



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

## AGILNÍ DATOVÝ SKLAD

AGILE DATA WAREHOUSE

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Martin Smetana

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Kříž, Ph.D.

BRNO 2019

# Zadání bakalářské práce

Ústav:	Ústav informatiky
Student:	<b>Martin Smetana</b>
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Manažerská informatika
Vedoucí práce:	<b>Ing. Jiří Kříž, Ph.D.</b>
Akademický rok:	2018/19

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

## Agilní datový sklad

### Charakteristika problematiky úkolu:

Úvod  
Cíle práce, metody a postupy zpracování  
Teoretická východiska práce  
Analýza současného stavu  
Vlastní návrhy řešení  
Závěr  
Seznam použité literatury  
Přílohy

### Cíle, kterých má být dosaženo:

Cílem je navrhnout, vytvořit a zautomatizovat plnění datového skladu společnosti na platformě Microsoft SQL server. Zdrojem budou jak interní informační systémy firmy, tak i externí data volně dostupná na internetu. Návrh datového skladu musí pokrýt propojení jednotlivých datových zdrojů, možnost historizace dat a implementaci požadavků zákazníka, které budou přicházet v průběhu projektu.

### Základní literární prameny:

BEN-GAN, Itzik, Dejan SARKA a Ron TALMAGE. Querying Microsoft SQL Server 2012: exam 70-461 training kit. Sebastopol, Calif.: Microsoft, 2012. ISBN 978-0-7356-6605-4.

GÁLA, Libor, Jan POUR a Prokop TOMAN. Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi, technologie informačních systémů, řízení a rozvoj podnikové informatiky. Praha: Grada, 2006. 1. vyd. 482 s. ISBN 80-247-1278-4.

LACKO, Ľuboslav. Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby. Brno: Computer Press, 2009. ISBN 978-80-251-2887-9.

LACKO, Ľuboslav. Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle. Brno: Computer Press, 2003. ISBN 80-7226-969-0.

LARSON, Brian. Delivering business intelligence with Microsoft SQL server 2012. 3rd ed. New York: McGraw-Hill, 2012. ISBN 0071759387.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně dne 28.2.2019

L. S.

---

doc. RNDr. Bedřich Půža, CSc.  
ředitel

---

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
děkan

## **Abstrakt**

Bakalářská práce je zaměřena na návrh a zautomatizování plnění vytvořeného datového skladu na platformě Microsoft SQL server. Navržený sklad bude umožňovat historizaci dat, možnost připojení několika různých externích zdrojů z jiných systémů a další specifické implementace na základě požadavků zákazníka v budoucnu.

## **Abstract**

Bachelor thesis is focused on design and automatization filling to datawarehouse on the platform of the Microsoft SQL server. This datawarehouse is designed for historization data, possibility to connect other different external sources and adding specific implementation of customer demand in the future.

## **Klíčová slova**

Datové sklady, ETL, Business Intelligence, Azure, Data Factory, Merge, Automatizace plnění, BI, Microsoft SQL Server.

## **Key words**

Data warehouse, ETL, Business Intelligence, Azure, Data factory, Merge, Automatic filling, BI, Microsoft SQL Server.

### **Bibliografická citace**

SMETANA, Martin. Agilní datový sklad. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/120066>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jiří Kříž.

### **Čestné prohlášení**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně.  
Prohlašuji, že citace použitých pramenů je úplná a že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/200 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne DD.MM.ROK

.....

podpis studenta

### **Poděkování**

Chtěl bych poděkovat svému vedoucímu bakalářské práce panu Ing. Jiřímu Křížovi za asistenci, trpělivost při tvorbě této práce a mnohé cenné podněty. Mé poděkování patří také panu Ing. Zdeňku Noskovi, který mi poskytnul informace o fungování v jeho firmě a za odborné rady při zpracování práce.

# OBSAH

ÚVOD.....	10
CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ.....	11
1 TEORETICKÁ VÝCHODISKA.....	12
<b>1.1 Databáze.....</b>	<b>12</b>
<b>1.1.1 Důležité pojmy v databázi.....</b>	<b>12</b>
<b>1.2 Databázové systémy.....</b>	<b>12</b>
<b>1.2.1 Typy databází.....</b>	<b>13</b>
<b>1.3 Business Intelligence.....</b>	<b>14</b>
<b>1.3.1 Obecná koncepce architektury BI.....</b>	<b>16</b>
<b>1.3.2 Budování architektury BI.....</b>	<b>16</b>
<b>1.3.3 Hlavní komponenty řešení BI.....</b>	<b>17</b>
<b>1.3.4 Architektura Business Intelligence na platformě SQL serveru.....</b>	<b>19</b>
<b>1.4 Datový sklad (DWH).....</b>	<b>20</b>
<b>1.4.1 Struktura datového skladu.....</b>	<b>21</b>
<b>1.6 OLAP.....</b>	<b>24</b>
<b>1.7 Reporting.....</b>	<b>24</b>
<b>1.8 Data Mining.....</b>	<b>25</b>
<b>1.9 T-SQL.....</b>	<b>25</b>
<b>1.10 Inkrementální Load.....</b>	<b>25</b>
<b>1.11 SCD.....</b>	<b>25</b>
<b>1.12 Merge.....</b>	<b>26</b>
2 ANALÝZA SOUČASNÉHO STAVU.....	27
<b>2.1 Informace o společnosti.....</b>	<b>27</b>
<b>2.1.1 Od historie až po současnost.....</b>	<b>27</b>
<b>2.1.2 Předmět podnikání.....</b>	<b>27</b>
<b>2.1.3 Obory podnikání.....</b>	<b>27</b>
<b>2.1.4 Poskytované výrobky a služby.....</b>	<b>28</b>
<b>2.1.5 Využívané technologie.....</b>	<b>29</b>
<b>2.2 Firemní vize, hodnoty a kulturu.....</b>	<b>30</b>
<b>2.3 Organizační struktura.....</b>	<b>30</b>
<b>2.4 Střediska společnosti.....</b>	<b>31</b>
<b>2.4.1 Česko.....</b>	<b>31</b>
<b>2.4.2 Rumunsko.....</b>	<b>31</b>
<b>2.4.3 Slovensko.....</b>	<b>31</b>
<b>2.4.4 Nizozemí.....</b>	<b>31</b>



2.4.5	Plánovaná střediska/expanze .....	31
2.5	Hardware a Software .....	32
2.5.1	Hardware .....	32
2.5.2	Software .....	32
2.6	Komunikace ve firmě a informační toky .....	34
2.7	Proces plnění dat v současnosti .....	34
2.8	Nedostatky stávajícího řešení .....	36
3	VLASTNÍ NÁVRH ŘEŠENÍ .....	37
3.1	Projektový plán .....	37
3.2	Postup k vytvoření datového skladu .....	39
3.3	Vytvoření databáze v Microsoft Azure .....	40
3.4	Iniciální proces plnění datového skladu .....	40
3.5	Struktura datového skladu .....	42
3.5.1	ER model datového skladu .....	42
3.5.2	Stagové tabulky .....	43
3.5.3	Dimenzionální tabulky .....	43
3.5.4	Faktové tabulky .....	46
3.6	Datové pumpy .....	46
3.6.1	Způsob vytvoření pump .....	46
3.6.2	Výběr a přesun tabulek z Navisionu .....	47
3.7	Procedury .....	50
3.7.1	Merge Wizardu .....	50
3.7.2	Popis jednotlivých procedur .....	52
3.8	Plnění faktových tabulek .....	55
3.8.1	Deklarace proměnných .....	55
3.8.2	Kód FactCosts .....	59
3.9	Možnosti pro budoucí rozvoj datového skladu .....	60
	ZÁVĚR .....	61
	SEZNAM POUŽITÝCH ZDROJŮ .....	62
	SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ .....	65
	SEZNAM OBRÁZKŮ .....	66
	SEZNAM TABULEK .....	68
	SEZNAM GRAFŮ .....	69
	SEZNAM PŘÍLOH .....	70

## ÚVOD

Téma své bakalářské práce Agilní datový sklad jsem si vybral z důvodu neustále narůstajícího množství vytvořených dat, které se každým rokem zdvojnásobí, a proto úprava, uložení a analýza těchto dat nabrala zcela jinou váhu. Samotná nezpracovaná data se pomocí nástrojů a postupů Business Intelligence přemění na poznatky a informace, které pro firmy mají velkou hodnotu z hlediska konkurenceschopnosti a pozicí na trhu. Znalosti z dat, které jsme získali z těchto kvalitativních nástrojů umožňují firmám lepší výchozí pozici při důležitých obchodních rozhodnutích.

Na celé této problematice mě nejvíce zaujala důležitost Business Intelligence, které tolik ovlivňuje ekonomickou stránku firem, a také uživatelsky přívětivé prostředí softwarů či aplikací, které se využívají, jak k vytvoření datového skladu nebo následného reportingu nad ním. V této bakalářské práci se zaměřuji pouze na jednu z těchto částí, a to na vytvoření datového skladu. Toto řešení se vytváří ve firmě, která má velké nedostatky, které podrobněji popíšu v analýze současného stavu firmy.

Na začátku objasním cíle této práce včetně podrobností postupu při zpracování, aby pochopení problematiky vytvářeného projektu bylo jednodušší. Teoretická část bude obsahovat základní pojmy týkající se databází, datového skladu, dále z oblasti Business Intelligence, postupy při tvorbě datového skladu a další teorii, která popíše použité specifické okruhy, o kterých jsem se zmínil v praktické části.

V praktické části popisuji vlastní vytvořený datový sklad. Detailně se zde zaobírám, jak probíhala příprava na tento projekt, nedostatky tohoto řešení a v neposlední řadě jednotlivými kroky ve všech systémech, které jsem pro tvorbu skladu použil.

V předposlední bodě mé bakalářské práce se zaměřuji na možnosti rozvoje tohoto řešení, které by firma ráda v budoucnosti zavedla. V úplném závěru této práce jsem zhodnotil celkový přínos pro firmu společně s poznatky, které jsem při analýze tvorby datového skladu získal.

## **CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ**

Cílem je navrhnout, vytvořit a zautomatizovat plnění datového skladu společnosti na platformě Microsoft SQL server. Zdrojem budou jak interní informační systémy firmy, tak i externí data volně dostupná na internetu. Návrh datového skladu musí pokrýt propojení jednotlivých datových zdrojů, možnost historizace dat a implementaci požadavků zákazníka, které budou přicházet v průběhu projektu.

V rámci projektu nejprve proběhlo několikátýdenní firemní školení, na kterém jsem získal potřebné teoretické a praktické základy, které byly nutností ke zpracování méjí bakalářské práce. Po ukončení školení mi byly od jednotlivých vedoucích pracovníků předneseny návrhy a připomínky, které by měly řešit požadavky na vytvoření datového skladu. Se zástupci firmy byl připraven kompletní projektový plán včetně rozpočtu, který by byl na realizaci tohoto projektu poskytnut.

V této části projektu byly nejprve zanalyzovány zdrojové databáze, z kterých byly vybrány pouze data, která vedení firmy požadovalo. Tento časově zdlouhavý postup je celý popsán v jedné z kapitol, kde se data třídila a připravovala. Podle vybraných dat se specifikovala kompletní datová struktura a poté se provedl iniciální proces pro nahrání dat do databáze. Jakmile prostředí bylo připraveno, postupovalo se k tvorbě datových pump, kterými se data ze zdrojů dostala do datového skladu.

Za použití různých nástrojů jako např. Azure Cloud zejména Azure Data Factory a Azure Database a dalších nástrojů byla vyřešena potřebná aktualizace a historizace dat v datovém skladu. Dále byla vytvořena generální procedura, která spouští všechny ostatní procedury spojené s datovým skladem. Celý tento proces se společně s importem dat naplánoval na každý den v ranních hodinách. Při dokončování projektu bylo stanoveno několik konkrétních rozšíření, které se do budoucna budou realizovat.

# 1 TEORETICKÁ VÝCHODISKA

V této kapitole se budu podrobně věnovat potřebné teorii, která je důležitá k pochopení výstupu mého řešení.

## 1.1 Databáze

Nejsnadněji si můžeme databázi představit jako papírovou kartotéku známou z lékařských ordinací. Jedná se o kolekci informací jedné nebo více tabulek, které jsou vzájemně propojeny. Data jsou v databázi následně seřazeny, aby v ni bylo snadné vyhledávání, řízení a úprava (1).

### 1.1.1 Důležité pojmy v databázi

V této podkapitole upřesním pojmy, které v praktické části používám (2).

Data – jsou údaje, která máme někde uložená, máme k nim přístup a jsou z důvěryhodného zdroje. Data jsou následně rozdělena na dílčí údaje, ty jsou navíc ještě uspořádány. Na základě toho lze získat potřebnou informaci, která je důležitá v rozhodovací procesu.

Typy dat – data mohou být textové, číselné, logické (binární) a datumové.

Datové entity – objekty v databázi (tabulky, procesy, pohledy apod.)

Atribut – je jednotlivý sloupec v tabulce. (jméno, příjmení, rodné číslo).

Záznam – jedná se o řádek v tabulce.

Primární klíč – je atribut, který je pro každý záznam jedinečný.

Cizí klíč – slouží k vyjádření propojením a vztahů mezi databázovými tabulkami. To znamená, že díky němu můžeme zjistit, jaká data v databázi spolu souvisí.

## 1.2 Databázové systémy

Systém řízení báze dat (SŘBD) slouží jako softwarový nástroj, kdy nám programová vrstva efektivně zajišťuje práci s velkým množstvím dat v databázi a umožňuje nám ji také spravovat. Patří sem takové nástroje jako např. Oracle, MS SQL server, Sybase, Informix a další (3).

V SŘBD existují tyto funkce pro práci s daty:

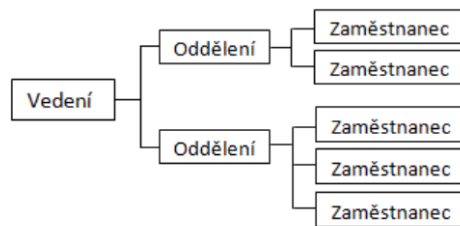
- CREATE - vytvoření báze dat
- INSERT - vkládání dat
- UPDATE - aktualizace dat
- DELETE – vymazání dat
- SELECT - výběr z báze dat

### 1.2.1 Typy databází

Databáze se dělí z hlediska způsobu ukládání dat a vazby mezi nimi na tyto základní typy:

#### Hierarchický model dat

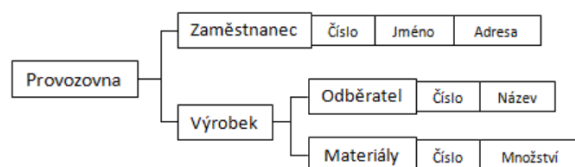
Data jsou zde organizována do stromové struktury. Vztahy mezi záznamy jsou typu rodič/potomek, kdy každý záznam představuje uzel ve stromové struktuře. Navigace ve struktuře je možné pouze směrem přes rodiče na potomka nebo do strany na dalšího potomka. Nevýhodou tohoto schématu je složitost při rušení záznamu (3).



Obrázek 1: Hierarchický model (3)

#### Síťový model dat

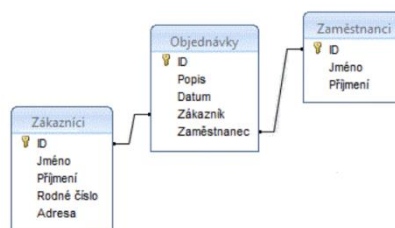
Jedná se o zobecněný hierarchický model, který obsahuje mnohonásobné vztahy, tzv. sety. Jednotlivých spojení může být realizováno na jeden či více záznamů, přičemž tyto sety propojují záznamy různého či stejného typu. K dispozici jsou zde operace jako posuny na potomky nahoru, stranou, dolů nebo i na jejich rodiče v jiném setu. Nevýhodou tohoto schématu je nepružnost a obtížnost struktury (3).



Obrázek 2: Síťový model dat (3)

### Relační model dat

Tento model je nejmladší a nejvyužívanější. Nejčastěji se používá v komerčních řešení pro SRŘBD. Model má velice jednoduchou strukturu, kdy jsou data organizována do tabulek z řádků a sloupců, kde se provádí všechny databázové operace (3).



Obrázek 3: Relační model dat (3)

### 1.3 Business Intelligence

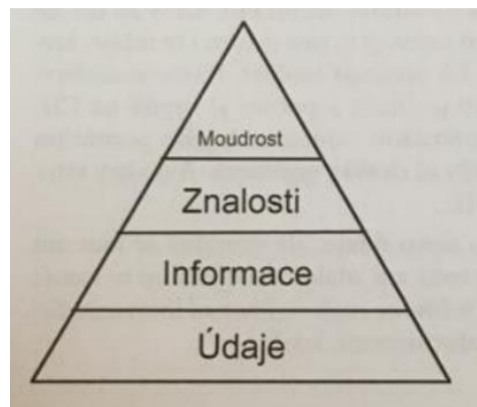
Tento termín definoval jako první Howard Dresner v roce 1989 jako:

*„Business Intelligence je množina konceptů a metodik, které zlepšují rozhodovací proces za použití metrik, nebo systému založených na metrikách. Účelem procesu je konvertovat velké objemy dat na poznatky, které jsou potřebná pro koncové uživatele. Tyto poznatky potom můžeme efektivně použít například v procesu rozhodování a mohou tvořit velmi významnou konkurenční výhodu“ (4, str. 14).*

Business Intelligence se používá pro takové úlohy v informatice, kde je nutná analytická, plánovací či rozhodovací podpora v činnostech, které podnik chce provést na základě získané moudrosti ze svých dat (5, str. 89).

Objem získaných údajů z podnikových oblastí je nadměrný, ale to je právě účel pro business intelligence, který se stará o převod velkého množství údajů na důležité a kvalitní

informace a znalosti pro koncové uživatele či analytiku. Přeměna dat se dá zobrazit na hierarchické pyramidě informačních úrovní (4, str. 15).



Obrázek 4: Hierarchie Informační úrovní (4, str.15)

Základem všeho jsou tedy data. Ta se mohou získávat z nejrůznějších oblastí podnikové řízení, např. nákup, prodej, zaměstnanci, zákazníci, výroba, dodavatele, informační systémy a další (5, str. 89).

