Podklady pro kurz Datové sklady

Jan Prokop

Bakalářská práce
2020

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav počítačových a komunikačních systémů

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Jan Prokop**
Osobní číslo:	**A15028**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informační technologie v administrativě**
Forma studia:	**Prezenční**
Téma práce:	**Podklady pro kurz Datové sklady**
Téma práce anglicky:	**Materials for the Data Warehouse Course**

Zásady pro vypracování

1. Seznamte se s problematikou a vypracujte literární rešerši na téma datové sklady.
2. Zpracujte teoretické podklady a navrhněte obsah kurzu.
3. Realizujte navržený kurz formou prezentací v PowerPointu.
4. Navrhněte a realizujte alespoň jeden vzorový příklad.
5. Připravte sadu 20 testovacích úkolů včetně řešení.

Rozsah bakalářské práce:
Rozsah příloh:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. LABERGE, Robert. Datové sklady, agilní metody a business intelligence. 1. vyd. Brno, Computer Press, 2012, 350 s. ISBN 978-80-251-3729-1.
2. RAINARDI, Vincent. Building a Data Warehouse, With Examples in SQL Server. Apress. United States of America. 2008.
3. LACKO, Ľuboslav. Databáze, datové sklady, OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle. Brno, Computer Press, 2003. ISBN 80-7226-969-0.
4. LACKO, Ľuboslav. Business Intelligence v SQL Serveru 2008, reportovací, analytické a další datové služby. Brno, Computer Press, 2009. ISBN 978-80-251-2887-9.
5. LARSON, Brian. Delivering business intelligence with Microsoft SQL server 2016. Fourth edition. San Francisco, McGraw-Hill Education, 2016. ISBN 978-1-25-964148-0.
6. KIMBALL, Ralph a Margy ROSS. The data warehouse toolkit, the definitive guide to dimensional modeling. 3rd ed. Indianapolis, Wiley, c2013, xxxiv, 564 s. ISBN 978-1-118-53080-1.

Vedoucí bakalářské práce: **doc. Ing. Zdenka Prokopová, CSc.**
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **19. prosince 2019**
Termín odevzdání bakalářské práce: **27. května 2020**

doc. Mgr. Milan Adámek, Ph.D.
děkan

doc. Ing. Martin Sysel, Ph.D.
garant oboru

Ve Zlíně dne 19. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 3.8.2020

Jan Prokop, v.r.
podpis diplomanta

ABSTRAKT

Bakalářská práce se zabývá návrhem a tvorbou datového skladu, který bude sloužit jako ukázkový příklad pro studenty kurzu Datové sklady. Teoretická část práce popisuje zařazení pojmu datové sklady do technického prostředí databází. V rámci praktické části jsou popsány jejich základní funkce: možnosti vstupů a výstupů, datové pumpy, reporting. Součástí bakalářské práce jsou rovněž výukové prezentace a sada testových otázek.

Klíčová slova: Datový sklad, DWH, ETL proces, datová pumpa, SQL

ABSTRACT

The bachelor's thesis deals with the design and creation of a data warehouse, which will serve as a sample example for students of the Data Warehouse course. The theoretical part of the thesis describes the inclusion of the concept of Data Warehouse in the technical environment of databases. Within the practical part are described their basic functions: input and output options, data pumps, reporting. The bachelor thesis also includes educational presentations and a set of test questions.

Keywords: Data warehouse, DWH, ETL process, data pump, SQL

Velmi rád bych poděkoval vedoucí své práce doc. Ing. Zdence Prokopové, CSc. za cenné rady, čas a především trpělivost. Dále bych rád poděkoval své rodině, která mě při studiu podporovala.

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

„Nepochválím-li se sám, nikdo to za mne neudělá.“

Jára Cimrman

OBSAH

ÚVOD

Na polích velkých firem vznikají obrovské objemy dat, které se musí nějak zpracovat a analyzovat. Tato data mají pro firmy velmi vysokou hodnotu, jelikož z kvalitní analýzy může vzniknout mocný rozhodovací nástroj. Správné rozhodnutí poté vede ke konkurenční výhodě na trhu.

Datové sklady jsou styčným místem mezi zdroji dat a analýzami. Za datovým skladem stojí velká řada činností např. pumpování dat, čištění dat, sledování integrity dat, přípravy reportů.

Cílem této bakalářské práce je vytvoření podkladů pro náplň kurzu Datové sklady. Tento kurz studenty seznámí s teoretickými okruhy, ale především také s jejich praktickým použitím na vytvořeném příkladu. Student bude mít možnost osahat si a vyzkoušet veškeré činnosti datového skladu. Studentovi budou představeny jak projekty v grafickém rozhraní, tak procedury SQL jazyka.

Součástí práce bude také sada testových otázek, která vyučujícímu a studentům poskytne zpětnou vazbu o pochopení problematiky datových skladů.

I. TEORETICKÁ ČÁST

1 ÚVOD DO TÉMATIKY

Hlavním prvkem všech institucí v jakémkoliv odvětví jsou v dnešní době především údaje, jejichž množství představuje až téměř nepředstavitelnou hodnotu. Hlavním úkolem datových skladů (Data Warehouse - DWH) je tyto údaje usměrňovat a ukládat tak, aby byla dosažena co nejjednodušší a nejvyšší dostupnost a zpracovatelnost.[1]

1.1 Historie databází

Za první předchůdce databází lze z historického hlediska považovat papírové kartotéky a dokumenty, v nichž byla data uložena vždy podle nějakého řádu. Následně se do kartoték začala ukládat data v podobě děrných štítků. Jednalo se o vstupy a výstupy elektromechanických počítačů. V 60. letech 20. století byly děrné štítky vytlačovány magnetickými úložišti, páskami a poté diskovými úložišti. Děrné štítky ovšem přetrvávali téměř až do 80. let Nástup diskových úložišť byl velmi rychle následován vývojem softwaru nazývaném Database Management System (DBMS), jehož největší výhodou byla možnost vyhledávání. Jeho princip spočíval ve stahování požadovaných dat z databáze organizace a umožňoval je přenášet, seřazovat, dále jinak zpracovávat a zase je pak navrátit na původní místo. Tento proces měl být navíc nesmírně rychlý.

V dalším desetiletí se ke slovu dostaly první komerční online aplikace, jako byly například vzdálené ovládací prvky výrobních procesů, bankovní systémy řídící bankomaty aj. V 80. letech získávají na popularitě relační databáze, jejichž hlavním plusem je uživatelská srozumitelnost na rozdíl od předchůdců. Spolu s nimi přichází také SQL jazyk, který se využívá až dodnes. Na konci této dekády se pro připojení k databázi běžně využívá model klient-server (méně výkonné pracovní stanice, výkonné servery).

S novým miléniem přichází také nový problém, a to velké množství dat z různých aplikačních systémů, které podniky využívaly, a nebylo možné tato data dále slučovat.

Potřeba slučování dat odstartovala vývoj první datových skladů.

Logicky se vývoj databází a systémů s nimi spojenými pohyboval kupředu tak, jak pokročil vývoj hardware (HW) a počítačových sítí.

Mezi osobnosti, které se historicky podílely na vzniku databází, a to konkrétně sběrem dat, patří také Tomáš Baťa. Jeho myšlenky vedli nepřímo k optimalizaci systémů ERP, mezi které se řadí velmi známý nástroj SAP. [2]

1.2 Aktuální trendy

S vývojem technologií roste i náročnost na kapacitu a výpočetní výkon. Databáze jsou v jednotkách až stovkách gigabyte a jejich velikost roste. S tím se odvíjí i techniky přístupu k získávání dat, rostou nároky na bezpečnost a odolnost proti výpadkům.

1.2.1 Business Inteligence

Business Inteligence je proces získávání nebo také transformace velkého množství údajů na informace a jejich převod na poznatky. Podle firmy Oracle se údaje stávají informacemi, jsou-li splněny tyto body:

- Máme údaje
- Víme, že máme údaje
- Víme, kde tyto údaje máme
- Máme k údajům přístup
- Zdroji údajů můžeme důvěřovat. [11]

Poznatky se poté stávají velmi důležitým prvkem v rozhodovacím procesu, kde jsou tyto informace sledovány také z dlouhodobého hlediska (informace tvoří trendy, vykazují souvislosti). Jedná se tedy o soubor několika činností, jichž je DWH součástí. [11]

1.2.1.1 OLAP – Analytické, Big Data

OLAP (On Line Analytical Processing) je online analytický multidimenzionální vyhledávací systém. Vyhledává data pro rozhodovací analýzy. O formátu těchto analýz = výstupů rozhodují uživatelé (pracovníci), především management.

Základní rozdíly mezi klasickým OLTP (On Line Transactional Processing) a OLAP jsou popsány v tabulce Tab. 1:

	OLTP	**OLAP**
Základní srovnání	Je online transakční systém, provádí změny v datech	Je online systém pro získávání a analýzu dat.
Zaměření	Vkládání, aktualizace, mazání dat v databázi	Načítání dat pro rozhodovací analýzy
Data	OLTP je nositelem dat, data jsou uložena přímo v databázi a je s nimi operováno	OLAP využívá OLTP jako svůj zdroj dat
Transakce (Spojení)	Krátké transakce	Dlouhé transakce
Čas provádění dotazu	Dotazy jsou prováděny rychle	Dotazy se provádějí dlouho
Dotazy	Jednoduché dotazy	Komplexní složitější dotazy
Normální formy	Je vyžadována třetí normální forma (3NF)	Normalizace dat není vyžadována
Důvěryhodnost	S každým příkazem se může změnit integrita dat	Nemění se často, důvěryhodnost dat není zasažena

Tab. 1 Srovnání OLTP a OLAP [7]

Big Data jsou velké složité sady dat, pocházející především z nových zdrojů. Množství dat dosahuje takových rozměrů, že nejsou zpracovatelné tradičním softwarem. Vyskytují se především v obchodním odvětví. Pro Big Data jsou specifické „tři v“:

- Volume (objem)
  - Rozmezí velikosti Big Data se pohybuje od desítek terabytů po stovky petabytů
- Velocity (rychlost)
  - Rychlostí se rozumí, že data musí být zpracována od přijetí až po zápis do paměti co nejrychleji. V některých případech je dokonce požadováno real-time zpracování
- Variety (rozmanitost)
  - Big Data nejsou jen strukturovaná data, ale také různé datové typy jako je text, zvuk nebo video. Aby získaly význam, procházejí tzv. předzpracováním

Big Data se vyžívají například v:

- Rozvoj produktů – predikce poptávky
- Zákaznická zkušenost – sledování návštěv webů
- Bezpečnost – sledování chování sítě, pro možný výskyt hackerů
- Strojové učení [20]

1.2.2 Technická realizace

- CloudComputing: Moderní je také využití cloudcomputingu a využívání tzv. DaaS (Database as a Service). Tímto řešením firmy šetří především nemalé částky za pořizování nového HW a licencí na potřebné systémy. Důležitou výhodou je vysoká dostupnost a odolnost proti výpadkům s použitím technologie georedundance (pokud dojde v lokalitě datového centra k havárii, živelným pohromám, regionálním výpadkům, jsou data stále dostupná ze záložní lokality např. Praha x Brno). Někteří dodavatelé dokonce v základních konfiguracích nabízí službu cloudcomputingu zcela zdarma a škálovatelnost výpočetního výkonu v těchto případech znamená jen zpoplatnění nebo zvýšení poplatků, které jsou účtovány za hodinu, místo nákupu, vývoje a nasazení (deploymentu). Další zajímavou funkcí, kterou datacentra nabízí, je automatické škálování. Pokud na serveru dosáhnu určitého výkonu, je automaticky výkon přidán.
- Bezpečnost: Důraz je čím dál tím více kladen také na bezpečnost databází zvláště pak na osobní údaje, k jejichž ochraně se zpravidla využívá dynamické maskování. Dynamické maskování zabraňuje čtení údajů pro neoprávněného uživatele. Textový řetězec, číslo a také časové údaje se zamění buď za výchozí hodnotu nebo za předem stanovený počet X. Takto maskovaný údaj může vypadat např. takto: JXXXXX PXXXXX. Za úniky osobních údajů jsou firmám účtovány až několikamilionové pokuty.
- Jednoduchost: Hlavním rysem moderních databází je především jednoduchost. Databáze by měla být jednoduchá pro administrátory, aby její údržbou neztráceli čas, datové analytiky, aby se ve struktuře dobře orientovali, i pro manažery, kteří by měli bez většího informačně-technického vzdělání tvořit výstupy (reporty). Ke zjednodušení dochází automatizací některých procesů.
- Tabulka s paměťovou optimalizací: Standardně SQL servery počítají přístupy do tabulek a dělají statistiku přístupů. Často navštěvované tabulky ukládá do cache, která

je uložena v paměti. Tabulky, které jsou méně používány, jsou z cache automaticky odebírány, aby se ušetřila paměť. V případě, že do některých tabulek přistupuji poprvé, dotazy trvají déle, opakované přístupy jsou poté rychlejší. Případně když chceme, aby tabulka nebyla z cache odebrána, přepneme tabulku do režimu paměťově optimalizovaná tabulka. [17]

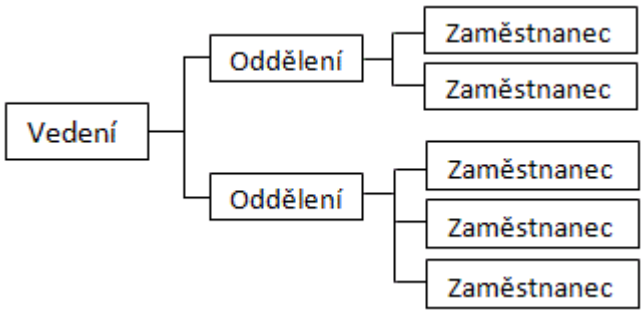
- OLTP (On – Line Transaction Procesing): Oblast použití OLTP je téměř neomezená. Úkolem OLTP je umožnit uživatelům databázového serveru vykonání velkého množství transakcí online. Ke zdroji údajů tedy ve stejném čase přistupuje velké množství uživatelů, někteří údaje čtou, jiní zapisují. Cílem transakčních databázových systémů je automatizace periodických činností. Transakční systémy jsou hojně využívány především z důvodu existence velkého množství administrátorů a vývojářů. Hlavní výhodou je široké spektrum použití. Pokud OLTP pokrývá většinu našich podnikových aktivit, mluvíme o ERP (enterprise resource planning). [3, 4, 5, 6, 11]

1.3 Datové modely

Existuje několik druhů možností ukládání dat. Vazby mezi nimi dělíme do několika druhů. Při návrhu DWH je možné se rozhodnout, který druh bude využitý, případně je možné využít i jejich kombinaci.

1.3.1 Hierarchická stromová struktura

Každý záznam představuje uzel ve stromové struktuře. Vzájemný vztah mezi záznamy je typu rodič - potomek. Výhodou je jasně daný vztah mezi záznamy. Nevýhodou je samotné hierarchické uspořádání, které je omezující pro operace jako je vkládání či rušení záznamů.[15]



Obr. 1 Stromová struktura [15]

1.3.2 Relační struktura

Relační databáze jsou takové databáze, které ukládají a přistupují k datům, které spolu souvisí přímočaře a intuitivně. V relačních databázích má každý záznam v rámci jedné tabulky svůj jedinečný identifikátor, který se nazývá klíč. Sloupce tabulky obsahují atributy. Zpravidla má každý záznam hodnotu pro každý atribut.

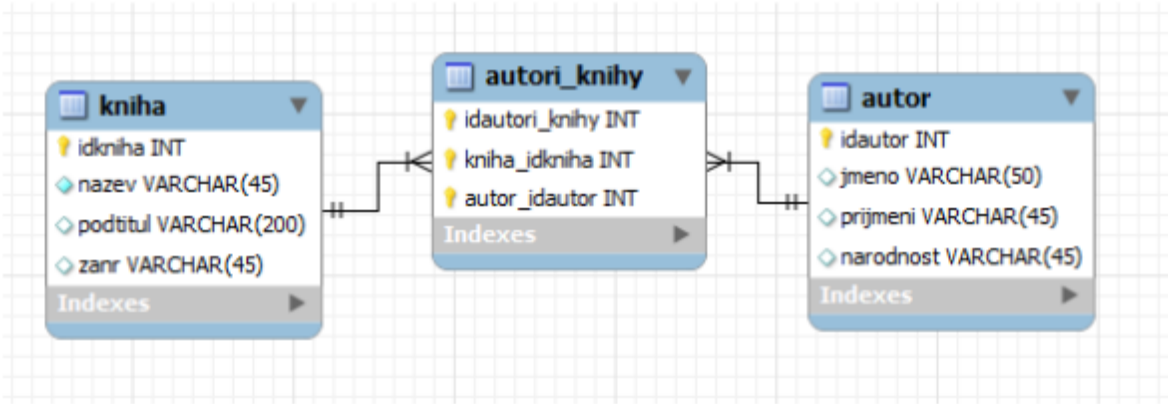
Struktury takové databáze jsou od sebe odděleny, zvlášť je struktura datová a zvlášť je struktura fyzického úložiště. U relačních databází je vyžadována nulová „redundance“, tj. záznamy v tabulkách nesmí být duplicitní.

Vztahy v databázích

1:1 - Pokud záznamu v tabulce odpovídá právě jeden záznam v jiné tabulce (jeden zákazník má právě jedno rodné číslo).

1:N - Pokud záznamu v tabulce odpovídá více záznamů v jiné tabulce (jeden zákazník má několik objednávek).

