

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

**Absolvování individuální odborné praxe**  
**Individual Professional Practice in the**  
**Company**

# Zadání bakalářské práce

Student: **Jan Polach**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

Jazyk vypracování: čeština

## Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Stora Enso Wood Products Zdirec s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

## Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Petr Olivka, Ph.D.**

Konzultant bakalářské práce: Ing. Pavel Meca

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty


Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 1. dubna 2016

  
.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 1. dubna 2016



.....

tora Enso Wood Products Ždírec s.r.o  
82 63 Ždírec nad Doubravou, Nádražní 6c

Rád bych poděkoval společnosti Stora Enso za umožnění absolvování odborné praxe v této firmě. Jmenovitě bych především poděkoval svému konzultantovi Ing. Pavlovi Mecovi za podporu, cenné rady a odborné vedení praxe. Dále bych poděkoval mému vedoucímu bakalářské práce Ing. Petru Olivkovi, Ph.D., jehož rady a připomínky mi pomáhaly při tvorbě Bakalářské práce.

## **Abstrakt**

Cílem této bakalářské práce je popsat úkoly a jejich řešení, se kterými jsem setkal v IT centru společnosti Stora Enso nacházející se v Ostravě, ve kterém jsem absolvoval odbornou praxi. Během této doby jsem plnil celou řadu úkolů v oblasti reportingu a analýzy dat. Tato práce obsahuje popis vybrané skupiny problémů a jejich řešení, ať už teoretické nebo i praktické. Dále jsou zde uvedeny informace o získaných znalostech a zkušenostech.

**Klíčová slova:** Business Intelligence, MSBI, SSAS, Reporting, SQL, Tabular Model

## **Abstract**

The main purpose of this bachelor thesis is to describe tasks and their solution, which I was carrying out in IT department of company Stora Enso located in Ostrava, where I was working as Intern. My tasks were mainly connected to reporting and analysis of large amount of data. This thesis is describing some of those tasks and their solution from perspective of theory and practice. Furthermore the gained experience and knowledge are described in detail.

**Key Words:** Business Intelligence, MSBI, SSAS, Reporting, SQL, Tabular Model

# Obsah

Seznam použitých zkratk a symbolů	7
Seznam obrázků	8
Seznam výpisů zdrojového kódu	9
<b>1 Úvod</b>	<b>10</b>
<b>2 Firma a popis pracovního zařazení</b>	<b>11</b>
<b>3 Použité technologie</b>	<b>12</b>
3.1 Business Intelligence . . . . .	12
3.2 SSIS . . . . .	13
3.3 SSAS . . . . .	14
3.4 SSAS Tabular Model . . . . .	15
3.5 TFS . . . . .	16
<b>4 Zadané úkoly</b>	<b>17</b>
4.1 Reporty v tabulárním modelu . . . . .	17
4.2 Report ze SAP BW . . . . .	19
4.3 Ostatní úkoly . . . . .	25
<b>5 Uplatněné znalosti a dovednosti získané v průběhu studia</b>	<b>26</b>
<b>6 Znalosti a dovednosti scházející v průběhu praxe</b>	<b>27</b>
<b>7 Závěr</b>	<b>28</b>
<b>Literatura</b>	<b>29</b>

## Seznam použitých zkratek a symbolů

BI	– Business Intelligence
MSBI	– Microsoft Business Intelligence
SSAS	– SQL Server Analysis Services
TAS	– Tabular Analysis Services
SSRS	– SQL Server Reporting Services
SSIS	– SQL Server INtegration Services
MDX	– MultiDimensional Expressions
DAX	– Data Analysis Expressions
SQL	– Structured Query Language
WPS	– Wood Products System
SAP	– účetnický systém
DM	– Data Mart
DW	– Data Warehouse
RAACO	– Reporting & Analysis Applications and Collaboration Team
ETL	– extract, transform, load process



## Seznam obrázků

1	Logo firmy Stora Enso [7] . . . . .	11
2	Ilustrační obrázek znázornění BI procesu [8] . . . . .	12
3	Ilustrační obrázek struktury SSIS balíčku [9] . . . . .	13
4	Ukázka editoru SSAS kostky . . . . .	14
5	Ukázka obrazovky tabulárního modelu . . . . .	15
6	Ukázka struktury tabulárního modelu OI_reportu . . . . .	18
7	Ukázka tabulky výsledné hierarchie . . . . .	25

## Seznam výpisů zdrojového kódu

1	Ukázka aktualizování faktových tabulek . . . . .	20
2	Ukázka funkce na naplnění hierarchické tabulky . . . . .	21

# 1 Úvod

I přes velké množství témat bakalářských prací jsem si nakonec vybral absolvování individuální bakalářské praxe ve firmě. Tuto cestu jsem zvolil, protože mi to přijde jako nejlepší způsob, jak získat zkušenosti z praxe, které se mi budou určitě v budoucnu hodit.

Z mnoha firem, které tuto praxi nabízely, jsem si nakonec vybral firmu Stora Enso. O této firmě jsem se dozvěděl na veletrhu práce KariéraPlus, kde se mi tato firma zamlouvala. Kontaktoval jsem firmu a byl jsem pozván na ústní pohovor, který probíhal jak v češtině, tak i v angličtině. U pohovoru byl také Ing. Pavel Meca, který se později stal mým nadřízeným a kolegou. Po několika dnech mě firma kontaktovala a od září 2015 jsem nastoupil.

Sepsáním této práce bych se rád podělil o zkušenosti, které jsem nabyl během bakalářské praxe. V bakalářské práci popisuji dva hlavní projekty, na kterých jsem se podílel ve firmě. Dále tato práce obsahuje shrnutí průběhu praxe.

## 2 Firma a popis pracovního zařazení

Firma Stora Enso s.r.o 1 je švédská společnost založená roku 1998. Dnes je to mezinárodní produkční firma zabývající se zpracováním dřeva, papíru, biomateriálu atd. Aktuálně zaměstnává kolem 26 tisíc zaměstnanců po celém světě v 35-ti státech. Celá společnost se rozděluje na 5 základních divizí:

- Consumer Board
- Packaging Solutions
- Biomaterials
- Wood Products
- Paper

Každá tato divize se stará o nějakou oblast v obchodní sféře. V Ostravě sídlí IT centrum této společnosti. Bylo založeno v roce 2006 jako strategické servisní centrum pro interní vývoj software pro produkční a obchodní oblasti firmy. Nyní jsou zde vyvíjeny a udržovány systémy na plánování, logistiku a prodej pro všechny divize firmy. IT centrum v Ostravě je největší IT centrum společnosti čítající víc než 140 zaměstnanců.



Obrázek 1: Logo firmy Stora Enso [7]

Byl jsem vybrán na pozici Microsoft Business Intelligence developer. Požadavky na tuto pozici byly hlavně znalost SQL jazyku, základní znalost o principech Business Intelligence (dále jen BI) a hlavně znalost angličtiny na komunikační úrovni. SQL jazyk jsem se učil na VŠB, tento požadavek jsem tedy splnil. O principech BI jsem se sice na škole nic nedozvěděl, ale rok zpátky jsem absolvoval Business Intelligence Academy ve firmě TIETO, kde jsme BI probírali celkem podrobně. Takže druhou podmínku jsem myslím splnil na výbornou a s angličtinou problém nemám.