Principy Business Intelligence (6, str. 16):

- a. „Jsou určeny pro analytické a plánovací aplikace a tomu odpovídá i organizace dat v jejich databázích,
- b. uchovává data na potřebných úrovních detailu (granularity), tedy detailní i agregovaná,
- c. pracuje primárně s daty podnikových ukazatelů a ty vyhodnocuje podle nejrůznějších dimenzí a jejich kombinací, je tedy založena na multidimenzionalitě
- d. uložení a zpracování dat,
- e. využívá časové dimenze, která umožňuje ukládání dat do databází postupně v časových intervalech a časových snímcích,
- f. je charakteristická vyššími nároky na kvalitu dat”

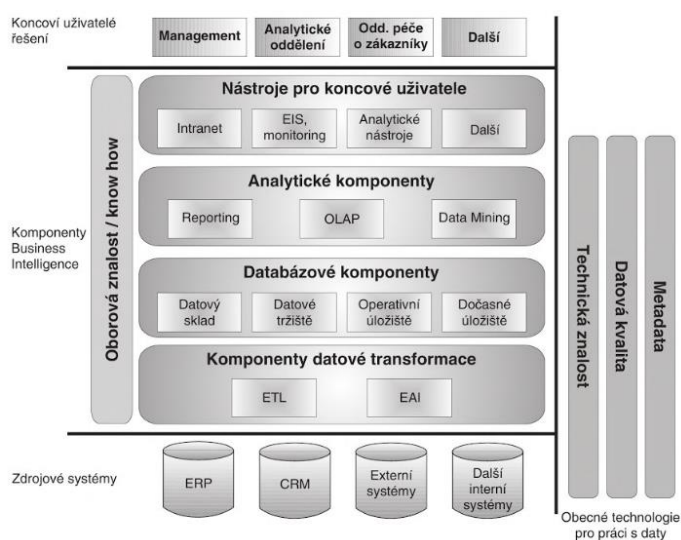
### **Výhody a nevýhody v aplikování BI:**

Jednoznačnou výhodou je to, že data jsou BI systému pohromadě a umožňuje pohled na firmu jako na celek. Tento pohled také umožňují transakční systémy, které firmy normálně používají. Rozdíl je zde takový, že systémy BI nám poskytují přidanou hodnotu

k těmto datům. Málokdo dokáže z celkových pohledů odpovědět na otázky typu: jaké faktory ovlivňují náš zisk, náklady nebo výnosy, jací zákazníci jsou nejvýhodnější nebo zjistit nejvýkonnějšího zaměstnanec. Není to nemožné, ale zároveň to stojí spoustu úsilí, času a peněz. Tyto problémy se systémem BI odpadnou, jelikož BI data zpracuje přehledně, přesně a bez nutnosti vnějšího zásahu. Naopak nevýhodou těchto systémů jsou náklady na provoz a údržbu těchto systémů (7).

### 1.3.1 Obecná koncepce architektury BI

Vyznačená architektura na obrázku a její konkrétní uspořádání se může podle dané situace měnit podle potřeb podniku. Rozmanitost nástrojů BI, kterými lze řešit rozmanité problémy vyústila v to, že tato architektura má několik vývojových větví a samotná aplikace se může v realitě dost lišit. Během vývoje v této oblasti se ustálila tato obecná koncepce (8, str. 114).



Obrázek 5: Obecná koncepce architektury BI (8, str.27)

### 1.3.2 Budování architektury BI

Níže heslovitě popíšu fáze pro úspěšné vybudování datového skladu. Úspěch při budování takové skladu v systému BI závisí na podpoře nejvyšších úrovní modelu, tak zejména na široké úrovni uživatelů, kteří budou datový sklad využívat. Při budování musíme myslet dopředu a už při chystání prototypu musíme výsledné řešení rozdělit na menší celky, abychom ve výsledku docílili rychlejší implementace (4, str.16).



Fáze na nejvyšší úrovni granularity (4, str.16):

- *„Zabezpečení víceúrovňové podpory*
- *Definice vize*
- *Analýza současného stavu*
- *Definice požadavků na nové řešení, určení očekávaných přínosů*
- *Definice/aktualizace procesů pro sběr a použití informací*
- *Výběr a implementace technologie*
- *Podpora koncových uživatelů“*

### **1.3.3 Hlavní komponenty řešení BI**

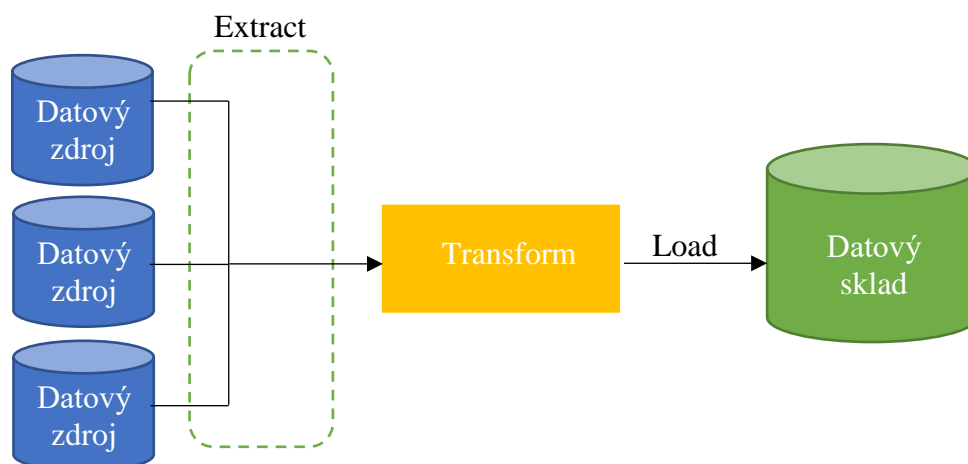
V dalším textu uvedu stručný popis a charakteristiku jednotlivých komponentů v řešení Business Intelligence:

#### **Produkční (zdrojová) databáze**

Jedná se o databázi aplikace, z kterého systémy Business Intelligence získávají data. Produkční databáze bývají jediným hlavním zdrojem pro BI. Příkladem mohou být databáze aplikací a softwarů jsou například ERP (Enterprise Resource Planning), CRM (Customer relationship Management), SCM (Supply Chain Management) nebo běžné excelovské soubory. Zdrojem mohou být i textové dokumenty, kde jednotlivá data jsou rozdělené oddělovači nebo externí soubory např. telefonní seznamy, údaje ze statistických úřadů nebo internetových databází (8, str.115).

#### **ETL (Extract, Transform, Load)**

Je jedním z nejvýznamnějších komponentů, který se nachází uvnitř BI systémů. V některých případech se tomuto procesu říká datová pumpa. Slouží pro přenos mezi dvěma či více libovolnými aplikacemi a databázemi. Jedná se o proces, ve kterém jsou data dávkována v předem určených časových intervalech např. denní, týdenní, měsíční apod. Proces data vybere ze zdrojové databáze (Extract) potom je uloží do cílové databáze (Load) a zde provede příslušné transformace, aby se shodovaly s příslušnou formou (Transform) (8, str.115).



Obrázek 6: Princip ETL (4, str.115)

### **Vrstva pro ukládání dat**

Do této vrstvy patří datové sklady a datová tržiště. Oběma těmto pojmem se budu věnovat v jedné z dalších kapitol (9).

### **Vrstva pro analýzu dat**

V této vrstvě se používají softwary a nástroje pro reporting, vytváří se kostky OLAP a využívá se metoda dolování dat (9).

### **Prezentační vrstva**

Jedná se zde o portálové aplikace nebo EIS (Executive Information Systems). Systémy a aplikace jsou důležitým pilířem u systému BI. Po vybudování fungujícího datového skladu je nutnost poskytnout čitelné, organizované, analyzovatelné a v reálném čase dostupné informace z maxima podnikových databází a externích zdrojů, které jsou využitelné pro řízení institucí a firem. EIS zajišťuje právě tyto činnosti a poskytuje komplexní upravené informace (10).

### **Vrstva oborové znalosti**

Uvnitř vrstvy jsou koncovým uživatelům poskytnuty znalosti z dříve nezpracovaných dat, které jim usnadní rozhodování při nasazování těch nejlepších řešení pro konkrétní situace při řízení firmy (9).

### **Dočasné uložení dat – DSA (Data Staging Area)**

Úkolem dočasného uložení dat je uložení dat z produkční databáze na krátkou dobu s cílem zajistit potřebnou kvalitu pro datový sklad. Hlavním účelem stage

vrstvy je co nejméně zatěžovat zdrojový systém. Tj. rychlá kopie dat bez zbytečných úprav, které zabírají čas. Případně nahrát data například z textových souborů. Tato část slouží k prvotnímu uložení nekonzistentních dat, která po přenosu a upravení se sama uvnitř DSA automaticky odstraní (19, str.26).

### **Operační uložení dat – ODS (Operational Data Store)**

Operační uložení dat je nepovinnou součástí BI systémů. ODS zajišťuje veškeré operativní informační potřeby, jelikož jeho primárním úkolem je pomáhat operativním procesům a nic ostatního nenarušovat nebo cokoli brzdit. Správně implementovaná ODS dokáže současně podporovat operativně-analytické stránky a zároveň dokáže zrychlit kritické operativní procesy několikanásobně (12).

ODS se dělí do několika tříd podle četnosti náhraní. Tyto třídy popsal Inmon následovně (13):

- 1. „V případě první třídy ODS, jsou transakce nahrávány do ODS okamžitě, jakmile nastanou. Koncový uživatel by tedy neměl podle dat rozlišit, zda přistupuje k primárnímu systému nebo k ODS*
- 2. V této třídě ODS jsou změny nahrávány v dávkách několikrát za den (např. po čtyřech hodinách). Je zde tedy znatelné zpoždění proti primárnímu prostředí. Na druhou stranu je implementace ODS třídy II jednodušší než třídy I*
- 3. Pro třetí třídu ODS je charakteristický update v průběhu noci, dochází zde tedy k jednodennímu zpoždění, ale implementace této třídy je nejjednodušší*
- 4. Na rozdíl od prvních třech tříd, které jsou plněny z primárních systémů, je čtvrtá třída ODS plněna analytickými daty z datového skladu. Tato data potom mohou sloužit pro potřeby primárních systémů“*

### **1.3.4 Architektura Business Intelligence na platformě SQL serveru**

Implementaci systému Business Intelligence rozdělujeme na několik nezávislých, ale vzájemně propojených částí. Jedná se o tyto části, které musí být v tomto pořadí (4, str.22):

### **Integrační služby**

Export ze zdrojových databází, poté import do datového skladu a případná transformace dat.

### **Datový sklad**

Vytvoření prostoru pro uložení dat, které bude využívat relačních vazeb mezi tabulkami.

### **Analytické služby**

V této části se původní nekonsolidovaná data stávají cennými informacemi za pomoci nejrůznějších výsledků analýz, nástrojů a predikcemi z data miningu, které následně slouží jako podpora při rozhodování.

### **Reportovací služby**

Informace, které jsme v předchozí části zjistili, tak za pomoci těchto služeb umožníme přístup k výsledkům pro koncové uživatele v určité vhodné formě a rozsahu.

## **1.4 Datový sklad (DWH)**

Datový sklad (Data warehouse – DWH) je relační databáze, která umožňuje řešit analytické úlohy nad obsáhlými daty, které jsou zde uloženy. V datovém skladu se musí ukládání dat probíhat co nejrychleji, které mohou mít různá umístění a formáty. Měl by být také dostatečně flexibilní a pro koncového uživatele přehledný a přívětivý (14).

Jedna z nejznámějších definicí datového skladu popsal Bill Inmon a zní následovně (4, str.38):

*„Datový sklad je podnikově strukturovaný depozitář subjektivě orientovaných, integrovaných, časově proměnných, historických dat použitých pro získání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.“*

Definice Inmona obsahuje pojmy, nad kterými je potřeba se zamyslet a přesněji definovat. Jeho definici tvoří pojmy (4, str.39-40):

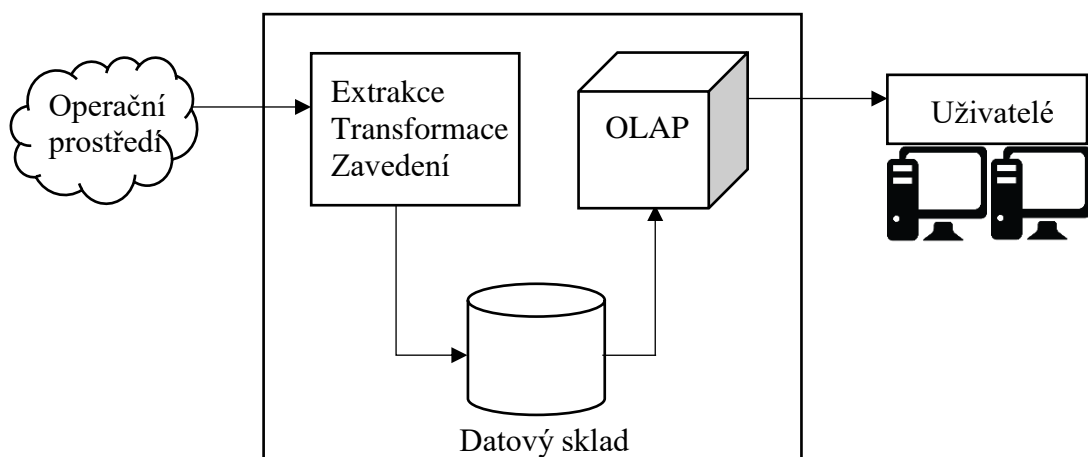
**Subjektová orientace** – *„data se zapisují do datového skladu podle předmětu zájmu než podle aplikace, ve které byla vytvořena“*. To znamená, že v datovém skladu

se data organizují a kategorizují podle subjektu, např. zákazník, dodavatel, výrobek a další. Pokud je systém zaměřen na aplikaci, tak jsou zde data uložena podle, toho z jaké aplikace pochází, např. data aplikace pro odbyt, pro fakturaci apod.

**Integrovanost** – „*datový sklad musí být jednotný a integrovaný*“. To znamená, že musíme zavést jednotnou terminologii a jednotky veličin, jelikož data pocházejí z různých externích zdrojů. Potom se do datového skladu budou ukládat pouze data, která jsou závislá na konkrétní předmět a v systému budou uložena pouze jednou.

**Časová variabilita** – „*data se ukládají jako série snímků, z nich každý reprezentuje určitý časový úsek*“. To znamená, že data se zde ukládají po několik roků, a proto na rozdíl od operační databáze, kde se data uchovávají mnohem kratší dobu. Atributy v datovém skladu obsahují časové údaje.

**Neměnnost** – „*data se v datovém skladu nemění ani neodstraňují, jen jsou v pravidelných intervalech přidávána nová data*“.



Obrázek 7: Procesní schéma datového skladu (4, str.41)

### 1.4.1 Struktura datového skladu

V datovém skladu se nachází převážně jenom dva typy tabulek – Fakta a dimenze tabulky. Členění je v datovém skladu velmi důležité kvůli přehlednosti, jelikož se datovém skladu může nacházet několik stovek i tisíců tabulek (15).

V následující tabulce znázorním hlavní rozdíly mezi těmito dvěma typy tabulek (15):

Tabulka 1: Rozdíly mezi fakty a dimenzemi (15)

<b>Fakta</b>	<b>Dimenze</b>
Obsahuje transakční (numerická) data nebo data, která jsme z dat získali analýzou.	Dimenze jsou popisné atributy faktů.
Obsahují složené primární klíče a cizí klíče, kterými odkazují na dimenzionální tabulky.	Jsou zde uložené textové popisy týkající se businessu.
Fakta obsahují fakta na nejnižší úrovni granularity.	Dimenze jsou denormalizované kvůli mnoha napojením.

### Druhy datových modelů

Každý datový sklad obsahuje komplexní datový model produkčního systému, které obsahují mnoho tabulek včetně jejich vazeb. To umožňuje se dostat k stejnému výsledku větším množstvím způsobům pomocí různých vazeb mezi tabulkami. Existují dvě základní schémata, přesněji „schéma hvězdy“ (STAR scheme) a „schéma sněhové vločky“ (SNOWFLAKE scheme). V centru obou schématů se nachází vždy tabulka faktů, na kterou jsou napojeny tabulky dimenzí. Star schéma je jedno z nejjednodušších typů, které můžeme použít pro datový sklad. Uprostřed je jedna faktová tabulka, na kterou jsou napojeny jedno-dimenzionální tabulky. U snowflake schématu je rozdíl ten, že uprostřed faktová tabulka je napojena na dimenze, které díky hierarchii se dělí do různých úrovní, a proto musí být rozdělena do několika samostatných tabulek (16).

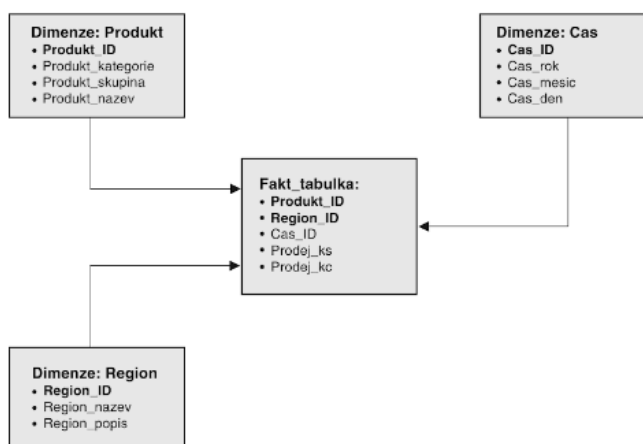
V následující tabulce znázorním výhody a nevýhody použití těchto schémat (16):

Tabulka 2: Rozdíl mezi schématy star a snowflake (16)

<b>Star</b>	<b>Snowflake</b>
Dimensionální tabulky nejsou normalizované	Dimensionální tabulky jsou normalizované

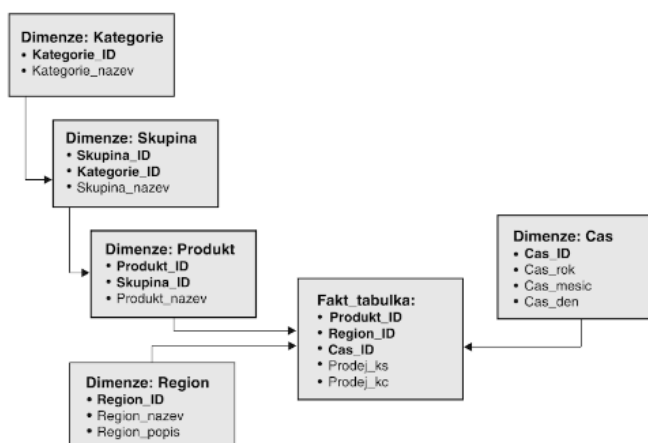
Hierarchie dimenzí je uložena uvnitř dimensionální tabulky.	Hierarchie dimenzí je rozdělena do více tabulek.
Jedna dimenze obsahuje všechna agregovaná data	Data z dimenze jsou rozděleny do různých dimenzí.
Obsahuje faktovou tabulku obklopenou tabulkami dimenzí.	Tabulka faktů je zde obklopena tabulkami dimenzí, které jsou obklopeny dalšími dimenzemi.

Schéma STAR:



Obrázek 8: Schéma hvězdy (19, str.24)

Schéma SNOWFLAKE:



Obrázek 9: Schéma sněhové vločky (19, str.25)

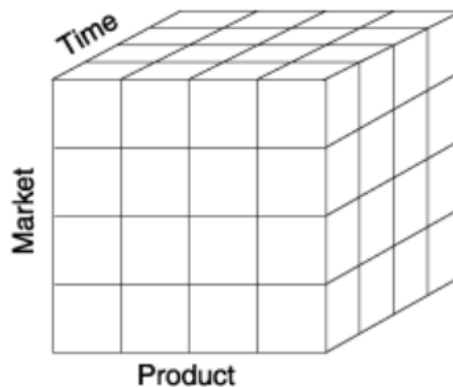
## 1.5 Datová tržiště

„Princip datových tržišť je obdobný jako v případě datových skladů. Rozdíl je v tom, že datová tržiště jsou určena pro omezený okruh uživatelů (oddělení, divize, pobočka, závod). Podstatou jsou tak decentralizované datové sklady, které se pak mohou postupně integrovat do celopodnikového řešení“ (19, str.25).