M:N - Pokud záznamu v jedné tabulce můžeme přiřadit libovolný počet záznamů z druhé tabulky a také záznam v druhé tabulce lze také přiřadit k více záznamům v tabulce první (jeden druh zboží je ve více objednávkách, v jedné objednávce je více druhů zboží). [8]



Obr. 2 Příklad relační databáze [14]

1.3.3 Objektově orientovaný datový model

Objektově orientovaný datový model se více podobá běžným objektům. Tento typ nepoužívá řádky, ale každý objekt má své třídy, které jej charakterizují. Objekty jsou uloženy v kolekcích, které mezi sebou mají logickou strukturu.

Objekty se skládají z datové složky a z metod, které objekt provádí. Metody jsou tudíž jediným způsobem přístupu k hodnotám v objektu. Každý objekt má svůj jedinečný identifikátor OID (object identifier).

Jsou podobné objektům z objektově orientovaného programování. [21]

Relační datový model	Objektově orientovaný model
Relace (tabulka)	Kolekce (množina objektů – i z více tříd)
N-tice (řádek)	Objekt
Atribut	Datová složka a metody objektu
Primární klíč (na logické úrovni)	OID (na fyzické úrovni)

Tab. 2 Srovnání relačního a objektového modelu [21]

1.3.4 NoSQL

NoSQL jsou databáze s rychle se měnícími typy dat, strukturovanými, nestrukturovanými a polymorfními. Hlavními znaky těchto databází je rychlý vývojový cyklus, vysoká dostupnost dat a vysoce optimalizované úložiště. [16]

1.4 Datový sklad

Bill Inmon, který je považován za otce datových skladů, ho definoval takto:

„Datový sklad je podnikově strukturovaný depozitář subjektově orientovaných, integrovaných, časově proměnlivých historických dat použitých na získávání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.“ [11]

Pokud chceme definici porozumět, je potřeba vysvětlit si její jednotlivé pojmy:

- Subjektová orientace: Údaje v datovém skladu jsou orientovány na konkrétní subjekt nikoliv aplikaci, ze které údaje pocházejí. V praxi to znamená, že jsou údaje za zákazníky soustředěny do kategorie zákazník, ať už pocházejí z objednávkového nebo marketingového informačního systému.
- Integrovanost: Datový sklad má jednotnou podobu. Údaje týkající se jednoho subjektu musí mít stejný tvar (jednotou terminologii; F ≠ False). Proto se údaje před uložením do DWH transformují.

- Časová variabilita: Údaje v DWH nejsou pouze aktuální, ale také za určité časové období, proto musí být jasné, jakému časovému období údaje odpovídají.
- Neměnnost: Údaje v DWH nejsou určeny k editaci ani mazání, ale pouze ke čtení.

Hlavní rozdíly mezi produkční databází a datovým skladem jsou popsány v tabulce níže.

Vlastnost	Produkční databáze	Datový sklad
Čas odezvy	Zlomky sekund až minuty	Sekundy až hodiny
Operace	Manipulace s daty	Čtení dat
Původ dat	Max 60 dní staří	Libovolný časový úsek
Organizace dat	Dle aplikace	Dle subjektu
Velikost	Malá až velká	Velká až velmi velká
Zdroje dat	Interní	Interní i externí
Činnosti	Procesy	Analýza

Tab. 3. Rozdíl mezi produkční databází a datovým skladem [11]

Z definice a srovnání vyplývá, že datový sklad není jen databáze, ale také pravidla, kterými se řídí.

1.4.1 Data Mart

Data Mart čili datové trhy jsou podmnožiny datového skladu. Čerpají z nich především menší oddělení firem. Vznikají před dokončením projektu tvorby datového skladu, kdy je tímto způsobem firmě předáno „aspoň nějaké“ řešení. Vznikat mohou ale i po dokončení. Mají na ně vliv způsoby tvorby datového skladu [11]

2 VÝVOJ DWH

Při vývoji a realizaci DWH existuje několik možností, jak postupovat. Tyto postupy jsou popsány v následujících kapitolách. Rozhodnutí, jak postupovat, záleží na konkrétních požadavcích, kritériích pro řešení, datových vstupech.

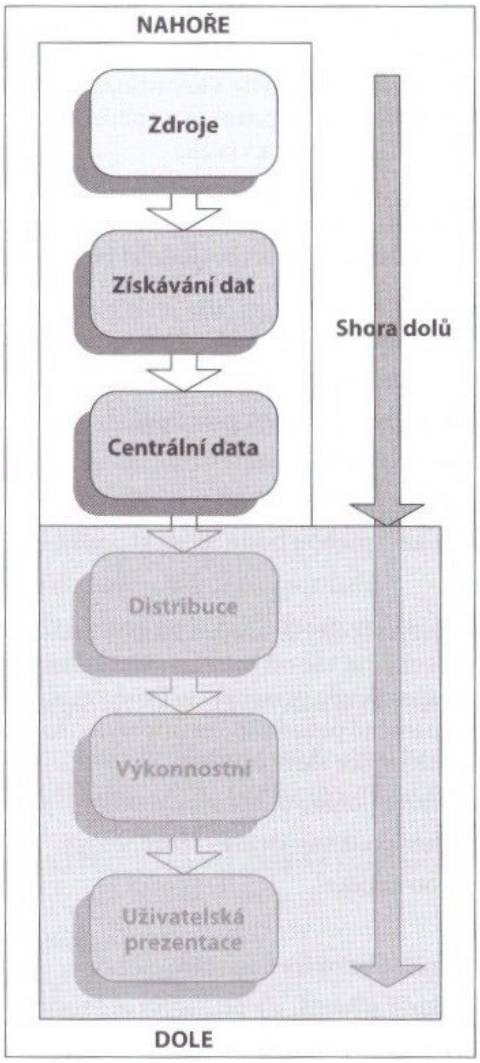
2.1 Metody tvorby datového skladu

Datové sklady vznikají zpravidla postupnou tvorbou již z infrastruktury nebo její transformací. Ve velmi malém procentu případů vzniká datový sklad jako celek od nuly. Jedním z takových možných případů je vznik rozsáhlého a sofistikovaného podnikání, za kterým stojí silný investor. V tom případě se bavíme o metodě vzniku, která se jmenuje „velký třesk“. [11,27]

V ostatních případech podniky ke tvorbě datových skladů přistupují postupně čili přírůstkově. Pracovní tým, který se zabývá tvorbou datového skladu, se skládá z několika členů z různých oddělení podniku (např.: programátoři, analytici, vedoucí oddělení, kteří budou z DWH čerpat data, stanoví a sumarizují požadavky). [27]

2.1.1 Přírůstková metoda „Shora dolů“

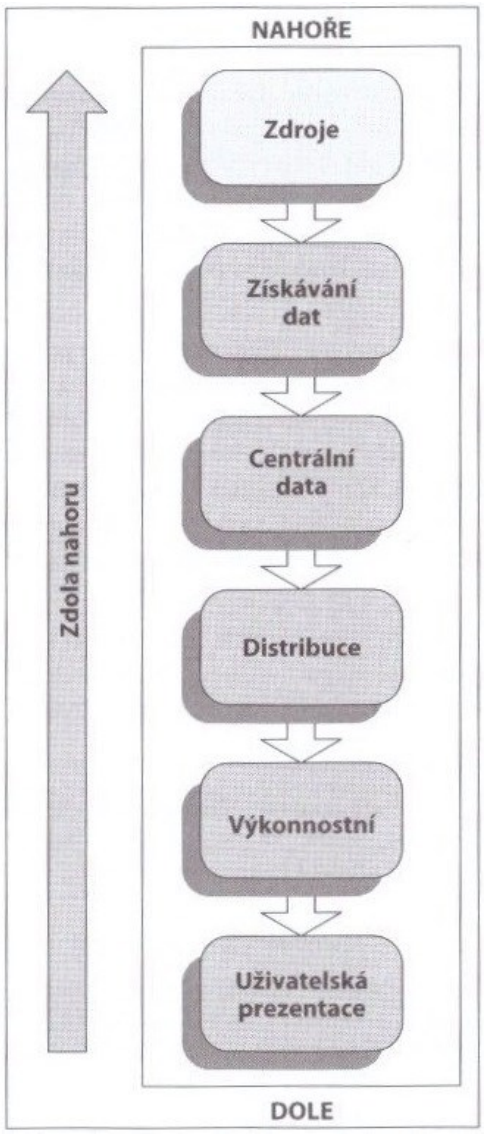
Na základě požadavků se vytvoří model datového skladu, přičemž se stanoví hierarchie předmětných oblastí. Následně se vytvoří datové trhy jednotlivých předmětných oblastí. Datový sklad uchovává atomické nebo transakční data, která jsou extrahována z jednoho či více zdrojových systémů a integrována do normalizovaného datového modelu. Z něho jsou data sumarizována a sestavují se z nich struktury dimenzí, následně jsou distribuována do datového tržiště. Organizace mohou doplnit datový sklad o dočasné úložiště pro ukládání dat zdrojových systémů, odkud jsou načítána do datového skladu. Takové dočasné úložiště může být užitečné, především pokud je zdrojových systémů několik nebo pokud jsou většího objemu. Tato metoda by se dala přirovnat ke kuchaři, který první zkontroluje suroviny, které má a poté z nich začne vařit. [9,11]



Obr. 3 Metoda Shora dolů [9]

2.1.2 Přírůstková metoda „Zdola nahoru“

Metoda je velmi podobná metodě „shora dolů“. Touto metodou je dosaženo podnikových obchodních přínosů vystavěním datových tržišť v co nejkratším možném čase. Na rozdíl od metody „shora dolů“ obsahují tato datová tržiště veškerá datová aktiva společnosti (atomická data), která mohou koncoví uživatelé datového skladu analyzovat nejen ihned po dokončení datového skladu, ale i v budoucnu. Každé další tržitě je pak stavěno vedle již existujícího a všechny využívají již vytvořených dimenzí a faktů, což umožnuje uživatelům pracovat napříč těmito tržišti. Hlavní výhodou této metody je zaměření na vytváření uživatelských struktur, což podniku přináší přínosy v kratším časovém období. Pokud opět použijeme přirovnání s kuchařem, znělo by takto: Kuchař nejprve od hostů zjistí, jaký pokrm by si přáli a poté jde vařit. [9,11]



Obr. 4 Metoda Zdola Nahoru [9]

2.2 Požadavky na systém

Při projektové části návrhu DWH je nutné stanovit požadavky, které bude datový sklad splňovat. Tyto požadavky formuluje zadávající útvar spolu s útvarem odpovídajícím za vývoj. Mezi nejdůležitější požadavky patří především typ čerpaných dat, jejich frekvence sběru z jednotlivých umístění a jak bude sběr prováděn, zda ručně nebo automaticky. Velmi důležitým požadavkem je také požadavek na zobrazování dat z DWH. Dalším nejčastěji diskutovaným požadavkem je cena celkového řešení. Tyto požadavky jsou poté zahrnuty do dokumentace, která pečlivě popisuje každou část datového skladu.[18]

2.2.1 Vstupní a výstupní data

Určení vstupních dat do datového skladu je nejdůležitější požadavek na celý datový sklad. Podnik určí, které systémy budou do DWH vstupovat, a zajistí, aby data mohly tento systém opustit k dalšímu zpracování. Mezi vstupní data patří kromě jednotlivých systémů, jakým je např. Money, také textové soubory, listy excelu, databáze a výpisy hovorů z telefonní ústředny.

V případě výstupních dat existuje několik možností realizace. Buď to se jedná o jednoduché tabulkové zobrazení (view), které mohou nastavit na online zobrazení přímo programátoři datového skladu, nebo může být výstup realizován pomocí sofistikovaných reportovacích nástrojů. O nastavení reportovacích nástrojů se starají datoví analytici. Ti ve spolupráci se zadavateli sestaví reporty, dashboardy atp., aby jejich využití pro podnik bylo co nejefektivnější. K výstupním datům mohou přistupovat i ostatní podnikové systémy. Pro ně je potřeba také nastavit výstupní formát. [11,18]

2.2.2 Požadavky na výkon, odolnost proti výpadku, bezpečnost

Další požadavky na datový sklad mohou být z pohledu výkonu, tj. jaký výkon bude v datacentru pronajat, jaká bude velikost úložiště, typy pamětí nebo třeba rychlost připojení. Další z pohledu odolností proti výpadku, tím se rozumí dostupnost služby, zda jsme schopní udržet servery v provozu i v případě oprav, výpadku energie či živelné pohromy. V tomto pohledu jasně vítězí datová centra, která jsou schopna se smluvně za dostupnost zaručit. Další pohled je na bezpečnost. Především se jedná o stanovení zabezpečeného spojení, určení osob oprávněných k administraci.

2.3 Volba vhodného databázového systému

Dle požadavků určených k provozu datového skladu je potřeba vybrat systém vhodný pro realizaci tak, aby vyhovoval většině požadavků a umožňoval růst systému do budoucna. Obecně by se projekty daly rozdělit na malé projekty a enterprise projekty (podniková architektura).

2.3.1 Malé projekty (MySQL, PGSQL, MSSQL Express...)

Jako malý projekt je možné si představit například databázi rybářských povolenek pro oblastní spolek rybářů, který se rozhodl dál nepoužívat Excel, protože přestal být pro použití

přehledný. Z větší části mezi malé projekty patří většina malých a středních projektů, které mají maximálně 1 – 2 databáze, a to také na serveru, který sami většinou neprovozují a databáze kupují například jako službu u poskytovatele webových stránek nebo např. Microsoft Azure. Obecně je možné říct, že malé projekty obsahují maximálně 5GB zpracovaných dat. Pokud se podniku daří a expanduje, je velmi jednoduché přejít na řešení Enterprise architektury.

2.3.2 Enterprise architektura

Mezi velké projekty se v dnešní době řadí většina řešení pro téměř všechny podniky z různých odvětví, ať už automotive, bankovnictví nebo prodej služeb. Společným měřítkem těchto projektů je především touha podniků za dosažením vyšších zisků. Nejčastěji jsou využívány technologie firem Microsoft (SQL Server, Azure) a Oracle. Velikost takových projektů se pohybuje mnohdy i v řádech TB a je využíváno především databázových serverů, které podniky spravují samy a mají je pronajaty v datových centrech.

2.4 Vývoj a provoz

Při vývoji datových skladů se postupuje podobně jako při vývoji software. Podle okolností a požadavků na DWH se vytvoří vývojově-testovací prostředí označované jako dev-test. Toto prostředí je zpravidla 1:1 stejné jako prostředí provozní. Před uvedením do produkce je potřeba důkladně otestovat veškeré součásti, jako jsou pumpy, procedury, views… Pokud vše funguje, jak má, tak je z vývojového prostředí vytvořen instalační balíček pomocí speciálního softwaru (např.: SQL Toolbelt) a poté je teprve proveden kompletní proces nasazení (deployment).

Některé firmy kromě stavby vlastních datových skladů provozují také vlastní IS, které DWH využívá. Pokud je informační systém zpětně kompatibilní a je u něj nasazena nová verze, může být nová verze datového skladu nasazena libovolně. V případě, že je vydána verze, která zpětně kompatibilní není, musí být verze DWH nasazena zároveň s verzí IS, a to ve stejný čas. Pokud v tomto případě vznikne problém, je velmi náročné provést obnovení všech dotčených systémů (rollback). Zpět na původní verzi musí jak datový sklad, tak i informační systém a také všechny ostatní součásti.

Je také nutné dbát na to, že i přes stanovené požadavky může dojít k jejich změnám anebo rozšíření, ať už během vývoje, nebo provozu. Tyto možné nuance je potřeba také zohledňovat a pokud možno s nimi dopředu počítat. [27]

2.4.1 ETL

ETL (extract, transform, load) je velmi důležitá a nepostradatelná součást datového skladu. Představuje komplexní proces získání, úpravu a přemístění dat do datového skladu. Tento proces je časově nejnáročnější ze všech aktivit, které se DWH týkají, může zabrat až cca 70 % času. Jeho hlavním cílem je sjednocení údajů, což znamená, že data shromáždí z několika různých zdrojů, ty tvoří především klasické OLTP databáze. Těmito údaji, které také vyčistí, poté datový sklad naplní. Jelikož data v datovém skaldu musí mít jednotnou formu, zahrnuje ETL také indexaci, sumarizaci, změnu struktury klíčů atd. Tato činnost není jednorázová, nýbrž běží periodicky po celou dobu trvání celého DWH. Procesem ETL nesmí být narušena kontinuita činností systémů, ze kterých jsou data čerpána. [10,11]

2.4.1.1 Extract - extrakce

Extrakce je prvním krokem ETL procesu. Údaje, které se snažíme přenést do datového skladu, jsou zpravidla rozmístěny v různých nesourodých prostředích, hardwarových platformách, operačních systémech (PC, Mac, Linux…), databázových systémech (Oracle, MS SQL Server, MySQL), podnikových systémech (SAP) a jiných systémech (Money, Pohoda). Údaje se zpravidla v těchto místech vyskytují v různých formátech.

Ne všechny zdroje musí být nutně vnitrofiremního původu. Těmto zdrojům se říká externí. Mohou být dostupné například volně na internetu. Sběr těchto údajů probíhá na rozdíl od všech ostatních nepřetržitě vždy, když jsou dostupné.

Úkolem extrakce je získat údaje z takovýchto zdrojů. Údaje jsou extrahovány v nezměněné podobě do vrstvy L0 (viz. kapitola 2.4.4.1) datového skladu.

Existují dva hlavní druhy extrakce, a to úplná a inkrementační (přírůstková).

