Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky



Business Intelligence v MS SQL 2016 z pohledu začínajícího vývojáře

BAKALÁŘSKÁ PRÁCE

Studijní program: Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Autor: Margarita Tsakunova

Vedoucí bakalářské práce: Ing. Tomáš Církovský

Praha, květen 2019

Prohlášení

Prohlašuji, že jsem bakalářskou práci „Business Intelligence v MS SQL 2016 z pohledu začínajícího vývojáře“ vypracovala samostatně za použití v práci uvedených pramenů a literatury.

V Praze dne 5. května 2019 .....

Margarita Tsakunova

Poděkování

Chtěla bych v první řadě vyjádřit vděčnost vedoucímu své bakalářské práce Ing. Tomášovi Církovskému za cenné rady, věcné připomínky a ochotu při konzultacích a vypracování bakalářské práce. Dále chci poděkovat své rodině a blízkým, kteří mi poskytli podporu a povzbuzení v průběhu celého studia.

Abstrakt

Cílem bakalářské práce je popsání Business Intelligence nástrojů od společnosti Microsoft ve verzi 2016 a představení možností využití těchto nástrojů v podniku z pohledu začínajícího vývojáře.

Práce se skládá ze dvou části – teoretické a praktické. První část se věnuje základům Business Intelligence, které dávají nahlédnout do dané problematiky, počínaje všeobecným přehledem podstaty BI a pojmů s ní souvisejících, po zaměření na technologie, které poskytuje Microsoft SQL Server 2016 pro využití v oblasti BI. Druhá část se zabývá praktickým seznámením s nástroji Microsoft. V rámci této části je čtenáři demonstrován návrh a reálná implementace kompletního BI řešení za pomoci integračních, analytických služeb a Power BI. Celý proces zahrnuje etapy od popisu dat, tvorby datového skladu, přes řešení datových pump a tabulárního modelu až po přípravu reportů ve výše zmíněném Power BI.

Po přečtení této práce by měl čtenář získat představu o možnostech využití Business Intelligence na příkladu produktů MS SQL Serveru.

Předpokladem dosažení cíle práce je získání teoretických znalostí v oblasti Business Intelligence a datových skladů, jejich principů a architektury na základě studia dostupných zdrojů. Těžištěm pro získání dat a informací je odborná literatura a elektronické informační zdroje – odborné články a publikace.

Klíčová slova

Business Intelligence, ETL, datový sklad, MS SQL Server, SSIS, SSAS, Power BI.

Abstract

The aim of this bachelor thesis is to describe the 2016 version of Microsoft Business Intelligence tools and to introduce the possibilities of using these tools in the company from the perspective of a beginning developer.

The thesis consists of two parts – theoretical and practical. The first part deals with the basics of Business Intelligence, that give insight into the issue, starting with a general overview of the nature of BI and related concepts and culminating in focusing on technologies that MS SQL 2016 provides for use in BI. The second part deals with practical introduction to the Microsoft tools. In this part, the reader is demonstrated the design and real implementation of a complete BI solution with the help of integration, analysis services and Power BI. The whole process includes stages from data description, data warehouse creation, all the way through data pump solutions and tabular modeling and finally report preparation in Power BI.

After reading this work, the reader should get an idea of the possibilities of using Business Intelligence on the example of MS SQL Server products.

The prerequisite for achieving the goal of the thesis is to obtain theoretical knowledge in Business Intelligence and data warehousing, its principles and architecture based on the study of accessible resources. The focus for data and information is academic literature and electronic information resources – academic articles and publications.

Keywords

Business Intelligence, ETL, data warehouse, MS SQL Server, SSIS, SSAS, Power BI.

Obsah

Úvod ... 9
1 Databázové systémy ... 10
1.1 Teoretický úvod do relačních databází ... 10
1.1.1 Relace mezi tabulkami ... 12
1.1.2 Normalizace databáze ... 13
2 Business Intelligence ... 16
2.1 Podstata Business Intelligence ... 16
2.2 Komponenty Business Intelligence ... 17
2.2.1 Dočasné úložiště dat ... 18
2.2.2 Operační datový sklad ... 18
2.2.3 Datový sklad ... 19
2.2.4 Datové tržiště ... 20
2.2.5 Extract Transform Load ... 20
2.2.6 OLAP databáze ... 21
2.2.7 Reporting ... 22
2.2.8 Dolování dat ... 23
3 Business Intelligence v MS SQL 2016 ... 24
3.1.1 SQL Server Integration Services ... 24
3.1.2 SQL Server Analysis Services ... 25
3.1.3 SQL Server Reporting Services ... 26
3.1.4 Database Engine ... 28
4 Dimenzionální modelování ... 29
4.1 Faktové tabulky ... 29
4.2 Dimenzionální tabulky ... 30
4.3 Typy databázových schémat ... 31
5 Popis zdrojových dat ... 33
6 Tvorba datového skladu ... 35
7 Nahrání dat do datového skladu ... 42
8 Tvorba tabulárního modelu ... 58
9 Tvorba reportu v nástroji Power BI ... 64
Závěr ... 77
Použitá literatura ... 78

Seznam obrázků

Obr. 1 Koncepce relačního modelu dat (SHAHID, 2018) Poznámka: vlastní úprava 11
Obr. 2 Relace typu 1:1 (vlastní zpracování) 12
Obr. 3 Relace typu 1:N (vlastní zpracování) 13
Obr. 4 Relace typu M:N (vlastní zpracování) 13
Obr. 5 Hlavní komponenty Business Intelligence (POUR et al., 2018) 18
Obr. 6 OLAP kostka (GÁLA et al., 2015) 22
Obr. 7 Propojení služeb Microsoft SQL Serveru (BEK, 2016) 24
Obr. 8 Star schéma (POUR , 2017) 31
Obr. 9 Snowflake schéma (POUR , 2017) 32
Obr. 10 Postup BI řešení pro analýzu zločinů (vlastní zpracování) 34
Obr. 11 Vytvoření databáze (vlastní zpracování) 35
Obr. 12 Konfigurace databáze (vlastní zpracování) 36
Obr. 13 Vytvoření tabulky (vlastní zpracování) 36
Obr. 14 Nastavení vlastností tabulky DimCrimeCategory (vlastní zpracování) 37
Obr. 15 Určení primárního klíče (vlastní zpracování) 37
Obr. 16 Přehled tabulek a jejich vlastností (vlastní zpracování) 38
Obr. 17 Vytvoření diagramu (vlastní zpracování) 38
Obr. 18 Přidání tabulek do databázového diagramu (vlastní zpracování) 39
Obr. 19 Vytvoření vztahu mezi dimenzionální a faktovou tabulkou (vlastní zpracování) .. 39
Obr. 20 Znázornění vztahu mezi dvěma tabulkami (vlastní zpracování) 40
Obr. 21 Výsledný databázový diagram (vlastní zpracování) 40
Obr. 22 Založení nového projektu (vlastní zpracování) 42
Obr. 23 Vytvoření SSIS projektu (vlastní zpracování) 42
Obr. 24 Nový správce připojení (vlastní zpracování) 43
Obr. 25 Přidání Multiflatfile Connection Manageru (vlastní zpracování) 43
Obr. 26 Obecná konfigurace Multiftafile Connection Manageru (vlastní zpracování) 44
Obr. 27 Nastavení datových typů v Connection Manageru (vlastní zpracování) 44
Obr. 28 Přidání úlohy Data Flow Task (vlastní zpracování) 45
Obr. 29 Přidání Flat File Source (vlastní zpracování) 46
Obr. 30 Connection Manager pro prostý soubor (vlastní zpracování) 46
Obr. 31 Úprava sloupců pro tabulku DimDate (vlastní zpracování) 47
Obr. 32 Přidání transformace Derived Column (vlastní zpracování) 48
Obr. 33 Odvození data pomocí transformace Derived Column (vlastní zpracování) 49
Obr. 34 Přidání dalších sloupců do tabulky DimDate (vlastní zpracování) 50
Obr. 35 Přidání sloupce Weekday (vlastní zpracování) 51
Obr. 36 Přidání cíle OLE DB Destination (vlastní zpracování) 52
Obr. 37 Mapování sloupců (vlastní zpracování) 53
Obr. 38 Datový tok balíčku pro tabulku DimDate (vlastní zpracování) 54
Obr. 39 Příkaz ověřující správné nahrání dat do tabulky DimDate (vlastní zpracování) ... 54
Obr. 40 Příklad datového toku pro zbývající dimenzionální tabulky (vlastní zpracování). 55
Obr. 41 Obecná nastavení pro transformaci Lookup (vlastní zpracování) 55
Obr. 42 Získání primárního klíče tabulky DimCrimeCategory (vlastní zpracování) 56
Obr. 43 Výsledný datový tok balíčku pro tabulku FactCrime (vlastní zpracování) 57

Obr. 44 Vytvoření SSAS Tabular Project (vlastní zpracování) 58
Obr. 45 Nastavení Tabular model designeru (vlastní zpracování) 59
Obr. 46 Import dat do SSAS projektu (vlastní zpracování) 59
Obr. 47 Nastavení připojení k MS SQL Server Database (vlastní zpracování) 60
Obr. 48 Úprava tabulek určených pro import (vlastní zpracování) 60
Obr. 49 Úspěšné dokončení procesu importu dat (vlastní zpracování) 61
Obr. 50 DAX funkce pro Homicide metriku (vlastní zpracování) 62
Obr. 51 Tabulární model (vlastní zpracování) 63
Obr. 52 Import dat do Power BI (vlastní zpracování) 64
Obr. 53 Přidání nadpisu (vlastní zpracování) 65
Obr. 54 Vložení obrázku (vlastní zpracování) 65
Obr. 55 Duplikování stránky reportu (vlastní zpracování) 65
Obr. 56 Přidání součtů (vlastní zpracování) 66
Obr. 57 Vložení filtru (vlastní zpracování) 66
Obr. 58 Vložení sloupcového grafu (vlastní zpracování) 67
Obr. 59 Filtr Top 5 zločinů (vlastní zpracování) 67
Obr. 60 Sloupcový graf Top 5 Crimes (vlastní zpracování) 68
Obr. 61 Skládaný sloupcový graf (vlastní zpracování) 68
Obr. 62 Parametry nastavení prstencového grafu (vlastní zpracování) 69
Obr. 63 Čárový graf podle neseřazených měsíců (vlastní zpracování) 69
Obr. 64 Přidání nového sloupce (vlastní zpracování) 70
Obr. 65 Čárový graf podle seřazených měsíců (vlastní zpracování) 70
Obr. 66 První stránka reportu (vlastní zpracování) 71
Obr. 67 Filt Year (vlastní zpracování) 71
Obr. 68 Nepřehledná mapa (vlastní zpracování) 72
Obr. 69 Nastavení mapy (vlastní zpracování) 72
Obr. 70 Přehledná mapa (vlastní zpracování) 73
Obr. 71 Interakce mezi mapou a kartou (vlastní zpracování) 73
Obr. 72 Nejčastější místa zločinů (vlastní zpracování) 74
Obr. 73 Tabulka se seznamem zločinů (vlastní zpracování) 74
Obr. 74 Tabulka výkonnosti policejních jednotek (vlastní zpracování) 75
Obr. 75 Druhá stránka reportu (vlastní zpracování) 75

Úvod

Moderní společnost se nedá představit bez informačních technologií. Počet lidí, kteří mají přístup k internetu se neustále zvětšuje. Množství informací a dat, které se vytvářejí, roste kvantovými skoky. Tento rychlý růst, zejména v rozvojových zemích, vedl IT k tomu, aby se staly základním kamenem každodenního života.

Mnoho společností je vystaveno rozsáhlému toku dat. Pro maximalizaci využití těchto nezpracovaných dat přišlo mnoho podniků se systémem nástrojů, které pracují přímo s příchozími daty a systematizují je nebo transformují za jediným účelem maximalizace efektivity datového úložiště společnosti a následného poskytování výhody na konkurenčním trhu a dlouhodobé stability. Jedním z těchto systémů je BI technologie. Umožňuje společnostem zpracovávat a snadno interpretovat velké množství strukturovaných dat, a tak navrhnout optimální strategii rozvoje podniku a zajistit jeho konkurenceschopnost. Tato technologie podporuje širokou škálu obchodních rozhodnutí, umožňují podnikům identifikovat nové příležitosti a aktivně modifikovat své provozní a strategické politiky založené na dynamickém a neustále se měnícím podnikatelském prostředí.

Cíl práce

Cílem této práce je implementace BI řešení prostřednictvím nástrojů MS SQL Serveru, a to na základě dat o zločinech v Chicagu, která si může čtenář stáhnout z webové stránky uvedené v kapitole 5. To pomůže čtenáři pochopit, jak využít nástroje MS SQL Serveru a zpracovat data tak, aby se dalo je v budoucnu použít pro různé analýzy. Po přečtení této práce by měl čtenář mít jistou představu o tom, jak pracovat s datovými sklady, provádět transformace dat a vytvářet jednoduché reporty za použitím produktů od Microsoft.

Cílova skupina

Tato práce je užitečná zejména pro začínající vývojáře, jelikož demonstruje reálnou implementaci kompletního BI řešení pomocí integračních, analytických služeb a Power BI, včetně technických podrobností využití těchto služeb.

Struktura práce

Bakalářská práce je strukturována tak, aby se do plné míry naplnil cíl této práce. Pro implementaci BI projektu se předpokládá znalost relačních databází, architektury BI a služeb Microsoft SQL Serveru a tato teorie je uvedena v prvních čtyř kapitolách. Praktická část se skládá ze zbývajících pěti kapitol, v nichž je postupně ukázáno praktické využití Microsoft BI nástrojů.

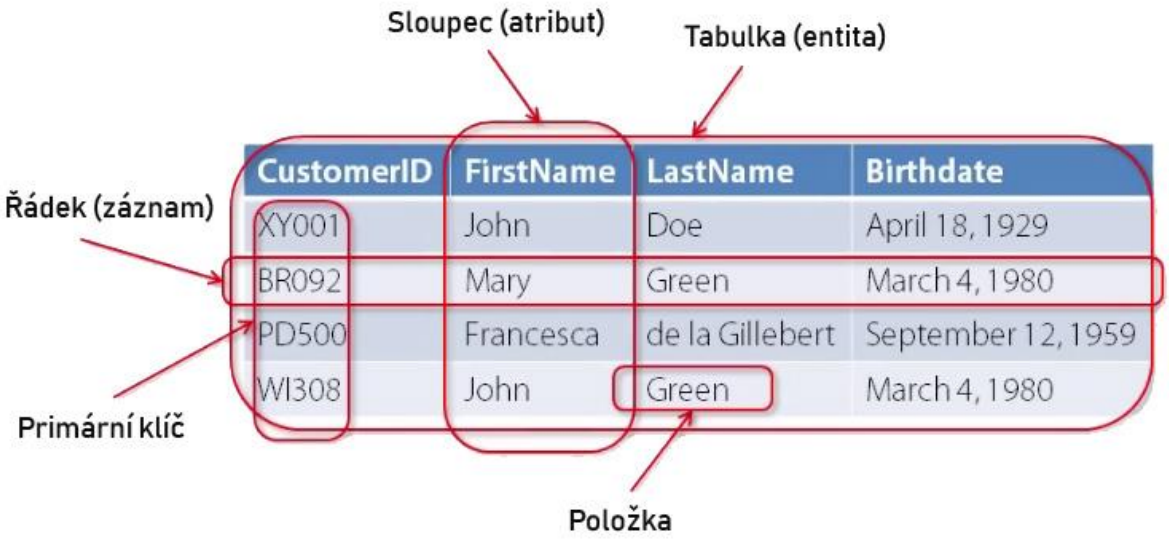
1 Databázové systémy

Pod pojmem *databáze* se rozumí místo, kam se ukládají nezbytná a relevantní data pro podnik. V dnešní době se databáze běžně používají téměř ve všech oblastech, včetně podnikání, vzdělávání, vědy, medicíny, protože databáze umožňují zadávání, zachování a správu datových záznamů. Využití databáze vysvětlí následující jednoduchý a názorný příklad. Firma vlastní svůj sklad, ve kterém se nachází zboží naskladněné a seřazené na základě jednotného systému, např. podle ID zboží, názvu nebo data naskladnění. Data o naskladněném a vyskladněném zboží se evidují do firemní databáze. S růstem objednaného zboží, které firma ukládá do svého skladu, se postupně zvětšuje obsah databáze skladu. Aby mohla firma rychle najít nějaké zboží, potřebuje určitý způsob, respektive určitý klíč, který usnadní vyhledávání daného zboží. Vyhledané zboží lze pak přemísťovat na jiné místo ve skladu, zařadit do nějaké skupiny zboží nebo vůbec ho vyskladnit. Úplně stejně to funguje s datovými záznamy v databázi, která navíc usnadňuje uživatelům práci s daty (VŠE, 2018; ELMASRI a NAVATHE, 2010).

Pojem databáze v informačních technologiích lidé často mylně považují za synonymum pojmu *systém řízení báze dat*, proto je nezbytné tyto dva pojmy rozlišovat. Systém řízení báze dat (též *SŘBD*) je software, který uživatelům pomáhá vytvářet databáze a pracovat s jednou nebo více databázemi. Mezi známé SŘBD patří mimo jiné MySQL, Oracle, PostgreSQL, MS SQL Server. Samotný pojem báze dat představuje data uložená v databázi. Spojení těchto dvou prvků pak lze charakterizovat jako termín databáze (SILBERSCHATZ et al., 2011).

1.1 Teoretický úvod do relačních databází

Většina dnešních SŘBD je založena na relačním modelu, proto se tato sekce zaměřuje na vysvětlení pojmu *relačních databází*. Relační databáze byla navržena Edgarem Coddem kolem roku 1970 a od té doby se stala dominantním databázovým modelem pro komerční aplikace. Relační databáze slouží pro popis zvolené výseči světa. Jako příklad může posloužit databáze určená pro pořadatele běžeckého kurzu, na jejíž základě se postupně vysvětlí základní pojmy relačních databází (viz Obr. 1). V databázi je možné evidovat jednotlivé účastníky kurzu, kteří se běhání věnují rekreačně i závodně. Ke sportovcům kurzu lze evidovat jednotlivé tréninkové skupiny, do kterých patří. Každý prvek reálného světa (např. sportovec, trénink, soutěž) se považuje za tzv. *entitu*, která je stavebním kamenem každé relační databáze. Entita představuje v podstatě tabulku, ve které jsou uložena data, jejich datové typy a vazby s jinými tabulkami. Relační databáze má obvykle více tabulek a tyto tabulky jsou propojeny mezi sebou prostřednictvím vazeb, které se nazývají *relace*. V případě propojení tabulky obsahující údaje o sportovci a tabulky s informacemi o jednotlivých soutěžích je možné pak zjistit z tabulky Soutěž, jakých soutěží se zúčastnil jeden vybraný sportovec (VŠE, 2018; KROENKE a AUER, 2012).



Obr. 1 Koncepce relačního modelu dat (SHAHID, 2018)
Poznámka: vlastní úprava

Základem každé tabulky jsou řádky a sloupce. Sloupce neboli *atributy* představují *„vlastnosti entit“* (VŠE, 2018) a obsahují různé typy informací, například u osoby lze definovat jméno, příjmení, rodné číslo, adresa, datum narození, pohlaví. Každý sloupec jedné tabulky je identifikován názvem, který musí být jedinečným. Sloupec musí navíc obsahovat pouze taková data, která mají stejný datový typ (VŠE, 2018; KROENKE a AUER, 2012).

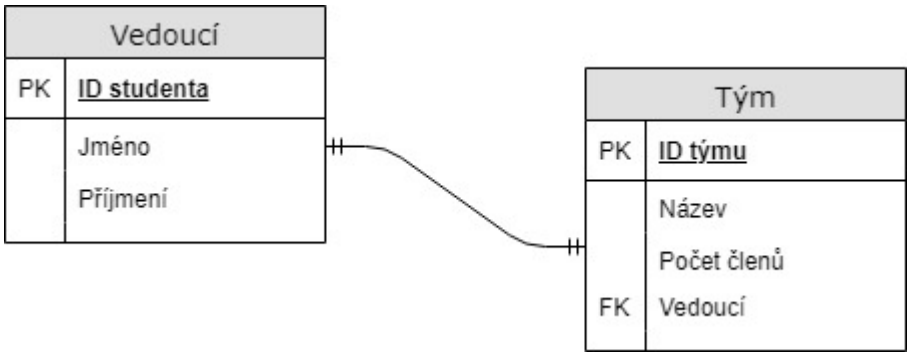
Jeden řádek v tabulce představuje jeden záznam, například záznam o jedné osobě. Řádky by měly obsahovat pouze takové hodnoty, které jsou přípustné pro příslušné sloupce. Jednoduše řečeno, každý z těchto řádků musí dodržovat stejnou strukturu. Kvůli tomu, že každá tabulka zpravidla obsahuje velký počet řádků, je nezbytné, aby měl každý řádek v rámci jedné tabulky tzv. *primární klíč*, který zaručuje jednoznačnou identifikaci jednotlivých záznamů. Dále může tabulka obsahovat *cizí klíč*, který slouží pro poskytnutí spojení mezi dvěma databázovými tabulkami. Pro lepší pochopení si lze vzít tabulku sportovců z předchozího příkladu, ve které jsou zaznamenány údaje o jednotlivých sportovcích, včetně tréninkových skupin, ve kterých sportovci cvičí. Pak cizím klíčem může být ID tréninkové skupiny, které odkazuje do jiné tabulky (VŠE, 2018; KROENKE a AUER, 2012).

Poslední část této sekce se jen krátce věnuje jazyku SQL, který je dnes používán převážnou většinou databází. SQL je *„standardizovaný strukturovaný dotazovací jazyk, který je používán pro práci s daty v relačních databázích“* (PROCHÁZKA, 2009). Původní verze jazyka SQL byla navržena a implementována v 70. letech společností IBM a byla označována jako SEQUEL (*angl. Structured English Query Language*). Nyní je SQL mezinárodně uznávaným databázovým jazykem, který slouží jednak pro tvorbu tabulek, jednak pro manipulaci s daty. Pod základní příkazy jazyka SQL spadají SELECT, INSERT, UPDATE a DELETE (ELMASRI a NAVATHE, 2010).

1.1.1 Relace mezi tabulkami

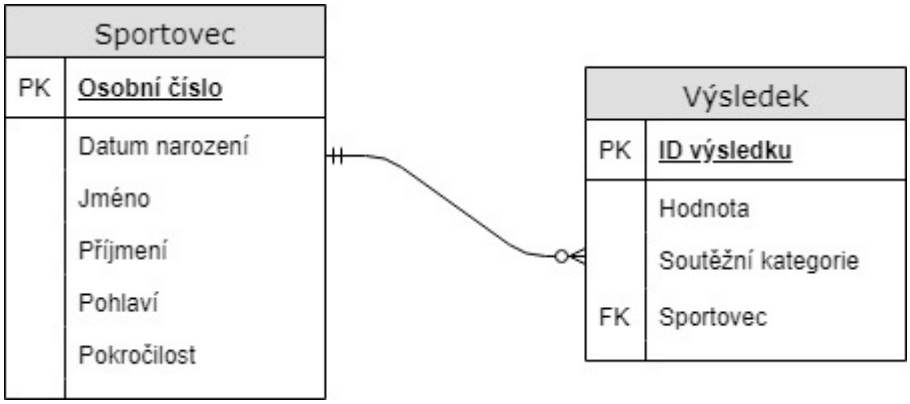
Jak již bylo zmíněno v kapitole 1.1.1, jednotlivé tabulky (*entity*) mají mezi sebou vztah neboli relaci. Tyto relace jsou specifikovány *kardinalitou*, která vyjadřuje *„počet možných kombinací entit, které se mohou účastnit odpovídající sady vztahů“* (ELMASRI a NAVATHE, 2010). V relačních databázích existují celkem tři druhy relací: 1:1, 1:N a M:N. Postupně budou všechny tři relace podrobně vysvětleny (ELMASRI a NAVATHE, 2010).

Nejjednodušší relací je relace 1:1. Tato relace znamená, že *„jednomu záznamu v tabulce odpovídá přesně jeden záznam v druhé tabulce“* (LASÁK, 2010). Jako příklad lze uvést následující případ (viz Obr. 2). Nějaká databáze obsahuje tabulku Vedoucí, která nese informace o studentovi, který je vedoucím týmu v rámci jednoho předmětu. Tabulka zahrnuje ID (případně rodné číslo), jméno a příjmení studenta. ID je primárním klíčem v této tabulce. Druhá tabulka databáze obsahuje data o Týmu: ID týmu, jeho název, počet členů týmu včetně vedoucího a ID vedoucího ve formě cizího klíče. Z tohoto vztahu je zřejmé, že každý tým může mít maximálně jednoho vedoucího a student nemůže být vedoucím dvou týmů najedou. (LASÁK, 2010).



Obr. 2 Relace typu 1:1 (vlastní zpracování)

Relace 1:N přiřazuje *„jeden záznam v první tabulce několika záznamům ve druhé tabulce“* (LASÁK, 2010). Jedná se o komplexnější, ale také mnohem používanější typ vazby při návrhu a tvorbě velkých databází. V praxi se tato relace řeší tak, že např. sportovec může dosáhnout libovolného počtu výsledků, ale jeden výsledek musí být dosažen jen jedním konkrétním sportovcem (viz Obr. 3). V rámci tohoto příkladu, tabulka Sportovec obsahuje informace o sportovci: datum narození, jméno, příjmení, pohlaví, pokročilost. Také obsahuje primární klíč Osobní číslo. Druhá tabulka Výsledek obsahuje následující informace: ID výsledku, hodnota a soutěžní kategorie. Dále obsahuje cizí klíč Osobní číslo sportovce, který odkazuje na primární klíč v tabulce Sportovec. S pomocí tohoto cizího klíče lze pak odvodit, jakého výsledku kdo dosáhl. Výhodou definování tohoto vztahu je, že umožňuje klást otázky typu: „Jakých tří nejlepších výsledků v kariéře dosáhl daný sportovec?“ (LASÁK, 2010).



Obr. 3 Relace typu 1:N (vlastní zpracování)

Poslední relací je relace M:N. Podstata této relace spočívá v tom, že *„více záznamu v první tabulce odpovídá více záznamům v druhé tabulce“* (LASÁK, 2010). Pro práci s této relací v databázi je zapotřebí nová pomocná neboli tzv. *vazební tabulka*. Jeden sportovec může mít libovolný počet tréninků a zároveň každý trénink je pro libovolné množství sportovců, jde tedy o relaci M:N (viz Obr. 4). Tabulka Trénink obsahuje informace o tréninku, tabulka Sportovec – údaje o sportovci. Propojení obou tabulek se realizuje právě prostřednictvím vazební tabulky a primárních klíčů. Vazební tabulka může obsahovat i další atributy charakterizující tuto tabulku. V rámci příkladu o sportovcích by to mohl být například datum zúčastnění tréninku sportovcem (LASÁK, 2010).



Obr. 4 Relace typu M:N (vlastní zpracování)

1.1.2 Normalizace databáze

Tato sekce je zaměřena na problematiku normalizace relačních databází. Obsahem této sekce je tedy pojem normalizace databáze a vysvětlení, v čem spočívá normalizace relační databáze. *Normalizace* je postup, který popisuje, jak správně a efektivně navrhnout a vytvořit databázovou strukturu tak, aby fungovala spolehlivě. Proces normalizace může zahrnovat přidávání více sloupců, vytvoření nových tabulek, definování klíčů a vztahů. (DePaul University).

Proces normalizace dat se skládá z několika kroků a každý krok se nazývá normální forma. Existuje několik normálních forem:

- První normální forma (1NF)

- Druhá normální forma (2NF)
- Třetí normální forma (3NF)
- Boyce-Coddova normální forma (BCNF)
- Čtvrtá normální forma (4NF)
- Pátá normální forma (5NF)

Obvykle se ale vyžaduje dodržení prvních tří normálních forem. Nicméně hlavní nevýhodou normalizace je slabá výkonnost reportingu (AMBLER).

První normální forma (1NF)

První normální forma vyžaduje, aby každá buňka v tabulce obsahovala jednu hodnotu, a nikoliv seznam hodnot, takže nesmí tady existovat žádný vícehodnotový atribut. Data, která mají tuto vlastnost, jsou označována jako *atomická*. 1NF také zakazuje opakované skupiny dat v jednotlivých tabulkách, což znamená, že každý sloupec v 1NF je jedinečný. Lze to ukázat na následujícím příkladu. Zaměstnavatel má přehled všech svých zaměstnanců ve formě tabulky, ve které eviduje jména, věk, adresy zaměstnanců a oddělení, v nichž zaměstnanci pracují. Takto navržená tabulka by však způsobovala spoustu potíží, protože by se špatně vyhledávalo podle příjmení nebo podle města. Proto je nutné, aby položky jméno a adresa obsahovaly pouze atomické hodnoty. Atributy jméno a adresa tedy lze rozdělit na následující části: ID zaměstnance, jméno, příjmení a město, část města, ulice, PSČ. V nově vytvořené jsou všechny atributy již atomické a nejdou dále rozdělit, čímž byly splněny požadavky první normální formy (DePaul University).

Druhá normální forma (2NF)

Tabulka se nachází ve druhé normální formě, pokud splňuje všechny databázové požadavky pro 1NF a *„každý neklíčový atribut tabulky je zcela závislý na celém primárním klíči“* (OTTE, 2013). Pro lepší pochopení se opět lze podívat na příklad. Nějaká firma má databázi, ve které sleduje stavy jednotlivých projektů, na nichž se podílejí zaměstnanci firmy. Databáze pochopitelně obsahuje tabulku Projekt, ve které se eviduje ID projektu, ID projektového manažera, hodinová sazba manažera, název projektu, rozpočet projektu a počet řešitelů. Primární klíč této tabulky tvoří dva sloupce: ID projektu a ID projektového manažera. Hodinová sazba je závislá pouze na ID projektového manažera, proto za účelem splnění podmínky druhé normální formy, je potřeba přemístit hodinovou sazbu do nové tabulky Projektový manažer, kde budou uvedeny pouze údaje o manažerovi (DePaul University).

Třetí normální forma (3NF)

Tabulka je považována za tabulku ve třetí normální formě, když je ve druhé normální formě a zároveň její *„neklíčové atributy jsou na sobě vzájemně nezávislé“* (OTTE, 2013). Jinými slovy, neklíčové sloupce jsou závislé pouze na primárním klíči. Příkladem by mohla být situace, kdy tabulka obsahuje informace o předmětech vyučovaných na konkrétním studijním oboru. Celá databáze pak zahrnuje tabulku Předměty a ta obsahuje: kód předmětu, název, číslo místnosti a kapacitu místnosti. Je zřejmé, že kapacita místnosti je však závislá na místnosti a nemá nic společného s kódem předmětu. Právě tento atribut je nezávislý na primárním klíči. Je nutné tedy oddělit údaje o místnostech do samostatné

tabulky. Takto byla získána nová tabulka s místnostmi. Všemi těmito úpravami byla zajištěna třetí normální forma (DePaul University).

2 Business Intelligence

Tato kapitola si klade za cíl vysvětlit čtenáři základní principy Business Intelligence a její roli v podnicích. V návaznosti na to kapitola poskytuje informace o hlavních komponentách BI řešení.