Datové tržiště je prostor pro elektronické uložení historických dat, které se neúčastní denních operací v organizaci. Místo toho jsou tyto data používány k vytvoření Business Intelligence. Data uvnitř datového tržiště jsou vždy aplikovány na specifickou procesní oblast uvnitř organizace (18, str.29).

## 1.6 OLAP

Online analytical processing (OLAP) systémy umožňují koncovým uživatelům rychle a jednoduše získat informace z dat v datových skladech nebo datových tržištích pro analýzu. OLAP systémy prezentují data pomocí měřítek, dimenzí, hierarchií a kostek. (18, str.44)



Obrázek 10: Princip multidimenzionální databáze na bázi OLAP (19)

## 1.7 Reporting

Business Intelligence reporting (BI reporting) umožňuje přijímat výstup nebo výsledky z BI řešení. Je to obvykle předkonfigurovaná funkce BI řešení. Na základě nastavených parametrů je reporting BI zcela zautomatizovaný proces, který reportuje analyzovaná data z datových skladů. Tyto reporty mohou mít různou formu jako např. statistická data, vizuální grafy nebo standartní text (21).



Reportingy se dělí na dva různé typy (19, str.34):

1. „**Standardní reporting**, kdy jsou v určitých časových periodách spouštěny předpřipravené dotazy“.
2. „**Ad hoc reporting**, kdy jsou na databáze (většinou) jednorázově formulovány specifické dotazy, explicitně vytvořené uživatelem“.

## 1.8 Data Mining

Data mining je proces využívající společnosti k tomu, aby surová data přeměnila v užitečné informace. Pomocí použití softwaru, který hledá vzory ve velkém množství dat, mohou firmy zjistit a naučit se něco více o svých zákaznících a vytvořit tak co nejvíce efektivní marketingovou strategii, která zvýší prodeje a sníží náklady. Data mining závisí na efektivním sběru dat, způsobu skladování a zpracování (22).

## 1.9 T-SQL

Transact-SQL (T-SQL) je sada programových rozšíření od Sybase a Microsoftu, které přidali několik funkcí do strukturovaného dotazovacího jazyka (Structured Query Language – SQL), včetně řízení transakcí, zpracování chyb a deklarování proměnných. T-SQL identifikátory se používají ve všech databázích, server a objektech využívající SQL server. To zahrnuje tabulky, pohledy, procedury, sloupce a datové typy. Identifikátory T-SQL musí mít každý jedinečný název, který je přiřazen při vytvoření objektu a používá se ke ztotožnění objektu (23).

## 1.10 Inkrementální Load

Jak už z názvu vyplývá, jedná se o proces, kdy se přidá pouze ta data do datového skladu, která byla změněna nebo jsou nová od posledního importu, což je rozdíl od „*full loadu*“, kde se importuje vše, bez ohledu na to, jestli to už není v datovém skladu. Využívá se například u importu Faktových tabulek. Zajistit flexibilitu DWH a nedovolit update dat, která mají být nezměněna není vždy jednoduché. Výhodou je snížení přenosu dat od jednoho systému do druhého. (24, str.228):

## 1.11 SCD

Pokud se rozhodneme aplikovat SCD v datovém skladu je toto jedna z nejlepších možností pro aktualizaci a historizaci dat. SCD spočívá v tom, že se každému nastaví SCD typ a na základě typu, pak dimenze vykoná určitou činnost. SSIS obsahuje tuto komponentu, která tyto sloupce může nastavit. Toto jsou tři základní typy (24, str.228):

SCD Type 1 – daná hodnota ve sloupci s tímto type, se přepíše novým údajem.

SCD Type 2 – umožňuje verzování. Přidá se nový řádek, pokud budeme chtít do sloupce s tímto typem přidat novou hodnotu, pod identifikačním číslem, které již je v datovém skladu.

SCD Type 0 – méně často využívaná. Jedná se o nepřepisovanou hodnotu, která se prezentuje jako „Originál“, např. datum narození.

### **1.12 Merge**

V SQL jazyku existuje možnost využití příkazu Merge, který se chová velmi podobně jako SCD, ale nevýhodou je zde, že všechny podmínky se zde musí naprogramovat. Příkaz se skládá ze dvou částí. V první části přesně definujeme, co má příkaz udělat pokud hodnoty ve sloupcích jsou stejné a jaké hodnoty má aktualizovat. V druhé zase definujeme opačnou situaci. Zde už ale neaktualizuje, nýbrž vkládá nové řádky, pokud hodnoty ve sloupcích nejsou stejné (24, str.234)

## 2 ANALÝZA SOUČASNÉHO STAVU

### 2.1 Informace o společnosti

#### 2.1.1 Od historie až po současnost

Společnosti skupiny Gesteem působí na evropském trhu pod různými názvy od roku 1993. Jejich primární zaměření bylo vždy na IT služby, Business Intelligence a Consulting. Založení české pobočky v roce 2015 vedlo k rychlému růstu outsourcingu a offshore služeb. V roce 2016 byly společnosti rebrandovány, takže všechny názvy zahrnovaly GS (Global Services). Od roku 2017 probíhá transformace našich společností s ručením omezeným na akciové společnosti, která podporuje expanzi na nové trhy a založení holdingové struktury.

#### 2.1.2 Předmět podnikání

Jako jedni z největších Evropských lídrů v oblasti Business Intelligence řešení a Near-shore development. Jedná se o společnost, která má české základy a podniká v dalších tří evropských zemích s budoucí další expanzí i mimo EU (Asie).

#### 2.1.3 Obory podnikání

##### PORADENSTVÍ

Společnost Gesteem poskytuje zkušené specialisty schopné pokrýt všechny fáze životního cyklu projektu i odborné poradenství v mnoha oblastech, jako je Business Process Analysis, Solution Architecture a Architecture Design.

##### DATA MANAGEMENT

Poskytují správu dat a vhléd, díky využití specializovaných softwarů, které jejich zaměstnanci bez problému ovládají. Pomocí správného nastavení a ovládání analýzy a integrace je možné získat důležité informace napříč všemi odděleními k obchodnímu rozhodování pro budoucí růst a dlouhověkost.

##### TECHNOLOGIE

Garantují, že jejich odborníci používají pouze ty správné technologie k tomu, aby poskytovali optimální služby s přiměřeným úsilím, časem a rozpočtem. Škála odvětví obchodních partnerů je velmi široká, od financí přes zdravotnictví až po telekomunikace a informační technologie. Spolupracují tak nejen s průmyslovými giganty, ale také se společnostmi malých a středních rozměrů.

## VÝVOJ

Poskytují soběstačné vývojové týmy schopné pracovat v agilním SCRUM prostředí nebo v prostředí Critical Chain Strategií je úzká spolupráce se zákazníky, aby bylo dosaženo jejich maximální spokojenosti.

### **2.1.4 Poskytované výrobky a služby**

CONSULTING & ADVISORY - Krátkodobé poradenství, sloužící k vyřešení konkrétního problému. Může se jednat o technickou expertní práci stejně jako o audit business procesů a architektury dané oblasti. Jejich certifikovaní odborníci jsou připraveni poskytnout potřebné know-how. Tito specialisté mají zkušenosti zejména v oblastech:

- Data Governance
- BI Solution Architecture
- Business Process Analysis
- Security and Penetration Testing
- Microsoft Office 365
- A další..

WEBFORMS - Aplikace pro výměnu dat a informací mezi terénními pracovníky a továrnami/kanceláři. Aplikace umožňuje pracovníkům v terénu efektivně komunikovat s odborníky poskytující druhý a třetí level podpory pro nástroje a výrobky zákazníků.

IoT SOLUTIONS - Inovativní produkt z oblasti IoT glancero.com je cloudový monitorovací systém pro kolaborativní roboty. Sledování produktivity, využití a výkonu v reálném čase, zasílání upozornění na nestandardní stavy prostřednictvím SMS, emailu nebo push notifikací.

BUSINESS INTELIGENCE SOLUTIONS – Pomáhají odemknout sílu v zákaznických datech tím, že poskytnou nástroje, techniky a procesy, které zákazník potřebuje k důležitým obchodní rozhodnutím. Jejich pracovníci umožní vytvořit a pokrýt nejrůznější zákaznické požadavky a zajistí mu tak perfektní analýzu a vizualizaci dat.

G-RESOURCE – Na základě požadavků na odborné znalosti a dovednosti pro konkrétní zakázky poskytují garanci externích expertů na tuto problematiku v přiměřeně krátkém čase. (6-8 týdnů) Tito experti jsou zárukou kvality

provedených prací, v případě nespokojenosti nahradí jiným odborníkem bez dalších nákladů.

G-SHORE – Klient má ve firmě kontakt na jednu osobu, která je tzv. Leager G-Shore týmu. Tento leader zůstává u klienta, což umožňuje rychle reagovat na nové požadavky, operativně snížit nebo zvýšit kapacitu týmu či touto kombinací snižovat náklady na vývoj až o 40%. Celkově G-Shore metodologie umožní několikanásobné zrychlení dodávky.

#### DALŠÍ POSKYTOVANÉ SLUŽBY

- Licence Microsoft
- Data warehouse automation Wherescape
- Produkty společnosti SAS

#### **2.1.5 Využívané technologie**

##### MICROSOFT –

- SQL (SSRS,SSAS,SSIS)
- Power BI
- Cloud Data Platform

WHEREESCAPE – WhereScape zjednodušuje a zrychluje vývoj vysoce flexibilních a auditovatelných datových skladů. Jeho modelově řízený přístup v kombinaci s vytvářením kódů založených na šablonách činí WhereScape spolehlivým doplňkem do každého zásobníku datových skladů našich zákazníků. Společnost Gesteem vytvořili s touto firmou partnerství a v současné době distribuují software pro správu datového skladu WhereScape.

CONCERTO PPM – projektový nástroj, který řídí procesy ve firmě, snaží se je vylepšovat a podporuje metodu critical chain.

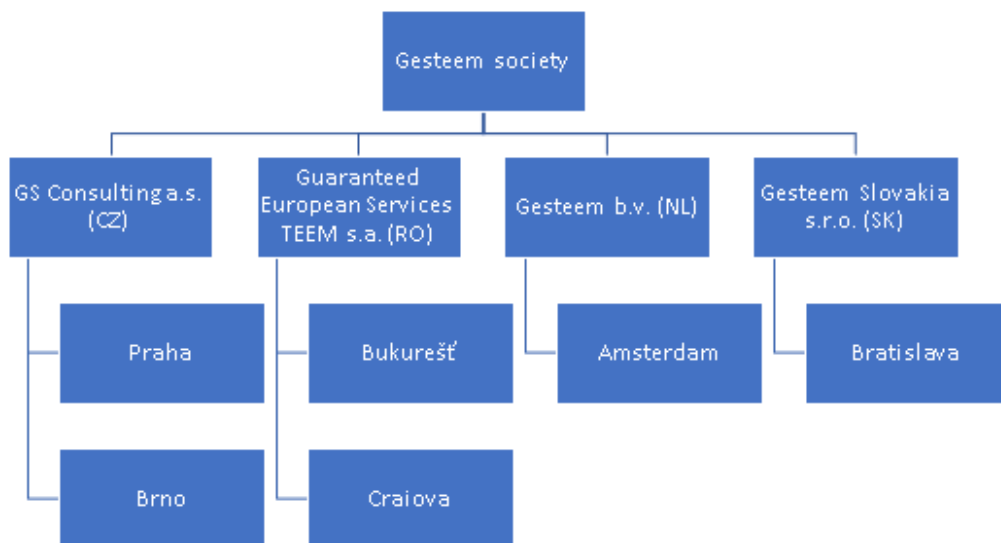
- Hadoop
- Informatica
- SAP Lumira

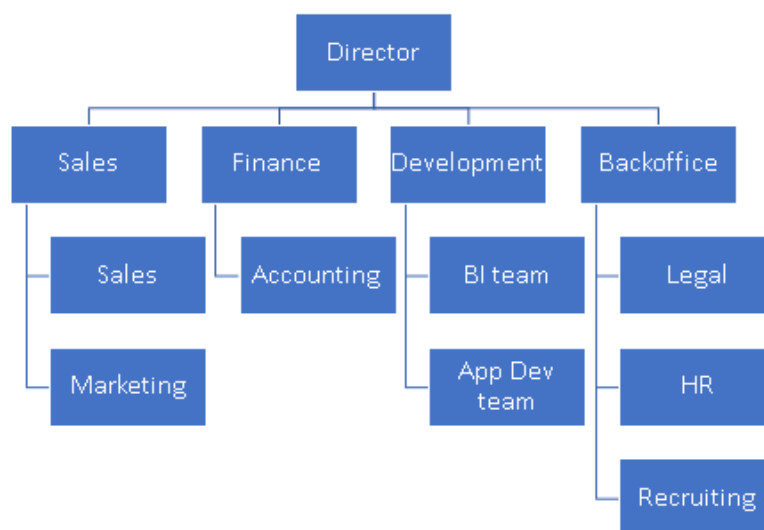
## 2.2 Firemní vize, hodnoty a kulturu

- Netvoříme firmu, ale rodinu
  - Preferujeme důvěru před kontrolou. Ale ztráta důvěry je bohužel nevratná.
  - Preferujeme spolehlivost před povinností. Dané slovo je víc než smlouva.
  - Neplýtváme penězi ani časem. Svými ani druhých.
  - Vytváříme prostředí, kde se každý cítí být vítán.
  - Nejsme armáda. Každý by měl projevit svůj názor.
  - I zákazníci a dodavatelé jsou členové naší rodiny.
- Jsme lidé
  - Pracujeme abychom žili. Zákazníka a osobní život považujeme za stejně důležité.
  - Selhání se stávají, ale nemohou být pravidlem. Když jim nedokážeme zabránit, informujeme o nich co nejdříve.

## 2.3 Organizační struktura

Na následujících obrázcích je znázorněna hierarchická struktura celé organizace Gesteem ve čtyřech zemích – Česká republika, Rumunsko, Nizozemsko a Slovensko. Na prvním obrázku je hierarchie holdingové společnosti. Na druhém je hierarchie funkcí ve firmě, které se mohou prolínat mezi pobočkami.





Obrázek 12: Hierarchické znázornění funkcí ve firmě Gesteem a.s. (autor)

## 2.4 Střediska společnosti

Firma působí v rámci Evropy v mnoha zemích, kdy každé středisko má svá specifická oddělení a aktivity, které jsou spjaty s jejich polohou a zemí v jaké sídlí.

### 2.4.1 Česko

Hlavní stan, řízení firmy

Praha – Velcí zákazníci, korporátní projekty

Brno – pokrytí menších zákazníků, junior program

### 2.4.2 Rumunsko

Offshore teamy, „migrující“ konzultanti.

Bukurešť – Security, penetrační testy

Craiova – Web development, offshore team pro Wherescape

### 2.4.3 Slovensko

Bratislava – delivery centrum zaměřující se na bankovní oblast

### 2.4.4 Nizozemí

Obchodní zastoupení. Žádný vývoj nebo konzultanti

Amsterdam – velký potenciál díky lokálním partnerům. Plán je využít offshore, onsite konzultantů a vývojářů z ostatních poboček

### 2.4.5 Plánovaná střediska/expanze – zatím v jednání mimo EU (Asie)

## 2.5 Hardware a Software

### 2.5.1 Hardware

Každý ze zaměstnanců potřebuje ke své práci notebook. Využívají přenosné zařízení z toho důvodu, jelikož často většina z nich cestuje mezi jednotlivými pobočkami nebo klienty. Jedná se o takové řešení, kdy v kancelářích mají zaměstnanci pouze dokovací stanice, na které se s notebookem připojí ke všem externím zařízením jako je například připojením k síti či uložišti a jiným komponentům. Běžná konfigurace je následující.

**procesor: Intel Core i3**

**operační paměť: 4GB DDR4 SODIMM**

**grafika: Intel GMA3100**

**pevný disk: 500 GB HDD**

**mechanika: bez mechaniky**

**další: 4x USB port, Bluetooth, 1x HDMI port, Wi-Fi**

Jednotlivé notebooky jsou připojeny pomocí dokovací stanice k externím LED monitorům typu: Dell SE2416H, 24" uhlopříčky. K síti jsou zaměstnanci připojeni pomocí WLAN a využívají prostřednictvím této sítě laserovou tiskárnu značky Konica Minolta, Bizhub C454.

### 2.5.2 Software

Na každém notebooku je nainstalován operační systém Windows 10, který je vzhledem k výkonu provozován bez větších problémů. Společnost má koupenou licenci na Office 365, takže využívají poštovního klienta Outlook. Používají ho ve spolupráci s Microsoft Exchange Server pro sdílení pošty a organizaci schůzek. Jako prohlížeč webových stránek používají nejnovější Google Chrome, který pravidelně aktualizují, popřípadě jim sekunduje prohlížeč Mozilla Firefox.

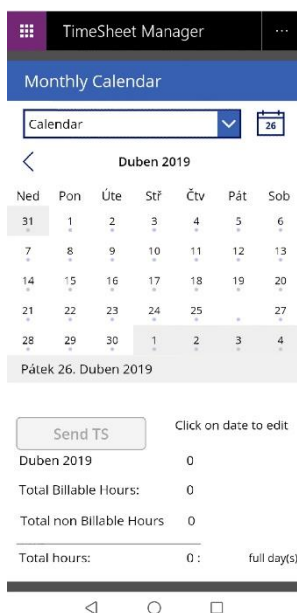
Software používaný v rámci bakalářské práce je Azure. Umožňuje nám skrze cloudovou platformu, kdy pomocí webového prostředí (bez jakékoliv instalace) konfigurovat tento operační systém a na základě potřeb firmy využívat velké množství služeb pro náš vývoj, které fungují i třeba v reálném čase.



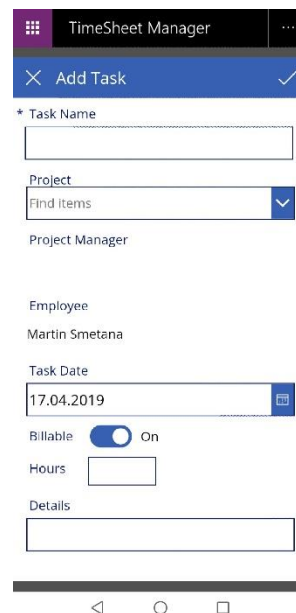
## Navision

Jedná se o podnikový informační systém pro účetnictví a ekonomiku. Firma na tento systém přešla koncem minulého roku 2018 a to z důvodu, že tento systém používá jako uložení databázový Microsoft SQL server, který je umístěn na cloudu. Každá země má vlastní licenci na tento software, a tudíž také svoje uložení (databázi).

**Mobilní aplikace pro generování odpracovaných hodin zákazníků (timesheets)** – firma si vytvořila vlastní aplikaci v nástroji „PowerApps“, kterou bude využívat pro dokumentaci práce svých zaměstnanců nebo externích pracovníků. Nyní tato aplikace běží pouze v testovacím režimu, proto z ní data nejsou ještě v datovém skladu. Níže přikládám snímky z mobilní aplikace.



