Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informatiky a kvantitativních metod

Návrh datového skladu pro analýzy databázové aplikace Kalorické tabulky

Bakalářská práce

Autor: Juraj Jankovič

Studijní obor: Informační management

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

Hradec Králové 04. 2020

Prohlášení:

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

V Hradci králové dne 25.4. 2020 Juraj Jankovič

Poděkování:

Děkuji vedoucí práce Barboře Tesařové Ph.D. za metodické vedení práce a cenné rady, jež mi napomohly k úspěšnému dokončení této bakalářské práce. Dále také chci poděkovat společnosti Dine4Fit za poskytnutí dat a umožnění práce na tomto tématu.

Anotace

V této bakalářské práci budou detailně rozebrány pojmy z oblasti Business Intelligence. Bude zde probírána hlavně tématika databází, okolo kterých se celé Business Intelligence orientuje. Teoretická část práce je primárně zaměřena na popis datových skladů, jejich komponent a technologií, jež se v této oblasti využívají. Konkrétně se jedná o další pojmy jako datová tržiště, OLTP, OLAP technologie, ETL procesy, dimenzionální modelování, či historizace dat. V praktické části potom dojde k představení aplikace Kalorické tabulky od společnosti Dine4Fit, a.s., což je jedna z oblíbených aplikací na monitorování denního režimu stravování jejích uživatelů. V první řadě bude rozebrána jejich aktuální databáze, a k ní poté bude navržen multidimenzionální datový sklad, včetně potřebných ETL procesů, který bude sloužit k účelům provádění analýz nad získanými daty.

Annotation

In this bachelor thesis, terms from Business Intelligence will be analyzed in detail. The main topic of the Business Intelligence are databases. The theoretical part is primarily focused on the description of data warehouses, their components and technologies that are used in this area. Specifically, there are other terms such as data markets, OLTP, OLAP technology, ETL processes, dimensional modeling, or data historization. The practical part will introduce the application of Kalorické tabulky from Dine4Fit, a.s., which is one of the most popular applications for monitoring the daily diet regime of its users. First of all, their current database will be analyzed and then a multidimensional data warehouse, including the necessary ETL processes, will be designed for analysis of acquired data.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Tato bakalářská práce se zabývá problematikou Business Intelligence, čili správou, a prací s daty využitelných k fungování, budoucí prosperitě a k větším ziskům firmy, jež s nimi nakládá. Shromažďování dat o uživatelích je v posledních letech jedním z nejzásadnějších a nejdůležitějších procesů, jelikož zjištěné informace z těchto dat mnohdykrát určují budoucí strategii podniku. Ať už se jedná o národní, či nadnárodní korporace, žádné by nemohly fungovat, aniž by neměly svou databázi, ve které mají ucelený přehled o všem, co potřebují.

K ukládání faktů slouží právě databáze, které za použití správných technologií, principů a postupů dokáží sama o sobě nicneříkající data transformovat na užitečné a použitelné informace. Jelikož data se v průběhu času mění, je potřeba udržovat i záznamy, které již nejsou aktuální, a to je úlohou datového skladu, o kterém se tu bude nejvíce mluvit.

1.1 Cíl práce

Cílem této práce je navrhnout a implementovat datový sklad, jež by sloužil k provádění potřebných analýz nad shromážděnými daty z aplikace Kalorické tabulky. Umožnil by tak pracovníkům firmy mít lepší přehled o datech jejich zákazníků a zjednodušil by tak tvorbu reportů.

2 Business Intelligence

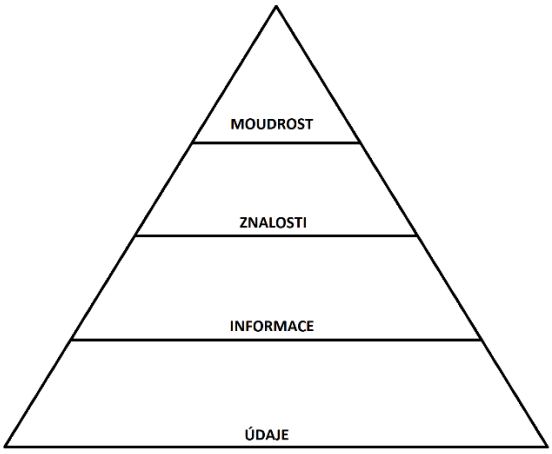
Business Intelligence je jedním z nejvíce se rozšiřujících pojmů v oddělení podnikové informatiky i informačních technologií jako takových. Tento pojem zahrnuje několik kroků, které společně vedou k získávání užitečných informací, jež lze využít k větší prosperitě podniku. Díky nim může firma dosáhnout lepšího obratu, dokáže snižovat náklady na výrobu a provoz podniku, umožňuje lepší flexibilitu na trhu. Zkrátka díky jednoduššímu a lepšímu rozhodování díky informacím dokáže firma využít lépe svůj potenciál. (L. Lacko, 2009)

Jednou v trefných definic pojmu business intelligence může být tato od: (L. Gála, 2009)

„Business inteligence je sada procesů, know-how, aplikací a technologií, jejichž cílem je účinně a účelně podporovat řídící aktivity ve firmě. Podporují analytické plánovací a rozhodovací činnosti organizací na všech úrovních a ve všech oblastech podnikového řízení, tj. prodeje, nákupu, marketingu, finančního řízení, controllingu, majetku, řízení lidských zdrojů, výroby a dalších.“ [1]

Jak je vidno z této definice, pojem zahrnuje opravdu širokou škálu procesů a technologií, jež vedou k vytouženému výsledku. Vše začíná sběrem dat z externích i interních firemních systémů, následné uložení do velké, zformátované a jednotné databáze, čili datového skladu, a poté práci analytiků, manažerů a jiných koncových uživatelů – zaměstnanců firmy, jež data využívají. Jelikož dnešní doba je velice uspěchaná, je i ve firmách vyžadována co nejrychlejší adaptace na jakékoliv nečekané změny jako třeba měnící se trh, různé potřeby zákazníků nebo vliv konkurence. Aby bylo možné všechny tyto faktory sledovat, je potřeba o nich mít správné informace. (L. Lacko, 2009)

Toho lze dosáhnout posloupností úprav, nimiž data musí projít, aby plnila svůj účel. Základním prvkem jsou zde data (údaje), která jsou posbírána z několika zdrojů. Sama o sobě tato data nic neříkají, ale po přidání určitých souvislostí nebo spojení několika dat jsme schopni vytvořit užitečnou informaci. Ta je poté vlivem lidského přemýšlení přetvořena na znalost. Z té může být po zobecnění ucelená moudrost, kterou lze pak v praxi použít. Tyto přeměny nám zajišťují technologie jako ETL procesy, datové sklady, OLAP analýzy, datové dolování, reporting.



Obrázek 1 - Hierarchie informačních úrovní

Zdroj: vlastní zpracování dle L. Lacka 2003

3 Databáze

Databáze je samotným pilířem pojmu business intelligence. (Oracle, 2020) Databáze je úložiště jakýchkoliv strukturovaných dat, které se skládá z řádků a sloupců. V dnešní době jsou nejpoužívanější dva typy: operační databáze a analytická databáze. Každá je postavena jiným způsobem a umožňuje mít uložená data jiným stylem. Přínosů a funkcí databází dnes využívá již skoro každá firma. Zaměstnavatelé uchovávají data o svých zaměstnancích, stavu skladovaných zásob, o výrobcích, údaje o zákaznících, dodavatelích nebo stavech prodejů. Velké společnosti uchovávají data o svých jednotlivých odděleních firmy. Dle (comparebusinessproducts, 2010) nejrozsáhlejší databáze však potřebují pro své podnikání internetové e-shopy jako například Amazon, mobilní operátoři, bankovní společnosti, databáze státních orgánů nebo samotný Google a YouTube. Údaje v databázích však samy o sobě přínosem nejsou. Většinou jsou dost zmatečná a běžný uživatel, jenž nezná souvislosti, z nich toho moc nevyčte. K práci s daty slouží různé analytické nástroje, které využívají proškolení a zkušení pracovníci, kteří pomocí analýz a reportů vytvářejí užitečné informace. Na základě těchto informací mohou poté manažeři firmy podnikat další kroky a budovat novou strategii podniku. Bez databází bychom podnikání, zábavu, sociální média, ani život neznali tak, jaké dnes je.

3.1 Operační databáze

První představenou databází bude operační. Jak tvrdí (L. Lacko, 2003), dříve se v podnikách používaly na úschovu dat pouze operační OLTP databáze (On-line Transaction Processing). Tyto databáze poskytují možnost vykonávání velkého množství online transakcí. Každodenně jsou do nich nahrávána data z různých informačních systémů a také s daty z těchto databází je zároveň každodenně vykonáváno velké množství operací. Proto jsou velice důležité pro společnosti jako jsou banky, mobilní operátoři, nákupní střediska, pojišťovny, a různé nadnárodní i národní společnosti. OLTP systémy jsou však navržené pro rychlé provedení transakce, ale nejsou uzpůsobeny k provádění složitého dotazování za účelem získání informací k tvorbě reportů a analýz.

3.1.1 Nevýhody OLTP systémů dle (L. Lacko, 2019)

Normalizované tabulky: Nejsou vhodné pro složité analýzy, které tak zaberou spoustu času, jelikož se málokdo v tabulkách vyzná.

Decentralizace OLTP systémů: Různé pobočky firmy, které mohou být v různých částech republiky, či v případě nadnárodních korporací i v jiných státech a kontinentech, mají každá svou vlastní transakční databázi a tím pádem i odlišná data pro své podnikové či geografické oddělení. Nelze tak úplně provádět ucelené analýzy nad daty celé firmy.

Nedochází k historizaci dat: Bohužel tyto systémy většinou mají nedostatečnou kapacitu dat pro jejich uchovávání. Následkem je poté to, že postupně mizí starší data a jsou nahrazována novějšími. Tím pádem nelze provádět komplexní analýzy a nelze je porovnávat například s analýzami přechozích časových období.

Nestrukturovaná data: Data v několika databázích, ale i pouze v jedné jediné mohou být různých formátech. Mohou mít odlišnou délku nebo tvar (např. mobilní číslo). Navíc může docházet ke vzniku duplicitních údajů, jelikož nikdo neeviduje a ani ho nezajímá, zda již stejný záznam v databázi jednou není.

Vzhledem k tomu, že firmy právě potřebují veškeré funkce, jež jim OLTP databázové systémy bohužel nenabízí a nepodporují je, přechází se na databáze relační, které jsou budovány za plněním právě tohoto účelu.

3.1.2 Relační databáze

Dle (Oracle, 2020) se jedná o databázi, která je používána již od sedmdesátých let dvacátého století. Jedná se o specifickou podmnožinu OLTP databáze, která je vystavěna dle určitých pravidel a standardů. V databázi na sebe jednotlivé tabulky navazují a odkazují na sebe za pomoci klíčů. ID je jedinečný identifikační kód a ve většině případů se za klíč uvádí právě on. Občas je klíč i kombinací ID a jiného sloupce. Relační databáze se řídí předem stanovenými pravidly, aby nedocházelo k ukládání nevhodně zformátovaných dat, či ke vzniku duplicitních záznamů. Ty jsou ohlídány právě jedinečným klíčem, a tak nemůže existovat v databázi nějaký řádek, či jemu podobný řádek vícekrát. Při vkládání záznamu jsou s ním vždy přidány i hodnoty, odkdy je platný a v případě jeho ukončení se vkládá i datum, kdy jeho platnost vypršela. Pokud je záznam se stejným klíčem použit v databázi vícekrát, znamená to, že nějaká z jeho hodnot byla upravena a za pomocí historizace se udržuje i původní záznam o tom, jaké měl tento řádek hodnoty dříve.

Jak (L. Lacko, 2003) říká, zavaděné údaje se do relační databáze musí zapisovat s určitými pravidly. Využívá se při tom procesu normalizace. Jedná se tedy o odstranění redundancí, což zajišťuje vyšší výkonost. Čím vyšší je normativnost tabulky, tím lépe by se s ní mělo pracovat. Nejvyšší formou je 3NF, tedy třetí normální forma, nejnižší je 1NF.

3.1.3 Normativní formy

Podle společnosti (Oracle, 2020), musí tabulky pro splnění normativní formy splňovat normy předchozích tabulek. Čili pro třetí tabulku to znamená, že bude splňovat zároveň i první a druhou normativní formu.

První normativní forma (1NF)

První normativní forma znamená, že data jsou atomická. Čili hodnoty ve všech sloupcích jsou dále nedělitelné, mají pouze jednu hodnotu, která je jednoznačná.