2.1 Podstata Business Intelligence

Existuje několik různých definic Business Intelligence, ale pro lepší pochopení nejdřív bude uvedena jedna z jednodušších a nejpopulárnějších definic, která zní následovně: *„Business Intelligence, nebo také BI spočívá v tom, že poskytuje relevantní a spolehlivé informace správným lidem a ve správný čas“* s cílem rychlejšího dosažení lepších rozhodnutí na všech úrovních organizace (BOGZA a ZAHARIE, 2008). Pro tyto účely BI vyžaduje metody a programy, které slouží pro shromažďování nestrukturovaných dat, přeměnu těchto dat do hodnotných informací a jejich následnou prezentaci pro podporu rozhodování v různých oblastech podniku. Jinými slovy, Business Intelligence přebírá obrovské množství dat, která jsou vytvářena podniky a zobrazuje je smysluplným způsobem (POUR et al., 2018).

Definice BI, která byla zmíněna výše, se velmi shoduje s názory odborníku v této oblasti. Forrester Research definuje pojem Business Intelligence jako *„soubor metodik, procesů, architektur a technologií, které transformují surová údaje do smysluplných a užitečných informací, které slouží k tomu, aby umožnily efektivnější strategické, taktické a operační pohledy a rozhodování“* (Forrester).

Uvedené definice skvěle zachycují myšlenku, že Business Intelligence znamená víc než soubor nástrojů, které jsou samy o sobě málo hodnotné. Navíc zdůrazňují, že i když pojmy data a informace se často používají zaměnitelně, v rámci BI existuje mezi těmito dvěma slovy výrazný rozdíl. Data jsou surová, neorganizovaná fakta používaná pro výpočet, odůvodnění nebo měření. Data mohou být shromažďována, uložena nebo zpracována, ale jsou něco jednoduchého a zdánlivě náhodného a zbytečného, dokud nejsou uvedena do kontextu. Zatímco informace jsou výsledkem shromažďování a uspořádání dat způsobem, který spojuje datové položky a tím pádem dává datům nějaký smysl nebo význam (Guru99).

BI přispívá ke zlepšení obchodních či výrobních procesů prostřednictvím získání velmi cenných informací, ale pokud jsou tyto informace ignorovány, nepřinesou žádnou hodnotu. I když se jedná o jednoduché koncepty, BI představuje ve skutečnosti docela rozsáhlou a složitou oblast, která zahrnuje analýzu, prediktivní modelování, dolování dat (*data mining*) a textů, performance management neboli řízení a mnohem více. Ale o jednotlivých komponentách BI se bude povídat až v následujících kapitolách (MOUNTAIN, 2015).

Závěrem této sekce je třeba zmínit OLTP systémy a popsat, jaká je jejich role v oblasti Business Intelligence.

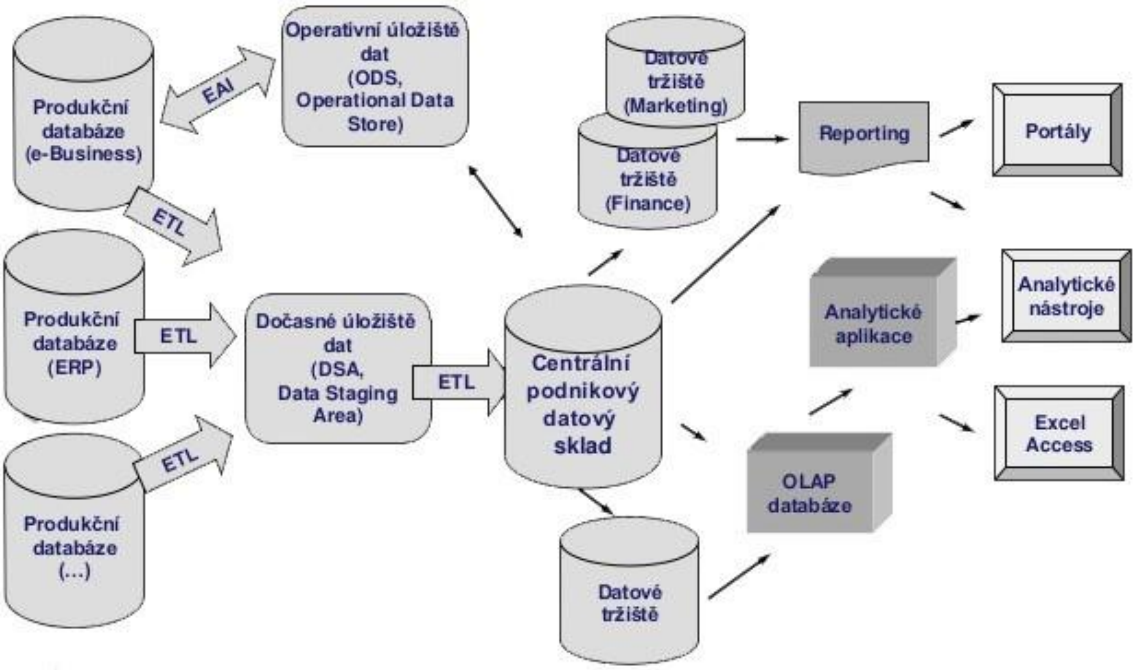
Úspěch celého trhu Business Intelligence byl předem definován skutečností, že velké množství společností shromažďuje obrovské množství dat během jejich každodenních operací. Dnes je těžké si představit podnik nebo organizaci fungující bez systému zpracování transakcí online (*OLTP* či Online Transaction Processing), protože OLTP systémy jsou nezbytné pro to, aby podnik řídil každodenní provoz. Jsou určeny k efektivnímu vykonávání velkého počtu malých online transakcí a spolehlivému přístupu k údajům uloženým v důsledku transakcí. Příkladem takových transakcí jsou obvykle platby obdržené od zákazníků, platby dodavatelům, produkty pohybující se na skladě, přijaté objednávky, poskytované služby a podobně (LACKO, 2013).

V transakčních databázích jsou vytvářená a ukládaná nová data, k nímž má ve stejném čase přístup velké množství uživatelů, kteří s těmito údaji pracují. Proto je kladen velký důraz na velmi rychlé zpracování dotazů, zachování integrity dat v prostředí s více přístupy a efektivitu měřenou počtem transakcí za sekundu. Dotazy OLTP jsou jednodušší a krátké, a proto vyžadují méně času při zpracování a vyžadují méně místa. Z výše uvedených důvodů OLTP databáze vyhovuje pravidlům třetí normální formy (3NF), která umožňuje systému OLTP nezávisle zpracovávat velké množství transakcí a vylučuje redundanci dat, která jsou v OLTP databázích zpravidla vysoce uspořádaná a komplexní (LACKO, 2013).

OLTP je systémem pro správu dat, ale na rozdíl od technologie OLAP nedosahuje vysoké výkony u složitých analýz, které vyžadují sadu jiných technik (LACKO, 2013). Podstata OLAP technologie je vysvětlena v kapitole 2.2.6.

2.2 Komponenty Business Intelligence

Návrh architektury hraje klíčovou roli pro implementaci systému, ale je mnohem důležitější v případě, kdy je na starosti BI řešení. Existuje více standardních komponent BI, jejímiž vlastnostmi se tato sekce bude dále podrobněji zabývat (viz **Error! Reference source not found.**). Komponenty BI vzájemně spolupracují a pokud vznikne v architektuře nějaká mezera, pak se pravděpodobně nedá získat požadované informace a splnit stanovený cíl (POUR et al., 2018).



Obr. 5 Hlavní komponenty Business Intelligence (POUR et al., 2018)

2.2.1 Dočasné úložiště dat

Dočasné úložiště dat či Data Staging Area (*DSA*) je prostor pro ukládání extrahovaných a transformovaných dat, která jsou připravena k načtení do datového skladu. Tato sada procesů se běžně označuje jako *extract, transform and load* (*ETL*). Dočasné úložiště dat datového skladu představuje vše, co je mezi operačními zdrojovými systémy a oblastí pro prezentaci dat, a tedy neposkytuje běžným podnikovým uživatelům dotazovací a prezentační služby. Jak již bylo výše zmíněno, k provedení přípravy dat je zapotřebí samostatné místo nebo komponenta. Nelze přesunout data z různých zdrojů do samotného úložiště datového skladu a teprve pak připravit data. Hlavním důvodem je, že v datovém skladu získáváme data z mnoha různých operačních systémů. Data v datovém skladu jsou předmětově orientovaná a prochází operačními aplikacemi, a proto je pro přípravu dat pro datový sklad nezbytná samostatné dočasné úložiště dat (KIMBALL a ROSS, 2002).

2.2.2 Operační datový sklad

Operační datový sklad neboli Operational Data Store (*ODS*) je součástí datové vrstvy, která slouží k podpoře včasného rozhodování a zkvalitnění a urychlení operativních byznys procesů, a to poskytnutím okamžitých zpráv o operativních výsledcích. Operační datový sklad je integrovaná verze zdrojového systému, který typicky obsahuje aktuální data pocházející z různých zdrojů. Tato integrovaná databáze může také sloužit jako zdroj pro datový sklad, ale nikoliv jako náhrada datového skladu. Před tím, než budou data zařazena do datového skladu, musí být nejprve připravena, což znamená, že jsou upravena pomocí ETL procesu. Pro ODS jsou charakteristické následující vlastnosti:

- ODS obsahuje aktuální hodnoty dat. Toto je nejdůležitější vlastností, která odlišuje operační datový sklad od datového skladu a data martu.

- *„ODS má tendenci se soustředit na operativní požadavky konkrétního byznys procesu (například zákaznické služby)“* (Gartner). Tato vlastnost je velmi podobná ty předchozí a v podstatě znamená, že operační datový sklad neobsahuje historická data, přičemž existuje také možnost, že v některých případech ODS odstraní data po určité době trvání (KIMBALL a ROSS, 2002).

2.2.3 Datový sklad

Datový sklad (angl. *Data Warehouse* či *DWH*) v podstatě představuje jádro systému BI. Datový sklad je vlastně úložiště dat, které je určeno pro centralizaci dat za účelem provádění optimalizované analýzy a reportingu. Jednoduše řečeno, datový sklad funguje jako primární centrální zdroj firemních informací, který využíváme pro získání dat. Tato vlastnost umožňuje uživatelům rychle přistupovat k datům, která sice pochází z různých zdrojů, ale jsou shromážděna na jednom místě, což podnikovým uživatelům usnadňuje využívání kritických dat podle svých potřeb (LACKO, 2013).

Data uložená v datovém skladu obvykle pocházejí z více operačních zdrojů (Oracle databáze, SAP, Microsoft SQL Server atd.). Tato data jsou následně shromažďována a archivována pro dlouhodobé skladování. Účelem datového skladu je poskytnout souhrnné údaje, součty, průměry, trendy apod., které jsou ve vhodném a srozumitelném formátu pro rozhodování. Uživatelé datového skladu nemají zájem, zda např. tento konkrétní zákazník zaplatil svůj účet nebo zda bylo toto konkrétní zboží zakoupeno v daný den. Místo toho se zajímají o obecné trendy a pracují se stovkami či tisíci řádky, aby mohli stanovit, kde lze nejlépe využít úsilí společnosti a kam tato úsilí správně nasměrovat (Dremio).

Další charakteristikou datových skladů je, že ve srovnání s klasickým informačním systémem mohou poskytovat historii velkého množství dat, která jsou v nich uložena. Historická data pomáhají byznys uživatelům analyzovat různé trendy a časové období, aby bylo možné předpovědět budoucí stavy (Dremio).

Co se týče architektury, tak datové sklady jsou postaveny na základě dimenzionálních modelů, které se snaží optimalizovat vyhledávání a získávání informací. Před tím, než budou surová data vložena do datového skladu, je nezbytné je pečlivě vyčistit a formátovat, aby bylo zajištěno, že data jsou správně pojmenována a jsou v konkrétních vztazích se ostatními daty, která jsou již uložena v datovém skladu. Takovou transformaci dat vykonává ETL proces a nástroje pro čištění dat. Teprve po transformaci mají data vysokou kvalitu, což znamená, že jsou vhodné pro spotřebu byznys uživateli, a tedy mohou být používány pro různé analýzy (Dremio).

Dalším aspektem, který ovládá celou strukturu datového skladu a definuje jeho architekturu, jsou metadata. Koncept metadat je docela jednoduchý a spočívá v tom, že metadata jsou v podstatě údaje o datech. Například všechny druhy datových souborů, včetně obrázků, videí, tabulkových procesorů a hudby, mají metadata. V tomto případě metadata jsou informace, které vypovídají o takových věcech, jako, kdo vytvořil daný soubor, kdy byl tento soubor vytvořen, jak je velký a podobně. Jinými slovy, metadata shrnují základní informace o datech, a to zachycením detailů o těchto datech (LOSHIN, 2003).

Z hlediska datového skladu se metadata používají pro definování objektů. Metadata toho docilují pomocí přesného vymezení zdrojů, použití, hodnot, konvencí pojmenování dat v datovém skladu a funkcí, jak lze data měnit a zpracovávat. Metadata hrají důležitou roli v datovém skladu, protože přeměňují data na znalosti a pomáhají řídit a zachovávat datový sklad (LOSHIN, 2003).

Metadata mohou být rozdělena do dvou kategorií: technická a business metadata. Lze si představit, že nějaká databáze obsahuje ID zákazníka. Technická metadata by pak mohla popisovat, v jaké tabulce se toto ID nachází nebo jaký datový typ může mít ID zákazníka. Oproti technickým metadatům business metadata poskytují podnikům smysluplný pohled na data, která ukládají. Z toho lze odvodit, že technická metadata obsahují strukturální informace, zatímco business metadata zahrnují definici podniku, která umožňuje koncovým podnikovým uživatelům snadno pochopit informace uložené v datovém skladu (LOSHIN, 2003).

2.2.4 Datové tržiště

Datové tržiště, anglicky *data mart*, je *„menší podmnožina datového skladu“*, která se obvykle používá ke skladování určité datové sady potřebné pro konkrétní skupinu uživatelů v rámci jedné organizace. Datové tržiště ve srovnání s datovým skladem neintegruje data z celé organizace, ale je postaveno a řízeno jedním oddělením s cílem podpořit své analytické potřeby. Uživatelé tedy nemusí pracovat se zbytečnými daty (LACKO, 2013).

Vzhledem k tomu, že se datová tržiště zaměřují pouze na jednu oblast firemních dat, např. logistiku nebo účetnictví, uchovávají uživatelské dotazy odděleně. Z toho lze odvodit, že datová tržiště lze snadno spravovat a uchovávat, protože se při vytváření oddělených datových tržišť zmenšuje pravděpodobnost, že se podnikoví uživatelé ztratí v hromadě dat (LOSHIN, 2003).

Za další velkou výhodu bych považovala skutečnost, že pokud jeden z podnikových uživatelů v rámci oddělení přichází s novou otázkou, pak se snaží tento problém vyřešit na menší množině relevantních dat. To znamená, že datová tržiště jsou mnohem menší než datové sklady a poskytují odpovědi trochu rychleji a efektivněji, protože nemusí prohledávat velké množství dat. Z toho také vyplývá, že datová tržiště představují samostatné entity: jakákoliv práce provedena jedním uživatelem se svým datovým tržištěm nebude mít žádný vliv na práci s jinými datovými tržišti ostatních uživatelů (LOSHIN, 2003).

2.2.5 Extract Transform Load

Extract, Transform, Load neboli *ETL* je jednou z nejpoužívanějších komponent BI řešení, jelikož zahrnuje v sobě tři funkce, které přispívají k migraci dat z jedné databáze do druhé za účelem podpory rozhodování. Díky svým vlastnostem tento nástroj je rovněž označován jako *datová pumpa* (POUR et al., 2018).

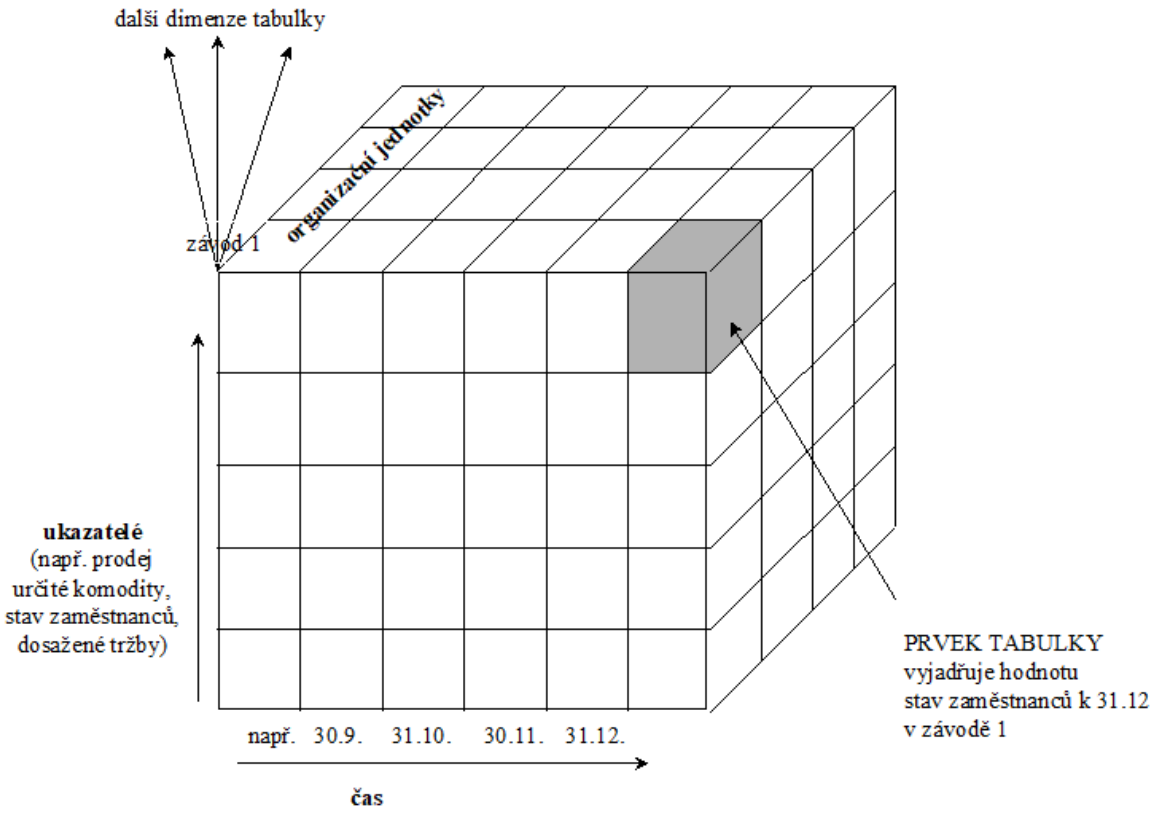
Funkce extrakce představuje první krok v ETL procesu a znamená čtení a porozumění zdrojovým datům a kopírování dat potřebných pro datový sklad do dočasného úložiště dat za účelem další manipulace. Po získání dat z různých operačních systémů a externích zdrojů

je potřeba tato surová operační data transformovat do datového skladu, přičemž data musí být vhodná pro dotazování a spotřebu uživateli. Důvodem je, že extrahovaná data pocházejí z několika různých zdrojů, a proto je třeba je změnit, převést na jiná data a připravit je ve formátu, který bude vhodný pro uložení s cílem dotazování a analýzy. Pro tyto účely poskytuje dočasné úložiště dat místo s řadou funkcí pro čištění dat (oprava chybných hlášení, vyřešení doménových konfliktů, problémy týkající se chybějících prvků nebo převod dat do standardních formátů), kombinování dat z více zdrojů, deduplikace dat a přípravu zdrojových dat pro uložení a použití v datovém skladu. Všechny tyto transformace jsou nezbytné pro to, aby data byla připravena k načtení do oblasti pro prezentaci dat v datovém skladu (KIMBALL a ROSS, 2002).

2.2.6 OLAP databáze

Na rozdíl od OLTP systémů OLAP (*Online Analytical Processing*) systémy jsou navrženy speciálně pro analýzu dat. OLAP nástroje poskytují prostředky pro prezentaci dat pocházejících z datového skladu nebo datového tržiště takovým způsobem, který umožňuje majiteli dat zobrazit srovnávací metriky v různých dimenzích (Microsoft, 2018).

OLAP systémy používají obvykle multidimenzionální struktury dat, které umožňují analytikům a manažerům analyzovat číselné hodnoty z různých hledisek jako je čas, datum, zákazník, produkt, umístění a další. Taková architektura systému umožňuje neustálý přístup k velkému množství dat. Aby to bylo zajištěno, OLAP systémy obvykle předběžně agregují data. Dimenze dat jsou uspořádány do tzv. *OLAP kostky* (viz **Error! Reference s ource not found.**), která umožňuje měnit vnímání dat pomocí alternativních základních dimenzí. OLAP umožňuje uživatelům otáčet vícerozměrná data nebo filtrovat je podle konkrétních hodnot. Tento proces se někdy nazývá „s*licing and dicing"* a pomáhá uživatelům odhalovat trendy a zkoumat data, aniž by museli znát podrobnosti o běžné analýze dat (LOSHIN, 2003; Microsoft, 2018).



Obr. 6 OLAP kostka (GÁLA et al., 2015)

2.2.7 Reporting

Podstatou reportingu je prezentace získaných informací uživatelsky přívětivým způsobem. Pojem *reporting* se odkazuje na vytváření reportů, které poskytují přehled o činnostech společnosti v nejrůznějších podobách: tabulky, diagramy, grafy and mnoho dalších pěkných způsobů vizualizace dat. Reporty mohou být definovány jako dokumenty, které obsahují různé informace shromážděné pro jeden společný účel v upravené podobě (POUR et al., 2018).

Nicméně obchodní činnosti jsou obvykle charakterizovány informacemi a ukazateli ve formě surových dat, která sama o sobě nejsou příliš užitečná. To je důvod, proč business uživatelé potřebují reporting, jehož účelem je vizualizovat data a dávat jim smysl přeměnou surových dat na informace, které mohou být použity pro přijímání byznys rozhodnutí (POUR et al., 2018).

Za účelem, aby mohl uživatel odpovědět na konkrétní otázku týkající se organizace, potřebuje report v užitečném formátu. Zrovna tato otázka přispívá k extrakci dat z databází nebo datového skladu. Z uživatelského pohledu jsou reporty generovány automaticky. Uživatel tedy má pasivní roli, kde stačí pouze interpretovat informace v daném reportu (POUR et al., 2018).

2.2.8 Dolování dat

Pojem dolování dat či *Data mining* je takový segment Business Intelligence, jehož význam v této oblasti velice rychle roste. *„Business intelligence data mining je proces využívání sofistikovaných zařízení k vyhledání a získávání relevantních obchodních informací z většího množství dat“* (HASAN, 2017). Tato forma dolování dat se snaží získat přístup k potřebným datům, která se týkají historie a současného stavu podnikatelské jednotky, a umožňuje logické uspořádání těchto dat. V rámci této problematiky je potřeba alespoň stručně probrat koncepci dolování dat (HASAN, 2017).

Data mining přispívá k extrahování skrytých prediktivních informací z velkých databází. Je to tedy výkonná technologie s velkým potenciálem, která pomáhá společnostem soustředit se na nejdůležitější informace ve svých datových skladech. *„Nástroje pro dolování dat předpovídají budoucí trendy a chování, která podnikům umožňují činit proaktivní rozhodnutí založená na znalostech“* (HASAN, 2017). Dolování dat se používá především k analýze vědeckých a obchodních dat pro odhalení nových vzorů nebo trendů v těchto datech. K tomu používá data mining předpovědní techniky. Výsledné vzory pak mají konečnou roli v procesu rozhodování, protože odhalují oblasti, ve kterých může dojít ke zlepšení v procesu (HASAN, 2017).

3 Business Intelligence v MS SQL 2016

Není pochyb o tom, že jednou z nejdůležitějších věcí pro moderní obchodní společnosti je odhalování vzorů a trendů v jejich datech, jakož i další cenné informace za účelem zlepšení organizačního rozhodování. Nástroje a funkce určené pro výše zmíněné účely musí být interaktivní a poměrně snadno použitelné, protože podnikové uživatelé jsou jediní, kteří musí s nimi pracovat (LACKO, 2013).

Business Intelligence byla a je stále hlavním cílem pro společnost Microsoft, která se snaží, aby se stal SQL Server funkční databázovou platformou pro podnikání. Aby bylo vůbec možné poskytnout organizaci klíčové informace, je důležité mít k dispozici správný BI systém, který by shromažďoval nesourodá data z různých zdrojů, a to po dlouhou dobu a navíc umožňoval společné prohlížení těchto dat. Proto Microsoft nabízí svůj produkt MS SQL Server, který obsahuje kompletní sadu Business Intelligence nástrojů (viz Obr. 7), a tudíž umožňuje uživatelům identifikovat cenná obchodní data, vytvářet reporty a vizuální pomůcky a sdílet data s ostatními uživateli bez jakýchkoliv problémů. Aktuální verze Microsoft SQL Server 2016 nabízí vestavěnou podnikovou platformu připravenou pro BI, která umožňuje více podnikovým uživatelům pracujících ve firmách různých velikostí přijímat rozhodnutí včas. MS SQL Server 2016 obsahuje celou sadu programů pro integraci dat (SSIS), analýzu dat (SSAS), reporting (SSRS) a také database engine (LACKO, 2013).



Obr. 7 Propojení služeb Microsoft SQL Serveru (BEK, 2016)

3.1.1 SQL Server Integration Services

SQL Server Integration Services (SSIS) je nástroj, který je součástí SQL Serveru, sloužící pro budování integrace podnikových dat, jejich migraci a řešení datových transformací. SSIS zahrnuje takové úlohy jako extrahování dat ze zdrojových operačních systémů, čištění dat, dolování dat a jejich standardizace a také nahrání dat do datového skladu. Z tohoto důvodu je vhodné využít SSIS pro řešení složitých obchodních problémů. Jinými slovy integrační služby řídí ETL proces, a proto jsou pak data použitelná pro celou organizaci (Microsoft, 2018).

Integrační služby mohou slučovat data ze široké škály zdrojů dat, i když jsou tyto zdroje odlišné: například ploché soubory, relační zdroje dat nebo XML datové soubory (Microsoft, 2018).

Pro tvorbu a správu ETL balíčků je možné použit SQL Server Data Tools (*SSDT*), kde pak existují dva způsoby, jak vytvářet BI řešení: buď zcela využít grafické nástroje integračních služeb, nebo naprogramovat objektový model integračních služeb. Pro oba tyto účely bude uživatel potřebovat vývojové prostředí Visual Studio, které je vlajkovou lodí vývojových nástrojů Microsoftu (Microsoft, 2018).

3.1.2 SQL Server Analysis Services

Analytické služby se používají k analýze obchodních dat, a to obvykle až po jejich transformaci. Je důležité si uvědomit, že SQL Server Analysis Services představuje analytický datový engine, který podporuje rozhodování a obchodní analýzu. Může ukládat kostky, dimenze, ukazatele a agregáty. Poskytuje tak uživatelům podnikové datové modely pro tvorbu business reportů pomocí klientských nástrojů, včetně SSRS, Excelu a Power BI. K těmto modelům patří multidimenzionální model (klasická OLAP kostka) a tabulární model a je nezbytné vysvětlit, jaký je mezi nimi rozdíl a při jakých situacích se tato řešení používají (Microsoft, 2018).

Tabulární model

Tabulární model (*tabular model*) byl představen v roce 2012 jako součást SQL Serveru a díky svým vlastnostem získal oblibu u profesionálů. Tento model je založen na principu relačního modelování, a proto je považován za snadno pochopitelný model, zejména když se jedná o jeho návrh. Navíc podporuje využití KPI a metrik (Microsoft, 2018; FANCKE, 2017).

Databáze tabulárního modelu může vycházet ze dvou technologií pro ukládání dat: in-memory nebo DirectQuery. Pro in-memory databázi platí, že se importovaná data ukládají do operační paměti počítače, přičemž musí být data před ukládáním komprimována. Takový databázový engine zaručuje rychlejší dobu odezvy a celkově lepší výkon a zároveň vyžaduje dostatečnou paměť (Microsoft, 2018; FANCKE, 2017).

Multidimenzionální model

Multidimenzionální model používá OLAP kostku, která byla důkladně popsána v kapitole 2.2.6, proto stačí vymezit některé vlastnosti, jimiž se odlišuje multidimenzionální model od tabulárního:

- Umí pracovat opravdu s velkým množstvím dat, jelikož drží data na disku
- Na rozdíl od tabulárního modelu podporuje agregaci dat
- Podporuje funkcionalitu zpětného zápisu (*write-back*), která umožňuje uživateli zadávat data přímo do OLAP databáze (Microsoft, 2018; FANCKE, 2017).

Původně SSAS složka zahrnovala OLAP engine založený na multidimenzionálních strukturách. SSAS byla přejmenována na analytické služby po zařazení stroje pro dolování dat. Vzhledem k tomu, že byly zabudovány funkce dolování dat, umožňuje SQL Server Analysis Services objevovat vzory v datech a vztahy, které nejsou okamžitě zjevné uvnitř velkého množství dat (WITHEE, 2010).

3.1.3 SQL Server Reporting Services

Služba SQL Server Reporting Services je koncovým nástrojem při řešením BI projektu, který zákazníci používají pro generování výstupů různými způsoby, ať už je zobrazují ve webovém prohlížeči, na mobilním zařízení nebo v podobě emailu ve své doručené poště. Rolí reportovacích služeb je vytvářet, publikovat a spravovat reporty ve vhodné podobě a následně poskytovat je různým uživatelům ve správný čas tak, aby dokázaly podpořit rozhodování na všech úrovních organizační infrastruktury. Funkčnost reportovacích služeb umožňuje zaměstnancům efektivně přistupovat k datům a vytvářet reporty za použití obsahu z různých zdrojů dat, stejně jako centrálně spravovat zabezpečení a příspěvky (Microsoft, 2018).

Reportovací služby nabízí pro SQL Server 2016 řadu různých typů reportů pro různé případy užití. Existují reporty, které jsou více určeny pro generování a tisk dokumentů (*paginated reports* neboli stránkované sestavy), reporty určené pro využití na mobilním zařízení, které mají více citlivé rozvržení (*mobilní reporty*) a pak i interaktivní průzkumné reporty (Microsoft, 2018).

Tradiční reporty

Tradiční reporty jsou založeny na konceptu, čím je víc dat, tím více stránek bude obsahovat výsledný report. Ty jsou skvělé pro vytváření dokumentů s pevným rozvržením, které lze exportovat do různých formátů, jako jsou PDF, Excel, PowerPoint a další možnosti (Microsoft, 2018).

Takové tradiční sestavy je možné vytvářet uvnitř Report Builderu nebo uvnitř SQL Server Data Tools. Report Builder je univerzální aplikace, která dává firemním uživatelům možnost pracovat v samostatném prostředí. Aplikace není součástí SQL Serveru, ale lze ji získat pouhým stažením z Microsoftu (Microsoft, 2018).