Příkladem úplné extrakce může být exportní soubor samostatné tabulky nebo vzdálený příkaz SQL skenující kompletní zdrojovou tabulku. U přírůstkové metody jsou extrahovány údaje, které byly vytvořeny nebo změněny pouze od doby stanovené v minulosti. Tímto „bodem“ může být například poslední provedení extrakce. Pro identifikaci těchto změn musí existovat jednoznačný identifikátor. Identifikátorem může být sloupec s časovým razítkem nebo historizační tabulka, kterou řídí dodatečný mechanismus.

Většina datových skladů nepoužívá žádnou techniku na zachycování změn jako součást extrakčního procesu. Jako identifikaci používá porovnání celých extrahovaných údajů oproti datovému skladu.

K extrakci můžeme využít několik způsobů postupů či technologií. Jednoduché i složitější procedury můžeme navrhovat sami v jazyce C++ nebo v nadstavbách jazyka SQL. [11,19]

2.4.1.2 Transform - transformace

Nekvalitní data nemají pro datový sklad žádný význam. Pokud je u dat snížena srozumitelnost a důvěryhodnost, ztrácí reporty a analýzy z datového skladu význam, protože by mohli vést ke špatnému rozhodnutí toho, kdo z nich čerpá. K tomu, aby data byla využita co nejefektivněji, musíme data transformovat do jednotného formátu. [11]

Transformace je tedy soubor činností a úloh, díky nimž dochází ke zkvalitnění údajů a odstranění chyb. Jedním z typických problémů, je například použití rozdílného kódování, gramatické chyby, sjednocení formátů veličin, datumů. [11]

Dá se říct, co zdroj vstupující do DWH, to jiná kvalita údajů. Některé můžou být dokonce nejednoznačné. Je to především díky lidskému faktoru, který zadává jednotlivé záznamy do zdrojových systémů, zvláště pak je-li vstupní formulář veřejně přístupný. Proto tyto údaje čistíme (třídíme – mažeme). Kupříkladu máme dotazník spokojenosti zákazníků v prodejně, kde se objeví něco takového:

id	jmeno	prijmeni	mesto	email	po-bocka	hodnoceni	poznamka
5521	Čert	Rohatý	Peklo	ahoj@vpe-kle.com	Praha	3/5	Brzo si vás od-nesu všechny

Tab. 4 Ukázka nechtěných údajů

V takovémto případě nevíme, jestli byl vyplňovatel spokojen nebo ne, nebo zda-li byl vůbec naším zákazníkem. Můžeme se z takového záznamu ale i poučit. Například ve formuláři můžeme pole „pobocka“ zaměnit za výběr ze seznamu. Budeme – li mít totiž více poboček v jednom městě, byla by odpověď „Praha“ nejednoznačná.

Během transformace musíme dbát také na nulové záznamy. Ty doplníme, pokud můžeme z jiného zdroje, pokud ne, zkusíme je doplnit vhodným příznakem. U měn dbáme hlavně na změny hodnoty. Např. na Slovensku došlo ke změně měny ze slovenské koruny na euro, 1 SKK ≠ 1 EUR (ve sloupečku cena). U číselných vstupů je potřeba dále dbát na poštovní směrovací čísla, rodná čísla a telefonní čísla.

Pokud čerpáme z databáze jiného typu například stromového, nesmíme zanedbat jejich vztahy, aby některá data ve větvi nezůstala opomenuta.

jmeno	pohlavi
Josef Vzteklý	muz
Martina Nováková	Žena
Čestmír Čestný	M
Soňa Ivanová	zena
Martin Šťastný	pan
Klára Dlouhá	paní

Tab. 5 Sloupec pohlavi před transformací [11]

jmeno	pohlavi
Josef Vzteklý	M
Martina Nováková	Z
Čestmír Čestný	M
Soňa Ivanová	Z
Martin Šťastný	M
Klára Dlouhá	Z

Tab. 6 Sloupec pohlavi po transformaci [11]

2.4.1.3 Load - načtení

Načtení nebo také přenos je poslední fází ETL. Během prvního přenosu zpravidla „teče“ do datového skladu velké množství dat. Následné datové dávky již nedosahují takových velikostí. Přenos by měl být plánovaný, pravidelný a automatizovaný. Po zavedení dat probíhá jejich indexace.[11]

2.4.2 Dostupné nástroje ETL

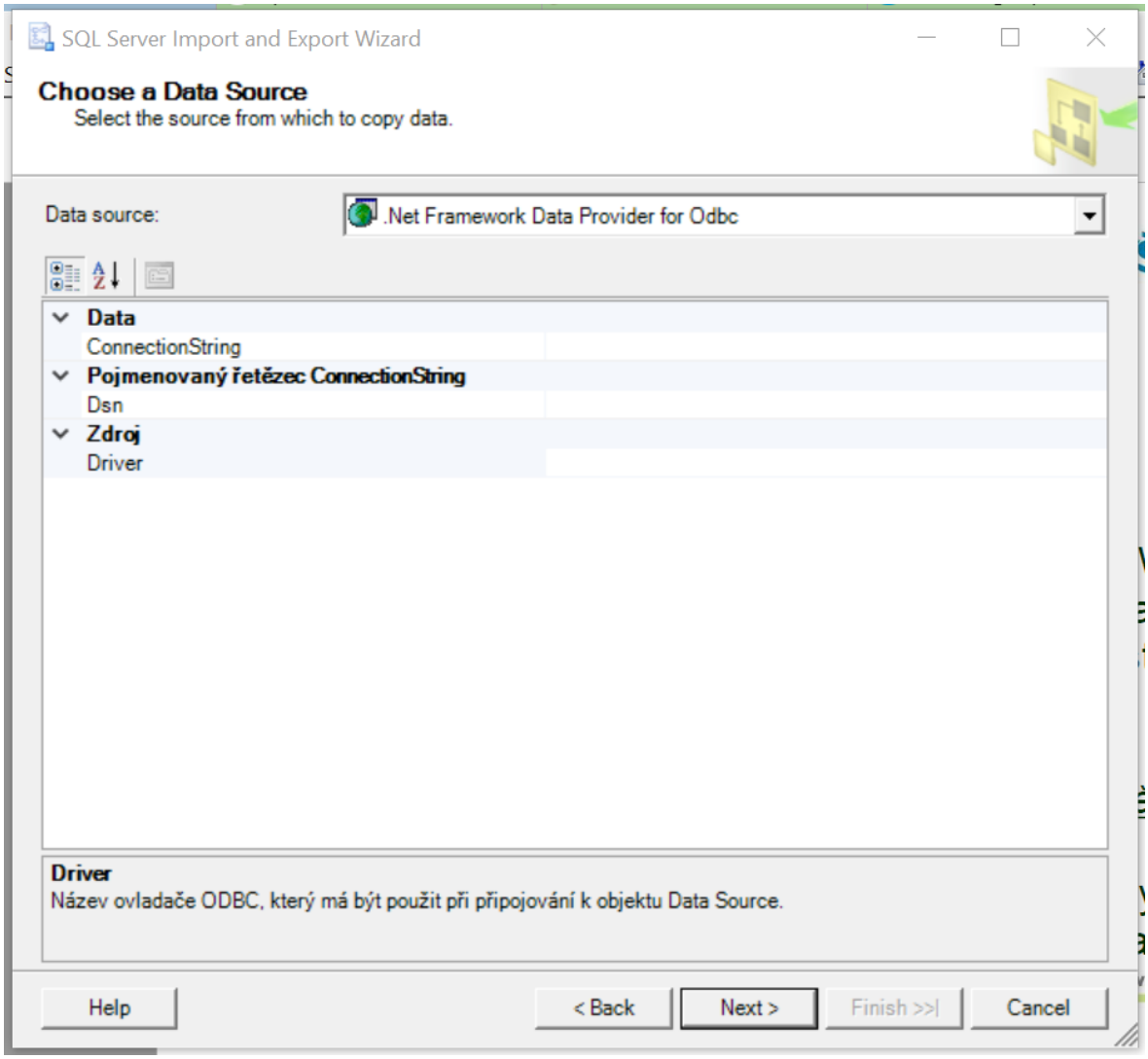
Volba nástrojů závisí na zvoleném operačním systému a SQL serveru. Můžeme také využít schopnosti našich programátorů, kteří ve spolupráci s datovými specialisty můžou pumpy vytvořit v jazycích C++, C#, Java atd.

Hlavní požadavky na výběr ETL nástrojů:

- Schopnost číst/zapisovat/z/do neomezeného počtu typů datových architektur (univerzální)
- Automatizované převzetí a předání metadat současně s daty
- Znalost historie tvorby ETL a jejich možnosti k budoucímu vývoji
- Snadné rozhraní

Nástroje ETL můžeme rozdělit do kategorií podle ceny na:

- Levné řešení: Levné řešení představují nástroje, které jsou dostupné na internetu zdarma. Typicky se jedná o open-source řešení, která se vyznačují požadavky na vysokou technickou zdatnost uživatele.
- Střední kategorie: Do střední kategorie spadají nástroje komerčních společností, jejichž cenová dostupnost umožňuje využití těchto nástrojů téměř u všech typů velikostí projektů. Tyto nástroje jsou více uživatelsky přívětivější. V konečném důsledku je rychlejší napsat např. pumpu v těchto nástrojích, než ji programovat v nízko úrovňových programovacích jazycích.

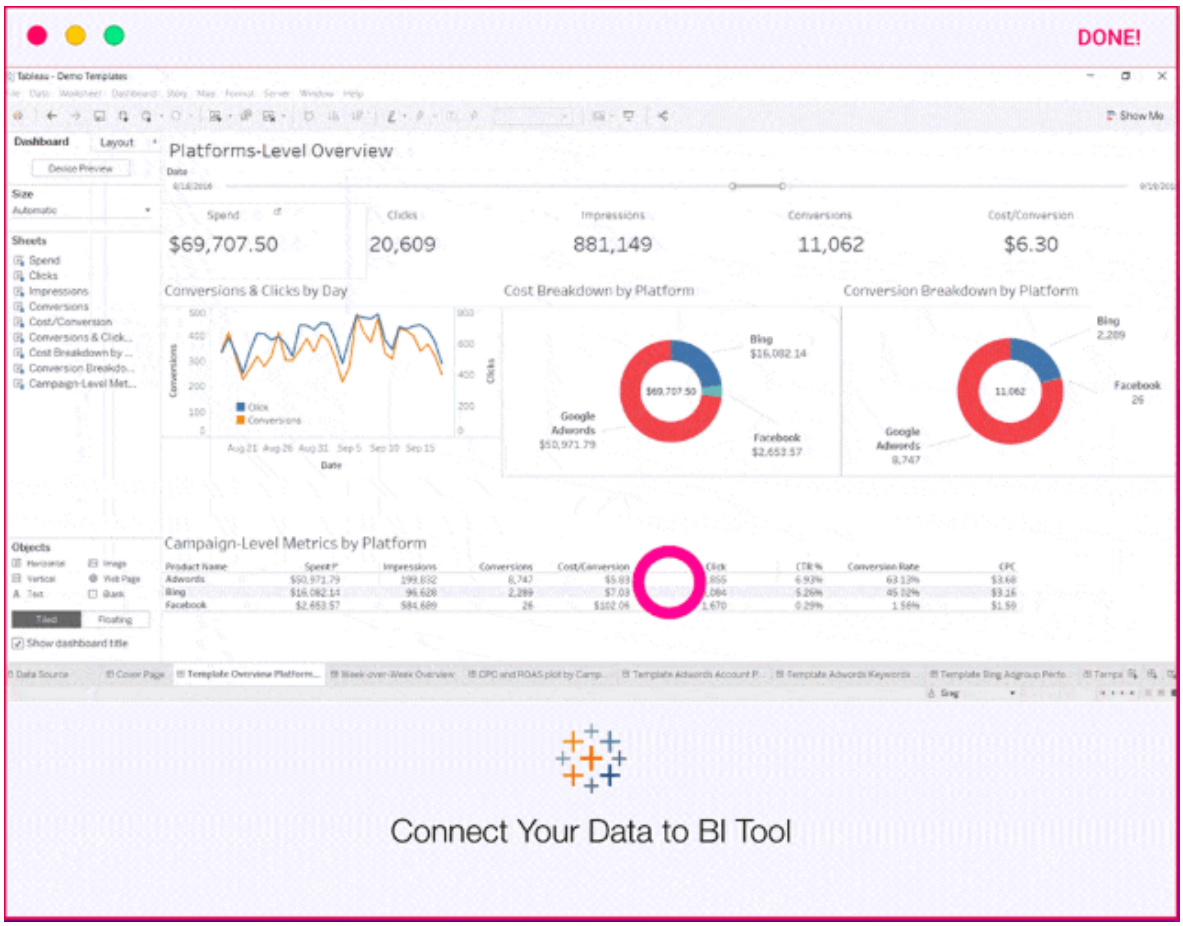


Obr. 5 Jeden z grafických Wizzardů

Výběr nástrojů je v dnešní době opravdu široký:

Improvado

Kompletní ETL nástroj obsahující i prvky základního reportingu. Je zaměřený primárně na marketing. Umožnuje spojení téměř 100 datových typů. Tento nástroj je placený.



Obr. 6 Prostředí Improvado [22]

Skyvia

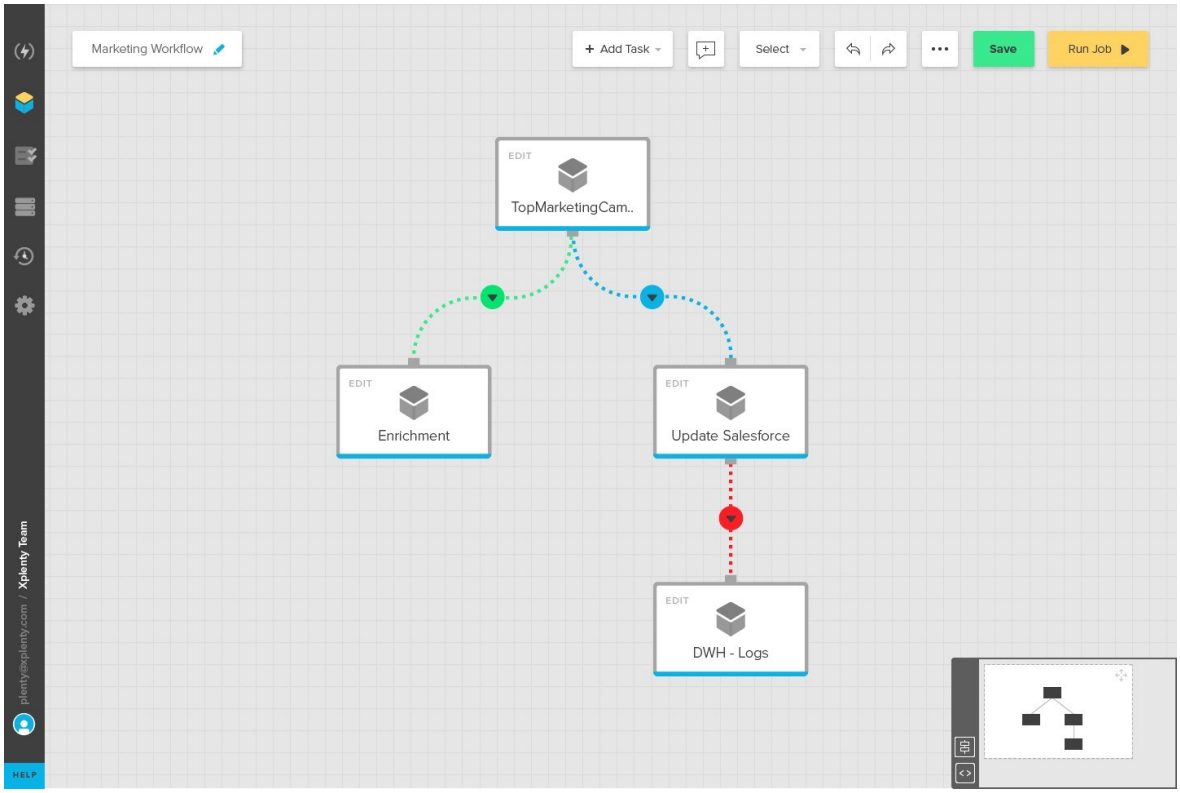
Cloudová platforma pro ETL, která zvládá také zálohování. Skyvia nabízí několik druhů předplatného, některé jsou i zdarma. Disponuje jednoduchým průvodcem při spojování dat.

Hevo

Hevo nabízí ETL jako grafický wizzard. Veškeré procedury je možné sestavit bez znalosti psaní kódu.

Xplenty

Je dalším cloudovým řešením pro ETL. Poskytuje jednoduché grafické rozhraní. Umí extrahovat data z Rest API.



Obr. 7 Prostředí Xplenty [23]

IRI Voracity

Voracity je placený ETL a datamanagementový nástroj v cloudovém prostředí. Zvládá velké množství vstupních datových typů, které dokáže připravit pro Business Intelligence (BI).