Druhá normativní forma (2NF)

Druhou normativní formu lze aplikovat pouze na tabulky, kde je více než jeden primární klíč. V případě, že se primární klíč skládá ze dvou a více sloupců, musejí všechny ostatní hodnoty ve sloupcích záviset právě na této kombinaci primárních klíčů, a ne na pouze jednom z nich. Pokud nějaký takový případ nastane, je nejlepší tabulku rozdělit na dvě samostatné, které na sebe poté odkazují za pomoci cizího klíče.

Třetí normativní forma (3NF)

V třetí normativní formě nesmí být neklíčové sloupce provázány mezi sebou. V případě, že tomu tak je, je vhodné ponechat v tabulce pouze jeden z těchto sloupců a na ten druhý, jež je v samostatné tabulce, ukazovat opět pomocí cizího klíče.

3.1.4 SQL

Jak píší na svých stránkách (Oracle, 2020), SQL je v dnešní době prozatím nejrozšířenějším dotazovacím jazykem. Jedná se o zkratku pro Structured Query Language. Používá se k provádění dotazů nad relačními databázemi. Umožňuje nejen zobrazování dat, ale také manipulaci s nimi. Umožňuje vkládat nové záznamy, upravovat i mazat stávající.

Soupis funkcí tohoto jazyka:

- DDL – data definition language – vytvářejí a upravují strukturu databáze a tabulek (CREATE, DROP, ALTER)
- DML – data manipulation language – jsou k zobrazení, ukládání, modifikaci a mazání dat (SELECT, INSERD, UPDATE, DELETE)
- DCL – data control language – slouží pro správu uživatelských rolí
- TCL – transaction control language – určené ke správě databázových transakcí

Pro běžné uživatele jsou nejdůležitější znát hlavně DDL a DML příkazy, jelikož je hojně využívají každý den.

3.1.5 Multidimenzionální databáze

Multidimenzionální databáze (Pedersen, 2009) jsou optimalizované k tomu, aby byly možné ukládat kromě běžných nenormalizovaných dat i agregovaná data. Jsou zde tedy nejen data v základní formě, ale i také jejich různé součty, průměry, výseče dat a další. Agregovaná data jsou ukládána v kontextu navržených dimenzí dle jejich stavby. Na ně jsou poté navázány reportovací nástroje, které umožňují analytikům z dat vyvést nějaké závěry. Pro vykonávání analýz jsou tyto databáze nejvhodnějšími, největším záporem jsou však vysoké nároky na kapacitu.

3.2 Technologie OLAP

Systémy OLAP slouží ke zpracování dat z databází do formy pochopitelné koncovými uživateli. Jednotlivá písmena jsou zkratkou pojmu Online Analytical Processing. Data získaná z OLTP systémů se transformují pomocí ETL procesů a následně ukládají do datových skladů. Z nich se poté čerpají do analytického databázového modelu, který je často multidimenzionální, a na kterém je tato technologie založena. Ten je uzpůsoben podnikatelským cílům a směrům konkrétní firmy. Na základě získaných informací z OLAP systémů, mají uživatelé jako

manažeři nebo business analytici lepší přehled o stavu podniku. Multidimenzionální databázový model je reprezentován jako kostka, o níž bude zmínka. Na rozdíl od OLTP systémů se do dat v těchto systémech pouze nahlíží, nijak nejsou upravovány nebo mazány. Tyto dotazy bývají složité, často ad-hoc a využívá se v nich joinování, agregací nebo filtrování, a tak jsou také z časového hlediska velice náročné. Samotné OLAP databázové systémy udržují v porovnání s OLTP systémy daleko větší množství dat. Existuje několik typů získávání dat pro OLAP analýzy, o kterých se bude mluvit později. (L. Lacko, 2003)

3.3 Nevhodnost OLTP databází pro analýzy

Vzhledem k tomu, že v OLTP databázích jsou data normalizována – většinou 3.NF (L. Lacko, 2003), dosahují tyto databázové systémy spíše vysokých výkonů během transakčních operací, nikoliv však při složitých analýzách, které hodně zatěžují výpočetní kapacitu procesorů. Analýzy také vyžadují pro jednodušší výpočty multidimenzionální schémata hvězdy nebo vločky a nenormalizované tabulky. Největším problémem je ale to, že OLTP databáze jsou decentralizované a neumožňují tak komplexní analýzu nad všemi daty najednou. Většina podniků má několik zdrojových databází a výsledek analýzy pouze jedné z nich by nebyl moc platný. Data se tak musí nejprve integrovat, což je velice časově náročné. Občas se ani nepovede takovou integraci provést. Z technického hlediska je OLAP analýza nad OLTP proveditelná, ale je to velice nevhodné řešení.

Odůvodnění nevhodnosti dat:

- Nelze jednoduše najít příčiny ani vysvětlení problémů,
- těžké vyhledávání závislostí subjektů,
- dlouhý čas výpočetního výkonu v transakční databázi,
- transakční databáze neprovádí historizaci,
- různorodá struktura dat,
- příprava dat vyžaduje hodně času.

4 Datový sklad

„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.“ [1]

Takto definici datového skladu, jež je použita v knize L. Lacka Business Intelligence v SQL Serveru 2008, vyřkl Bill Inmon, jeden z největších průkopníků a osobností v tomto počítačovém oboru.

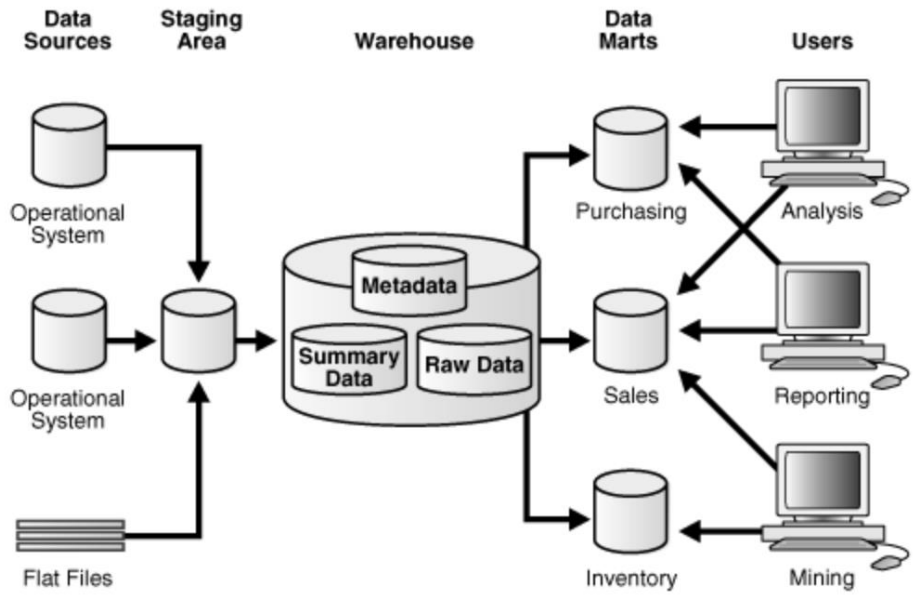
Vysvětlení pojmů z definice:

Subjektová orientace – data se zapisují do datového skladu tak, aby byla vázána na předmět-subjekt (zaměstnanec, zákazník, produkt, nemovitost), nikoliv na nějaký proces, jež se ve firmě odehrává. Například data aplikací pro prodeje firmy nebo pro fakturaci.

Integrovanost – data se zde nesmí vyskytovat vícekrát (duplicitně), musí mít všechna stejný formát a stejné veličiny.

Časová proměnlivost – data jež se ukládají do datového skladu mají všechna na rozdíl od operačních databází také vždy časový údaj, díky čemuž spadá do nějakého časového období a díky tomu lze na tomto základě například simulovat předchozí stav zdrojových OLTP systémů v určitém období.

Neměnnost dat – jelikož datový sklad je takový „archiv“, data se do něj ukládají, čtou se, ale nijak nedochází k jejich mazání, či jakékoliv jiné modifikaci.



Obrázek 2 - Datový sklad [19]

Datový sklad je tedy formou databáze, jež je orientována podle určitých pravidel, a jež schraňuje veškerá firemní data z několika různých druhů zdrojů (Oracle, 2020). Nejedná se pouze o samotnou databázi, ale i souhrn všech nástrojů a technologií, jež umožňují skladování, úpravu a práci s daty. Slouží výhradně k provádění dotazů a analýz. Jednou z hlavních výhod takovéhoto skladu je, že uchovává historizovaná data. Jdou zde tedy dohledat data několik let dozadu. Stává se tak centrálním zdrojem dat pro celou firmu a díky nim může být neuvěřitelným přínosem. Umožňuje uživatelům vykonávat ad-hoc dotazy, neboli dotazy, jež nebyly předem definované a podporované. Datový sklad může být postaven na základě relačního modelu nebo multidimenzionálního.

Prvky datového skladu (dle Oracle, 2020):

- Procesy pro extrakci, transformaci a načítání dat
- Statistické analýzy, vykazování a dolování dat
- Analytické nástroje pro prezentaci dat v podniku

V tabulce níže lze porovnat rozdíl mezi OLTP databází a datovým skladem.

	Datový sklad	Systém OLTP
Zatížení	Přizpůsobuje dotazy ad hoc a analýzu dat	Podporuje pouze předdefinované operace
Úprava dat	Pravidelná automatická aktualizace dat	Aktualizace koncovými uživateli vydávajícími příkazy
Návrh schématu	Používá částečně denormalizovaná schémata	Používá plně normalizovaná schémata
Skenování dat	Zahrnuje tisíce až miliony řádků	Přistupuje pouze k několika záznamům najednou
Historická data	Ukládá data po několik let	Ukládá data pouze po dobu týdnů nebo měsíců

Tabulka 1 - Porovnání datového skladu a OLTP systému

Zdroj: vlastní tvorba dle Oracle

4.1 Datové tržiště

Jedná se o podmnožinu datového skladu, která je vytvořena a specializuje se na plnění funkcí určité složky/oddělení firmy. Podmnožinou jsou většinou celé tabulky, mohou to být ale i pouze vybrané sloupce daných tabulek. Například se tvoří datové trhy pro potřeby oddělení účetnictví, personalistiky, distribuce, výroby, prodeje a ta mají oprávnění pracovat pouze s tímto poskytnutým datovým trhem. Nemusí se složitě dotazovat v celém datovém skladu a mají tak veškerá potřebná data na jednom místě, což je pro ně přehlednější a také díky kratší časové odezvě lepší. Datové tržiště není pouze kopií dat z datového skladu, jelikož může obsahovat svou vlastní tabulku faktů. Ta na sebe navazuje ostatní tabulky dimenzí a mají vždy schéma sněhové vločky nebo hvězdy.

Datové tržiště	Datový sklad
Data jsou soustředěna na 1 odvětví (finance, prodej, marketing)	Data ze všech firemních odvětví
Data zde jsou většinou detailní, ale mohou být již osekána pouze na potřebné údaje	Uchovává všechna data, která jsou detailní
Vybudován pro hvězdicový dimenzionální model	Sklad nemusí být multidimenzionální databáze, pouze jeho data se používají do dimenzionálních modelů

Tabulka 2 - Porovnání datového skladu a datového tržiště [20]

4.2 Metody vzniku datového skladu

Datový sklad lze budovat více způsoby, avšak nejznámější a nejhojněji využívanými jsou metoda „velkého třesku“ a přírůstková metoda. Jak je řečeno v knize (L. Lacka, 2003), právě návrh a koncepce datového skladu představují největší procento hodnoty takového skladu. Hned za nimi jsou výdaje za hardware a software. Správně navržený datový sklad je základem, jelikož se špatným návrhem takový sklad není pro podnik vůbec přínosem, pouze vyhozenou investicí.

4.2.1 Metoda „velkého třesku“

Při této metodě se datový sklad vytvoří jako celek a následně je dodán zadavateli práce. Metoda „velkého třesku“ nabízí jednu jedinou výhodu. Tou je možnost navržení a vypracování celého datového skladu ještě před tím, než se zrealizuje. Tato výhoda je zároveň také velkou nevýhodou. Jelikož se datový sklad začne budovat na základě určitých dosavadních znalostí, technologií a aktuálního stavu podnikových dat, může vzniknout problém v tom, že se jakákoliv z těchto proměnných během budování změní. Vzhledem k tomu, že vývoj technologií jde stále mílovým krokem kupředu, prakticky se výskytu tomuto problému nelze vyvarovat.

4.2.2 Přírůstková metoda

Přírůstková metoda nabízí mnohem dynamičtější řešení. Datový sklad je budován po jednotlivých etapách, které postupně vytvářejí celek. Jednotlivé části jsou ale i samy o sobě funkční. Výhodou je, že zadavatel práce během krátké doby již vidí nějaké výsledky práce. Většinou se začíná od vytvoření pár datových tržišť, která se poté v praxi otestují a zjistí se, zda fungují správně. Následně jsou doplňovány další a další části, až se vytvoří celý datový sklad.