Report Builder je aplikace určená pro analytiky a další podnikové uživatele, ale SSDT je nástrojem, který je dobře známý pro převážnou část lidí, pokud mluvíme o většině vývojových úkolů. SQL Server Data Tools je univerzální nástroj jak pro Business Intelligence, tak pro vývojáře databází, který je založen na Microsoft Visual Studio prostředí. SSDT představuje pouze kolekci šablon projektů, které se instalují do Microsoft Visual Studio. SQL Server Data Tools podporuje vývoj řešení pro Business Intelligence online i offline a umožňuje otevírat, upravovat, zobrazovat, ukládat části reportu a zveřejňovat reporty nejen na SQL Serveru, ale také na SQL Azure. Je nutné dodat, že SSDT není součástí SQL Serveru (Microsoft, 2017).

Mobilní reporty

Vývoj mobilní výpočetní techniky změnil život mnoha lidí. To způsobilo celosvětový nárůst penetrace mobilních telefonů a vznik nových požadavků na podnikové uživatele, jelikož mají dnes lidé jiné reportingové potřeby. Pevný layout reportů zobrazovaných na menších obrazovkách tabletů a telefonů ničí obecný přehled, použitelnost reportu a celkovou zkušenost uživatele. Je zřejmé, že něco, co je navrženo pro použití na široké obrazovce

počítače nemůže být optimálním řešením na malém displeji mobilního zařízení (Microsoft, 2018).

Namísto pevného rozvržení budou podnikové uživatelé pravděpodobně potřebovat responzivní layout s přizpůsobujícími se řádky a flexibilními prvky mobilních reportů, které se dokážou přizpůsobit jakýmkoliv zařízením a různým velikostem obrazovky. Pro ty, kdo potřebují reporty, které poskytují takovou optimální zkušenost při přístupu a prohlížení reportů na mobilních zařízeních, přidala společnost Microsoft nový typ reportu – mobilní reporty. Mobilní reporty jsou součástí získané technologie Datazen, která poskytuje více zkušeností s reporty typu dashboard. Pokud potřebuje uživatel přenést stávající obsah Datazen Serveru (dashboardy nebo KPI) do Reporting Services, pak můžem pro takové účely vyzkoušet SQL Server Migration Assistant pro Datazen (Microsoft, 2018).

Optimalizované mobilní reporty lze rychle vytvářet prostřednictvím aplikace Mobile Report Publisher. Mobilní reporty mají široký sortiment vizualizací, včetně vlastních map, srovnávacích grafů a stromových map. Když podnikoví uživatelé vytvářejí různé vizualizace, Mobile Report Publisher automaticky generuje vzorová data pro každou z těchto vizualizací, takže uživatelé mohou vidět *„jak vizualizace vypadá s použitím ukázkových dat a jaký typ dat dobře vyhovuje pro každou vizualizaci“* (Microsoft, 2018). Navíc uživatel může uložit mobilní reporty na server reportovacích služeb a přistupovat k nim v prohlížeči nebo v aplikacích Power BI určených pro mobilní zařízení. To znamená, že při publikování mobilního reportu je možné se připojit k Power BI Report Serveru prostřednictvím mobilní aplikace Power BI a zobrazovat ten samý report uvnitř mobilní aplikace (Microsoft, 2018).

Webový portál

Webový portál umožňuje prohlížet reporty v libovolném moderním prohlížeči. Na portálu pak lze spravovat, seskupovat a zobrazovat všechny reporty, stránkované sestavy a ukazatele výkonnosti (*KPI*) (Microsoft, 2018).

Reporting Services nyní obsahuje relativně novou funkci pod názvem *branding webového portálu*, která byla poprvé zavedena v reportovacích službách v roce 2016. Tato funkce umožňuje uživatelům měnit vzhled webového portálu, a to tak, že použijí vlastní obchodní značku. Branding existuje ve službě Reporting Services pro to, aby umožňoval uživateli přejít od původního rozvržení k vícebarevnému finálnímu produktu, který by se vyznačoval svou obchodní značkou. Speciální balíček umožňuje skutečně upravovat téma reportovacích služeb, aby odpovídalo požadovanému barevnému schématu v případě, že uživatel nechce základní barevné schéma, které je obvykle dodáváno s produktem. Branding webového portálu také umožňuje přidat obrázek, který se může vztahovat na konkrétní značku nebo společnost (Microsoft, 2018).

Přímo na webovém portálu je možné vytvářet KPI. KPI jsou ukazatele výkonnosti nebo klíčové metriky. Jsou skvělým způsobem, jak rychle získat určitý pohled na podnik, aby vedení firmy dovedlo posoudit, zda je na správné cestě nebo je pozadu. Nyní jsou ukazatele výkonnosti integrovány do reportovacích služeb. Je také možné označovat některé položky jako oblíbené, aby mohl uživatel rychle prohlédnout věci, které jsou pro něj nejdůležitější (Microsoft, 2018).

Obsah na webovém portálu je uspořádán podle typu: KPI, stránkované sestavy, sdílené datové sady a sdílené zdroje dat, které se používají jako základ pro reporty uživatelů. Reporty a dřívější kopie reportů jsou bezpečně uloženy v hierarchii složek. Uživatel může prohlížet složky Report Serveru pomocí webového portálu nebo vyhledávat konkrétní reporty a jejich obecné vlastnosti, které jsou zachyceny v historii reportů (Microsoft, 2018).

3.1.4 Database Engine

Složka Database Engine je pravděpodobně nejznámější součástí SQL Serveru, protože přispívá k ukládání, zpracovávání a zabezpečení dat. Database Engine vytváří běžné relační databáze, což je nejzákladnějším a nejdůležitějším databázovým úkolem. Database Engine pomáhá splňovat požadavky nejnáročnějších aplikací pracujících s podnikovými daty, a to poskytováním řízeného přístupu a rychlého zpracování transakcí. Z toho plyne, že Database Engine představuje základní službu pro vytváření relačních databází, což zahrnuje vytváření tabulek pro ukládání dat a databázových objektů, jako jsou indexy, pohledy, které lze navíc spravovat pomocí aplikace SQL Server Management Studio (Microsoft, 2017).

4 Dimenzionální modelování

V rámci této práce se bude opakovaně odkazovat na dimenzionální modelování, jehož podstata a účel budou vysvětleny dále. Dimenzionální modelování je alternativní technikou, jejíž účelem je logický návrh datového skladu. Na rozdíl od tradičních Entity-relationship modelů, které využívají normalizovaný přístup pro návrh databází, dimenzionální modelování neklade důraz na optimalizaci uložení dat. Naproti tomu datové sklady obvykle akceptují mírnou redundanci dat a nešetří místem, jelikož jde tady především o to, aby práce s nimi byla pochopitelná dokonce i pro netechnicky orientované uživatele. Základním přístupem k modelování dat je již zmíněná srozumitelnost pro uživatele, která evidentně vede ke zvýšení efektivity práce. Proto je dimenzionální modelování mnohem vhodnější pro návrh datamartů nebo datových skladů (KIMBALL a ROSS, 2002; LOSHIN, 2003).

Cílem kapitoly je detailněji popsat podstatu dimenzionálních modelů. Dimenzionální modelování je přímo spojeno s návrhem řešení BI a tedy aplikacemi BI, které jsou typicky vyvíjeny na míru podle potřeb podniku. Díky velkému množství technik, principů a přístupů, které se uplatňují při dimenzionálním modelování, lze pak mluvit o datových skladech, které mimo jiné poskytují rychlou odezvu analytických dotazů, přehledné, snadno rozšiřitelné a uživatelsky přívětivé modely a jednoduché připojení reportingových nástrojů (POUR et al., 2018).

Dimenzionální modelování se skládá především z dimenzí a faktů. To jsou dva typy entit, o nichž se bude dále povídat a je lépe začít vymezením pojmu *dimenze*. Dimenze tvoří převládající většinu všech dat v datovém skladu. Z uživatelského pohledu je možné říci, že dimenze popisuje obchodní události, například prodej, objednávku nebo nákup zboží. Jsou to informace a nejde tady o žádné množství (POUR et al., 2018).

Výsledný dimenzionální model by měl odrážet různé typy otázek, které by byly pro podnik zajímavé. Model nesmí reflektovat jakousi údržbu dat. Základní myšlenkou je návrh modelu tak, aby byl optimální pro lidi, kteří s daty pracují, a aby byla tato data uspořádána odpovídajícím způsobem. Uživatel by mohl se například ptát, jaké jsou prodeje v letošním roce nebo oproti minulému roku (POUR et al., 2018).

4.1 Faktové tabulky

Výsledkem multidimenzionálního modelování jsou databázová schémata, s jejichž pomocí se podporuje rozhodování v podniku a která jsou formulována na základě uživatelských dotazů na datový sklad (LOSHIN, 2003).

Tato schémata umožňují ukládat v datových skladech data a související s nimi operace na principu multidimenzionality. V těchto datových skladech se vyskytují dva typy tabulek: faktové a dimenzionální tabulky. *Faktová tabulka* či tabulka faktů je základním stavebním kamenem dimenzionálního modelování, resp. dimenzionálního modelu a obvykle se nachází uprostřed databázového schématu. Faktové tabulky obsahují takové podstatné

informace o výkonnostních ukazatelích firmy, jako jejich hodnoty. Pro lepší pochopení lze uvést jednoduchý příklad. Vedení logistického podniku, který převáží různý materiál a zboží, chce zjistit finanční zdraví firmy. Pak může sledovat různé finanční ukazatele (tržby, náklady) a výkonnostní ukazatele (počet reklamací, objem prodeje v Kč atd.). Lze si všimnout, že pro popis jednotlivých ukazatelů se používají převážně číselné údaje, které lze agregovat. Numerická fakta považujeme za nejužitečnější, protože mají větší vypovídací schopnost. Jinými slovy, jednotlivé tabulky faktů uchovávají fakta nebo tzv. metriky, které byly stanoveny konkrétní organizací, např. obchodním podnikem, jako klíčové pro analýzu efektivnosti dané organizace (KIMBALL a ROSS, 2002).

Kromě ukazatelů obsahuje navíc faktová tabulka složený primární klíč tvořený cizími klíči nebo vlastní umělý primární klíč. Každá metrika faktové tabulky je identifikována cizím klíčem, který odpovídá primárnímu klíči z dimenzionální tabulky. Cizí klíče jedné tabulky faktů pak jsou vázány na příslušné dimenzionální tabulky v rámci jednoho schématu. Kardinalita vztahů mezi tabulkou faktů a dimenzionálními tabulkami je vždycky 1:N, zatímco samotné dimenzionální tabulky navzájem mají kardinalitu vazeb M:N (KIMBALL a ROSS, 2002).

V souvislosti s tabulkami faktů je třeba zmínit granularitu. Granularita u faktových tabulek určuje úroveň podrobnosti sledovaných ukazatelů. Existuje několik úrovní granularity, kde vysoká granularita znamená, že údaje uložené v tabulce faktů jsou velmi podrobné, což umožňuje získat jejich přesný význam, zatímco při nízké granularitě naopak není možné podrobit údaje detailní analýze. Na rozdíl od nízké úrovní granularity vysoká garnularita nese s sebou výrazně vyšší nároky na kapacity analytické databáze, jelikož přibývá v tomto případě počet záznamů v tabulce v závislosti na definované úrovní. Granularitu je možné aplikovat jak na tabulky faktů, tak i dimenzionální tabulky. Co se týká granularity ve faktové tabulce, tak tady platí pravidlo, že fakta uložená v rámci jedné tabulky musí mít stejnou granularitu. V opačném případě musí být rozdělena do různých tabulek faktů (POUR et al., 2018).

4.2 Dimenzionální tabulky

Hlavním principem dimenzionální tabulky je, že představuje jakési úložiště, kam se ukládají formou atributů informace popisující a doplňující hodnoty ukazatelů. Tyto informace zpravidla obsahují textové detaily o číselných faktech sledovaných ve faktové tabulce, což umožňuje business analytikům lépe porozumět datům a vytvořeným z nich reportům. Například při návrhu datového skladu pro prodej musí dimenzionální tabulka obsahovat takové atributy jako název a typ prodávaného zboží, kdy a z jakého skladu nebo obchodu bude zboží vyzvednuto atd (KIMBALL a ROSS, 2002).

Pro dobře navržený dimenzionální model je typické, že obsahuje tabulky dimenzí s velkým množstvím sloupců nebo atributů. Lze dokonce říci, že dimenzionální tabulky jsou široké, protože mají tendenci osahovat alespoň 50 sloupců. Z toho tudíž vyplývá, že data v dimenzionálních tabulkách jsou denormalizována a popisují řádky uložené v těchto tabulkách. Základní příčina, proč tabulky dimenzí jsou velmi často denormalizovány, se skrývá v účelu těchto tabulek. Denormalizovaná struktura tabulek neslouží ke správě

transakcí, ale je určená pro uživatele, aby jim umožnila snadnou analýzu dat. Tím je zajištěno, že výsledky dotazů mohou být rychle načteny z tabulek dimenzí (KIMBALL a ROSS, 2002).

Stejně tak, jako i faktová tabulka, každá dimenzionální tabulka musí být identifikována jednoznačným identifikátorem, tj. svým primárním klíčem. Ten ale není složený oproti tabulce faktů a zachycuje spojení mezi dimenzionální tabulkou a faktovou tabulkou. Primární klíč pak slouží jako základ pro zachování referenční integrity s jakékoliv tabulkou faktů, ke které samozřejmě musí být připojen. Kardinalita vazby dimenzionální tabulky vůči řádkům ve faktové tabulce je 1:N (KIMBALL a ROSS, 2002).

Nyní je třeba se ještě vrátit k atributům tabulky dimenzí a popsat je trochu podrobněji. Vzhledem k tomu, že atributy v dimenzionálních tabulkách slouží jako hlavní zdroj pro vytváření reportů, hrají v datovém skladů jednu z klíčových rolí. Atributy jsou zodpovědné za to, aby byla data srozumitelná a použitelná pro celou řadu uživatelů, kteří mají velmi odlišné informační a kvalifikační dovednosti. Proto by měli atributy převážně obsahovat bohatou, jasnou a plnou obchodní terminologii, přičemž hodnoty atributů by měli spíše být textové. Navíc je nevhodné používat běžné zkratky nebo standardní kódy různých typů (KIMBALL a ROSS, 2002).

4.3 Typy databázových schémat

Jak již bylo uvedeno v kapitole 4.1, dimenzionální tabulky jsou prostřednictvím cizích klíčů spojeny se svými faktovými tabulkami, avšak propojení tabulek neboli jejich uspořádání může mít dvě základní podoby. Relační dimenzionální model může mít strukturu tzv. STAR schématu (hvězdy) nebo schématu SNOWFLAKE (sněhové vločky). Podstata a využití těchto schémat budou vysvětleny dále v této sekci (POUR et al., 2018).

Pro implementaci datového modelování v datových skladech se nejčastější používá STAR schéma (viz **Error! Reference source not found.**). Faktová tabulka se nachází v centru S TAR schématu a je obklopena příslušnými dimenzionálními tabulkami připojenými prostřednictvím cizích a primárních klíčů. Jednotlivé tabulky dimenzí se vztahují pouze k tabulce faktů a nikoliv jiným dimenzionálním tabulkám. Pojem STAR se používá proto, že grafické znázornění struktury připomíná hvězdu. Níže je uveden obrázek, který demonstruje STAR schéma (POUR et al., 2018).

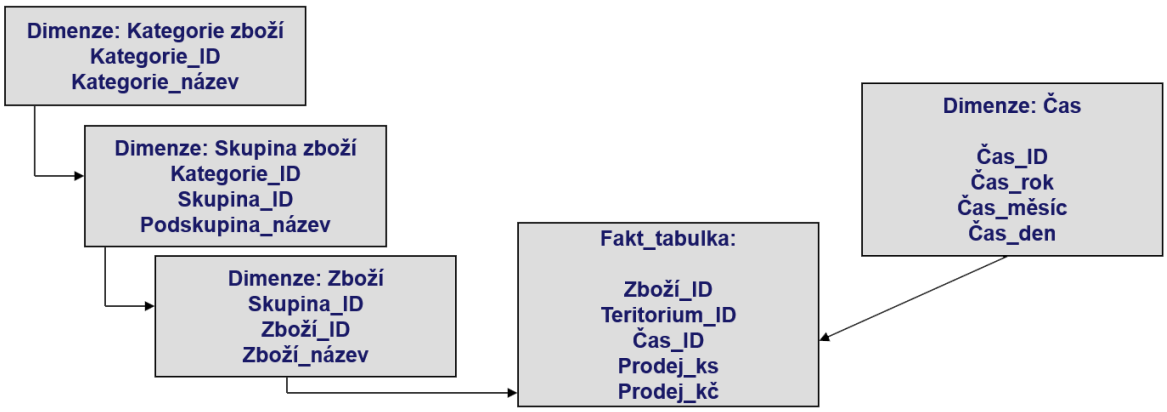


Obr. 8 Star schéma (POUR , 2017)

STAR schéma má následující charakteristiky:

- STAR schéma je považováno za jednodušší, protože má strukturu, které budou uživatelé schopni hned porozumět, což jim umožňuje s tímto schématem pohodlně pracovat.
- Takové jednoduché a předvídatelné uspořádání vede ke zjednodušení procesu extrakce dat, což znamená rychlejší dobu odezvy na klientské dotazy. Příčinou je nutnost pouze jednoho spojení mezi faktovou tabulkou a damenzionální tabulkou (POUR et al., 2018).

SNOWFLAKE schéma je schéma, ve kterém na rozdíl od STAR schématu dochází k normalizaci dat v dimenzionální tabulce , což znamená, že jde tady o dimenzi, která pravděpodobně obsahuje hierarchii atributů daného druhu a tyto atributy byly umístěny do samostatných dimenzí, které byly potom propojeny cizími klíči (POUR et al., 2018).



Obr. 9 Snowflake schéma (POUR , 2017)

Tento princip znázorňuje **Error! Reference source not found.**. Je ale třeba upozornit, ž e schéma sněhové vločky v dimenzionálním modelování není něco, co by mohlo být doporučeno, a to z několika důvodů. Za prvé, SNOWFLAKE schéma komplikuje model. Pokud bude uživatel využívat toto schéma, pak je zřejmé, že skončí s mnohem více dimenzí, které musí být spojeny dohromady, aby uživatel pak mohl pracovat s daty kdykoliv si bude přát. Za druhé, schéma sněhové vločky navíc značně komplikuje práci ETL systému, který musí zachovávat všechny tyto dimenze (POUR et al., 2018).

5 Popis zdrojových dat

Pro vykonání praktické části byla zvolena databáze o kriminálních incidentech ve městě Chicago za období let 2016 až 2018. Data byla získána z datového portálu města Chicago, jehož veřejně dostupná adresa je https://data.cityofchicago.org. Původní data odráží nahlášené zločiny za rok 2001 až do současnosti, ale pro účely bakalářské práce budou data o jednotlivých zločinech omezena pouze na tři roky. Zdrojem základních dat jsou datasety nazvané Crimes – 2016, Crimes – 2017 a Crimes – 2018, které je možné najít na https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data.
Jedná se o data extrahována ze systému policejního oddělení města Chicago, přičemž policejní oddělení prohlašuje, že trestné činy mohou být založeny na předběžných informacích poskytnutých zpravodajskými jednotkami a tyto informace nejsou ověřovány. Za účelem ochrany soukromí obětí trestných činů jsou adresy uvedeny pouze na úrovni bloků a konkrétní místa nejsou identifikována, jelikož se jedná o citlivé osobní údaje.

Zvolená data se skládají ze tří textových souborů a jsou dostupná ke stažení v různých formátech. Vzhledem k tomu, že soubor obsahuje velký počet záznamů, které nelze zobrazit v plném rozsahu v Excelu, byla tato data stažena ve formátu CSV.

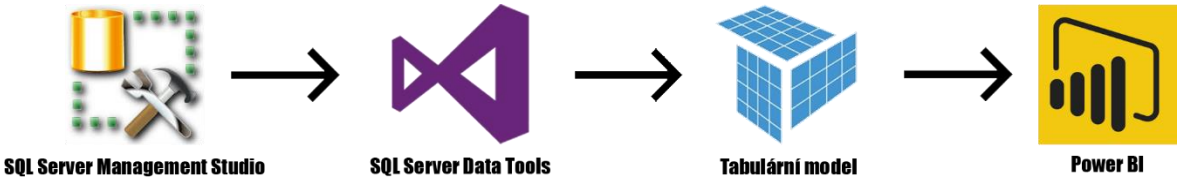
Nyní je možné blíže se seznámit s daty a zjistit základní charakteristiky jednotlivých sloupců, které jsou popsány níže v Tab. 1. Některé sloupce byly ze zdrojových dat vynechány z důvodu jejich nerelevantnosti pro následující analýzu dat a tvorbu reportů.

Tab. 1 Seznam využívaných sloupců a jejich význam (vlastní zpracování)

Sloupec	Popis sloupce
Date	Datum spáchání trestního činu (datum a čas)
Block	Částečně upravená adresa, kde k incidentu došlo
Primary Type	Kategorie trestného činu
Description	Detailní popis trestného činu
Location Description	Popis místa, kde byl spáchán zločin
Arrest	Určení, zda bylo provedeno zatčení pachatele
Domestic	Určení, zda se jedná o domácí násilí
Beat	Část města, kde operuje daná policejní jednotka
Ward	Okrsek v Chicagu
Year	Rok spáchání zločinu
Latitude	Zeměpisná šířka místa incidentu
Longitude	Zeměpisná délka místa incidentu

Pro naplnění cíle této práce projdou data jednotlivými kroky procesu implementace BI řešení, kde budou postupně prezentovány jednotlivé možnosti MS SQL Serveru ve verzi 2016 (viz

). Prakticky bude realizován datový sklad demonstrační databáze, ETL proces a tabulární model. Finální etapa celého procesu spočívá v tvorbě reportu.



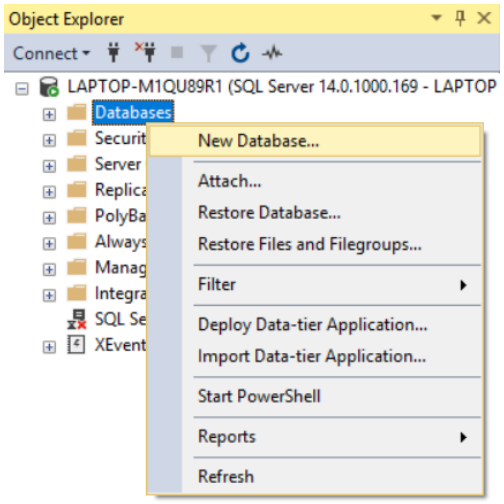
Obr. 10 Postup BI řešení pro analýzu zločinů (vlastní zpracování)

V průběhu tohoto procesu jsou postupně využity nástroje Microsoft. Za účelem realizace BI úlohy je nejdřív vytvořen datový sklad v SQL Server Management Studiu, do kterého se musí nahrát data. Tato data nejsou ale připravena k migraci, a proto prochází ETL procesem, který je realizován v SQL Server Data Tools. Poté po nahrání transformovaných dat do datového skladu se vytváří SSAS Tabular model, pomocí kterého lze připravit data k analýze. Pak data jsou importována z SQL Server Analysis Servises do Power BI, kde se provádí realizace reportů a jejich zveřejnění na web.

6 Tvorba datového skladu

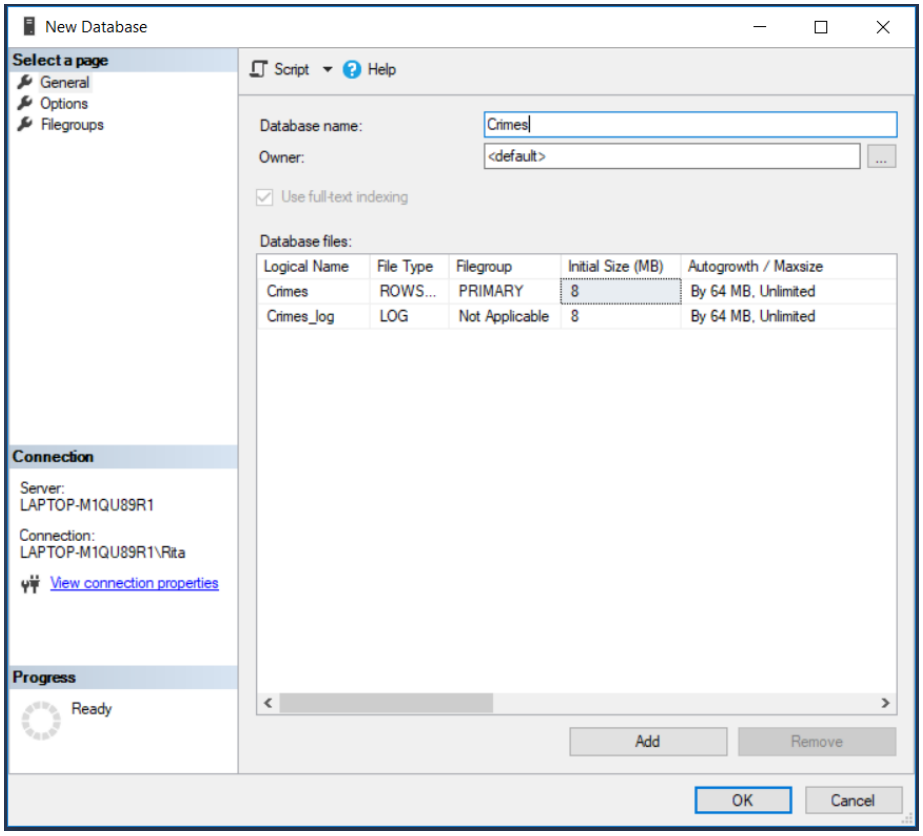
Tato kapitola se zaměřuje na vytváření datového skladu pro ukládání dat o zločinech. Prvním krokem je vytvoření nové databáze datového skladu. Nejprve je nutné rozhodnout, jaké schéma je vhodné použít pro propojení dimenzionálních tabulek s tabulkou faktů. Pro konceptuální model skladu bylo vybráno STAR schéma, protože toto schéma poskytuje několik výhod ve srovnání se SNOWFLAKE schématem. Design STAR schématu je velmi snadno pochopitelný a vyžaduje jen velmi málo joinů pro vytvoření smysluplného dotazu. Vzhledem k jednoduchosti modelu, STAR schéma poskytuje lepší a rychlejší výkon při dotazování dat než normalizovaný model (POUR et al., 2018). Jak již bylo nastíněno, cílem datového skladu je poskytnout uživateli velké objemy dat pro reportování analytických úloh a díky svým vlastnostem STAR schéma pomáhá dosáhnout tohoto cíle.

K vytvoření nového datového skladu lze použít například SQL Server Management Studio (SSMS), což je administrační nástroj pro návrh, správu a ladění databází. Při konfigurování nové databáze je nejprve nutné v okně **Object Explorer** kliknout pravým tlačítkem na položku **Databases**, která se nachází ve stromové struktuře serveru a vybrat z kontextového menu **New database** pro založení nové databáze (viz Obr. 11).



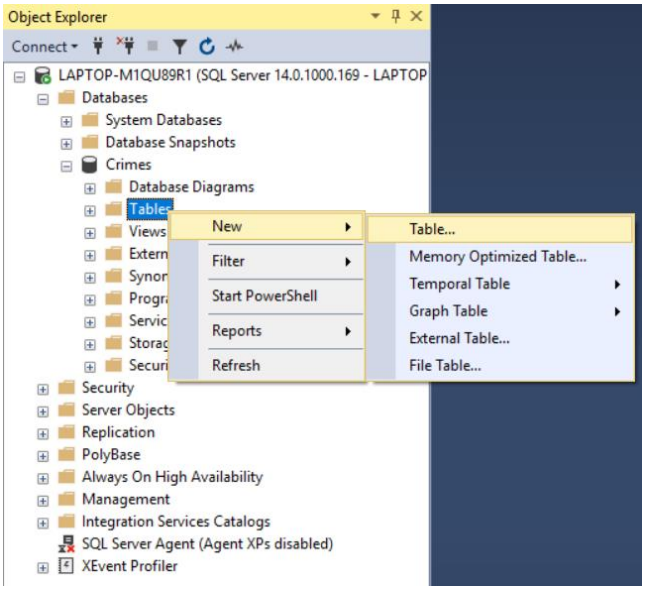
Obr. 11 Vytvoření databáze (vlastní zpracování)

Objeví se vyskakovací okno pro operace nad databázemi (viz Obr. 12). Standardní konfigurace databáze bude pro účely této bakalářské práce stačit, proto jediným nastavením je název databáze. Databázi pojmenovala **Crimes** . Nově vytvořená databáze **Crimes** se po potvrzení zobrazí v seznamu **Databases**.



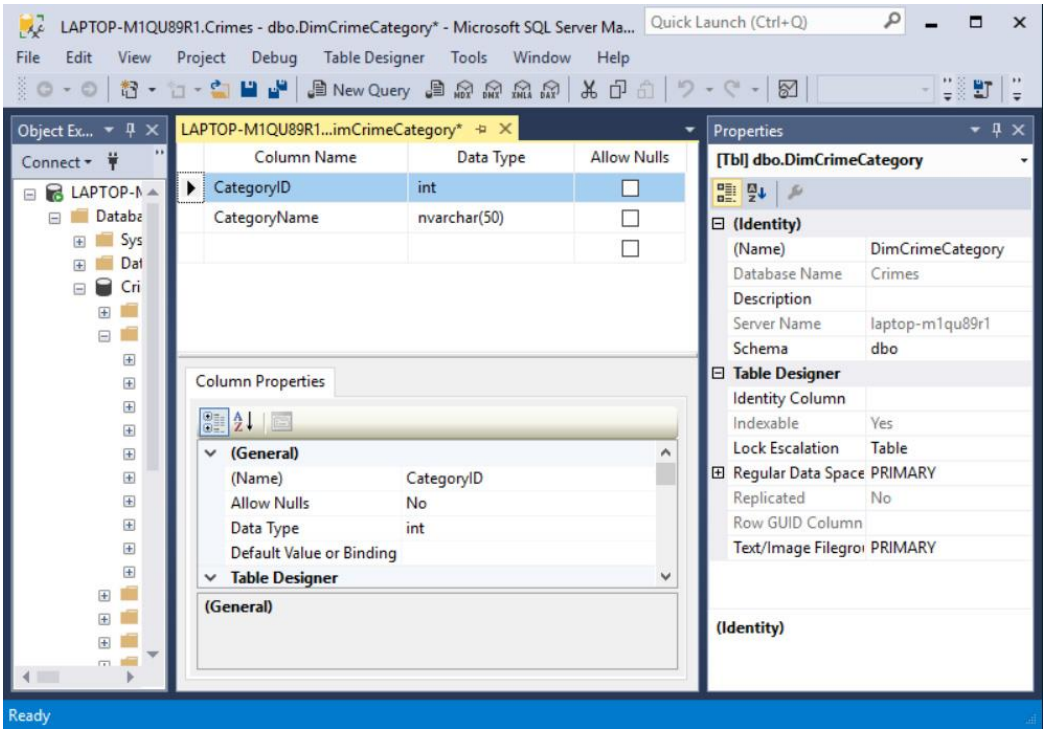
Obr. 12 Konfigurace databáze (vlastní zpracování)

Dalším krokem je návrh cílových tabulek do databáze **Crimes**, kde čtyři tabulky jsou dimenzionální a jedna je tabulka faktů, do nichž se pak v následující kapitole 7 budou nahrávat data. Postup je skoro stejný jako předchozím kroku při vytváření nové databáze. V databázi **Crimes** je nutné zvolit textovou nabídku větve **Tables** a spustit příkaz **New -> Table** (viz Obr. 13).