IBM – Infosphere Information Server

Vysoce profesionální placený nástroj pro integraci dat, primárně určený pro datové společnosti. Je možné jej rozšiřovat pomocí pluginů. Kompatibilní s ostatním řešením od IBM

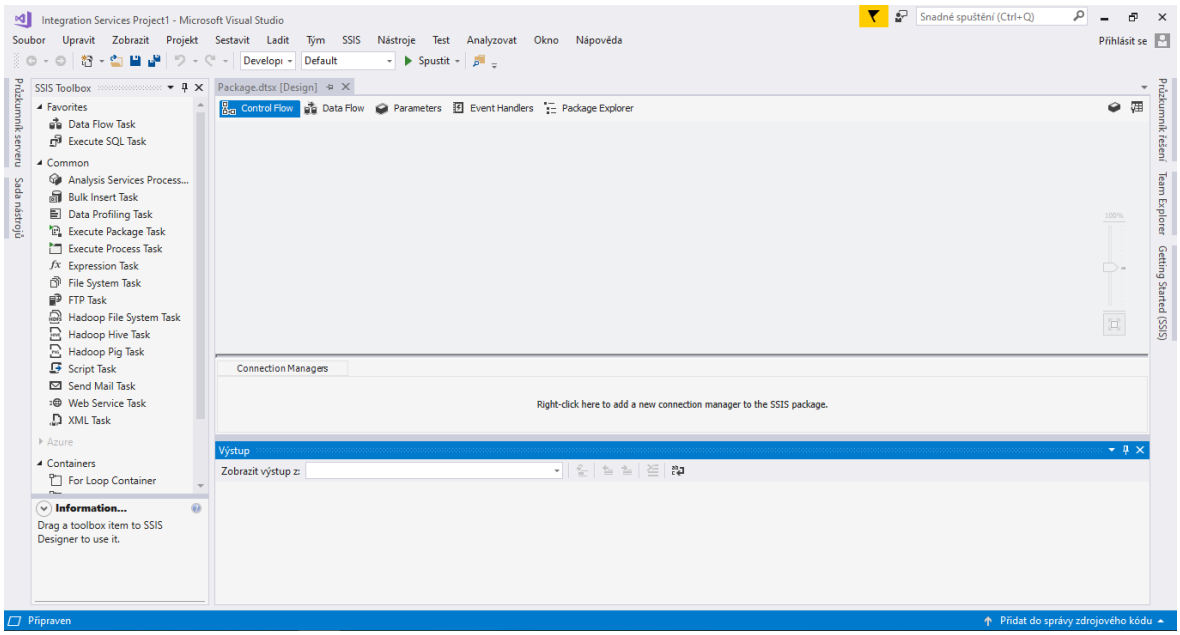
Obr. 8 prostředí IBM – Infosphere information server [22]

Oracle Data Integrator

Grafické prostředí pro vytváření a správu datové integrace. Je vhodný pro velké organizace, které vyžadují častou migraci dat. Automaticky identifikuje vadná data a recykluje je před přesunem do cílové aplikace

Microsoft – SQL Server Integrated Services (SSIS)

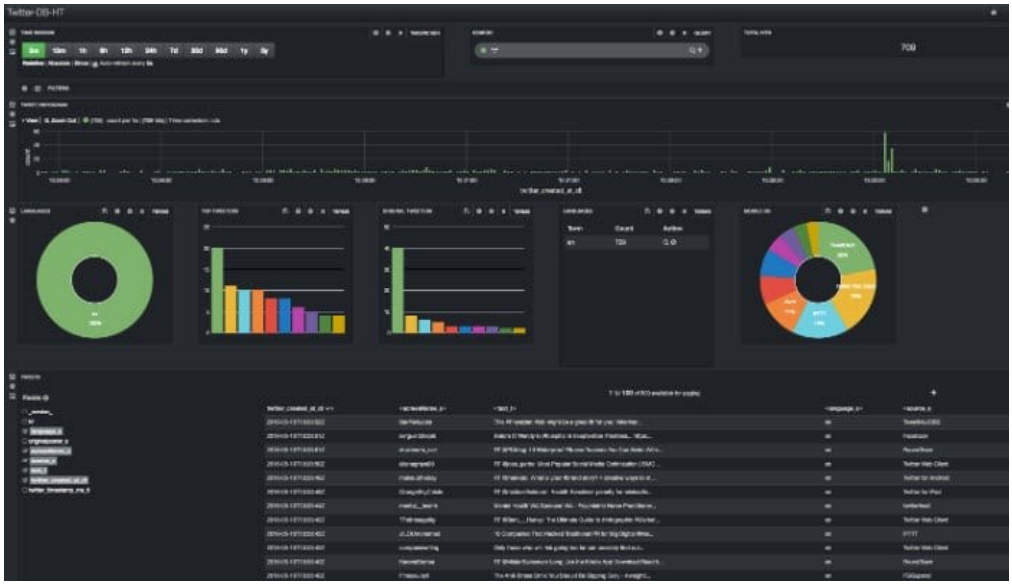
SSIS je produkt společnosti Microsoft a byl vyvinut pro migraci dat. Integrace dat je mnohem rychlejší, protože proces integrace a transformace dat se zpracovává v paměti. Kompatibilní pouze s Microsoft SQL Serverem. Pro komerční využití je licencovaný. Obsahuje Drag and Drop uživatelské rozhraní pro úpravy SSIS balíčků a prostředí pro psaní kódu. SSIS balíček je dostupný z prostředí Microsoft Visual Studio s rozšířením SQL Server Data Tools.



Obr. 9 Prostředí SSIS ve Visual studio (SSDT)

Apache Nifi

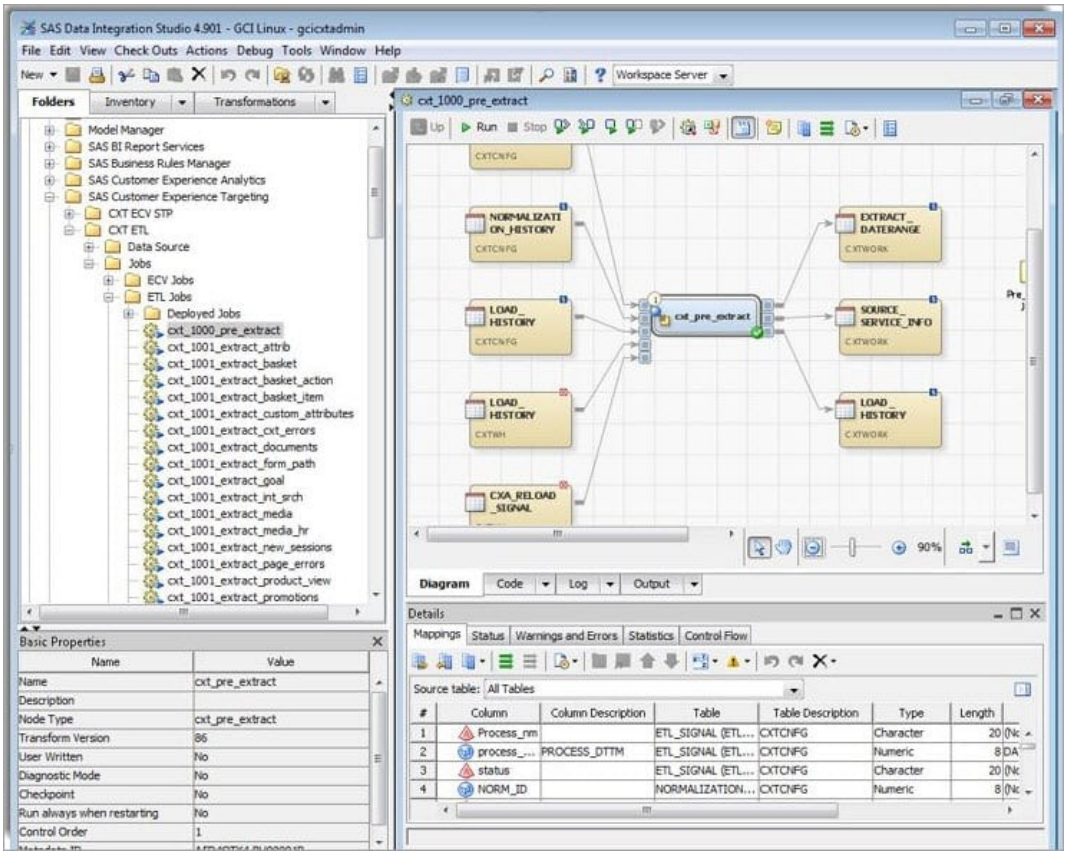
Apache Nifi zjednodušuje tok dat mezi různými systémy pomocí automatizace. Datové toky se skládají z procesů. Uživatel si vytváří vlastní procesy. Tyto toky lze uložit jako šablony a později je lze integrovat do složitějších projektů. Jedná se o open-source.



Obr. 10 Prostředí Apache Nifi [22]

SAS – Data Integration Studio

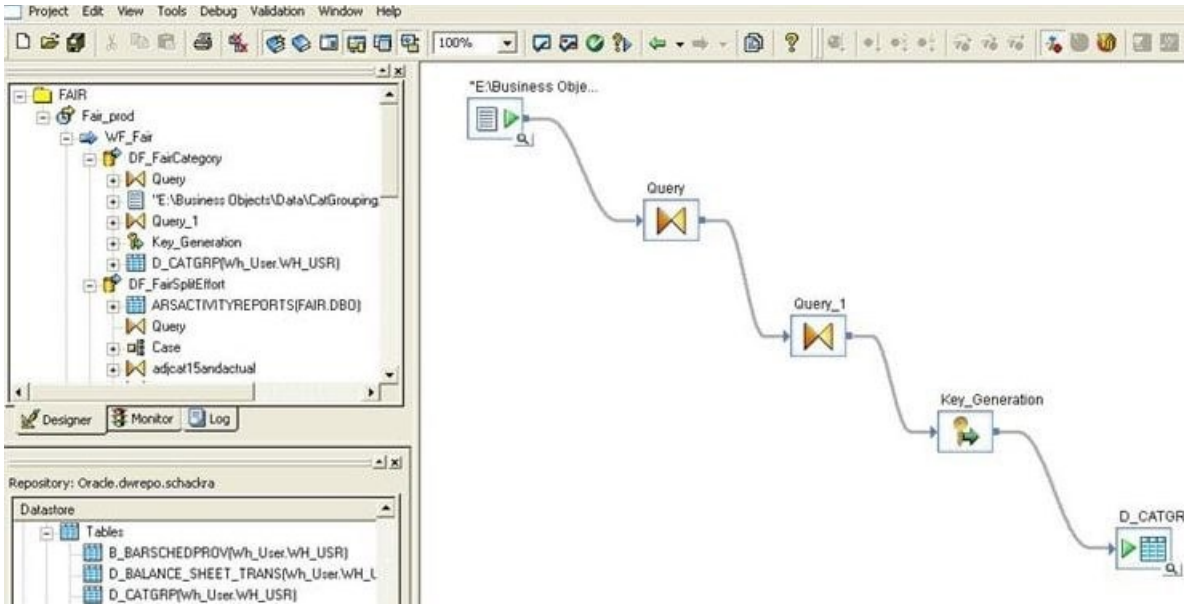
Je grafický nástroj pro vytváření a správu procesů integrace dat. Zvládá zpracovávat velké množství druhů vstupních dat. Snadno se používá díky rozhraní, které je navigováno průvodcem.



Obr. 11 Prostředí SAS – Data Integration Sudito [22]

SAP – BusinessObjects Data Integrator

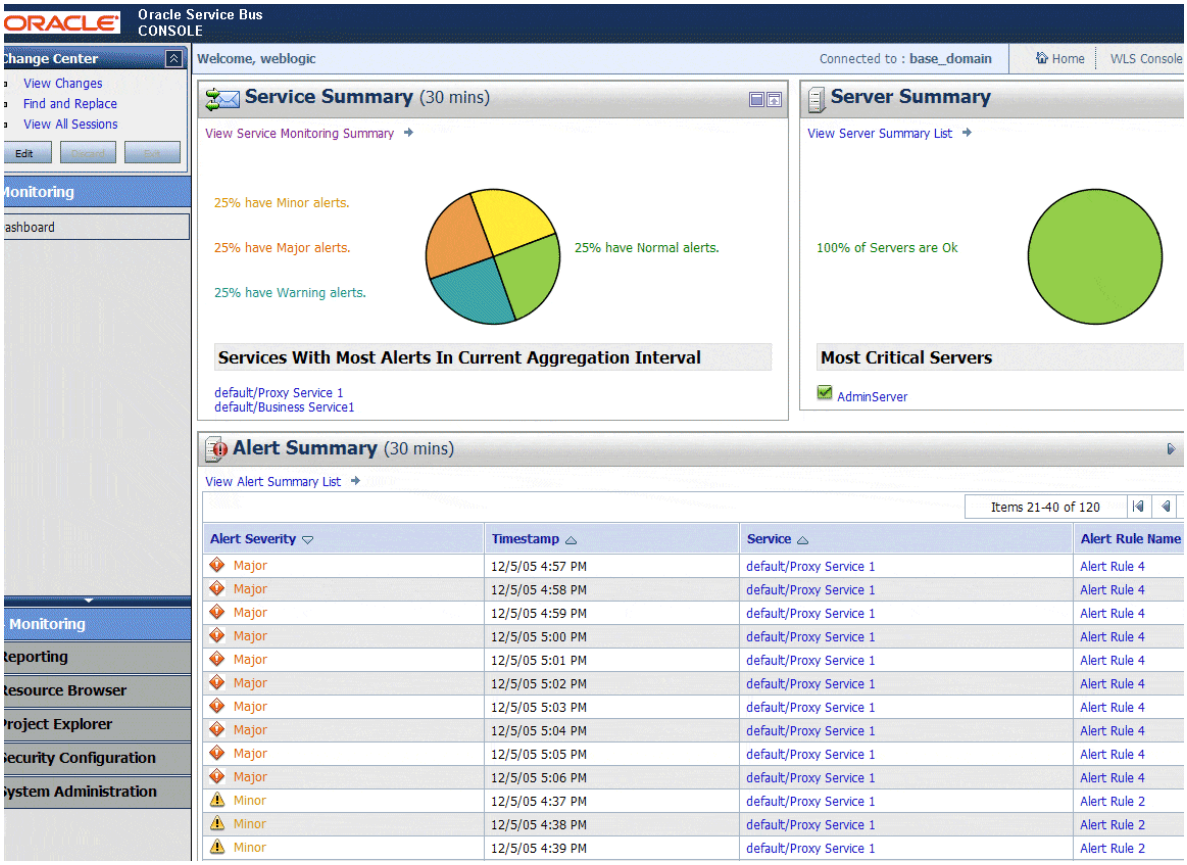
Pomocí SAP BusinessObjects Data Integrator lze data extrahovat z jakéhokoli zdroje a načíst do jakéhokoli datového skladu. Správcem datového integrátoru je webové rozhraní, které umožňuje správu různých úložišť, metadat, webových služeb a serverů úloh. Podporuje platformy Windows, Sun Solaris a Linux.



Obr. 12 Prostředí SAP BusinessObjects Data Integrator [22]

Oracle Warehouse Builder

Grafické prostředí od firmy Oracle. Jeho hlavní funkcí je profilování dat, čištění dat, plně integrované modelování dat a audit dat.



Obr. 13 Prostředí Oracle Warehouse Builder [22]

V konečném důsledku je pro nás výhodné vybírat nástroje jednoho konkrétního výrobce, vzhledem k vysoké kompatibilitě jednotné formě uživatelského rozhraní atd.

2.4.3 Tvorba datových struktur

Struktura datového skladu se dělí na několik dílčích částí a činností. Všechny mají svůj význam.

2.4.3.1 Dočasná úložiště dat (Vrstva L0)

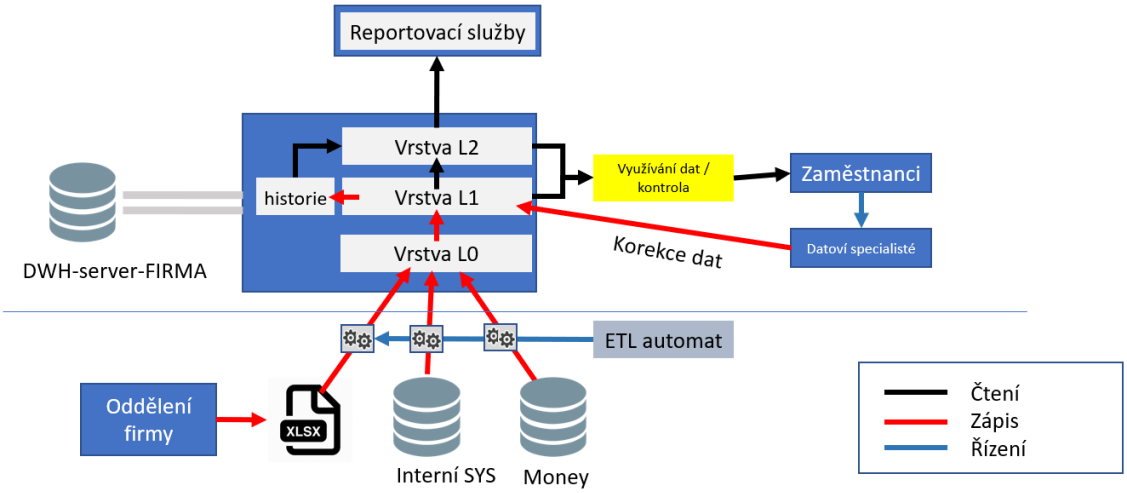
Dočasné úložiště dat je první „nultou“ vrstvou v datovém skladu. Slouží jako dočasné úložiště pro data extrahované z externích produkčních systémů. Data jsou v této vrstvě ještě netransformována, nejsou naprosto detailní, nejsou upravena do formátu datového skladu a nejsou historizována. Obecně platí, že jsou to data otisknutá 1:1 k datům zdrojovým. Před každým novým nalitím dat se vrstva L0 vyčistí. Používá se u zdrojů, které jsou v naprosto jiném než databázovém formátu, např. plain text. Tato vrstva je zpravidla označována jako L0_nazev. [18]

2.4.3.2 Operativní úložiště (Vrstva L1)

Operativní úložiště je další vrstvou v datovém skladu. Může být popsáno jako místo, kde se shromážděná data integrují do jednotné formy před nahráním do datového skladu. Data se zde například převádí do číselníků, agregují se. Z této vrstvy už k datům mohou přistupovat uživatelé a podnikové aplikace. Data v této vrstvě nesmí být měněna. Změny se provádí pomocí takzvaných korekčních tabulek. O každé případné změně a korekci je veden historizační záznam. Vrstvu označujeme jako L1_Nazev. [18]

2.4.3.3 Reportovací vrstva datového skladu (Vrstva L2)

Nejvyšší vrstva v datovém skladu. V této vrstvě se žádná data nenacházejí, obsahuje pouze náhledy (views) na vrstvu předchozí. Jediným možným výstupem z této vrstvy jsou přípravy pro vyšší reportovací nástroje (zpravidla nástroje OLAP) nebo reporting samotný.