Na základě úspěšného pohovoru jsem byl zařazen do týmu RAACO - Reporting & Analysis Applications and Collaboration Team pod vedením Ing. Pavla Mecen. Úkolem tohoto týmu je starat se o reporting a analýzy z divize Wood Products. Skupina Pavly Mecen pracuje s daty a snaží se je transformovat z různých zdrojových systémů do Microsoft MSBI reportů, ať se kolegové z businessu můžou na základě těchto informací rozhodovat o dalších postupech a případně analyzovat historická data. Tomuto odvětví se říká obecně Business Intelligence.

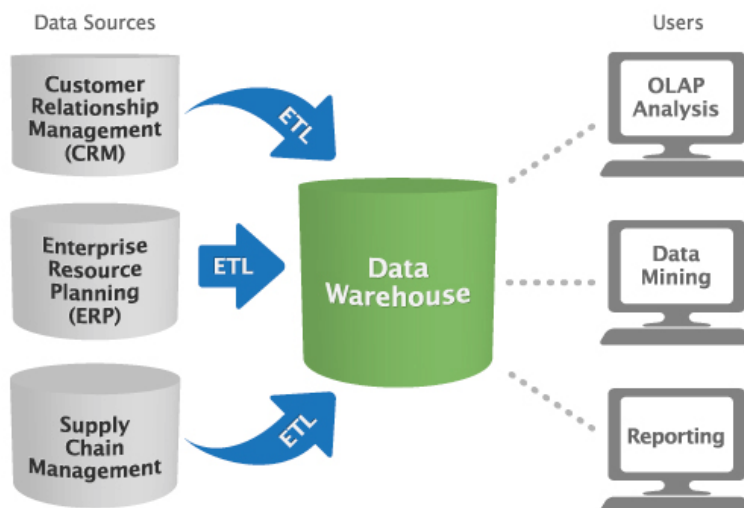
### 3 Použité technologie

V následujících kapitolách popisují technologie, se kterými jsem se setkal na bakalářské praxi. Jedná se o technologie firmy Microsoft. Tyto technologie jsem musel nastudovat hned na začátku praxe, abych s nimi mohl dále pracovat. Veškeré informace jsem se dozvěděl hlavně z [2], a také z různých stránek na internetu, kde jsem si ujasnil potřebné znalosti.

#### 3.1 Business Intelligence

Nejdřív bych uvedl, co vlastně Business Intelligence [2] je a následně řeknu něco o tom, s jakými BI technologiemi jsem se setkal. BI je rámcový termín označující paletu softwarových aplikací využívaných k analýze syrových dat společnosti. BI je disciplínou složenou z několika navzájem souvisejících aktivit jako je data mining, OLAP (Online Analytical Processing), dotazování na databázi a vytváření zpráv, čili reporting.

BI proces začíná u zdrojových systémů. Převážně se jedná o strukturovaná data v relační databázi, data v excelu atd. V první fázi zpracování dat je ETL. ETL je zkratka pro extrakci, transformaci a nahrání dat do datového skladu. Data jsou extrahována ze zdrojových systémů, poté transformována do datového skladu v požadované podobě. ETL procesy se provádějí přes SSIS balíčky. Další fází je OLAP kostka. Microsoft tuto OLAP kostku nazval SSAS, s touto technologií jsem se už také setkal. Poslední fází celého BI je vytváření reportů a zpráv, které analyzují data získaná ze zdrojových systémů. Tady existuje více způsobů, jak zobrazovat data v reportech. Microsoft pro tuto část nachystal technologie jako SSRS, PowerPivot, Tabulární model atp. Tyto technologie rozvedu později v následujících kapitolách.

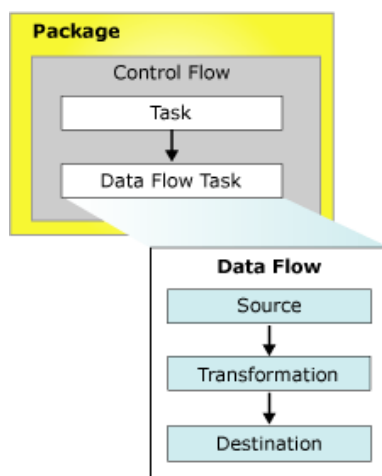


Obrázek 2: Ilustrační obrázek znázornění BI procesu [8]

V neposlední řadě bych poznamenal, že výše zmíněné technologie jsou sice od firmy Microsoft, ale existuje mnoho dalších technologií od různých společností, které plní stejný účel. Například pro reporting to je Qlikview, Cognos a další.

### 3.2 SSIS

SSIS (SQL Server Integration Services) je nástroj od firmy Microsoft, který slouží k řešení úkolů ohledně migrace dat. Výsledkem tohoto nástroje jsou SSIS balíčky, které se mohou spouštět automaticky nebo ručně. Migrace dat je část ETL procesu, kdy chceme dostat data ze zdrojových systémů do jednotného formátu.



Obrázek 3: Ilustrační obrázek struktury SSIS balíčku [9]

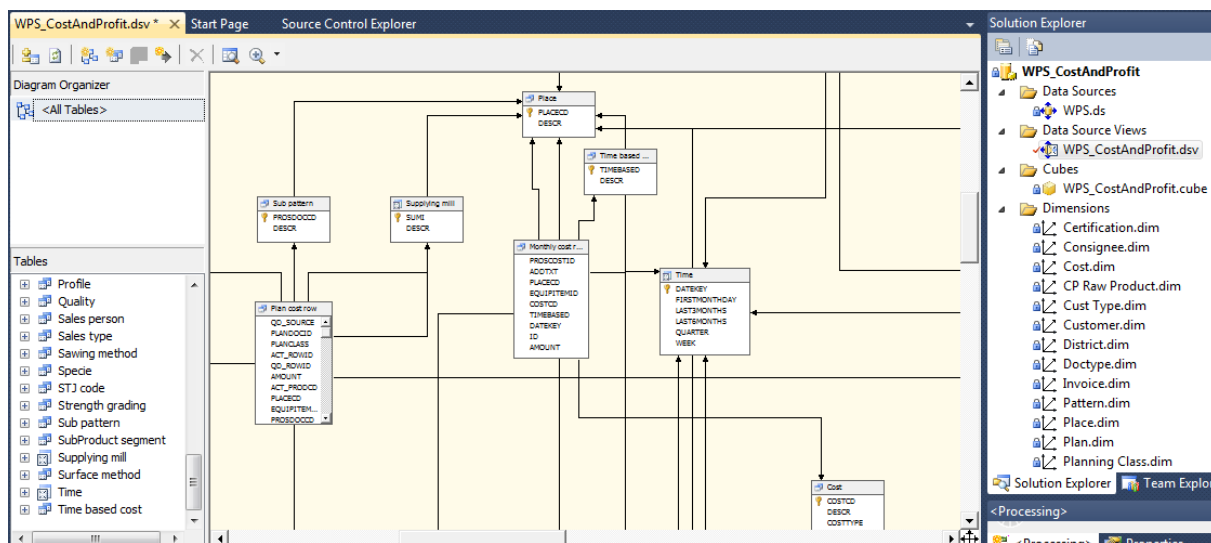
V SSIS balíčku je možné dělat mnoho funkcí. Od jednoduchého provedení SQL operace na databázi až po složité transformace a korekce dat. SSIS balíček tvoří dvě základní Flow komponenty. Na obrázku 3 jsou obě komponenty zobrazeny, první je Control Flow a další je Data Flow. Control Flow slouží k operacím pro přípravu databáze jako volání procedur a funkcí nebo také větvení celého balíčku podle různých podmínek. Příjemně mě překvapila i možnost vložení C# kódu, kterou jsem během praxe také využil. Control Flow obsahuje více Data Flow. Data Flow už pracuje přímo s daty. To znamená, že vždycky musí obsahovat zdroj a cíl. Mezi zdrojem a cílem může být opět mnoho druhů transformací jako je přidávání sloupce, převádění dat na jiné formáty, plnit sloupce podle různých podmínek atd.