Obrázek 14: Úvodní obrazovka ve firemní aplikaci (autor)



Obrázek 13: Ukázka funkčnosti a možnosti firemní aplikace (autor)

Na prvním snímku je při načtení aplikace vidět kalendář s celkovým přehledem odpracovaných hodin a po následném vyplnění se odblokuje i tlačítko, kterým tento timesheet odešlete příslušnému vedoucímu pracovníkovi. Druhý snímek nám ukazuje, jakým způsobem se dané pracovní dny vyplní. Uvádí se, co zaměstnanec daný den dělal, jakého projektu se to týkalo, počet hodin a případně detaily.

### Další využívaný software:

- Práce s databázemi - Visual Studio, SMSS
- Reporty - Report Builder, Power BI

## **2.6 Komunikace ve firmě a informační toky**

Ve firmě je komunikace zajištěna pravidelnými schůzkami vedení nebo hovory z konferenční místnosti. Každý týden jsou jasně stanoveny cíle a plány pro každého manažera a jeho tým. Problém zde nastává při přerozdělování práce či samotné koordinaci. Vyšší management je zde rozdělen na více projektů současně a následně dochází k mírnému chaosu, kdy například nejsou přístupová hesla k serverům, správně nastavená přístupová a uživatelská práva nebo vypršené licence, aniž by o tom kdokoliv z firmy věděl.

Komunikace, která probíhá vně firmy pro získání nových zákazníků nebo zaměstnanců se pomalu zlepšuje. Firma má několik důležitých klientů, které jsou zároveň jejich hlavními projekty, avšak získávání menších klientů se v posledních měsících muselo razantně změnit. Krok kupředu zvolili tím, že vytvořili a posílili marketingové oddělení, které se stará o prezentaci firmy, např. zkvalitnění webových stránek, vytvoření propagačních materiálů. Pro inzerci a celkové zviditelnění firmy se rozhodli ve formě externí firmy, která jim udělá reklamu na sociálních sítích a jiných internetových portálech, které jim umožní, tak zapůsobit na větší množství lidí a získání tak nových klientů a zaměstnanců.

## **2.7 Proces plnění dat v současnosti**

Jedná se o manuální proces, kdy se data musí pravidelně v měsíčních intervalech kontrolovat a aktualizovat. Nynější řešení nám poskytuje komplexní pohled na pobočku z hlediska nákladů a příjmů. Problém nastává, pokud firma má tyto náklady úměrně rozdělit mezi klienty nebo zjistit celkové náklady na zaměstnance. Tento zdoluhavý proces nyní zajišťuje jejich finanční ředitelka, která nad těmito zdroji dat z jednotlivé pobočky stráví mnoho času, než by bylo nutné. Toto řešení je neefektivní v tom, že musí zkombinovat několik externích zdrojů dohromady, což je ve výsledku velmi

problematické a může docházet k vysoké chybovosti. (finanční software, excel, docházkový systém apod.) Pokud se na tuto situaci podíváme z praktického hlediska data nejsou nijak centralizována, tzn. že zde dochází k časovým prodlevám, než oprávněná osoba dané pobočky pošle potřebné zdroje dat danému nadřízenému. Také je v každém státě jiná legislativa, na kterou se musí přihlížet, při následném reportingu, kdy se toto může opomenout případně špatně nastavit. V souhrnu nynější řešení je zdlouhavé, chybové a data pro následný report vytváří osoba, která to nemá v popisu své práce a brání to tak firmě v celkovém ekonomickém růstu.

Obrázek 15: Ukázka excelovského sešitu používaný pro analýzu bankovního účtu firmy (autor)

Obrázek 16: Ukázka excelovského sešitu používaný pro analýzu všech rozvahových účtů firmy (autor)

Na obrázcích výše jsou znázorněny excelovské sešity exportované z transakčních systémů.

## **2.8 Nedostatky stávajícího řešení**

Vzhledem k vlastní podrobné analýze současného stavu firmy vyšly najevo nedostatky stávajícího řešení. Tyto nedostatky budou hrát důležitou roli při návrhu a tvorbě nového řešení, které by je mělo zcela eliminovat. Nedostatky jsou následující:

- doba naplnění
- doba na celkové vyhodnocení dat
- vysoká míra chybovosti
- data uložená na různých místech (emaily, Excelovské sešity)
- neexistuje možnost reportingu
- nedostatečný monitoring finančních toků

## 3 VLASTNÍ NÁVRH ŘEŠENÍ

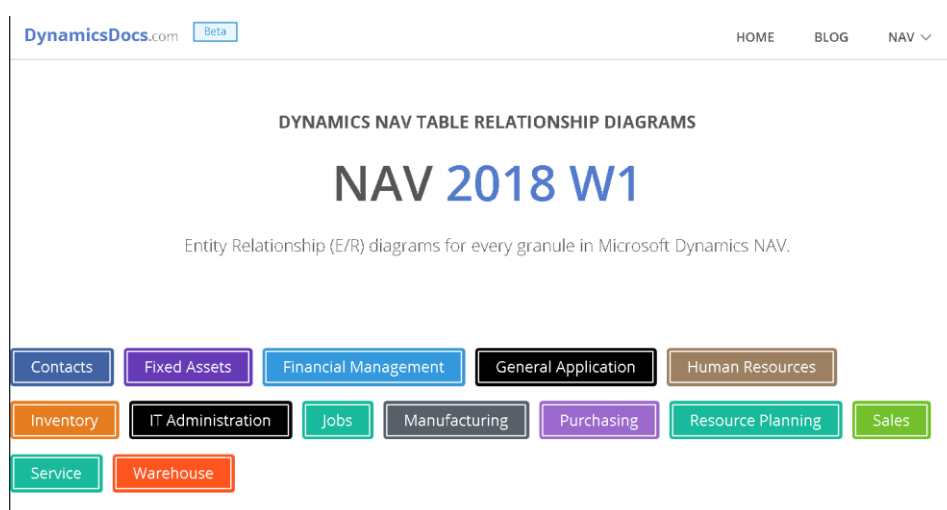
V této kapitole se budeme zabývat návrhem a vytvoření funkčního datového skladu.

### 3.1 Projektový plán

Předtím, než se pustíme do fyzického vytváření datového skladu, musíme přesně určit formulace a rozsah projektu. Na to, aby byl takový plán kvalitní, je důležitá komunikace se zadavatelem a pochopení celkového projektového plánu pro takové řešení.

Projektový plán obsahuje několik částí:

- **Definování cílů** – úplně jako první nám bylo vysvětleno jakého cíle bychom měli dosáhnout a okrajově nám nastínili praktičnost vybudovaného řešení. Bylo to sezení, kterého se účastnil vyšší management této firmy, jelikož se jedná o interní projekt. Formulace a definice těchto cílů by měly vycházet z toho, aby přímo odpovídali tomu, k čemu chce firma daný datový sklad využívat.
- **Analýza zdrojů** – tento krok zabral více času, než by se na první pohled zdálo. Datové zdroje pro náš projekt jsou ERP databáze softwaru Microsoft Dynamics NAV. Tyto databáze obsahují přes 2 tisíce různých tabulek, které se museli projít a zanalyzovat. K zjištění vazeb a dalších užitečných parametrů tabulek jsem využil tuto stránku: [www.dynamicsdocs.com](http://www.dynamicsdocs.com), na které vše bylo přehledně rozděleno podle kategorií a to mi napomohlo si rychleji ujasnit, jaká je struktura celého systému a vybrat tak správné tabulky pro datový sklad.



Obrázek 17: Ukázka webu dynamicsdocs.com (25)

- **Analýza požadavků** – projekt, který vytvoříme, je pro široké spektrum koncových uživatelů a každý z nich má jiné představy a požadavky, a proto jsem to s každým zvlášť musel konzultovat. Dalším důvodem bylo zmapování klíčových činností ve firmě, které mi poskytly informace podstatné pro stavbu a strukturu tabulek v datovém skladu, kdy jsem tak mohl předcházet problémům s chybějícími daty a zbytečně se nezdržovat.
- **Rozčlenění úkolů** – rozčlenění je zde dvojitý. Z globálního pohledu se projekt dělí na back-end a front-end. Následně z lokálního pohledu má back-end hned několik úkolů. Back-end je zde právě pro analýzu, export, transformaci a načtení dat, vytváření datové struktury a zautomatizování.  
Kolega, který má druhou část se pak stará o kompletní reporting v několika různých softwarech a jejich zabezpečení.
- **Definování doby trvání** – celková doba projektu se zde dopředu neuváděla, jelikož se jedná o interní projekt, nikoliv projekt pro zákazníka.
- **Sekvence úkolů** – u projektů je posloupnost úkolů velmi důležitá, protože při nepostupování podle nějakého plánu se velmi rychle dokážete ztratit v kódech a stovkách tabulek. Na druhou stranu samotný software při vytváření datové struktury vám logicky nedovolí postupovat jinak. Například musíte nejprve vytvořit tabulky, abyste do nich mohli načíst data.
- **Rozpočet** – na tento interní projekt jsme se dohodli, že firma vyhradí přibližně 200 tisíc korun. Rozdělení financí bude následovné: ¼ se použije na nákup licencí a datových prostor, který umožní provoz datového skladu na minimálně rok dopředu. Zbylé ¾ se použijí na platy.

Po ujasnění a průzkumu zdrojů je dalším velmi důležitým krokem určit čeho je možné dosáhnout (in scope) a čeho ne (out of scope). Předejde se tím zbytečným problémům, nejasnostem a každý už od počátku projektu ví, co přesně očekávat jako výsledek.

V mém projektu je to následovně:

#### **IN SCOPE**

- Integrace dat ze všech systémů a zdrojů
- Vytvoření sumarizačních souhrnů z jednotlivých detailů dat
- Připravená data pro reporting

- Aktualizace a historizace dat
- Automatické plnění na denní bázi
- Převod měn do fixních kurzů

### **OUT OF SCOPE**

- Nahrávání aktuálních denních kurzů měn – pro funkčnost datového skladu prozatím nepotřebné – používají se fixní kurzy měn.
- Zjištění zaplacených faktur – nutnost propojení s bankou – toto řešení by vyžadovalo velké množství času, které jsme neměli.
- Detailní pohledy až na zaměstnance – firma neposkytla zdroje z aplikace pro timesheety, která běží v testovacím provozu. Řešení je na přidání připravené.

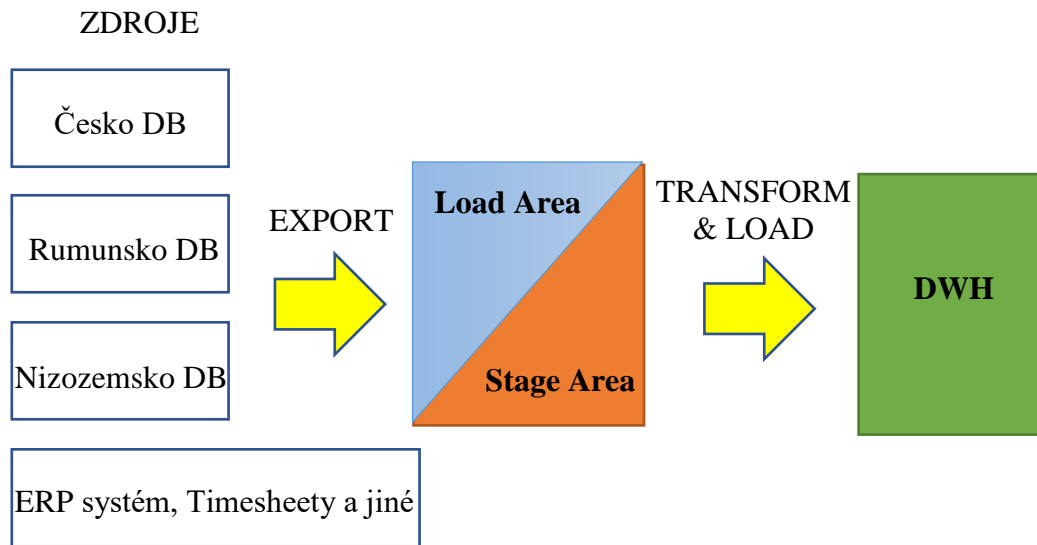
### **3.2 Postup k vytvoření datového skladu**

Abych úspěšně vybudoval datový sklad, splnil všechny požadované cíle a na nic nezapomněl, tak důležitým prvním krokem je si nejprve ujasnit kompletní postup, který rozdělím do jednotlivých kroků, které následně znázorním ve schématu.

Postupoval jsem podle těchto následujících kroků:

1. Vytvoření databáze na cloudu
2. Získání přístupu ke zdrojovým datům
3. Vybrání datových zdrojů a posouzení struktury dat
4. Vybrání způsobu exportu/přenosu dat na základě využívané platformy
5. Vytvoření přípravného prostoru
6. Spuštění importu do přípravného prostoru (stage)
7. Vytvoření procedur pro aktualizaci dat
8. Vytvoření star-schématu (dimenzionální a faktové tabulky)
9. Vložení dat do star-schématu ze stage
10. Zautomatizování

### Schéma postupu při vytváření datového skladu:



Obrázek 18: Schéma postupu při vytváření datového skladu (autor)

### 3.3 Vytvoření databáze v Microsoft Azure

Ještě předtím, než začnu popisovat daný postup, tak bych rád odůvodnil, proč jsem si vybral zrovna tuto platformu. Důvody jsou velmi jednoduché. První je ten, že firma nevlastní žádné potřebné hardwarové komponenty a také nemá prostory, kam by je umístila, případně personál, který by se o něj staral. Druhý je spíše skupina důvodů, kam spadá např. jednoduchost při zavádění, náklady na provoz či aplikace velkého množství služeb, které lze implementovat téměř okamžitě.

Tato platforma mi umožnila vytvoření databáze na cloudu, kdy se využila licence, kterou mi firma zakoupila. Úplně prvopočátkovým řešením mělo být využití poukazu pro cloudového Azure datawarehouse, jenže zde si Microsoft účtoval hodinově několik dolarů a databáze by tak vydržela přístupná jen pár desítek hodin. Vysokou cenu Azure DWH nevyvážily ani jeho výhody, a proto jsem zvolil cloudovou databázi, kde se platí pouze za prostor, který využijete. Pro mé řešení stačila celková velikost databáze 2 GB.

### 3.4 Iniciální proces plnění datového skladu

V této části jsem už fyzicky provedl některé kroky, které jsem dříve popsal. Jedná se o proces, který se skládá z pěti kroků. Proces je něco jako univerzální postup pro přípravu datového skladu, kdy naše databáze postrádá jakoukoliv strukturu nebo ji žádná databáze



není. Toto se používá jako odrazový můstek, kterým se pokryje většina požadavků a následně se pouze na tuto inicializaci přidávají pouze individuální požadavky zákazníka.

- **Vytvoření datové struktury** – všechny zdrojové databáze jsou stejné, jelikož se používá Navision. Pro ulehčení tvorby skladu jsem vybral pouze tabulky, které budeme potřebovat a se stejnou strukturou sloupců jsem je vytvořil jako přípravné místo (stage) v našem datawarehouse (DWH). Zde nesmíme zapomenout, že pro každý zdroj musíme mít specifický stage. Při znalosti struktury zdrojů a požadovaného výstupu už může také tvořit zbylé Dimové tabulky a v návaznosti na ně i Faktové, které budou dvě. První pro náklady a druhá pro výnosy.
- **Naplnění dimenzí neznámým údajem** – tímto bodem se myslí to, že všechny dimenzionální tabulky musí obsahovat záznam s identifikačním číslem -1. Důvodem je to, že pokud by se stalo a nějaký záznam by nám v dimenzi chyběl, tak bychom díky agregačním podmínkám vyřadily určité záznamy z Faktové tabulky, což by pak znehodnotilo výsledná data pro reporting.
- **Provedení Importu** – nachystaná struktura a připravený DWH mi umožní ze zdrojových databází připravit datové pumpy neboli „*pipelines*“, které nám přenesou/zkopírují data do přípravného prostoru v datovém skladu. Tyto pumpy se nastaví v přehledném uživatelském prostředí Azure, kde se dají i načasovat. Podrobněji celý tento proces popíši později.
- **Úprava a přesun dat do dimenzionálních tabulek** – tabulky v Navisionu bývají velmi obsáhlé, a tak je nutné vybrat opravdu pouze ty, které budeme nyní nebo v budoucnu využívat. Je to hned z několika důvodů. Závisí na tom rychlost načítání dat a také to je zbytečná komplikace, pokud nějaký sloupec vynecháme a následně zjistíme, že tento údaj potřebujeme. V tomto případě ho pak musíme složitě znovu implementovat. Celkově u sloupců tabulek v DWH je nutné nastavit aktualizaci a přidávání nových údajů. Tohle se zde řeší pomocí MERGE,

kdy každý sloupec nastavíme, co má provést při importu nových dat. MERGE je zde kvůli tomu, abychom nezatěžovali systém importem všech dat, které již v tabulkách máme a pro historizaci např. pokud se něco odstraní ze zdrojových tabulek.

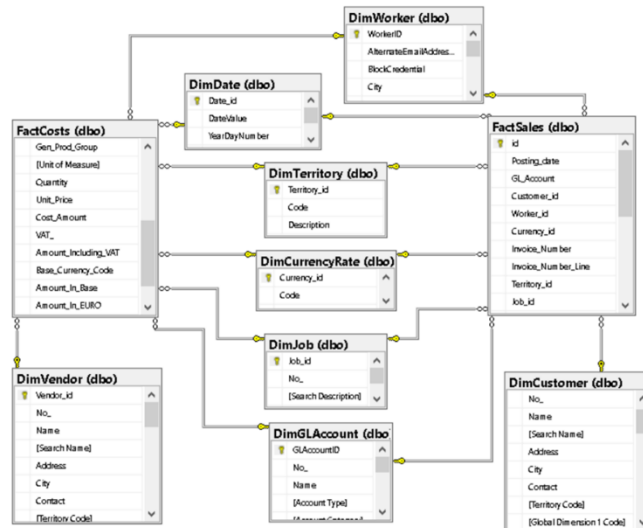
- **Naplnění Faktových tabulek** – v mém projektu se nachází dvě tyto tabulky. Každá má jiný účel, ale jejich struktura je až na pár odlišností velmi podobná. Pro každou z nich je vytvořena stage tabulka, která obsahuje více druhů dokladů z účetnictví. Podmínkou je zde to, na jaký účet byl daný doklad zaúčtován a podle toho se roztrídí. Zaobíráme se zde pouze účty začínající 5-kou (náklady) a 6-kou (výnosy). Tabulky se následně propojí se všemi dimenzionálními tabulkami pomocí cizích klíčů. Díky tomu se údaje na dokladech překonfigurují na identifikační čísla uložená v mých dimenzionálních tabulkách a následně pomocí skriptu „*insert into select*“ se obě dvě tabulky celé naplní.

### 3.5 Struktura datového skladu

V následující kapitole se budu věnovat grafickému znázornění datového skladu a jeho vazbám uvnitř. Popíši zde i části, do kterých se datový sklad dělí – přípravná část tzv. stage, dimenzionální a faktová část.

