Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Procesy ETL

Bakalářská práce

Autor: Jiří Šimůnek
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Barbora Tesařová, Ph. D.

Hradec Králové listopad 2019

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 12.11.2019 Jiří Šimůnek

Poděkování:

Děkuji vedoucí bakalářské práce Ing. Barboře Tesařové, Ph. D. za metodické vedení práce, podporu a výpomoc.

Anotace

Cílem této bakalářské práce je objasnit problematiku načítání dat do databázového systému pomocí procesu ETL. Práce se zajímá především o nasazení nástrojů vyvinutých společností Microsoft, konkrétně pak Microsoft SQL Server Integration Services a Microsoft SQL Server 2017. Součástí práce ovšem je i seznámení s problematikou z všeobecného hlediska, běžnými nedostatky a komplikacemi při implementaci a nasazení, různými prostředími, kde lze ETL uplatnit, a tomu příslušnými nástroji.

Nejprve se práce zaměří na obecnou problematiku, seznámí čtenáře s významem databází a v tom jak rozdílná jsou prostředí, kde lze ETL aplikovat. Dále osvětlí průběh importu dat včetně komplikací, které je třeba mít na paměti v jednotlivých krocích. Poté představí výčet několika různých dostupných řešení a porovná je s SSIS. Nakonec demonstruje konkrétní použití SSIS v modelovém prostředí, kde se zaměří na přenos dat mezi dvěma databázemi při migraci do nové firemní aplikace.

Klíčová slova: DBMS, RDBMS, ETL, SQL, MS SQL Server, MS SSIS, databáze, migrace dat

Annotation

ETL Processes

This thesis is aimed to clarify the issue of loading data into a database system using the ETL process. It is mainly interested in deployment of tools developed by Microsoft, specifically Microsoft SQL Server Integration Services and Microsoft SQL Server 2017. Part of the work, however, is familiarization with issues from a general perspective, common weaknesses and complications in implementation and deployment, various environments where to apply the ETL with the relevant tools.

First, the work will focus on general issues, acquaint readers with the importance of databases and how different are the environments where ETL can be applied. It also explains the process of data import, including the complications that need to be kept in mind in each step. It then presents a list of several different solutions available and compares them with SSIS. Finally, it demonstrates the specific use of SSIS in a model environment where it focuses on data transfer between two databases when migrating to a new business application.

Keywords: DBMS, RDBMS, ETL, SQL, MS SQL Server, MS SSIS, database, data migration

Obsah

Seznam obrázků

Seznam grafů

Seznam tabulek

1 Úvod

Počítačový software a hardware se již několikátým desetiletím těší velkému rozkvětu. V posledních skoro třiceti letech tomu největší měrou přispívá světová síť Internet. Právě počítačové sítě jsou klíčové pro použití podnikových aplikací, ve kterých větší množství koncových uživatelů přes různá počítačová zařízení přistupují k jednotným datům, skrze aplikaci k jednomu databázovému systému.

Se stárnutím softwaru se podniky čas od času dostanou do bodu, kdy je za potřebí přechod na nový program. V podniku též může dojít k nasazení nové podnikové aplikace kvůli nějaké funkčnosti, kterou stávající neumí zastat. V obou těchto případech firmy nerady přichází o svá, často i léta sbíraná, data. Existují také aplikace, které se instalují cíleně do prostředí, kde už běží jiný software, pracují s jeho daty a fungují jako forma rozšíření doplňující jiný vhled do datové struktury.

Všechny tyto situace spojuje jedno, je při nich třeba řešit přenos dat. Data jsou často nekompatibilní kvůli různým strukturám, či zcela rozdílným databázovým systémům. V takových situacích přichází ke slovu procesy Extract Transform Load (dále jen ETL), které mají za cíl vyřešit mapování jedné struktury do jiné a data správně načíst, převést a následně uložit.

2 Cíl práce

Cílem této práce je nejprve se obecně seznámit s tím, jak ETL funguje a kdy je vhodné jej použít, s nástroji pro něj určenými, objasnit si v jakých prostředích se používají, čím se různá řešení liší a kdy je vhodné o nasazení takového nástroje uvažovat. Na konkrétní implementaci s použitím SQL Serveru a SQL Server Integration Services (dále jen SSIS) pak bude předvedeno, jak může v praxi vypadat použití takového nástroje.

Implementace bude mít za cíl vytvořit jednoduchou databázi, která vyjde z již existující aplikace a její databáze, poté provést přenos dat z jedné databáze do druhé a demonstrovat tím, jak by mohl vypadat přechod z jedné aplikace na jinou.

Při takové migraci je pro zajištění úspěšného přenosu zapotřebí vyřešit množství odlišností, které pramení z různých formátů dat a obecně jiných struktur. Pro opravení těchto nuancí bude využito SSIS balíčku. Ten by měl načíst data z předem definované databáze a pomocí potřebných transformací je naplnit do nově vytvořené.

Výsledkem měl být SSIS balíček nasazený na SQL Serveru a spouštěný v pravidelných intervalech tak, aby zbytečně nezatěžoval serverovou infrastrukturu.

3 Metodika zpracování

Cílem migrace dat by měl být přechod na novou, čistší a výrazně jednodušší firemní aplikaci, jež by měla řešit výpočet plánu výrobních kapacit. Návrh nové databáze tedy nachází inspiraci ve starší aplikaci, která již několik let běží na interní síti nejmenovaného výrobního podniku. Tato starší aplikace měla sloužit jako koncept pro vývoj firemního softwaru pro plánování zakázkové výroby na architektuře client-server. Během vývoje již byla aplikace testována na "ostrých" datech výrobního podniku, který si spoustu funkcí přál doplnit, čímž prototyp aplikace výrazně přerostl původní představy.

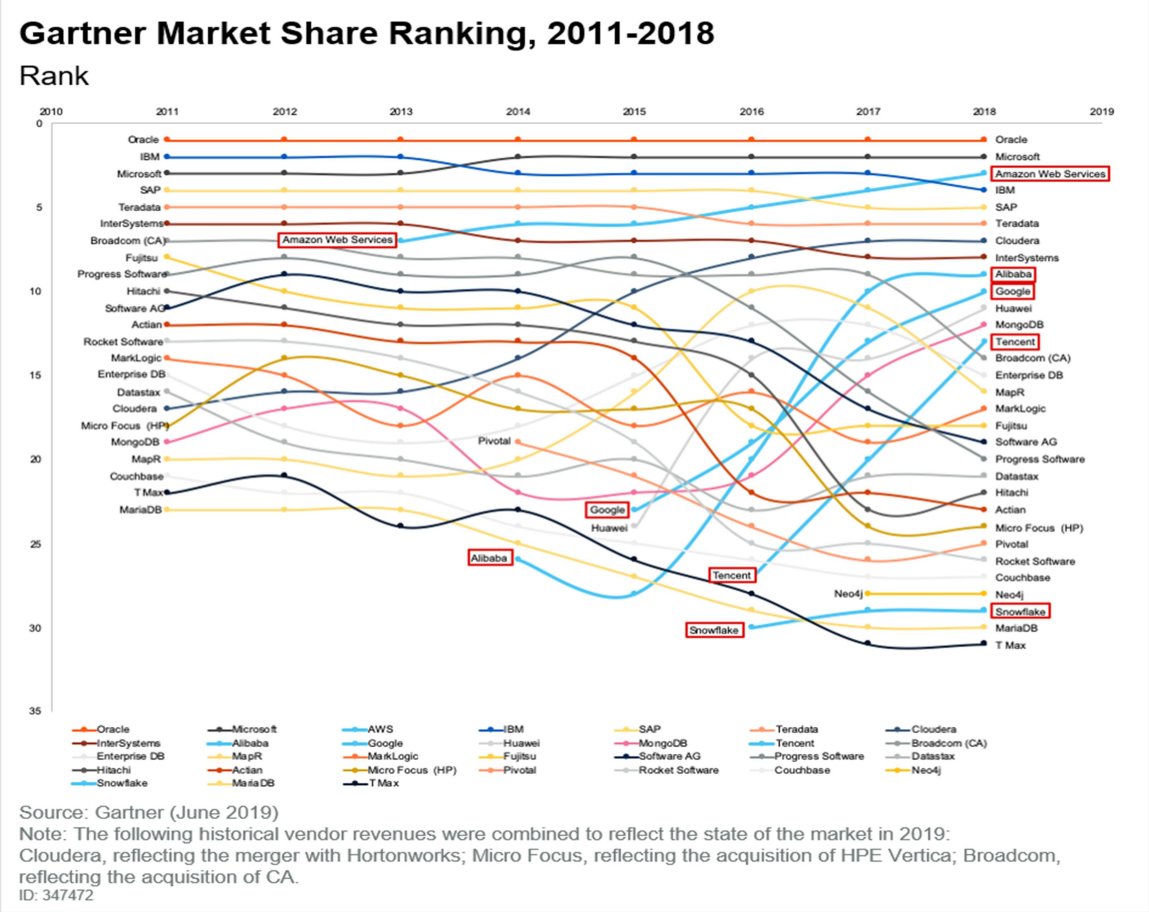
Nový program má za úkol oprostit se od konkrétních potřeb firmy, jež testovala prototyp, navrhnout jádro databáze, tedy ponechat pouze část pro výpočet plánu, a zajistit přenos dat do nové struktury pomocí SSIS. Pro zajištění plynulého přechodu a možnosti provozovat po nějakou dobu obě aplikace souběžně bude ETL spouštěno v pravidelných intervalech a načítání dat do nové databáze bude probíhat vždy v nočních hodinách, aby větší zátěž serveru při přenosech nemohla omezit plánovače při práci.

4 ETL procesy

Z hlediska toho, jakými funkcemi nástroje pro ETL běžně disponují, by se nabízelo vyvozovat, že ETL představuje obecný termín označující přenosy dat vyžadující nějakou formu transformace, neboť zdrojová i cílová úložiště často mohou být krom databází i různá další. Tak tomu ovšem není. ETL je bráno především jako databázové funkce spojené s migrací dat.