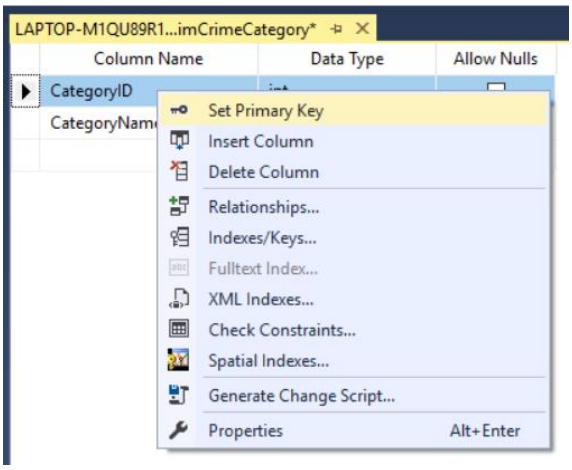
Obr. 13 Vytvoření tabulky (vlastní zpracování)

Objeví se nové okno, které dovoluje nastavit některé vlastnosti vytvořené tabulky (viz Obr. 14). Místo defaultního jména bude tabulka obsahovat požadovanou hodnotu **DimCrimeCategory**. Nyní je možné přidat do této tabulky nové sloupce, tedy vlastnosti, přičemž je nutné zadat název a zvolit datový typ jednotlivých sloupců a navíc rozhodnout, zda u sloupců povolit hodnoty **NULL** (zaškrtnutí políčka **Allows Nulls**).



Obr. 14 Nastavení vlastností tabulky DimCrimeCategory (vlastní zpracování)

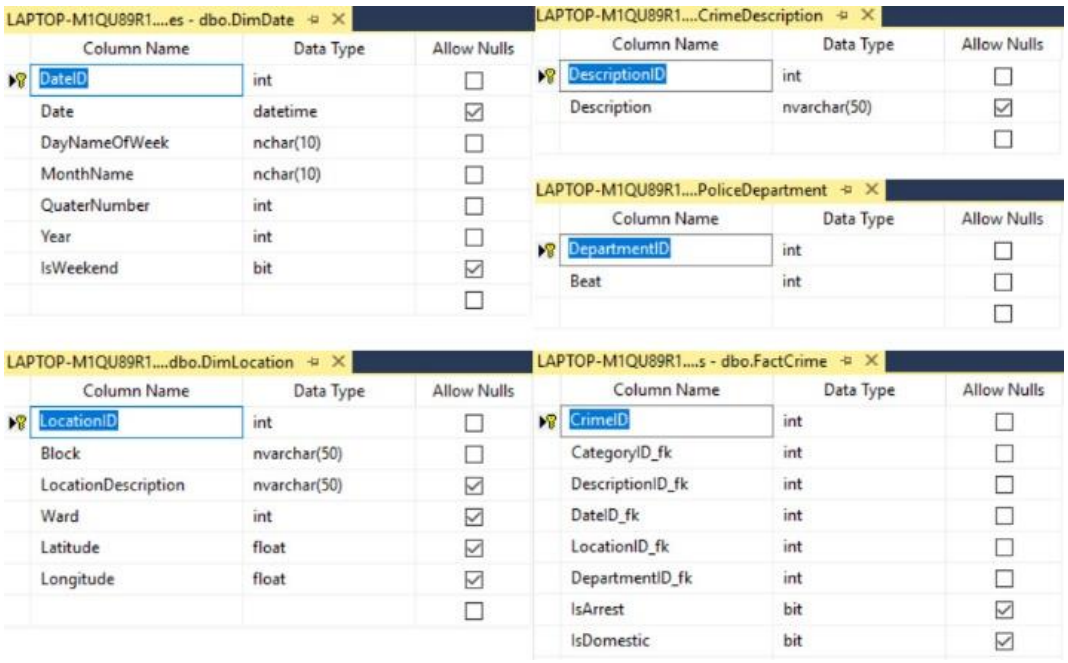
U příslušného sloupce se musí nastavit primární klíč kliknutím na název sloupce a zvolením možnosti **Set Primary Key** (viz Obr. 15).



Obr. 15 Určení primárního klíče (vlastní zpracování)

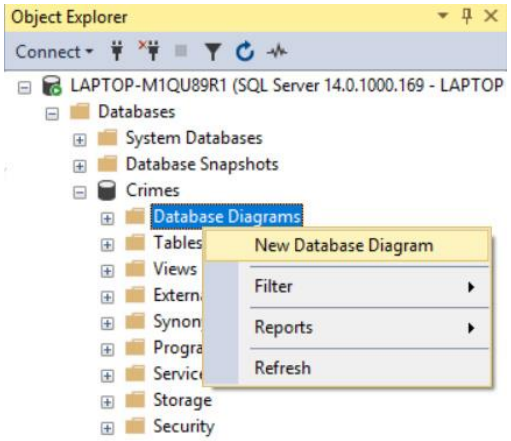
Symbol **Hvězdička** vedle názvu tabulky označuje, že tabulka nebyla uložena od poslední změny, takže tabulku je potřeba uložit, a tímto byla založena jedna tabulku.

Výše uvedené kroky je potřeba zopakovat pro tvorbu zbývajících tabulek **DimCrimeDescription**, **DimDate**, **DimLocation**, **DimPoliceDepartment** a **FactCrime**, jak je zobrazeno níže na Obr. 16.



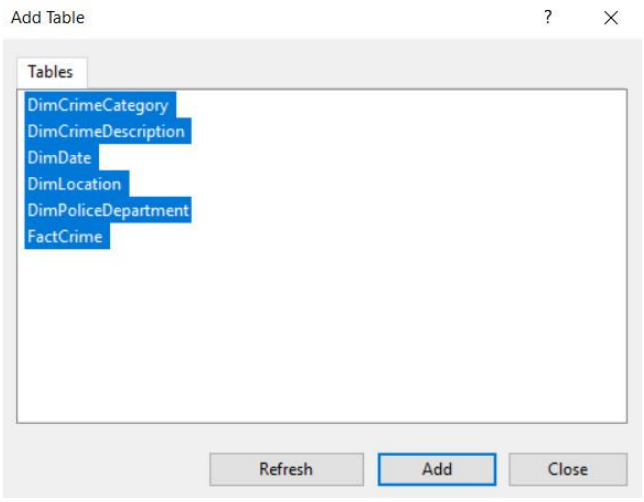
Obr. 16 Přehled tabulek a jejich vlastností (vlastní zpracování)

Dalším krokem je vytvoření vztahu mezi dimenzionálními tabulkami a tabulkou faktů. Tabulka faktů obsahuje cizí klíče jednotlivých propojených dimenzionálních tabulek, což bylo detailně vysvětleno v kapitole 4.1. Proto pro propojení tabulek, tedy zachování referenční integrity mezi tabulkami, nejprve je zapotřebí nový databázový diagram, který musí být vytvořen v rámci databáze **Crimes** kliknutím pravým tlačítkem na položku **Database diagrams -> New Database Diagram** (viz Obr. 17).



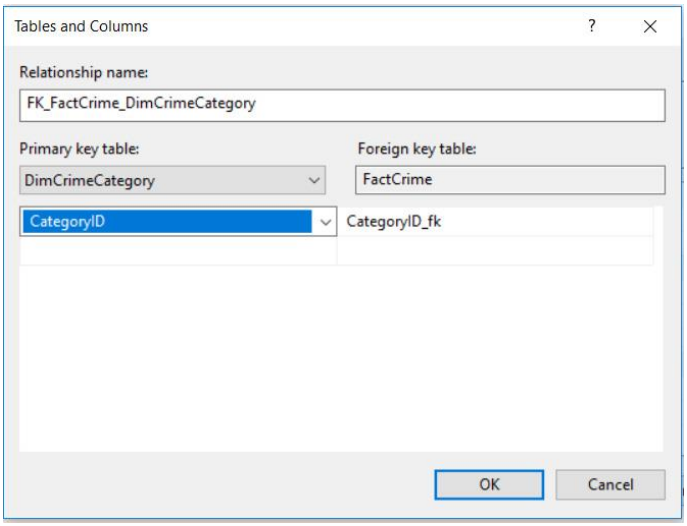
Obr. 17 Vytvoření diagramu (vlastní zpracování)

Otevře se okno se seznamem všech tabulek v databázi (viz Obr. 18). V tomto okně je nutné zvolit ty tabulky, které se pak zobrazí v diagramu, a kliknout na tlačítko **Add**. Po přidání všech požadovaných tabulky lze okno zavřít klepnutím na tlačítko **Close**.



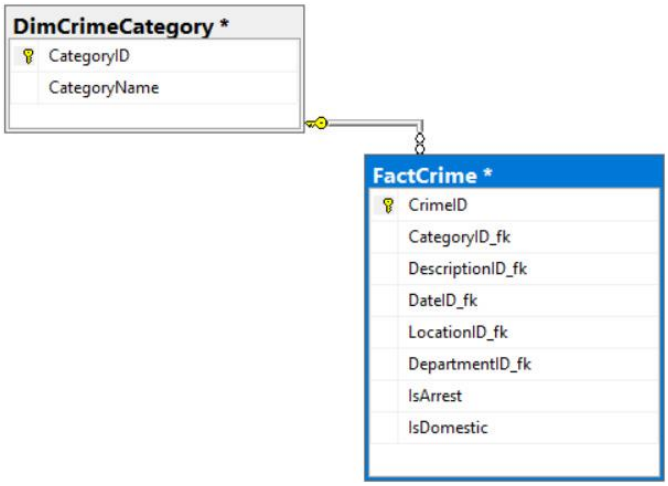
Obr. 18 Přidání tabulek do databázového diagramu (vlastní zpracování)

Po otevření panelu s diagramem je možné začít vytvářet relace mezi vybranými tabulkami. To se udělá přetažením ikonky klíče například z tabulky **DimCrimeCategory** na tabulku **FactCrime**. Okamžitě se zobrazí dvě vyskakovací okna. V prvním otevřeném okně je uveden název vztahu **FK_FactCrime_DimCrimeCategory** (viz Obr. 19). Pokud jsou klíče správně nastaveny, resp. pokud je nastaveno, že se tabulka **FactCrime** pomocí sloupce **CategoryId** připojuje na primární klíč tabulky **DimCrimeCategory**, lze panel zavřít.



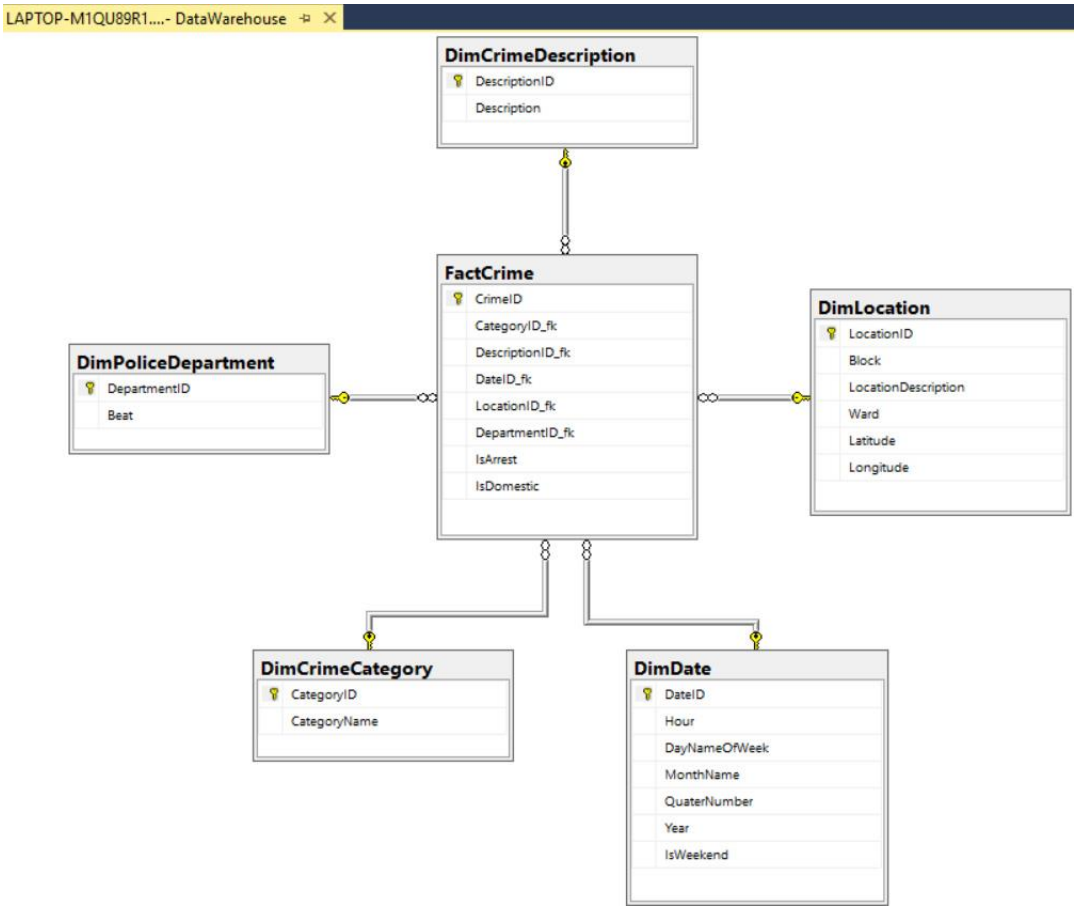
Obr. 19 Vytvoření vztahu mezi dimenzionální a faktovou tabulkou (vlastní zpracování)

Tímto způsobem byl vytvořen vztah mezi tabulkou **DimCrimeCategory** a tabulkou **FactCrime** (viz Obr. 20). Vztah mezi těmito dvěma tabulkami je 1:N.



Obr. 20 Znázornění vztahu mezi dvěma tabulkami (vlastní zpracování)

Pro přidání dalších vztahů mezi faktovou tabulkou a všemi jejími ostatními dimenzionálními tabulkami je nutné zopakovat všechny kroky uvedené v předchozím postupu. Výsledný diagram by pak měl vypadat jako na Obr. 21.



Obr. 21 Výsledný databázový diagram (vlastní zpracování)

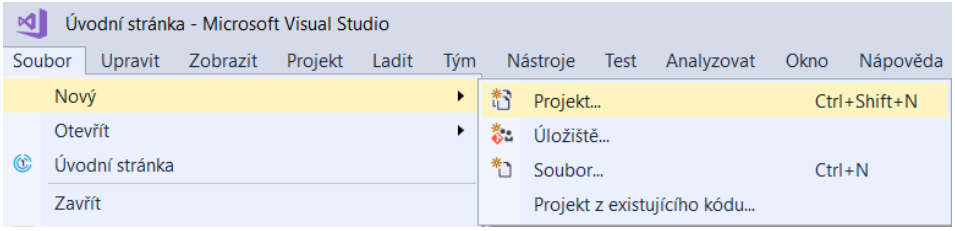
Tato kapitola byla zaměřena na tvorbu STAR schématu v SQL Server Management Studiu. V souvislosti s touto kapitolou byl realizován relační model datového skladu, na jehož základě se bude v následující kapitole vytvořen tabulární model. Nejprve však je nutné

vytvořit ETL balíček k naplnění tabulek datového skladu příslušnými hodnotami, aby byla data připravena pro analýzu a reporting.

7 Nahrání dat do datového skladu

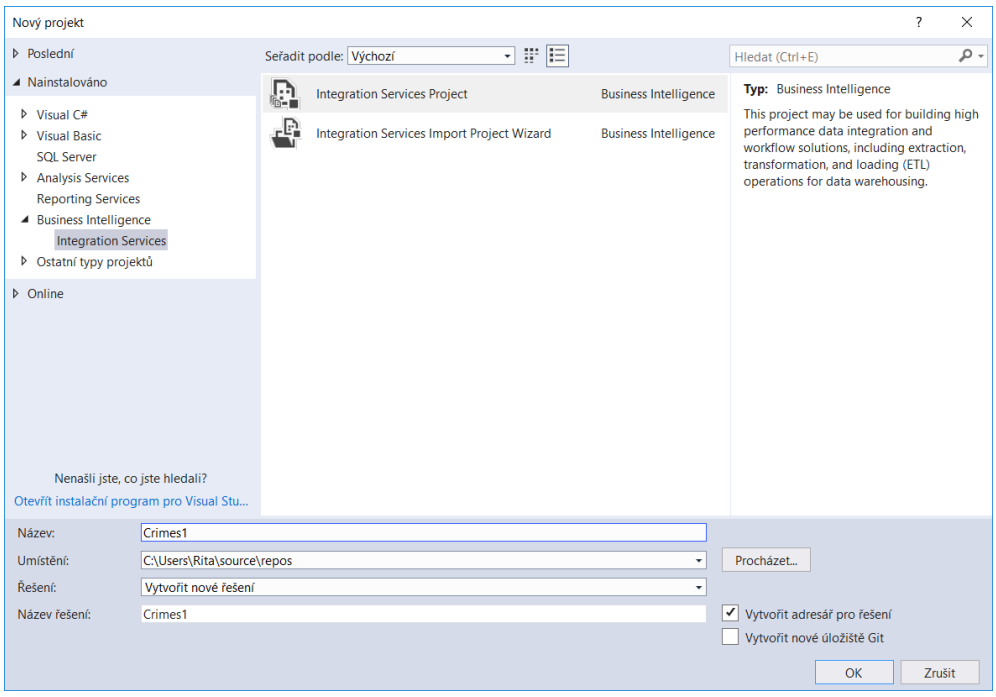
V této kapitole bude prezentován postup tvorby ETL projektu za účelem transformace dat a jejich následné nahrání do datového skladu. Je nutné si uvědomit, že se vytvořený projekt bude skládat z několika balíčků, které pak budou obsahovat jednotlivé procesy extrakce, transformace a načítání dat pro každou tabulku z datového skladu. Pro dosažení cíle této kapitoly, je nutné postupovat podle níže uvedených kroků.

Po spuštění Visual Studio zvolit **Soubor -> Nový -> Projekt** pro založení nového projektu (viz Obr. 22).



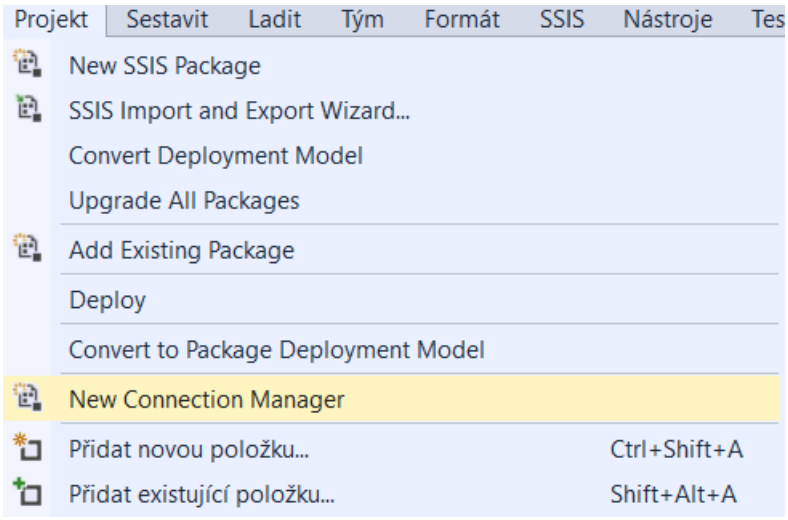
Obr. 22 Založení nového projektu (vlastní zpracování)

V dialogovém okně **Nový projekt** rozbalit větev **Business Intelligence** a vybrat **Integration Services -> Integration Services Project**, jelikož podstata této kapitoly je realizace ETL procesu (viz Obr. 23). Pak jenom zadat název projektu a určit jeho umístění. Dle výchozího nastavení se vytvoří prázdný balíček, který je automaticky přidán do nově založeného projektu ve kterém pak budou probíhat ETL procesy.

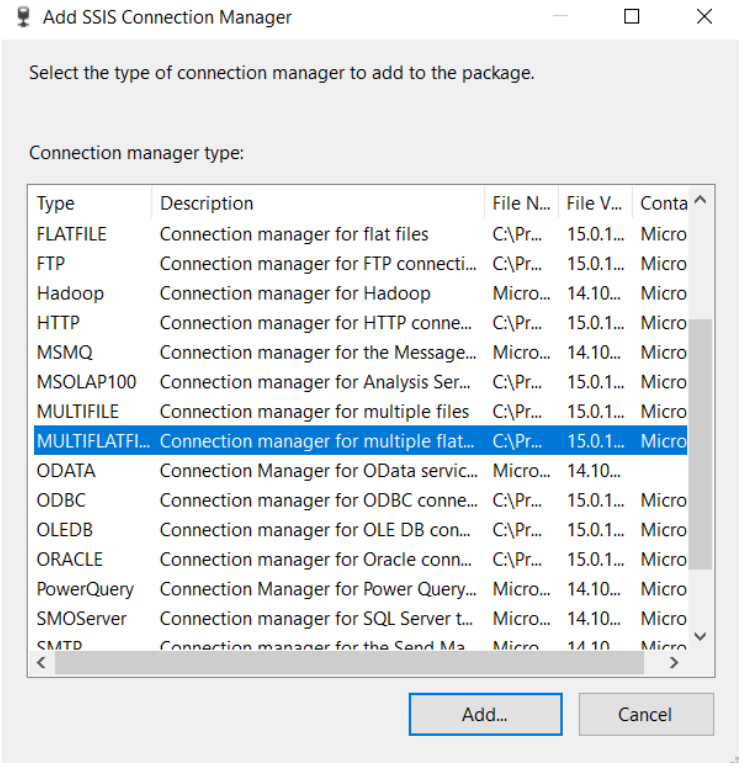


Obr. 23 Vytvoření SSIS projektu (vlastní zpracování)

Aby bylo možné začít pracovat s tímto projektem, je třeba načíst demonstrační data. Kvůli tomu, že tato data se skládají z více prostých souborů, které mají stejný formát dat, stačí přidat a nakonfigurovat **MULTIFLATFILE Connection Manager**. Ten umožňuje extrahovat záznamy ze všech souborů adresáře do jedné tabulky. Prvním krokem je tedy zvolení záložky **Projekt -> New Connection Manager** (viz Obr. 24) a přidání nového správce připojení dvojklikem na možnost **MULTIFLATFILE** (viz Obr. 25).



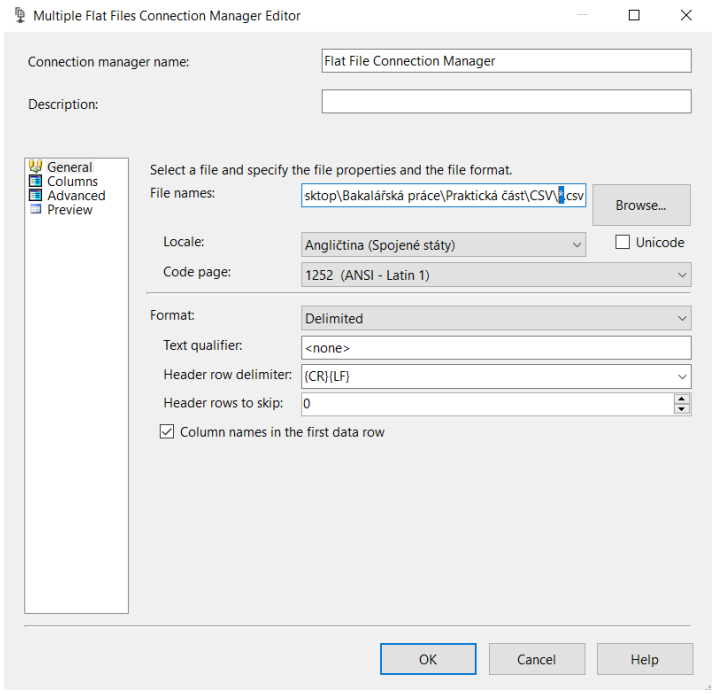
Obr. 24 Nový správce připojení (vlastní zpracování)



Obr. 25 Přidání Multiflatfile Connection Manageru (vlastní zpracování)

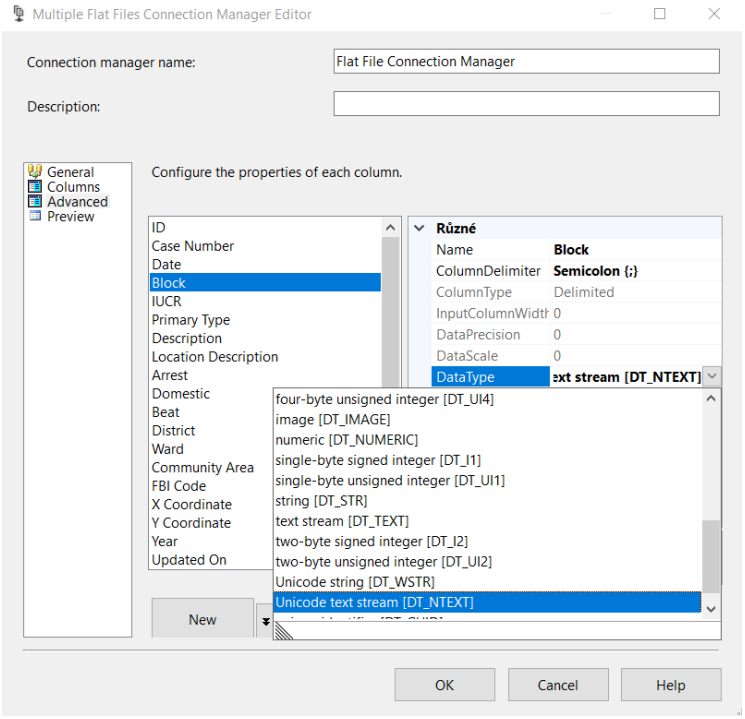
Otevře se okno editoru připojení (viz Obr. 26). Zde je možné mimo jiné zadat cestu k souborům. Nejjednodušším způsobem je kliknout na tlačítko **Browse** a přejít do složky, ve

které jsou soubory umístěny. Pak jenom vybrat libovolný soubor ze složky a nahradit název souboru hvězdičkou, která určuje nahrání více souboru najednou z jednoho adresáře.



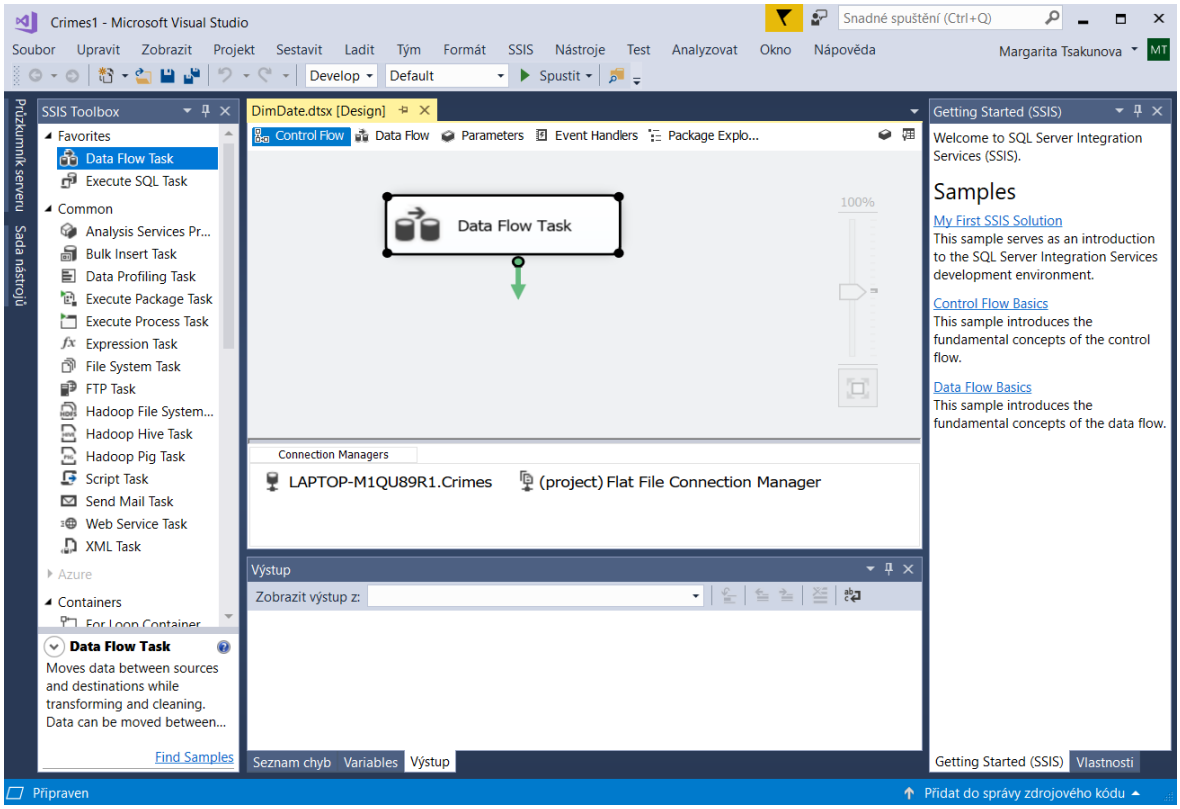
Obr. 26 Obecná konfigurace Multiftafile Connection Manageru (vlastní zpracování)

Na záložce **Advanced** je třeba nastavit správné datové typy sloupců a uložit všechny provedené změny (viz Obr. 27).