Samozřejmě SSIS balíčky podporují mnoho dalších prostředků, které umožňují velmi široké využití. Umí pracovat s proměnnými, volat jiné externí balíčky, pracovat s parametry a další. Za svou praxi jsem měl bohužel možnost setkat se s pouhým základem tohoto nástroje s porovnáním s tím, co vše se dá přes SSIS balíčky udělat. Ale tento nástroj má také negativní stránku. Tím, že je tento nástroj tak sofistikovaný a umí tolik věcí, tak pro základní věci je lepší použít jiný nástroj. S tímto problémem jsem se setkal, když jsem potřeboval jednoduchý csv soubor (asi s 500 tisíci řádky) převést do databáze. Přes SSIS balíček tato akce trvala asi 20 minut. Přemýšlel

jsem, kde by mohla být chyba a kolega mi poradil, že na takové jednoduché jednorázové operace je lepší použít nástroj Import and Export Wizard. Nakonec byl tento proces hotový během minuty a mohl jsem dál pokračovat v úkolu.

### 3.3 SSAS

SSAS (SQL Server Analysis Services) je nástroj od Microsoftu, o kterém jsem se nejvíc dozvěděl z [3]. Reprezentuje OLAP kostky, pokud se bavíme v terminologii BI. Kostka obsahuje dvě hlavní části. Jsou to measury (fakta) a dimenze. Obě tyto položky jsou v základu tabulky s daty. Rozdíl je v tom, že measury obsahují data, nad kterými chceme dělat nějaké výpočty, filtrovat je, prostě jsou to syrová data. Dimenze na rozdíl od measur obsahují hlavně číselníky a jiné věci, na základě kterých se pak measury filtrují a počítají. Measury tvoří takzvané faktové tabulky a dimenze zase dimenzní tabulky. Nejjednodušší podoba kostky v SSAS je jedna faktová tabulka a k ní napojené dimenzní tabulky pomocí vedlejších klíčů. Aby jste si mohli představit, co třeba dimenzní tabulka obsahuje za data, tak bych rád uvedl pár příkladů, se kterými jsem se setkal. Dimenzní tabulka může obsahovat číselník kalendáře, dřevozpracujících podniků, typů dřeva, fáze objednávky atp. Faktové tabulky na rozdíl od dimenzních obsahují taková data, která chceme sledovat v dané kostce (reportu). Pro získávání dat z SSAS kostky se používá dotazovací jazyk MDX [4].

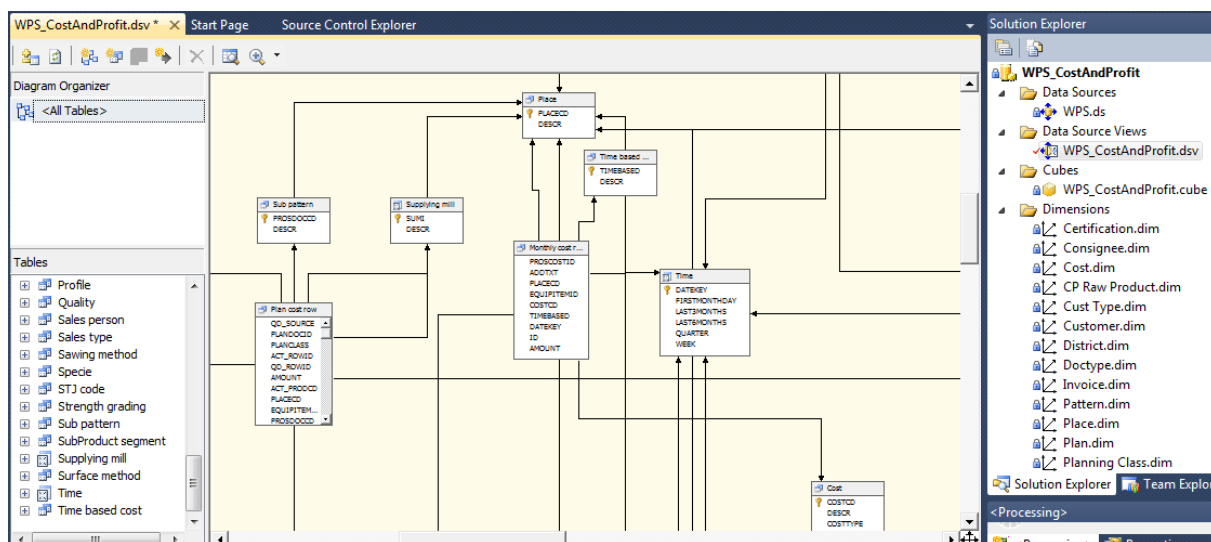


Obrázek 4: Ukázka editoru SSAS kostky (CostAndProfit)

Na obrázku číslo 4 můžeme vidět kousek obrazovky editoru SSAS kostky ve Visual Studiu. V pravém okně najdeme různé zdroje dat, pohledy na zdroje dat, samotnou kostku a v neposlední řadě seznam dimenzí. V hlavním okně můžeme vidět opravdu malý kousek celého plátna všech tabulek. Můžeme zde vidět vazby mezi dimenzními a faktovými tabulkami.

### 3.4 SSAS Tabular Model

Tabulární model je poměrně nová technologie od Microsoftu, která slouží k reportingu [1]. Předchůdce tabulárního modelu byl doplněk v Excelu s názvem PowerPivot [5]. K porovnání těchto dvou technologií se dostanu až v následujících kapitolách na názorném příkladě. K této technologii jsem se dostal v prvním zadaném úkolu, kdy jsem měl za úkol najít alternativu k již zmíněnému doplňku PowerPivot, protože s narůstajícími daty PowerPivot jednoduše nestačil. Dále bych tento problém blíže vysvětlil a popsal v kapitole o zadaných úkolech.