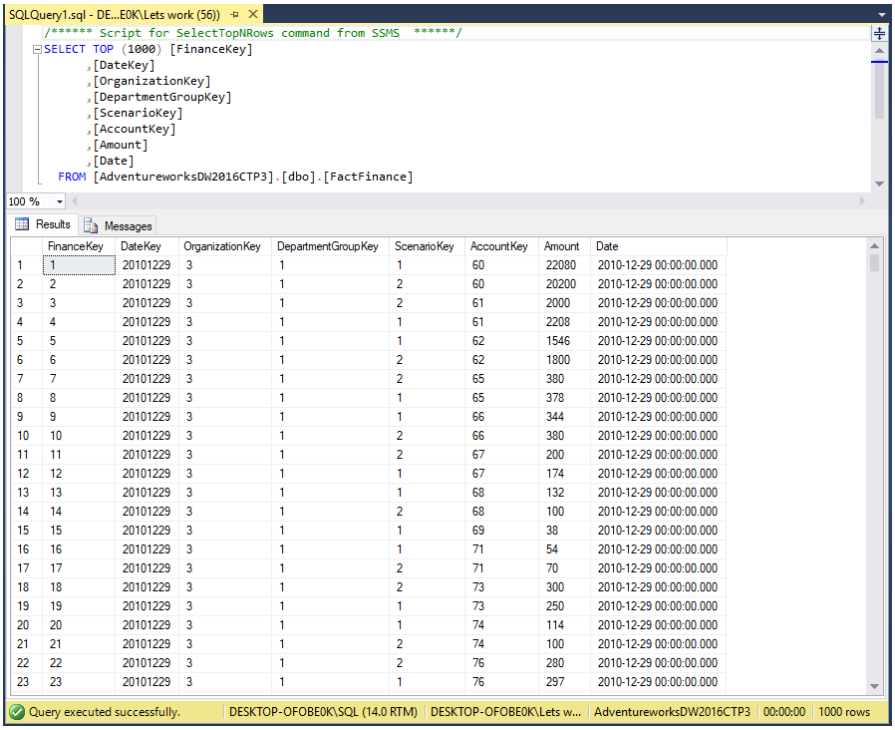


Obr. 14 Příklad struktury DWH [autor]

2.4.3.4 Tabulky dimenzí a faktů

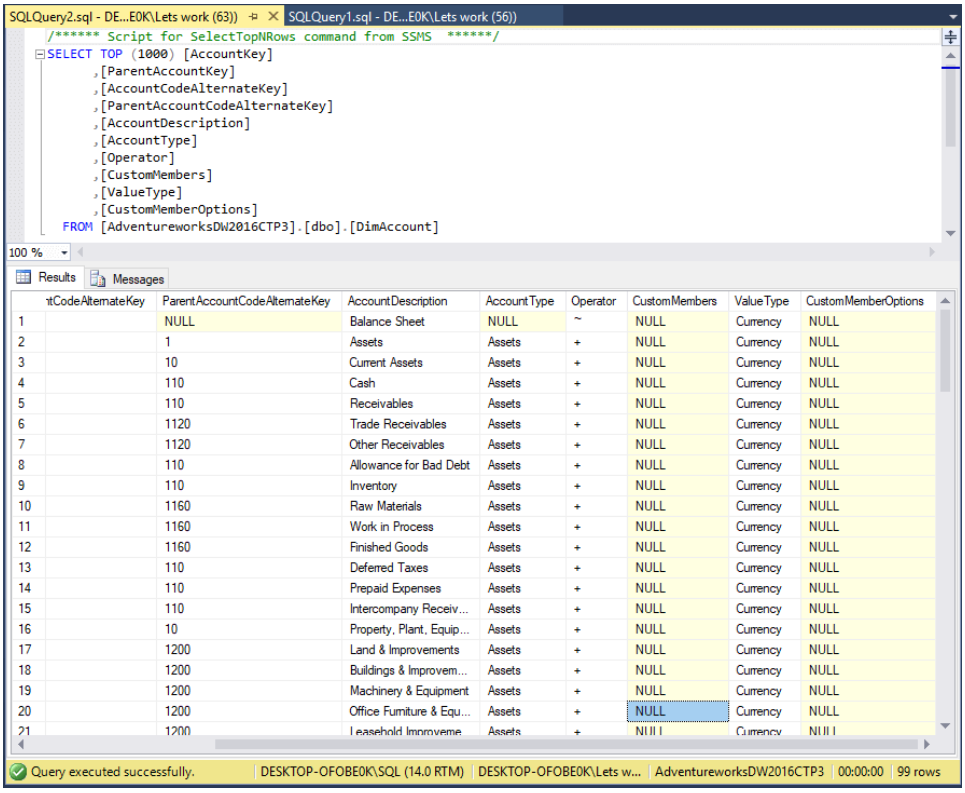
Ve vrstvě L1 jsou uloženy tabulky dvojího druhu. Jedná se o tabulky dimenzí a tabulky faktů. Tyto dva druhy tabulek vznikají proto, aby se data mohla lépe strukturovat a byla přehlednější.

Tabulky faktů: Tabulky faktů obsahují měřitelná data (náklady, výnosy, platby atd.). Záznamy jsou složeny z primárního klíče a dále také cizích klíčů do dimenzionálních tabulek. Nežádoucí je, aby tabulky faktů obsahovali textové pole atd. V názvu se označují jako: FactNázev



Obr. 15 Tabulka faktů [29]

Tabulky dimenzí: Tabulky dimenzí slouží především k tomu, aby data v tabulkách faktů měla význam a byla čitelná. Při zobrazování dat se pomocí JOIN tabulky dimenzí připojují k tabulkám faktů. V názvu se označují jako: DimNázev [29]



Obr. 16 Tabulka dimenzí [29]

2.4.3.5 Historizace

Protože z pohledu auditu nemůžou být v L1 změny, provádějí se formou korekcí. Uživatel tedy nezapisuje změny přímo do vrstvy L1, protože pak by další spuštění ETL pump jeho změny přepsalo zdrojovými daty. V L1 je proto vytvořena tabulka se sufixem korekce, která je historizovaná. Pak uživatel zapíše změny do zdrojového souboru nebo tabulky přímo v L0. Obsah souboru je napumpovaný do L0 korekční tabulky. Pak se provede očistění, transformace a nalijí se data do korekční tabulky v L1. Změny v L1 korekční tabulce se zapíší do historizační tabulky. V L2 je poté vytvořen pohled (view), který načítá data z originální tabulky a navíc data koriguje daty načtenými z korekční (historizační) tabulky.

2.4.3.6 Dokumentace

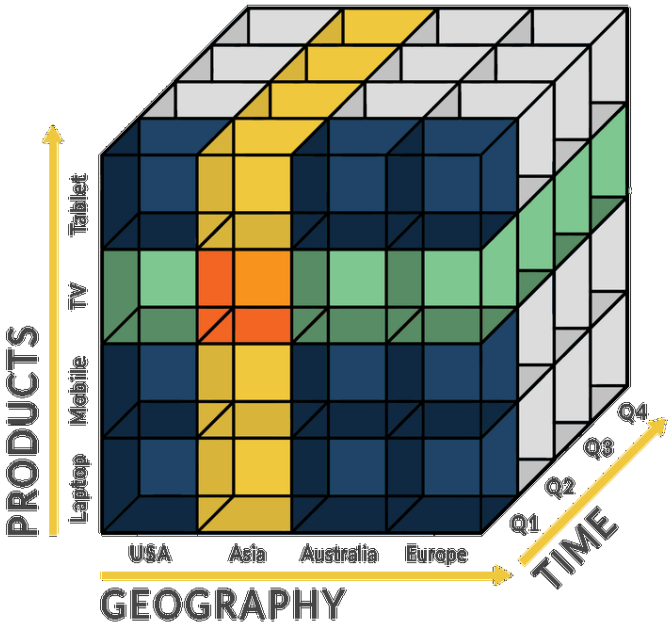
Ke každé struktuře datového skladu je potřeba vést podrobnou dokumentaci. Ta ulehčuje práci při dalším rozvoji v budoucnu. Pokud bude v novém požadavku například připojit několik nových tabulek, uživatel si v dokumentaci dohledá, k čemu je bude napojovat a nemusí studovat celou strukturu.

2.4.4 Reporting

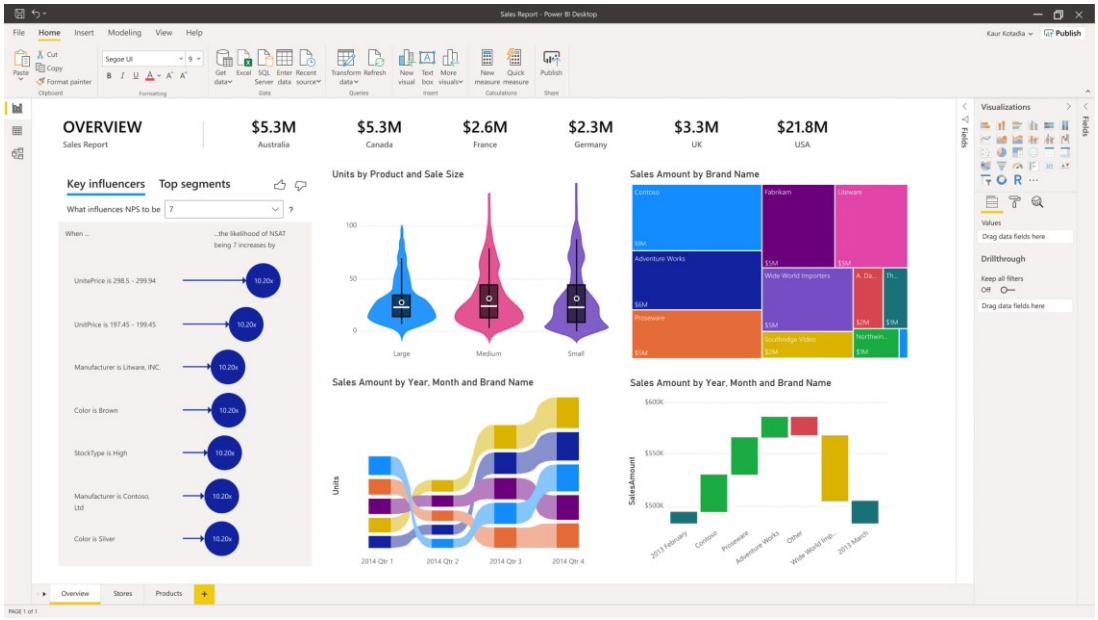
Reporting je jednou z posledních fází při tvorbě datového skladu. Pokud nejsou vysoké požadavky na formát reportů, můžeme využít jednoduchých reportovacích nástrojů, jako jsou například SQL Server Reporting Services (SSRS). Případně lze pro prezentaci dat využít Excel, což je ale krajní a nejméně vhodné řešení. [11]

Obr. 17 Vzorové reporty sestavené v SSRS [24]

V opačném případě přichází ke slovu specializované BI nástroje, které stojí na principu OLAP kostek, které jsou sestavovány pomocí dotazovacího jazyka MDX. Do nich se data z DWH dostávají pomocí svých vlastních pump. OLAP kostky jsou sestavovány za spolupráce datových analytiků a specialistů jednotlivých odvětví, pro které je reporting prováděn. Mezi tyto nástroje patří například IBM Cognos Analytics nebo PowerBI od společnosti Microsoft.[11]



Obr. 18 OLAP Kostka [25]



Obr. 19 Nástroj PowerBI od Microsoftu [26]

II. PRAKTICKÁ ČÁST

3 REALIZACE DWH

Záměrem této práce je seznámit studenty s problematikou datových skladů v rozsahu krátkého kurzu. Kurz navazuje na vzdělání z předmětů týkajících se databází, které studentům poskytly dostatečnou orientaci v relačních databázích, jazyce SQL a znalosti s některými vývojovými prostředími. Cílem je naučit studenta, jak se v datovém skladu orientovat a především, jak DWH vytvořit, napumpovat do něj data, zajistit správnost dat a vytvořit výstupy.

3.1 Výběr prostředí

K vytvoření, údržbě a provozu datového skladu je nutné zvolit softwarové řešení. Při výběru je důležité zohlednit všechny stanovené požadavky. Hlavním požadavkem pro bakalářskou práci je představit téma datové sklady studentům oboru Informační technologie v administrativě. Pro splnění tohoto požadavku bylo použito řešení od společnosti Microsoft, které je studentům známo již z předchozích předmětů. Konkrétně se jedná o:

- Microsoft SQL Server
- Microsoft SQL Server Management Studio (SSMS)
- Microsoft SQL Server Data Tools (SSDT) -> Microsoft Visual studio
- Microsoft SQL Server Integration services (SSIS)
    - instaluje se společně s SQL Server
- Microsoft SQL Server Reporting Services (SSRS)
    - instaluje se společně s SQL Server
- Microsoft Report Builder

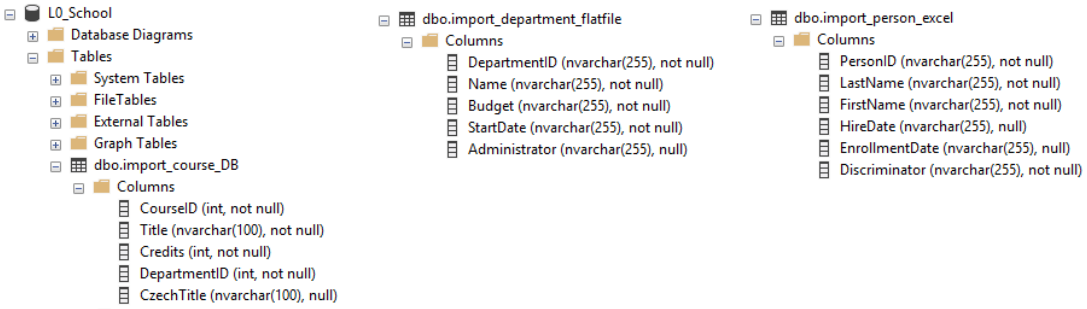
4 VYTVOŘENÍ JEDNOTLIVÝCH VRSTEV (L0, L1, L2)

Protože není stanoveno, jaké výsledné reporty má datový sklad generovat a je připravena vzorová databáze School Sample Database[28], využijeme tvorbu datového skladu shora dolů. Pro vytvoření vrstev bylo využito SSMS. Dostupné zdroje, které budou do databáze pumpovány, jsou:

- Tabulka ve formátu xlsx (xls)
- Flat file (text s oddělovači) csv
- Databáze (import_course_DB)

4.1 Vytvoření vrstvy L0

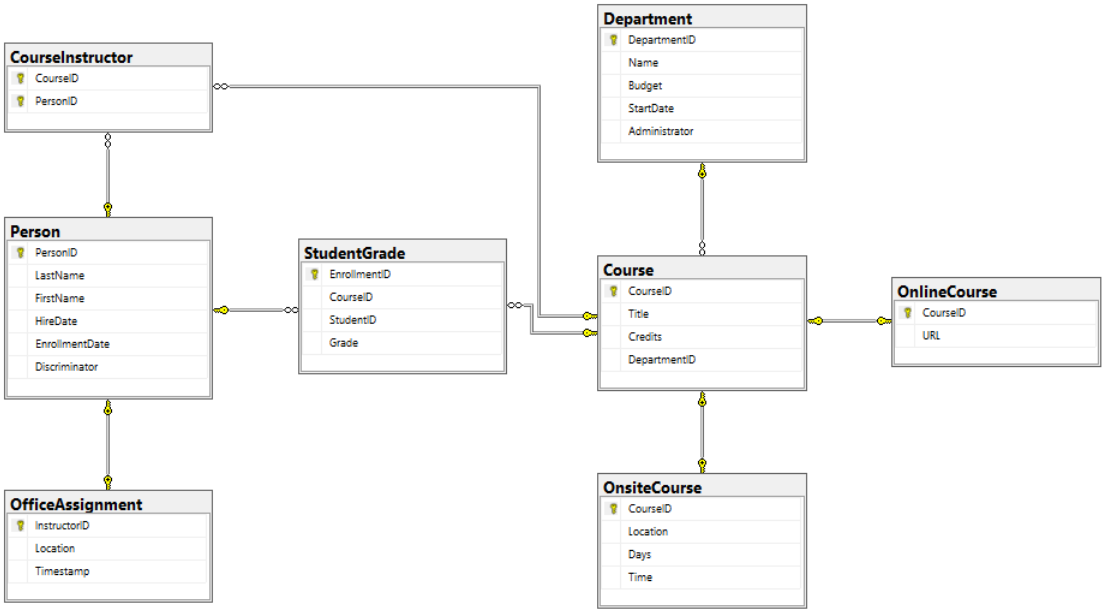
Pro každý zdroj je nutné vytvořit samostatnou tabulku. Aby bylo dodrženo, že data ve zdroji a ve vrstvě L0 jsou totožná (1:1), musí mít tabulky stejné názvy sloupců jako ve zdroji, a především také datové typy. Název databáze (vrstvy) musí obsahovat L0 pro přehlednost.