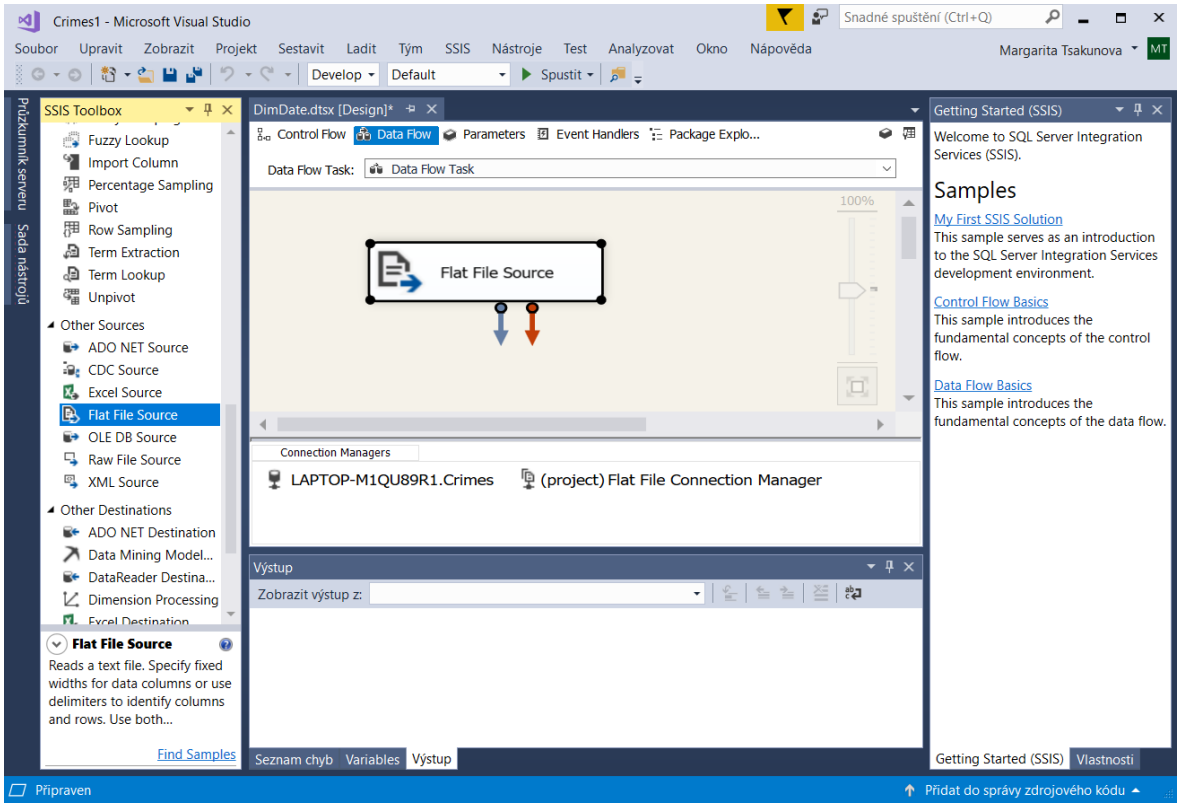
Obr. 27 Nastavení datových typů v Connection Manageru (vlastní zpracování)

Nyní už je možné začít tvořit jednoduchou datovou pumpu, a ta bude určena pro dimenzionální tabulku **DimDate**. Prvním krokem je výběr úlohy **Data Flow Task** z panelu nástrojů **SSIS toolbox** a přidání této úlohy do prostoru **Control Flow** pouhým přetažením (viz Obr. 28).



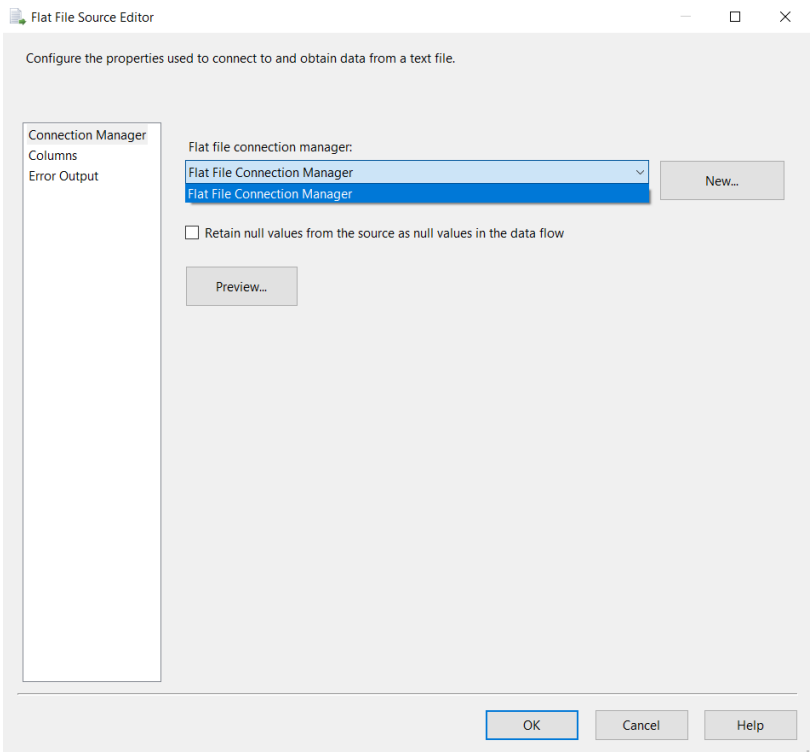
Obr. 28 Přidání úlohy Data Flow Task (vlastní zpracování)

Po dvojkliknutí na **Data Flow Task** se zobrazí **Data Flow prostor**, do kterého je nutné nejdřív přidat a nakonfigurovat **Flat File Source** pro daný balíček (viz Obr. 29). Jak je patrné z obrázku nahoře, při přechodu na plátno datového toku se SSIS toolbox změnil. Všechny úkoly jsou nyní nahrazeny transformacemi, zdroji a cíli určených pro **Data Flow**.



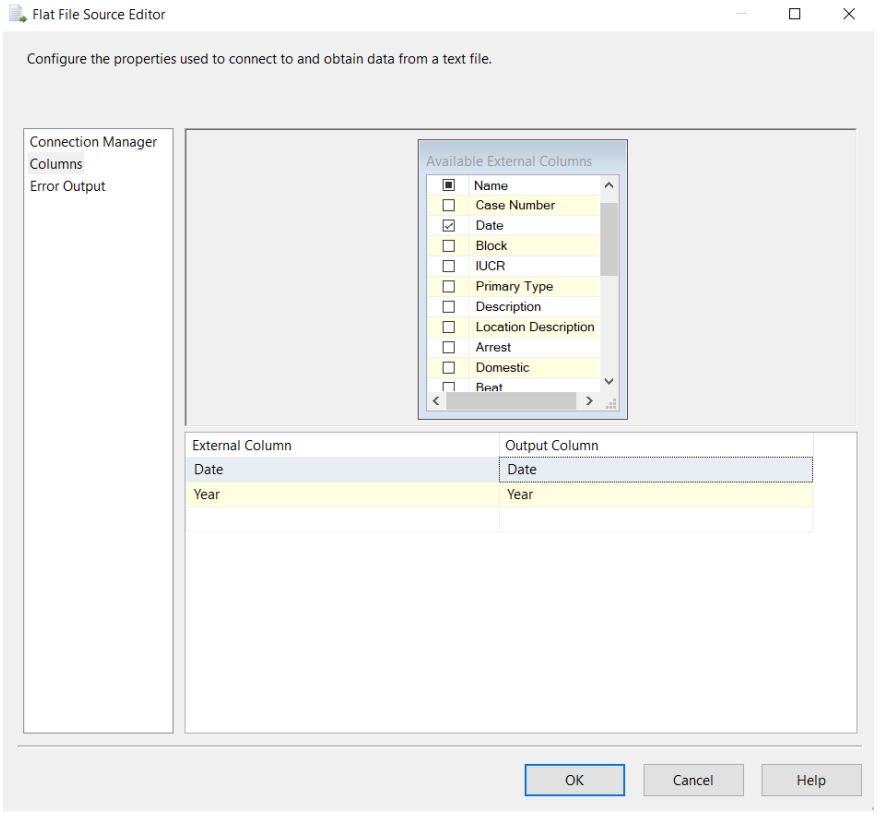
Obr. 29 Přidání Flat File Source (vlastní zpracování)

Dvojklik na **Flat File Source** pro úpravu zdroje a výběr dříve definovaného **Multiflatfile připojení** jako zdroje dat (viz Obr. 30).



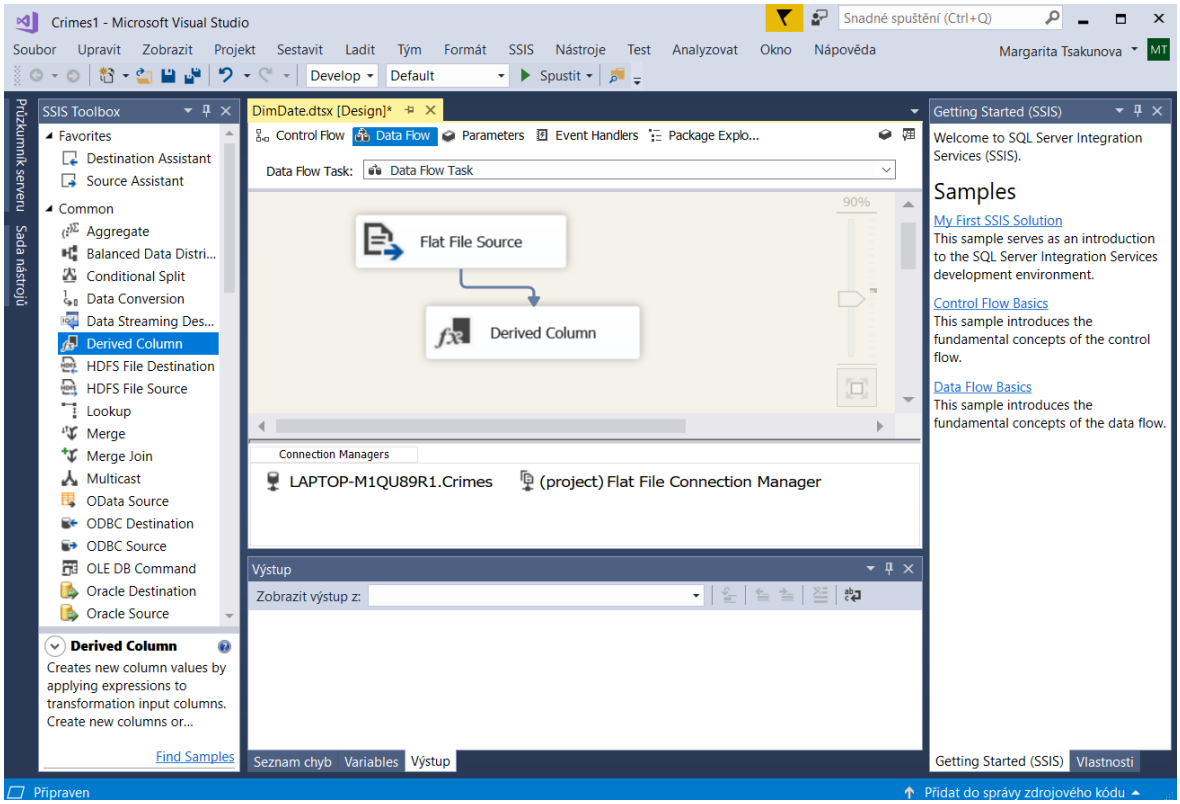
Obr. 30 Connection Manager pro prostý soubor (vlastní zpracování)

Na záložce **Columns** lze zkontrolovat všechny sloupce, které byly nahrány a zrušit výběr zbytečných sloupců (viz Obr. 31). V případě tabulky **DimDate** je třeba odstranit z výstupu všechny sloupce kromě **Date** a **Year**, a tím končí veškeré úpravy související se zdrojem dat.



Obr. 31 Úprava sloupců pro tabulku DimDate (vlastní zpracování)

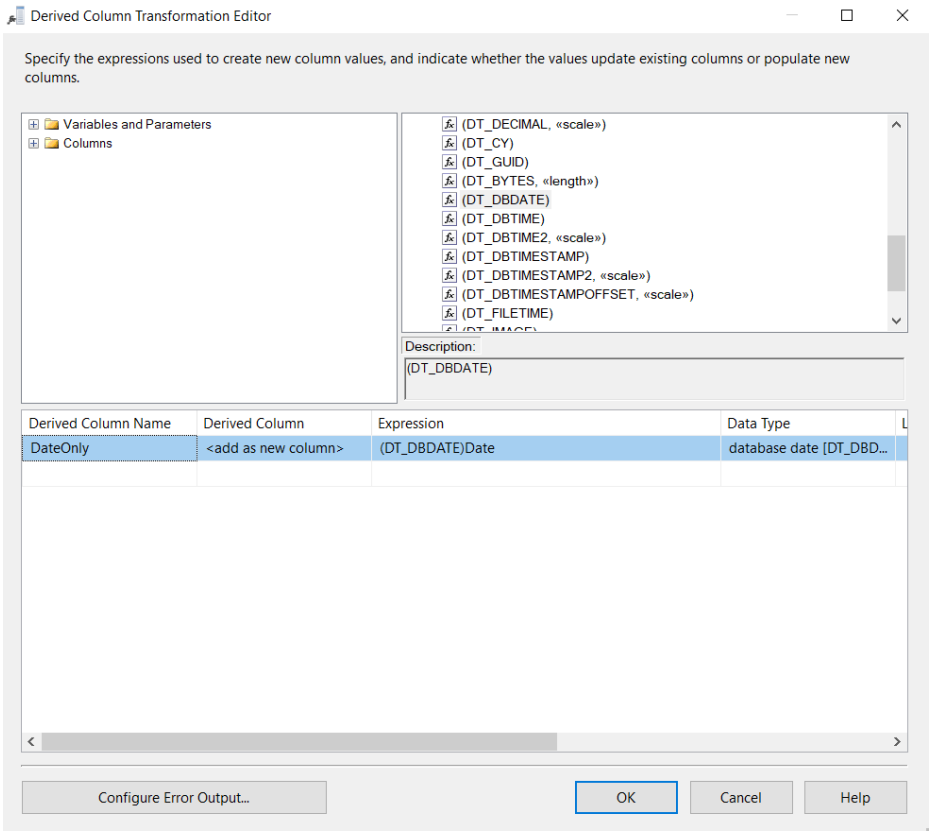
Dále budou popsány různé transformace dat pro tabulku **DimDate** založené na přidání odvozených sloupců. Pro tento účel je třeba přetáhnout **Derived Column** transformaci do **Data Flow prostoru**, propojit ji šipkou s počáteční komponentou **Flate File Source** (viz Obr. 32) a otevřít editor (viz Obr. 33).



Obr. 32 Přidání transformace Derived Column (vlastní zpracování)

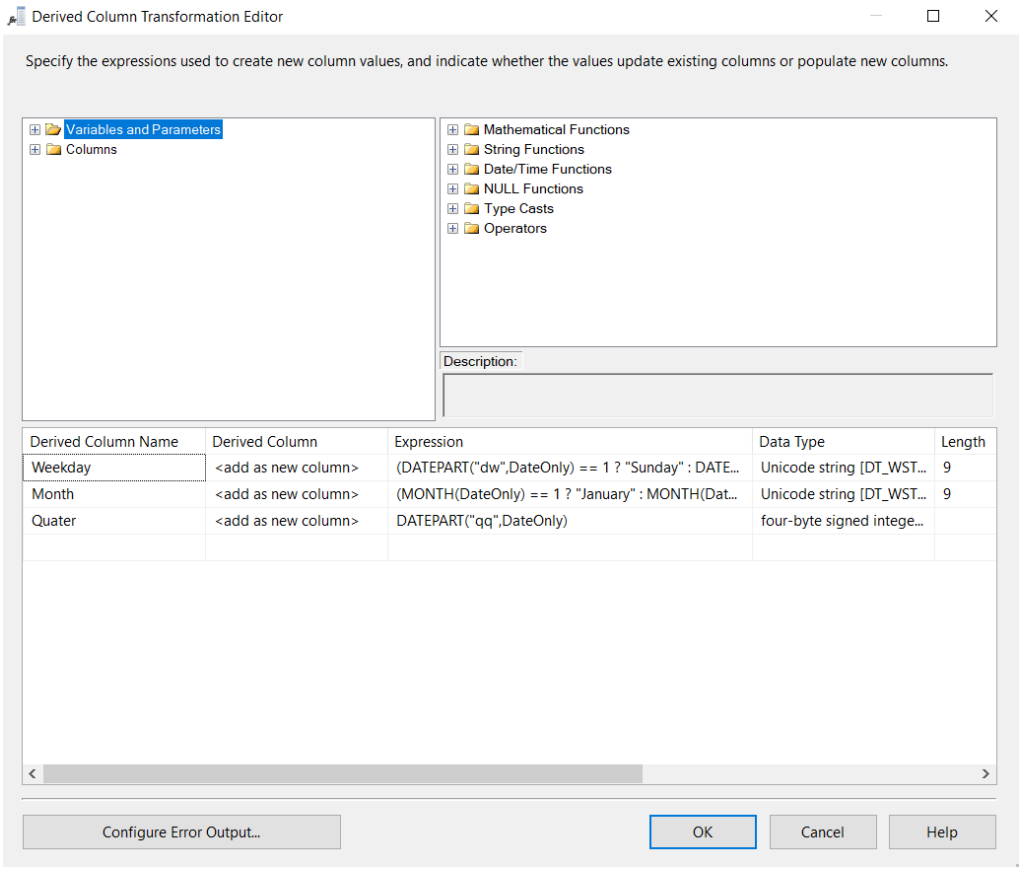
Tento editor je určen k psaní různých výrazů, do nichž je možné přetahovat existující sloupce, proměnné a parametry. Na pravé straně editoru je k dispozici seznam funkcí. Editor také nabízí uživatelům možnost, zda nahradit existující sloupec nebo přidat nový.

Sloupec **Date**, který obsahuje tabulka **DimDate** je datového typu *database timestamp*, což znamená, že sloupec obsahuje hodnoty data a času. Pro budoucí analýzu dat je možné přidat nový sloupec **DateOnly**, který bude obsahovat pouze datum. Použitý výraz pro tento sloupec je vidět na Obr. 33.



Obr. 33 Odvození data pomocí transformace Derived Column (vlastní zpracování)

Jak je patrné z datového skladu, který byl vytvořen v kapitole 6, tabulka **DimDate** by měla obsahovat více sloupců, proto za je nutné přidat ještě jednu komponentu do datového toku pro vytvoření dalších sloupců, které budou odvozeny již z nového sloupce **DateOnly** (viz Obr. 34).

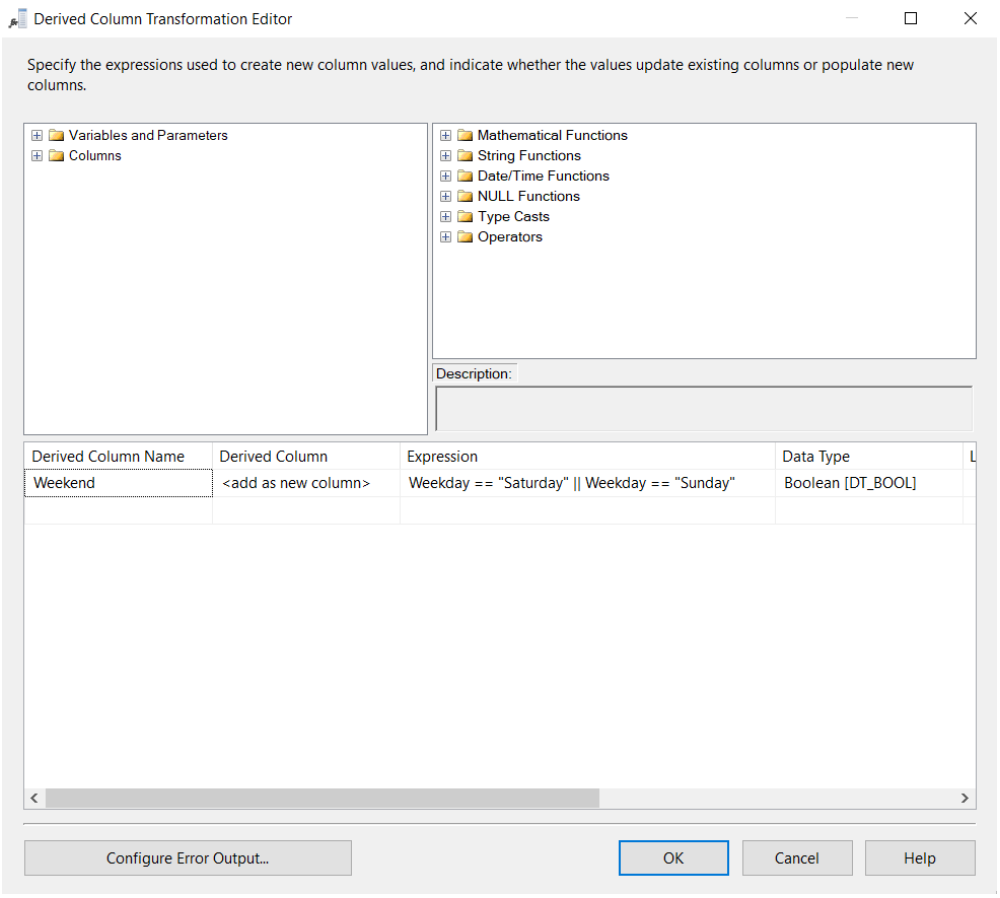


Obr. 34 Přidání dalších sloupců do tabulky DimDate (vlastní zpracování)

Uvedené výrazy odvozují z data následující hodnoty:

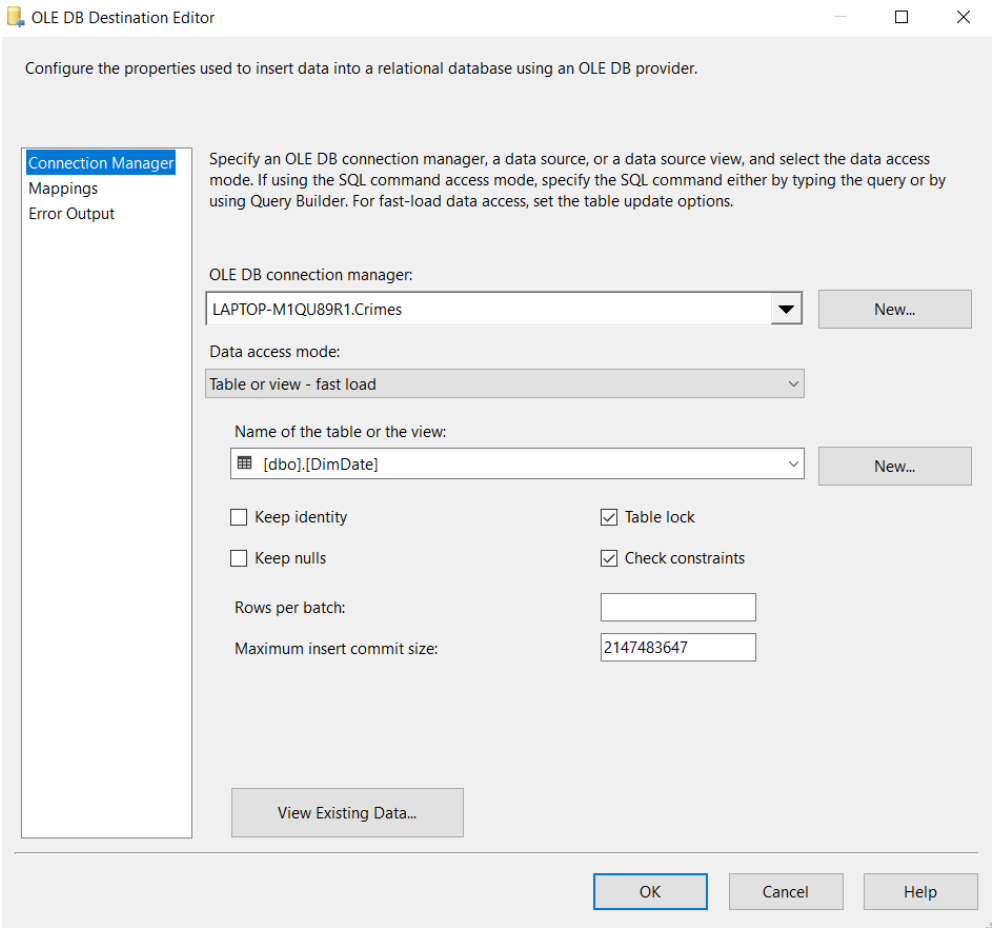
- Den týdne
  `(DATEPART("dw",DateOnly) == 1 ? "Sunday" : DATEPART("dw",DateOnly) == 2 ? "Monday" : DATEPART("dw",DateOnly) == 3 ? "Tuesday" : DATEPART("dw",DateOnly) == 4 ? "Wednesday" : DATEPART("dw",DateOnly) == 5 ? "Thursday" : DATEPART("dw",DateOnly) == 6 ? "Friday" : DATEPART("dw",DateOnly) == 7 ? "Saturday" : "")`
- Název měsíce
  `(MONTH(DateOnly) == 1 ? "January" : MONTH(DateOnly) == 2 ? "February" : MONTH(DateOnly) == 3 ? "March" : MONTH(DateOnly) == 4 ? "April" : MONTH(DateOnly) == 5 ? "May" : MONTH(DateOnly) == 6 ? "June" : MONTH(DateOnly) == 7 ? "July" : MONTH(DateOnly) == 8 ? "August" : MONTH(DateOnly) == 9 ? "September" : MONTH(DateOnly) == 10 ? "October" : MONTH(DateOnly) == 11 ? "November" : MONTH(DateOnly) == 12 ? "December" : "")`
- Číslo kvartálu.

Nakonec poslední sloupec, který bude přidán do tabulky DimDate je sloupec **Weekend**, který na základě sloupce **Weekday** uvažuje, zda se jedná o víkend nebo ne (viz Obr. 35).



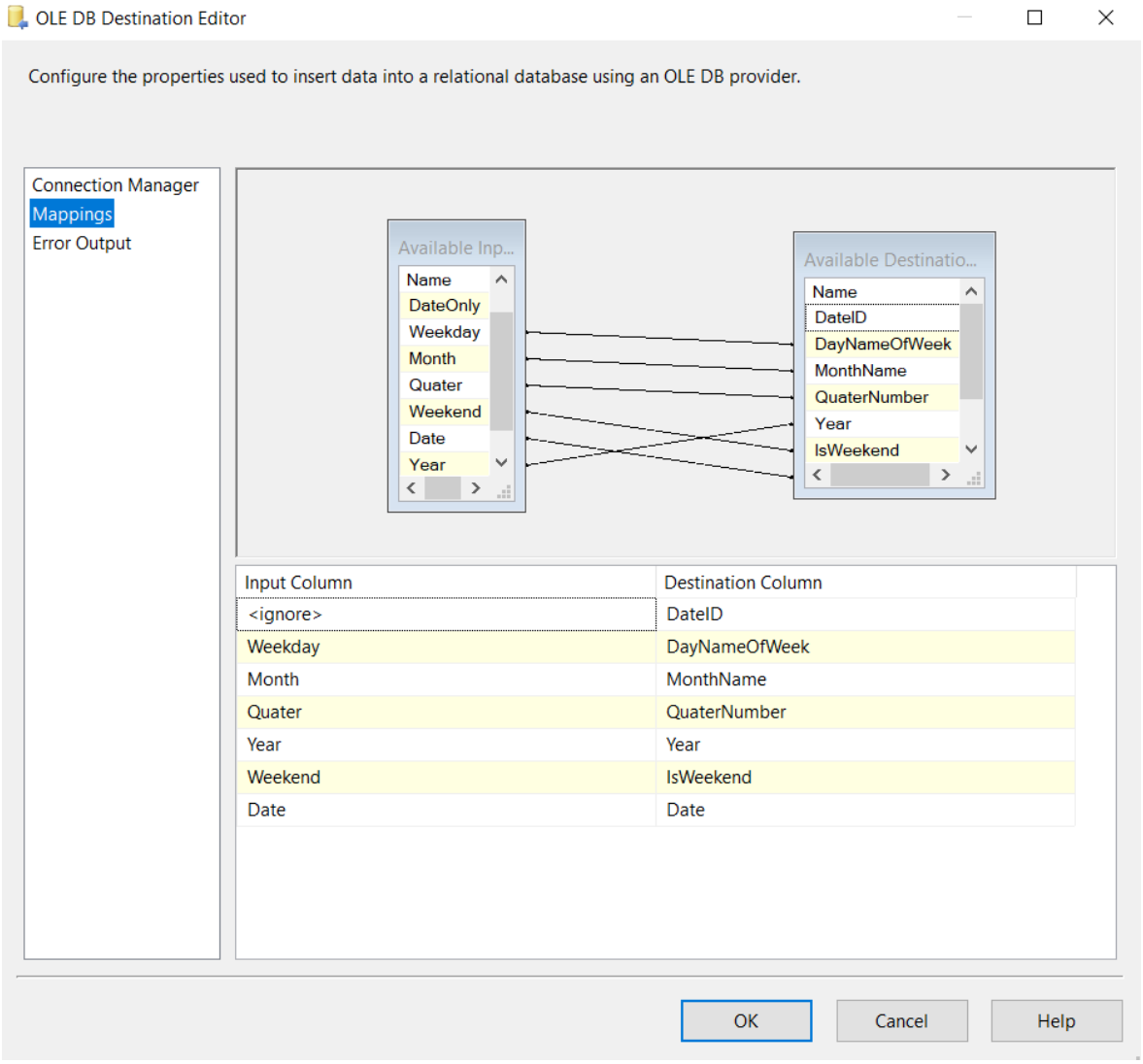
Obr. 35 Přidání sloupce Weekday (vlastní zpracování)

Všechny tyto transformace budou stačit, a proto lze přejít do finální fáze, jelikož daný balíček nyní může extrahovat data z prostého souboru a transformovat tato data do formátu kompatibilního s cílem. Proto dalším krokem je načíst transformovaná data do cílové tabulky **DimDate**, a to přidáním do datového toku komponenty **OLE DB Destination** (viz Obr. 36).



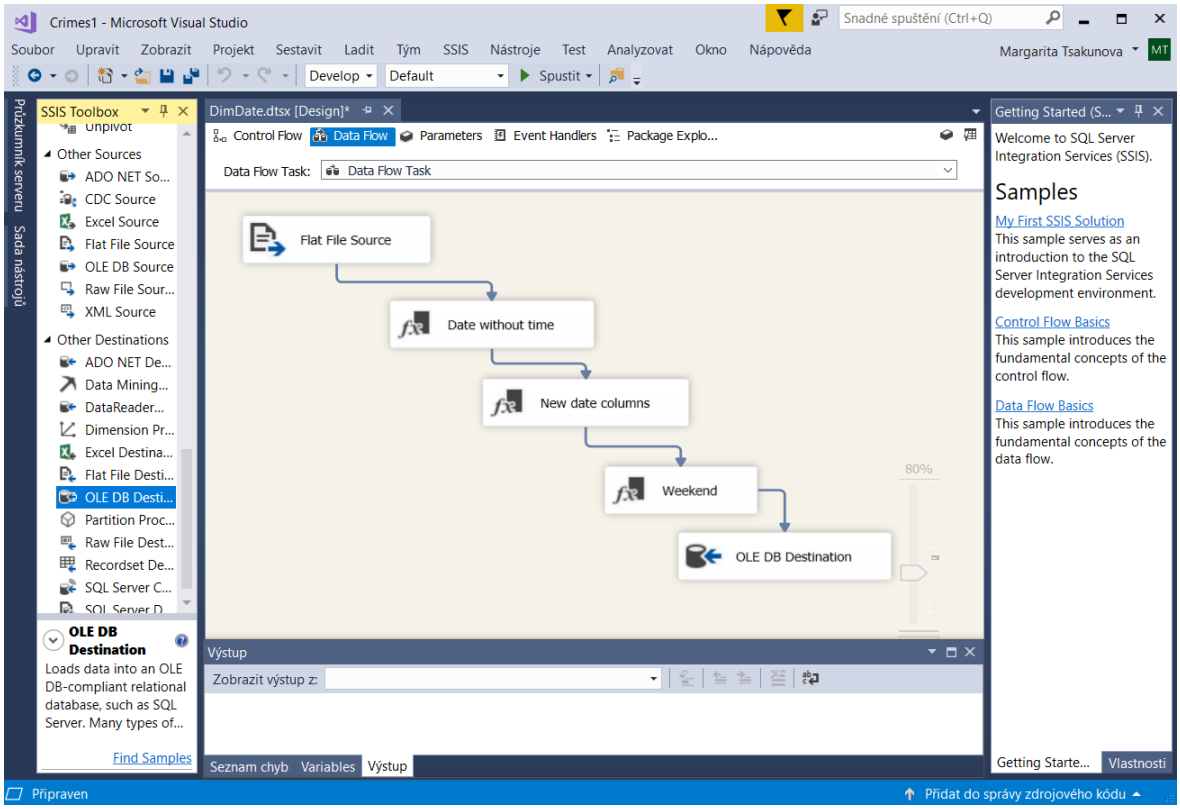
Obr. 36 Přidání cíle OLE DB Destination (vlastní zpracování)

Na záložce **Mappings** lze mapovat sloupce ze vstupu do sloupců cílové tabulky (viz Obr. 37). Jelikož vstupní data neobsahují žádné údaje pro sloupec **DateID**, tak není zmapován a je nastaven na hodnotu *ignore*, a tím pádem budou mu automaticky vygenerována identifikační čísla počínaje 1.