Obrázek 5: Ukázka obrazovky tabulárního modelu

Na obrázku číslo 5 si můžeme prohlédnout jeden screenshot tabulárního modelu z Visual Studia a projdeme si, co na obrázku můžeme vidět. Hlavní okno obsahuje čistá data, která existují v dané tabulce. Ke každé tabulce můžeme vytvořit skoro neomezený počet počítaných vzorců nad daty (jedná se o buňky s modrým pozadím). Dále můžeme do tabulárního modelu přidávat další kalkulované sloupce a nad těmito sloupci provádět taktéž různé kalkulace.



### 3.5 TFS

TFS (Team Foundation Server) je verzovací systém od firmy Microsoft. Ve firmě se tento systém používá z více důvodů. Za prvé při takovém množství lidí a projektů je potřeba nějak organizovat a verzovat projekty a různé verze projektů. Bez verzovacího systému by to asi šlo obtížně organizovat. Další důvod je ten, že tento nástroj je přímo importovaný a podporovaný Visual Studiem, takže jeho použitelnost a využití je určitě výhodné. Největší problém v tomto verzovacím systému (vlastně ve všech verzovacích systémech) jsou binární data. Například pokud se bavíme o SSIS balíčcích, tabulárním modelu nebo SSAS kostce, ty všechny projekty obsahují nějaká binární data nebo automaticky generované soubory. Integrace TFS ve firmě je dle mého pouze taková pomocná a hlavně na to, ať dostupný projekt mají všichni členové týmu. Konflikty v tomto systému se ve firmě moc neřešily, protože pokud začal někdo se souborem pracovat, soubor se zamkl a nemohl s ním nikdo jiný dělat do té doby, než byla práce na souboru hotová. Dalším důvodem, proč se vyhnout konfliktům ve slučování práce různých spolupracovníků, je fakt, že se ve firmě často pracuje s binárními daty, která se špatně slučují ve verzovacím systému.

## 4 Zadané úkoly

V následujících kapitolách bych rád přiblížil zadání a vypracování úkolů, které jsem ve firmě řešil. Jedná se o:

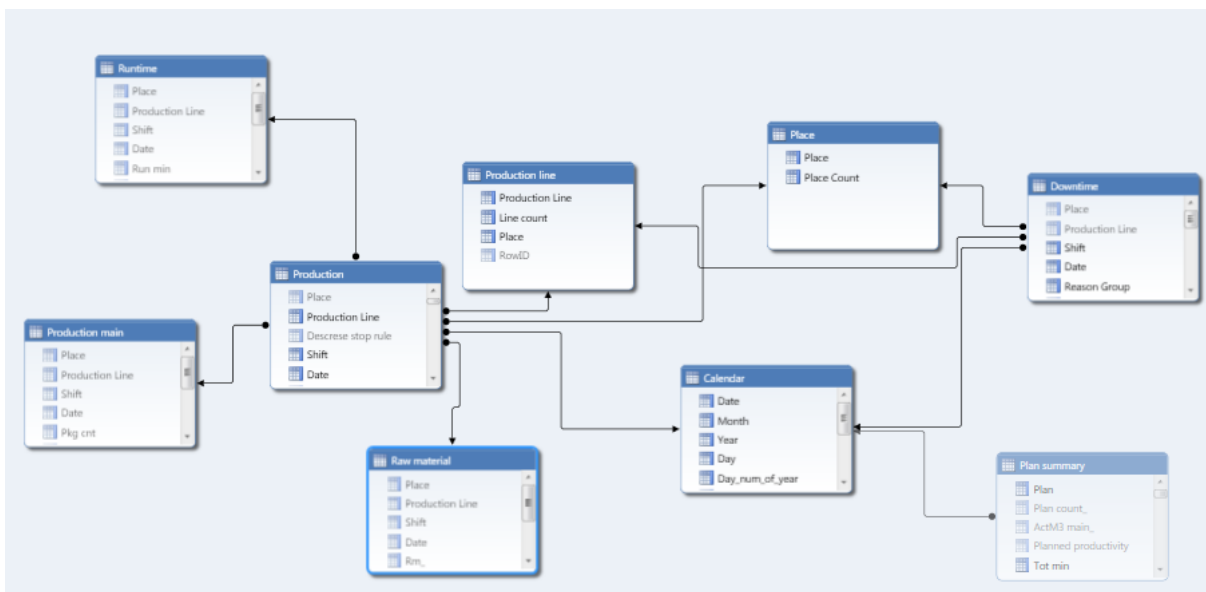
1. Reporty v tabulárním modelu
2. Reporty ze SAP BW
3. Další dílčí úkoly

### 4.1 Reporty v tabulárním modelu

Po nástupu do firmy jsem se první dva týdny seznamoval s používanými technologiemi. Prošel jsem si plno ukázkových příkladů, abych vůbec pochopil technologii, než budu pracovat s konkrétními daty na zadaných úkolech. Mým prvním větším úkolem bylo vymyslet, jak optimalizovat reporty v nástroji PowerPivot [5], případně jakou jinou technologii použít, která by fungovala lépe než již zmíněné PowerPivot reporty.

Ve firmě bylo vytvořených 6 SSAS OLAP kostek, které byly využívány 12. reporty. Jedná se o reporty z divize WoodProducts, které používali kontroloři na dřevozpracujících podnicích a různí manažeři. Všechny reporty byly implementovány v Excelu přes doplněk PowerPivot. Tento nástroj funguje tak, že veškerá data jsou uložena přímo v Excelu lokálně na disku. Z toho vyplývá, že největší problém při větších množstvích dat je velikost výsledných .xlsx souborů, větší reporty měly až přes 100 MB. Tato nevýhoda se časem stala celkem zásadním důvodem, proč najít něco lepšího, protože uživatelům se samozřejmě nelíbilo, že při nové verzi musí stáhnout soubor o velikosti 100 MB, aby se mohli podívat na data. Další nevýhodou tohoto řešení byla bezpečnost a aktualizace dat (aktualizace všech dat v PowerPivotu). Když si uživatelé stáhli .xlsx soubor a byla v něm již stáhnutá data jiného uživatele, který měl vyšší pravomoc v rámci společnosti, tak před aktualizací PowerPivotu mohli vidět i nepřístupná data. Jak už jsem zmínil, další problém byl v tom, jak aktualizovat data v reportu pomocí PowerPivotu. Uživatel musel nejdříve otevřít daný report, poté přejít do doplňku PowerPivot (kliknutím na ikonku Manage v horním panelu), a poté kliknout na “Refresh All”. Tento postup museli udělat uživatelé pokaždé, když chtěli vidět v reportu aktuální data. Toto aktualizování zabralo v některých případech i několik minut (záleželo na připojení k internetu a na výkonu počítače).