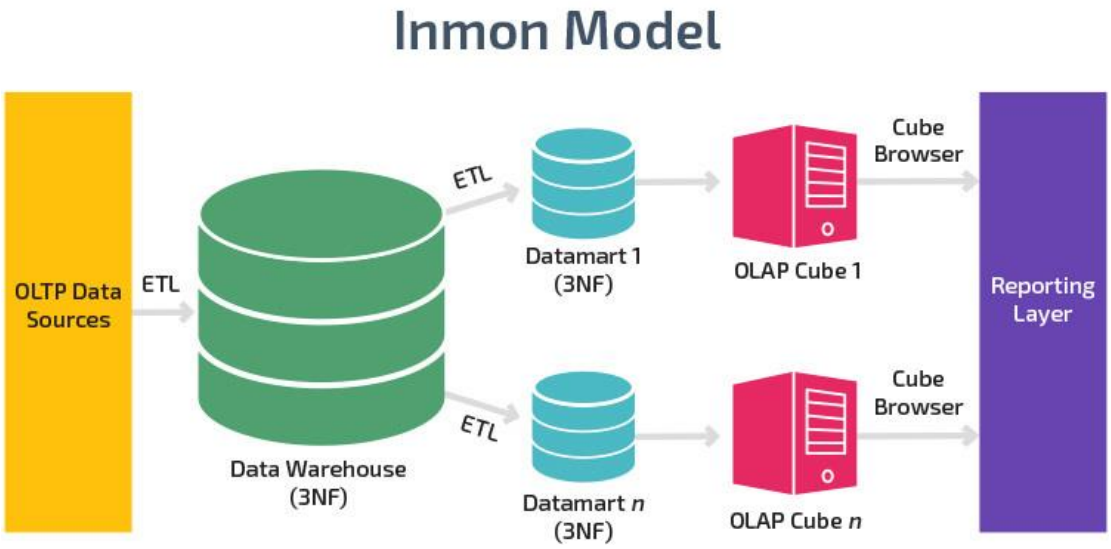
Tento přístup k budování tak umožňuje projekt v průběhu času přizpůsobovat novým, doplňujícím, či opravným požadavkům nebo novějším technologiím.

4.3 Vrstvy datového skladu

Při budování datového skladu přírůstkovou metodou je možné postupovat dvěma způsoby. Buď lze zvolit metodu směrem „shora dolů“ (Top Down) nebo „zdola nahoru“ (Bottom Up).

4.3.1 Metoda Top Down

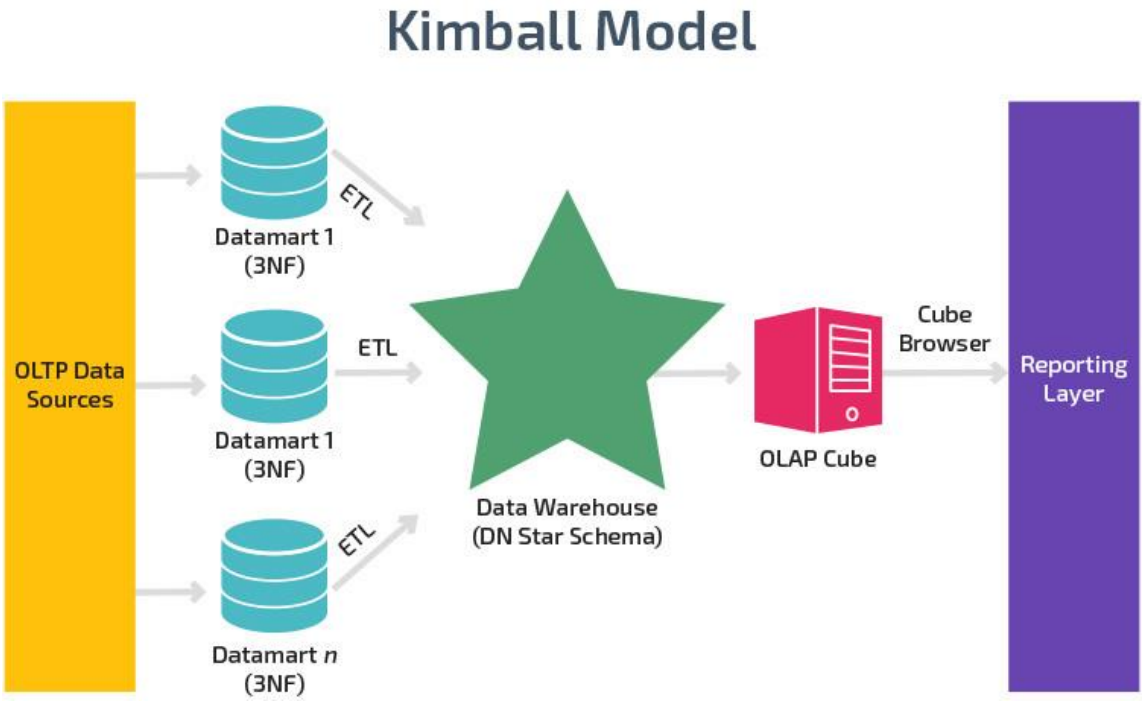
Podle (L. Lacka, 2003), při této metodě se vytvoří celkový model datového skladu. Ten je postupně doplňován po částech. Dochází tedy k vybudování základního celistvého skladu a pak teprve menších oblastí – datových tržišť, které jsou zaměřené na určitou část podniku. Tuto strukturu je jednodušší udržovat z hlediska přehlednosti a lépe se k ní přidávají nová potřebná datová tržiště. Tento princip více směřuje k zisku z jednotlivých odvětví. Prvotní investice jsou zde ale vcelku vysoké a trvá, než se ukáže nějaký pokrok. Tuto metodu prosazoval Bill Inmon, americký počítačový vědec a jeden ze dvou hlavních průkopníků ve světě datových skladů.



Obrázek 3 - Inmonův model datového skladu [21]

4.3.2 Metoda Bottom Up

U této metody je více kladen důraz na samotná data (computerweekly, 2012). Nejdříve jsou vybudovány datové trhy, které sbírají data jednotlivých oddělení podniku a až poté na ně navazuje celý datový sklad. Výhodou je, že tato metoda poskytuje rychlejší řešení budování datového skladu. Prvotní náklady jsou zde nižší, ale s každým novým tržištěm stojí tento projekt více peněz. Dále je tento model také méně přehlednější. Co se týče toku dat, je zde poloha datového skladu a datových tržišť prohozena. Tento postup byl typický zase pro Ralpha Kimballa.



Obrázek 4 - Kimballův model datového skladu [21]

4.4 Datový tok skrze řešení datových skladů

Data jsou v prvopočátku zaváděna do zdrojových systémů pomocí běžných uživatelů. Zdrojová data mohou být uložena v klasických transakčních databázích, v excelovských tabulkách nebo ve formě textových souborů, takzvaných „flat files“ a další. Pokud je ve zdrojových transakčních systémech velké množství dat, a ještě nedošlo k automatickému plánovanému přesunu dat do datového skladu, mohou být přenesena do DSA. DSA je dočasné úložiště dat, které slouží k ulevení OLTP systémů. Jednou za určité časové období jsou data ze zdrojových systémů nebo z DSA převáděna buď do datového skladu nebo datových tržišť, záleží na tom, jak je sklad navržen. To zajišťují ETL procesy.

5 ETL procesy

ETL procesy jsou zkratkou pro tři anglické výrazy – funkcionality. Zkratka označuje názvy tří procesů, jež s daty probíhají: E – Extract, T – Transform, L – Load. V prvním kroku dochází k extrakci dat ze zdrojových systémů, následně podstoupí data nějakou transformaci a jsou upravena do požadované struktury a v poslední fázi jsou data nahrána do datového skladu (Guru99, 2020). Hlavním důvodem je, že data jsou uložena na několika firemních databázových serverech a nejsou centralizována. Zároveň nejsou homogenizována, a tak jsou velice nepřehledná pro běžného uživatele, který by se jimi chtěl probírat. Pro takovéto procesy jsou využity nástroje například od firem Oracle, Teradata, SAP, IBM nebo Informatica.

5.1 Extract

V počátečním kroku se data vyextrahují ze zdrojových systémů (Vithal S., 2018). To mohou být různé interní i externí relační databázové systémy, tzv. „flat files“, což jsou textové soubory s daty, XML dokumenty a další. Data z těchto zdrojů jsou tak extrahována a uložena do přechodné staging area, kde se podrobí úpravám. Během extrakce by neměla být ovlivněna výkonnost zdrojových systémů, proto se také data přesouvají do stageového úložiště. Zdrojové systémy potřebují běžet s co největší výkonností, jelikož v reálném čase mají velké množství transakcí, které musejí vykonat.

Datová extrakce rozlišuje dvě metody:

- Plná extrakce – extrakce celé tabulky v takové formě, v jaké se ve zdrojovém systému nachází
- Částečná přírůstková extrakce – data pro extrakci jsou vybrána jen částečně, ta co byla již dříve extrahována ze zdrojové tabulky jsou v tomto případě vynechána

Dle informací uváděných na (Guru99, 2020), během procesu extrakce se nad daty provádí několik úkonů:

- Porovnání záznamů se zdrojovými daty
- Zajištění, že se nenačtou nechtěná data/spamy
- Kontrola datového typu
- Odstranění duplicitních dat a fragmentů
- Kontrola správnosti klíčů (primárních, cizích)

5.2 Transform

Po nahrání dat do stageového úložiště dochází k jejich úpravě. Toto je jedním z nejdůležitějších kroků (R. Kimball, 2002), jelikož právě zde data dostávají svou kvalitu. Mezi daty získanými ze zdrojových systémů se mohou vyskytnout taková, která nepotřebují být nijak transformována. Takováto data se nazývají „pass through data“ nebo „direct move“ (Guru99, 2020). Ostatní data jsou při transformaci podrobena vyfiltrování nepotřebných atributů, očištění dat, sjednocením měrných jednotek, dalším požadovaným úpravám jako spojováním tabulek nebo výběr, či rozdělení některých potřebných sloupců. Provádí se různé abecední nebo numerické řazení. Dále také nahrazením nullových hodnot konkrétními čísly nebo textovými řetězci. Nullové (prázdné) hodnoty v datovém skladu nesmí být. Dále také sjednocení číselných i textových formátů, kupříkladu datumů, telefonních čísel, čísel bankovních účtů a karet nebo úprava popisu pohlaví z muž a žena na ‚M‘ a ‚Ž‘. V tomto kroku se také přidává datům časový údaj, aby bylo možno data v datovém skladu poté dobře vyhledávat.

Největší překážky a problém s vytvořením unikátních dat se dle (Guru99, 2020) může vyskytovat v těchto případech:

- Jméno člověka může více variant (Petr, Péťa, Peter)
- Názvy společností se mohou lišit (Tyco Electronics, Tyco Electronics EC Trutnov s.r.o., Tyco Trutnov)
- Různé použití názvů (Praha, Prag, Praga)
- Odlišné formátování telefonních čísel nebo čísel bankovních účtů (+420 776 423 551, 776423551)
- Prázdné hodnoty ve zdrojovém systému, které ale v datovém skladu musí být nenulové
- Nevalidní záznam, který byl do systému manuálně špatně zadán uživatelem

5.3 Load

Ve finální fázi se upravená a osekaná data vezmou z přechodné databáze a uloží se do datového skladu. V praxi k tomuto přenosu dochází většinou automaticky, běžně se tento proces spustí kolem půlnoci, aby je pracovníci, jež aktualizovaná a nová data druhý den potřebují, mohli využít. Často se datům ještě přiřadí i nové primární klíče, jelikož spousta z nich je ve zdrojových systémech vůbec neměla.

Ukládání dat se dělí na tři typy (Guru99, 2020):

- Počáteční načtení – naplnění všech tabulek datového skladu
- Přírůstkové načtení – periodicky plánované změny tabulek dle potřeby
- Celkové obnovení – vymazání obsahu jedné celé tabulky a následné uložení nových čerstvých dat

5.4 Testování ETL procesů

ETL vývojáři musí před zavedením procesů otestovat, zda jimi navržené transformace dat fungují správně, jak by měly. V praxi tak ve vývojovém prostředí vyvinou ETL proces, který za pomocí testovacích dat podrobí zkoušce, zda vše funguje, jak se předpokládá. Dále se ETL proces vyzkouší v testovacím prostředí, které obsahuje zkopírovaná data z pravých databázových systémů. Pokud vytvořený ETL proces pracuje správně, dojde k jeho nasazení do oficiálního produkčního prostředí firmy a ETL proces je ještě po nějakou dobu sledován, zda probíhá správně.