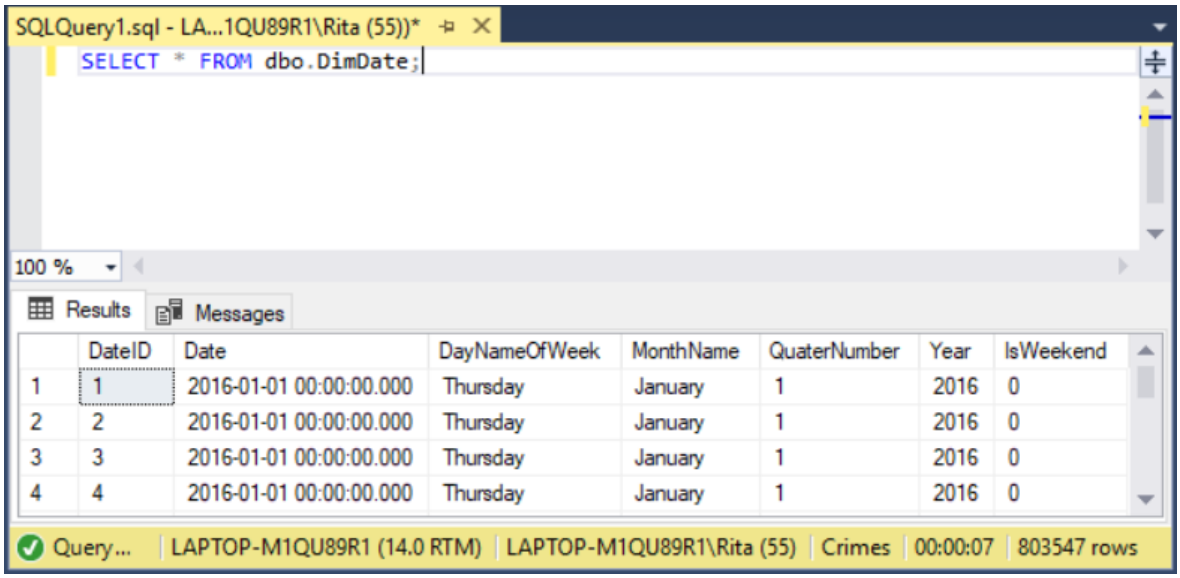
Obr. 37 Mapování sloupců (vlastní zpracování)

Celý **Data Flow** pro balíček tabulky **DimDate** pak vypadá jako na Obr. 38.



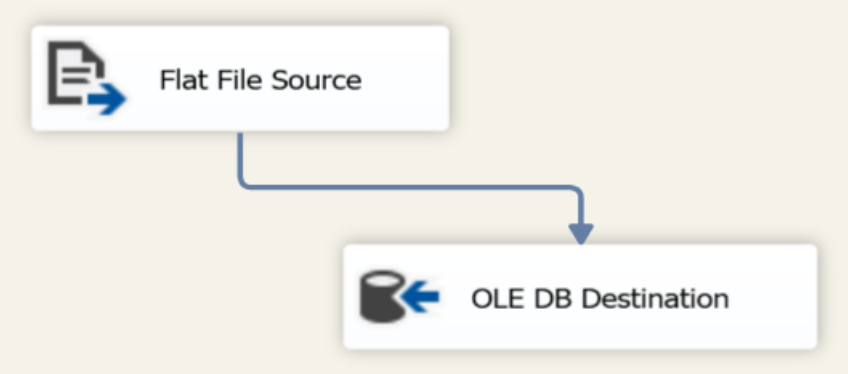
Obr. 38 Datový tok balíčku pro tabulku DimDate (vlastní zpracování)

Po spuštění celého balíčku se všechny záznamy nahraji do cílové tabulky **DimDate**. Celkový počet záznamů je 803 547, a to lze ověřit příkazem v SQL Server Management Studiu (viz Obr. 39).



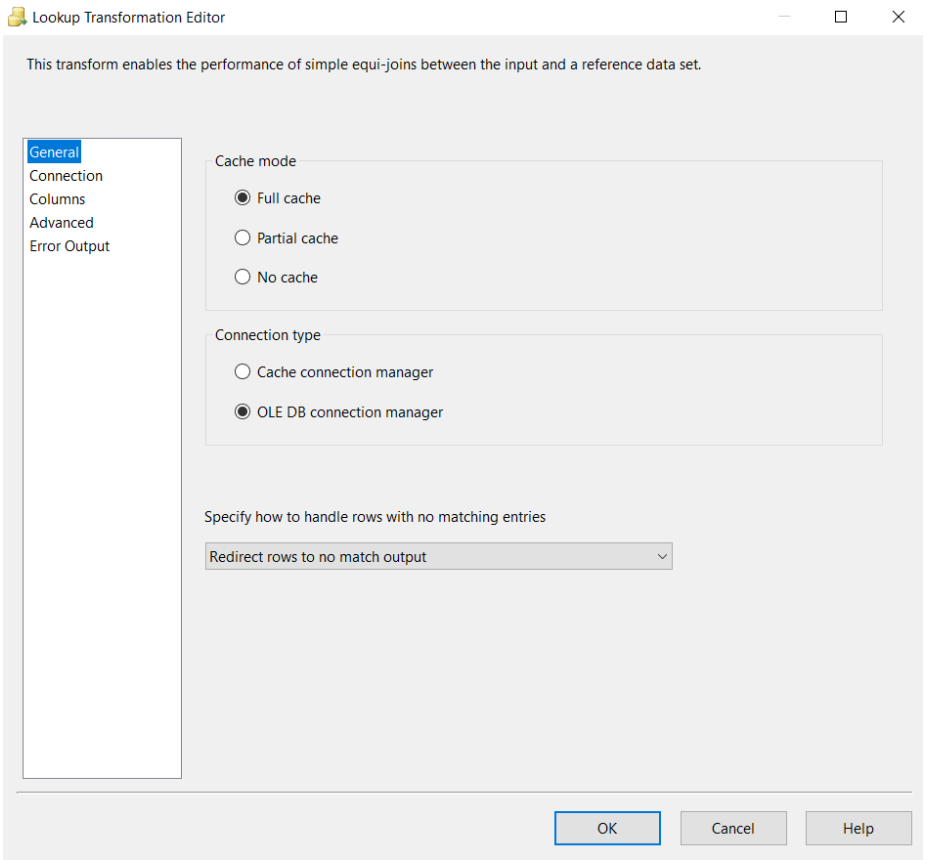
Obr. 39 Příkaz ověřující správné nahrání dat do tabulky DimDate (vlastní zpracování)

Ostatní dimenzionální tabulky **DimCrimeCategory**, **DimCrimeDescription**, **DimLocation** a **DimPoliceDepartment** nepotřebují žádné transformace, proto jejich datové toky se skládají pouze z komponent **Flat File Source** a **OLE DB Destination** se stejnými nastaveními jako u tabulky **DimDate** (viz Obr. 40).



Obr. 40 Příklad datového toku pro zbývající dimenzionální tabulky (vlastní zpracování)

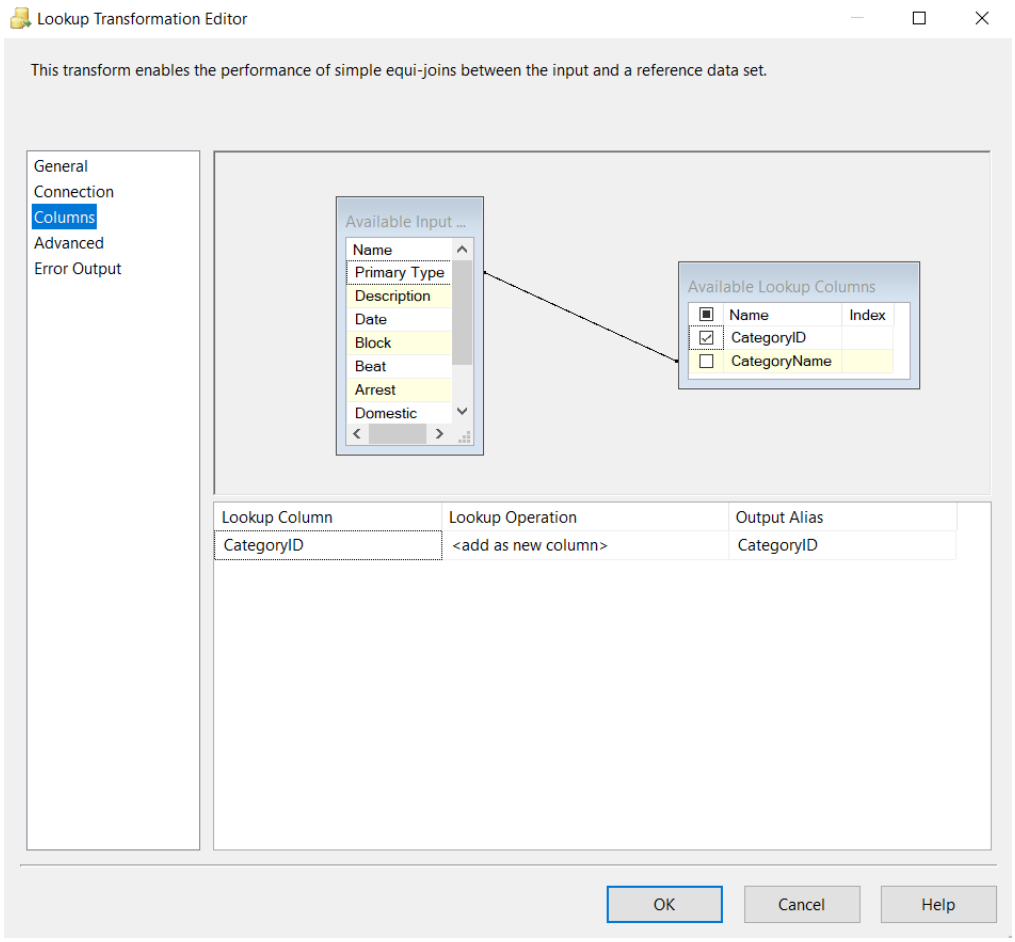
Teď jen zbývá vytvořit poslední balíček pro faktovou tabulku a načíst do ní data, ale obsah datového toku se bude trochu lišit od datových toků dimenzionálních tabulek. Obecně platí, že u faktové tabulky proces zahrnuje tři běžné fáze: přidání zdroje dat, získání primárních klíčů dimenzionálních tabulek a přidání destinace. Za účelem získání primárních klíčů z tabulek dimenzí se používají transformace nazývané **Lookup**, které umožňují spojení dat ve sloupcích vstupní tabulky se sloupci tabulky jiné. Každá taková komponenta se vytváří pro jednotlivé dimenzionální tabulky. Po přidání **Lookupu** do **DataFlow** je třeba nastavit obecné parametry následujícím způsobem (viz Obr. 41).



Obr. 41 Obecná nastavení pro transformaci Lookup (vlastní zpracování)

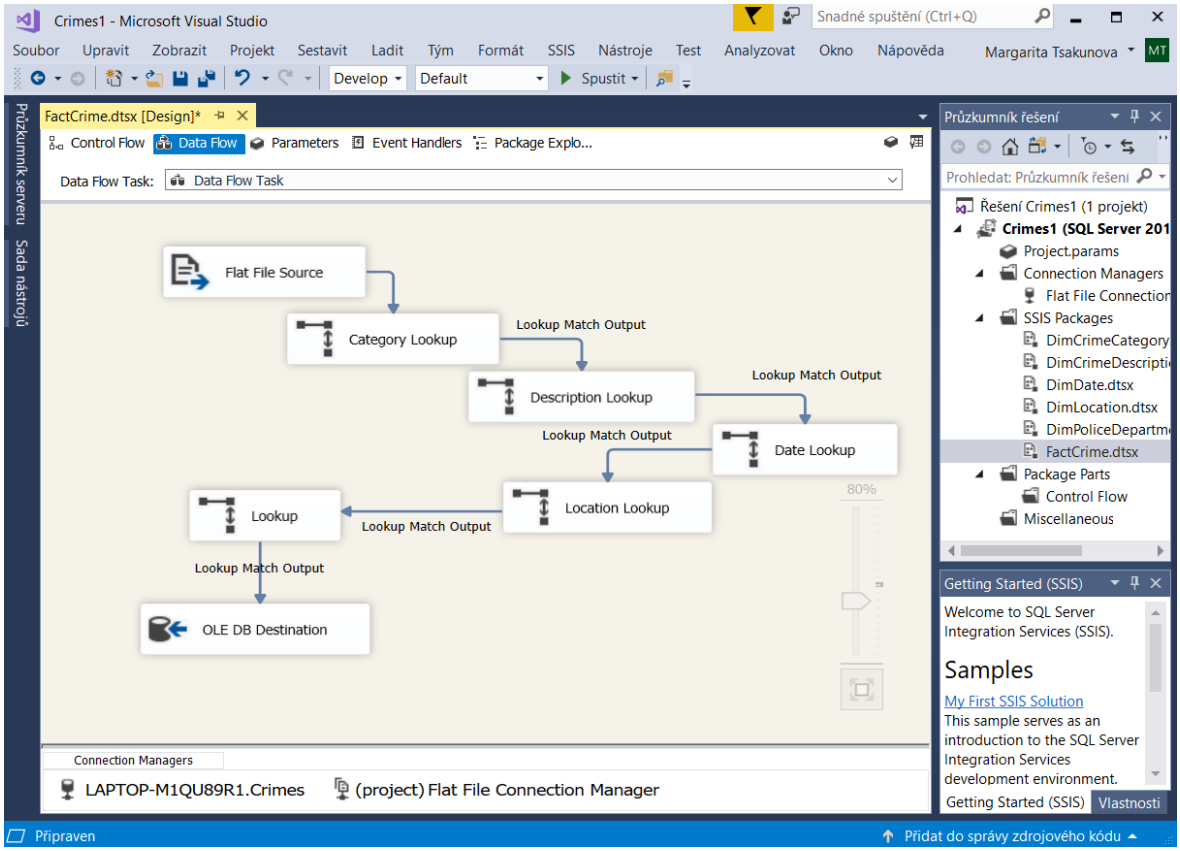
Na záložce **Connection** by měla mít vybrána tabulka, jejíž klíč je potřeba získat, např. tabulka **DimCrimeCategory**. Na záložce Columns pak stačí propojit **Primary Type** ze

vstupní tabulky a **CategoryName** z tabulky **DimCrimeCategory** a uvést **CategoryID** jako výstup (viz Obr. 42).



Obr. 42 Získání primárního klíče tabulky DimCrimeCategory (vlastní zpracování)

Úplně stejné kroky lze provést pro získání ostatních primárních klíčů dimenzionálních tabulek. Až budou všechny **Lookup** hotové, datový tok pro faktovou tabulku **FactCrime** pak vypadá jako na Obr. 43.



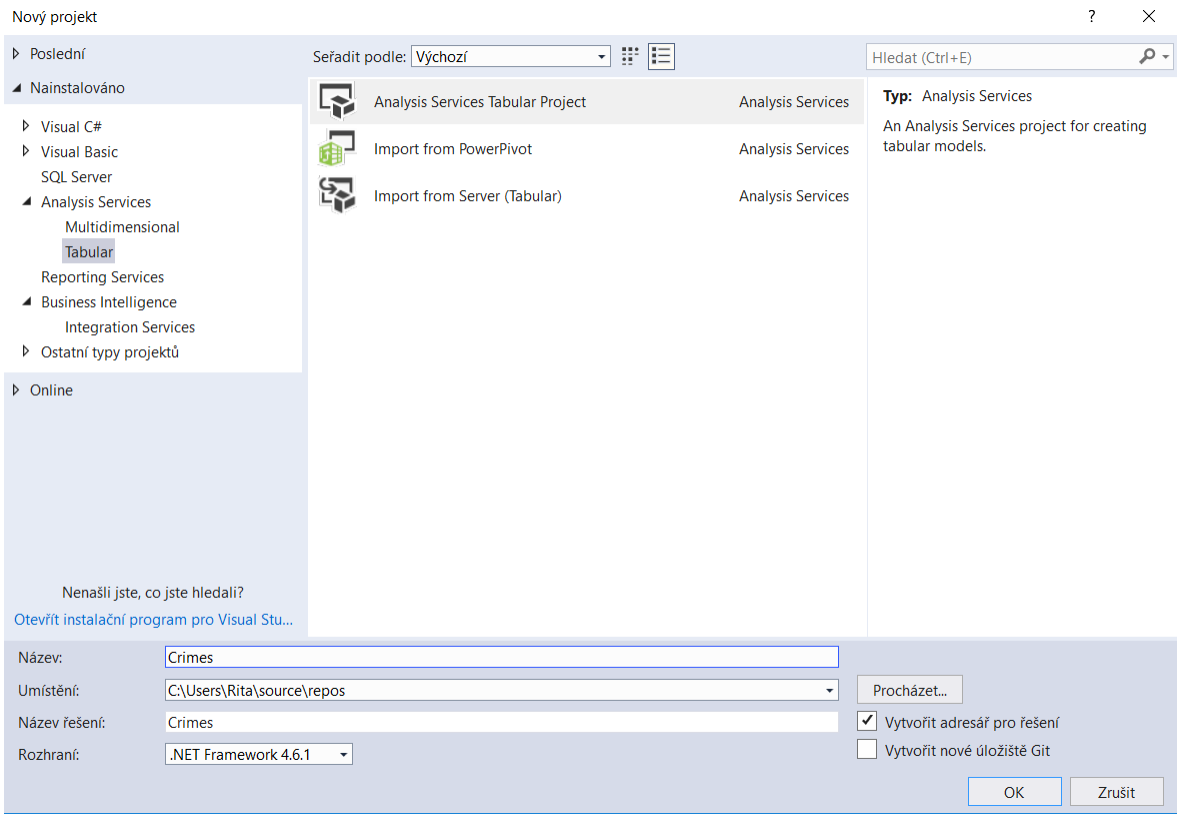
Obr. 43 Výsledný datový tok balíčku pro tabulku FactCrime (vlastní zpracování)

Po spuštění tohoto balíčku se všechna potřebná data, včetně cizích klíčů faktové tabulky, nahrají do cílového datového skladu. Tím pádem proces nahrávání všech potřebných dat do datového skladu končí.

8 Tvorba tabulárního modelu

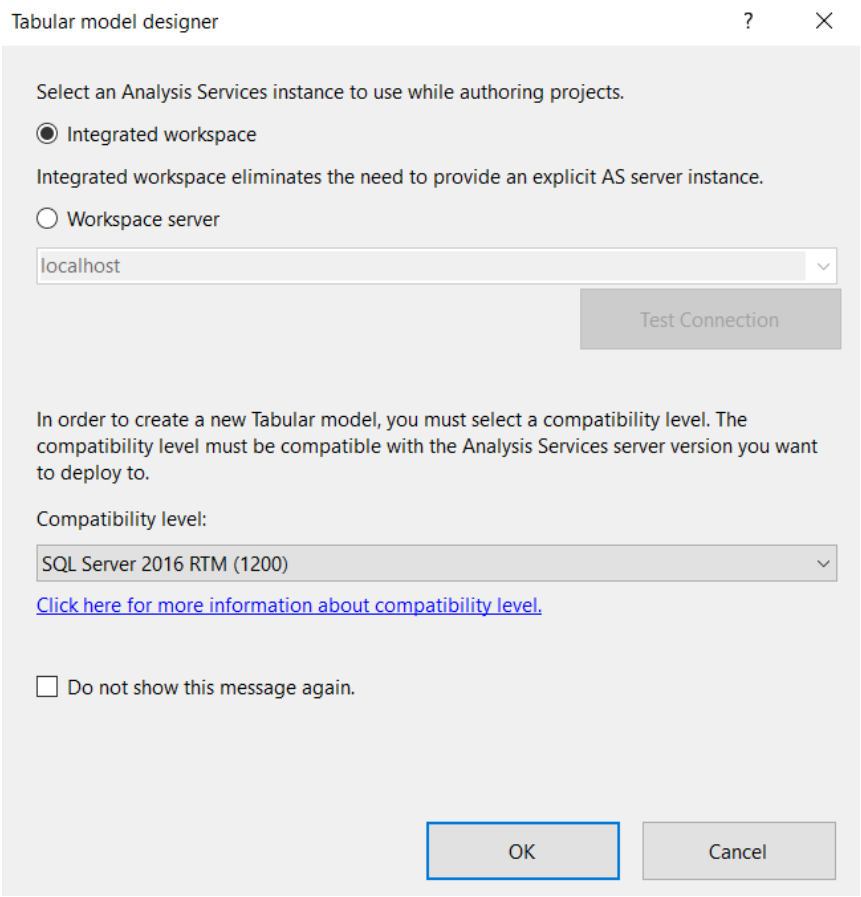
Předtím, než se pustí do samotné tvorby reportů, navrhuji vytvořit tabulární model prostřednictvím využití analytických služeb od Microsoft, protože je optimalizován na rychlé dotazování.

Proces tvorby tabulárního modelu začíná tak, že se nejdřív vytvoří nový projekt v SSDT **New -> Project**, kde je nutné specifikovat, že se jedná o **Analysis Services Tabular Project** (viz Obr. 44).



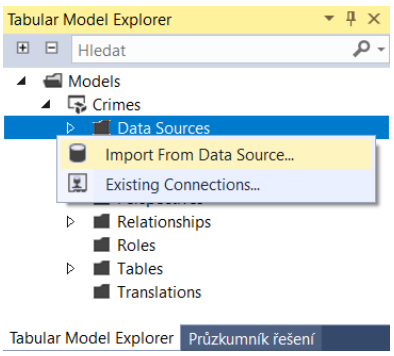
Obr. 44 Vytvoření SSAS Tabular Project (vlastní zpracování)

Objeví se vyskakovací okno, které umožňuje ověřit, zda je nastavena možnost **Integrated workplace** a zda je zvolena správná úroveň kompatibility (viz Obr. 45).



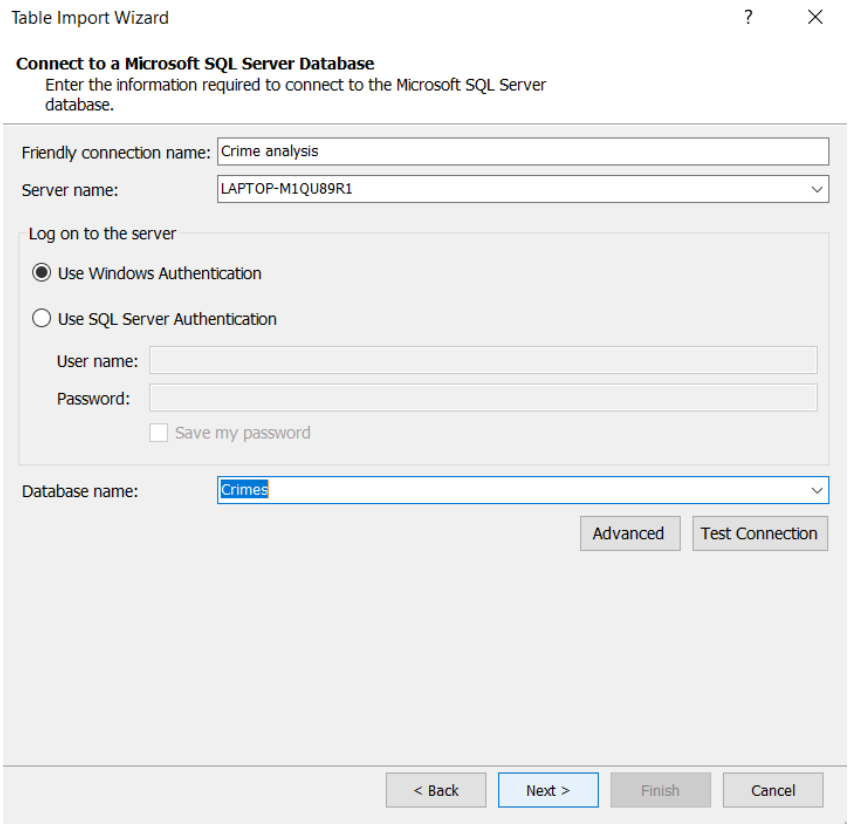
Obr. 45 Nastavení Tabular model designeru (vlastní zpracování)

Nyní lze importovat data do projektu kliknutím na **Data Sources -> Import From Data Source** v okně **Tabular Model Explorer**, které se nachází v pravé části panelu (viz Obr. 46).



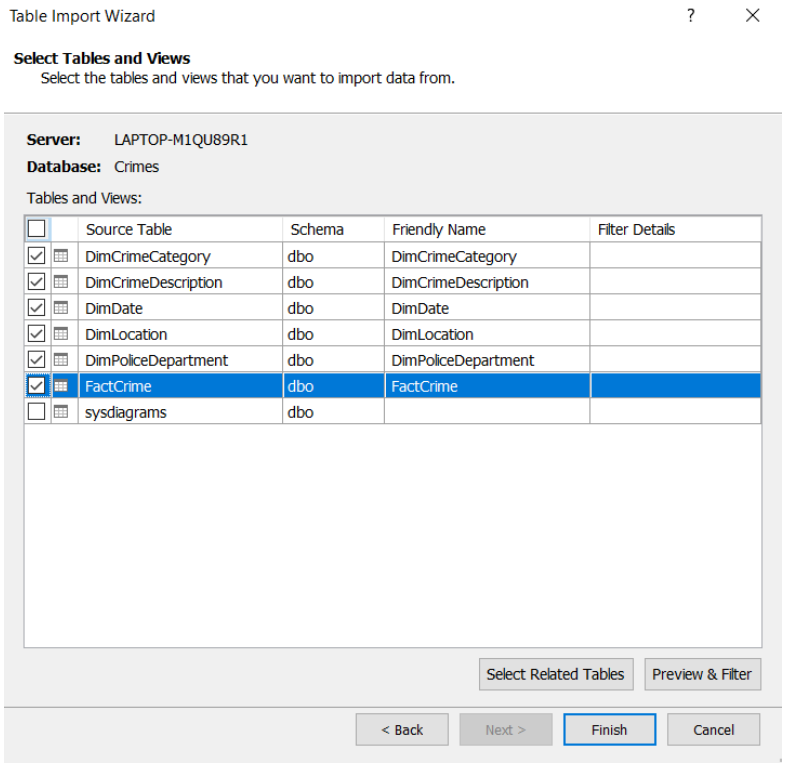
Obr. 46 Import dat do SSAS projektu (vlastní zpracování)

Zobrazí se **Table Import Wizard**, kde je nutné zvolit **Microsoft SQL Server -> Next**. V dalším okně musí být nastaveny parametry zobrazené na Obr. 47.

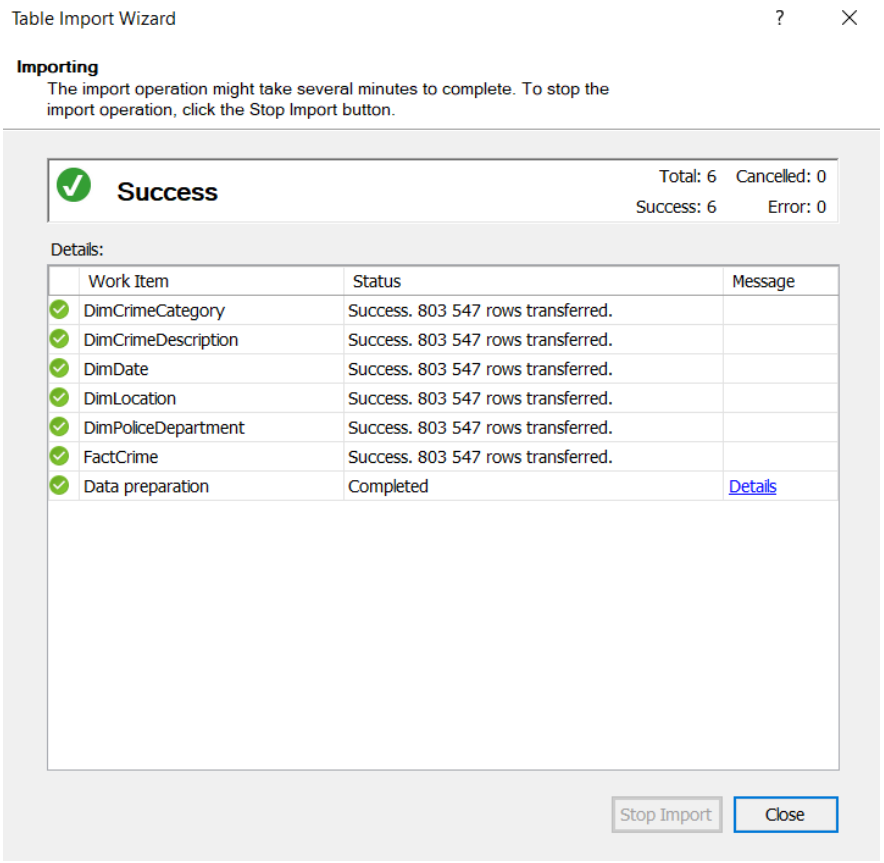


Obr. 47 Nastavení připojení k MS SQL Server Database (vlastní zpracování)

Dalším krokem je filtrování a přejmenování potřebných tabulek, pocházejících ze zdroje dat (viz Obr. 48) a dokončení celého procesu importu dat (viz Obr. 49).