Obr. 18 Tabulky ve vrstvě L0

4.2 Vytvoření vrstvy L1

Vrstva L1 je uložištěm datového skladu. Pro snadnější představení práce s DWH je použita vzorová databáze School Sample Database, která je do vytvořené vrstvy vložena ručně pomocí SQL dotazu.



Obr. 19 ER Diagram nově vytvořené vrstvy L1

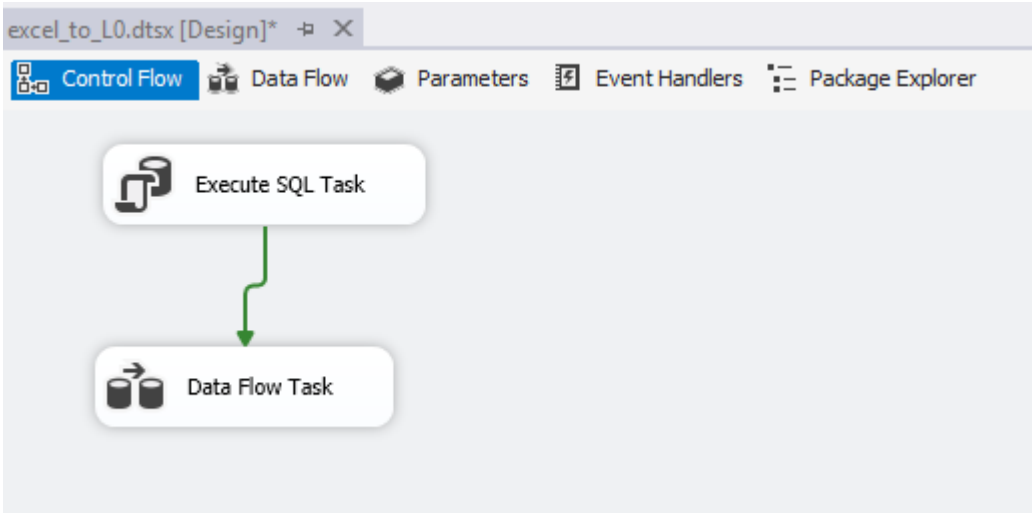
4.3 Vytvoření vrstvy L2

Vrstva L2 (reportovací) je určena především pro tvorbu pohledů (views). Těm se věnuje kapitola 9. Na počátku tvorby DWH tedy zůstává „prázdná“.

5 NAPUMPOVÁNÍ DAT DO VRSTVY L0

Ve vrstvě L0 vznikly 3 tabulky, na které připadají 3 zdroje. K vytvoření pump bylo použito Microsoft Visual Studio (SSDT), ve kterém byl založen nový projekt: Business Intelligence -> Integration Services -> Integration Services Project. Jeden projekt se může skládat z několika balíčků (Packages). Pro balíček, který pumpuje data z externího zdroje do tabulky v L0, jsou v Control Flow použity dva kroky:

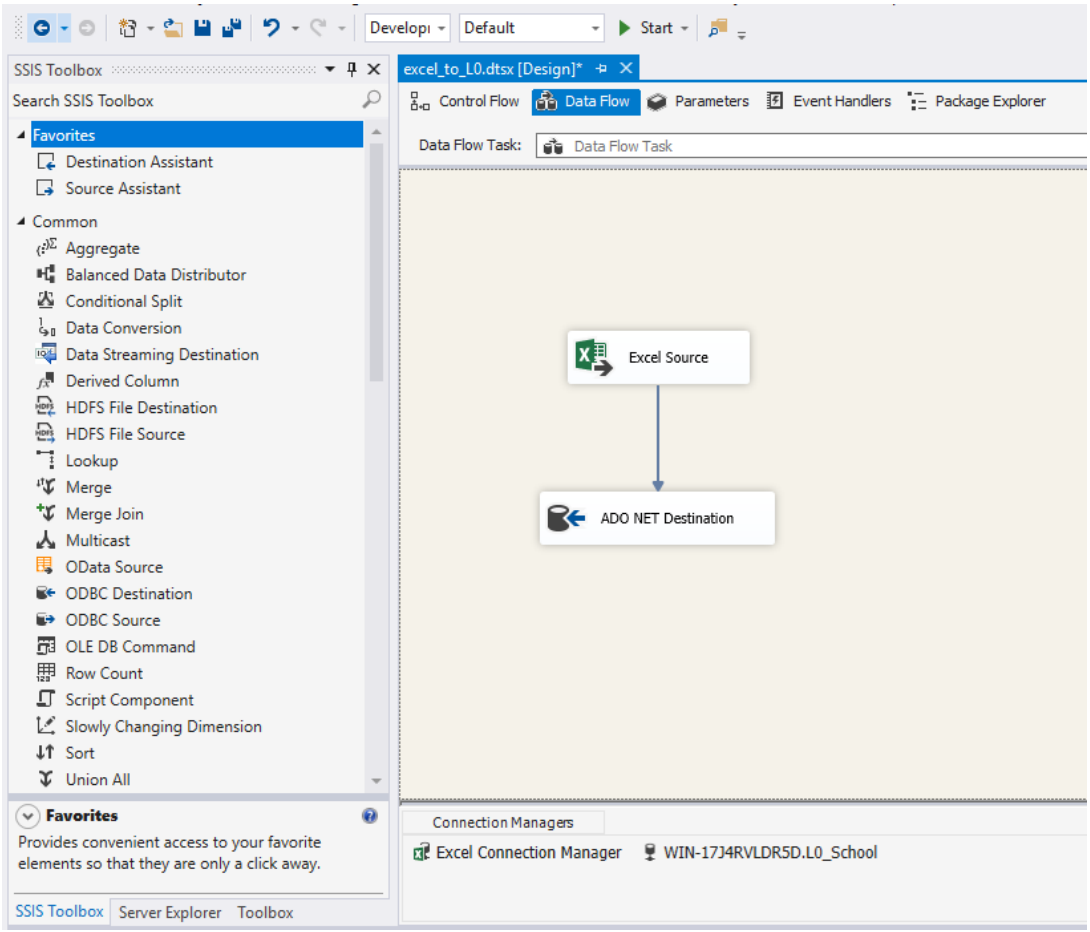
1. Execute SQL Task -> vyprázdní cílovou tabulku před načtením dat pomocí SQL dotazu, který je do něj zadán
2. Data Flow task -> naplní data do tabulky, skládá se z dalších kroků



Obr. 20 Control Flow pro napumpování tabulky v L0

5.1 Data Flow Task pro tabulku v L0

Data flow se skládá z nástrojů z nabídky SSIS. Obsahuje minimálně zdroj dat a cíl dat. Zdroj i cíl mají každý svůj vlastní Connection manager, který určuje, ze které části zdroje bude bráno a kde bude napumpováno (List excelu, Konkrétní tabulka v L0). V nástroji cíle je nutné nastavit mapování sloupců PersonID = PersonID atd. Pumpa se spouští tlačítkem start.

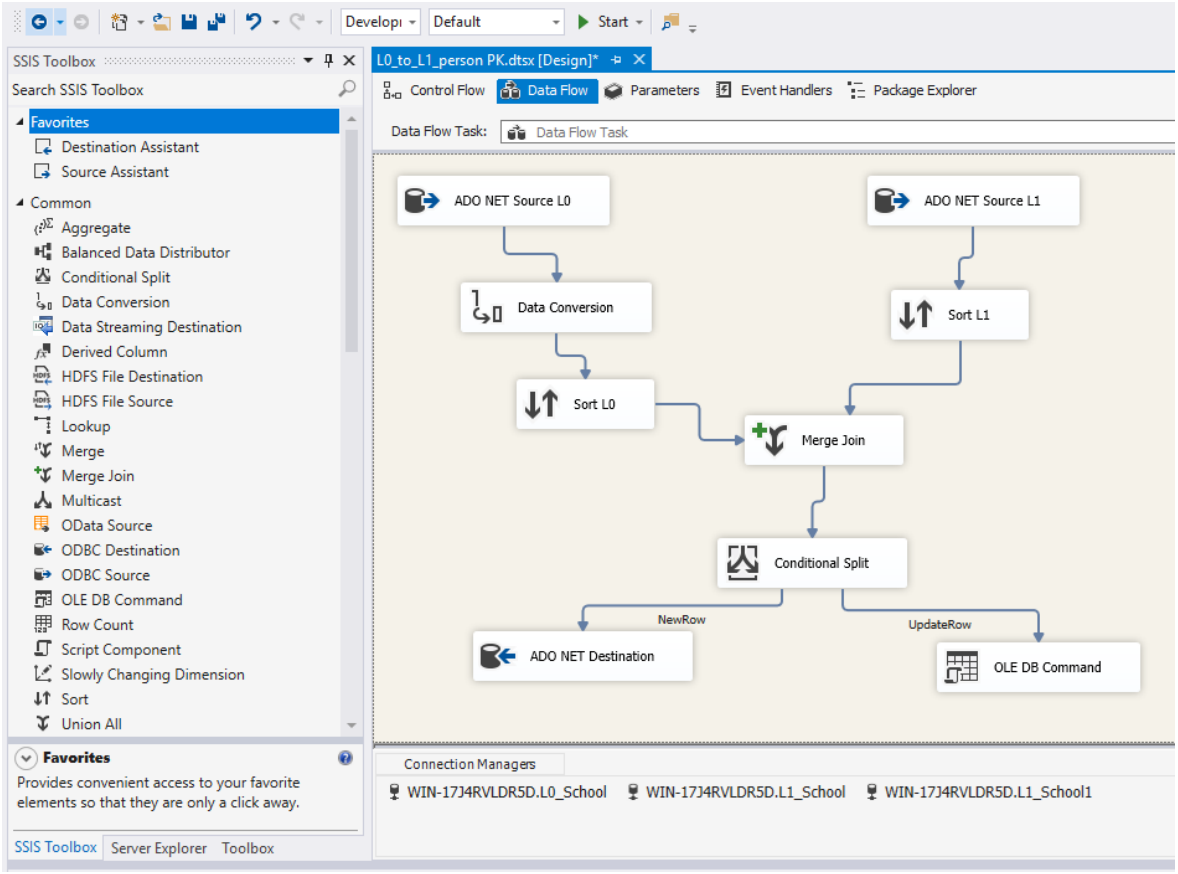


Obr. 21 Data Flow pro napumpování tabulky v L0

Obdobné pumpy jsou vytvořeny také pro zdroje CSV a Databázi.

6 PUMPOVÁNÍ Z VRSTVY L0 DO L1

Po úspěšném napumpování dat do vrstvy L0 je nutné je převést do vrstvy L1. Tento krok je poněkud složitější. V Data Flow je nutné zohlednit až tři možné výstupy: prvním je zápis nového řádku, druhým je aktualizace již existujícího řádku a třetí možností je smazání řádku, pokud se ve zdrojových datech nenachází. Třetí možnost se používá jen velmi zřídka dle konkrétních případů.



Obr. 22 Data Flow pro pumpu z L0 do L1

6.1 Zdroje pro napumpování L1

Na rozdíl od pumpy, která pumpuje data do vrstvy L0, je v Data Flow nutné zadat zrovna dva zdroje. První zdroj je L0, kde jsou data ve stejném formátu jako v původním úložišti dat. Druhým zdrojem jsou data ve vrstvě L1, se kterými jsou v následujících krocích data spojena a dle podmínek roztřízena.

6.1.1 Data Conversion & Sort

Z vrstvy L0 přicházejí do datového skladu data v „surovém“ formátu a je potřeba je přizpůsobit formátu cíle. Prvním krokem je úprava datových typů tak, aby k dalšímu kroku přišla data z obou zdrojů ve stejných datových typech. Pro zdroj L0 je použit Data Conversion, který převádí jednotlivé datové typy sloupců na datové typy použité ve vrstvě L0. Data se ještě dále mohou upravovat – čistit - viz kapitola 7.

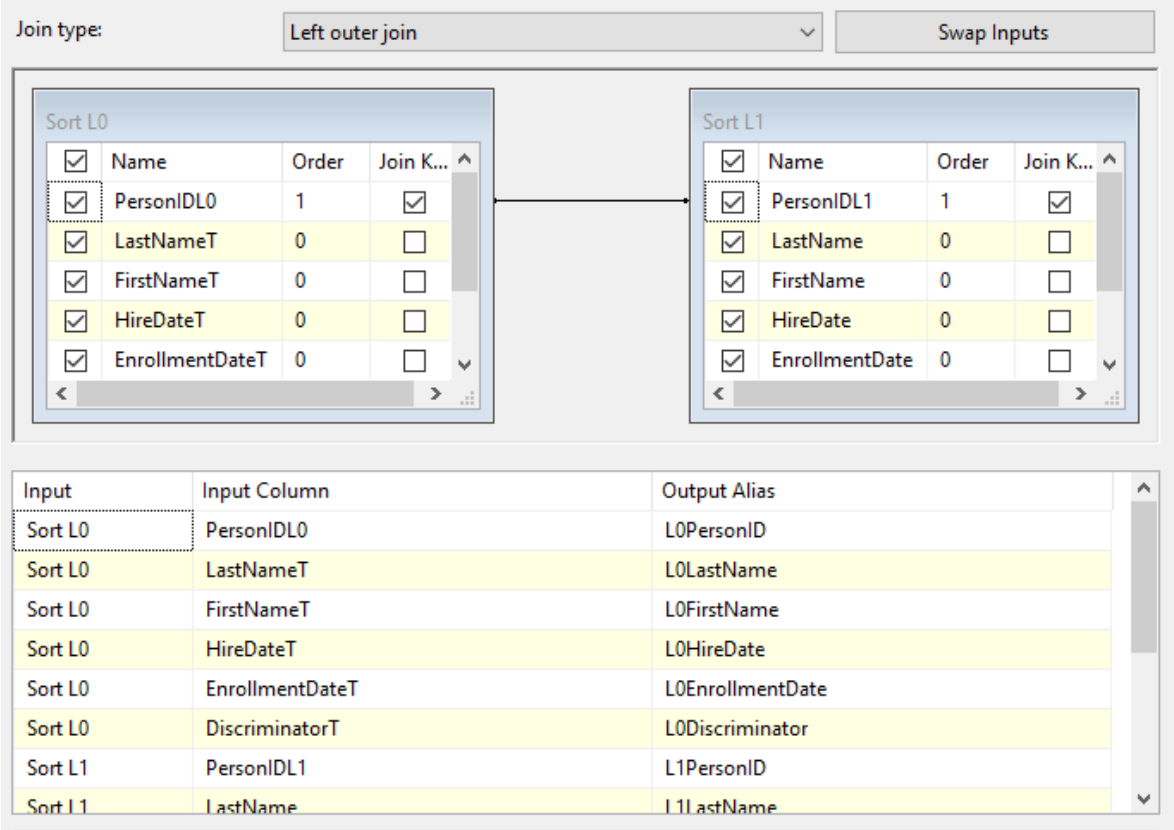
Aby mohlo dojít ke sloučení dat, je nutné, aby byla data seřazena podle identifikátoru. K tomu byla použita funkce Sort. Identifikátorem je zpravidla sloupec, který nese jedinečné ID. Seřadit je nutné oba vstupy.

Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
PersonID	PersonIDT	four-byte signed integer ...				
LastName	LastNameT	Unicode string [DT_WSTR]	50			
FirstName	FirstNameT	Unicode string [DT_WSTR]	50			
HireDate	HireDateT	database timestamp [DT...				
EnrollmentDate	EnrollmentDateT	database timestamp [DT...				
Discriminator	DiscriminatorT	Unicode string [DT_WSTR]	50			

Obr. 23 Konverze datových typů

6.2 Merge Join

Seřazená a transformovaná data se v kroku Merge Join za použití funkce Left outer join spojí pomocí PersonID. V tomto kroku je důležité taky pojmenovat „Output Alias“, aby sloupce z L0 a L1 byly jednoznačně identifikovatelné v dalším kroku.



Obr. 24 Left outer join

6.3 Conditional split a načtení do L1

Sloučená data dále putují k třídění pomocí podmínek. V případě načtení do tabulky Person vznikly dvě podmínky aktualizace záznamu a nový záznam. Smazání v tomto případě není požadováno.