5.5 Fakta

Dle (Kimball, 2002) je tabulka faktů tou hlavní primární tabulkou dimenzionálního modelování. Jsou v ní uloženy záznamy podnikových dat. V těchto faktových tabulkách je mnoho cizích klíčů, jež jsou napojovány na primární klíče tabulek dimenzí. S faktovou tabulkou je úzce spojen pojem granularita. Tento pojem určuje míru podrobnosti dat. Čím je granularita nižší, tím jsou data detailnější a zase naopak. Faktové tabulky zabírají až devadesát a více procent multidimenzionálních databází. Neobsahují tolik sloupců, zato řádků v nich bývá většinou nespočet. Typickým příkladem může být tabulka faktů denního prodeje.

Dělení faktových tabulek dle (Zentut, 2020)

- Transakční
- Periodicky snímkové
- Akumulované snímkové

Transakční faktové tabulky obsahují nejvíce detailní informace a jsou spojeny s mnoha dimenzemi (Zentut, 2020). Periodicky snímkové tabulky pořizují data v určitých cyklech. Zdrojovými daty pro ně jsou právě transakční faktové tabulky.

Samotné tabulky faktů mohou mít jako primární klíč sloupec s ID řádku nebo může primární klíč být složený z několika cizích klíčů. Tento klíč se nazývá kompozitní (složený). Faktové

tabulky se nalézají vždy ve středu schémat (sněhové vločky, hvězdy). Záznamy v těchto tabulkách jsou všechny na stejné úrovni a nejsou rozděleny do žádných hierarchií.

I samotná fakta v tabulkách lze rozdělit na tři druhy (Guru99, 2020):

- Aditivní
- Semi-aditivní
- Neaditivní

Aditivní fakta lze agregovat do všech dimenzí. V případě semi-aditivních faktů lze agregovat pouze některé dimenze. Neaditivní fakta uchovávají pouze základní jednotky agregací.

5.6 Dimenze

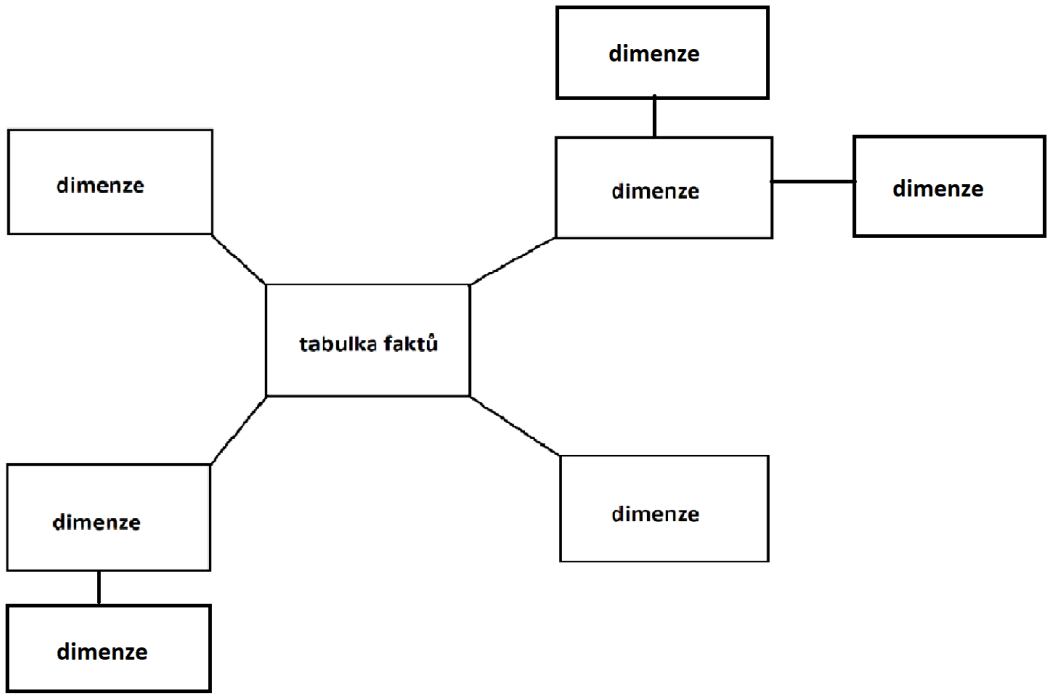
Jak píše (Kimball, 2002), dimenzionální tabulky jsou nedílnou součástí dimenzionálního modelu. Obsahují textový popis dané problematiky. Obsahují velké množství sloupců, které popisují řádky. Jejich počet se může pohybovat i až okolo sta. Každá dimenze má svůj primární klíč, kterým je odkazována na další tabulku (faktovou, či další dimenzi). Každá taková tabulka dimenzí představuje souvislosti k faktům. Dimenzionální tabulky často představují hierarchické vztahy uvnitř podniku. Dimenze vlastně určují úhel pohledu na věc. Nejtypičtějšími dimenzemi jsou dimenze časová, dimenze lokace, dimenze produktu. Příkladem pro hierarchii časové dimenze je možnost zabalit do vyšší úrovně či detailněji rozdělit časové úseky na roky, kvartály, měsíce, týdny, dny.

5.7 Schéma datového skladu

Datový sklad může být vystavěn na základě dvou přístupů. Tabulky dimenzí jsou propojeny s tabulkami faktů za pomoci cizích klíčů. Nejvíce se používají schémata sněhové vločky (snowflake) a hvězdy (star).

5.7.1 Schéma sněhové vločky

Toto schéma vizualizací své struktury připomíná tvar sněhové vločky. Jednotlivé dimenze na sebe postupně navazují a všechny vyúsťují do tabulek faktů. Dimenze jsou v tomto schématu normalizované a jsou rozdělené do několika mezi sebou propojených tabulek. Tato struktura umožňuje menší nároky na velikost úložiště, jelikož data nejsou redundantní, a zároveň snižuje nároky na celkový výkon systému a rychlost jeho odezvy.

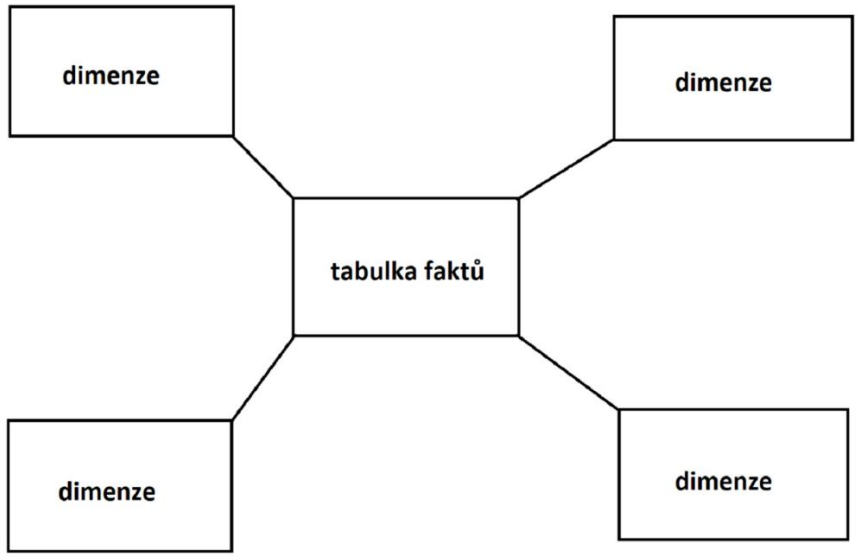


Obrázek 5 - Schéma sněhové vločky [23]

Zdroj: vlastní tvorba dle informací z Guru99

5.7.2 Schéma hvězdy

Dle (teradatapoint, 2020) se jedná o nejjednodušší schéma používané pro datové sklady. Zároveň je také nejčastěji používaným schématem. Tabulky dimenzí jsou zde na rozdíl od schématu sněhové vločky nenormalizované a dimenze jsou orientované pouze do jedné tabulky. Dimenze jsou pak napojeny na jednu a více tabulek s fakty. Svým vzhledem připomínají právě tvar hvězdy.



Obrázek 6 - Schéma hvězdy [23]

Zdroj: vlastní tvorba dle informací z Guru99

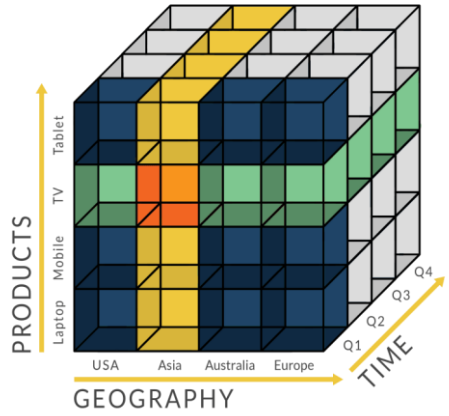
6 Historizace

Pojem historizace je v datových skladech jedním z nejzásadnějších pojmů. Historizování dat znamená uchovávání záznamů nejen aktuálních, ale i těch původních, které jim přecházely. Pro spoustu podniků je podstatné mít údaje o tom, kolik jejich výrobek či služba stála dříve, jaké byly díky nim zisky a další informace. Firemním analytikům a manažerům toto pak usnadňuje práci a umožňuje vytvářet reporty za minulá období, či kupříkladu porovnávat stávající období s těmi předchozími. V případě vyskytnutí nějakého problému pocházejícího z nějaké anomálie v datech v minulosti, lze jednodušeji dohledat, kde je chyba a co za tím stálo. V transakčních databázích jsou při plném naplnění úložiště původní záznamy přepisovány. K tomuto účelu slouží datové sklady, které tuto funkci podporují. V datovém skladu má každý záznam časový atribut, kdy byl vytvořen, popřípadě modifikován nebo ukončena jeho platnost. Data se zde nikdy nemažou.

7 Kostka OLAP

Po zpracování ETL procesem jdou očištěná data do datového skladu a následně do datových tržišť (v případě Top Down metody) či do jednotlivých datových tržišť a poté do skladu (v případě Bottom Up metody). Odtud se data přenáší do OLAP kostky. Na základě struktury hvězdy či sněhové vločky a rozdělení datových tabulek do příslušných dimenzí a faktů, je postaven datový sklad. Následně z tohoto modelu se pak odebírají data pro analytické účely datové kostky, jež je na nich vystavěna.

Jedná se o aplikaci pro podporu rozhodování (Humpries a kol., 2002), která umožňuje náhled na dřívější a současná podniková data. Poskytuje také funkce na zobrazení ve formách grafů a diagramů. Kostka OLAP je základním kamenem celé této technologie a také posledním dílem v řešení datového skladu. Datová krychle je multidimenzionální struktury a svou vysokou rychlostí při provádění analýz dat tak převyšuje na relačními databázemi. Umožňuje jednoduché prohledávání kostky a sčítání velkého množství dat (microsoft.docs, 2019). Krychle tak obsahuje již agregovaná data, jejichž dotazování a výpočet by zabral spoustu času. Tyto agregované výsledky jsou poté připravené k budoucím analýzám.



Obrázek 7 - Názorný příklad OLAP krychle [24]

7.1 Druhy OLAP systémů

OLAP systémy se rozdělují podle jejich výstavby a funkcionalit. Nejznámější dělení těchto úložišť je na multidimenzionální OLAP (MOLAP), relační OLAP (ROLAP) a hybridní OLAP (HOLAP). Existuje ještě několik druhů úložišť, ty však nejsou v praxi tak často využívané. (Guru99, 2020)

7.1.1 ROLAP

Tento OLAP pracuje s daty z relačních databází. Fakta a tabulky jsou uchovávány jako relační tabulky. Uživatelé na ně mají však náhled, jako by byly uložené v multidimenzionální struktuře.

Data jsou však fyzicky uložená pouze v původní relační databázi, proto při tomto procesu nevzniká redundance. Jedná se o nejrychleji se rozšiřující větev OLAP technologií. Nevýhodou je, že SQL dotazy zde trvají delší dobu než kupříkladu v MOLAP úložištích. (L. Lacko 2003)

7.1.2 MOLAP

Dle slov (L.Lacka, 2003) pro multidimenzionální úložiště se data získávají buďto z datového skladu nebo přímo z transakčních zdrojových databází. Ta se ukládají do multidimenzionální struktury. Na základě získaných dat se začnou vypočítávat různé agregační funkce, které mají dopředu vypočítané všechny možné kombinace faktů a dimenzí, a které budou předem připravené pro další využití (Guru99, 2020). Tento přístup je nejvíce typický pro multidimenzionální modelování. Výhodu je, že MOLAP nástroje dokáží obsluhovat i méně zkušení uživatelé, jelikož jsou dobře navržené. Kostka je také uzpůsobena k provádění operací „slice“ a „dice“, o kterých bude zmíněno později. Nevýhodou je redundance dat, které jsou uložené jak v relační databázi, tak v multidimenzionální a také vyšší nároky na kapacitu úložiště. MOLAP úložiště také neposkytují veškeré detailní data.