4.1 Databáze

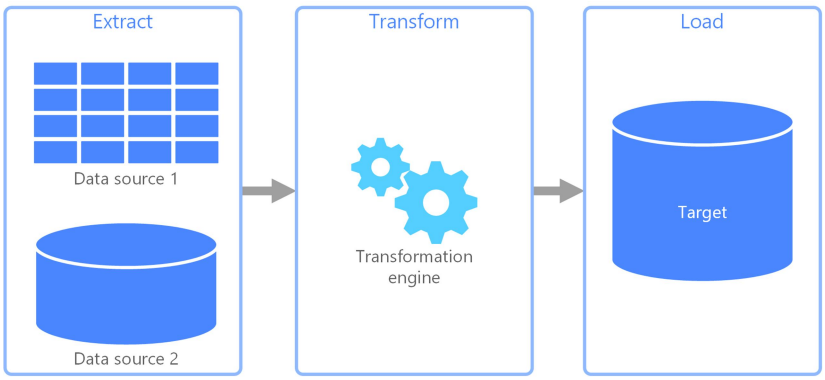
Většina moderního softwaru se jen stěží obejde bez potřeby zaznamenávat data do organizované datové struktury. K těmto účelům slouží databázové systémy existující dnes již ve spoustě různých podob. Běžně se tyto systémy označují česky systém řízení báze dat (SŘBD) nebo anglicky database management system (DBMS), česká varianta je spíše historického charakteru a v praxi se povětšinou neobjevuje. Na základě potřeb neustále se vyvíjejícího trhu vzniká neustále čím dál více jak implementačně, koncepčně, tak i licenčně rozdílných řešení. Relační databáze jako je např. Microsoft SQL Server stále drží majoritní podíl trhu, budoucnost však míří do virtuálního světa cloudových technologií, jak naznačuje výzkum korporace Gartner. [1] Za rok 2018 zaznamenal trh DBMS největší procentuální růst za poslední dekádu, v konkrétních číslech je řeč o nárůstu 18,4%, z čehož je připisováno celých 75% dvěma nejrychleji rostoucím společnostem, a sice Amazonu a Microsoftu, z nichž Amazon poskytuje pouze cloudově hostované databáze a v případě Microsoftu se jejich podíl na růstu blíží ke 100%. Graf ukazuje podíl jednotlivých firem na trhu během let 2011 až 2018, červeně o značené jsou pak společnosti s nejdynamičtějším růstem.



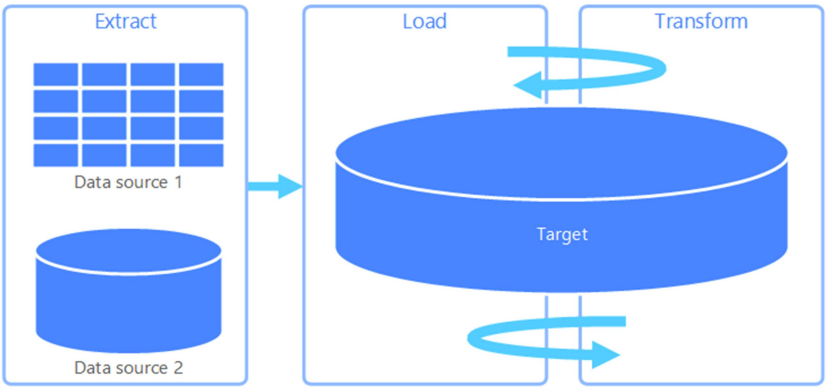
Graf 1 Pořadí četnosti databázových enginů dle Gartneru
Zdroj: [1]

4.2 Proč a kdy využít ETL

ETL je obecně popisováno jako proces, který načte z jednoho či více zdrojů data, přizpůsobí je a uloží v cílovém systému, jenž s načtenými daty pracuje odlišným způsobem. Existuje též varianta prohazující načtení a transformaci známá jako ELT. Ta se od ETL liší především tím, že nejprve data načte do cílového skladu a teprve poté je začne přizpůsobovat nové datové struktuře. Jelikož při tomto řešení cílový sklad uchovává jak transformovaná tak i netransformovaná data, vyžaduje si toto řešení podstatně více diskového prostoru. Typicky je též náročnější i na ostatní systémové prostředky, a proto se takové řešení nejvíce nasazuje v cloudovém prostředí, kde lze výkon jednoduše škálovat. [2]



Obr. 1 Obecný průběh ETL
Zdroj: [2]



Obr. 2 Obecný průběh ELT
Zdroj: [2]

Podobně jako existuje nepřeberné množství DBMS systémů je též k dispozici spousta různých ETL řešení, která slouží různým účelům a potřebám a reflektují odlišné preference v závislosti na tom, kde, jak a proč jsou implementovány. Rozlišnosti potřeb v různých prostředích lze demonstrovat pomocí dvou modelových situací. Jedním typickým příkladem nasazení je synchronizace bankovních transakcí. Z uživatelského hlediska je vidět, že při převodu peněz mezi různými bankami peníze nejsou převedeny okamžitě, nýbrž až následující den. Příčinou této prodlevy je použitá technologie, jež funguje na principu předem plánovaných importů dat mezi bankami. Tyto operace probíhají v nočních hodinách neb právě to je čas, kdy bankovní systémy mají nejnižší zátěž od uživatelů. Druhou modelovou situací mohou být sociální sítě. Uživatel chce sdílet fotografii z jedné sítě do druhé, klikne na příslušné tlačítko, fotka se ihned začne přenášet a během chvilky je sdílena i na síti druhé. Co plyne z těchto dvou příkladů? Některé aplikace potřebují, aby přenosy dat probíhaly plánovaně, nebo aby nevytěžovaly hardware serverů ve výkonově kritických časech, a na druhé straně jsou aplikace, které vyžadují co nejbrzčí přenos bez ohledu na možnou zátěž.

4.3 Extrakce

Prvním krokem ETL je obecně extrahování dat ze zdrojových systémů. Ty se však mohou v menší či větší míře lišit od cílového skladu. Před začátkem extrakce je tedy nutné vědět, jaká data bude potřeba načítat.

4.3.1 Možné datové zdroje

Zdroj dat lze vybírat ze spousty různých možností počínaje vybranými aplikacemi, přes různé databázové systémy, strukturované či nestrukturované datové soubory, po komunikační protokoly jako jsou např. FTP či HTTP/HTTPS. Dle TechNetu lze jednotlivé zdroje rozčlenit do čtyř skupin, jimiž jsou aplikační systémy, CRM systémy, databázové systémy a protokoly či systémy front. [3] Nutno dodat, že byť SSIS ihned po nainstalování nabízí nemalý a pro velkou část uživatelů jistě i dostačující výběr možností, lze je dále rozšiřovat pomocí instalovatelných nástaveb třetích stran, tzv. Component. Jako typický příklad lze uvést načítání dat z NoSQL databází. Pro ty SSIS v jeho "čisté" podobě podporu nenabízí, existuje však množství Component, které si s nimi rozumí. Příkladem je možno uvést SSIS Productivity Pack od společnosti KingswaySoft [4], jenž umí zpracovávat data ze tří různých NoSQL databázových systémů: Amazon DynamoDB, Apache Cassandra a MongoDB. SSIS Productivity Pack ovšem je poměrně rozsáhlá nástavba a NoSQL není jeho jediným, byť silným, lákadlem. Dalšími lákadly v podobě datových zdrojů mohou být třeba cloudová úložiště (Amazon S3, Dropbox, OneDrive a další), JSON, XML, SFTP nebo FTPS, některé služby Googlu (konkrétně Ads, Analytics, BigQuery a Sheets) a další. Componenty však nemusí nutně rozšiřovat pouze podporované datové zdroje, umí přidat i různé další nástroje. Zmíněná nástavba takto přidává například funkcionalitu pro anonymizování dat, kompresi, šifrování či změnu časového pásma.

4.3.1.1 Relační databáze

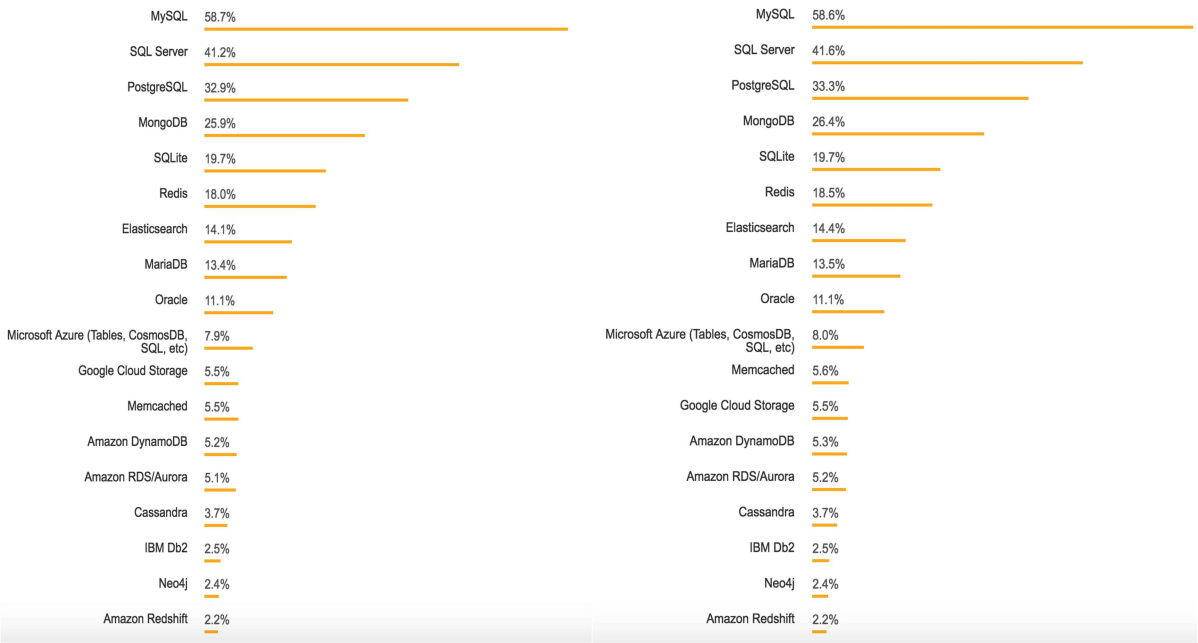
Dlouhodobě nejrozšířenější databázovou technologií jsou relační databáze. Z výzkumu podílu používání jednotlivých databázových technologií společnosti EverSQL prováděného mezi uživateli vývojářského portálu StackOverflow [5] lze zpozorovat rostoucí podíl moderních technologií jako např. MongoDB, Elasticsearch nebo Redis. Na druhou stranu první tři místa stále pevně drží právě relační databáze v čele s MySQL. Co víc, podíl SQL Serveru a PostgreSQL roste. V případě SQL Serveru jej lze vysvětlit

distribucí v rámci Windows Server a umělou inteligencí, jenž slouží k automatické optimalizaci výkonu cloudové distribuce Azure SQL. [6] V případě PostgreSQL je možno zásluhu přičítat spokojenosti programátorů s funkcionalitou a výkonem.

Z většiny běžně používaných relačních databází dokáže SSIS načíst data bez nutnosti instalování doplňků. Konkrétně je řeč o databázových systémech Oracle, Sybase, IBM DB2, Teradata, MS Access, MySQL, PostgreSQL a samozřejmě MS SQL Server. [6]

4.3.1.2 NoSQL databáze

Databáze typu NoSQL v posledních letech patří mezi postupně se rozšiřující technologie, jak také dokládá statistika EverSQL [5]. Žebříček používanosti čítá 18 databázových systémů, z nichž polovina jsou relační a polovina NoSQL. Nejrozšířenější distribucí NoSQL databáze, s níž má zkušenosti lehce přes čtvrtinu dotázaných vývojářů, je MongoDB. Ve zmíněném článku se nachází dva grafy. Z levého grafu lze vyčíst, že celkový podíl vývojářů, kteří mají zkušenost s MongoDB, je 25,9%. V pravém grafu vykazuje MongoDB ještě o něco málo vyšší podíl, konkrétně 26,4%. Tento graf se zaměřuje pouze na profesionální vývojáře. Je tedy zřejmé, že v produkčním prostředí má MongoDB silnější obsazení. Při porovnání těchto dvou grafů je též zřejmé, že na rozdíl od relačních databází mají s NoSQL databázemi profesionální programátoři vždy vyšší nebo při nejmenším stejné procento. Ač jsou rozdíly veskrze marginální, je jasné, že popularita této technologie není zanedbatelná. Možnost zpracovávat tento typ databází jako zdroj pro SSIS však poskytují pouze nadstavby třetích stran formou Component.