- Aktualizace záznamu:
  - (L0PersonID != L1PersonID) || (L0LastName != L1LastName) || (L0First-Name != L1FirstName) || (L0HireDate != L1HireDate) || (L0EnrollmentDate != L1EnrollmentDate) || (L0Discriminator != L1Discriminator)
- Nový záznam:
  - ISNULL(L1PersonID) && !ISNULL(L0PersonID)

6.4 Merge v SQL proceduře

Pro pumpování dat z L0 do L1 lze také využít proceduru napsanou v SQL, která je odrazem balíčku vytvořeném v grafickém prostředí SSDT. Tato procedura může být k vytvoření

mnohdy jednodušší. Někdy ale může nastat problém s datovými typy. Je důležité určit, jakým způsobem budou data transformovány (vyčištěny).

```
 1 WITH SRC AS (
 2 SELECT CAST(PersonID AS int) AS PersonIDC
 3       ,CAST(LastName AS nvarchar(50)) AS LastNameC
 4       ,CAST(FirstName AS nvarchar(50)) AS FirstNameC
 5       ,CONVERT(datetime,CAST(HireDate AS varchar),104) AS HireDa-
   teC
 6       ,CONVERT(datetime,CAST(EnrollmentDate AS varchar),104) AS
   EnrollmentDateC
 7       ,CAST(Discriminator AS nvarchar(50)) AS DiscriminatorC
 8   FROM [L0_School].[dbo].[import_person_excel]
 9 )
10
11 MERGE L1_School.dbo.[Person] AS DST
12 USING SRC AS SRC
13 ON DST.[PersonID]=SRC.[PersonIDC]
```

Kód 1 Část procedury SYNC_PersonFromExcel

Tato procedura zároveň řeší čištění dat. Nejdůležitějším bodem je vyčištění datumu. To se řídí podle pravidel funkce CONVERT. V parametru funkce se nastavuje vstupní formát datumu, který funkce poté rozpozná.

7 ČIŠTĚNÍ DAT

Některá data se před napumpováním do datového skladu musí transformovat (vyčistit). Ať už se jedná jen o změnu datového typu nebo opravení dat, která nejsou zadána korektně. Nejčastěji se jedná o přebytečné mezery či nežádoucí diakritiku. Čištění se také týká případu, když nějakou slovní hodnotu můžeme nahradit „číselníkem“. Například u uživatelů to můžou být role, které jsou v tabulce zapsány jako text. Tato textová hodnota je nahrazována ID. Samotné role jsou poté definovány v samostatné tabulce.

7.1 Odstranění mezer

K odstranění mezer, například u poštovního směrovacího čísla, není potřeba provádět zvláštní konverzi. Konverze se provede v DataConversion, kde byly nastavovány datové typy. Pokud je požadavek na odstranění mezer z textového řetězce, je využit nástroj Derived Column s nastavenou funkcí REPLACE.

7.1.1 Odstranění mezer v proceduře

Odstranění je procesováno ve stejném kroku SELECT, tak jako je řešen převod mezi datovými typy. Použita je stejná funkce REPLACE.

```
1 SELECT CAST(DepartmentID AS int) AS DepartmentIDC
2       ,CONVERT(nvarchar(50),REPLACE(Name,' ', '')) AS NameC
3       ,CONVERT(int,REPLACE(Budget, ' ', '')) AS BudgetC
4       ,CONVERT(datetime,CAST(StartDate AS varchar),121) AS Start-
  DateC
5       ,CAST(Administrator AS int) AS AdministratorC
6   FROM [L0_School].[dbo].[import_department_flatfile]
```

Kód 2 Select s odstraněním mezer ve sloupcích Name a Budget

7.2 Odstranění diakritiky

Pro odstranění diakritiky za pomocí SSIS ve Visual Studiu není definován žádný odpovídající nástroj. Vyřešení tohoto a složitějších problémů je možné pomoci C# v tool Scripts. V daném prostředí je možné zpracovat data na vstupu a dávat je upravená na výstup. Alternativně je možné i data ukládat – načítat na disk nebo přes síť. Jsou zde pokročilé metody zpracování dat pomocí regulérních výrazů nebo pomocí enkódovacích nástrojů. [30]

7.2.1 Odstranění diakritiky v proceduře

K odstranění diakritiky v proceduře se využívá „přenesení“ do jiné znakové sady. Důležité je také zvolit datový typ varchar. Select v proceduře poté vypadá takto:

```
1 SELECT CourseID AS CourseIDC
2       ,Title AS TitleC
3       ,Credits AS CreditsC
4       ,DepartmentID AS DepartmentIDC
5       ,CAST(CzechTitle as varchar(100)) Collate
  SQL_Latin1_General_CP1253_CI_AI  AS CzechTitleC
6   FROM [L0_School].[dbo].[import_course_DB]
```

Kód 3 Select čistící mezery v proceduře

7.3 Nahrazení hodnoty číselníkem

V tabulce Person_Cist je oproti tabulce Person vytvořen nový sloupec DiscriminatorID naplněn hodnotami „1“ a „2“. Vytvoříme tedy tabulku „Discriminator“ se sloupci

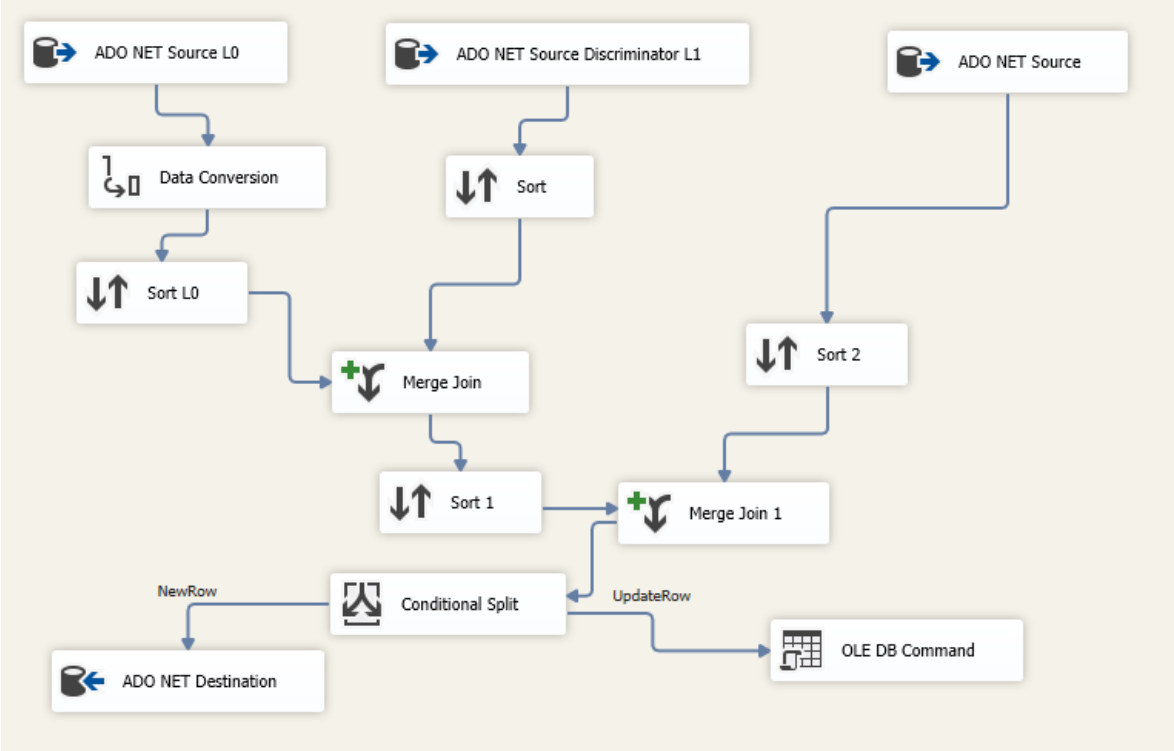
- ID – INT
- Name- nvarchar(50)

A tabulku pomocí SSMS naplníme:

	ID	Name
1	1	Student
2	2	Instructor

Obr. 25 Obsah tabulky Discriminator

Poté je sestaven v Data Flow. Tabulka Discriminator je postavena na stranu zdrojů. V data flow jsou použity dva Merge Joiny



Obr. 26 Datová pumpa s nahrazením hodnoty v sloupci

7.3.1 Nahrazení hodnoty číselníkem v proceduře

V proceduře nahrazení hodnoty probíhá pomocí Left Join s tabulkou Discriminator.

```
1  WITH SRC AS (
2  SELECT CAST(PersonID AS int) AS PersonIDC
3        ,CAST(LastName AS nvarchar(50)) AS LastNameC
4        ,CAST(FirstName AS nvarchar(50)) AS FirstNameC
5        ,CONVERT(datetime,CAST(HireDate AS varchar),104) AS HireDa-
   teC
6        ,CONVERT(datetime,CAST(EnrollmentDate AS varchar),104) AS
   EnrollmentDateC
7           ,Discriminator.ID as DiscriminatorID
8    FROM [L0_School].[dbo].[import_person_excel]
9     LEFT JOIN L1_School.dbo.Discriminator On Discriminator.Name =
   import_person_excel.Discriminator
10 )
```

Kód 4 Left Join v proceduře SYNC_PersonFromExcel_Cist

8 HISTORIZACE, HISTORIZAČNÍ A KOREKČNÍ TABULKY

Změny dat, mimo ty, které provede ETL, jsou v datovém skladu nežádoucí, i přes to jsou někdy nevyhnutelné. Proto je nutné všechny sledovat a evidovat. K evidování slouží historizační tabulky.

8.1 Historizace

Základní historizační tabulka je vytvořena ručně. Je nutné ji rozšířit o další 4 sloupce oproti tabulce, ve které jsou změny sledovány. Jsou to sloupce ID, PlatnostOd, PlatnostDo a UzivatelID. Další změnou je název sloupce, který v tabulce nesl ID. Do názvu sloupce je nutné přidat celý název tabulky. Parametry pro sloupce jsou:

- ID – INT Identity(1,1) NOT NULL
- PlatnostOd - datetime NOT NULL
- PlatnostDo – datetime NOT NULL
- UzivatelID – nvarchar(50) NOT NULL

Historizační tabulka (nahoře) pro tabulku StudentGrade:

	ID	StudentGradeEnrollmentID	CourseID	StudentID	Grade	PlatnostOd	PlatnostDo	UzivatelID
1	1	1	1045	1	5.00	2020-07-16 15:38:34.583	2020-07-16 16:22:56.963	WIN-17J4RVLDR5D\Prokop
2	2	2	1045	2	1.00	2020-07-16 15:38:51.027	3000-01-01 00:00:00.000	WIN-17J4RVLDR5D\Prokop
3	3	1	1045	1	6.00	2020-07-16 16:22:56.963	3000-01-01 00:00:00.000	WIN-17J4RVLDR5D\Prokop

	EnrollmentID	CourseID	StudentID	Grade
1	1	1045	1	6.00
2	2	1045	2	1.00

Obr. 27 Tabulky StudentGrade_Hist a StudentGrade

Touto historizační tabulkou jsou sledovány změny, které jsou provedeny přímo na datech ve vrstvě L1. Správné řešení historizace probíhá až za pomocí korekčních tabulek viz kapitola 8.2

8.1.1 Historizační trigger

Historizační tabulka sama o sobě pozbývá smyslu bez historizačního triggeru. Ten historizační tabulku naplňuje. Trigger běží na sledované tabulce (StudentGrade) a sleduje změny, INSERT, UPDATE, DELETE.

```
 1 CREATE TRIGGER [dbo].[trg_StudentGrade]
 2    ON  [dbo].[StudentGrade]
 3    AFTER INSERT,UPDATE,DELETE
 4 AS
 5 BEGIN
 6 SET NOCOUNT ON;
 7
 8 DECLARE @platnostOd datetime = GETDATE();
 9 DECLARE @platnostDo datetime = dbo.MaxDateTime();
10 -- print 'update historie - ukonceni platnosti puvodne platne po-
   lozky'
11
12
13 UPDATE StudentGrade_HIST SET PlatnostDo=@platnostOd
14
15 FROM StudentGrade_HIST th
16 LEFT JOIN StudentGrade t ON t.EnrollmentID = th.StudentGradeEn-
   rollmentID
17   WHERE
18        th.StudentGradeEnrollmentID IN (
19        SELECT EnrollmentID FROM
20              (
21              SELECT [EnrollmentID],[CourseID],[StudentID],[Grade]
   FROM INSERTED i1
22              EXCEPT
23              SELECT [EnrollmentID],[CourseID],[StudentID],[Grade]
   FROM DELETED  d1 -- zmenene
24
25              UNION
26
27              SELECT [EnrollmentID],[CourseID],[StudentID],[Grade]
   FROM DELETED  d1 -- smazane
28              EXCEPT
29              SELECT [EnrollmentID],[CourseID],[StudentID],[Grade]
   FROM INSERTED i1
30
31              ) chng
32        )
33        AND
34        PlatnostDo = @platnostDo;
35
36
```

```
37 -- print 'insert historie - vlozeni noveho zaznamu (pro in-
   sert/update)'
38 INSERT INTO StudentGrade_HIST ([StudentGradeEnrollmen-
   tID],[CourseID],[StudentID],[Grade],[UzivatelID],[Plat-
   nostOd],[PlatnostDo])
39        SELECT [EnrollmentID],[CourseID],[Studen-
   tID],[Grade],SYSTEM_USER, @platnostOd, @platnostDo FROM
40              (
41              SELECT [EnrollmentID],[CourseID],[StudentID],[Grade]
   FROM INSERTED i1
42              EXCEPT
43              SELECT [EnrollmentID],[CourseID],[StudentID],[Grade]
   FROM DELETED  d1 -- zmenene
44              ) chng
45 END
```

Kód 5 Historizační trigger pro tabulku StudentGrade

V triggeru je použita funkce dbo.MaxDateTime() ta v historizačním záznamu ukazuje čas daleko v budoucnosti a říká nám, že tento záznam je aktuálně platný.

```
46 CREATE FUNCTION [dbo].[MaxDateTime]()
47
48 RETURNS date
49 AS
50 BEGIN
51
52        DECLARE @date date;
53        SET @date = '3000-01-01';
54        RETURN @date;
55 END
```

Kód 6 Funkce MaxDateTime()

8.2 Korekční tabulky

Ke správnému použití historizace je nutné použít korekční tabulku. Vytvoříme tedy tabulku s názvem StudentGrade_Korekce se sloupci: StudentGradeEnrollmentID a Grade. Pro tuto tabulku vytvoříme také tabulku historizační, StudentGrade_Korekce_Hist a vytvoříme historizační trigger, který bude do historizační tabulky zapisovat změny známek.

	EnrollmentID	CourseID	StudentID	Grade
1	1	1045	1	6.00
2	2	1045	2	1.00

	StudentGradeEnrollmentID	Grade
1	1	3.00

	ID	StudentGrade_KorekceStudentGradeEnrollmentID	Grade	PlatnostOd	PlatnostDo	UzivatelID
1	1	1	1.00	2020-07-16 16:00:02.150	2020-07-16 16:00:06.540	WIN-17J4RVLDR5D\Prokop
2	2	1	5.00	2020-07-16 16:00:06.540	2020-07-16 16:00:11.050	WIN-17J4RVLDR5D\Prokop
3	3	1	3.00	2020-07-16 16:00:11.050	3000-01-01 00:00:00.000	WIN-17J4RVLDR5D\Prokop

Obr. 28 Tabulky Student Grade, StudentGrade_Koreke a StudentGrade_Korekce_Hist

Pro zjištění aktuální známky bude sloužit „funkční view“ ve vrstvě L2, které bude mít vstupní parametr datum a bude složeno z tabulek: Student Grade, StudentGrade_Koreke a StudentGrade_Korekce_Hist Pro korekci je vytvořen package, který převádí data z excelovské tabulky přes L0 do L1.

9 VIEWS A REPORTOVÁNÍ

Ve vrstvě L2 jsou uloženy views (pohledy). Jsou to v podstatě připravené selecty. Ty slouží pro zobrazování dat tak, aby uživatel, který má view vidět, nemusel mít žádnou znalost SQL jazyka. Z views čerpají také reportovací nástroje.

9.1 Views

Pro potřebu práce a reporting bylo vytvořena Multistatement table valued function. V této funkci je uložen Select s parametrem @datumplatnosti. Po zavolání této funkce vrací data formátované do tabulky.