7.1.3 HOLAP

Tento nástroj je kombinací obou předchozích úložišť, tedy MOLAP a ROLAP. Jsou zde zkombinovány výhody jak relačního, tak multidimenzionálního modelu. Agregovaná a vypočítaná data se ukládají do multidimenzionální struktury – kostky, kdežto detailní data jsou uchovávána v relačním modelu. Vzhledem k tomu, že se jedná o kombinaci dvou odlišně pracujících prostředí, vyžaduje si to větší nároky na zkušenosti uživatelů.

7.2 Analytické operace nad daty v OLAP kostce

Data v OLAP kostkách slouží účelům usnadnění tvorby reportů a různých analýz. Známe pět základních analytických operací. Jednotlivé operace jsou přiblíženy na webových stránkách (Guru99, 2020).

Roll-up

Tato operace umožňuje nahlédnout do jednotlivých hierarchických úrovní dat. Umožňuje tedy shlukovat data podle některých parametrů. Kupříkladu počet prodaných automobilů za týden/měsíc/kvartál. Čili jedná se o sbalení jednotek nižší úrovně do té vyšší (z jednotlivých týdnů se to shlukuje do celého měsíce).

Drill-down

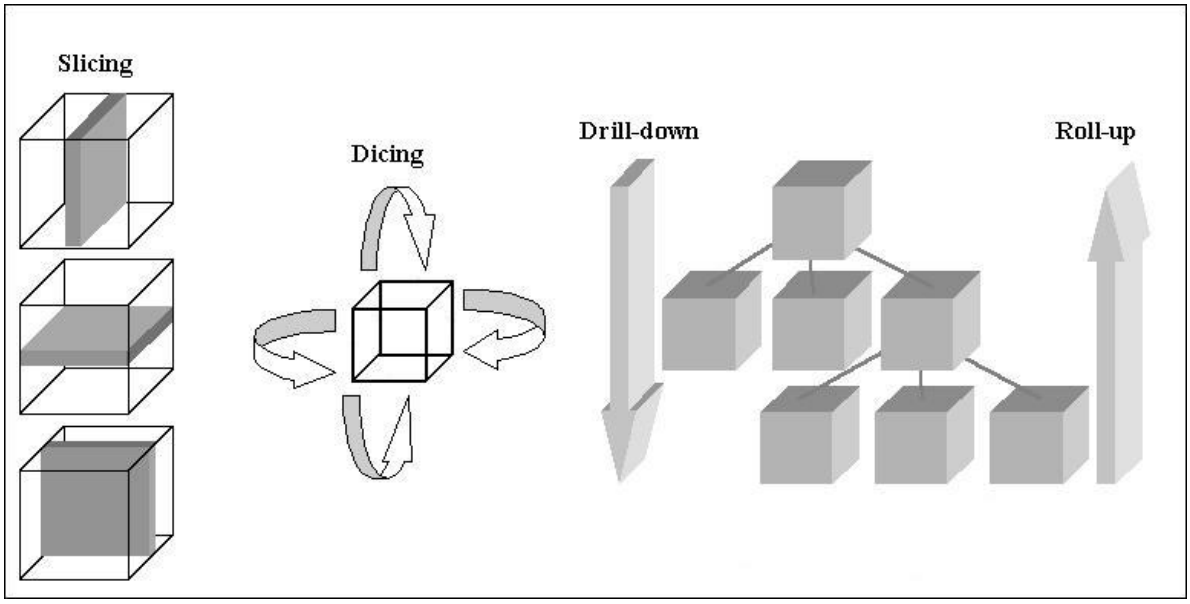
Je přesným opakem operace roll-up, kdy jsou data vyšší úrovně rozebrána na menší kousky (měsíce podrobněji na týdny).

Slice

V tomto případě je vybrána určitá hodnota jedné z dimenzí, která zůstává po celou dobu konstantní. Z následujících dimenzí je vytvořen „plátek“, na který je dále nahlíženo. Lze si to představit jako uříznutí jedné strany rubikovy kostky, kterou pak zkoumáme.

Dice

Při této operaci lze vybrat dvě nebo více dimenzí, které zobrazí výsledek v podobě nové „osekané“ krychle.



Obrázek 8 - Operace nad OLAP kostkou [25]

8 PRAKTICKÁ ČÁST

V praktické části dojde k představení konceptu datového skladu a jeho následné implementaci.

8.1 Firma Dine4Fit a aplikace Kalorické tabulky

Nejprve bude firma krátce představena (kaloricketabulky.cz, 2020). Firma Dine4Fit, a.s. je firma sídlící v Hradci Králové, jež podniká v několika odvětvích a byla založena v roce 1999. Jejich hlavní oblastí podnikání je provoz databázové aplikace Kalorické tabulky. Tato aplikace napomáhá jejím uživatelům ke snadné kontrole denního příjmu energie, což jim usnadňuje sledovat své cíle.

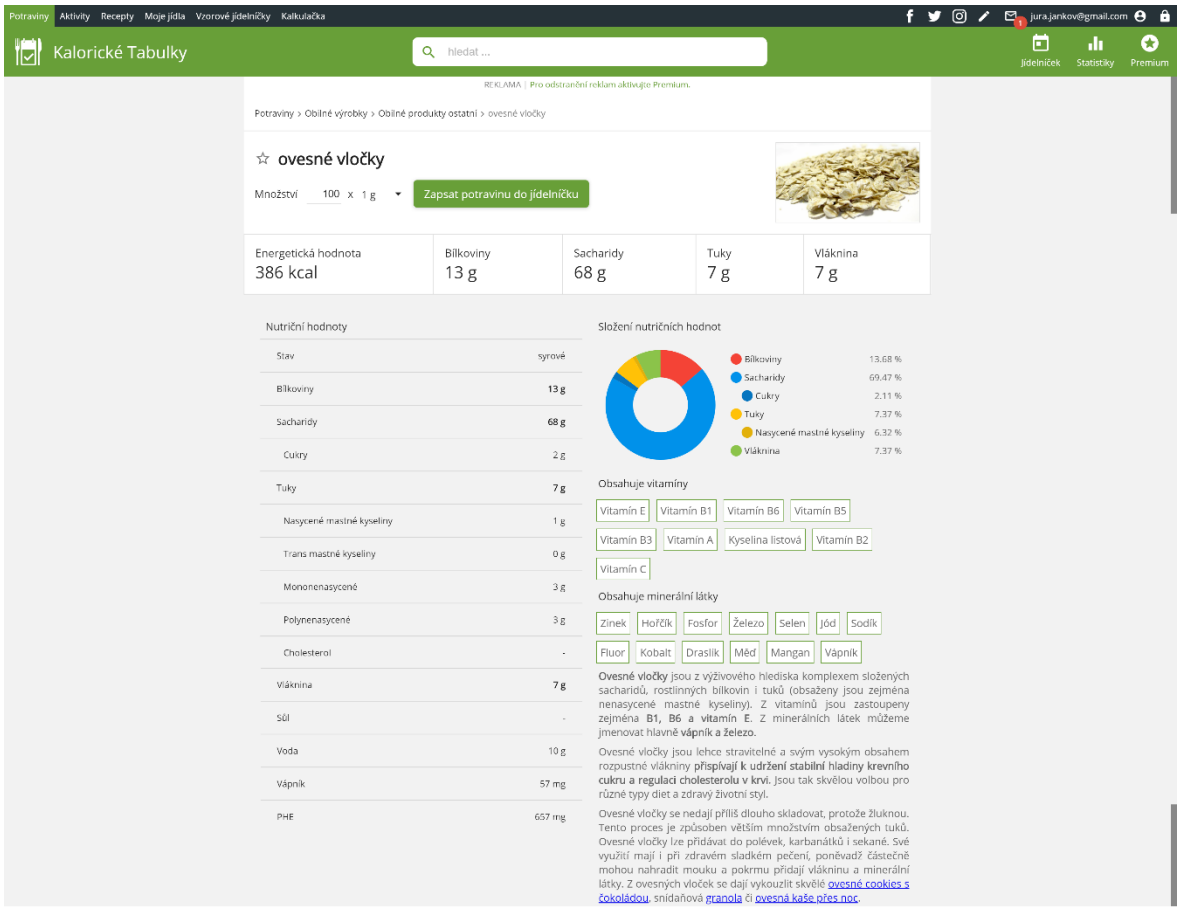
8.2 Uživatelské funkce

Uživatelé si zde mohou prohlížet energetické hodnoty jednotlivých potravin, jež jsou prodávány v obchodech, ale i již uvařených jídel, aby zjistili, kolik takovýto pokrm obsahuje energie.

Informace uváděné u jednotlivých potravin:

- Energetická hodnota (kJ/kcal)
- Makroživiny – obsažené množství (bílkoviny, sacharidy, tuky), které se pak ještě dělí na jednotlivé podkategorie
- Mikroživiny – výpis druhů a množství mikronutrientů jako jsou vitamíny a minerály
- Množství vlákniny, vody, soli
- Koláčový graf procentuálního složení potraviny
- Jednoduchý slovní popis dané potraviny

U každé potraviny si lze také upravit její množství přesně na gramy, aby si to každý uživatel mohl upravit dle své potřeby.



Obrázek 9 – Náhled na nutriční hodnoty potraviny Ovesná kaše ve webovém prohlížeči

Zdroj: https://www.kaloricketabulky.cz/potraviny/ovesne-vlocky

Aplikace ale také nabízí uživatelům založit si svůj vlastní účet. Uživatel zadá několik požadovaných atributů, podle kterých se bude jídelníček řídit. Následně si pak může sledovat průběh svého postupu.

Zadávané hodnoty:

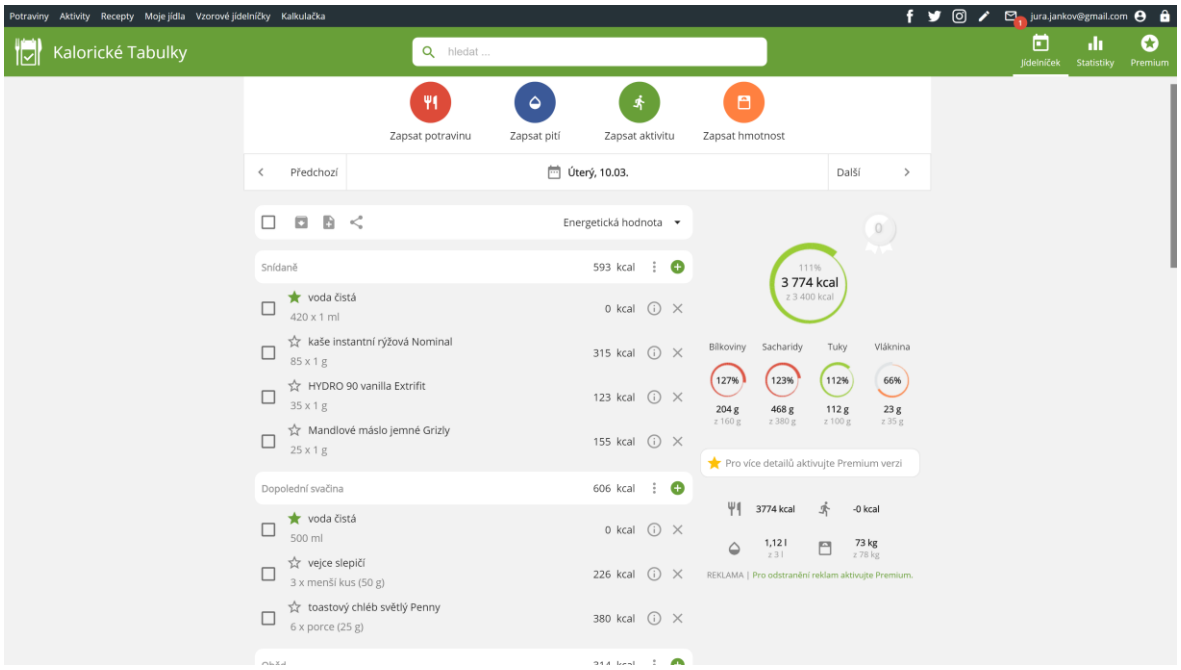
- Pohlaví
- Výška
- Hmotnost
- Rok narození

Nastavení jídelníčku:

- Cíl (být fit, zhubnout, nabrat svaly)
- Cílová hmotnost
- Pitný režim

- Denní výdej energie (sedavý životní styl, určitý stupeň fyzické aktivity, profesionální sportovec)
- Popřípadě si lze i nastavit své vlastní množství makronutrientů, pokud nechce, aby mu je automaticky vypočítala aplikace

Díky tomu uživatel může zaznamenávat svůj celodenní příjem a výdej energie pomocí sestavování si jídelníčku. Uživatel si zapisuje, které potraviny během dne snědl a vypil, do jednotlivých chodů. Zároveň si také může zaznamenávat i aktivitu, kterou během dne vynaložil. Její druh, délku a náročnost. Aplikace mu poté jednotlivá jídla sčítá a uvádí celodenní energetickou bilanci včetně množství přijatých makroživin a vody.