Graf 2 Zkušenosti s DBMS mezi uživateli StackOverflow
Zdroj: [5]

4.3.1.3 Webové API

Na rozdíl od NoSQL je podpora HTTP/HTTPS k dispozici bez nutnosti instalování pluginů. Asi nejobecnější způsob, jakým lze pomocí SSIS načítat data pomocí protokolu je parsování XML dokumentů. Co ovšem chybí, to je podpora formátu JSON. K tomuto účelu existují různé alternativy v podobě instalovaných komponent. Společnost ZappySys nabízí dvě možná řešení. Buď jde o JSON Integration Pack zaměřující se pouze na parsování tohoto konkrétního formátu, nebo o Web API Integration Pack, což je komplexnější řešení, jež rozšiřuje možnosti dále o protokol REST, SOAP, OData, či autorizaci pomocí OAuth 2.0, popř. i starší 1.0 [7].

4.3.1.4 Datové soubory

Při použití souboru uloženém na disku počítače coby zdroje je též nutné počítat s tím, že pro některé běžné formáty není k dispozici kompatibilita již v základní podobě SSIS. Formáty souborů, pro které není třeba nic instalovat, jsou MS Excel, CSV a textové soubory. Pro načtení souborů formátu JSON lze rovněž použít komponentu zmíněnou ve webových API JSON Integration Pack [7].

4.4 Transformace

4.4.1 Konverze datových typů

Jedním z velmi běžných problémů při transformaci dat jsou jistě datové typy. I na první pohled podobné databázové systémy se mohou lišit v interpretaci některých datových typů. Dobrým příkladem tohoto problému může být hodnota jednoho bitu pro programátory známá spíše jako boolean. MS SQL Server tento datový typ interpretuje jednou možností, a sice typem BIT reprezentovaným hodnotou 0 nebo 1, též dovolujícím zápis v podobě řetězce 'TRUE' pro 1, popř. 'FALSE' pro 0. V případě databáze MySQL se nabízí dva možné typy: BIT či BOOLEAN. BIT se chová velmi podobně jako v SQL Serveru, ovšem nepodporuje zápis v podobě řetězce. BOOLEAN pak navíc k číselným hodnotám přidává i literály TRUE a FALSE. Oba tyto typy byly do verze 5.0.3 datově reprezentovány jako TINYINT o délce 1, později však implementace typu BIT byla vylepšena. Je tedy třeba počítat s tím, že v datech se mohou nacházet i hodnoty vyšší než 1. Naproti tomu SQL Server při použití BITu pokus o uložení vyšší hodnoty automaticky opravuje na 1. V databázi SQLite ovšem žádný ekvivalent neexistuje a oficiální dokumentace doporučuje použít INTEGER. Zde tedy integrita dat není vůbec zaručena. [8, 9, 10]

4.4.2 Nalezení a odstranění duplicit

Typickou součástí transformace dat je též řešení duplicitně uložených záznamů. Nejčastěji vznikají v číselnících, což může být ku příkladu seznam adresátů, zákazníků, výrobků apod. Tento problém však není úplně triviální. Je třeba brát v potaz, že se nemusí jednat pouze o totožné hodnoty s různými primárními klíči, ale rovněž o různé textové reprezentace téhož subjektu.

4.4.3 Doplnění hodnot

Též se může stát, že chybí některé hodnoty. Děje se tak tak proto, že daný sloupec může dovolovat ukládání hodnoty NULL. Pokud však v nové databázi uložení NULL tento sloupec nedovoluje, musí se tyto údaje pro přenos doplnit. Tento nedostatek lze řešit několika různými způsoby v závislosti na tom, jaká data chybí. Ku příkladu těžké by bylo vymyslet adresu pouze na základě jména osoby. Takový záznam můžeme buď ignorovat, nebo opatřit implicitní hodnotou (např. "neuvedeno"). Na druhou stranu také

může nastat situace, kdy doplnění hodnoty není nikterak složitý problém. Jako modelová situace může být třeba záznam objednávky s uvedenou jednotkovou cenou a množstvím, ale chybějící celkovou cenou. Pak jednoduše stačí údaje navzájem vynásobit a požadovaný údaj je na světě.

	EmpID	Name	Salary	Designation
1	1	Amit	12000.00	SE
2	7	Amit	12000.00	SE
3	2	Mohan	15000.00	SE
4	3	Monu	27000.00	SSE
5	6	Monu	27000.00	SSE
6	4	Riyaz	16000.00	SE
7	5	Riyaz	16000.00	SE

Obr. 3 Příklad tabulky s duplicitními daty
Zdroj: [14]

4.4.4 Sjednocení formátu dat

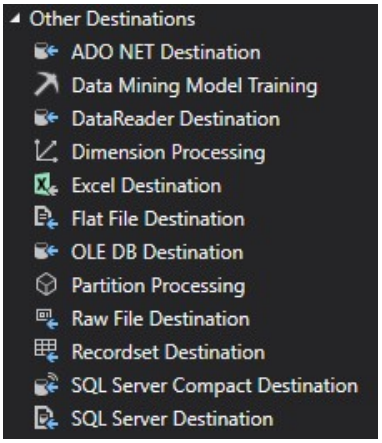
Stejně jako v případě předešlých dvou diskutovaných problémů, i různě formátovaná data mohou vzniknout nepozorností uživatelů, nedostatečným ošetřením vstupů ze strany programátora, či kombinací obojího.

4.4.5 Referenční integrita

K zajištění referenční integrity nabízí designer SSIS uživatelské rozhraní pro Check Database Integrity Task, jež je nativní funkcí T-SQL. Tato funkce využívá příkazu DBCC CHECKDB, jehož vyvolání v sobě zapouzdřuje tři další příkazy a sice DBCC CHECKALLOC, DBCC CHECKTABLE a DBCC CHECKCATALOG a několik dalších operací zajišťujích validaci indexovaných pohledů, odkazů na data uložená ve file systémů. [11]

4.5 Načtení (Load)

Poté, co proběhne extrakce a transformace, jsou data připravena pro uložení do nové databázové struktury. Podobně jako při extrakci se i tomto případě nabízí široká paleta možností. SSIS bez jakýchkoliv instalovaných doplňků nabízí dvanáct možných cílů, kam data zapisovat. Některé z nich je však možné konfigurovat pro více různých úložišť. Dobrým příkladem této nastavitelnosti je destinace OLE DB. Ta nabízí konektivitu pro SQL Server, databázi Jet, Microsoft Access, Oracle a několik dalších možností.



Obr. 4 Možné cíle načítání dat v SSIS
Zdroj: Vlastní tvorba

4.6 Ošetření chyb

Dojde-li k chybě, nabízí se vícero možných přístupů, jak pokračovat. Užitečnou praktikou jistě je zápis do logu, což ovšem neřeší ošetření chyby, nýbrž pouze její zaznamenání. Dle závažnosti chyby je otázkou, zda má smysl pokračovat ve zpracovávání dat, či proces ukončit. Typickou záminkou ukončení může být například narušení referenční integrity. Primárním cílem ETL však je přenést veškerá data a vyvinout vše tak, aby se chybám předcházelo. Pro většinu případů použití tedy je postačující zapisovat logy a v případě, že k chybě dojde, ošetřit ji.

5 Přehled dostupných ETL nástrojů

V následujících kapitolách budou postupně představeny vybrané nástroje používané pro ETL. Přednostní bude představení SSIS, poté budou následovat další možné alternativy. Ty budou následovat seskupené do logických celků dle možných preferencí u cílových uživatelů.

5.1 SQL Server Integration Services

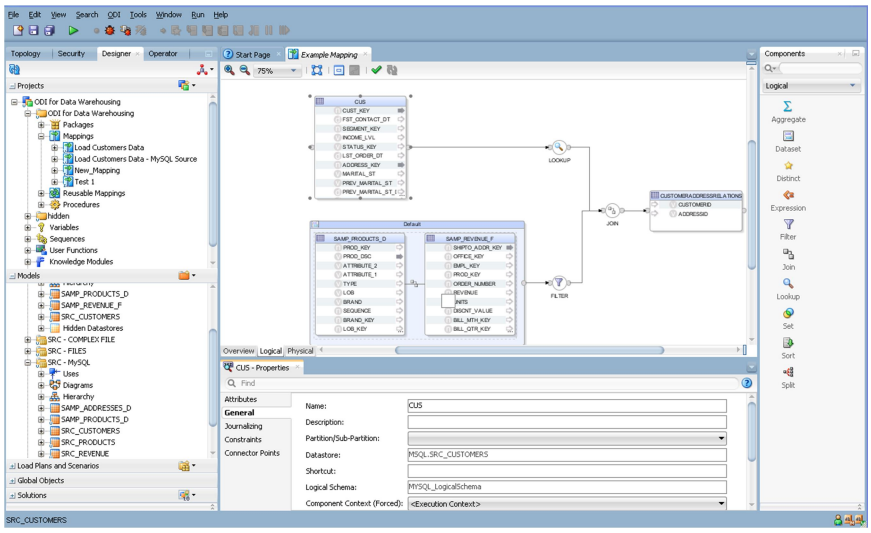
SSIS je nástroj určený pro plnění databáze Microsoft SQL Server daty z různých zdrojů. Jedná se o proprietární nástroj vyvíjený firmou Microsoft určený primárně pro SQL Server a další nástroje této společnosti, podporuje však i jiná úložiště například databáze Oracle. [12] Základem při načítání dat pomocí SSIS jsou balíčky. Pomocí nich lze definovat z jakého zdroje se mají data získávat, jakým způsobem se musí zpracovat a kam se mají uložit. Definování těchto balíčků probíhá ve vývojovém prostředí Microsoft Visual Studio. Aby bylo možné Visual Studio využít k tomuto účelu, je nutné zajistit, aby byla nainstalována také nástavba SSDT (dále jen SQL Server Data Tools), jejíž součásti je i SSIS.

5.2 Ostatní dostupné nástroje

Na trhu dostupných ETL nástrojů zdaleka není Microsoft jediným majoritním hráčem, ovšem faktem též je, že v tomto odvětví se věnuje vývoji softwarových nástrojů už delší dobu a SSIS v jeho stávající podobě rozhodně není prvním počinem. Stejně jako na poli databázových systémů i zde je významnou konkurencí firma Oracle.