Obr. 48 Úprava tabulek určených pro import (vlastní zpracování)

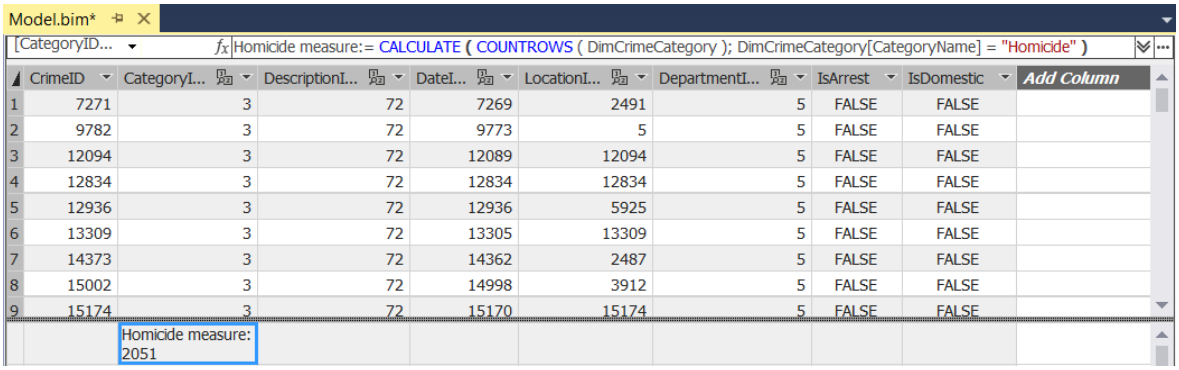


Obr. 49 Úspěšné dokončení procesu importu dat (vlastní zpracování)

Pro ulehčení práce při tvorbě reportů v Power BI je nutné do vytvořeného tabulárního modelu přidat nové metriky, a to pomocí jazyka DAX (*Data Analysis Expressions*). Metriky jsou číselné hodnoty, které se počítají za účelem poskytnutí významu datům, se kterými se pracuje (Microsoft, 2018). Metriky lze vytvářet i v Power BI, ale pro naplnění účelu bakalářské práce bude využít SSAS tabulární model. Stačí v **Measure Grid** faktové tabulky zadat příslušný vzorec a nechat ho vyhodnotit. Například je vhodné vytvořit metriku **Homicide** s následujícím vzorcem (viz Obr. 50):

```
Homicide measure := CALCULATE ( COUNTROWS ( DimCrimeCategory );
DimCrimeCategory[CategoryName] = "Homicide" )
```

Tento vzorec slouží pro určení počtu řáků, jejichž hodnoty odpovídají *"Homicide"*, a je počítán ze sloupce **CategoryName** tabulky **DimCrimeCategory**. Na základě této metriky se dá posoudit, kolik vražd bylo spácháno v Chicagu za tři roky. Výsledkem funkce jsou 2 051 záznamů.

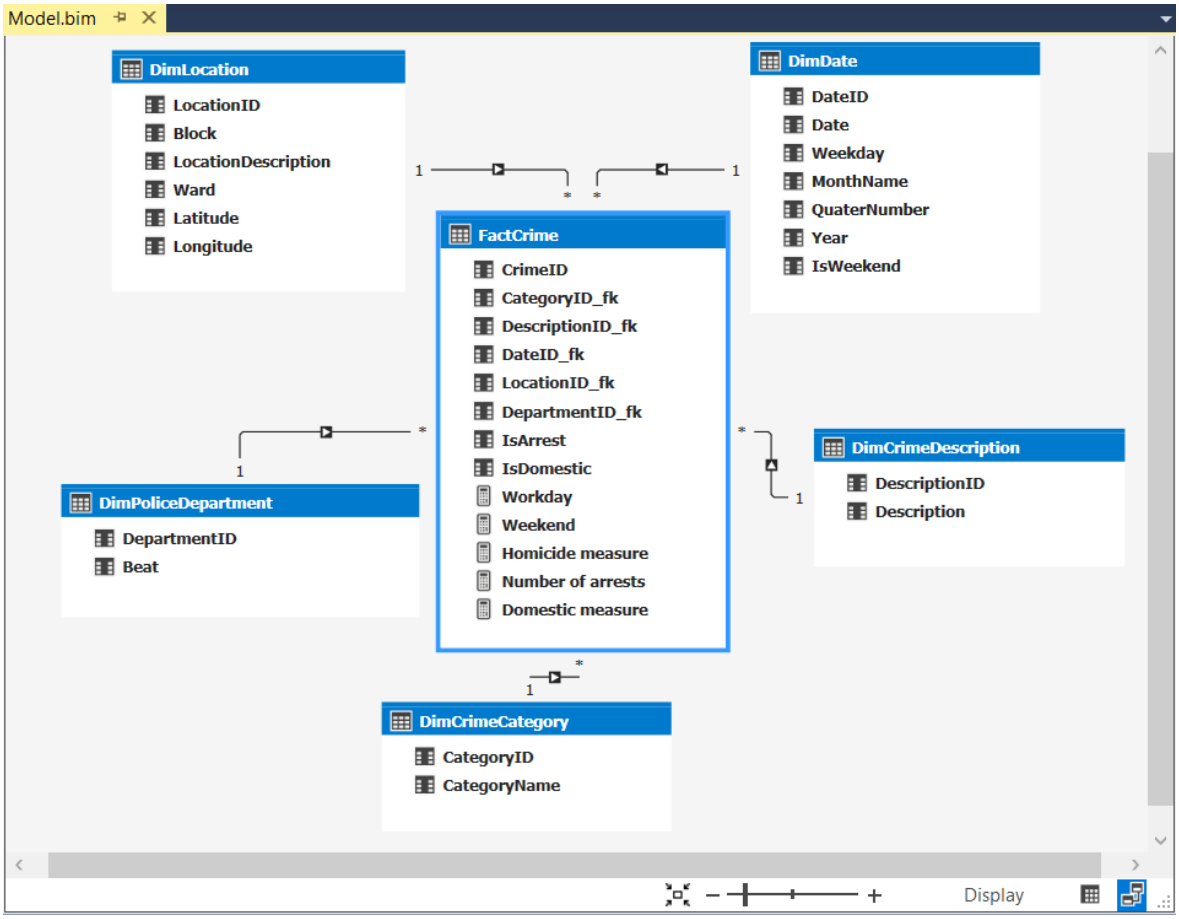


Obr. 50 DAX funkce pro Homicide metriku (vlastní zpracování)

Dále jsem vytvořila ještě čtyři různé metriky, které se taky ukládají do tabulky faktů:

- Domestic measure
  `Domestic measure := COUNTAX(FILTER('FactCrime';[IsDomestic]=TRUE());TRUE())`
  Tato funkce se vztahuje ke sloupci **IsDomestic** z tabulky FactCrime a vypisuje počet hodnot *TRUE*, což v daném případě znamená, u kolika zločinů se jedná o domácí násilí. Tento počet se určuje porovnáváním hodnot sloupce s požadovanou hodnotou *TRUE*.
- Number of arrests
  `Number of arrests := COUNTAX(FILTER('FactCrime';[IsArrest]=TRUE());TRUE())`
  Funkce je založena na stejnem principu jako ta předchozí a vyhodnocuje celkový počet zločinu, u nichž bylo provedeno zatčení pachatele. Metrika je odvozena ze sloupce **IsArrest** faktové tabulky.
- Workday
  `Workday := COUNTAX(FILTER('DimDate';[IsWeekend]=FALSE());FALSE())`
  Není nutné tuto metriku moc rozepisovat, jelikož její funkce je stejná jako u dvou předchozích metrik. Metrika počítá, kolik zločinů vychází na pracovní den.
- Weekend
  `Weekend := COUNTAX(FILTER('DimDate';[IsWeekend]=TRUE());TRUE())`
  Na rozdíl od předchozí metriky určuje, kolik zločinu bylo spácháno o víkendu.

Všechny požadované metriky jsou vytvořeny, proto je nezbytné ukázat nový datový model s metrikami. Tabulární model vypadá jako na Obr. 51 a jak je patrné z tohoto obrázku, obsahuje již nově vytvořené metriky, které jsou uloženy do faktové tabulky.



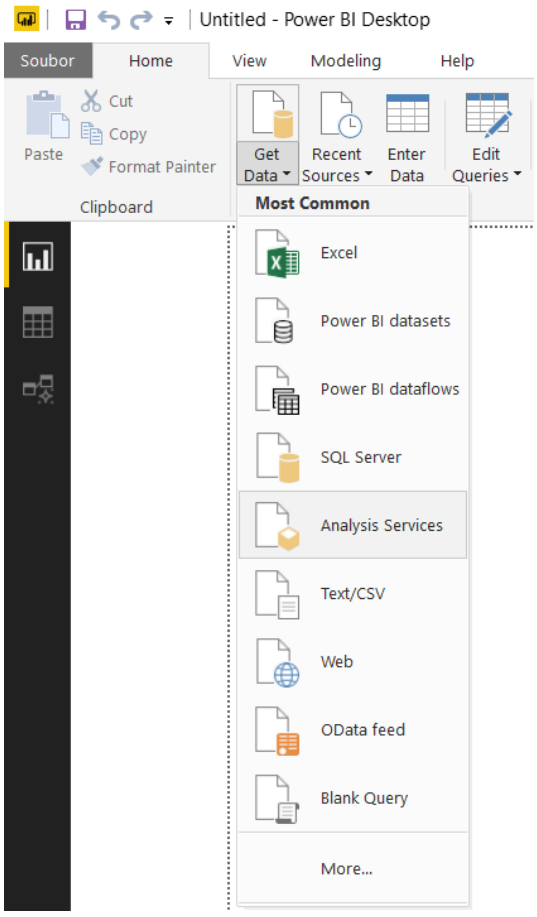
Obr. 51 Tabulární model (vlastní zpracování)

Nyní je již možné nasadit tento model na server služby Analysis Services, a tímto krokem končí proces tvorby modelu.

9 Tvorba reportu v nástroji Power BI

Poslední etapa implementace BI řešení spočívá ve vytváření reportů v Power BI na základě dat, která jsou uložena v tabulárním modelu z předchozí kapitoly 8.

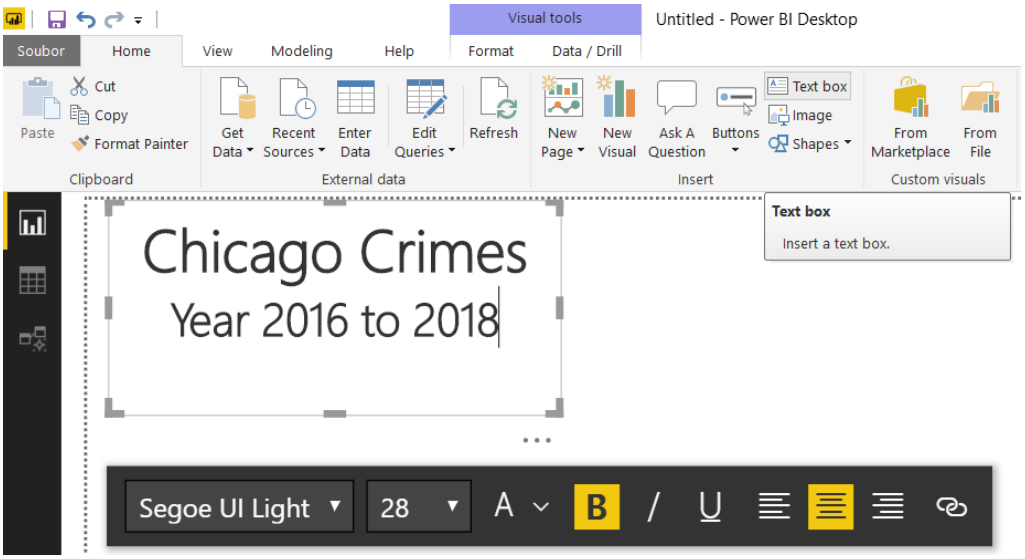
Pro tvorbu reportu je používán Power BI Desktop s jehož pomocí lze reporty upravovat a následně publikovat. Při importu dat do Power BI lze uvažovat o různých datových zdrojích. Jelikož v předchozí kapitole byl vytvořen tabulární model, bude tady zdrojem dat služba Analysis Services. Pro import dat musí uživatel najít na úvodní obrazovce menu **Home** a zvolit **Get Data -> Analysis Services** (viz Obr. 52).



Obr. 52 Import dat do Power BI (vlastní zpracování)

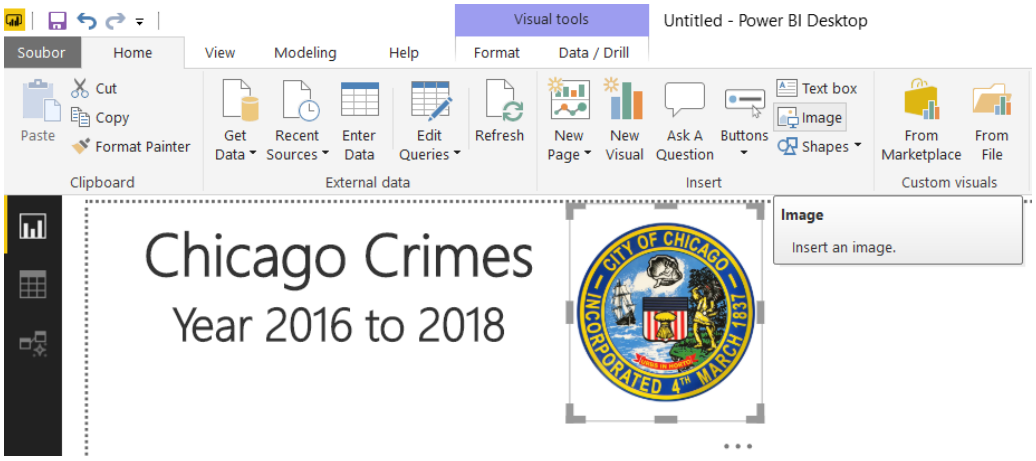
Objeví se okno, ve kterém může uživatel vyplnit pole s názvem serveru a nepovinné pole s názvem databáze. Po připojení k serveru, zobrazí se nové okno, kde uživatel zvolí tabulky, s nimiž se chystá pracovat. Já jsem naimportovala všechny své tabulky.

Nyní lze začít vytvářet samotný report. První stránka reportu bude nabízet celkový pohled na zločiny ve městě Chicago. Je dobré vložit nadpis reportu ještě předtím, než se budou do reportu přidávat vizualizace a filtry. Možnost vložení textového pole se nachází v záložce **Home -> Text Box** (viz Obr. 53**Error! Reference source not found.**).



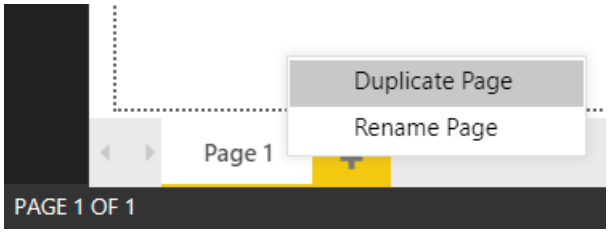
Obr. 53 Přidání nadpisu (vlastní zpracování)

Vedle nadpisu lze vložit znak Chicaga **Home -> Image** (viz Obr. 54**Error! Reference source not found.**).



Obr. 54 Vložení obrázku (vlastní zpracování)

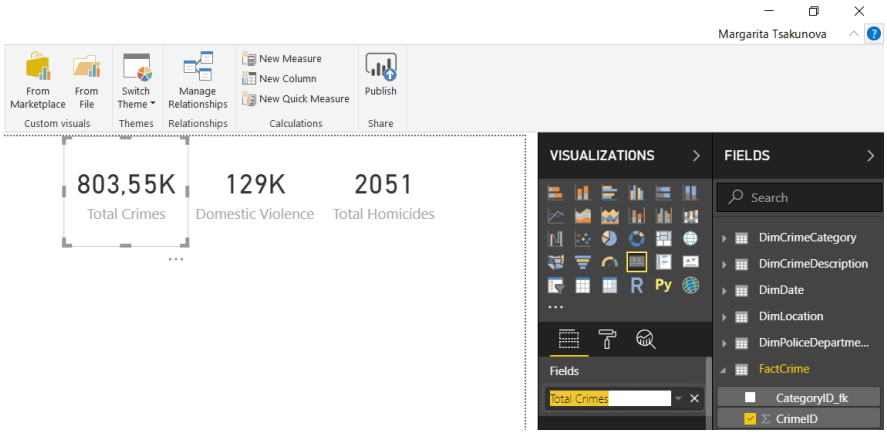
Teď je potřeba tuto stránku duplikovat, jelikož celý report se bude skládat celkem ze dvou stránek a obě stránky budou obsahovat stejný nadpis s obrázkem (viz Obr. 55).



Obr. 55 Duplikování stránky reportu (vlastní zpracování)

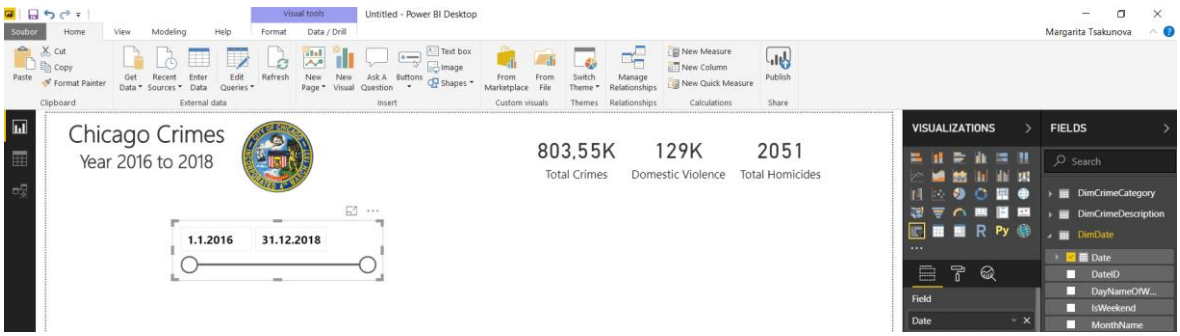
Druhá stránka byla přidána a lze pokračovat s úpravy na první stránce. Úplně doprava nahoru může uživatel přidat různé součty pro přehlednost určité statistiky zločinů. Toho lze docílit pouhým přetažením hodnot ze seznamu **Fields** do zobrazení **Visualizations**,

přičemž je nutné zvolit **Card** vizualizaci. Co se týče mého příkladu, tak jsem přetáhla součet všech ID zločinů, **Domestic measure** a **Homicide Measure** z tabulky **FactCrime** (viz Obr. 56**Error! Reference source not found.**) a změnila jsem názvy jednotlivých součtů pro lepší pochopitelnost.



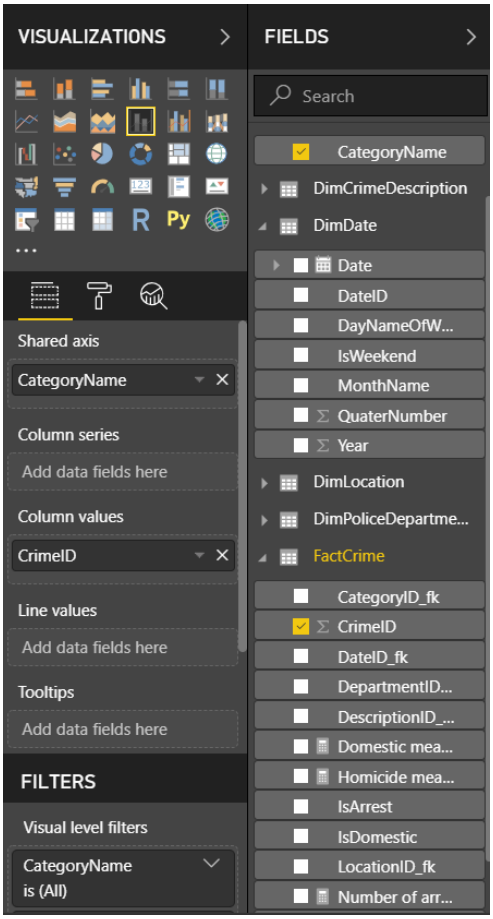
Obr. 56 Přidání součtů (vlastní zpracování)

Dalším krokem je vložení filtru celé stránky, který umožní uživateli pracovat pouze s těmi daty, na nichž se bude chtít zaměřit v rámci jedné stránky. Je nutné zvolit **Slicer** vizualizaci v podokně **Visualizations** a přetáhnout hodnotu **Date** do pole **Fields,** jelikož se budou data třídit podle datumu (viz Obr. 57).



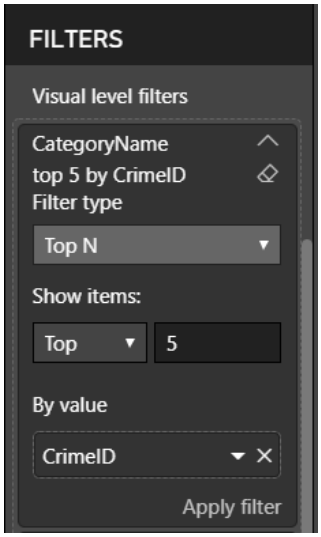
Obr. 57 Vložení filtru (vlastní zpracování)

Teď je na radě vložení sloupcového grafu, který bude odrážet 5 nejčastějších druhů incidentů. Je potřeba provést následující kroky: za prvé, zvolit vizualizaci **Line and stacked column chart** a pak přidat hodnotu **CategoryName** z tabulky **DimCrimeCategory** do pole **Shared axis** a hodnotu **CrimeID** do **Column values**, což určuje, že se v grafu zobrazují celkové počty jednotlivých druhů zločinů (viz Obr. 58).



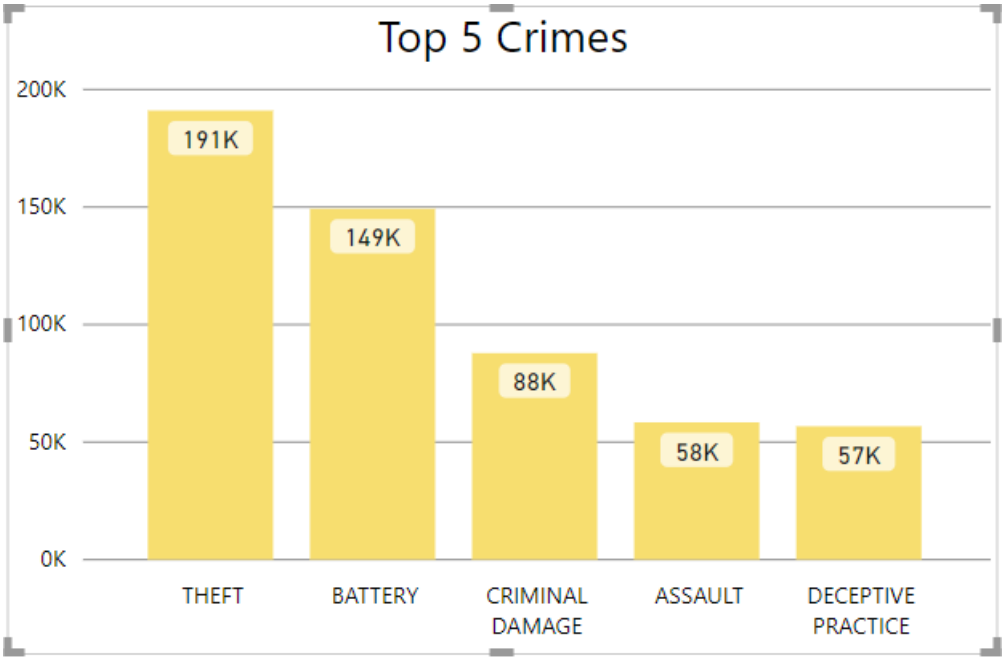
Obr. 58 Vložení sloupcového grafu (vlastní zpracování)

Za druhé, je nutné přidat filtr, který odebere z grafu všechny druhy zločinů kromě prvních pět, jejichž počty jsou největší. Lze to udělat v podokně Filters, které se vztahuje k danému grafu. Nastavení filtru jsou uvedena na Obr. 59.



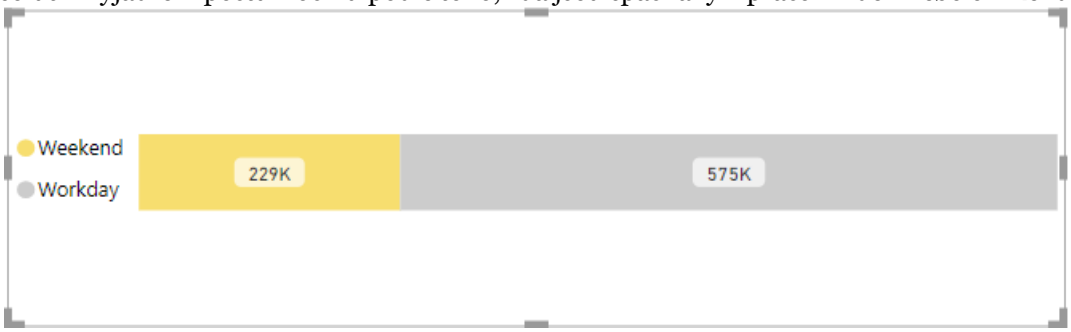
Obr. 59 Filtr Top 5 zločinů (vlastní zpracování)

Po všech úpravách vypadá sloupcový graf jako na Obr. 60. Podle tohoto grafu lze určit, čeho se musí člověk varovat, pokud se nachází nebo žije v Chicagu, a jaká bezpečnostní opatření je třeba přijmout.

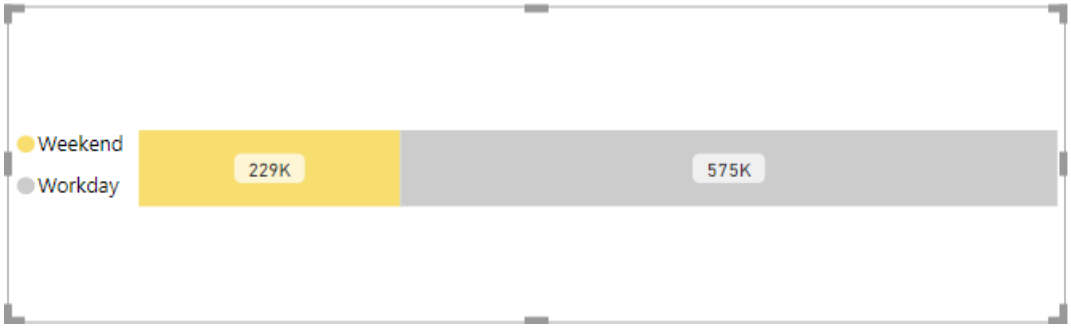


Obr. 60 Sloupcový graf Top 5 Crimes (vlastní zpracování)

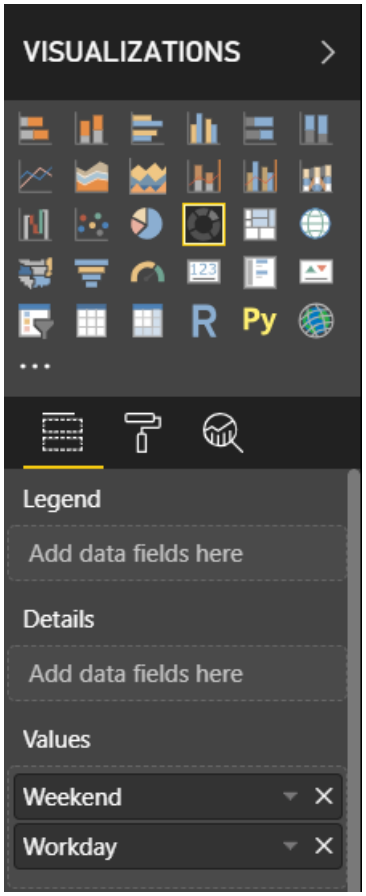
Následně lze přidat na list skládaný sloupcový graf (**Stacked bar chart**), který bude zobrazovat absolutní vyjádření počtu zločinů podle toho, zda jsou spáchány v pracovní den nebo o víkendu (



Obr. 61). Při tvorbě tohoto grafu se použijí metriky **Weekend** a **Workday** a žádná další nastavení nejsou (viz Obr. 62).

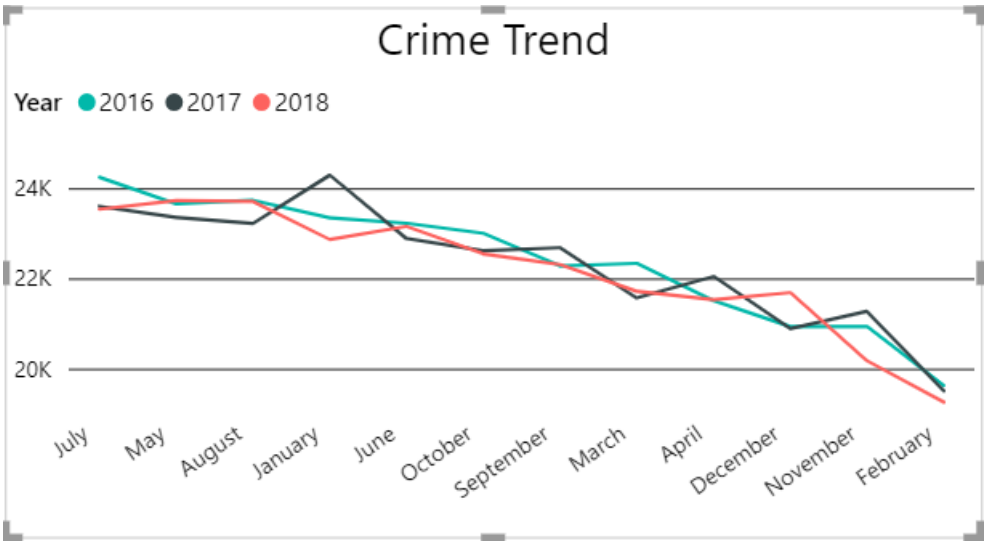


Obr. 61 Skládaný sloupcový graf (vlastní zpracování)



Obr. 62 Parametry nastavení prstencového grafu (vlastní zpracování)

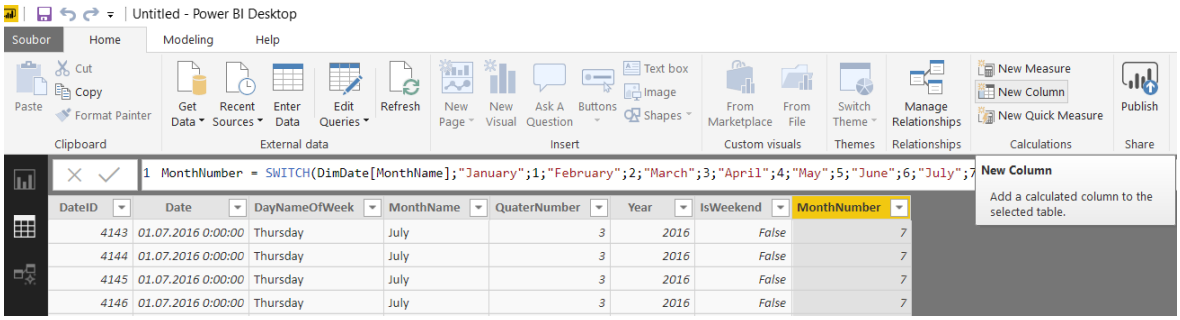
Poslední vizualizací, která se umístí na první stránku reportu, je čárový graf (**Line chart**). Účelem tohoto grafu je znázornění tří trendových čar (reprezentujících roky 2016, 2017 a 2018), které vyjadřují úroveň kriminality podle jednotlivých měsíců. Po nastavení os *x* a *y* a legendy grafu lze pozorovat, že měsíce jsou uspořádány chaoticky (viz Obr. 63), což je třeba napravit.



Obr. 63 Čárový graf podle neseřazených měsíců (vlastní zpracování)

Pro správné seřazení měsíců je potřeba mít sloupec, který obsahuje čísla měsíců. Žádná tabulka z demonstračního datového modelu nemá takový sloupec, ale lze ho snadno a rychle vytvořit, resp. odvodit ze sloupce s názvy měsíců.

