ismll.uni-hildesheim.de/lehre/bi-10s/script/bi-00-Lecture9.pdf>
- [15] Data Warehousing - OLAP. [online], [cit. 22.03.2017]. Dostupné z: [https://www.tutorialspoint.com/dwh/dwh\\_olap.htm](https://www.tutorialspoint.com/dwh/dwh_olap.htm)
- [16] Balke, S., Wolf-Tilo a Homoceanu: Data Warehousing & Data Mining. [online], [cit. 23.03.2017]. Dostupné z: [http://www.ifis.cs.tu-bs.de/webfm\\_send/922](http://www.ifis.cs.tu-bs.de/webfm_send/922)
- [17] What is XMLA and how does it carry MDX? [online], [cit. 22.03.2017]. Dostupné z: <http://www.simba.com/blog/what-is-xmla-and-how-does-it-carry-mdx/>
- [18] Neoral, J.: MDX tutorial. [online], [cit. 25.03.2017]. Dostupné z: <http://www.neoral.cz/2015/05/mdx-tutorial-1-uvod.html>
- [19] Neoral, J.: MDX tutorial – 2. práce se setem. [online], [cit. 25.03.2017]. Dostupné z: <http://www.neoral.cz/2015/05/mdx-tutorial-2-prace-se-setem.html>
- [20] Neoral, J.: MDX tutorial – 3. počítané členy a currentmember. [online], [cit. 25.03.2017]. Dostupné z: <http://www.neoral.cz/2015/05/mdx-tutorial-3-pocitane-cleny.html>
- [21] CurrentMember (MDX). [online], [cit. 29.03.2017]. Dostupné z: <https://docs.microsoft.com/en-us/sql/mdx/currentmember-mdx>

- 
- [22] IBM Knowledge Center, Design phase. [online], [cit. 01.04.2017]. Dostupné z: [https://www.ibm.com/support/knowledgecenter/en/SS9UM9\\_8.5.0/com.ibm.datatools.dimensionai.ui.doc/topics/c\\_dm\\_design\\_cycle\\_2\\_idgrain.html](https://www.ibm.com/support/knowledgecenter/en/SS9UM9_8.5.0/com.ibm.datatools.dimensionai.ui.doc/topics/c_dm_design_cycle_2_idgrain.html)
- [23] Babcock, B.: Topics in Data Warehousing. [online], říjen 2004, [cit. 01.04.2017]. Dostupné z: <https://web.stanford.edu/class/cs345/slides/Lecture3.ppt>
- [24] Bertrand, A.: Creating a date dimension or calendar table in SQL Server. [online], [cit. 02.04.2017]. Dostupné z: <https://www.mssqltips.com/sqlservertip/4054/creating-a-date-dimension-or-calendar-table-in-sql-server/>
- [25] Configure HTTP Access to Analysis Services on IIS 8.0. [online], [cit. 14.03.2017]. Dostupné z: <https://docs.microsoft.com/en-us/sql/analysis-services/instances/configure-http-access-to-analysis-services-on-iis-8-0>
- [26] FOIL – PHP template engine, for PHP templates. [online], [cit. 16.03.2017]. Dostupné z: <http://www.foilphp.it/>
- [27] XML for Analysis (XMLA). [online], [cit. 20.03.2017]. Dostupné z: <https://help.sap.com/doc/400066065a1b46cf91df0ab436404ddc/2.0.00/en-US/56fea2d06bc24547b47159d8bc5813eb.html>



## Seznam použitých zkratk

|             |                                |
|-------------|--------------------------------|
| <b>BI</b>   | Business Intelligence          |
| <b>CDB</b>  | Centrální databáze             |
| <b>CRK</b>  | Centrální registr kont         |
| <b>DW</b>   | Data warehouse                 |
| <b>ETL</b>  | Extract Transform Load         |
| <b>FK</b>   | Foreign key                    |
| <b>GUI</b>  | Graphical user interface       |
| <b>IIS</b>  | Internet Information Services  |
| <b>JDBC</b> | Java Database Connectivity     |
| <b>KPI</b>  | Key performance indicator      |
| <b>MDX</b>  | MultiDimensional Expressions   |
| <b>MIS</b>  | Medicon Informační Systém      |
| <b>OLAP</b> | On-line Analytical Processing  |
| <b>OLTP</b> | On-line Transaction Processing |
| <b>PK</b>   | Primary key                    |
| <b>SQL</b>  | Structured Query Language      |
| <b>SSAS</b> | SQL Server Analysis Services   |
| <b>SSDT</b> | SQL Server Data Tools          |
| <b>TCP</b>  | Transmission Control Protocol  |

## A. SEZNAM POUŽITÝCH ZKRATEK

---

**UI** User interface

**UML** Unified Modeling Language

**UX** User experience

**XML** Extensible markup language

**XMLA** XML for Analytics



---

## Obsah přiloženého CD

|                     |  |
|---------------------|--|
| readme.txt.....     | stručný popis obsahu CD  |
| exe .....           | adresář se spustitelnou formou implementace                                      |
| src                 |  |
| ├─ impl.....        | zdrojové kódy implementace   |
| ├─ diagrams.....    | zdrojové soubory návrhových diagramů   |
| ├─ thesis .....     | zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ |
| text .....          | text práce   |
| ├─ thesis.pdf ..... | text práce ve formátu PDF  |