5.2.1 Dávkové zpracování

Do nedávné doby byl jediný dostupný způsob jak řešit procesy ETL v podobě dávkového zpracování. Typické pro tento přístup je spouštění zdlouhavých a též pro systém hardwarově náročných procesů načítání dat do databáze v nočních hodinách, kdy na server nejsou kladeny takové nároky od uživatelů. SSIS, IBM InfoSphere a Oracle Data Integrator jsou typickými zástupci této skupiny. [18, 19]



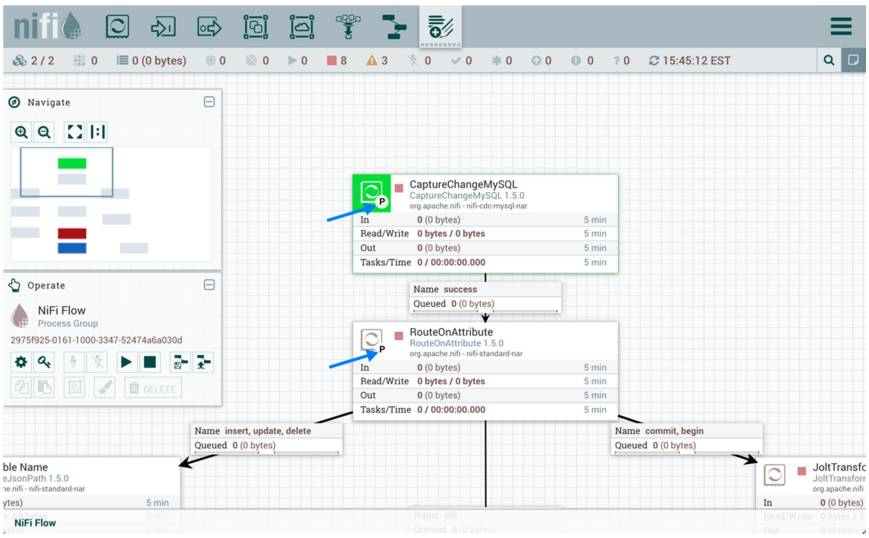
Obr. 5 Oracle Data Integrator 12c
Zdroj: [13]

5.2.2 Cloudové ETL nástroje

V posledních letech se ve velkém migruje IT do cloudu. Spolu s tím ruku v ruce vznikla nový typ ETL nástrojů - cloudové. Spojuje je tedy přizpůsobení pro běh ve virtuálním prostředí cloudu, v realizaci importu se však tyto nástroje často různí. Některé fungují na principu "klasických" dávkových, jiné zase načítají data v reálném čase. Příklady mohou být Alooma nebo Fivetran. [18, 19]

5.2.3 Open-source nástroje

Na poli open-source softwaru též existuje velké množství dostupných řešení. Velkým zástupcem této kategorie je bezesporu Apache, neboť nabízí hned tři různá řešení. Airflow, Kafka, NiFi jsou sice ETL řešení od jedné firmy, ovšem jejich určení se prakticky neprolíná, nelze je tedy považovat za vzájemnou konkurenci a z pravidla je ani nelze zaměnit jedno za druhé. Neboť Apache je pouze nadace sdružující vývojáře open-source softwaru je dobré zmínit, jaké firmy stojí za vývojem jednotlivých řešení. Kafka je dílem sociální sítě LinkedIn [20], AirBnB je autorem Airflow a NiFi vyvíjí americká NSA. Tyto nástroje, jak je možná z autorů zmíněné trojice zřejmé, vznikají pro specifické potřeby společnosti, jenž je vyvíjí. Teprve později pak jsou uvolněny pod otevřenou licencí. [18]



Obr. 6 Náhled prostředí Apache NiFi
Zdroj: [15]

5.2.4 Real-time nástroje

Pravděpodobně největší technologickou inovací jsou nástroje pracující v reálném čase. Typicky se opírají o frontu zpráv, která oznamuje datové změny. Tento přístup je užitečný, pokud je třeba držet data co nejvíce aktuální. Své užití může najít ve spoustě scénářů od sdílení obrázku na sociální sítě, přes odeslání objednávky do e-shopu, po transakci elektronickou platební bránou. Tyto a spousty dalších úkonů vyžadují okamžitou synchronizaci dat. Možnými zástupci této skupiny jsou Alooma, Confluent a StreamSets. [18, 19]

6 Implementace v praxi

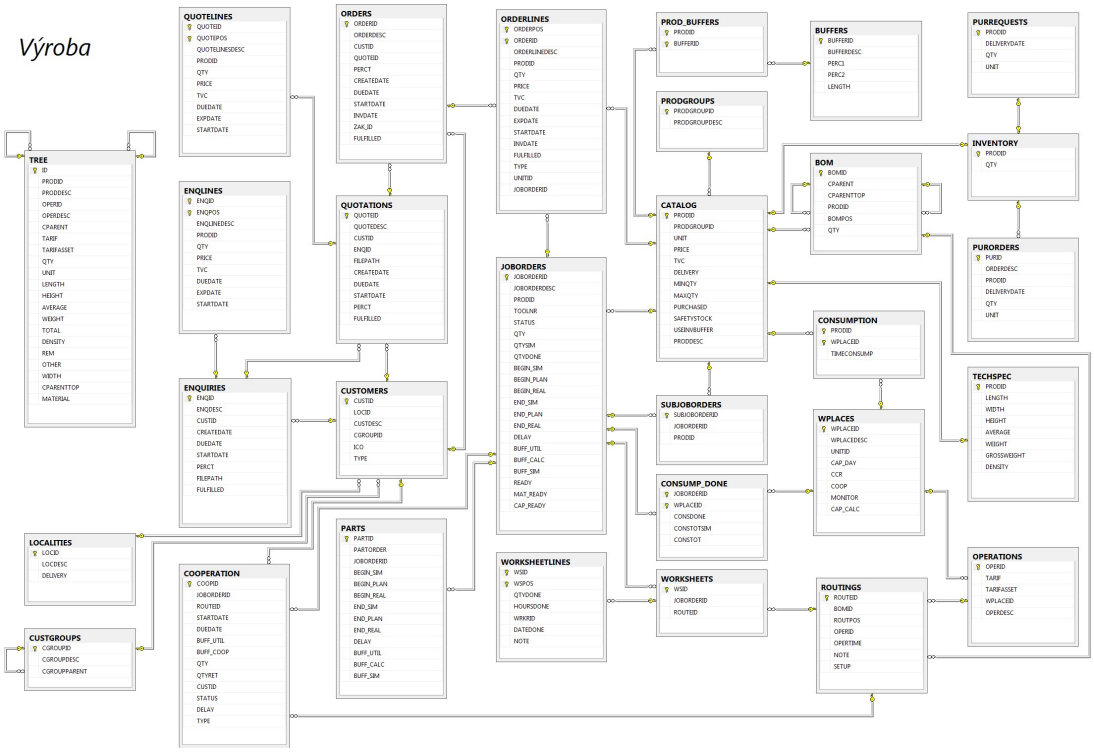
Praktická ukázka nasazení procesu ETL bude provedena na datech client-serverové podnikové aplikace. Cílem je demonstrovat, jak může vypadat reálná implementace pomocí SSIS. V průběhu bude třeba instalovat SSDT, SQL Server a další potřebný software, navrhnout a realizovat novou zjednodušenou databázi, vytvořit SSIS balíček pro přenos dat ze staré do nové databáze a nastavit cyklické spouštění.

6.1 Definice cílové databáze

Důvodem k migraci je především přílišná složitost, zastaralé použité technologie a nižší robustnost kódu, neboť aplikace byla z velké části vyvíjena za běhu. Prakticky všechny technologie použité při vývoji jsou dnes již překonány a používají se při nejmenším jejich novější verze. Jedná se o webovou aplikaci napsanou v ASP.NET, nikoliv však s použitím moderní architektury Model View Controller, nýbrž v dnes již několikátým rokem nepodporovaných WebForms. Aplikace byla kompilována ve Visual Studiu 2012 a verzi .NET Frameworku 4.5, je však plně kompatibilní i s běhovým prostředím verze 4.0 a byla pod ním i testována. Ze začátku byla též vyvíjena ještě ve Visual Studiu 2010. Serverovou logiku obstarává webová služba naprogramovaná v jazyce C#, s níž komunikuje klient ve webovém prohlížeči využívající starší JavaScript splňující standart ECMAScript verze 5.

6.1.1 Původní databáze

Co se týče zbytečné složitosti, ta je především problémem databázové struktury. Ta se za prvé skládá ze dvou víceméně oddělených skupin tabulek. Jednou z těchto skupin je infrastruktura, tou druhou pak výroba, což je právě ta část, která je zajímavá, neboť v ní jsou data, na která se nová databáze má soustředit. Zbytečná složitost struktury staré databáze reflektuje především přibývající požadavky na funkcionalitu, z nichž od některých bylo později upuštěno a od části implementovaných bude oproštěna nová aplikace. Při návrhu nové struktury je též na místě zbavit se nešvarů v podobě používání plurálů v názvech tabulek a nekonzistence v používání podtržítek pomocí jasnější jmenné konvence.



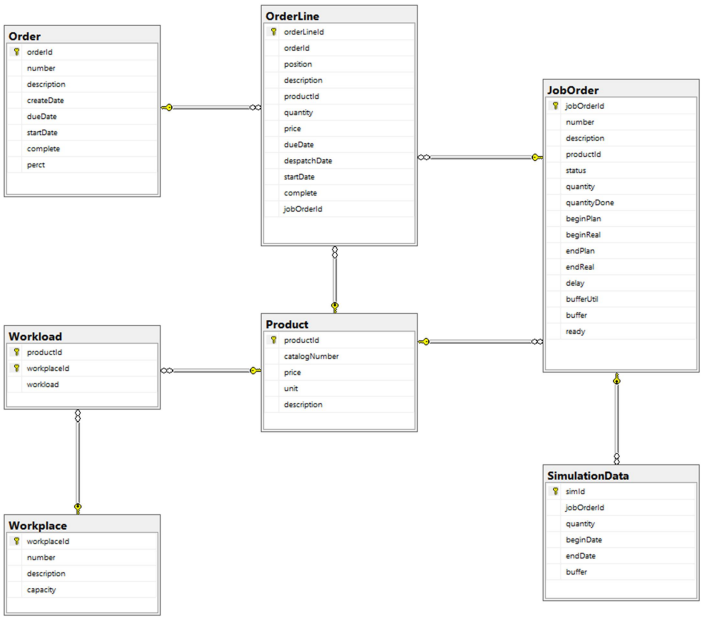
Obr. 7 Náhled části struktury databáze původní aplikace pro výrobu
Zdroj: Vlastní tvorba

6.1.2 Zjednodušení struktury

Příkladem zjednodušení může být zjednodušení životního cyklu objednávky. V závislosti na procesech jednotlivých firem se i tento životní cyklus může různit, celý tedy lze popsat následovně. Zákazník zašle firmě poptávku (tabulky ENQUIRIES a ENQLINES ve staré struktuře), firma poté poptávku zpracuje a vystaví nabídku (QUOTATIONS a QUOTELINES). Poté zákazník objedná zboží a zašle objednávku (ORDERS a ORDERLINES), na základě přijaté objednávky pak lze vytvořit výrobní zakázky (JOBORDERS), z nichž lze vypočíst plán výroby. Je možné si všimnout, že poptávky, nabídky a objednávky jsou složeny dvojicemi tabulek, ovšem zakázky nikoliv. To lze jednodušše vysvětlit tím, že zakázky pro výrobu jsou tvořeny čistě na základě výrobků, zatímco ostatní mají společnou hlavičku a jednotlivé řádky odpovídající jednotlivým výrobkům. První dva kroky tohoto procesu jsou však pro plán irelevantní a velká část firem je navíc ve svých procesech vůbec nemá. Nová aplikace má za cíl být především univerzální a nebude tedy řešit celý cyklus včetně poptávek a nabídek, soustředit se bude pouze na objednávky a výrobní zakázky.