Obrázek 10 - Příklad vyobrazení celodenního příjmu energie z mého vlastního účtu

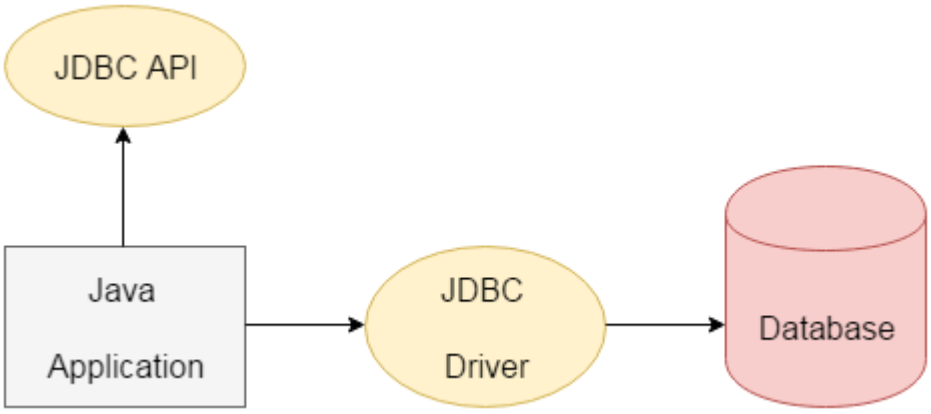
Zdroj: https://www.kaloricketabulky.cz/user/diary (vlastní účet)

Uživatel si také může v aplikaci vytvářet vlastní jídla (složením jednotlivého množství ingrediencí). Aplikace také nabízí velké množství převážně fitness receptů. Jejich autoři sepisují postup a množství potřebných ingrediencí a aplikace poté dopočítá, kolik energie a makroživin v tom pokrmu bude. Dále v aplikaci lze nalézt vzorové jídelníčky od soukromých autorů i samotného výživového týmu Kalorických tabulek. V neposlední řadě je tu kalkulačka, která vypočítává na základě zadaných parametrů uživatelův bazální metabolismus a BMI index. Dále také doporučí denní příjem kalorií a poměr makroživin v závislosti na cílech uživatele. V záložce statistik lze sledovat týdenní plnění přednastaveného jídelníčku, pitného režimu a fyzických aktivit. Zároveň taky lze graficky zobrazit množství přijaté energie, vody a změny hmotnosti v průběhu času zpětně až k prvopočátečním záznamům.

8.3 JDBC připojení

JDBC je uživatelské rozhraní pro programátory, kteří pracují s jazykem Java. Zkratka označuje pojem Java Database Connectivity. Tento ovladač je potřebný pro uživatele, jež se chtějí připojit k databázovému serveru (interval.cz, 2003). Díky JDBC rozhraní může vývojář software naprogramovat tak, aby uměl přistupovat k přes JDBC a nemusí rozlišovat jednotlivé druhy databází. JDBC vychází z původního standardu ODBC a dělí se na čtyři typy:

- JDBC-ODBC Bridge Driver
- Native Driver
- Network Protocol Driver
- Thin Driver



Obrázek 11 - JDBC connection

Zdroj: https://www.javatpoint.com/java-jdbc

Tato aplikace napsaná v jazyce Java, dokáže přistupovat k datům uloženým v tabulkách relační databáze. Dokáže nad nimi provádět požadované změny – mazání, aktualizace, přidávání dat.

8.4 DataGrip

K samotné extrakci dat z poskytnuté databáze Kalorických tabulek byl využit nástroj DataGrip od společnosti JetBrains (JetBrains, 2020). Firma se zabývá vývojem softwarových programů pro programátory a projektové manažery. Konkrétně program DataGrip nabízí databázové vývojové prostředí sloužící SQL vývojářům (JetBrains, 2020). Nabízí několik užitečných funkcí jako je kupříkladu smart doplňování rozepsaného kódu, detekci a opravení chyb v kódu, či provedení různých úkonů pouhým jednoduchým klikáním, místo psaní celého kódu. Nástroj slouží ke správě i vytváření databází a k provádění dotazů nad nimi. Spravovat lze nejen databázi lokální, ale díky JDBC ovladači lze DataGrip připojit k několika cloudovým databázovým prostředím (MySQL, MSSQL, Oracle, Vertica, Azure SQL Database, atd…).

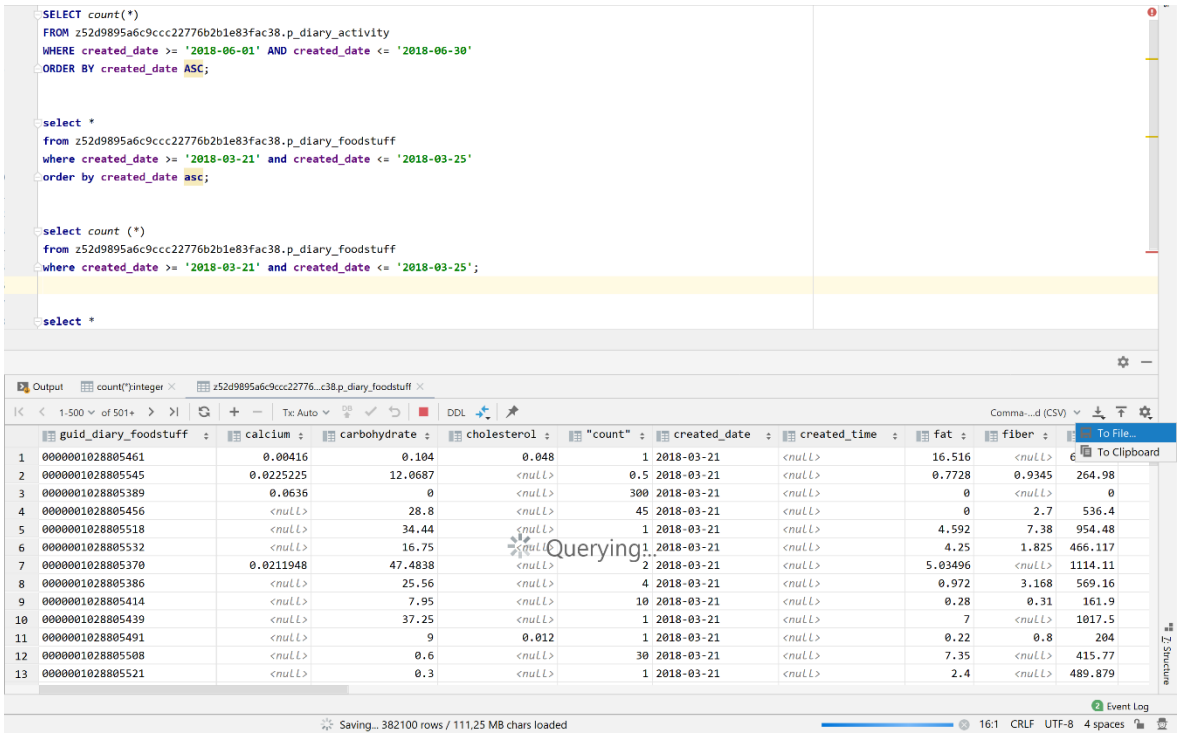
8.5 Extrakce dat

Extrahovat data z DataGripu lze do několika formátů. Program nabízí možnosti jako CSV, JSON, XML. Pro tuto práci byla vybrána nejpoužívanější forma a tou je CSV. Poskytnutá databáze od firmy Dine4Fit obsahuje několik desítek tabulek. Některé z nich jsou svým množstvím dat opravdu malé, ale některé z nich mají miliony záznamů. Nejvíce záznamů se pohybuje v tabulkách, kde si uživatelé zapisují každodenní snědené jídlo (diary_foodstaff) či vyprodukovanou aktivitu (diary_activity) nebo v seznamu všech uživatelů. V těchto tabulkách je dohromady v poskytnutých datech z časového období let 2017 – 2019 kolem 370 milionů řádků. Extrakce takovéhoto množství dat by trvala opravdu hrozně dlouho, takže pro účely této bakalářské práce byla exportována data pouze z časového období 1.1.2018 – 30.6.2018.

Tabulky, jež byly exportovány:

- p_diary_foodstuff
- p_diary_time
- p_foodsuff
- p_foodstuff_translation
- p_foodstuff_type
- p_foodstuff_type_translation
- p_foodstuff_unit
- p_user_statistics_view

Málo obsáhlé tabulky bylo možné exportovat celé najednou. Jelikož program DataGrip občas uprostřed exportování dat vyhodil chybu a export byl přerušen, u obsáhlejších tabulek byl export rozdělen do několika menších dávek.



	guid_diary_foodstuff	calcium	carbohydrate	cholesterol	"count"	created_date	created_time	fat	fiber	
1	0000001028805461	0.00416	0.104	0.048	1	2018-03-21	<null>	16.516	<null>	
2	0000001028805545	0.0225225	12.0687	<null>	0.5	2018-03-21	<null>	0.7728	0.9345	264.98
3	0000001028805389	0.0636	0	<null>	300	2018-03-21	<null>	0	<null>	0
4	0000001028805456	<null>	28.8	<null>	45	2018-03-21	<null>	0	2.7	536.4
5	0000001028805518	<null>	34.44	<null>	1	2018-03-21	<null>	4.592	7.38	954.48
6	0000001028805532	<null>	16.75	<null>	1	2018-03-21	<null>	4.25	1.825	466.117
7	0000001028805370	0.0211948	47.4838	<null>	2	2018-03-21	<null>	5.03496	<null>	1114.11
8	0000001028805386	<null>	25.56	<null>	4	2018-03-21	<null>	0.972	3.168	569.16
9	0000001028805414	<null>	7.95	<null>	10	2018-03-21	<null>	0.28	0.31	161.9
10	0000001028805439	<null>	37.25	<null>	1	2018-03-21	<null>	7	<null>	1017.5
11	0000001028805491	<null>	9	0.012	1	2018-03-21	<null>	0.22	0.8	204
12	0000001028805508	<null>	0.6	<null>	30	2018-03-21	<null>	7.35	<null>	415.77
13	0000001028805521	<null>	0.3	<null>	1	2018-03-21	<null>	2.4	<null>	489.879

Obrázek 12 - Export dat do CSV souboru

Zdroj: záznam exportování dat pomocí programu DataGrip

Tabulku **user_statistics_view** bylo možné exportovat podle data založení účtu v intervalech tří let pomocí kódu:

```
SELECT *
FROM z52d9895a6c9ccc22776b2b1e83fac38.p_user_statistics_view
WHERE created >= '2010-01-01' and created <= '2013-12-31';
```

Tabulka **diary_foodstuff** byla ale mnohem obsáhlejší. Nakonec tak byla data exportována většinou po pětidenních intervalech. Kód se opět pouze měnil ve **WHERE** klauzuli. Příklad kódu:

```
SELECT *
FROM z52d9895a6c9ccc22776b2b1e83fac38.p_diary_foodstuff
WHERE created_date >= '2018-03-16' AND created_date <= '2018-03-20'
ORDER BY created_date ASC;
```

Exportovány byly pro jistotu všechny sloupce dané tabulky a pro mou vlastní přehlednost byla data seřazena vzestupně podle data jejich zápisu (created_date). Jedna taková extrakce většinou obsahovala přes dva miliony řádků.

Za účelem vybudování datového skladu nebylo potřeba extrahovat veškerá data. Byly vynechány veškeré tabulky, jež souvisely s aktivitami, tabulky o časech a zařízeních uživatelských přihlášení, předplatném, atd… Toto byla první část transformací, jež byly nad daty prováděny.

8.6 Vybudování datového skladu v lokální databázi

Na základě dat exportovaných ze vzdáleného úložiště, byla vytvořena lokální Microsoft SQL Server databáze. Databáze byla vytvářena opět za pomoci programu DataGrip od JetBrains a pojmenována Dine4Fit_official. Toto datové schéma slouží jako před přípravná databáze pro pozdější vybudování hvězdicově navrženému datovému skladu obsahujícímu tabulky faktů a dimenzí. Sklad se zaměřuje na sledování deníku jídelního režimu uživatele. Z tohoto důvodu tabulek pro toto schéma již nebylo potřeba takové množství, jako bylo v databázi firmy Dine4Fit.