Rozdíl mezi klasickým view a multistatement table valued function je především v tom, že funkce má vstupní parametr, který uživatel zadá. [31]

```
 1 CREATE FUNCTION [dbo].[UF_View_Znamky]
 2 (
 3  @datumplatnosti datetime
 4
 5 )
 6 RETURNS TABLE
 7 AS
 8 RETURN
 9 (
10 --     Declare @datumplatnosti datetime = '2020-07-16
   16:00:02.150';
11
12
13 WITH znamky AS
14 (
15
16 SELECT [StudentGradeEnrollmentID]
17       ,[CourseID]
18       ,[StudentID]
19       ,[Grade]
20   FROM [L1_School].[dbo].[StudentGrade_Hist] where @datumplatnosti
   BETWEEN PlatnostOd AND PlatnostDO
21  )
22  ,
23  znamky_korekce AS
24 (
```

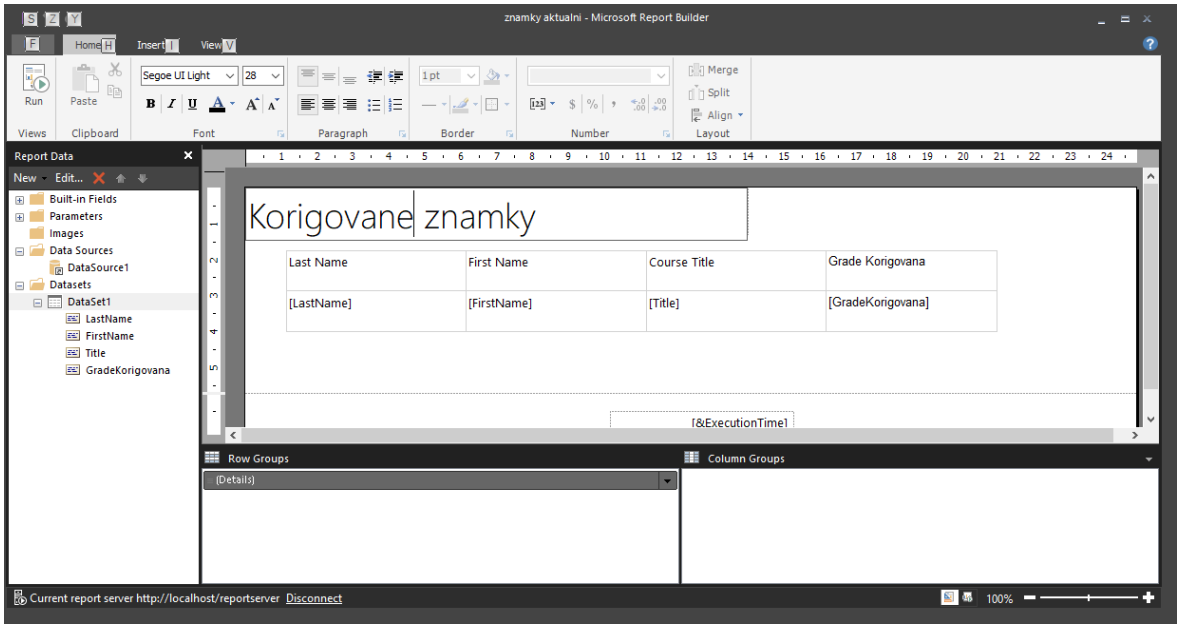
```
25
26 SELECT StudentGrade_KorekceStudentGradeEnrollmentID as [Student-
   GradeEnrollmentID]
27
28       ,[Grade]
29   FROM [L1_School].[dbo].[StudentGrade_Korekce_Hist] where @da-
   tumplatnosti BETWEEN PlatnostOd AND PlatnostDO
30  )
31
32  SELECT
33
34   znamky.StudentGradeEnrollmentID,
35   znamky.CourseID,
36    znamky.StudentID,
37    COALESCE(znamky_korekce.Grade, znamky.Grade) as GradeKorigovana
38  FROM znamky
39  FULL JOIN  znamky_korekce ON znamky.StudentGradeEnrollmentID =
   znamky_korekce.StudentGradeEnrollmentID
40 )
```

Kód 7 Vytvoření Multistatement table valued function UF_View_Znamky

Ve funkčním zobrazení je použita funkce COALESCE. Tato funkce může mít dva a více parametrů. Funkce pak bere jeden parametr po druhém a zobrazí první, který není NULL. Na zobrazeném příkladu funguje takto: Pokud existuje korekční hodnota (Student-Grade_Korekce), tak ji zobrazí. Pokud je korekční hodnota NULL, tak zobrazí hodnotu ze StudentGrade.

9.2 Reportování

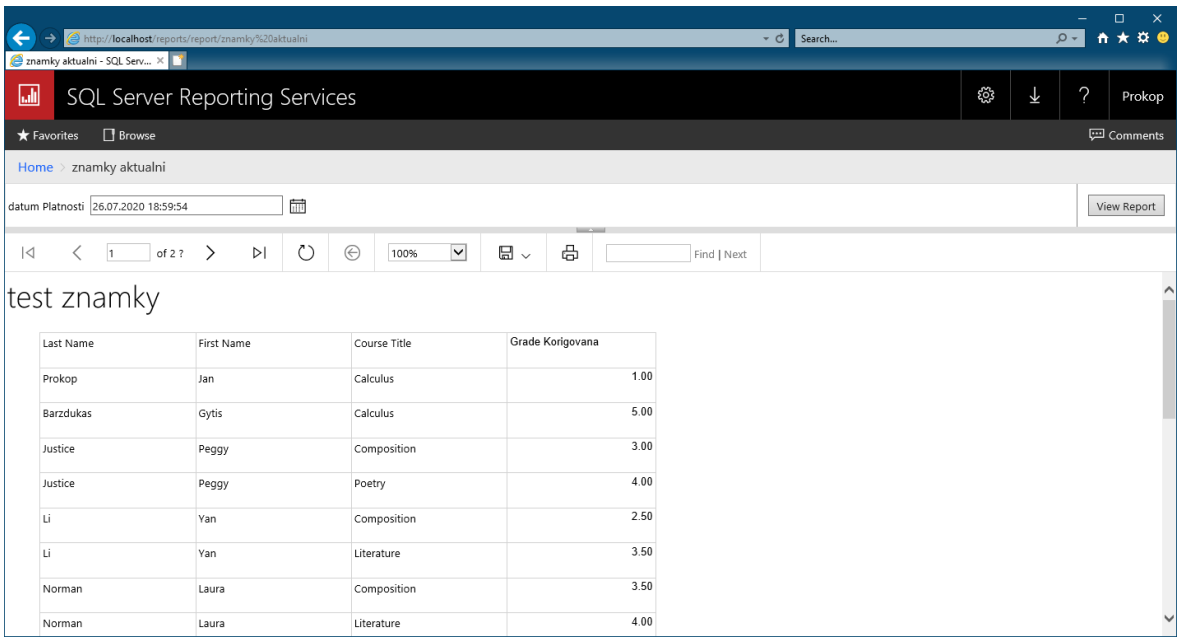
Pro reportování dat z DWH je nutné mít nainstalovaný, nastavený a zapnutý Reporting services. Proces nastavení Report Serveru začíná nastavením Report Server Configuration Manageru. Dále je potřeba se připojit a vytvořit nový report. V novém souboru je nutné nastavit DataSet, ve kterém je zapsán select, který tvoří report.



Obr. 29 Nastavení reportu v Report Builderu

```
1 SELECT LastName, FirstName, course.Title, GradeKorigovana FROM
  [dbo].[UF_View_Znamky] (@datumPlatnosti)
2 left join L1_School.dbo.Person on StudentID = PersonID
3 left join L1_School.dbo.Course on UF_View_Znamky.CourseID =
  Course.CourseID
4 order by PersonID asc
```

Kód 8 Select zobrazující report



Obr. 30 Online report pro UF_View_Znamky

10 VYLEPŠENÍ A USNADNĚNÍ PROVOZU

Pro usnadnění, především pro vyučujícího kurzu byla připravena procedura, která při zachování pravidel vytvoří automaticky historizační trigger pro zadanou tabulku. Jedná se konkrétně o tyto „pravidla“

- Historizační tabulka je rozšířena o 4 sloupce a to:
  - ID (int, identitny 1,1) NOT NULL
  - PlatnostOd (datetime) NOT NULL
  - PlatnostDo (datetime) NOT NULL
  - UzivtelID(nvarchar(50 NOT NUL)
- Název sloupce který nesl ID se musí změnit na NazevtabulkyNazevSloupceID

```
 1 DECLARE @colNames VARCHAR(max) ;
 2 SELECT @colNames = COALESCE(@colNames + ',', '') +
   '['+COLUMN_NAME+']'
 3 FROM INFORMATION_SCHEMA.COLUMNS
 4 WHERE TABLE_NAME = @tablename;
 5
 6 PRINT @colNames;
 7
 8
 9 DECLARE @trgname varchar(100) = 'trg_'+@tablename;
10
11 DECLARE @pkeyWithBracets varchar(200)=
   RTRIM(SUBSTRING(@colnames,0,CHARINDEX(',', @colnames, 0)));
12 DECLARE @pkey varchar(200) =
   REPLACE(REPLACE(@pkeyWithBracets,'[',''),']','');
13
14 DECLARE @colnamesHist varchar(2000) =
   REPLACE(@colnames,'['+@pkey+']','['+@tablename+@pkey+']') ;
15
16 print  @pkeyWithBracets;
17 print @pkey;
18 print '---------------'
```

Kód 9 Část procedury vytvářející historizační triggery

ZÁVĚR

Předkládaná bakalářská práce byla zaměřena na tvorbu výukových materiálů pro kurz Datové sklady, vytvořeného ve formě přednáškové prezentace, úkolů pro cvičení a sadou testových úkolů.

V teoretické části je obsažena literární rešerše, která se zabývá historickým kontextem, charakteristikou pojmů z oblasti databází, které s datovými sklady souvisí. Mezi tyto pojmy patří např. Business Intelligence a OLAP. Dále také charakteristikou pojmů, které přímo souvisí s tvorbou a provozem datových skladů jako je např. ETL, L0 atd. Dále seznamuje čtenáře s několika platformami, které jsou schopné provozovat datový sklad a proces ETL.

Praktická část se zabývá datovými sklady, kde jsou postupně popsány jednotlivé úkoly z běžného provozu datového skladu na platformě od společnosti Microsoft. Tato platforma je studentům dobře známá z předchozí výuky. Konkrétně se jedná o produkty Microsoft SQL Server a Microsoft SQL Managemet Studio a Visual studio. Jsou zde popsány jednotlivé vrstvy skladu, datové pumpy, čištění dat, historizační tabulky, historizační triggery, korekční tabulky, pohledy a základy reportingu.

Přílohou bakalářské práce jsou dvě prezentace. Jedna prezentace je zaměřena na teoretické základy datových skladů a bude sloužit jako podklad k výuce na přednáškách. Druhá prezentace sloužící k podpoře výuky na cvičeních. Na vzorovém příkladu detailně popisuje, jak krok za krokem tvořit datový sklad, datové pumpy atd. a to jak v grafickém prostředí, tak také v prostředí jazyka SQL. Třetí přílohou bakalářské práce je připravena sada 20 testovacích otázek včetně klíče k jejich řešení.

SEZNAM POUŽITÉ LITERATURY

[1] KIMBALL, Ralph a Margy ROSS. *The data warehouse toolkit: the definitive guide to dimensional modeling*. Third edition. Indianapolis, IN: John Wiley & Sons, [2013]. ISBN 1118530802.

[2] Dataversity. *A Brief History of the Data Warehouse* [online]. [cit. 2020-03-14]. Dostupné z: https://www.dataversity.net/brief-history-data-warehouse/#

[3] Systemonline. *Trendy moderních databází* [online]. [cit. 2020-03-14]. Dostupné z: https://www.systemonline.cz/business-intelligence/trendy-modernich-data-bazi.htm

[4] BussinesIT. *Nejžhavější trendy v databázích* [online]. [cit. 2020-03-14]. Dostupné z: http://www.businessit.cz/cz/nejzhavejsi-trendy-databaze-in-memory-cloud.php

[5] Azure Resiliency. *Microsoft Docs* [online]. [cit. 2020-03-14]. Dostupné z: https://azure.microsoft.com/en-us/features/resiliency/

[6] Overview and Usage Scenarios. *Microsoft SQL Docs* [online]. [cit. 2020-03-14]. Dostupné z: https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-oltp/overview-and-usage-scenarios?view=sql-server-ver15

[7] TechDifferences. *Difference Between OLTP and OLAP* [online]. [cit. 2020-03-14]. Dostupné z: https://techdifferences.com/difference-between-oltp-and-olap.html

[8] Oracle Česká Republika. *Co je relační databáze* [online]. [cit. 2020-03-14]. Dostupné z: https://www.oracle.com/cz/database/what-is-a-relational-database/

[9] LABERGE, Robert. *Datové sklady: agilní metody a business intelligence*. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1

[10] RAINARDI, Vincent. *Building a data warehouse with examples in SQL Server*. Berkeley, CA: Apress ; Distributed to the book trade worldwide by Springer-Verlag New York, 2008. ISBN 1590599314.

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

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

[13] LARSON, Brian. *Delivering business intelligence with Microsoft SQL server 2016*. Fourth edition. San Francisco: McGraw-Hill Education, [2017]. ISBN 9781259641480.

[14] Rozdělení vztahu m:n na dva vztahy 1:n. In: *Greendot* [online]. [cit. 2020-03-17]. Dostupné z: http://vyuka.greendot.cz/materialy/material-4.pdf

[15] Databázové modely. *Databáze* [online]. [cit. 2020-03-18]. Dostupné z: http://www.databaze.chytrak.cz/modely.htm

[16] NoSQL Databases Explained. *MongoDB* [online]. [cit. 2020-03-18]. Dostupné z: https://www.mongodb.com/nosql-explained

[17] Introduction to Memory-Optimized Tables. *Microsfot SQL Docs* [online]. [cit. 2020-03-19]. Dostupné z: https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-oltp/introduction-to-memory-optimized-tables?view=sql-server-ver15

[18] POUR, Jan. *Business Intelligence: Jak využít bohatství ve vašich datech*. Praha: Grada, 2004. ISBN 978-802-4766-850.

[19] Oracle Docs. *Extraction in Data Warehouses* [online]. [cit. 2020-03-24]. Dostupné z: https://docs.oracle.com/cd/B19306_01/server.102/b14223/extract.htm

[20] Oracle. *What Is Big Data?* [online]. [cit. 2020-03-31]. Dostupné z: https://www.oracle.com/big-data/guide/what-is-big-data.html?fbclid=IwAR1iyQdQ7wEl78a3oHd-SIF75OOlepD9olTTjqyvI71xOtmSFiXHr-CEfeIw

[21] GEYER, Jakub. *Porovnání objektových a relačních databázových systémů*. České Budějovice, 2012. Bakalářská práce. Jihočeská univerzita v Českých Budějovicích Přírodovědecká fakulta. Vedoucí práce Mgr. Miloš Prokýšek.

[22] Software Testing Help. *15 Best ETL Tools In 2020 (A Complete Updated List)* [online]. [cit. 2020-04-02]. Dostupné z: https://www.softwaretestinghelp.com/best-etl-tools/

[23] Medium. In: *The New Xplenty* [online]. 2018 [cit. 2020-04-02]. Dostupné z: https://miro.medium.com/max/1400/1*UhoiXXKsMrHKW1q4W7eyeg.jpeg

[24] What is SQL Server Reporting Services (SSRS)? In: *Microsoft Docs* [online]. 2019 [cit. 2020-04-02]. Dostupné z: https://docs.microsoft.com/en-us/sql/reporting-services/media/ss-reporting-services-all-together.png?view=sql-server-ver15

[25] Holistics Blog. In: *The Rise and Fall of the OLAP Cube* [online]. 2020 [cit. 2020-04-02]. Dostupné z: https://www.holistics.io/blog/content/images/2020/01/olap-3d-cube.png

[26] Explore PowerBI. In: Microsoft Power BI [online]. [cit. 2020-04-02]. Dostupné z: https://powerbicdn.azureedge.net/cvt-c16b5f70497a5a499474025d24ff5acdb188e36198c6e2cddfd7367e388a90cd/pictures/pages/index/blade2/powerbi.jpg

[27] HAMMERGREN, Thomas C. a Alan R. SIMON. *Data Warehousing for Dummies*. 2nd ed. Indianapolis: Wiley Publishing, 2009. ISBN 978-0-470-40747-9.

[28] School Sample Database. In: *Microsoft Docs* [online]. [cit. 2020-07-16]. Dostupné z: https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-4.0/bb399731(v=vs.100)?redirectedfrom=MSDN

[29] ZEDNÍČEK, Jan. Fakta a dimenze – Tabulky v datovém skladu. *BI Portál* [online]. 23.7.2018 [cit. 2020-07-22]. Dostupné z: https://biportal.cz/fakta-dimenze-tabulky-v-datovem-skladu/

[30] Regulární výrazy .NET. *Microsoft Docs* [online]. [cit. 2020-07-25]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/standard/base-types/regular-expressions

[31] Postupy: Použití uživatelem definovaných funkcí s tabulkovými hodnotami. *Microsoft Docs* [online]. [cit. 2020-07-26]. Dostupné z: https://docs.microsoft.com/cs-cz/dotnet/framework/data/adonet/sql/linq/how-to-use-table-valued-user-defined-functions

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

3NF	Třetí normální forma
API	Application programming interface (Rozhraní pro programování aplikace)
Apod.	A podobně
Atd.	A tak dále
BI	Business Intelligence
DaaS	Database as a Service (databáze jako služba)
DBMS	Database Management System
Dev-Test	Development-Testing (Vývojové a testovací prostředí)
DWH	Data Warehouse (Datový sklad)
ER	Entity Relationship
ERP	Enterprise resource planning (Plánování podnikových zdrojů)
ETL	Extract Transform Load (Extrakce Transformace Načtení)
EUR	Euro
GB	GigaByte
HANA	High-performance ANalitic Appliance software
HW	Hardware
IBM	International Business Machines (Technologická společnost)
MDX	MultiDimensional eXpressions (Dotazovací jazyk)
Např.	Například
OID	Object Identifer (Identifikátor objektu)
OLAP	On-Line Analyst Procesing
OLTP	On-Line Transaction Procesing
SAP	Systeme, Anwendungen, Produkte in der Datenverarbeitung; německá softwarová firma vyvíjející stejnojmenný informační systém
SKK	Slovenská koruna

SQL	Structured Query Language (Strukturovaný dotazovací jazyk)
SSDT	SQL Server Data Tools
SSIS	SQL Server Integrated Services
SSRS	SQL Server Reporting Services
TB	TeraByte
Tj.	To je.
Tzv.	Takzvaný

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM KÓDU

SEZNAM PŘÍLOH

PŘÍLOHA P I: OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje:

- fulltext.pdf – bakalářská práce
- prihlohy.zip
    - Podklady pro cvičení – Kurz datové sklady
    - Podklady pro přednášky – Kurz datové sklady
    - Sada testových úloh – Kurz datové sklady
    - DB-Backup – Zálohy jednotlivých vrstev DWH
    - Solutions – datové pumpy vypracované ve Visual Studiu
    - Ukoly – podpůrné materiály k jednotlivým úkolům