6.1.3 Návrh nové databáze

Tvorba nové databáze pak proběhla pomocí nástroje instalovaným pro správu SQL Serveru, SQL Server Management Studio (dále jen SSMS). Pomocí tohoto nástroje se navázalo spojení s lokální instancí databázového enginu. Založila se nová databáze pod jménem easy_plan a následně se příslušným editorem vytvořili potřebné tabulky. Nakonec byl vytvořen Database Diagram pro definici vazeb primárních a vzdálených klíčů. Výsledný diagram je přiložen v podobě obrázku (Obr. 9). Všechny tyto operace byly provedeny grafickým editorem, jenž je pouze uživatelsky přívětivější alternativou SQL příkazů CREATE DATABASE, CREATE TABLE a ALTER TABLE ADD FOREIGN KEY. Pro úplnost je též vhodné zmínit, že v diagramu chybí tabulka ImportLog, která nesouvisí přímo s přenášenými daty, nýbrž k tomu, aby nesla informace o proběhlých přenosech.



Obr. 8 Náhled nové struktury pro plánování výroby
Zdroj: Vlastní tvorba

6.2 Prostředí

Pro co nejjistější kontrolu nad během ETL byl zvolen běh v nativním prostředí. Přenos velkého množství dat vyžaduje pro svižný chod co nejvyšší výkon a toho lze docílit především pomocí přímého přístupu operačního systému k hardwaru. V rámci zajištění co nejlepší kompatibility s vývojovým prostředím Visual Studio a SQL

Serverem je ideální použít operační systém Windows, jelikož tyto nástroje byly po dlouhá léta vyvíjeny výhradně pro tento operační systém.

6.2.1 Hardware

Roli pracovní stanice zastal stolní počítač s osmijádrovým procesorem řady AMD FX běžící na frekvenci 4,2GHz doplněný 16GB operační paměti DDR3 na frekvenci 1600MHz. Jako úložné médium pro zdrojový i cílový datový sklad posloužil pevný disk typu SSD pro rozhraní NVM Express. Přestože tato konfigurace hardwaru je v dnešní době už poněkud zastaralá, pro potřebné účely stále slouží bez sebemenšího zaváhání. Vysoká frekvence procesoru a pevný disk s minimální latencí a vysokými čtecími rychlostmi více než dostatečně kompenzují postarší architekturu.

6.2.2 Windows 10

Jako operační systém byl vybrán Windows 10 Pro především kvůli kompatibilitě s potřebným softwarem. Je sice faktem, že SQL Server 2017 je už v dnešní možné spustit i na operačních systémech mimo rodinu systémů Windows, ovšem rozhodující faktor sehrála kompatibilita s Visual Studiem. Plnohodnotná verze Visual Studia je stále kompatibilní pouze se systémem Windows, byť existuje i multiplatformní Visual Studio Code a specifická verze Visual Studio for Mac přizpůsobená pro prostředí systém Mac OS X. Tyto varianty jsou však zcela jinými produkty a spojuje je pouze název, ani jednu z nich nelze použít pro vyvíjení SSIS balíčku.

6.2.3 SQL Server 2017

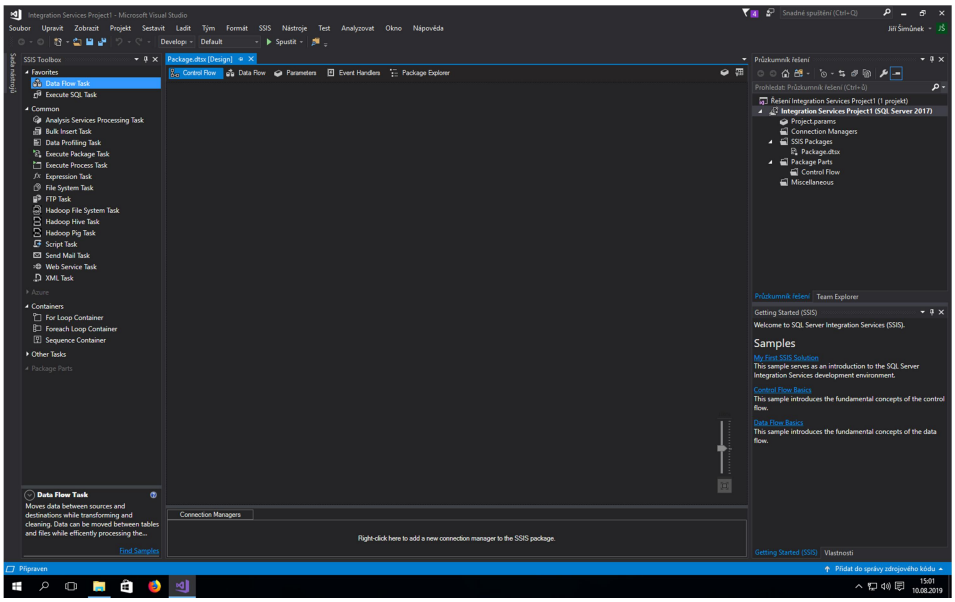
DBMS v podobě SQL Serveru je v podstatě středobodem práce s SSIS a též nutnou podmínkou jeho použití, neboť SSIS je uzpůsoben na míru právě pro SQL Server. SQL Server je relační databázový systém vyvíjený firmou Microsoft. Jedná se o plně vybavenou databázi primárně určenou jako konkurenci pro Oracle Database a MySQL. Jako všechny významné RDBMS, SQL Server podporuje ANSI SQL, neboli standartizovaný jazyk SQL. SQL Server ovšem také obsahuje T-SQL, což je jeho vlastní implementace SQL. Správu obsluhuje SSMS (dříve známé jako Enterprise Manager), jež podporuje jak 32-bitová tak i 64-bitová prostředí. SQL Server se také někdy označuje jako MSSQL nebo Microsoft SQL Server.[16] K databázovému enginu byla též instalována nástavba SSMS, která přidává grafické rozhraní pro jednodušší správu. Lze s

její pomocí upravovat strukturu, prohlížet a spravovat uložená data, přidělovat oprávnění k jednotlivým objektům a mnoho dalšího.

6.2.4 Visual Studio 2017

SSIS je součástí balíčku SSDT, jež je pluginem pro vývojové prostředí Visual Studio. Použita byla verze Visual Studio 2017 v edici Community. Tato edice je vybavením shodná s variantou Professional, oproti ní však má podstatně jinou licenční politiku. Community je sice zdarma, ale není dovoleno používat jí v podniku vykazujícím zisk na úrovni 1 milion USD nebo vlastnícím 250 PC. [17]

SSDT je též možné instalovat i za předpokladu, že v počítači dosud není instalováno Visual Studio. V takovém případě se spolu s SSDT nainstaluje i Visual Studio Integrated Shell, což je v podstatě plnohodnotná distribuce pouze bez předinstalované podpory kteréhokoliv programovacího jazyku. V obou případech je tedy nutné mít nainstalovanou nějakou formu tohoto softwaru.



Obr. 9 Prostředí Visual Studio 2017 při tvorbě SSIS balíčku
Zdroj: Vlastní tvorba

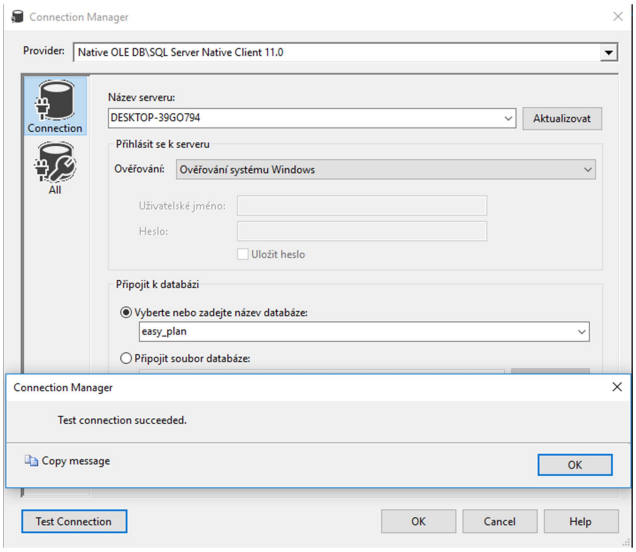
6.3 Příprava balíčku

Tvorba SSIS balíčku pro ETL probíhá, jak již bylo zmíněno, ve vývojovém prostředí Visual Sudio. Prvním krokem je tedy vytvoření nového projektu typu “Integration Services Project” a stanovení jeho jména. Při výběru jména jako inspirace posloužil název databáze a výchozí jméno SSIS projektu. Po vytvoření projektu se Visual Studio přepne do rozhraní pro tvorbu SSIS balíčků, kde se při výchozím rozložení ovládacích

prvků v prostřední části okna zobrazí pracovní plocha s několika záložkami, z nichž nejdůležitější jsou Control Flow a Data Flow. V levé části okna se zobrazí SSIS Toolbox obsahující nástroje k příslušné záložce pracovní plochy, v dolní části seznam dostupných Connection Managerů a v pravé části průzkumník projektu.

6.3.1 Connection Manager

První místo, které je zajímavé v rámci tvorby balíčku, je záložka Control Flow. Zde se definuje obecný průběh procesu v podobě sekvence tasků. Task může mít mnoho různých podob, ovšem ne všechny jsou pro přenos dat z databáze do databáze přínosné. Mezi takové patří ku příkladu FTP Task nebo XML Task sloužící k práci s FTP serverem, respektive s XML soubory. Ovšem zcela určitě se hodí použít Data Flow Task, neboť ten je určen přesně pro potřebný účel. Než však bude přidán první takový task, je vhodně nadefinovat si pořebné Connection Managery. Pro připojení k SQL Serveru je třeba specifikovat poskytovatele, název serveru, metodu přihlášení včetně uživatele a hesla pro případ SQL autentizace a vybrat konkrétní databázi pomocí jejího názvu či cesty k souboru. Jako poskyvatele je možné využít standart OLE DB, ADO.NET nebo ODBC. Jako kompromis mezi výkonem a komplexností padla volba na OLE DB klienta. Pro ověření správnosti nastavení je k dispozici tlačítko pro otestování konektivity. Na přiloženém obrázku (Obr. 10) je vidět kompletní nastavení pro databázi easy_plan a dialogové okno oznamující, že test připojení byl úspěšný. Obdobně nastavíme připojení i ke druhé databázi.