Základními kameny této databáze byly tabulka **user**, jež obsahovala informace o uživatelích a tabulka **diary_foodstuff**, ve které byly veškeré zapsané potraviny za období prvního půlroku roku 2018. I zde došlo k určitým transformacím tabulek, jelikož pro potřeby následného použití dat z datového skladu nebylo nezbytně nutné uchovávat veškeré sloupce.

Script pro vytvoření tabulky diary_foodstuff:

```
create table Dine4Fit_official.diary_foodstuff
(
   guid_diary_foodstuff Char(32) PRIMARY KEY ,
   calcium Float(15),
   carbohydrate Float(15),
   cholesterol Float(15),
   created_date Date default '',
   fat Float(15),
   fiber Float(15),
   kj Float(15),
   protein Float(15),
   salt Float(15),
   sodium Float(15),
   sugar Float(15),
   title nVarchar(255) default '',
   unit nVarchar(45) default '',
   water Float(15),
   guid_diary_time Char(32) FOREIGN KEY REFERENCES
Dine4Fit_official.diary_time(guid_diary_time),
   guid_foodstuff Char(32) FOREIGN KEY REFERENCES
Dine4Fit_official.foodstuff(guid_foodstuff),
   guid_user Char(32) FOREIGN KEY REFERENCES Dine4Fit_official.[user](guid_user),
);
```

Script pro vytvoření tabulky user:

```
create table Dine4Fit_official.user
(
    guid_user char(32) PRIMARY KEY,
    birth_year int,
    language varchar(5)  default '',
    sex varchar(45) default '',
    weight int,
    height int
);
```

Dále byly vytvořeny tabulky:

diary_time – guid_diary_time (char, **PK)**, name (varchar)

foodstuff – guid_foodstuff (char, **PK**), kj (float), protein (float), carbohydrate (float), sugar (float), fat (float), created (date), fiber (float), cholesterol (float), salt (float), calcium (float), sodium (float), water (float), guid_foodstuff_type (char, **FK**), guid_user (char, **FK**)

foodstuff_translation – guid_foodstuff_translation (char, **PK**), language (varchar), title (nvarchar), guid_foodstuff (char, **FK**)

foodstuff_type – guid_foodstuff_type (char, **PK**), guid_parent_foodstuff_type (char)

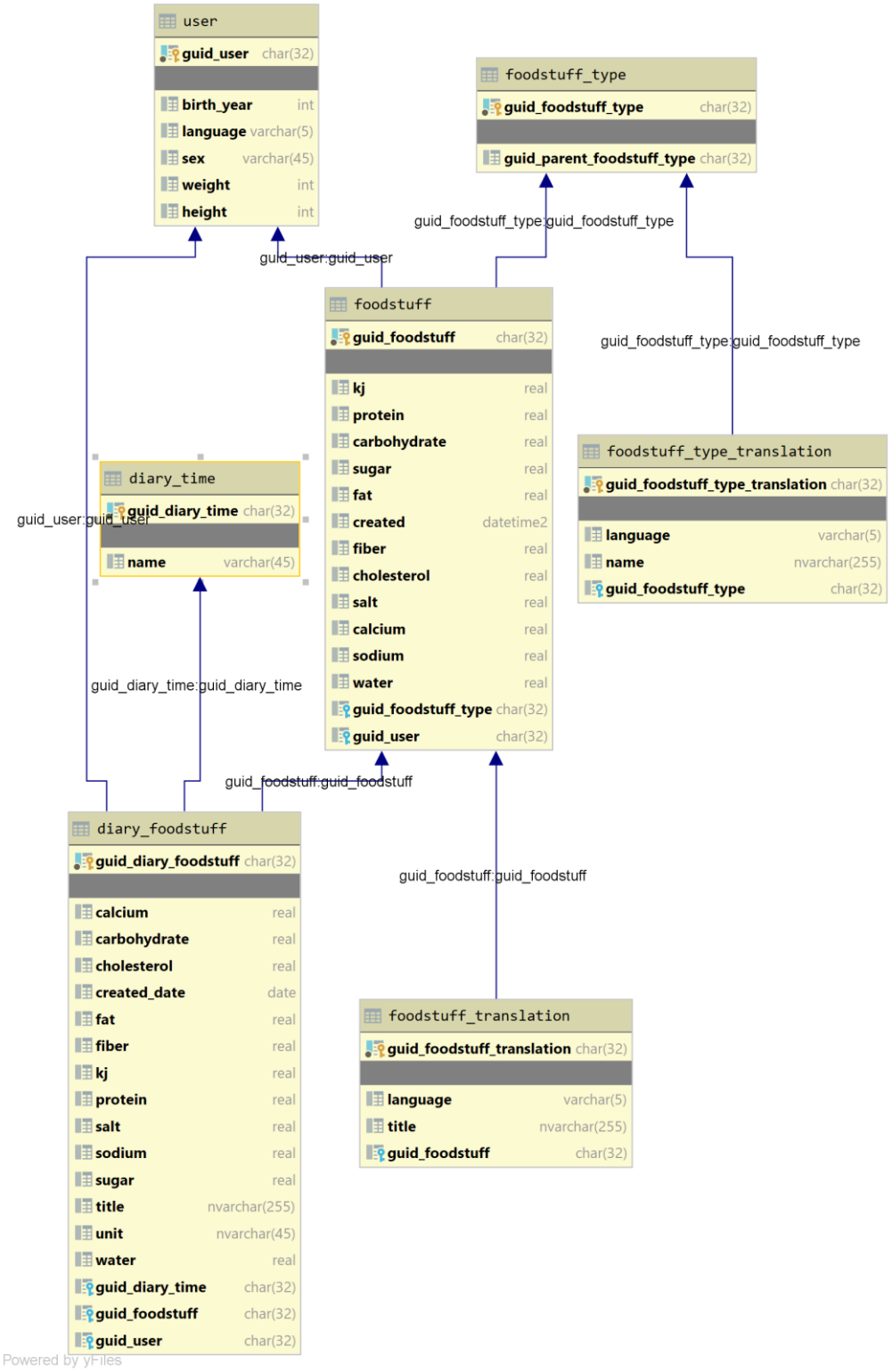
foodstuff_type_translation – guid_foodstuff_type_translation (char, **PK**), language (varchar), name (nvarchar), guid_foodstuff_type (char, **FK**)

Vzhledem k tomu, že aplikace Kalorické tabulky je provozována i v zemích mluvících ruským jazykem, bylo nutné v některých tabulkách zvolit nvarchar datový typ namísto běžného varcharu, který nedokáže uchovat unicodové znaky a nebyl by tak schopný zapsat znaky azbuky.

Tabulky foodstuff_translation a diary_foodstuff byly problematické při importu, jelikož jejich cizí klíče (foreign key) nešlo spojit s primárními klíči v patřičných tabulkách. Pravděpodobně v datech exportovaných z firemní databáze některé chyběly. Stejně tomu bylo i při návaznosti cizího klíče diary_foodstuff na primární klíč v tabulce diary_time, kde chyběl záznam s primárním klíčem „1“, který měl reprezentovat snídani jako jeden z mořných časů jídla. Byl doplněn jednoduchým kódem:

```
INSERT INTO Dine4Fit_official.diary_time VALUES ('1','snidane');
```

Samotné provázání databáze pomocí primárních a cizích klíčů je znázorněno na níže uvedeném schématu:

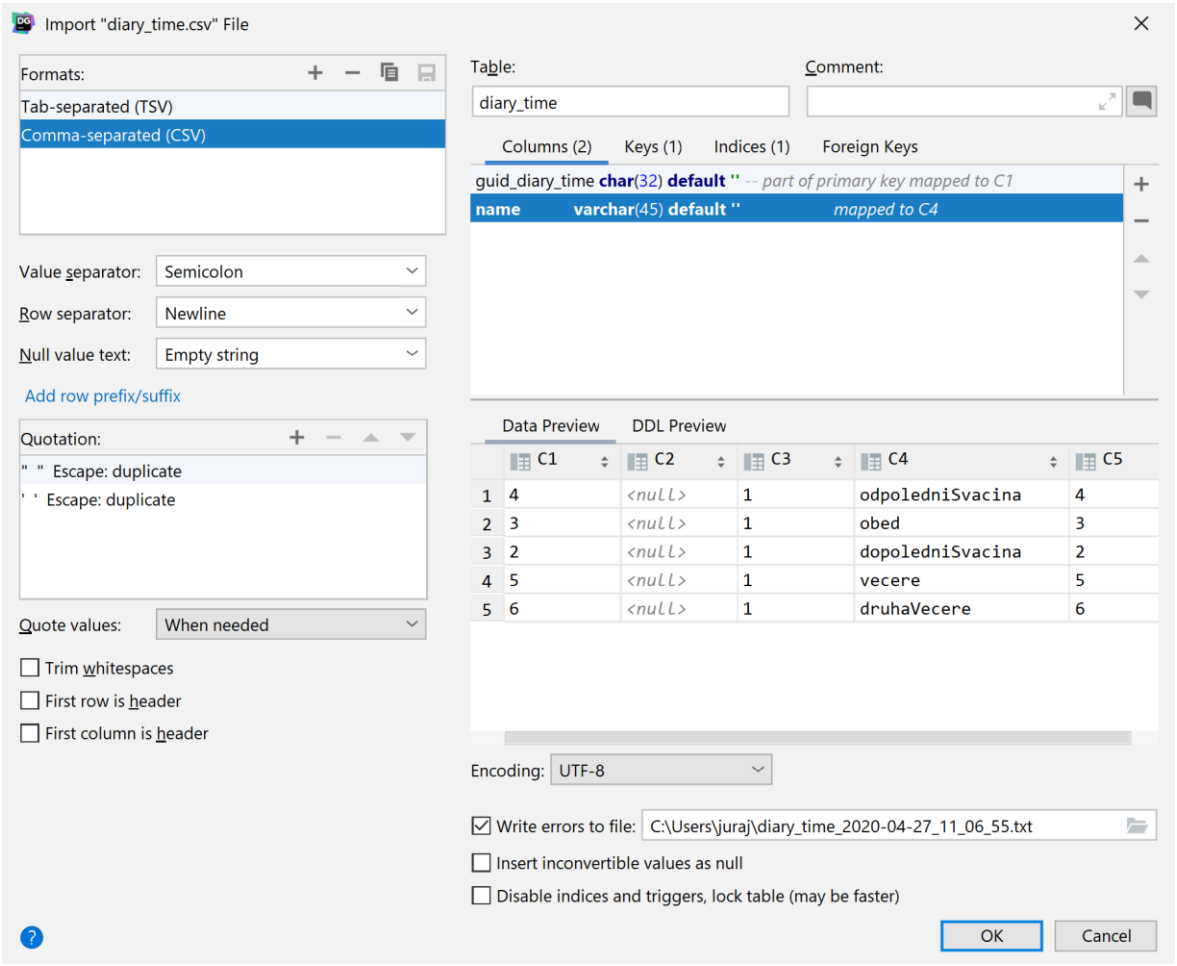


Obrázek 13 - Schéma datového skladu

Zdroj: vlastní tvorba – export obrázku z programu DataGrip

8.6.1 Import dat

Data byla importována do databáze opět v programu DataGrip. Program při importu z flat filu umožňuje vybírat separátor sloupců i dalších řádků pomocí čárky, tabulátoru, nového řádku nebo středníku. Dále také poskytuje možnost k jednotlivým sloupcům tabulky přiřadit jakýkoliv sloupec ze zdrojového souboru. V případě přeházení sloupců se ale import velkých souborů značně zpomaluje. Dále také umožňuje nullové hodnoty rovnou přepisovat na cokoliv jiného.



Obrázek 14 - Ukázka importu dat do tabulky diary_time

Zdroj: vlastní tvorba (program DataGrip)

Největší problém nastal při importování dat do foodstuff_translation. V této tabulce jsou překlady jednotlivých potravin do dalších jazyků. Často se zde používají uvozovky, které potom import nechtěl zpracovat a jednotlivé řádky pak vyhodnotil chybou nezapsal je. Zároveň je v ruských názvech pro potraviny občas i k vypsání uvozovek sled písmen a znaků ", takže bylo nutné vynechat i středníky. Nakonec byla tato tabulka vyexportována znovu, ale jednotlivé sloupce odděloval tabulátor, takže se jednalo o TSV soubor. Uvozovky, jež dělaly neplechu byly pro účel správného importu smazány.