#### 3.5.1 ER model datového skladu

Na následujícím obrázku je umístěný vygenerovaný diagram vytvořeného datového skladu. Zde můžeme vidět, že je datový sklad vytvořen ve schématu hvězdy (star), kde každá faktová tabulka je napojena pouze do dimenzí. Faktové tabulky se ve skladu nachází dvě a jsou rozděleny podle toho, jaké dokumenty budou obsahovat. Toto řešení by šlo udělat i s jednou faktovou tabulkou, ale kvůli přehlednosti a budoucím záměrům vedoucích členů firmy, se musely vytvořit oddělené faktové tabulky.



Obrázek 19: ER diagram vytvořeného datového skladu (autor)

### 3.5.2 Stagové tabulky

Tyto přípravné tabulky slouží k exportování dat ze zdrojů, a tak mají stejnou strukturu sloupců jako zdroje, aby při mapování nedocházelo k problémům. Zde se tabulky dělí na loadové a stageové. Je to kvůli problému, který zde nastal tehdy, když jsem zjistil, že Azure – Data Factory neumí přidávat samotné sloupky s daty. Pracuje pouze s uloženými procedurami uvnitř datového skladu. Každou z procedur bych musel složitě parametrizovat. Tento problém se vyřešil rozložením každé databáze státu do vlastního loadu, který se následně spojí v jednu přípravnou tabulku stage. Toto spojení se realizovalo pomocí operátoru „union“, které jsem vytvořil jako procedury. Další úprava dat probíhá za pomoci merge do dimenze, který umožňuje historizaci a aktualizaci dat v jednom kroku. Celkově o těchto procesech se zmíním v některé z následujících kapitol.

### 3.5.3 Dimenzionální tabulky

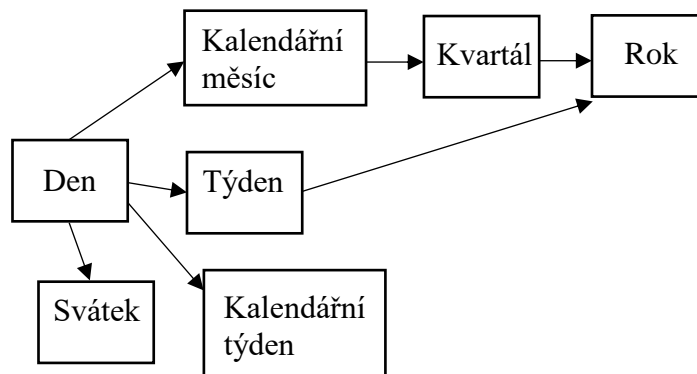
- **DimCustomer** – tabulka zákazníka, měla by být obsažena pouze ve faktové tabulce FactSales. Tabulka obsahuje detailní informace o zákazníkovi a je propojena navázána na jeho bankovní účet. Pro ukázkou funkčnosti jsem zde vybral pouze jeden z bankovních účtů každého zákazníka, který se tímto prioritizoval. Dimenze nemá žádnou hierarchii. Atributy této dimenze můžeme rozdělit do několika skupin. Do první patří

automaticky generovaný primární klíč, který se přidá ke každému řádku tabulky, následně odkdy je daný řádek platný a dokdy (Valid From/To). Druhou skupinou jsou kontaktní údaje, např. jméno, příjmení, kód zákazníka, priorita. Třetí skupinou jsou zeměpisné údaje, např. adresa, směrovací číslo, země původu apod. Poslední skupinou jsou finanční údaje, např. jméno banky, číslo bankovního účtu, dobu splatnosti, výši zdanění apod.

Tyto atributy jsou nastaveny jak na SCD1, tak i na SCD2, jelikož je pro nás důležité zachovávat si informace o zákaznících do budoucna a také z důvodu geografické analýzy. Právě zeměpisné údaje a IČO jsou nastaveny na SCD2.

- **DimVendor** – tabulka dodavatelů, oproti tabulce zákazníků se liší pouze v tom, že se nachází ve faktové tabulce FactCosts, jinak mezi nimi strukturově ani v nastavení není žádný rozdíl.
- **DimCurrencyRate** – tabulka kurzů měn. Jedná se o fixně vytvořenou tabulku kurzů třech měn, ve kterých firma obchoduje. Je připojena k oběma faktovým tabulkám. Tabulka obsahuje zkratku měny, název měny a dvě hodnoty s daty, které označují, zda jsou hodnoty validní. Není zde nastavena žádná aktualizace, jelikož do budoucna se bude řešit celkové načítání kurzů denně. První využití je pro centralizaci faktur v jedné měně ze všech států a druhým využitím je to, že majitelé chtějí zjistit v jaké měně se nejvíce vyplatí obchodovat – fixním nebo denním kurzem.
- **DimDate** – tabulka času. Jedná se o jedinou hierarchickou tabulku uvnitř datového skladu. Tabulka je vytvořena přes skript od roku 1900 až do konce roku 2035. Jsou zde uvedeny měsíce, roky, pořadí dnů v roce, kvartály, den v týdnu, název dne, svátky apod.

Výhodou u této hierarchické dimenze je ten, že se dá automaticky vygenerovat dopředu do již existující dimenze.



Obrázek 20: Hierarchie dimenze datumu (autor)

- **DimGLAccount** – tabulka rozvahových účtů. Kromě atributu pro identifikační číslo a dvou časových hodnot na validitu se v tabulce nachází pouze detailní popis jednotlivých účtů a přesná specifika, např. název, typ, kategorie, nebo jestli je účet blokový. Zde jsou všechny atributy nastaveny na SCD1, ale celkově se zde nepředpokládá, že by se měnily nebo se nějaké nové účty přidávaly.
- **DimJob** – tabulka prací v projektech. Tabulka obsahuje seznam prací, které je možné provádět na projektech ve firmě a jejich časovou náročnost. Opět tabulka obsahuje mimo jiné i automaticky generované identifikační číslo a také atributy validity. Nastavení atributů je stejné jako u účtů, vše SCD1.
- **DimTerritory** – tabulka zemí. Fixně vytvořená tabulka pro budoucí hierarchické rozdělení poboček pod danou zemi. Obsahuje identifikační číslo, zkratku země, název země a validitu.
- **DimEmployee** – tabulka zaměstnanců. Prozatím není napojena na žádnou z faktových tabulek. Důvodem je to, že chybí zdroje z aplikace a také se plánuje upravení v zapisování dat, kdy budou k jednotlivým fakturám přiřazovat čísla zaměstnanců, kteří na zakázce pracovali, aby je šlo propojit. Nyní je tedy tabulka předpřipravená. U atributů je nastaveno jak SCD1, tak i SCD2. SCD2 je nastaveno jen u zeměpisných údajů.
- **DimWorker** – tabulka pracovníků. Tabulka není také prozatím na žádnou z faktových tabulek napojena. Jde o externí excelovský soubor, který byl nahrán do databáze. Obsahuje zaměstnance i dodavatele (externisty), kteří

pro firmu pracují. Tabulka se vytvořila z toho důvodu, že někteří dodavatelé nejsou umístěni v tabulce DimVendor, a proto v budoucnu dojde za pomoci této tabulky k přidání těchto externistů k dodavatelům. Nyní jsou zde jen předpřipravení.

### 3.5.4 Faktové tabulky

- **FactSales & FactCosts** – obě dvě faktové tabulky obsahují skoro všechny dimenze. Rozdílné jsou pouze v tom, že FactSales obsahuje zákazníky a FactCosts zase dodavatele. Dále obě obsahují datum vystavení, číslo účtu, zaměstnance, měny, území, číslo dokumentu, řádek dokumentu a poté už jenom další finanční hodnoty. Tyto finanční hodnoty bere každá z jiného Stage, kde se načítají z více různých detailních zdrojových tabulek, které slouží pro účetnictví.

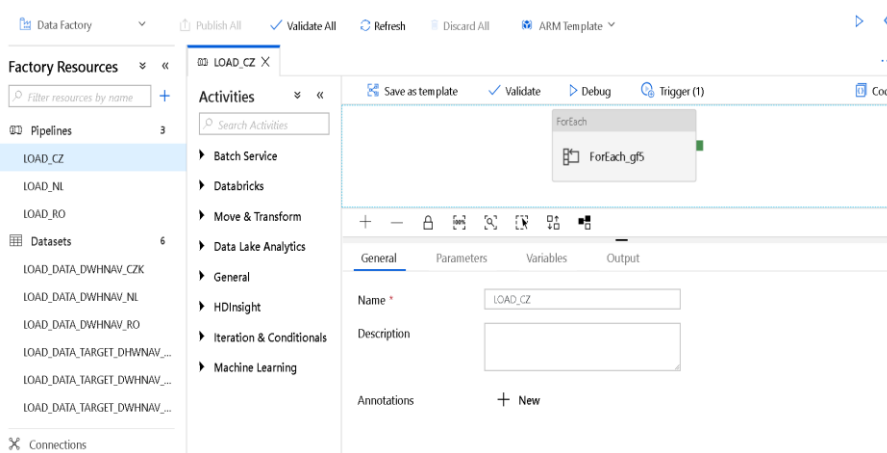
## 3.6 Datové pumpy

Vytvoření těchto pump mi umožňuje velmi jednoduše a v pravidelných intervalech překopírovat potřebná provozní data každé země ze zdrojových databází. Pumpy se vytvořily v nástroji v Microsoftu Azure v sekci Data Factory. V této kapitole se tedy zaměřím na postup, kde popíši, jak jsem tyto pumpy vytvořil.

### 3.6.1 Způsob vytvoření pump

V Azure jako první krok musíme dokoupit nejprve rozšíření nazvané Data Factory. Po zakoupení a nastavení základních atributů pro jeho fungování se mi zobrazí úvodní obrazovka, která je znázorněna na obrázku níže. Pracovní plocha nástroje se dělí na dvě části – zdroje a aktivity. Nejprve bych se zaměřil na zdroje a jakým způsobem se zde s daty pracuje. Jako nejjednodušší možnost, kterou jsem i zvolil, je „Copy data“, která umožní přesun více tabulek zároveň a stačí u této možnosti vybrat pouze zdrojový (link) a cílový (sink) server. Následně se vyberou tabulky, které chceme přenést, poté se vše automaticky namapuje a data mohou procházet. Ve výsledku se automaticky vytvoří „for each“ cyklus, který obsahuje parametry jako název struktury, aby věděl, z jaké zdroje data brát. Také je zde parametr název tabulky „@item().destination.tableName“ a na základě daného mapování prochází vybrané tabulky jednu po druhé. Důležitou podmínkou je zde

to, že struktura ve skladu musí být perfektně nachystaná a také se nesmí zapomenout na „pre-skript“, který před každým takovým importem dat promaže dosavadní tabulky, aby nedocházelo k chybám skrz duplicitu. Oproti tomu v aktivitách je nespočetné množství funkcí a úprav jakým lze přenos dat optimalizovat. Například použít vlastní kód v Pythonu nebo JSONu (celý tento nástroj je v tomto jazyku naprogramovaný). Během vytváření těchto pump jsem přišel na velkou nevýhodu tohoto nástroje a ta je v tom, že jakékoliv pokročilejší rozšíření a usnadnění při kopírování stojí další peníze, takže při návrhu datového skladu i těchto pump jsem s tímto musel počítat. Nakonec jako nejrychlejší a nejlevnější řešení nakonec vyšlo to, že se vytvořila pro každou zemi pumpa zvlášť, což také vidíte i na obrázku.

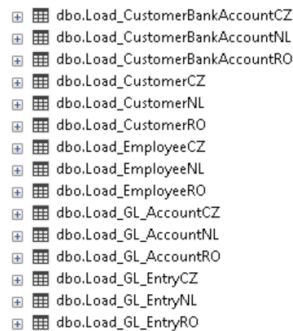


Obrázek 21: Úvodní obrazovka nástroje Azure Data Factory (26)

### 3.6.2 Výběr a přesun tabulek z Navisionu

V této podkapitole na následujících bodech popíšu a znázorním pomocí fotek kroky, které vedly k vytvoření spolehlivých pump.

1. Vytvoření struktury v mém datovém skladu. Na obrázku můžete vidět ukázkou z toho, jak pro tyto tři země vypadá přípravné místo, kam budou data importována.

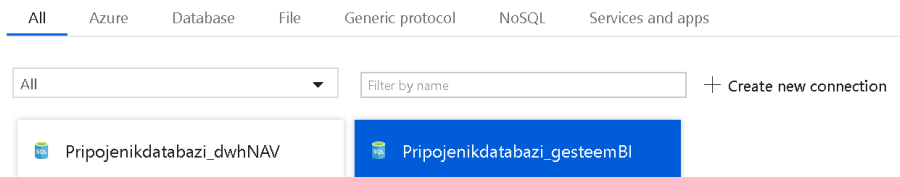


Obrázek 22: Přípravné tabulky pro každou zemi (autor)

2. Dalším krokem je vybrat zdrojovou databázi a z ní následně všechny tabulky, které budu chtít přenést do vlastního datového skladu.

### Source data store

Specify the source data store for the copy task. You can use an existing data store connection or specify a new data store.



Obrázek 23: Výběr zdrojů v Data Factory (26)

3. Tento krok opakuji i pro cílovou databázi s tím, že navíc zde musím namapovat vybrané tabulky a jejich sloupce. Zde je nutné dodržet stejné pojmenování sloupců jako je na zdroji, jinak dochází k problémům v mapování.



## Column mapping

Choose how source and destination columns are mapped

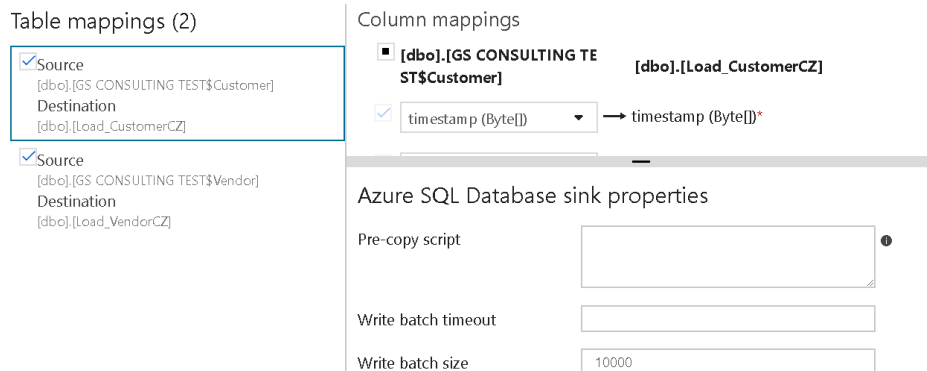


Table mappings (2)

- Source [dbo].[GS CONSULTING TEST\$Customer] Destination [dbo].[Load\_CustomerCZ]
- Source [dbo].[GS CONSULTING TEST\$Vendor] Destination [dbo].[Load\_VendorCZ]

Column mappings

- [dbo].[GS CONSULTING TEST\$Customer] [dbo].[Load\_CustomerCZ]
- timestamp (Byte[]) → timestamp (Byte[])\*

Azure SQL Database sink properties

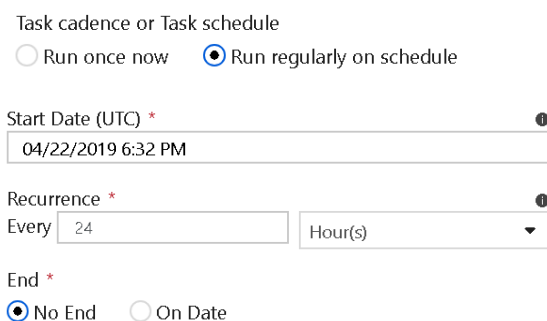
Pre-copy script

Write batch timeout

Write batch size

Obrázek 24: Mapování sloupců uvnitř Data Factory (26)

4. Posledním krokem v přenosu bylo nastavit automatizaci těchto pump. Ze začátku jsem prováděl kopii pouze jednou, aby při chybě ve skladu mi do něj nechodily nepřetržitě data. Bohužel už na vytvořenou pumpu, nelze aplikovat automatizace jinak než kódem v JSONu. Nástroj má také nevýhodu v tom, že vám umožní přepisovat/upravovat kód, ale na většinu změn reaguje negativně a nepovolí vám je aplikovat. Jediné řešení bylo tedy všechny pumpy udělat od znovu už s automatizací. Nastavuje se zde první začátek a také kdy se má provést naposledy. Umožňuje i možnost „no end“, kterou jsme využili. Dále se nastavil 24hodinový interval.



Task cadence or Task schedule

Run once now  Run regularly on schedule

Start Date (UTC) \*

Recurrence \*  Hour(s)

End \*  No End  On Date

Obrázek 25: Nastavení periodického importu (26)

### 3.7 Procedury

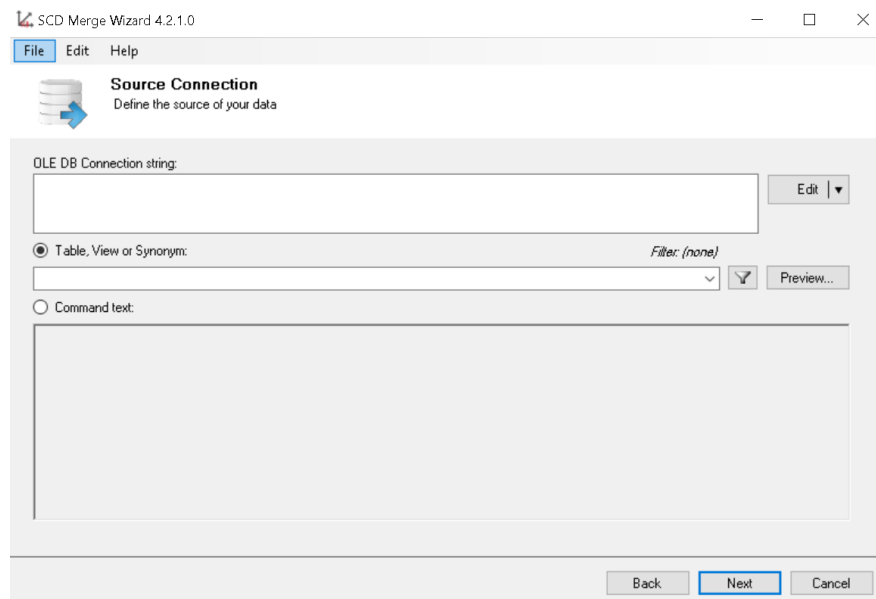
Tato kapitola bude obsahovat popis veškerých uložených procedur včetně programu, kde popíši postup, jakým způsobem jsem řešil aktualizaci dat v dimenzionálních tabulkách.

#### 3.7.1 Merge Wizardu

U dat, která vkládám do skladu ze zdrojů potřebuji, aby byla jednoznačná, nevznikaly duplicity a aby každý sloupek v dimenzi věděl, co udělat při změně v již zapsaných datech. K této aktualizaci dat jsem využil SQL příkaz merge. Pro zjednodušení vytváření SQL skriptů jsem využil program nazvaný „SCD-MergeWizard 4.2.1.0“, který mi pomohl v rozdělení SCD typů (Slowly-changing-dimension) každé sloupku ve všech dimenzích.