Obr. 10 Nastavení připojení k databázi easy_plan
Zdroj: Vlastní tvorba

6.3.2 Control Flow

Na záložce Control Flow se definuje obecný průběh celého přenosu. Jako stavební kameny pro tuto záložku slouží tasky a constrainty. Task reprezentuje dílčí operaci nebo ucelenou skupinu operací. Může jím být na příklad spuštění procesu, odeslání e-mailu, ověření integrity dat, nebo třeba vykonání SQL dotazu. Specifickým příkladem pak je Data Flow Task. Ten vyžaduje připravit svůj průběh skrze druhou záložku zvanou Data Flow. Druhým pomyslným stavebním kamenem v Control Flow je constraint, což je prvek znázorněný šipkou a udávající návaznost jednotlivých tasků. Spojuje vždy dva po sobě jdoucí tasky, ovšem na vstupu i výstupu daného tasku jich může být vícero. Více constraint na výstupu značí, že následující tasky se vykonávají paralelně. Více vstupních constraint znamená, že následující task může být vykonán, pouze pakliže všechny předcházející úspěšně proběhly.

Na základě těchto faktů byla sestrojena taková posloupnost tasků, aby při respektování struktury cílové databáze mohlo být dosaženo co možná nejvíce paralelně zpracovávaných tabulek. Pro přenos dat do jednotlivých tabulek cílové databáze jsou použity Data Flow Tasky. První z nich načte data do tabulky Product, jelikož její záznamy jsou nejčastěji odkazovány pomocí vzdáleného klíče do ostatních tabulek. Dále následují hned tři Data Flow Tasky paralelně, jmenovitě pro tabulky Order, JobOrder a Workplace. Na ně pak navazuje načtení tabulek OrderLine, SimulationData a Workload. Pro zpětné uchování již proběhlých přenosů je Control Flow doplněno též SQL tasky, které ukládají několik základních informací o průběhu do tabulky ImportLog.

Obr. 11 Control Flow
Zdroj: Vlastní tvorba

6.3.3 Data Flow

Vlastní průběh Data Flow Tasku se konkretizuje v záložce Data Flow. V případě jednoduchých přenosů, kdy data lze mapovat přímo z tabulky do tabulky, stačí přidat pouze zdroj a cíl, neboť u zdroje dat se specifikuje konkrétní tabulka a její sloupce, se kterými task pracuje, u cíle pak cílová tabulka a mapování sloupců ze zdroje. Tímto jednoduchým postupem však nelze mapovat přenos žádné z tabulek v databázi, neboť při opakovaném spuštění by poté docházelo k duplikování záznamů, což je efekt nežádoucí.

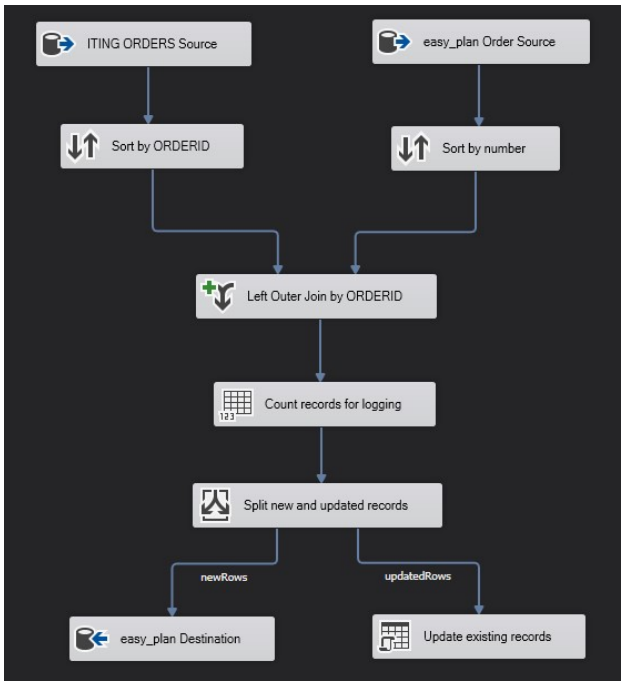
6.3.3.1 Použité transformace

Pro dosažení požadovaných výsledků bude využito několika transformací. Jmenovitě se jedná o následující: Sort, Merge Join, Conditional Split, Derived Column, OLE DB Command a Row Count. Sort je funkcí k seřazení řádků použitou z pravidla před funkcí Merge Join, jež vyžaduje setříděné vstupy. Merge Join slučuje řádky dvou vstupních tabulek pomocí shodných hodnot v daných dvou sloupcích. Conditional Split pak dělá přesný opak, jelikož dle specifikovaných podmínek rozděluje data do vícero proudů. Derived Column vytvoří pomocí daného pravidla nový sloupec. OLE DB

Command pošle SQL příkaz. Poslední zmíněný, Row Count, slouží ke spočítání řádků a uložení této informace do proměnné.

6.3.3.2 Zamezení duplicit

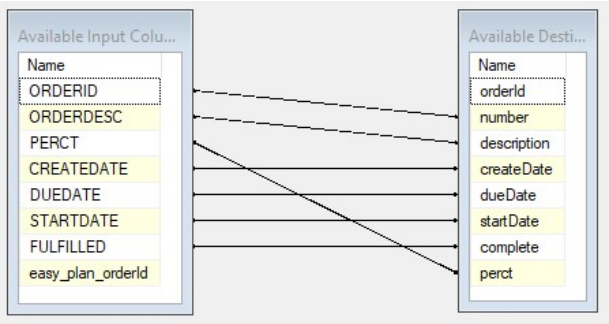
Jak předejít těmto duplicitám lze nejlépe předvést na Data Flow Tasku pro tabulku Order, neboť v něm nedochází k žádné transformaci dat. Diagram tedy obsahuje pouze logiku, která je nezbytná k načtení nových, po případě aktualizaci stávajících záznamů. V horní části přiloženého diagramu (Obr. 12) se nachází dva zdroje dat: jak zdrojová tabulka ORDERS v databázi ITING, tak i cílová tabulka Order v databázi easy_plan. Záznamy z těchto tabulek se propojí SSIS komponentou Merge Join pomocí left outer joinu, čímž je zajištěno, že všechny nové záznamy z ORDERS (ITING) jsou načteny a ke všem již existujícím je připojen příslušný řádek z Orders (easy_plan). Na konci diagramu se záznamy rozdělují pomocí SSIS komponenty Conditional Split. Využitím jednoduché podmínky, která SQL funkcí ISNULL ověří, zdali pro daný záznam existuje primární klíč v databázi easy_plan, se rozdělí záznamy na nové nebo upravené a ty se poté příslušným způsobem uloží. Pro úplnost je též správné poznamenat, že bylo-li by cílem záznamy též mazat, musí se využít full outer join a rozšířit split o třetí větev.



Obr. 12 Data Flow pro naplnění tabulky Order
Zdroj: Vlastní tvorba

6.3.3.3 Mapování transformovaných dat do tabulky

Nyní, když je přenos ošetřen proti duplikování dat, je třeba zajistit, aby jednotlivé sloupce vzniklého setu byly napojeny na odpovídající sloupce v nové tabulce. Na přiloženém obrázku je zachyceno napojení tabulky Order v editoru mapování. Sloupec orderId v cílové tabulce není napojen k žádnému zdroji, jelikož při vložení nového řádku se nový primární klíč v databázi vygeneruje sám. Ve zdrojové tabulce též můžeme vidět nepřipojený sloupec easy_plan_orderId. Ten je napojen k primárnímu klíči (orderId) v druhém výstupu splitu a slouží k identifikaci řádků, jež se mají aktualizovat.



Obr. 13 Mapování tabulky Order
Zdroj: Vlastní tvorba

6.3.3.4 Nalezení primárních klíčů

Ani takovéto řešení řešení však nelze aplikovat na všechny tabulky. Při plnění většiny ostatních tabulek je třeba mít na paměti, že stará a nová databáze mají rozdílné primární a kvůli tomu i vzdálené klíče. Tabulka Workload má vazby na tabulky Product a Workplace, stejně tak tabulka CONSUMPTION má odpovídající vazby na CATALOG a WPLACES. Pro nalezení odpovídajících primárních klíčů v databázi easy_plan se musí provést dvakrát operace Merge Join a sice pro připojení tabulek Product a Workplace. Před provedením Merge Join ještě Data Flow SSIS vyžaduje setřídit data dle odpovídajících sloupců, na kterých Merge Join proběhne. Na konci přiloženého diagramu (Obr. 14) je join a nezakončený constraint, jenž je následován splitem detekujícím duplicity obdobně jako v tabulce Order.

Obr. 14 Data Flow pro naplnění tabulky Workload
Zdroj: Vlastní tvorba

Obdobným způsob nalezení vzdálených klíčů je aplikován i u ostatních tabulek. Specifický přístup vyžaduje přenos dat z JOBORDERS. Aby nová aplikace umožňovala uložit vícero plánů, rozděluje nová databáze JOBORDERS na dvě tabulky a sice JobOrder a SimulationData. Pro přehlednost je načítání dat do jednotlivých tabulek rozděleno do samostatných tasků. V tomto případě je JOBORDERS hlavním zdrojem dat u dvou tasků. V prvním z nich se naplní tabulka JobOrder pomocí joinu JOBORDERS ze staré a Product z nové databáze. Teprve poté je možné pokračovat druhým taskem s naplněním tabulky SimulationData joinem JOBORDERS (ITING) a JobOrder (easy_plan).

6.3.3.5 Oprava datových typů

Nejednou opakovaným krokem v data flow je též oprava kompatibility datových typů. Dobrým příkladem takové opravy je sloupec Description v tabulce OrderLine, jehož datovým typem je VARCHAR o maximální délce 80 znaků. Zdrojem dat pro tento sloupec je ORDERLINEDESC z tabulky ORDERLINES s datovým typem VARCHAR maximální délky 255. Zkrácení řetězce znaků lze provést transformací Derived Column. S využitím SQL funkce LEFT dojde k oříznutí řetězce od počátku do maximální délky stanovené parametrem, v tomto případě 80 znaků. Pro takto upravený text je dále třeba

nastavit datový typ a pojmenovat sloupec. Jako inspirace pro pojmenování posloužil původní sloupec a jeho nová délka – ORDERLINEDESC_80. Nad datovým typem pak není třeba příliš přemýšlet, zde je třeba zvolit string délky 80.

6.3.4 Logování

Pro možnost trasování chyb v přenosech nebo k nahlížení do historie již proběhlých přenosů je vhodné zaznamenávat log. SSIS nabízí možnost automaticky zaznamenávat vybrané typy událostí do několika možných úložišť. Jmenovitě lze log ukládat na SQL Server, do textového nebo XML souboru anebo do Windows Event Logu. Na výběr je k dispozici více než 20 různých typů událostí, jejichž zaznamenávání lze omezit na konkrétní tasky, či na celý balíček. Pro diagnostiku možných chyb postačí ukládat pouze několik událostí a zapisovat log do textového souboru. Běžnému uživateli však takovýto záznam příliš mnoho informací neprozradí, protože nese systémové hlášky, jimž lze porozumět s patřičným technickým povědomím.