Na základě těchto požadavků měl náš tým najít nějaký způsob, jak se těchto problémů co nejrychleji zbavit. Rozhodovali jsme se, jestli se nepřiklonit k reportovacím balíčkům SSRS (opět od Microsoftu) nebo najít nějaké vhodnější řešení. Nakonec jsme našli technologii od firmy Microsoft, jedná se o Tabulární modely. Tabulární model je něco jako PowerPivot verze na serveru. To znamená, že všechna data jsou uložena na tabulárním serveru (administrace se podobá již zmíněnému PowerPivotu) a data se získávají jen na základě toho, co chce uživatel zobrazit. Tím jsme se zbavili skoro všech problémů. Reporty, které zabíraly předtím těch 100



Obrázek 6: Ukázka struktury tabulárního modelu OI\_reportu

MB, zabíraly v novém řešení maximálně pár MB. Zároveň zmizel i problém s tím, že neoprávnění uživatelé viděli data, ke kterým správně neměli přístup.

Po získání těchto informací jsem dostal za úkol jeden ukázkový report převést na tabulární model a zjistit, jak tento model používat a také zjistit, zda to funguje tak, jak potřebujeme. Tímto reportem byl středně velký report s názvem WPS\_OI\_report. Prvním krokem bylo vytvořit projekt na tabulárním serveru a vytvořit tam stejnou strukturu jako v ukázkovém reportu s PowerPivot doplňkem. To znamená vytvořit connection na OLAP kostky, které daný report používá a vytvořit tabulky tak, jak to bylo v zdrojovém reportu. Dalším úkolem bylo vytvořit role, které k tomuto reportu mohou přistupovat. Předtím tuto správu rolí řešila OLAP kostka, teď se tato pravomoc přesunula na vyšší vrstvu, na tabulární model. Role se filtrovaly pomocí toho, která skupina windows účtů ve společnosti má přístup k určitému Place (tzn. jeden nebo více dřevozpracujících podniků ze seznamu společnosti). O těchto rolích jsem samozřejmě neměl žádné tušení, jak to funguje. Nakonec jsem po pár hodinách bádání a hledání pochopil, co mám zadat do formuláře, aby to data filtrovalo správně. Muselo se vytvořit asi 10 rolí právě na základě pravomocí, v různých dřevozpracujících podnicích.

Poté jsem vytvořil nový report, připojil jsem ho na tabulární model, vytvořil filtry a sloupce podle předlohy, jak vypadala původní verze reportu. Po kontrole, zda report vrací stejné hodnoty a změnění formátu čísel, byl tento ukázkový report hotový. Poslední věc, která se potřebovala vyřešit, bylo automatické aktualizování serverových tabulárních reportů. Aby byly tabulární modely vždy aktuální, tak po každém aktualizování SSAS kostky se musely zaktualizovat všechny tabulární modely, které z této kostky čerpaly.

Další otázkou bylo, jak tento model aktualizovat, na výběr byl full process, default process

a nebo přes script. Full proces z nějakého neznámého důvodu trval strašně dlouho, takže to bylo nepřípustné. Default process zase nezaručoval, že když se změní nebo přidá sloupec do tabulárního modelu, že to processne se správnými daty. Poslední možnost byla udělat full process, ale přes script. Zdá se, že to je stejné jako full process, ale v tomto případě možnost přes script byla mnohonásobně rychlejší. Teď už bylo zaručeno, že tabulární model obsahuje vždy ta nejčerstvější data.

V dalším kroku se tento report poslal do produkce, zda jsou uživatelé spokojeni s výsledkem a zda o tuto změnu mají zájem. Po potvrzení, že jim tyto reporty vyhovují, jsem začal ve spolupráci s dalším studentem Vratislavem Gorčíkem předělávat PowerPivot reporty na tabulární model podle ukázkového reportu. Toto, dá se říci nahození, bylo časově náročné, ale později se tento proces zautomatizoval a člověk automaticky postupoval tak, jak se to po pár reportech naučil.

Reporty se poslaly kolegovi do Finska, kde to rozeslal testovací skupině uživatelů. Uživatelům se to líbilo, ale chtěli do mnohých reportů udělat plno změn jako například změnit data, popřípadě přejmenovat sloupce, aby to bylo víc přívětivé pro konečné uživatele. Nakonec se tyto reporty řešily po jednom a probíhala komunikace přes emaily.

Dalším podúkolem bylo zaokrouhlit všechny číselné hodnoty na 4 desetinná místa a zbavit se nečíselných hodnot (nekonečno není číslo), a to se dalo provést tak, že se ošetřilo dělení nulou. S kolegou jsme našli jednotný způsob, jak detekovat tyto nečíselné hodnoty a ošetřit je. Nakonec z toho vzešel DAX příkaz, který toto ošetřoval. Poté jsme museli opět projít všech 12 reportů a každou číselnou hodnotu opravit a zaokrouhlit. Po těchto krocích jsme opět pro jistotu kontrolovali hodnoty v reportech, zda je vše v pořádku a neudělali jsme někde chybu.

Dále jsme poskytovali podporu pro tabulární reporty. To znamená, že když napsal konečný uživatel, že s tím má nějaký problém, tak jsme mu pomohli. Šlo většinou o neznalost samotné technologie, jak funguje a nebo o špatně nastavený přístup v roli.

## 4.2 Report ze SAP BW

Dalším větším úkolem bylo udělat zcela nové reporty ohledně HR dat. Tento úkol nepatří přímo pod WoodProducts divizi, ale je to pro celou společnost obecně. Zdrojová data jsou uložena v databázi, kterou využívá SAP. Reporty ze SAPu, které se doteď používaly, využívaly nástroj SAP BW, jedná se konkrétně o 2 reporty. Tato technologie není od Microsoftu a asi to byla nejjednodušší a neoptimalnější možnost, jak tyto reporty udělat. Výhodou těchto reportů je to, jak implementují hierarchii celé společnosti, jdou tam dělat věci, které se v Microsoft nástrojích dělají složitěji. Po čase chtěla společnost všechny reporty pod záštitou jedné společnosti pomocí jednotného nástroje. Touto společností je Microsoft a nástrojem je nějaký typ reportu, jehož výsledek je v Excelu.

K dispozici jsme dostali specifikaci dat, která jsou uložena v SAPu a popis fungujících reportů, které jsou k dispozici v SAP BW. Naším úkolem bylo dostat data ze SAPu, přetransformovat je a uložit do naší SQL Microsoft databáze. Další krok byl vytvořit z toho OLAP SSAS kostku a vytvořit report, který funguje podobně jako ze SAP BW.

Prvním krokem bylo vytvořit tabulky v naší SQL databázi. Tyto tabulky jsme vytvořili společně s dalším studentem Vratislavem Gorčíkem. Vratislav měl za úkol vytvořit tabulky, které se naplňovaly ručně, já jsem měl na starosti ten zbytek.

Následujícím krokem bylo naplnit tabulky daty. Tento proces se dělal přes SSIS balíček, to jsem musel dělat přes vzdálenou plochu jiného počítače, protože na data ze SAPu se přistupovalo přes SAP HUB, což byl jeden doplněk, který musel být nainstalovaný na počítači, kde se spouštěl SSIS balíček. Než zprovoznit tento doplněk na vlastním počítači, tak bylo daleko lepší zvolit tento postup přes vzdálenou plochu severu. V první fázi jsem vytvořil pro každou dimenzi svůj SSIS balíček, který jsem zatím spouštěl ručně, abych tam měl data. Balíček v této fázi jen načel data ze SAPu, přetypoval sloupce a odstranil prázdné hodnoty. Tento proces se musel udělat pro každý sloupec, kde byla čísla, aby se data stala více korektními. Tak jsem udělal 8 SSIS balíčků a šel jsem na další krok s plánem, že se ještě k těmto balíčků vrátím podle situace.