V krocích níže popíši přesný postup při tvorbě procedur merge na aktualizace dimenzionálních tabulek:

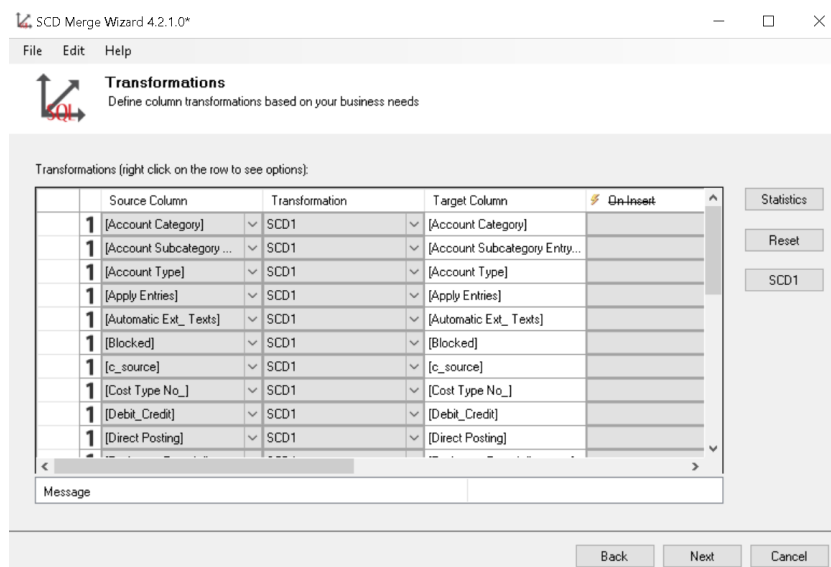
1. Celý program je naprogramovaný C#, tudíž lze spustit ve Microsoft Visual Studiu. Po spuštění se zobrazí okno, ve kterém si můžeme vybrat jaké typy SCD chceme, aby program mohl používat.
2. Při dalším postupu nám program nabídne připojení na zdrojový server pomocí „connection string“ a možnost výběru tabulky. Tabulku můžeme vybrat za pomoci rozklikávacího menu nebo pomocí vlastního kódu. Vlastní kód se použil pouze ve dvou případech – u dimenze zákazníka a dodavatele, kdy jsme obě tabulky propojovali s jejich účty. Obrovskou nevýhodou, kterou jsem zjistil ze začátku je v tom, že v kódu neobsahuje žádné „záchytné body“ a nastává tak spadnutí celého programu při jakémkoliv překlepu nebo špatně zadaném údaji. Program nám nabídne dále to samé pro cíl a tam postupujeme úplně stejně jako zde.



Obrázek 26: Program SCD MergeWizard 4.2.1.0 (autor)

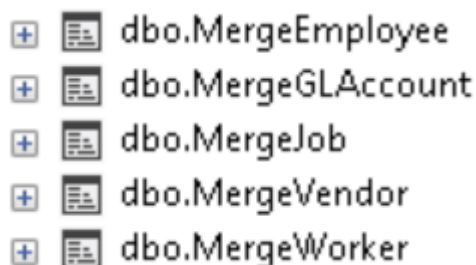
3. Nyní mě program přesunul na určení transformací SCD typů pro jednotlivé sloupce. Mapování proběhne v pořádku, pokud jsou v obou tabulkách (zdrojové a cílové) tytéž sloupce a žádný není vícekrát v případě, kdy propojujeme více tabulek dohromady. Vybírám zde poté business klíč, který je zde velmi důležitý. Je nutné, aby business klíč byl složený kvůli tomu, jelikož vkládáme data do jedné dimenze ze tří zdrojů stejného systému. Problémem je že systém v každé zemi přiděluje stejná identifikační čísla a procedura importu by nedokázala vložit identická identifikační čísla do jedné dimenze. Vyřešilo se to složeným business klíčem z identifikačního čísla a zemí původu databáze, kterou jsem jinou procedurou vždy při importu přiřadil. Jako poslední se musela vyřešit sloupečky validity, kterou merge také řeší. Do dvou sloupců „Valid From“ a „Valid To“, se zapisují datумы na základě různých akce s daty.

- **Import** – Valid From – CurrentDateTime, Valid To – MaxDateTime
- **Update** – ValidFrom – CurrentDateTime, ValidTo – MaxDateTime
- **Delete** – ValidTo – CurrentDateTime



Obrázek 27: Transformace v nástroji SCD Merge Wizard 4.2.1.0 (autor)

4. Dále mi program nabídl kontrolu duplicit předtím, než se vygeneruje kód. Tento kód poté vložíme do Microsoft SQL Server Management Studia, kde z toho vytvoříme proceduru, kterou následně spustíme. Toto provedeme pro každou dimenzi, která není fixní a přidávají se do ní nová data.

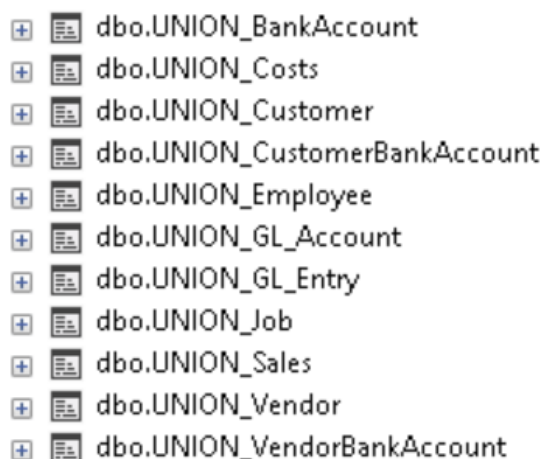


Obrázek 28: Procedury Merge v datovém skladu (autor)

### 3.7.2 Popis jednotlivých procedur

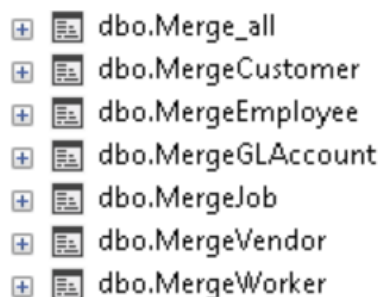
Uvnitř datového skladu se nachází kromě merge několik další důležitých procedur, které v této podkapitole popíši a uvedu k čemu jsou.

- UNION procedury – každá země má vlastní přípravnou tabulku, a právě pomocí operátoru „union“ je možné všechny přípravné tabulky spojit dohromady do jedné. Uvnitř těchto procedur se navíc ke každému zdroji přidává i země z jaké databáze data pochází, aby je bylo možné rozlišit databáze a nevznikaly duplicity. Abych nemusel spouštět proceduru pro každý stage zvlášť, vytvořil jsem pomocnou proceduru, která pouští všechny tyto procedury zároveň – „Exec\_Union“.



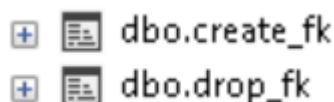
Obrázek 29: Procedury Union v datovém skladu (autor)

- MERGE procedury – Tyto procedury se vytvořily, kvůli aktualizaci dimenzionálních tabulek. Celý proces vytváření procedur je popsán v kapitole výše. Důležitost takovýchto procedur v datovém skladu je velká. Důvody k zavedení jsou takové, že by se neprováděla historizace dat a docházelo by k zatěžování systému, kdy by se musely pokaždé přenášet kompletní data. Zde jsem také vytvořil stejnou pomocnou proceduru, pro celkový merge – „Merge-all“.



Obrázek 30: Kompletní Merge procedury v datovém skladu (autor)

- CREATE/DROP cizí klíče – během tvorby merge jsem zjistil, že pokud jsou faktové tabulky navázané pomocí cizích klíčů z dimenzí, nelze spustit procedury merge aktualizovat data uvnitř dimenzí. Z toho důvodu jsou vytvořeny tyto dvě procedury, které umožní „shodit“ klíče, načíst data a pak znovu klíče vytvořit.



Obrázek 31: Vytvořené procedury pro cizí klíče (autor)

- INSERT procedura – tato procedura slouží k nahrání aktuálních dat do faktových tabulek na základě několika podmínek. Celý kód i podmínky popíší v následující kapitole.
- MASTER procedura – celý datový sklad běží na cloudu, kde oproti běžné databázi se složitěji specifikuje „agent“. Tato funkce umožňuje, aby v pravidelném intervalu udělal určité interakce, které mu předtím nastavíme. Onen problém jsem obešel pomocí Azure, kdy jsem si předpřipravil v datovém skladě proceduru, která spouštěla všechny již vytvořené procedury. V Azure se pouze upravily načasované importy, ke kterým se po dokončení přiřadila tato nově vytvořená procedura a tímto se zajistila celková automatizace plnění datového skladu.

```

CREATE PROCEDURE master_proc
AS
BEGIN
    EXEC Exec_UNION
    EXEC drop_fk
    EXEC Merge_all
    EXEC create_fk
    EXEC insert_fact
END;

```

Obrázek 32: Kód master procedury (autor)



Obrázek 33: Navázání master procedury na periodický proces plnění (autor)

### 3.8 Plnění faktových tabulek

Jako jeden z posledních bodů, který mi zůstal na projektovém pláň bylo umožnit načíst data z dimenzí a stageové části do faktových tabulek na základě podmínek, které jsme si museli předem definovat.

#### Podmínky pro insert:

##### **FactSales**

- Obsahuje pouze dokumenty, které se zaúčtovali na 6kové účty (výnosy).
- Insertovat se budou pouze data, která budou mít datum „vyšší“, jak datum posledního insertu v sloupečku „ValidFrom“.
- Musí se zde počítat i s nekompletními daty, které nahradíme -1, při nenalezení v dané dimenzi.

##### **FactCosts**

- Obsahuje pouze dokumenty, které se zaúčtovali na 5kové účty (náklady).
- Insertovat se budou pouze data, která budou mít datum „vyšší“, jak datum posledního insertu v sloupečku „ValidFrom“.
- Musí se zde počítat i s nekompletními daty, které nahradíme -1, při nenalezení v dané dimenzi.

#### **3.8.1 Deklarace proměnných**

Nad hlavní procedurou importu se nachází dvě proměnné, pro které jsem musel zjistit hodnoty, kvůli načítání dat do obou faktových tabulek. Proměnné se deklarují jako celé číslo, do kterých se přiřadí vždy maximální hodnota datumu řádku ve faktové tabulce. Každou z nich poté použijeme v sekci „WHERE“ u insertu. To umožní vybírat pouze data, která do faktových tabulek nebyla vložena. Řeší to tedy, jak nový import nebo i vynechání importu.

```

declare @maxS int
declare @maxC int

:select @maxS = max(FactSales.[Posting_date])
from FactSales

:select @maxC = max(FactCosts.[Posting_date])
from FactCosts

```

Obrázek 34: Deklarace proměnných pro proceduru (autor)

### 3.8.2 Kód FactSales

Kvůli čitelnosti a přehlednosti jsem kód rozdělil do dvou částí. V první části popíšu strukturu vybraných sloupců do faktových tabulek a ve druhé zase propojení na dimenzionální tabulky společně s podmínkami výběru insertu.

Data se vybíraly z přípravné tabulky StageSales. Faktová tabulka začíná datumovým údajem, která značí den, kdy byl dokument vystaven. Následuje sloupec, ve kterém jsou rozvahové účty, na které byl dokument zaúčtován. Vedle něj je sloupec zákazníka s podmínkou, že pokud by se nenacházel v dimenzionální tabulce, tak se nahradí -1, která znamená neznámý údaj. Tato podmínka je zde i v dalších sloupcích kvůli tomu, jelikož sloupce v obou faktových tabulkách jsou nastaveny na „NOT NULL“, tzn. že nesmějí být prázdné. Další je zde nachystaný sloupec na pracovníka, který je prozatím staticky vyplněný -1, jelikož datové zdroje pro zaměstnance/externisty nejsou v požadovaném tvaru. Potom se zde nachází identifikační číslo měny, faktury, řádek faktury, země pobočky odkud údaj přišel, typ práce, za kterou se daná částka vyfakturovala a jejich třídění do produkčních skupin. Nyní se ve faktové tabulce dostáváme k cenám a množstvím. Většina sloupců je už v detailu daného dokumentu vypočítaná, což mi ulehčuje práci. Výpočty probíhají pouze u sloupců, kdy cenu včetně daně převádíme do měny dané země a celkově pak na eura. Zbylé sloupce se doplňují pomocí podmínek CASE. První dvě podmínky CASE jsou zde kvůli špatné kvalitě dat, kdy se u některých dokumentů nevyplňují všechny údaje. Další podmínku CASE jsem vytvořil, aby se zjistilo datum splatnosti u zákazníků, které jsou zapsány v dimenzi. Zjištěné údaje poté podmínka přidá k datumu vystavení faktury a tím je nastaveno datum splatnosti.



V druhé části se nachází jednotlivá propojení na tabulky dimenzí. Všechny propojení jsou na základě identifikačních čísel každé dimenze. Navíc tu jsou u některých sloupců podmínky, které nám zajišťují, aby se při importu vybrala vždy správná verze daného údaje, pokud se v dimenzi nachází např. zákazník víckrát, kvůli tomu, že se stěhoval. Tato podmínka je také důležitá u kurzů, kterou budeme potřebovat v budoucnu až budeme nahrávat denní kurzy, abychom mohli převést hodnoty u dokumentů vždy k danému dni. Celkově je zde 4x JOIN a 3x LEFT JOIN. První samostatné JOINy nám umožňují filtraci řádků, které neobsahují data pro všechny sloupečky. Zbylé LEFT JOINy jsou zde kvůli špatné kvalitě zdrojových dat a vyfiltrovali bychom si kompletně všechny údaje. Podmínky ke konci insertu jsou dvě. První vybírá pouze dokumenty, které byly zaúčtovány na rozvahové účty, které začínají na číslo 6, což vyznačuje výnosy. Druhá podmínka zajišťuje při opakovaném importu, aby se insertovala pouze data, která ve faktové tabulce ještě nejsou a nedocházelo k duplikacím.

## Kód pro plnění FactSales:

```
INSERT INTO FactSales
SELECT DimDate.Date_id
, DimGLAccount.GLAccountID
, COALESCE(DimCustomer.Customer_id,-1) AS Customer_id
, -1 AS WorkerID
, DimCurrencyRate.Currency_id
, SS.[Document No_], SS.[Line No_]
, DimTerritory.Territory_id
, COALESCE(DimJob.Job_id,-1) AS Job_id
, SS.[Gen_ Prod_ Posting Group]
, CASE
  WHEN SS.[Quantity] = '1' THEN 'Project / Projekt'
  WHEN SS.[Unit of Measure] = '' THEN 'Other / Ostatní'
  ELSE SS.[Unit of Measure]
END AS [Unit of Measure]
, SS.[Quantity],SS.[Unit Price]
, SS.Amount, SS.[VAT _], SS.[Amount Including VAT]
, DimCurrencyRate.Currency_id AS Base_Currency_Code
, SS.[Amount Including VAT]* DimCurrencyRate.Value AS Amount_In_Base
, (SS.[Amount Including VAT]/CurrEUR.Value) AS 'Amount_In_EURO'
, CASE
  WHEN DimCustomer.[Payment Terms Code] = '' THEN 'none'
  WHEN DimCustomer.[Payment Terms Code] IS NULL THEN 'none'
  ELSE DimCustomer.[Payment Terms Code]
END AS Payment_Terms_Code
, CASE
  WHEN DimCustomer.[Payment Terms Code] = '' THEN SS.[Posting Date]
  WHEN DimCustomer.[Payment Terms Code] = '10D' THEN DATEADD(day,10,SS.[Posting Date])
  WHEN DimCustomer.[Payment Terms Code] = '14D' THEN DATEADD(day,14,SS.[Posting Date])
  WHEN DimCustomer.[Payment Terms Code] = '21D' THEN DATEADD(day,21,SS.[Posting Date])
  WHEN DimCustomer.[Payment Terms Code] = '30D' THEN DATEADD(day,30,SS.[Posting Date])
  WHEN DimCustomer.[Payment Terms Code] = '60D' THEN DATEADD(day,60,SS.[Posting Date])
  WHEN DimCustomer.[Payment Terms Code] = '90D' THEN DATEADD(day,90,SS.[Posting Date])
  WHEN DimCustomer.[Payment Terms Code] = '1M' THEN DATEADD(month,1,SS.[Posting Date])
  WHEN DimCustomer.[Payment Terms Code] = 'EOM' THEN EOMONTH(SS.[Posting Date])
  WHEN DimCustomer.[Payment Terms Code] IS NULL THEN DATEADD(day,30,SS.[Posting Date])
END AS Effective_Date
FROM Stage_Sales SS

JOIN DimDate ON ss.[Posting Date] = DimDate.DateValue
JOIN DimGLAccount ON ss.[No_] = DimGLAccount.No_
JOIN DimCustomer ON ss.[Sell-to Customer No_] = DimCustomer.No_
AND ss.[Posting Date] BETWEEN DimCustomer.ValidFrom and DimCustomer.ValidTo
JOIN DimTerritory ON DimTerritory.Territory_id = CASE
  WHEN ss.c_source LIKE 'NL' THEN 1
  WHEN ss.c_source LIKE 'CZ' THEN 2
  WHEN ss.c_source LIKE 'RO' THEN 3
END

LEFT JOIN DimJob ON ss.[Job No_] = DimJob.No_
LEFT JOIN DimCurrencyRate CurrEUR ON CurrEUR.Currency_id = 1
--AND SS.[Posting Date] BETWEEN DimCurrencyRate.ValidFrom AND DimCurrencyRate.ValidTo
LEFT JOIN DimCurrencyRate ON DimCurrencyRate.Currency_id = Territory_id
AND SS.[Posting Date] BETWEEN DimCurrencyRate.ValidFrom AND DimCurrencyRate.ValidTo
WHERE SS.[No_] LIKE '6%' AND @maxS >= SS.[Posting date]
```

Obrázek 35: Kód procedury pro insert FactSales (autor)

### 3.8.2 Kód FactCosts

Struktura tabulky i propojení s dimenzemi jsou totožné s předchozí. Rozdílná je pouze ve zdrojové tabulce, kdy se data berou ze StageCosts. Podmínka pro výběr dokumentů je zde taková, že se musí zaúčtovat na rozvahové účty začínající číslem 5, což vyznačuje náklady.