Jako zajímavější pro uživatele byla implementována tabulka ImportLog. Je navržena tak, aby z ní bylo možné zobrazovat přehledy o proběhlých importech dat přímo v aplikaci. Jsou do ní zapisovány základní statistiky o proběhlých importech v podobě začátku, konce, načtení jednotlivých tabulek a případných chyb. U každého záznamu v tabulce ImportLog je uložen čas, typ události a popis, volitelně pak jsou uloženy navíc název entity a počet zpracovaných řádků. Statistiky jsou ukládány pomocí SQL Tasků přímo v Control Flow (Obr. 10). Pod názvem entity se ukládá název zpracované tabulky, nebo název tasku, který vyvolal chybu, zjištěný počet zpracovaných řádků je uložen do proměnné v rámci příslušného Data Flow. Chyby jsou zaznamenány pomocí Event Handleru reagujícího na události OnError a OnTaskFailed. Zaznamenávání probíhá kontextuálně na úrovni balíčku, aby bylo zajištěno, že se všechny chyby uloží. Pro vytvoření akce pro Event Handler se využívá editor s obdobnými možnostmi jako pro Data Flow. K uložení záznamu do tabulky ImportLog byl využit jeden task v podobě SQL skriptu. Možný výstup do logu je znázorněn v následujícím obrázku. Zde jsou pod sebou zaznamenány dva běhy, kde první z nich byl záměrně spuštěn s chybou pro otestování zápisu chyby.

dateOfEvent	eventName	description	entityName	rowsAffected
2019-11-12 21:16:05.963	import_started	Import job started.	NULL	NULL
2019-11-12 21:16:07.067	table_loaded	Product table updated.	Product	1480
2019-11-12 21:16:07.917	error	Executing the query "INSE...	Log error task	NULL
2019-11-12 21:16:07.957	error	Execute SQL Task	Log workpla...	NULL
2019-11-12 21:16:08.063	table_loaded	Order table updated.	Order	529
2019-11-12 21:16:08.617	table_loaded	JobOrder table updated.	JobOrder	550
2019-11-12 21:16:09.580	table_loaded	Workload table updated.	Workload	1047
2019-11-12 21:16:10.087	table_loaded	SimulationData table updat...	SimulationData	534
2019-11-12 21:16:10.853	table_loaded	OrderLine table updated.	OrderLine	681
2019-11-12 21:16:10.883	import_finished	Import job fiished.	NULL	NULL
2019-11-12 21:16:51.110	import_started	Import job started.	NULL	NULL
2019-11-12 21:16:52.150	table_loaded	Product table updated.	Product	1480
2019-11-12 21:16:53.037	table_loaded	Workplace table updated.	Workplace	15
2019-11-12 21:16:53.180	table_loaded	Order table updated.	Order	529
2019-11-12 21:16:53.590	table_loaded	JobOrder table updated.	JobOrder	550
2019-11-12 21:16:54.653	table_loaded	Workload table updated.	Workload	1047
2019-11-12 21:16:55.027	table_loaded	SimulationData table updat...	SimulationData	534
2019-11-12 21:16:55.760	table_loaded	OrderLine table updated.	OrderLine	681
2019-11-12 21:16:55.783	import_finished	Import job fiished.	NULL	NULL

Obr. 15 Záznamy v tabulce ImportLog
Zdroj: Vlastní tvorba

6.3.4.1 Proměnné

Při vytváření vlastních logů našly své uplatnění především proměnné. SSIS nabízí jak paletu předdefinovaných proměnných, tak i možnost spravovat vlastní. Ty se od sebe liší prefixem v názvu. Vlastní proměnné jsou označeny prefixem User, ty, jenž jsou součástí SSIS, začínají slovem System. Systémové proměnné jsou při logování do tabulky ImportLog použity pouze při ukládání chyb. Pro uložení záznamu za pomocí eventu typu OnError jsou využity systémové proměnné ErrorDescription jako popis chyby a TaskName jako název tasku, pro event OnTaskFailed jsou to SourceDescription pro popis a SourceName pro název tasku. Mimo vlastní logování do databáze je využívá i automatický log do souboru, jehož záznamy sestávají v podstatě pouze ze systémových proměnných. Příkladem takové proměnné může být CreatorName. Tato proměnná nese název uživatele systému, zodpovědného za vytvoření SSIS balíčku. Vlastnoručně definované proměnné pak jsou počítadla zpracovaných záznamů a typy událostí, obojí slouží pro ImportLog.

6.3.5 Optimalizace

S funkčním přenosem dat i logováním je na čase zaměřit se na možné optimalizace. Jako solidní pomůcku pro optimalizování lze použít záložku Execution Results. Na této záložce se nachází podobné informace jako je možné pomocí SSIS zaznamenat do logu, nicméně jsou podrobnější a přehledněji zobrazeny, pokrývají jen poslední spuštění, díky

čemuž se nehromadí a lze je sledovat v reálném čase. Díky tomuto nástroji jde například velmi jednoduše identifikovat nadbytečné sloupce, které se extrahují ze zdrojového skladu a neuloží do cílového. Také zde je možné sledovat, jak dlouho trvá zpracování jednotlivých tasků. Zpracovávání nepotřebných dat vytěžuje procesor i operační paměť, při dostatečně velkém vzorku dat by tedy měl být měřitelný pozitivní dopad na rychlost po odstranění těchto nadbytečných sloupců.

Nejobsáhlejší tabulkou v databázi ITING je CATALOG. Tato tabulka obsahuje všechny výrobky, díly a materiály, jež se zpracovávají ve výrobě. Není tedy divu, že tato tabulka čítá 375296 řádků. Provedením empirického testu s několika opakováními je možné změřit dopady na výkon před a po odebrání nadbytečných sloupců z Data Flow. Po prvním dokončení běhu balíčku se odstraní nadbytečné sloupce a zaznamená čas potřebný pro zpracování tasku pro naplnění tabulky Product. Při druhém spuštění je změřen čas bez zpracovávání nadbytečných dat. Několik opakování tohoto pokusu poskytne dostatečný vzorek.

Neoptimalizované spuštění (ms)	Optimalizované spuštění (ms)	Rozdíl (ms)	Zlepšení
4672	4203	469	10,0 %
4718	4187	531	11,3 %
4860	4281	579	11,9 %
5047	4172	875	17,3 %
Průměrné hodnoty			
4824	4211	613	12,7 %

Tab. 1 Test dopadu optimalizace na výkon
Zdroj: Vlastní tvorba

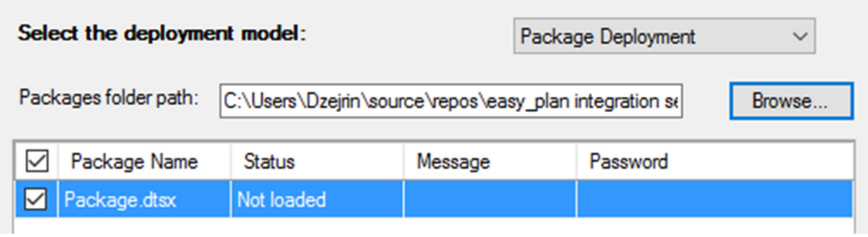
Přiložená tabulka (pozn. vypočtené hodnoty jsou zaokrouhleny) zachycuje data zaznamenaná během čtyř spuštění bez optimalizace a čtyř dalších s ní. Pozitivní vliv je nepopiratelný, v průměru task běžel o 12,7 % kratší dobu.

6.4 Nasazení

Nasazení balíčku SSIS na produkční server je možné spustit přímo skrze Visual Studio nebo SSMS. Obě tyto cesty vyvolají sekvenci kofiguračních oken zvanou Integration Services Deployment Wizard. Pomocí tohoto wizardu, který lze spustit i samostatně, se v několika jednoduchých krocích nakonfiguruje napojení SSIS balíčku na běžící instanci SQL Serveru. Dříve než se s konfigurací začne, musí se na SQL Serveru připravit katalog pro SSIS. K vygenerování tohoto katalogu postačí implicitní nastavení, čímž se vytvoří databáze SSISDB.

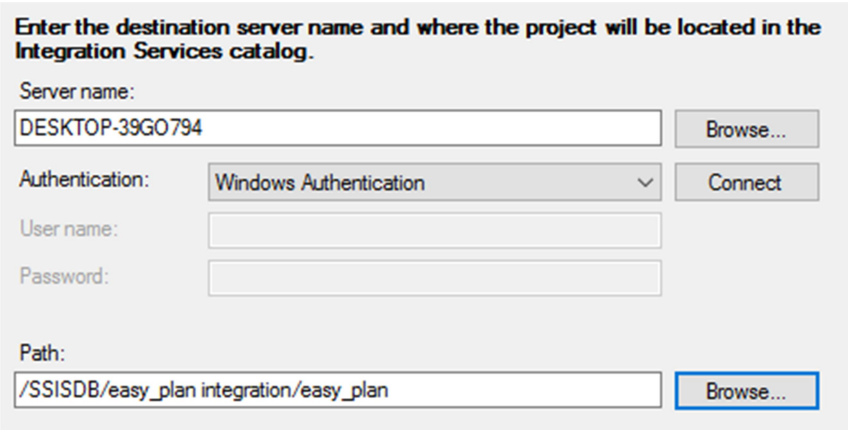
6.4.1 Integration Services Deployment Wizard

Po úspěšném vygenerování SSISDB následuje zmíněný wizard. Předmětem popisu bude samostatné spuštění, neboť otevření přímo z Visual Studia či SSMS předvyplní část informací automaticky, ovšem cílem tohoto popisu je popsat celý průběh. První stránka pouze stručně informuje o pěti následujících krocích. V prvním kroku proběhne výběr zdroje, v tomto případě SSIS balíček, který se otevře ze složky s projektem z Visual Studia.



Obr. 16 Výběr zdroje pro nasazení SSIS
Zdroj: Vlastní tvorba

Druhým krokem je výběr cíle. Zde se vyplní lokální instance SQL Serveru, metoda zabezpečení s případným přihlašovacím jménem a heslem a vybere se nebo vytvoří nový projekt v SSISDB katalogu.