Dalším dílčím krokem bylo napojit pomocí vedlejších klíčů všechny dimenze na faktové tabulky. Tento proces se zdá být jednoduchý a v polovině případů také byl, prostě se vytvořil vedlejší klíč na primární klíč dimenze. Tento postup ale nešel úplně u všech tabulek. V několika (asi 6) tabulkách se stalo to, že kód v dimenzní tabulce nebyl unikátní, to byl celkem problém. Zjistil jsem, že problém je v tom, že v těch tabulkách jsou stejné kódy, ale jiné hodnoty ve sloupcích jako jsou: data (ValidFrom, ValidTo), jazyk (LanguageKey) atd. Řešení bylo takové, že se musel udělat nový sloupec, který byl jednoduchý klíč typu int s autoinkrementem. Poté se vytvořily nové sloupce v těch faktových tabulkách, které právě využívaly tyto dimenze. Dalším logickým krokem bylo přepočítat po každém zaktualizování dimenze a aktualizovat nově vytvořený sloupec ve faktové tabulce. Tento proces jsem dělal přes proceduru, která byla uložena na serveru. Tělo procedury obsahoval update příkaz, který aktualizoval danou faktovou tabulku a na základě dat v dimenzi se přiřadil správný primární klíč viz 1.

---

```
ALTER PROCEDURE [dbo].[usp_HR_HRPositionChanged]

AS
BEGIN
--zaktualizovani tabulky FTEFigures, na zaklade jazyku a data
UPDATE [TrainingMSBI].[hr].[FTEFigures] SET PositionID = (
SELECT PositionID FROM [TrainingMSBI].[hr].[HRPosition] A
WHERE A.[Position] = [TrainingMSBI].[hr].[FTEFigures].[Position] AND A.[
    LanguageKey] = 'E' AND [TrainingMSBI].[hr].[FTEFigures].[CalendarDay]
    BETWEEN A.[Validfrom] AND A.ValidTo)

--zaktualizovani tabulky PersonnelActionFigures, na zaklade jazyku a data
UPDATE [TrainingMSBI].[hr].[PersonnelActionFigures] SET PositionID = (
SELECT PositionID FROM [TrainingMSBI].[hr].[HRPosition] A
WHERE A.[Position] = [TrainingMSBI].[hr].[PersonnelActionFigures].[Position]
```

```

AND A.[LanguageKey] = 'E' AND [TrainingMSBI].[hr].[
PersonnelActionFigures].[CalendarDay] BETWEEN A.[Validfrom] AND A.ValidTo
)

```

END

---

### Výpis 1: Ukázka aktualizování faktových tabulek

Když jsem měl hotové všechny procedury, které aktualizovaly požadované vedlejší klíče, tak jsem se vrátil zpátky do SSIS balíčků, kde jsem vložil zavolání procedury po nahrání dat do SQL databáze. Při testování, zda to všechno funguje, jsem narazil na celkem velký problém, se kterým jsem nepočítal. Nešly smazat dimenze, protože existovala faktová tabulka, která odkazovala do této dimenzní tabulky. Bylo více způsobů, jak tento problém vyřešit. Prvním řešením bylo smazat při každém aktualizování dimenze i faktovou tabulku (zapomněl jsem poznamenat, že ve faktových tabulkách bylo kolem 1 000 000 záznamů), a poté vytvořit znovu dimenzi, pak opět natahat ze SAPu tabulku s miliony záznamy a všechno přepočítávat. Toto řešení bylo víc než nepraktické. Dalším řešením bylo smazat závislost dané dimenze na faktové tabulce, a pak zase tuto závislost vytvořit. Toto řešení jsem opět zavrhl, protože je to vysoce náchylné na chyby. Poslední a nejlepší možnost byla taková: sloučit všechny SSIS balíčky do jednoho. V tomto řešení se nejdřív smažou všechny faktové tabulky, poté všechny dimenzní tabulky. V dalším kroku se naplní daty dimenzní tabulky, a pak faktové tabulky a následně se musí zavolat všechny procedury nad faktovými tabulkami (to znamená napojení těch složitých klíčů z dimenzí).

---

```

ALTER VIEW [hr].[v_HierarchyOrgUnit]
As

--common table expression
with tree (parentnodeid,nodeid, pathh) AS (

--rekurzivni tvoreni stringu, obsahujici level1/Level2...
select A.parentnodeid , A.nodeid, CAST(A.parentnodeid as varchar(8000)) + '/'
      + CAST(A.nodeid as varchar(8000)) as pathh
from [TrainingMSBI].hr.orgunit A
where A.nodeid in (select B.nodeid from [TrainingMSBI].hr.orgunit B)
OR A.parentnodeid = A.nodeid

union all
(
select c.parentnodeid,c.nodeid, CAST(c.parentnodeid as varchar(8000)) + '/'
      + p.pathh

```

```

from [TrainingMSBI].hr.orgunit c, tree p WHERE c.nodeid = p.parentnodeid

AND c.parentnodeid <> p.nodeid )
)

--v tomto selectu se musi string naparsovat, a ulozit do kazdeho sloupce
zvlast
--zaroven se musi otestovat zda vysledny retezec neni prazdny
SELECT A.nodeid,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 2, '/') = '' then A.nodeid else
    dbo.UFN_SEPARATES_COLUMNS(A.pathh, 2, '/') END as level1,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 2, '/') = '' THEN NULL ELSE (
    SELECT 01.Shortdescription from [TrainingMSBI].hr.orgunit 01 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 2, '/')) = 01.
    nodeid) END as level1Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 3, '/') = '' then A.nodeid else
    dbo.UFN_SEPARATES_COLUMNS(A.pathh, 3, '/') END as level2,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 3, '/') = '' THEN NULL ELSE (
    SELECT 02.Shortdescription from [TrainingMSBI].hr.orgunit 02 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 3, '/')) = 02.
    nodeid) END as level2Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 4, '/') = '' then A.nodeid else
    dbo.UFN_SEPARATES_COLUMNS(A.pathh, 4, '/') END as level3,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 4, '/') = '' THEN NULL ELSE (
    SELECT 03.Shortdescription from [TrainingMSBI].hr.orgunit 03 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 4, '/')) = 03.
    nodeid) END as level3Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 5, '/') = '' then A.nodeid else
    dbo.UFN_SEPARATES_COLUMNS(A.pathh, 5, '/') END as level4,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 5, '/') = '' THEN NULL ELSE (
    SELECT 04.Shortdescription from [TrainingMSBI].hr.orgunit 04 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 5, '/')) = 04.
    nodeid) END as level4Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 6, '/') = '' then A.nodeid else
    dbo.UFN_SEPARATES_COLUMNS(A.pathh, 6, '/') END as level5,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 6, '/') = '' THEN NULL ELSE (
    SELECT 05.Shortdescription from [TrainingMSBI].hr.orgunit 05 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 6, '/')) = 05.