Kód pro plnění FactCosts:

```
INSERT INTO FactCosts
SELECT DimDate.Date_id
, DimGLAccount.GLAccountID
, COALESCE(DimVendor.Vendor_id,-1) AS VendorID
,-1 AS WorkerID, DimCurrencyRate.Currency_id
, SC.[Document No_]
, SC.[Line No_]
, DimTerritory.Territory_id
, COALESCE(DimJob.Job_id,-1) AS Job_id
, SC.[Gen_Prod_Posting Group]
, CASE
  WHEN SC.[Quantity] = '1' THEN 'Project / Projekt'
  WHEN SC.[Unit of Measure] = '' THEN 'Other / Ostatní'
  ELSE SC.[Unit of Measure]
  END AS [Unit of Measure]
, SC.[Quantity]
, SC.[Direct Unit Cost]
, SC.Amount, SC.[VAT _]
, SC.[Amount Including VAT]
, DimCurrencyRate.Currency_id as Base_Currency_Code
, SC.[Amount Including VAT]* DimCurrencyRate.Currency_id AS 'Amount_In_Base'
, (SC.[Amount Including VAT]/CurrEUR.Value) AS 'Amount_In_EURO'
, CASE
  WHEN DimVendor.[Payment Terms Code] = '' THEN 'none'
  WHEN DimVendor.[Payment Terms Code] IS NULL THEN 'none'
  ELSE DimVendor.[Payment Terms Code]
  END as Payment_Terms_Code
, CASE
  WHEN DimVendor.[Payment Terms Code] = '' THEN SC.[Posting Date]
  WHEN DimVendor.[Payment Terms Code] = '100' THEN DATEADD(dsy,10,SC.[Posting Date])
  WHEN DimVendor.[Payment Terms Code] = '140' THEN DATEADD(dsy,14,SC.[Posting Date])
  WHEN DimVendor.[Payment Terms Code] = '210' THEN DATEADD(dsy,21,SC.[Posting Date])
  WHEN DimVendor.[Payment Terms Code] = '300' THEN DATEADD(dsy,30,SC.[Posting Date])
  WHEN DimVendor.[Payment Terms Code] = '600' THEN DATEADD(dsy,60,SC.[Posting Date])
  WHEN DimVendor.[Payment Terms Code] = '900' THEN DATEADD(dsy,90,SC.[Posting Date])
  WHEN DimVendor.[Payment Terms Code] = '1M' THEN DATEADD(month,1,SC.[Posting Date])
  WHEN DimVendor.[Payment Terms Code] = 'EOM' THEN EOMONTH(SC.[Posting Date])
  WHEN DimVendor.[Payment Terms Code] IS NULL THEN DATEADD(day,30,SC.[Posting Date])
  END as Effective_Date
FROM Stage_Costs SC

JOIN DimDate ON SC.[Posting Date] = DimDate.DateValue
JOIN DimGLAccount ON SC.[No_] = DimGLAccount.No_
LEFT JOIN DimVendor ON SC.[Buy-from Vendor No_] = DimVendor.No_
AND SC.[Posting Date] between DimVendor.ValidFrom AND DimVendor.ValidTo
JOIN DimTerritory ON DimTerritory.Territory_id = CASE
  WHEN SC.c_source LIKE 'NL' THEN 1
  WHEN SC.c_source LIKE 'CZ' THEN 2
  WHEN SC.c_source LIKE 'RO' THEN 3
END
LEFT JOIN DimJob ON SC.[Job No_] = DimJob.No_
LEFT JOIN DimCurrencyRate CurrEUR ON CurrEUR.Currency_id = 1
--AND SC.[Posting Date] BETWEEN DimCurrencyRate.ValidFrom AND DimCurrencyRate.ValidTo
LEFT JOIN DimCurrencyRate ON DimCurrencyRate.Currency_id = Territory_id
AND SC.[Posting Date] BETWEEN DimCurrencyRate.ValidFrom AND DimCurrencyRate.ValidTo
WHERE SC.[No_] LIKE '5%' AND @maxC >= SC.[Posting Date]
```

Obrázek 36: Kód procedury pro insert FactCosts (autor)

### 3.9 Možnosti pro budoucí rozvoj datového skladu

Po dokončení mé části pro bakalářskou práci jsem ve firmě s vedením konzultoval možnosti dalšího rozšíření, které by firma uvítala. Jako první, na čem jsme se shodli, je zlepšení kvality zapisovaných dat. Především u dokumentů, kde není zaznamenáván detailní pohled, což nám ve výsledku neumožňuje propojení na některé dimenze. Také chybí spousta návazností, které je nutné v ekonomickém systému vytvořit jako např. produkty/předměty, které firma vlastní nebo doplnění jednotlivých prací, které zaměstnanci vykonávají během projektu.

Dalším bod rozvoje se týká také doplnění. Zde se zvažovaly dva možné postupy. Jednodušší postup by byl, aby přímo do popisu faktury se zapisovalo jméno daného zaměstnance, který na projektu pracoval – přesněji jeho identifikačního číslo přidělené systémem. Druhý složitější způsob by byl ten, že bychom museli využít a nastavit službu strojového učení, které by v textu rozeznávalo jména zaměstnanců. K těmto dvěma postupům by bylo nutné také propojit aplikaci, která je určena pro získávání přesných odpracovaných hodin každého zaměstnance. Aplikace by byla propojená přímo se systémem a daný počet hodin by se automaticky přidělil k projektu. Tímto by se pak mohla v reportingu zjistit efektivita každého zaměstnance či externisty.

Zatím posledním bodem, o kterém vím je propojení banky s datovým skladem. Zde bychom dostávaly informace o tom, zda nám určitá platba na účet přišla, měla zpoždění nebo vůbec nepřišla. Zjišťovali bychom tímto celkovou likviditu svých zákazníků. Z celkového pohledu bychom zde navíc mohli získávat historické údaje o finančních prostředcích na bankovním účtu.

## ZÁVĚR

Cílem mé bakalářské práce bylo vytvořit agilní datový sklad, který umožní historizaci dat, aktualizaci údajů a také bude připraven na implementaci jakýkoliv požadavku ze strany zákazníka, které mohou přijít v budoucnu. Pro začátek byla databáze vytvořená na cloudové platformě v nástroji Azure od Microsoftu. V této databázi se připravila struktura datového skladu, pro který byla identifikována zdrojová data z externích zdrojů. Následně se pomocí softwaru vygenerovaly procedury, které umožňují aktualizaci dimenzionálních tabulek a tím také zajišťují verzování jednotlivých údajů. (historizaci). Poté se vytvořily datové pumpy v Microsoft Azure v sekci Data Factory, které umožňují periodicky plnit datový sklad. Vytvořením procedur UNION jsem docílil sjednocení dat ze všech různých poboček do jednoho stage. Dále se vytvořila insert procedura pro obě faktové tabulky, které doplňujeme na základě předefinovaných podmínek firmy. Úplně nakonec se vytvořila master procedura, která spouštěla všechny ostatní procedury. Ta se potom přidala do nástroje Azure, přesněji k datovým pumpám, které měly nastaveny časování a tímto se vyřešilo i časování zbylých akcí uvnitř datového skladu. Tímto se splnily všechny dílčí cíle, které byly předem stanoveny.

Na úplný závěr mohu napsat, že vytvořené řešení má reálné uplatnění, jelikož se nad tímto skladem vytvořilo kompletní analytické řešení a také firma počítá s mnoha budoucími rozšířeními, které následně jako kompletní produkt, mohou nabízet svým novým zákazníkům.

## SEZNAM POUŽITÝCH ZDROJŮ

- (1) What is database (DB)? - Definition from WhatIs.com. SQL Server: Covering today's SQL Server topics [online]. Dostupné z: <https://searchsqlserver.techtarget.com/definition/database>
- (2) Základní pojmy a funkce databáze – informatika | Studijni-svet.cz. Studijni-svet.cz | Studijní materiály do školy a k maturitě [online]. Dostupné z: <https://studijni-svet.cz/zakladni-pojmy-a-funkce-databaze-informatika/>
- (3) Databáze. Databáze [online]. Copyright © 2010 Misha [cit. 06.05.2019]. Dostupné z: <http://www.databaze.chytrak.cz/index.htm>
- (4) LACKO, Ľuboslav. Business Intelligence v SQL Serveru 2008: reportovací, analytické a další datové služby. Brno: Computer Press, 2009. ISBN 978-80-251-2887-9.
- (5) GÁLA, Libor, Jan POUR a Prokop TOMAN. Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi, technologie informačních systémů, řízení a rozvoj podnikové informatiky. Praha: Grada, 2006. 1. vyd. 482 s. ISBN 80-247-1278-4.
- (6) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2., přeprac. a aktualiz. vyd. Praha, 2009, 496 s. Expert (Grada). ISBN 978-80-247-2615-1.
- (7) KONVIČKA, Lukáš. *Potřebuje vaše firma business intelligence?* [online] Dostupné z: <https://www.systemonline.cz/business-intelligence/potrebuje-vase-firma-business-intelligence.html>
- (8) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi. 3., aktualizované vydání. Praha: Grada Publishing, 2015, 240 stran : ilustrace, portréty. ISBN 978-80-247-5457-4.

- (9) Display e-study documents. University information system MENDELU [online].  
Dostupné z: <https://is.mendelu.cz/eknihovna/opory/index.pl?cast=5145>
- (10) TVRDÍKOVÁ, Milena. *EIS je nezbytnou součástí BI* [online]. Dostupné z:  
<https://www.systemonline.cz/clanky/eis-nezbytna-soucast-business-intelligence.html>
- (11) LACKO, Ľuboslav. *Databáze: datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle*. Brno: Computer Press, 2003. ISBN 80-7226-969-0.
- (12) Operational Data Store – Adastra. Adastra – Úvodní stránka [online]. Copyright © Adastra. Dostupné z: <https://www.adastra.cz/ict-reseni/ods>
- (13) INMON, William H. ODS Types. In: Information Management [online].  
Dostupné z: <http://www.information-management.com/issues/20000101/1749-1.html>
- (14) Datový sklad – dwh [online]. Dostupné z:  
<https://www.itbiz.cz/slovník/informacni-technologie-it/datovy-sklad-dwh>
- (15) Fakta a dimenze - Tabulky v datovém skladu | BI Portál. Business Intelligence portál [online]. Copyright © VŠECHNA PRÁVA VYHRAZENA. Dostupné z:  
<https://biportal.cz/fakta-dimenze-tabulky-v-datovem-skladu/>
- (16) Star and snowflake in data warehouse [online]. Dostupné z:  
<https://www.guru99.com/star-snowflake-data-warehousing.html>
- (17) Types Of Facts And Dimensions In Data Warehouse. Online Certification Training | Corporate Training - Mindmajix [online]. Copyright © 2019 Mindmajix Technologies Inc. All Rights Reserved. Dostupné z: <https://mindmajix.com/data-warehousing/types-of-tables-in-dwh-facts-and-dimensions>

- (18) LARSON, Brian. Delivering business intelligence with Microsoft SQL server 2012. 3rd ed. New York: McGraw-Hill, 2012. ISBN 0071759387.
- (19) NOVOTNÝ, Ota, Jan POUR a David SLÁNSKÝ. Business intelligence: jak využít bohatství ve vašich datech. Praha: Grada, 2005, 254 s. ISBN 80-247-1094-3.
- (20) IBM Knowledge Center. [online]. Copyright © Copyright IBM Corp. 2017. Dostupné z: [https://www.ibm.com/support/knowledgecenter/en/SSGU8G\\_12.1.0/com.ibm.whse.doc/ids\\_ddi\\_350.html](https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.whse.doc/ids_ddi_350.html)
- (21) What is Business Intelligence Reporting (BI Reporting)? - Definition from Techopedia. Techopedia - Where Information Technology and Business Meet [online]. Copyright © 2019 Techopedia Inc. Dostupné z: <https://www.techopedia.com/definition/30217/business-intelligence-reporting-bi-reporting>
- (22) Data Mining Definition. Investopedia - Sharper Insight. Smarter Investing. [online]. Dostupné z: <https://www.investopedia.com/terms/d/datamining.asp>
- (23) BEN-GAN, Itzik, Dejan SARKA a Ron TALMAGE. Querying Microsoft SQL Server 2012: exam 70-461 training kit. Sebastopol, Calif.: Microsoft, 2012. ISBN 978-0-7356-6605-4.
- (24) LEONARD, Andy. SQL Server 2012 integration services design patterns. New York: Distributed to the book trade worldwide by Springer Science+Business Media, 2012, 444 s. ISBN 1430237716.
- (25) DynamicsDocs.com. DynamicsDocs.com [online]. Copyright ©. Dostupné z: <https://dynamicsdocs.com/nav/2018/w1>
- (26) Azure Data Factory. Azure Data Factory [online]. Dostupné z: <https://adf.azure.com>



## SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

**BI** – Business Intelligence

**DB** – Database, databáze

**SQL** – Structured Query Language, strukturovaný dotazovací jazyk

**SŘBD** – Systém Řízení Báze Dat, databázový systém

**ERP** – Enterprise Resource Planning, plánování podnikových zdrojů

**CRM** – Customer Relationship Management, řízení vztahů se zákazníky

**SCM** – Supply Chain Management, správa dodavatelského řetězce

**ETL** – Export, Transact, Load

**OLAP** – Online Analytical Processing, technologie uložení dat v DB

**EIS** – Executive Information Systems

**DSA** – Data Staging Area, dočasné uložení dat

**ODS** – Operational Data Store, operační uložení dat

**T-SQL** – Transact-SQL, rozšířený strukturovaný dotazovací jazyk

**NAV** – Navision, ERP systém od Microsoftu

**DWH** – Data Warehouse, datový sklad

**SCD** – Slowly Changing Dimensions, umožňuje dimenzím historizaci dat

**JSON** – JavaScript Object Notation, odlehčený formát pro výměnu dat

## SEZNAM OBRÁZKŮ

Obrázek 1: Hierarchický model (3) .....	13
Obrázek 2: Síťový model dat (3) .....	14
Obrázek 3: Relační model dat (3) .....	14
Obrázek 4: Hierarchie Informační úrovně (4, str.15) .....	15
Obrázek 5: Obecná koncepce architektury BI (7, str.27) .....	16
Obrázek 6: Princip ETL (4, str.115) .....	18
Obrázek 7: Procesní schéma datového skladu (4, str.41) .....	21
Obrázek 8: Schéma hvězdy (18, str.24) .....	23
Obrázek 9: Schéma sněhové vločky (18, str.25) .....	23
Obrázek 10: Princip multidimenzionální databáze na bázi OLAP (19) .....	24
Obrázek 11: Hierarchická struktura firmy Gesteem a.s. (autor) .....	30
Obrázek 12: Hierarchické znázornění funkcí ve firmě Gesteem a.s. (autor) .....	31
Obrázek 13: Ukázka funkčnosti a možností firemní aplikace (autor) .....	33
Obrázek 14: Úvodní obrazovka ve firemní aplikaci (autor) .....	33
Obrázek 15: Ukázka excelovského sešitu používaný pro analýzu bankovního účtu firmy (autor) .....	35
Obrázek 16: Ukázka excelovského sešitu používaný pro analýzu všech rozvahových účtů firmy (autor) .....	35
Obrázek 17: Ukázka webu dynamicsdocs.com (23) .....	37
Obrázek 18: Schéma postupu při vytváření datového skladu (autor) .....	40
Obrázek 19: ER diagram vytvořeného datového skladu (autor) .....	43
Obrázek 20: Hierarchie dimenze datumu (autor) .....	45
Obrázek 21: Úvodní obrazovka nástroje Azure Data Factory (24) .....	47
Obrázek 22: Přípravné tabulky pro každou zemi (autor) .....	48
Obrázek 23: Vyběr zdrojů v Data Factory (24) .....	48
Obrázek 24: Mapování sloupců uvnitř Data Factory (24) .....	49
Obrázek 25: Nastavení periodického importu (24) .....	49
Obrázek 26: Program SCD MergeWizard 4.2.1.0 (autor) .....	51
Obrázek 27: Transformace v nástroji SCD Merge Wizard 4.2.1.0 .....	52
Obrázek 28: Procedury Merge v datovém skladu (autor) .....	52
Obrázek 29: Procedury Union v datovém skladu (autor) .....	53
Obrázek 30: Kompletní Merge procedury v datovém skladu (autor) .....	53

Obrázek 31: Vytvořené procedury pro cizí klíče (autor).....	54
Obrázek 32: Kód master procedury .....	54
Obrázek 33: Navázání master procedury na periodický proces plnění (autor).....	55
Obrázek 34: Deklarace proměnných pro proceduru (autor) .....	56
Obrázek 35: Kód procedury pro insert FactSales (autor) .....	58
Obrázek 36: Kód procedury pro insert FactCosts (autor).....	59

## **SEZNAM TABULEK**

Tabulka 1: Rozdíly mezi fakty a dimenzemi (14) .....	22
Tabulka 2: Rozdíl mezi schématy star a snowflake (15) .....	22

## SEZNAM GRAFŮ

## SEZNAM PŘÍLOH

Příloha 1: Struktura DimCurrencyRate.....	i
Příloha 2: Struktura DimCustomer .....	ii
Příloha 3: Struktura DimDate .....	iv
Příloha 4: Struktura DimEmployee.....	v
Příloha 5: Struktura DimGLAccount.....	vii
Příloha 6: Struktura DimJob .....	viii
Příloha 7: Struktura DimTerritory .....	ix
Příloha 8: Struktura DimVendor.....	x
Příloha 9: Struktura DimWorker.....	xi
Příloha 10: Struktura FactCosts .....	xii
Příloha 11: Struktura FactSales.....	xiii

Příloha 1: Struktura DimCurrencyRate

<b>Název sloupce</b>	<b>Datový typ</b>	<b>Not Null</b>	<b>SCD typ</b>	<b>Popis sloupce</b>
Currency_id	Int	Ano	-	Primární klíč
Code	Nvarchar(10)	Ano	-	Kód měny
Description	Nvarchar(50)	Ano	-	Popis měny
Value	Decimal(5,2)	Ano	-	Hodnota měny
ValidFrom	Date	Ne	-	Datum od
ValidTo	Date	Ne	-	Datum do

## Příloha 2: Struktura DimCustomer

Název sloupce	Datový typ	Not Null	SCD typ	Popis sloupce
Customer_id	Int	Ano	Business Klíč	Primární klíč
No_	Nvarchar(20)	Ano	SCD1	Číslo zákazníka
Name	Nvarchar(50)	Ano	SCD1	Jméno
Search Name	Nvarchar(50)	Ano	SCD1	Vyhledávací jméno
Address	Nvarchar(50)	Ano	SCD2	Adresa
City	Nvarchar(30)	Ano	SCD2	Město
Contact	Nvarchar(50)	Ano	SCD1	Kontakt
Territory Code	Nvarchar(10)	Ano	SCD2	Kód země
Global Dimension 1 Code	Nvarchar(20)	Ano	SCD1	Globální kód první dimenze
Currency Code	Nvarchar(10)	Ano	SCD1	Kód měny
Payment Terms Code	Nvarchar(10)	Ano	SCD1	Kód platebního termínu
Fin_Charge Terms Code	Nvarchar(10)	Ano	SCD1	Kód termínu pro finanční platby
Salesperson Code	Nvarchar(20)	Ano	SCD1	Kód zaměstnance
Shipment Method Code	Nvarchar(10)	Ano	SCD1	Kód metody doručení
Customer Disc_Group	Nvarchar(20)	Ano	SCD1	Skupiny podle slev pro zákazníky
Country_Region Code	Nvarchar(10)	Ano	SCD2	Kód pro región
Blocked	Int	Ano	SCD1	Blokovaný zákazník
Priority	Int	Ano	SCD1	Priorita zákazníka
Prices Including VAT	Tinyint	Ano	SCD1	Cena včetně daně
Location Code	Nvarchar(10)	Ano	SCD2	Kód lokace
VAT Registration No_	Nvarchar(20)	Ano	SCD2	IČO
Gen_Bus_Posting Group	Nvarchar(20)	Ano	SCD1	Obchodní skupiny
Country	Nvarchar(30)	Ano	SCD2	Kód země
Reserve	Int	Ano	SCD1	Rezerva
Partner Type	Int	Ano	SCD1	Typ partnerství
Privacy Blocked	Tinyint	Ano	SCD1	Blokování soukromí
Preffered Bank Account Code	Nvarchar(30)	Ano	SCD1	Kód preferovaného bankovní účtu
Bank Account Code	Nvarchar(30)	Ano	SCD1	Kód bankovního účtu