Nejdřív je nutné přejít na záložku **Data**, která se nachází vlevo od listu a zvolit tabulku, která obsahuje sloupec **MonthName**. Poté vybrat v menu **Home** -> **New column** a zadat výraz, který z názvu odvozuje číslo měsíce (viz Obr. 64).

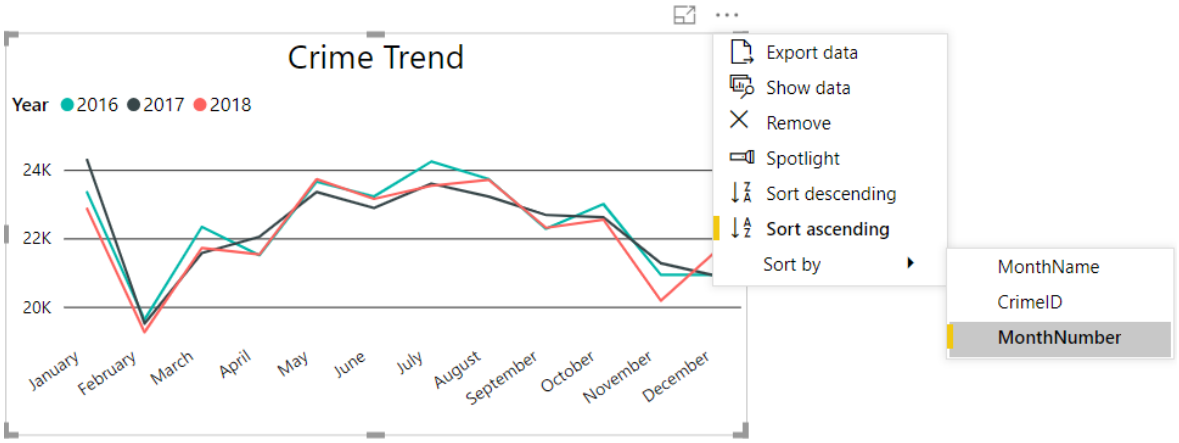


Obr. 64 Přidání nového sloupce (vlastní zpracování)

Pro odvození nového sloupce **MonthNumber** je použít následující výraz:

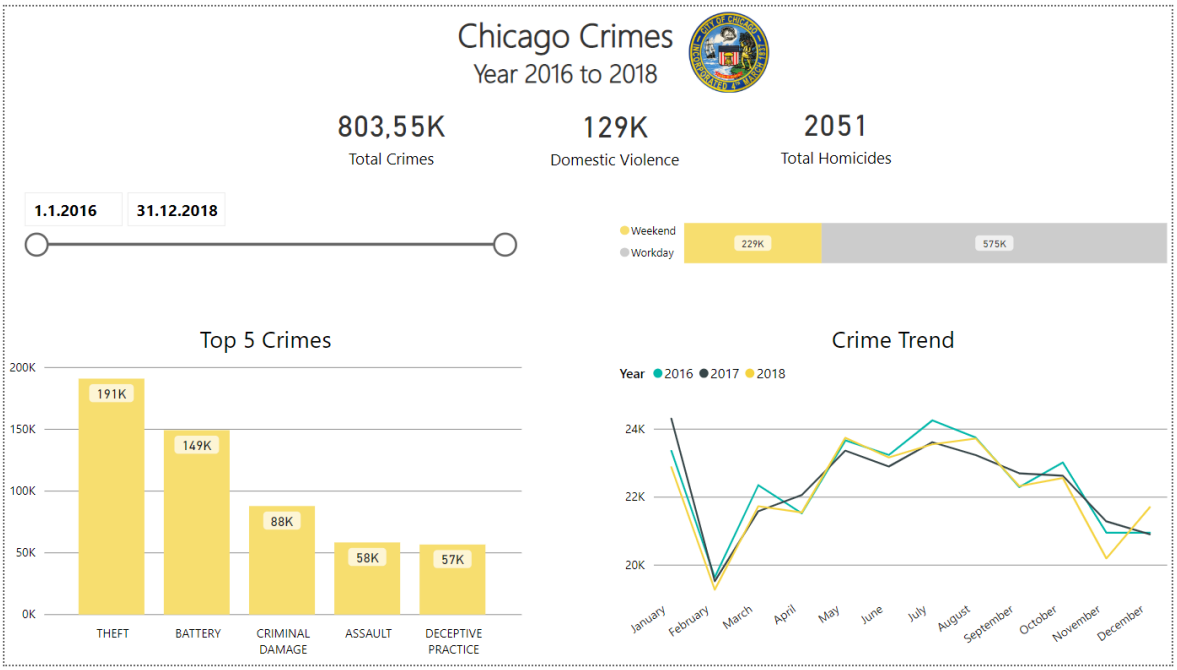
```
MonthNumber =
SWITCH(DimDate[MonthName];"January";1;"February";2;"March";3;"April";4;"May";5;"June";6;"July";7;"August";8;"September";9;"October";10;"November";11;"December";12)
```

Teď už jenom zbývá přejít zpátky na záložku **Report**, kliknout na čárový graf a přetáhnout do pole **Tooltips** v podokně **Visualizations** hodnotu **MonthNumber**. Pak na samotném grafu kliknout na tři tečky a zvolit **Sort by -> MonthNumber**, což vede k tomu, že hodnoty ose *x* začínají od prvního kalendářního měsíce (viz Obr. 65).



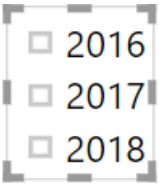
Obr. 65 Čárový graf podle seřazených měsíců (vlastní zpracování)

Práce s první stránkou reportu je dokončena a výsledný list vypadá jako na Obr. 66. Umožňuje získat okamžitý přehled o kriminalitě ve městě Chicago za období 2016 – 2018 a jejím vývoji.



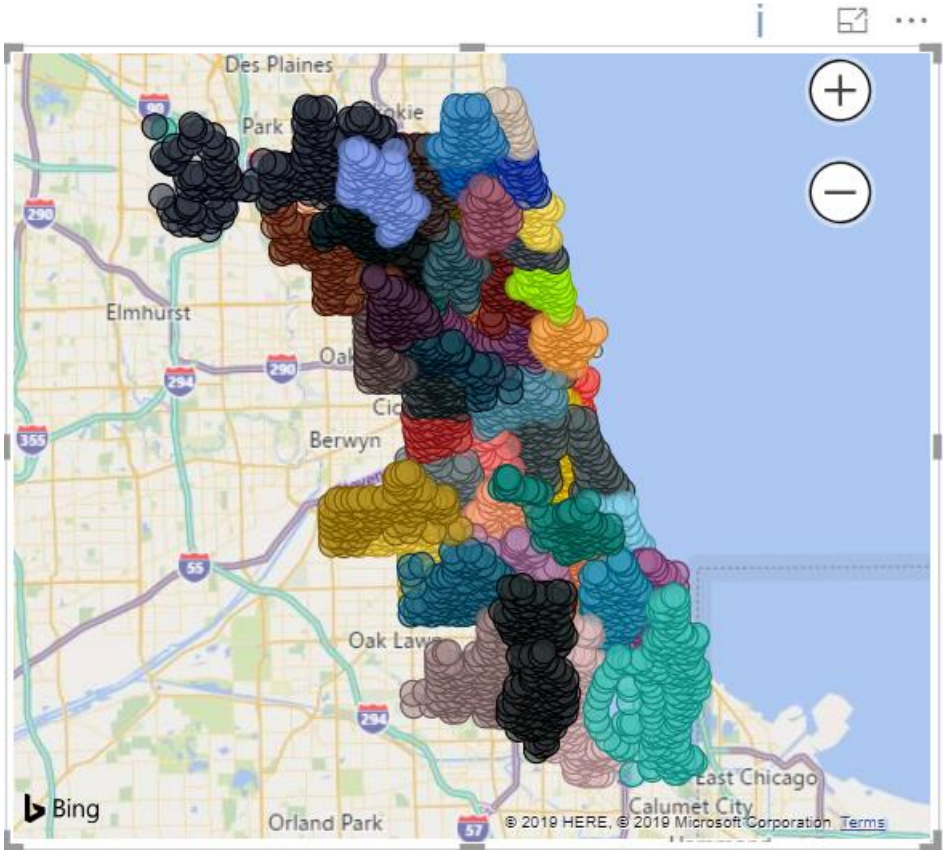
Obr. 66 První stránka reportu (vlastní zpracování)

Po přechodu na druhou stránku, která byla vytvořena ještě na začátku této kapitoly, lze hned přidat jednoduchý filtr (**Slicer**), který bude třídit všechna data na tomto listu podle roku (sloupec **Year**) (viz Obr. 67).



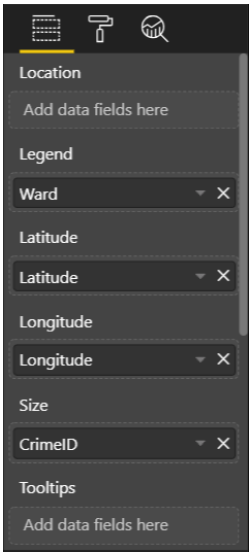
Obr. 67 Filt Year (vlastní zpracování)

Nejdůležitější vizualizací na tomto listu bude mapa, kterou je možné přidat zvolením **Map** v seznamu **Visualizations**. Do polí **Latitude** a **Longitude** je nutné přidat příslušné souřadnice z tabulky **DimLocation** a do pole **Legend** přetáhnout hodnotu **Ward**. Mapa pak zobrazuje všechna místa spáchání zločinů podle jednotlivých okrsků Chicaga, které jsou vyznačeny různými barvami (Obr. 68).



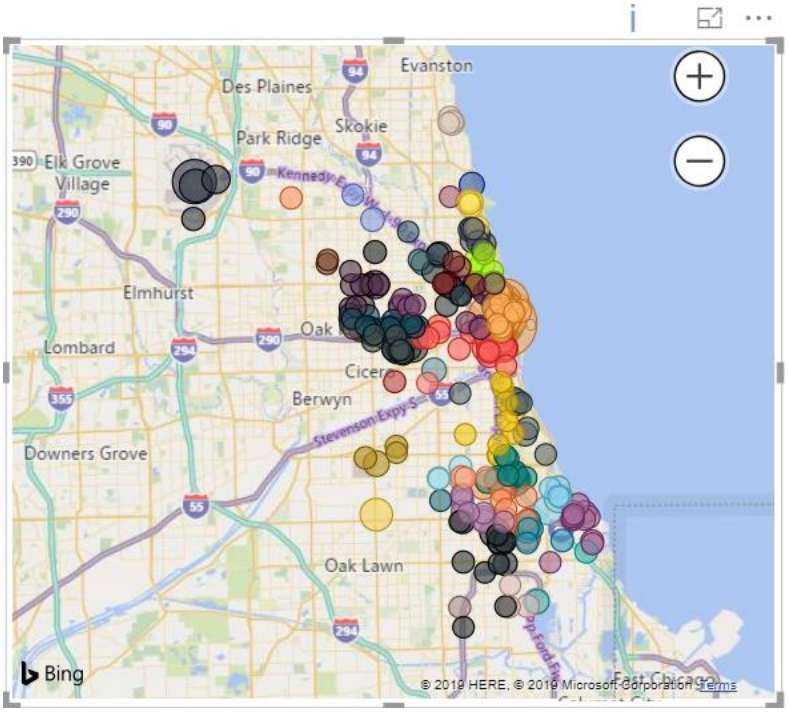
Obr. 68 Nepřehledná mapa (vlastní zpracování)

Taková mapa není přehledná, proto je potřeba snížit počet bublin vložením sloupce **CrimeID** do pole **Size** (viz Obr. 69).



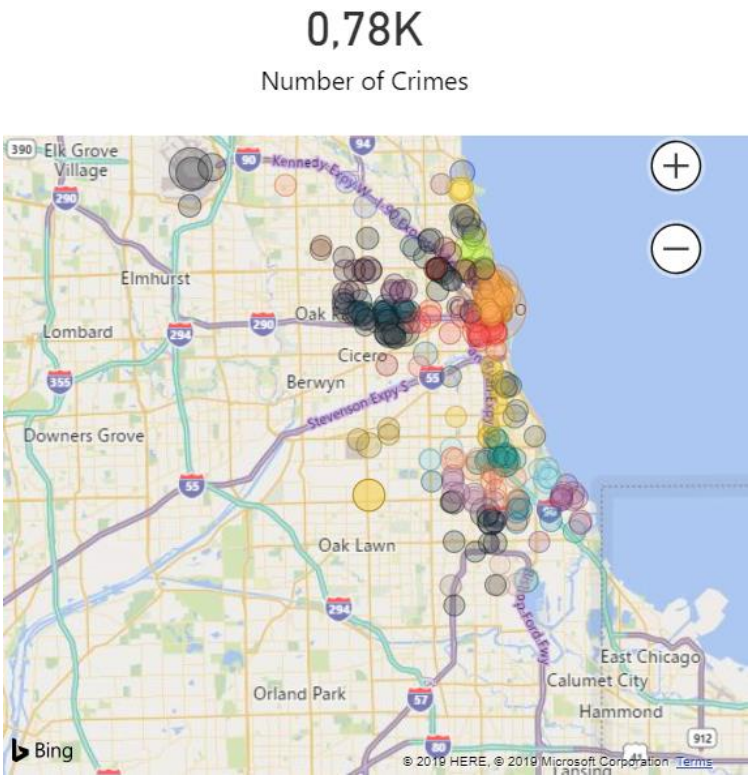
Obr. 69 Nastavení mapy (vlastní zpracování)

Teď mapa vypadá trochu jinak, protože velikost bublin (**Size**) představuje průměrný počet zločinů, a čím větší je bublina, tím větší je počet zločinů v dané lokalitě Chicaga (viz Obr. 70). Taková mapa umožňuje snadno zkoumat data pro každou lokalitu a určovat, jakého okrsku se tato data týkají.



Obr. 70 Přehledná mapa (vlastní zpracování)

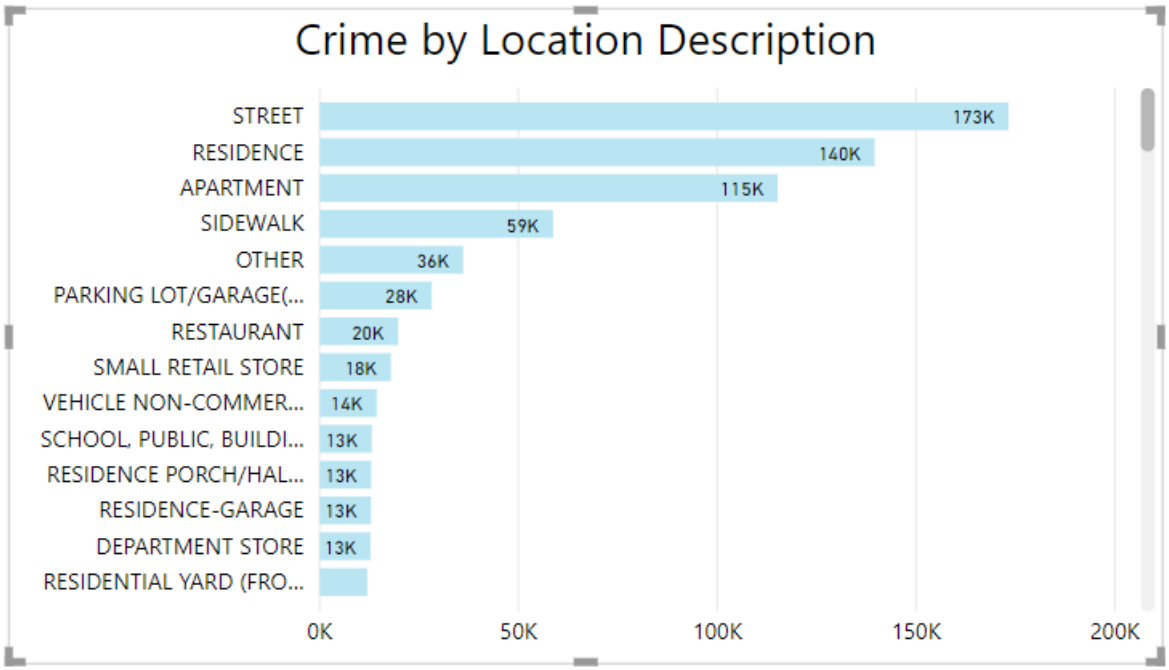
Dále bude list obsahovat kartu (**Card**), která odráží celkový počet zločinu za období 2016 – 2017. Stejná karta se nachází na první stránce reportu, proto stačí ji označit, zkopírovat a vložit na druhou stránku. Tato karta bude zobrazovat různé počty zločinů v závislosti na vybrané podmínce vizualizace. Například pokud uživatel klepne na jakoukoli bublinu na mapě, karta zobrazí průměrný počet zločinů spáchaných na tomto místě (Obr. 71).



Obr. 71 Interakce mezi mapou a kartou (vlastní zpracování)

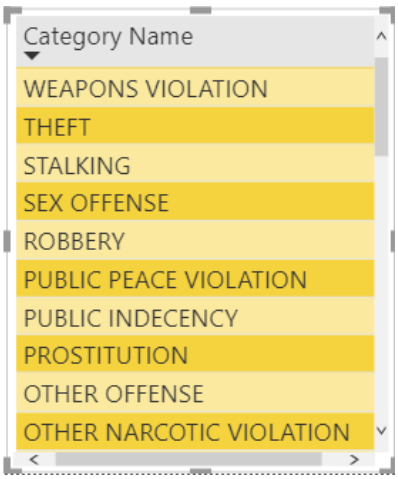
Dále pak už následuje přidání pár jednoduchých vizualizací:

Aby bylo možné sledovat, kde nejčastěji dochází k incidentům, lze přidat skládaný sloupcový graf (**Stacked bar char**), který se bude rolovat dolů (viz Obr. 72).



Obr. 72 Nejčastější místa zločinů (vlastní zpracování)

Dále je vhodné přidat tabulku (**Table**) se seznamem všech zločinů, aby mohl uživatel pozorovat, jak se mění vizualizace v závislosti na výběru jednotlivých kategorií zločinů (viz Obr. 73).



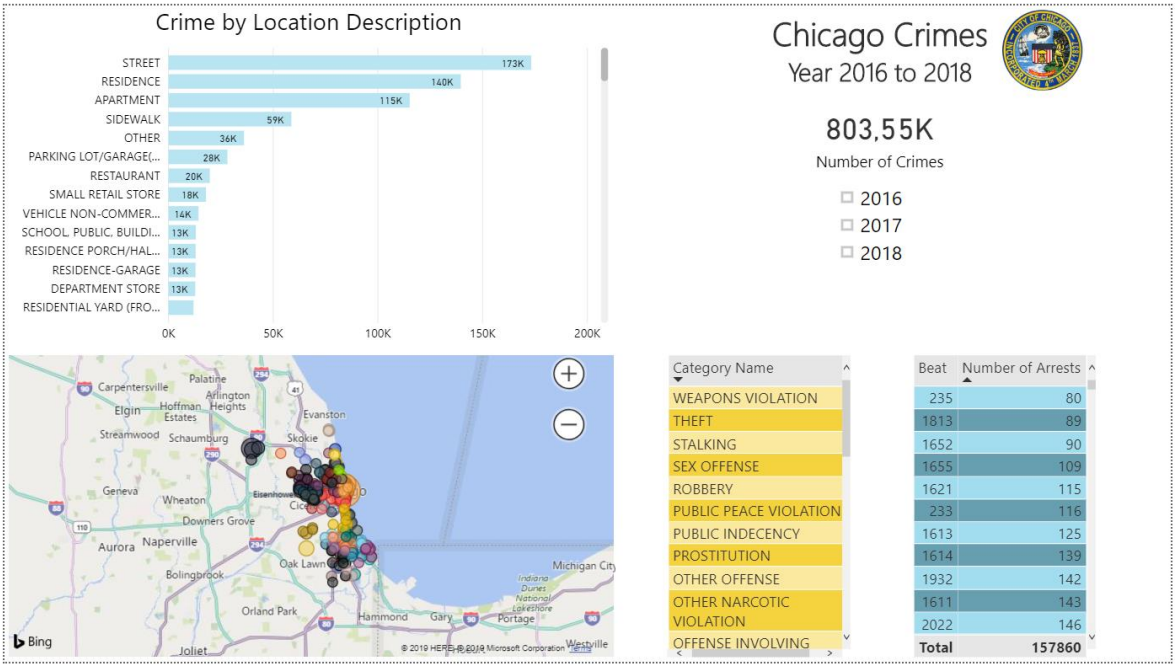
Obr. 73 Tabulka se seznamem zločinů (vlastní zpracování)

Posledním krokem je vložení ještě jedné tabulky, ze které lze vyčíst výkonnost jednotlivých policejních jednotek, která se měří počtem zatčených pachatelů (viz Obr. 74). Do pole **Values** je nutné přidat dvě hodnoty: **Beat** a **Number of arrests**.

Beat	Number of Arrests
235	80
1813	89
1652	90
1655	109
1621	115
233	116
1613	125
1614	139
1932	142
1611	143
2022	146
Total	**157860**

Obr. 74 Tabulka výkonnosti policejních jednotek (vlastní zpracování)

Všechny přidané vizualizace je potřeba narovnat a druhá stránka reportu je hotová (viz Obr. 75). Na rozdíl od první stránky dává podrobný náhled na problematiku kriminality v Chicagu a umožňuje prohlížet mapu, ze které se lze např. dozvědět, jak jsou rozmístěny okrsky Chicaga.



Obr. 75 Druhá stránka reportu (vlastní zpracování)

Teď už zbývá jenom publikovat konečný report na web, čehož lze docílit kliknutím v menu **Home -> Publish**.

Data použitá k vytváření reportu mohou být dobrá i pro jiné účely než reporting, např. pro data mining s cílem provést analýzu a poskytnout užitečné informace lidem, kteří mají zájem o prozkoumání zločinů spáchaných v Chicagu. Analýza může být dělána např. pro potřeby stavebníka, který se chystá postavit obytnou budovu v jedné z čtvrtí Chicaga. Bude ho zajímat takové vztahy v datech, které odhalí nebezpečná místa Chicaga a frekvenci

vyskytujících se kriminálních činností v jednotlivých čtvrtích tohoto města. Po provedené analýze bude stavebník schopen určit teoreticky nejbezpečnější, nejpříznivější, a tedy nejlepší čtvrť pro svou stavbu. V případě doplňování, ověřování nebo opakování analýzy v budoucnu lze využít větší dataset.

Závěr

Problematika Business Intelligence je mnohem rozsáhlejší, než bylo popsáno, avšak cílem této bakalářské práce bylo definovat principy Business Intelligence a provést analýzu BI nástrojů společnosti Microsoft za účelem jejích následného využití na reálném případu. A tento cíl byl splněn díky náplni teoretické a praktické části.

Cílem teoretické části bylo popsat základy Business Intelligence, popsat hlavní pojmy a seznámit čtenáře s klíčovými vlastnostmi BI služeb. Aby mohl čtenář vůbec pochopit podstatu Business Intelligence a její nejdůležitější technologické komponenty, byla nejdřív popsána teorie databázových systémů.

Po teoretickém seznámení s databázemi byl představen samotný pojem Business Intelligence a koncepce architektury BI. Následně byly popsány jednotlivé služby MS SQL Serveru, které jsou určeny k analýze podnikových dat s cílem podpory procesu rozhodování. Mimo jiné byl popsán rozdíl mezi multidimenzionálním a tabulárním modelem analytických služeb za účelem usnadnění výběru typu analytické služby v budoucích řešeních.

Poslední kapitola teoretické části byla zaměřena na dimenzionální modelování, které představuje základ pro návrh datového skladu. V rámci této kapitoly byly také nastíněny typy datových skladů, jejich charakteristiky a možná využití.

Praktická část se věnovala demonstraci praktického využití Microsoft BI nástrojů na příkladu konkrétních dat, a to na základě aplikací teoretických znalostí. Nejdřív byla popsána použitá data, aby měl čtenář představu o jejich struktuře. Další kapitoly praktické části zahrnovaly popis jednotlivých kroků tvorby datového skladu, ETL procesu a tabulárního modelu. Výsledkem praktické části byl vznik reportů, které představují poslední etapu celého procesu realizace BI projektu. Při tvorbě reportů byly využity mimo jiné metriky vytvořené v tabulárním modelu, které umožnily přidat do reportu zajímavé grafy.

Po dosažení cíle práce je teď zřejmé, že Business Intelligence je zásadní pro efektivní řízení dat. Implementace BI řešení umožňuje podnikům získat náhled do interních dat a přidat jim význam prostřednictvím reportů. BI přináší do podniků nové příležitosti a konkurenční výhodu.

Použitá literatura

AMBLER, S. W. [online]. Introduction to Data Normalization: A Database "Best" Practice. In: Agile Data. [cit. 26.04.2019]. Dostupné z: http://agiledata.org/essays/dataNormalization.html

BEK, D. Koncepce Microsoft prostředí pro Business Intelligence a jeho praktická aplikace. Praha, 2016. Bakalářská práce. Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky. Katedra informačních technologií.

BOGZA, R., ZAHARIE, D. 2008. Business intelligence as a competitive differentiator. IEEE International Conference on Automation, Quality and Testing, Robotics. Piscataway: IEEE. ISBN: 978-1-4244-2576-1

DePaul University. [online]. Relational Database Normalization Process. [cit. 26.04.2019]. Dostupné z: http://condor.depaul.edu/gandrus/240IT/accesspages/normalization3.htm

Dremio. [online]. Data Warehouses Explained by Dremio. [cit. 26.04.2019]. Dostupné z: https://www.dremio.com/what-is-a-data-warehouse/

ELMASRI, R., NAVATHE, S. B. 2010. Fundamentals Of Database Systems. 6. vyd. Boston: Addison-Wesley. 1172 s. ISBN-13: 978-0-136-08620-8.

FANCKE, J. 2017. [online]. BI Theory #7: Multidimensional vs Tabular. In: SelfServedBI.com. 26.11.2017 [cit. 26.04.2019]. Dostupné z: https://selfservedbi.com/2017/11/26/bi-theory-7-multidimensional-vs-tabular/

Forrester. [online]. Business Intelligence. [cit. 26.04.2019]. Dostupné z: https://www.forrester.com/Business-Intelligence

GÁLA, L., POUR, J., ŠEDIVÁ, Z. 2015. Podníková informatika: Počítačové aplikace v podnikové a mezipodnikové praxi. Praha: Grada Publishing. ISBN: 978-80-247-5457-4

Gartner. [online]. Operational Data Store. [cit. 26.04.2019]. Dostupné z: https://www.gartner.com/it-glossary/ods-operational-data-store

Guru99. [online]. Difference between Information and Data. [cit. 26.04.2019]. Dostupné z: https://www.guru99.com/difference-information-data.html

HASAN, D. 2017. [online] Data mining . In: issuu. [cit. 26.04.2019]. Dostupné z: https://issuu.com/danihasan/docs/data_mining.docx

KIMBALL, R., ROSS, M. 2002. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. 2. vyd. New York: Wiley Computer Publishing. 421 s. ISBN: 0-471-20024-7.

KROENKE, D. M., AUER, D. J. 2012. Database Processing: Fundamentals, Design, and Implementation. Boston: Pearson. 612 s. ISBN: 978-0-13-214537-4.

LACKO, Ľ. 2013. Mistrovství v SQL Server 2012: Kompletní průvodce databázového experta. Brno: Computer Press. 640 s. ISBN: 978-80-251-3773-4.

LASÁK, P. 2010. [online] Relace v databázích – teorie. In: Office školení konzultace. 19.10.2010 20:47 [cit. 26.04.2019]. Dostupné z: https://office.lasakovi.com/access/tabulky/relace-vazby-v-databazich-teorie/index.php

LOSHIN, D. 2003. Business Intelligence: The Savvy Manager's Guide. Burlington: Morgan Kaufmann Publishers. 270 s. ISBN-13:978-1-55860-916-7.

Microsoft. 2017. [online]. Reporting Services in SQL Server Data Tools (SSDT). [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/en-us/sql/reporting-services/tools/reporting-services-in-sql-server-data-tools-ssdt?view=sql-server-2016

Microsoft. 2017. [online]. SQL Server Database Engine. [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/cs-cz/sql/database-engine/sql-server-database-engine-overview?view=sql-server-2016

Microsoft. 2018. [online]. About SQL Server Analysis Services [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/en-us/sql/analysis-services/analysis-services?view=sql-server-2016

Microsoft. 2018. [online]. Comparing tabular and multidimensional solutions. [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/en-us/sql/analysis-services/comparing-tabular-and-multidimensional-solutions-ssas?view=sql-server-2017

Microsoft. 2018. [online]. Measures. [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/en-us/sql/analysis-services/tabular-models/measures-ssas-tabular?view=sql-server-2017

Microsoft. 2018. [online]. Online analytical processing (OLAP). [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing

Microsoft. 2018. [online]. Report Builder in SQL Server. [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/en-us/sql/reporting-services/report-builder/report-builder-in-sql-server-2016?view=sql-server-2017

Microsoft. 2018. [online]. SQL Server Integration Services. [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-2016

Microsoft. 2018. [online]. What is SQL Server Reporting Services (SSRS)? [cit. 26.04.2019]. Dostupné z: https://docs.microsoft.com/en-us/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-2016

MOUNTAIN, J. 2015. [online]. What is Business Intelligence? In: Wolfgroup. [cit. 26.04.2019]. Dostupné z: http://www.wolfgroup.co.uk/what-is-business-intelligence/

OTTE, L. 2013. Databázové systémy: Normalizace dat. Ostrava, 2013. Vysoká škola báňská – Technická univerzita Ostrava. Fakulta strojní.

POUR, J. 2017. F400 – BI – charakteristika komponent a přístupů. [PowerPoint prezentace]. Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky. Katedra informačních technologií. Slide 8.

POUR, J., MARYŠKA, M., STANOVSKÁ, I., ŠEDIVÁ, Z. 2018. Self Service Business Intelligence: Jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. Praha: Grada Publishing. ISBN: 978-80-271-0616-5

PROCHÁZKA, D. 2009. Oracle: Průvodce správou, využitím a programováním. Praha: Grada Publishing. ISBN: 978-80-247-2762-2.

SHAHID, U. 2018. [online]. SQL Server and Relational Database - Part One. In: C#Corner. 31.07.2018 [cit. 26.04.2019]. Dostupné z: https://www.c-sharpcorner.com/article/sql-server-and-relational-database-part-one/

SILBERSCHATZ, A., KORTH, H. F., SUDARSHAN, S. 2011. Database System Concepts. 6. vyd. New York: McGraw-Hill. 1349 s. ISBN: 978-0-07-352332-3.

Vysoká škola ekonomická v Praze. [online]. Základy relačních databází, jejich využití v programování webu. 16.8.2018 [cit. 26.04.2019]. Dostupné z: https://kme.vse.cz/wp-content/uploads/page/534/4.-Z%C3%A1klady-rela%C4%8Dn%C3%ADch-datab%C3%A1z%C3%AD-jejich-vyu%C5%BEit%C3%AD-v-programov%C3%A1n%C3%AD-webu.pdf

WITHEE, K. 2010. Microsoft Business Intelligence for Dummies. Hoboken: Wiley Publishing. 405 s. ISBN: 978-0-470-52693-4.