```

```

nodeid) END as level5Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 7, '/') = '' then A.nodeid else
dbo.UFN_SEPARATES_COLUMNS(A.pathh, 7, '/') END as level6,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 7, '/') = '' THEN NULL ELSE (
SELECT 06.Shortdescription from [TrainingMSBI].hr.orgunit 06 WHERE
CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 7, '/')) = 06.
nodeid) END as level6Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 8, '/') = '' then A.nodeid else
dbo.UFN_SEPARATES_COLUMNS(A.pathh, 8, '/') END as level7,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 8, '/') = '' THEN NULL ELSE (
SELECT 07.Shortdescription from [TrainingMSBI].hr.orgunit 07 WHERE
CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 8, '/')) = 07.
nodeid) END as level7Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 9, '/') = '' then A.nodeid else
dbo.UFN_SEPARATES_COLUMNS(A.pathh, 9, '/') END as level8,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 9, '/') = '' THEN NULL ELSE (
SELECT 08.Shortdescription from [TrainingMSBI].hr.orgunit 08 WHERE
CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 9, '/')) = 08.
nodeid) END as level8Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 10, '/') = '' then A.nodeid
else dbo.UFN_SEPARATES_COLUMNS(A.pathh, 10, '/') END as level9,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 10, '/') = '' THEN NULL ELSE (
SELECT 09.Shortdescription from [TrainingMSBI].hr.orgunit 09 WHERE
CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 10, '/')) = 09.
nodeid) END as level9Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 11, '/') = '' then A.nodeid
else dbo.UFN_SEPARATES_COLUMNS(A.pathh, 11, '/') END as level10,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 11, '/') = '' THEN NULL ELSE (
SELECT 010.Shortdescription from [TrainingMSBI].hr.orgunit 010 WHERE
CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 11, '/')) = 010.
nodeid) END as level10Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 12, '/') = '' then A.nodeid
else dbo.UFN_SEPARATES_COLUMNS(A.pathh, 12, '/') END as level11,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 12, '/') = '' THEN NULL ELSE (
SELECT 011.Shortdescription from [TrainingMSBI].hr.orgunit 011 WHERE
CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 12, '/')) = 011.
nodeid) END as level11Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 13, '/') = '' then A.nodeid
else dbo.UFN_SEPARATES_COLUMNS(A.pathh, 13, '/') END as level12,

```



```

CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 13, '/') = '' THEN NULL ELSE (
    SELECT O12.Shortdescription from [TrainingMSBI].hr.orgunit O12 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 13, '/')) = O12.
    nodeid) END as level12Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 14, '/') = '' then A.nodeid
    else dbo.UFN_SEPARATES_COLUMNS(A.pathh, 14, '/') END as level13,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 14, '/') = '' THEN NULL ELSE (
    SELECT O13.Shortdescription from [TrainingMSBI].hr.orgunit O13 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 14, '/')) = O13.
    nodeid) END as level13Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 15, '/') = '' then A.nodeid
    else dbo.UFN_SEPARATES_COLUMNS(A.pathh, 15, '/') END as level14,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 15, '/') = '' THEN NULL ELSE (
    SELECT O14.Shortdescription from [TrainingMSBI].hr.orgunit O14 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 15, '/')) = O14.
    nodeid) END as level14Name,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 16, '/') = '' then A.nodeid
    else dbo.UFN_SEPARATES_COLUMNS(A.pathh, 16, '/') END as level15,
CASE WHEN dbo.UFN_SEPARATES_COLUMNS(A.pathh, 16, '/') = '' THEN NULL ELSE (
    SELECT O15.Shortdescription from [TrainingMSBI].hr.orgunit O15 WHERE
    CONVERT(numeric(8,0), dbo.UFN_SEPARATES_COLUMNS(A.pathh, 16, '/')) = O15.
    nodeid) END as level15Name

FROM (
--zde se vyberou jen ty nejaktualnejsi slozene retezce, zbytek se zahodi
select (SELECT RIGHT(pathh, CHARINDEX('/', REVERSE(pathh))-1)) as nodeid,
    pathh
from tree
WHERE pathh like '0/%') as A
GO

```

---

#### Výpis 2: Ukázka funkce na naplnění hierarchické tabulky

Dalším velkým úkolem bylo nějak vymyslet, jak implementovat dimenzní tabulku obsahující hierarchii celé společnosti. Ze SAPu jsem dostal tabulku, která obsahovala seznam všech různých částí organizace a jejich rodičů v hierarchii. Tuto tabulku jsem musel vzít a předělat tak, aby se z toho dala v SSAS vyklikat dimenze. První problém, který jsme museli vyřešit, byl takový, že hierarchie obsahovala až 15 úrovní a zaměstnanci mohli být uloženi v kterékoli úrovni. Po hledání na internetu jsem nakonec zvolil takový postup, že si vytvořím tabulku o takovém počtu sloupců, jaký má daná hierarchie maximální počet úrovní. Vznikne tabulka, která obsahuje sloupce jako:

Level1, Level2, Level3 ... až Level15. S takovou tabulkou se dá už jednoduše naklikat dimenze v kostce. Prvním krokem byl SQL příkaz (viz výpis 2), který vytvoří pro každý řádek řetězec, kde je daná část společnosti zanořená. K rekurzivnímu projití celé hierarchie jsem použil Common Table Expression (dále jen CTE), který mi vrátil zanoření jednotky v hierarchii. Výsledek tohoto kroku vypadal asi takto: když bychom si vzali například jednotku společnosti, která náleží třetí úrovni: Level3ID/Level2ID/Level1ID. Tento řetězec jsem musel rozdělit a uložit ID jednotlivých úrovní do náležitých sloupců. Tento krok jsem musel dělat ručně pro každý sloupec, protože v SQL jazyce neexistuje alternativa pro smyčku. Dalším stěžejním krokem bylo vyhledání a následné přiřazení ke každé úrovni její název pro zobrazení v reportu. Výsledek tohoto celého procesu můžete vidět na obrázku 7.

level1	level1Name	level2	level2Name	level3	level3Name	level4	level4Name	level5	level5Name	level6	level6Name
1	NULL	49683	GX-ISSU	49687	ISHR	49702	IS-HR-PLHR	49721	GX_HRA	49721	NULL
1	NULL	49683	GX-ISSU	49687	ISHR	49702	IS-HR-PLHR	49720	GX_HRA	49720	NULL
1	NULL	49683	GX-ISSU	49687	ISHR	49699	GX_HRA	49719	IS-HR-ADM	49719	NULL
1	NULL	49683	GX-ISSU	49687	ISHR	49699	GX_HRA	49718	GX_HRA	49718	NULL
1	NULL	49683	GX-ISSU	49687	ISHR	49699	GX_HRA	49717	GX_HRA	49717	NULL

Obrázek 7: Ukázka tabulky výsledné hierarchie

Poslední krok k dokončení hierarchie bylo správně vyklikat dimenzi a napojit ji na faktovou tabulku. Tento konečný proces, týkající se hierarchie, byl snadný.