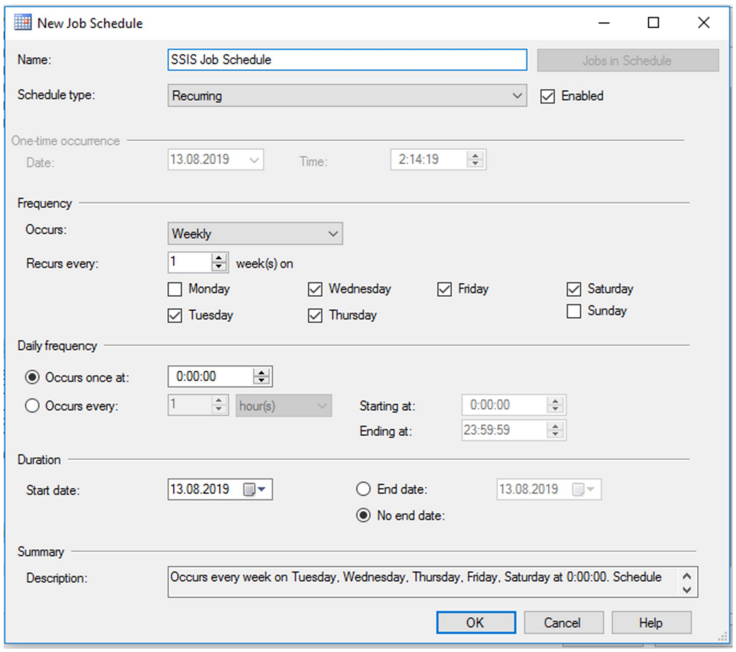
Obr. 17 Výběr cíle pro nasazení SSIS
Zdroj: Vlastní tvorba

Poté již následuje jen rekapitulace. Pakliže vše sedí, projekt se nasadí. V opačném případě se nabízí poslední možnost vrátit se zpět a konfiguraci upravit.

6.4.2 Nastavení plánu spouštění

Pro vytvoření pravidelného plánu spouštění bude využit SQL Server Agent. Nastavení opět proběhne skrze SSMS. Pro agenta se vytvoří nový Job, jenž bude mít

právě jeden Step - spuštění SSIS balíčku. Poté se nastaví plán spouštění, čímž je docíleno opakování dle specifických požadavků.



Obr. 18 Nastavení plánu spouštění SSIS
Zdroj: Vlastní tvorba

7 Shrnutí výsledků

Hlavním cílem práce bylo demonstrovat použití ETL, předtím však bylo nutné seznámit se obecně s problematikou. Pro správné porozumění celé problematiky byly nejprve představeny postupy a paradigmata v obecné podobě, jak je lze aplikovat na většinu dostupných nástrojů. Popsána byla extrakce s ohledem na to, jaké zdroje dat lze použít s primárním zaměřením na databáze, poté transformace a několik nejčastěji řešených kroků v této fázi, dále pak načtení a nakonec ošetření chyb.

Následující část práce přináší letmý vhled mezi dostupné softwarové nástroje. Záměrem bylo představit v první řadě SSIS a teprve poté ostatní možné alternativy. V dnešní době již existuje velké množství nástrojů pro ETL s různými přístupy i cílovými prostředími, ovšem to je výsada teprve několika posledních let. Vývoj v tomto odvětví probíhá dle možností hardwaru a potřeb softwaru. Představená řešení byla seskupena do několika skupin dle jejich určení a jednotlivé skupiny chronologicky uvedeny.

Dále pak následuje praktická ukázka použití. Jedním z hlavních cílů bylo navrhnout novou, strukturálně jednodušší databázi. Ta vychází z původní, ovšem vynechává tabulky, které nejsou nezbytně potřeba, převzaté tabulky pak drobně poupravuje. Jak ze softwarového, tak i z hardwarového hlediska bylo definováno běhové prostředí. Poté konečně přichází na řadu vlastní tvorba balíčku v prostředí SSIS. To se ukázalo jako velmi intuitivní a přímočaré. Jsou demonstrována řešení konkrétních potřeb při transformacích a další postupy aplikované při tvorbě balíčku. Implementována též byla vlastní funkcionalita pro logování. Ve finální fázi je v prostředí SQL Serveru přidán a spuštěn vytvořený balíček včetně nastavení cyklického spuštění na základě časového plánu.

8 Závěr

SQL Server Integration Service představuje poměrně přívětivý a přehledný nástroj na práci s ETL. S jeho pomocí můžeme vytvořit komplexní databázové operace bez nutnosti ovládat dotazovací jazyk SQL nebo psaní programového kódu. Obecně lze hlavní výhody ETL popsat jako snadnou správu, jednoduchost, robustnost.

Výběr nástrojů pro ETL je velký, ovšem především také rozmanitý. Byť existuje spousta nástrojů, ne vždy o nich lze mluvit jako o konkurenci. Zářným příkladem tohoto faktu budiž firma Apache distribuující tři různé nástroje.

Budoucnost výpočetní techniky spočívá v cloudu a nástroje pro ETL na to musí být připraveny. Během posledních let přibývají nová řešení především pro cloudové databáze a SSIS samozřejmě není vyjímkou, neboť spolu s růstem cloudové platformy Microsoft Azure, roste i její podpora v rámci SSIS.

9 Použitá literatura

[1] The Future of the DBMS Market Is Cloud - Donald Feinberg, Merv Adrian, Adam Ronthal. Gartner [online]. Copyright © 2019 Gartner, Inc. and [cit. 12.08.2019]. Dostupné z: https://www.gartner.com/document/3941821

[2] Extract, transform, and load (ETL) | Microsoft Docs. [online]. Dostupné z: https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl

[3] SSIS and Data Sources - TechNet Articles - United States (English) - TechNet Wiki. [online]. Copyright © 2015 Microsoft Corporation. All rights reserved. [cit. 26.4.2019]. Dostupné z: https://social.technet.microsoft.com/wiki/contents/articles/1947.ssis-and-data-sources.aspx

[4] SSIS Components. KingswaySoft - Data Integration Solutions [online]. Copyright © 2011. [cit. 28.4.2019] Dostupné z: http://www.kingswaysoft.com/ssis-components

[5] Most popular databases in 2018 according to StackOverflow survey. EverSQL - SQL Query Optimization Tool Online [online]. Copyright ©2019 EverSQL.com. [cit. 23.5.2019]. Dostupné z: https://www.eversql.com/most-popular-databases-in-2018-according-to-stackoverflow-survey/

[6] Artificial Intelligence tunes Azure SQL Databases | Blog | Microsoft Azure. Object moved [online]. Copyright © 2019 Microsoft. [cit. 23.5.2019]. Dostupné z: https://azure.microsoft.com/en-us/blog/artificial-intelligence-tunes-azure-sql-databases/

[7] ZappySys [online]. USA. [cit. 28.4.2019]. <https://www.zappysys.com/>

[8] SQLite Documentation. [online]. Dostupné z: https://www.sqlite.org/docs.html

[9] MySQL :: MySQL Documentation. MySQL :: Developer Zone [online]. Copyright © 2019, Oracle Corporation and [cit. 14.07.2019]. Dostupné z: https://dev.mysql.com/doc/

[10] STANEK, William R. Microsoft SQL Server 2012: kapesní rádce administrátora. Brno: Computer Press, 2013. Microsoft (Computer Press). ISBN 978-80-251-3797-0.

[11] Check Database Integrity Task (Maintenance Plan) - SQL Server | Microsoft Docs. [online]. Dostupné z: https://docs.microsoft.com/en-us/sql/relational-databases/maintenance-plans/check-database-integrity-task-maintenance-plan?view=sql-server-2017

[12] SQL Server Integration Services - SQL Server Integration Services (SSIS) | Microsoft Docs. [online]. Dostupné z: https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-2017

[13] INITIAL THROUGHTS ON ORACLE DATA INTEGRATOR 12C. OptimalBI | Making Information Visible [online]. Dostupné z: https://optimalbi.com/blog/2014/10/31/initial-throughts-on-oracle-data-integrator-12c

[14] Delete Duplicate Rows in SQL Server From a Table. Dot Net Tricks : Learn to Code, Prepare for Interviews, and Get Hired [online]. Copyright © 2019 Dot Net Tricks Innovation Pvt. Ltd. All rights Reserved. The course names and logos are the trademarks of their respective owners. [cit. 13.08.2019]. Dostupné z: https://www.dotnettricks.com/learn/sqlserver/remove-duplicate-records-from-a-table-in-sql-server

[15] Apache NiFi User Guide. Apache NiFi [online]. Dostupné z: https://nifi.apache.org/docs/nifi-docs/html/user-guide.html

[16] What is SQL Server? - Definition from Techopedia. Techopedia - Where IT and Business Meet [online]. Copyright © 2019 Techopedia Inc. [cit. 12.08.2019]. Dostupné z: https://www.techopedia.com/definition/1243/sql-server

[17] Porovnání nabídek produktů sady Visual Studio | Visual Studio. Visual Studio IDE, Code Editor, Azure DevOps, & App Center - Visual Studio [online]. Dostupné z: https://visualstudio.microsoft.com/cs/vs/compare/?rr=https%3A%2F%2FwwF.quora.com%2FWhat-are-the-differences-between-Microsoft-Visual-Studio-Community-and-Microsoft-Visual-Studio-Professional

[18] ETL Tools: A Modern List | Alooma. Alooma | Enterprise Data Pipeline Platform [online]. Copyright © 2019 Alooma, Inc. [cit. 10.11.2019]. Dostupné z: https://www.alooma.com/blog/etl-tools-modern-list

[19] ETL Tools: An Overview of ETL Technologies for 2019 and Beyond | Xplenty. [online]. Copyright © Xplenty Ltd. All rights reserved. [cit. 10.11.2019]. Dostupné z: https://www.xplenty.com/blog/etl-tools-an-overview-of-etl-technologies-for-2019-and-beyond/

[20] LinkedIn engineers spin out to launch 'Kafka' startup Confluent | Fortune. Fortune - Fortune 500 Daily & Breaking Business News [online]. Copyright © 2019 Fortune Media IP Limited. All Rights Reserved. Use of this site constitutes acceptance of our [cit. 10.11.2019]. Dostupné z: https://fortune.com/2014/11/06/linkedin-kafka-confluent/

10 Přílohy

1) Podklad k zadání
2) CD s SSIS balíčkem a databázovými soubory

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2018/2019

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Šimůnek Jiří	Lipnice 76, Dvůr Králové nad Labem - Lipnice	I1500439

TÉMA ČESKY:

Procesy ETL

TÉMA ANGLICKY:

ETL Processes

VEDOUCÍ PRÁCE:

Ing. Barbora Tesařová, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cílem práce je seznámit se s procesem ETL, který slouží pro přenášení, transformaci, čištění dat z datových zdrojů do datových skladů a datových tržišť z různých zdrojů, analýzovat specializované nástroje pro ETL a použití ETL procesů v Integration Services a MS SQL Server na reálných datech.

Osnova:
-ETL procesy
  -popis jak fungují ETL nástroje
  -rozbor jednotlivých částí procesu
  -řešení problematických situací
-Přehled dostupných ETL nástrojů
  -rozdělení do logických skupin
  -výčet běžných a oblíbených SW řešení se zaměření na silné a slabé stránky
-SQL Server Integration Services
  -ukázka použití SSIS
  -popis funkcionality a uživatelské prostředí
  -porovnání s ostatními řešeními
-Použití v praxi
  -popis zdrojového datového skladu
  -návrh a realizace cílové databáze
  -tvorba a nasazení ETL balíčku

SEZNAM DOPORUČENÉ LITERATURY:

Data Warehouse ETL Toolkit, Ralph Kimball
Microsoft SQL Server 2012 - Kapesní rádce administrátora, William R. Stanek

Podpis studenta: ..... **Datum:** 17.10.2018

Podpis vedoucího práce: ..... **Datum:** 17.10.2018