Bank Account Name	Nvarchar(50)	Ano	SCD1	Jméno banky
ValidFrom	Date	Ne	SCD2 ValidFrom	Datum od
ValidTo	Date	Ne	SCD2 ValidTo	Datum od
c_source	Nvarchar(10)	Ne	Business Klíč	Zdroj databáze

## Příloha 3: Struktura DimDate

Název sloupce	Datový typ	Not Null	SCD typ	Popis sloupce
Date_id	Int	Ano	-	Primární klíč
DateValue	Date	Ne	-	Datum
YearDayNumber	Int	Ne	-	Číslo dne v roce
QuaterDayNumber	Int	Ne	-	Číslo dne ve kvartálu
MonthDayNumber	Int	Ne	-	Číslo dne v měsíci
YearMonthNumber	Int	Ne	-	Číslo měsíce v roce
MonthDays	Int	Ne	-	Počet dní v měsíci
YearQuarterNumber	Int	Ne	-	Číslo kvartálu v roce
QuaterDays	Int	Ne	-	Počet dní v kvartálu
YearNumber	Int	Ne	-	Číslo roku
PreviousYear	Int	Ne	-	Předchozí rok
PreviousQuarter	Int	Ne	-	Předchozí kvartál
PreviousMonth	Int	Ne	-	Předchozí měsíc
WeekDayNumber	Int	Ne	-	Číslo dne v týdnu
DayName	Varchar(10)	Ne	-	Název dne
YearWeekNumber	Int	Ne	-	Číslo týdne v roce
MonthName	Varchar(8)	Ne	-	Název měsíce
WeekFirstDay	Date	Ne	-	První den týden
WeekLastDay	Date	Ne	-	Poslední den týdne
MonthFirstDay	Date	Ne	-	První den měsíce
MonthLastDay	Date	Ne	-	Poslední den měsíce
YearMonth	Varchar(8)	Ne	-	Datum první dne v měsíci v roce jako text
WeekFirstDay	Date	Ne	-	První den v týdnu
WeekLastDay	Date	Ne	-	Poslední den týdne
MonthFirstDay	Date	Ne	-	První den v měsíci
MonthLastDay	Date	Ne	-	Poslední den v měsíce
QuarterFirstDay	Date	Ne	-	První den v kvartálu
QuarterLastDay	Date	Ne	-	Poslední den v kvartálu
SalesHoliday	Int	Ano	-	Značí jestli je svátek
SalesHolidayName	Varchar(16)	Ano	-	Název svátku
ServiceHoliday	Int	Ano	-	Značí jestli je svátek
ServiceHolidayName	Varchar(16)	Ano	-	Název svátku

Příloha 4: Struktura DimEmployee

Název sloupce	Datový typ	Not Null	SCD typ	Popis sloupce
Employee_id	Int	Ano	Business Klíč	Primární klíč
No_	Nvarchar(20)	Ano	SCD1	Číslo zaměstnance
First Name	Nvarchar(30)	Ano	SCD1	Jméno
Last Name	Nvarchar(30)	Ano	SCD1	Příjmení
Initials	Nvarchar(30)	Ano	SCD1	Iniciály
Job Title	Nvarchar(30)	Ano	SCD1	Pracovní pozice
Search Name	Nvarchar(250)	Ano	SCD1	Vyhledávací jméno
Address	Nvarchar(50)	Ano	SCD2	Adresa
City	Nvarchar(30)	Ano	SCD2	Město
Post Code	Nvarchar(20)	Ano	SCD2	Směrovací číslo
Birth Date	Datetime	Ano	SCD1	Datum narození
Social Security No_	Nvarchar(30)	Ano	SCD1	Sociálního zabezpečení
Gender	Int	Ano	SCD1	Pohlaví
Manager No_	Nvarchar(20)	Ano	SCD1	Číslo vedoucího
Emplymt_ Contract Code	Nvarchar(10)	Ano	SCD1	Kód zaměstnanecké smlouvy
Statistics Group Code	Nvarchar(10)	Ano	SCD1	Kód statistické skupiny
Employment Date	Datetime	Ano	SCD1	Datum nástupu
Status	Int	Ano	SCD1	Status
Global Dimension 1 Code	Nvarchar(20)	Ano	SCD1	Kód globální 1 dimenze
Resource No_	Nvarchar(20)	Ano	SCD1	Číslo zdroje
Extension	Nvarchar(30)	Ano	SCD1	Rozšíření
Title	Nvarchar(20)	Ano	SCD1	Titul
Salespers__Purch_ Code	Nvarchar(20)	Ano	SCD1	Kód prodejce
Employee Posting Group	Nvarchar(20)	Ano	SCD1	Zaměstnanecké skupiny
Bank Account No_	Nvarchar(30)	Ano	SCD1	Číslo bankovního účtu
Privacy Blocked	Tinyint	Ano	SCD1	Zablokované soukromí
Cost Center Code	Nvarchar(20)	Ano	SCD1	Kód nákladového střediska
Line No_	Int	Ano	SCD1	Číslo řádku
Qualification Code	Nvarchar(10)	Ano	SCD1	Kód kvalifikace
Description	Nvarchar(50)	Ano	SCD1	Popis
Institution_Company	Nvarchar(50)	Ano	SCD1	Firma

ValidFrom	Date	Ne	SCD2 ValidFrom	Datum od
ValidTo	Int	Ne	SCD2 ValidTo	Datum do
c_source	Varchar(16)	Ne	Business Klíč	Zdrojová databáze

Příloha 5: Struktura DimGLAccount

<b>Název sloupce</b>	<b>Datový typ</b>	<b>Not Null</b>	<b>SCD typ</b>	<b>Popis sloupce</b>
GLAccountID	Int	Ano	Business Klíč	Primární klíč
No_	Nvarchar(20)	Ano	SCD1	Číslo rozvahového účtu
Name	Nvarchar(50)	Ano	SCD1	Název účtu
Account Type	Int	Ano	SCD1	Typ účtu
Account Category	Int	Ano	SCD1	Kategorie účtu
Income_Balance	Int	Ano	SCD1	Balance příjmů
Debit_Credit	Int	Ano	SCD1	Debetní kredit
Blocked	Tinyint	Ano	SCD1	Blokace
Direct Posting	Tinyint	Ano	SCD1	Přímé zveřejnění
No_ of Blank Lines	Tinyint	Ano	SCD1	Číslo prázdných řádků
Indentation	Int	Ano	SCD1	Členění
Totaling	Nvarchar(250)	Ano	SCD1	Celkové sumarizace
Automatic Ext_ Texts	Tinyint	Ano	SCD1	Automaticky vygenerované texty
Tax Liable	Tinyint	Ano	SCD1	Úroveň zdanění
VAT Bus_ Posting Group	Nvarchar(20)	Ano	SCD1	Daňová skupina pro business
VAT Prod_ Posting Group	Nvarchar(20)	Ano	SCD1	Daňová skupina pro výrobu
Exchange Rate Adjustment	Int	Ano	SCD1	Číslo kurzu pro převod
Account Subcategory Entry No_	Int	Ano	SCD1	Subkategorie rozvahového účtu
Cost Type No_	Nvarchar(20)	Ano	SCD1	Typ nákladů
Apply Entries	Tinyint	Ano	SCD1	Použité položky
ValidFrom	Date	Ne	SCD1	Datum od
ValidTo	Date	Ne	SCD1	Datum do
c_source	Nvarchar(10)	Ne	Business Klíč	Zdrojová databáze

Příloha 6: Struktura DimJob

<b>Název sloupce</b>	<b>Datový typ</b>	<b>Not Null</b>	<b>SCD typ</b>	<b>Popis sloupce</b>
Job_id	Int	Ano	Business Klíč	Primární klíč
No_	Nvarchar(20)	Ano	SCD1	Číslo práce
Search Description	Nvarchar(50)	Ano	SCD1	Vyhledávací popis
Description	Nvarchar(50)	Ano	SCD1	Popis
Bill-to Customer No_	Nvarchar(20)	Ano	SCD1	Číslo vyúčtování zákazníkovi
Creation Date	Datetime	Ano	SCD1	Datum vytvoření
Starting Date	Datetime	Ano	SCD1	Počáteční datum
Ending Date	Datetime	Ano	SCD1	Konečné datum
Status	Int	Ano	SCD1	Status
Person Responsible	Nvarchar(20)	Ano	SCD1	Zodpovědná osoba
Job Posting Group	Nvarchar(20)	Ano	SCD1	Skupina prací
Blocked	Int	Ano	SCD1	Blokace
Customer Disc_Group	Nvarchar(20)	Ano	SCD1	Skupina zákaznických slev
Language Code	Nvarchar(10)	Ano	SCD1	Kód jazyka
Bill-to Name	Nvarchar(50)	Ano	SCD1	Vyúčtování na jméno
Bill-to Address	Nvarchar(30)	Ano	SCD1	Vyúčtování na adresu
Bill-to City	Nvarchar(30)	Ano	SCD1	Vyúčtování na město
Bill-to Post Code	Nvarchar(20)	Ano	SCD1	Vyúčtování na směrovací číslo
No_Series	Nvarchar(20)	Ano	SCD1	Číslo série
Bill-to Country_Region Code	Nvarchar(10)	Ano	SCD1	Vyúčtování na kód regionu
Reserve	Int	Ano	SCD1	Rezerva
WIP Method	Nvarchar(20)	Ano	SCD1	VIP metoda
Currency Code	Nvarchar(10)	Ano	SCD1	Kód měny
Bill-to Contact No_	Nvarchar(20)	Ano	SCD1	Vyúčtování na číslo kontaktu
Bill-to Contact	Nvarchar(50)	Ano	SCD1	Vyúčtování na kontakt
Project Manager	Nvarchar(50)	Ano	SCD1	Projekt Manager
ValidFrom	Datetime	Ne	SCD2 ValidFrom	Datum od
ValidTo	Datetime	Ne	SCD2 ValidTo	Datum do
c_source	Nvarchar(10)	Ne	Business Klíč	Zdrojová databáze

Příloha 7: Struktura DimTerritory

<b>Název sloupce</b>	<b>Datový typ</b>	<b>Not Null</b>	<b>SCD typ</b>	<b>Popis sloupce</b>
Territory_id	Int	Ano	-	Primární klíč
Code	Nvarchar(3)	Ano	-	Kód země
Description	Nvarchar(25)	Ano	-	Popis země
ValidFrom	Date	Ne	-	Datum od
ValidTo	Date	Ne	-	Datum do

## Příloha 8: Struktura DimVendor

Název sloupce	Datový typ	Not Null	SCD typ	Popis sloupce
Vendor_id	Int	Ano	Business Klíč	Primární klíč
No_	Nvarchar(20)	Ano	SCD1	Číslo dodavatele
Name	Nvarchar(50)	Ano	SCD1	Jméno dodavatele
Search Name	Nvarchar(50)	Ano	SCD1	Vyhledávací jméno
Address	Nvarchar(20)	Ano	SCD2	Adresa
City	Nvarchar(30)	Ano	SCD2	Město
Contact	Nvarchar(50)	Ano	SCD1	Kontakt
Territory Code	Nvarchar(10)	Ano	SCD2	Kód země
Vendor Posting Group	Nvarchar(20)	Ano	SCD1	Dodavatelská skupina
Currency Code	Nvarchar(10)	Ano	SCD1	Kód měny
Payment Terms Code	Nvarchar(10)	Ano	SCD1	Kód platebního termínu
Purchaser Code	Nvarchar(20)	Ano	SCD1	Kód kupujícího
Shipment Method Code	Nvarchar(10)	Ano	SCD1	Kód metody poslání
Country_Region Code	Nvarchar(10)	Ano	SCD2	Kód regionu
Blocked	Int	Ano	SCD1	Blokace
Priority	Int	Ano	SCD1	Priorita
Prices Including VAT	Tinyint	Ano	SCD1	Cena včetně daně
Location Code	Nvarchar(10)	Ano	SCD2	Kód lokace
VAT Registration No_	Nvarchar(20)	Ano	SCD2	IČO
Gen_ Bus_ Posting Group	Nvarchar(20)	Ano	SCD1	Generální business skupina
Country	Nvarchar(30)	Ano	SCD1	Země
VAT Bus_ Posting Group	Nvarchar(20)	Ano	SCD1	Daňová skupina
Partner Type	Int	Ano	SCD1	Typ partnera
Privacy Blocked	Tinyint	Ano	SCD1	Blokace soukromí
Preferred Bank Account Code	Nvarchar(20)	Ano	SCD1	Preferovaný bankovní účet dodavatele
Vendor Bank Account Code	Nvarchar(30)	Ano	SCD1	Bankovní účet dodavatele
Vendor Bank Account Name	Nvarchar(50)	Ano	SCD1	Název banky
ValidFrom	Datetime	Ne	SCD2 ValidFrom	Datum od
ValidTo	Datetime	Ne	SCD2 ValidTo	Datum do
c_source	Nvarchar(10)	Ne	Business Klíč	Zdrojová databáze



## Příloha 9: Struktura DimWorker

<b>Název sloupce</b>	<b>Datový typ</b>	<b>Not Null</b>	<b>SCD typ</b>	<b>Popis sloupce</b>
WorkerID	Int	Ano	Business Klíč	Primární klíč
AlternateEmail Addresses	Nvarchar(255)	Ne	SCD1	Emailová adresa
BlockCredential	Bit	Ne	SCD1	Odpovědnost
City	Nvarchar(50)	Ne	SCD2	Město
Country	Nvarchar(20)	Ne	SCD2	Země
Department	Nvarchar(30)	Ne	SCD2	Oddělení
DisplayName	Nvarchar(50)	Ne	SCD1	Jméno pro zobrazení
Fax	Nvarchar(50)	Ne	SCD1	Fax
FirstName	Nvarchar(30)	Ne	SCD1	Jméno
LastName	Nvarchar(50)	Ne	SCD1	Příjmení
LastPasswordChange Timestamp	Nvarchar(30)	Ne	SCD1	Časová známka poslední změny hesla
LicenseAssignment Details	Nvarchar(50)	Ne	SCD1	Detaily přidělené licence
Licenses	Nvarchar(50)	Ne	SCD2	Licence
MobilePhone	Float	Ne	SCD1	Mobilní telefon
OathTokenMetadata	Nvarchar(255)	Ne	SCD1	Metadata
ObjectId	Nvarchar(50)	Ne	SCD1	Objekt id
Office	Nvarchar(30)	Ne	SCD2	Kancelář
PasswordNever Expires	Bit	Ne	SCD2	Vypršení hesla
PhoneNumber	Nvarchar(20)	Ne	SCD1	Telefonní číslo
PostalCode	Float	Ne	SCD2	Směrovací číslo
PreferredData Location	Nvarchar(30)	Ne	SCD1	Preferovaná data lokace
PreferredLanguage	Nvarchar(20)	Ne	SCD1	Preferovaný jazyk
ProxyAddresses	Nvarchar(40)	Ne	SCD1	Proxy adresy
State	Nvarchar(30)	Ne	SCD2	Stát
StreetAddress	Nvarchar(50)	Ne	SCD2	Ulice
StrongPassword Required	Bit	Ne	SCD1	Síla hesla
Title	Nvarchar(10)	Ne	SCD1	Titul
UsageLocation	Nvarchar(50)	Ne	SCD1	Místo výkonu
Externist	Bit	Ne	SCD1	Externista
UserPrincipalName	Nvarchar(50)	Ne	SCD1	Přihlašovací jméno
WhenCreated	Nvarchar(50)	Ne	SCD1	Datum vytvoření
ValidFrom	Datetime	Ne	SCD2 ValidFrom	Datum od
ValidTo	Datetime	Ne	SCD2 ValidTo	Datum do

## Příloha 10: Struktura FactCosts

<b>Název sloupce</b>	<b>Datový typ</b>	<b>Not Null</b>	<b>SCD typ</b>	<b>Popis sloupce</b>
id	Int	Ano	-	Primární klíč
Posting_date	Int	Ano	-	Datum vystavení
GL_Account	Int	Ano	-	ID rozvahové účty
Vendor_id	Int	Ano	-	ID dodavatele
Worker_id	Int	Ano	-	ID pracovníka
Currency_id	Int	Ano	-	ID měny
Document_Number	Int	Ano	-	Číslo faktury
Document_Number_Line	Nvarchar(20)	Ano	-	Řádek faktury
Territory_id	Int	Ano	-	ID země
Job_id	Int	Ano	-	ID práce
Gen_Prod_Group	Nvarchar(20)	Ano	-	Produkční skupina
Unit of Measure	Nvarchar(20)	Ano	-	Jednotka použitá pro výpočet
Quantity	Decimal(38,5)	Ano	-	Množství
Unit_Price	Decimal(38,5)	Ano	-	Jednotková cena
Cost_Amount	Decimal(38,5)	Ano	-	Náklady celkové
VAT_	Decimal(38,5)	Ano	-	Výše daně
Amount_Including_VAT	Decimal(38,5)	Ano	-	Cena včetně daně
Base_Currency_Code	Int	Ano	-	Kód základní měny
Amount_In_Base	Decimal(38,5)	Ano	-	Hodnota v základní měně
Amount_In_EURO	Decimal(38,5)	Ano	-	Hodnota v eurech
Payment_Terms_Code	Nvarchar(10)	Ano	-	Termín pro platbu
Effective_date	Date	Ano	-	Datum splatnosti

## Příloha 11: Struktura FactSales

<b>Název sloupce</b>	<b>Datový typ</b>	<b>Not Null</b>	<b>SCD typ</b>	<b>Popis sloupce</b>
id	Int	Ano	-	Primární klíč
Posting_date	Int	Ano	-	Datum vystavení
GL_Account	Int	Ano	-	ID rozvahové účty
Customer_id	Int	Ano	-	ID zákazníka
Worker_id	Int	Ano	-	ID pracovníka
Currency_id	Int	Ano	-	ID měny
Invoice_Number	Int	Ano	-	Číslo faktury
Invoice_Number_Line	Nvarchar(20)	Ano	-	Řádek faktury
Territory_id	Int	Ano	-	ID země
Job_id	Int	Ano	-	ID práce
Gen_Prod_Group	Nvarchar(20)	Ano	-	Produkční skupina
Unit of Measure	Nvarchar(20)	Ano	-	Jednotka použitá pro výpočet
Quantity	Decimal(38,5)	Ano	-	Množství
Unit_Price	Decimal(38,5)	Ano	-	Jednotková cena
Sales_Amount	Decimal(38,5)	Ano	-	Celkové prodeje
VAT_	Decimal(38,5)	Ano	-	Výše daně
Amount_Including_VAT	Decimal(38,5)	Ano	-	Cena včetně daně
Base_Currency_Code	Int	Ano	-	Kód základní měny
Amount_In_Base	Decimal(38,5)	Ano	-	Hodnota v základní měně
Amount_In_EURO	Decimal(38,5)	Ano	-	Hodnota v eurech
Payment_Terms_Code	Nvarchar(10)	Ano	-	Termín pro platbu
Effective_date	Date	Ano	-	Datum splatnosti