V tuto chvíli jsem už měl hotovou funkční kostku k HR reportům. Posledním krokem bylo vytvořit .xlsx soubory, které vypadaly podobně jako reporty ze SAP BW a data se musela shodovat.

### 4.3 Ostatní úkoly

Dále jsem ve firmě řešil různé dílčí úkoly, které neměly zdaleka takový rozsah jako již zmíněné projekty. Jedná se o tvoření různých procedur, prohledávání a hledání spojitostí mezi daty, tvoření a úprava tabulek, vytváření SSIS balíčků k ETL procesu a další. V těchto úkolech se jednalo většinou o rutinní záležitosti, se kterými jsem se setkal při větších projektech.

## 5 Uplatněné znalosti a dovednosti získané v průběhu studia

Vzhledem k mojí pracovní náplni ve firmě mi velice prospělo absolvování databázových předmětů jako jsou Úvod do Databázových Dystémů (UDBS) a Databázové a Informační Systémy (DAIS). Předmět UDBS probíhal třetí semestr a dozvěděl jsem se zde základy o relačních databázích, ujasnil jsem si pojmy, které se v tomto odvětví používají a hlavně jsem se zde naučil tvořit i složitější dotazy do databáze pomocí dotazovacího jazyku SQL. Bez znalosti SQL bych ani nebyl přijat do firmy, protože tato znalost je zcela základní podmínkou pro přijetí na mou pracovní pozici a tvoření složitějších dotazů je bráno jako samozřejmost.

Dalším databázovým předmětem na škole bylo DAIS. Obsahem tohoto předmětu bylo procedurální programování pomocí T-SQL a P-SQL. Znalost T-SQL jsem využil v dílčích úkolech, a hlavně když po mě chtěl můj vedoucí projektu nějakou proceduru nebo funkci, tak jsem nebyl vyveden z míry a všechno jsem zvládl.

Dalším předmětem, který mi pomohl k absolvování praxe, jsou předměty týkající se obecně vývoje různých projektů. Jsou to předměty jako Vývoj Informačních Systémů (VIS) nebo také Úvod do Softwarového Inženýrství (SWI), kde jsem se dozvěděl o celém průběhu a jednotlivých fázích vývoje a práci v týmu.

## 6 Znalosti a dovednosti scházející v průběhu praxe

V průběhu praxe mi chyběly znalosti z odvětví Business Intelligence. Tuto nevýhodu jsem zmenšil tím, že jsem během studia v prvním ročníku absolvoval Business Intelligence Academy, kterou pořádala firma Tieto. Myslím si, že absolvování tohoto kurzu mi výrazně pomohlo při výběrovém řízení, protože jsem nebyl v BI úplně nový, ale měl jsem alespoň základní přehled, co to vůbec znamená. Další znalost, kterou mě škola nenaučila, úzce souvisí s Business Intelligence a je to práce s BI nástroji. Konkrétně s nástroji od firmy Microsoft, ale chápu, že tak specifické nástroje se na škole ani nemohou probrat. Myslím, že by mi pomohl předmět, který je až v navazujícím studiu. Jedná se o předmět Business Intelligence and Data Warehouse I, kde se tyto věci okolo Business Intelligence probírají.

## 7 Závěr

Navzdory prvním obavám, které jsem měl, se praxe ukázala jako vynikající pracovní zkušenost. Praxe byla časově dost náročná a někdy bylo obtížné to zkombinovat se školou, ale vše se dalo zvládnout s pomocí kolegů v týmu. Praxe byla výborně zorganizována, bylo mi poskytnuto příjemné a pohodové pracovní prostředí a vstřícný přístup. Co se týče zkušeností a očekávání, tak bakalářská praxe předčila moje očekávání a za těch pár měsíců jsem si mohl vyzkoušet, jaké to je pracovat na projektech z reálného světa v časové tísní v podstatě tak, jak to chodí všude. Na škole jsem si nedovedl představit databázi obsahující víc než stovky řádků, nechápal jsem, co tam ti lidi mohou ukládat a na co potřebují tabulky, kde jsou statisíce až milion řádků. Teď už chápu, že tabulka se statisíci řádky je jedna z těch menších. Dalším zjištěním bylo, že když se ve škole bavili o výkonnosti SQL dotazů a jak tyto dotazy zlepšovat, tak už chápu, že to nebyly jenom plané řeči. Když si spustíte dotaz, který pracuje a spojuje 6 tabulek a tabulky obsahují až milion řádků, tak se dotaz vykonává opravdu dlouho, pokud ho napíšete špatně. Tento fakt mi na škole, kde jsme pracovali maximálně se 100 řádky, nedocházel. Praxe mi dala hodně praktických zkušeností a nových znalostí. Před praxí mě databáze moc nezajímaly a nebavily, měl jsem z nich respekt, ale po praxi musím říct, že práci s databázemi mám rád. Takové změně v mém přístupu k databázím jsem vděčný právě této bakalářské praxi.

Svou bakalářskou praxi hodnotím víc než pozitivně a určitě většinu znalostí uplatním i v budoucím zaměstnání a doufám, že i já jsem byl pro kolegy v týmu platným a užitečným členem.

## Literatura

- [1] Larson Brian. Microsoft SQL Server 2012 Reporting Services, Fourth Edition. : The McGraw-Hill Companies, 2012. ISBN 978-0-07-176047-8.
- [2] Larson Brian. Delivering Business Intelligence with Microsoft SQL Server 2012, Third Edition. : The McGraw-Hill Companies, 2012. ISBN 978-0-070175938-0.
- [3] Richard Tkachuk and Thomas Kejser. SQL Server 2008 Analysis Services Performance Guide [online]. 2008. [cit. 2016-03-21]. Dostupné z: <https://www.microsoft.com/en-us/download/details.aspx?id=17303>
- [4] Multidimensional Expressions (MDX) Reference [online]. June 2012. [cit. 2016-03-21]. ISBN . Dostupné z: <https://msdn.microsoft.com/en-us/library/ms145506.aspx>
- [5] Barry Ralston. PowerPivot for Business Intelligence Using Excel and Sharepoint [online]. 2015. [cit. 2016-03-21]. ISBN . Dostupné z: [www.it-ebooks.info](http://www.it-ebooks.info)
- [6] Patrick LeBlanc. SQL Server 2012 [online]. 2012. [cit. 2016-03-21]. ISBN . Dostupné z: [www.it-ebooks.info](http://www.it-ebooks.info)
- [7] Stora Enso. [online]. [cit. 2016-03-23]. Dostupné z: <http://www.storaenso.com/>
- [8] Software Advice. [online]. 2006-2016 [cit. 2016-03-23]. Dostupné z: <http://www.softwareadvice.com/bi/>
- [9] MSDN. Microsoft MSDN. [online]. 2016 [cit. 2016-03-23]. Dostupné z: <https://msdn.microsoft.com/>