8.6.2 Transformace dat

Všechny tabulky v tomto schématu, až na foodstuff_type_translation a diary_time a foodstuff_translation, byly podrobeny transformacím, jelikož se zde vyskytovalo spousta záznamů, které měly hodnotu null. Proto byly pro aktualizaci tabulek využity tyto skripty:

Tabulka user:

Tato tabulka obsahuje informace o uživatelích = rok narození, jazyk uživatele, jeho pohlaví, výška a hmotnost. Pro tuto tabulku byly přepsány nullové údaje o výšce a hmotnosti na hodnotu „0“, údaj o pohlaví a jazyce uživatele na „empty“ a nevyplněný rok narození nastaven na „1890“.

```
UPDATE Dine4Fit_official.[user]
SET sex = 'empty'
WHERE sex IS NULL;

UPDATE Dine4Fit_official.[user]
SET height = '0'
WHERE height IS NULL;

UPDATE Dine4Fit_official.[user]
SET weight = '0'
WHERE weight IS NULL;

UPDATE Dine4Fit_official.[user]
SET birth_year = '1890'
WHERE birth_year IS NULL;

UPDATE Dine4Fit_official.[user]
SET language = 'empty'
WHERE language IS NULL;
```

Tabulka foodstuff_type:

Na tuto tabulku se pomocí cizího klíče odkazuje tabulka foodstuff. Je zde číselník rozdělení potravin na jednotlivé kategorie, z nichž některé jsou nadřazené ještě jiným kategoriím. Proto je zde sloupec guid_parent_foodstuff_type, který může a nemusí být vyplněný. Nullové hodnoty byly opět přepsány na „empty“.

```
UPDATE Dine4Fit_official.foodstuff_type
SET guid_parent_foodstuff_type = 'empty'
WHERE guid_parent_foodstuff_type IS NULL;
```

Tabulka foodstuff:

Tato tabulka obsahuje informace o jednotlivých potravinách jako množství makroživin a mikroživin, množství energie v potravinách, ze kterých uživatelé mohou vybírat pro zápis do svého jídelníčku. V tomto případě byly všechny hodnoty null pro sloupce kj, protein, carbohydrate, sugar, fat, fiber, cholesterol, salt, calcium, sodium, water nahrazeny hodnotou „0“. Vždy opět jednoduchým skriptem, který se pouze měnil v závislosti na sloupci:

```
UPDATE Dine4Fit_official.foodstuff
SET protein = '0'
WHERE protein IS NULL;
```

Tabulka diary_foodstuff:

V této tabulce je zaznamenána každý zápis každého uživatele do jejich deníčku zapsaných potravin. Proto je zde ohromné množství dat, až 400 tisíc řádků za jeden den. Během updatu byly proto updaty pro každý sloupec rozděleny na měsíční intervaly. Pro calcium, carbohydrate, cholesterol, fat, fiber, kj, protein, salt, sodium, sugar a water se opět přepisovaly hodnoty z null na „0“. V případě sloupců title a unit z null na „empty“. Příklad aktualizace dat ve sloupci title:

```
UPDATE Dine4Fit_official.diary_foodstuff
SET title = 'empty'
WHERE title IS NULL AND created_date >= '2018-01-01' AND created_date <= '2018-01-
31';
```

8.7 Vytvoření faktové tabulky s dimenzemi.

Po tom, co byla data v lokální databázi transformována do lepší formy, byla vytvořena druhá databáze hvězdicového schématu. Tento datový sklad je určen ke sledování stravovacích návyků. Faktová tabulka nám umožňuje pozorovat přijímané makroživiny, mikroživiny nebo energii jednotlivých uživatelů v průběhu času.

Nejprve byly vytvořeny tabulky dimenzí:

- Dimenze časová (date_dimension),
- Dimenze názvu potraviny (food_name_dimension),
- Dimenze typu potraviny (food_type_dimension),
- Dimenze doby zápisu potraviny do jídelníčku (diary_time_dimension)
- Dimenze uživatele (user_dimension)
- Dimenze typu sledované hodnoty (measurement_type_food_dimension)

Následně byla vytvořena faktová tabulka odkazující se pomocí cizích klíčů na tabulky dimenzí:

- Faktová tabulka (diary_food_fact_table)

Jednotlivé tabulky byly vytvořeny pomocí následujících skriptů.

8.7.1 Skripty pro vytvoření jednotlivých tabulek

Dimenze časová

První z nich byla tabulka časová. Byla vytvořena tabulka s několika sloupci a následně do ní byla nalita pomocí kódu jednotlivá data.

```
--------------DATE_DIMENSION--------------
CREATE TABLE star_fact_table.date_dimension (
    date_ID int NOT NULL,
    date date NOT NULL,
    day tinyint NOT NULL,
    month tinyint NOT NULL,
    year int NOT NULL,
    PRIMARY KEY CLUSTERED (date ASC)
)

DECLARE @ActualDate DATE = '2018-01-01'
DECLARE @EndDate DATE = '2018-06-30'

WHILE @ActualDate <= @EndDate
BEGIN
    INSERT INTO star_fact_table.date_dimension (
        date_ID, date, day, month, year
    )

    SELECT date_ID = YEAR(@ActualDate) * 10000 + MONTH(@ActualDate) * 100 +
DAY(@ActualDate),
    date = @ActualDate,
    day = DAY(@ActualDate),
    month = MONTH(@ActualDate),
    year = YEAR(@ActualDate)

    SET @ActualDate = DATEADD(DD, 1, @ActualDate)
END
```

Dimenze názvu potraviny

Druhou byla tabulka názvů jednotlivých potravin, ve které je krom unikátního identifikátoru také název jídla a jazyk ve kterém je udávána.

```
--------------FOOD_NAME_DIMENSION--------------
SELECT guid_foodstuff, title, language INTO star_fact_table.food_name_dimension
FROM Dine4Fit_official.foodstuff_translation
```

Dimenze typu potraviny

Následuje dimenze typu potraviny, ve které je rozdělení potravin podle jejich druhů.

```
--------------FOOD_TYPE_DIMENSION--------------
SELECT f.guid_foodstuff, f.guid_foodstuff_type,
ftt.guid_foodstuff_type_translation, ftt.language, ftt.name
INTO star_fact_table.food_type_dimension
FROM Dine4Fit_official.foodstuff AS f
LEFT JOIN Dine4Fit_official.foodstuff_type_translation AS ftt ON
f.guid_foodstuff_type = ftt.guid_foodstuff_type
```

Dimenze doby zápisu potravy do jídelníčku

V této dimenzi rozlišuje, zda si uživatel potravinu zapisuje jako součást snídaně, dopolední svačiny, oběda, odpolední svačiny nebo večeře.

```
--------------DIARY_TIME_DIMENSION--------------
SELECT guid_diary_time, name INTO star_fact_table.diary_time_dimension
FROM Dine4Fit_official.diary_time
```

Dimenze uživatele

Obsahuje informace o tom, který uživatel si potravinu do deníku zapsal a také jakého je pohlaví, jaký používá jazyk, ročník narození a jakou má váhu a výšku.

```
--------------USER_DIMENSION--------------
SELECT guid_user, birth_year, language, sex, weight, height
INTO star_fact_table.user_dimension
FROM Dine4Fit_official.[user]
```

Dimenze typu sledované hodnoty

Tato dimenze je takovým číselníkem, který rozlišuje ve faktové tabulce, jakou hodnotu chceme sledovat. Opět nejdříve byla kódem vytvořena a pak naplněna potřebnými hodnotami.

```
--------------MEASUREMENT_TYPE_FOOD_DIMENSION--------------
CREATE TABLE star_fact_table.measurement_type_food_dimension (
    guid_measurement_type VARCHAR (1) PRIMARY KEY,
    measurement_title VARCHAR(45) NOT NULL
)

INSERT INTO star_fact_table.measurement_type_food_dimension VALUES
('1','protein'), ('2','carbohydrate'), ('3','sugar'), ('4','fat'), ('5','kj'),
('6','fiber'), ('7','cholesterol'), ('8','salt'), ('9','calcium');
```

Faktová tabulka:

Faktová tabulka je naplněna cizími klíči, jež odkazují na ostatní dimenze. Pro naše sledování byly vybrány parametry: proteiny, sacharidy, cukry, tuky, kJ, vláknina, cholesterol, soli a vápník.

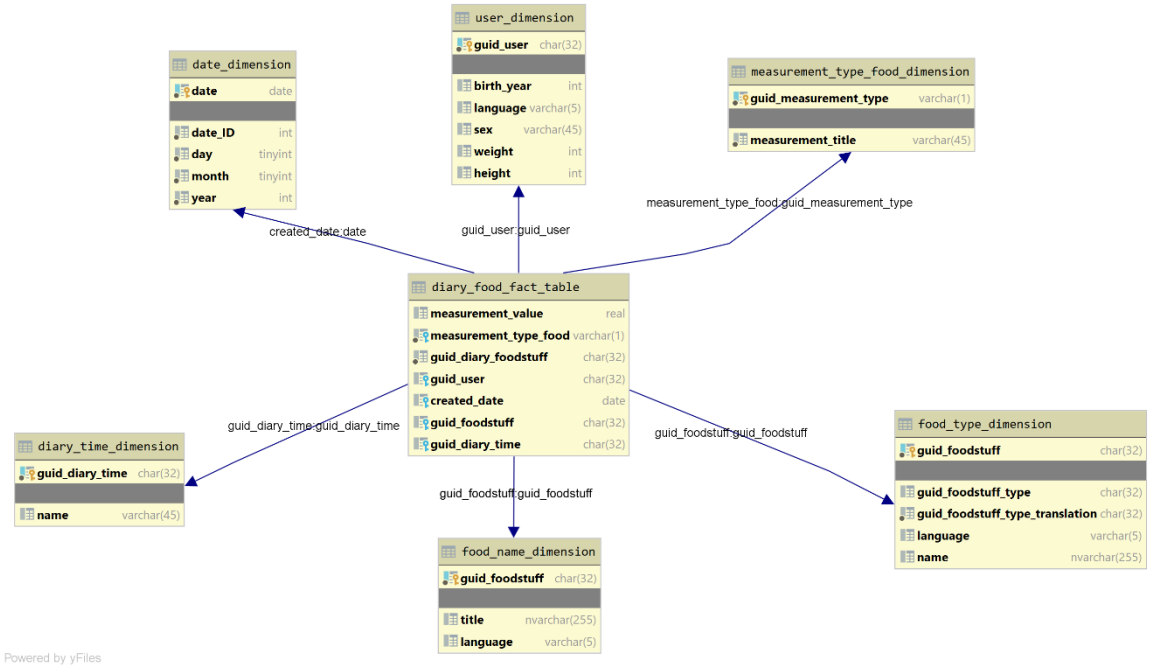
```
--------------DIARY_FOOD_FACT_TABLE--------------
SELECT protein AS measurement_value, '1' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
INTO star_fact_table.diary_food_fact_table
FROM Dine4Fit_official.diary_foodstuff
UNION
SELECT carbohydrate AS measurement_value, '2' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
FROM Dine4Fit_official.diary_foodstuff
UNION
SELECT sugar AS measurement_value, '3' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
FROM Dine4Fit_official.diary_foodstuff
UNION
SELECT fat AS measurement_value, '4' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
FROM Dine4Fit_official.diary_foodstuff
UNION
SELECT kj AS measurement_value, '5' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
FROM Dine4Fit_official.diary_foodstuff
UNION
SELECT fiber AS measurement_value, '6' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
FROM Dine4Fit_official.diary_foodstuff
UNION
SELECT cholesterol AS measurement_value, '7' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
FROM Dine4Fit_official.diary_foodstuff
UNION
SELECT salt AS measurement_value, '8' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
FROM Dine4Fit_official.diary_foodstuff
UNION
SELECT calcium AS measurement_value, '9' AS measurement_type_food,
guid_diary_foodstuff, guid_user, created_date, guid_foodstuff, guid_diary_time
FROM Dine4Fit_official.diary_foodstuff
```

Cizí klíče:

Cizí klíče z faktové tabulky byly napojeny na jednotlivé dimenze tímto způsobem:

- diary_food_fact_table (guid_foodstuff) → food_name_dimension (guid_foodstuff)
- diary_food_fact_table (created_date) → date_dimension (date)
- diary_food_fact_table (guid_user) → user_dimension (guid_user)

- diary_food_fact_table (measurement_type_food) → measurement_type_food_dimension (guid_measurement_type)
- diary_food_fact_table (guid_foodstuff) → food_type_dimension (guid_foodstuff)
- diary_food_fact_table (guid_diary_time) → diary_time_dimension (guid_diary_time)



Obrázek 15 - Schéma databáze s tabulkou faktů a dimenzí

Zdroj: vlastní zdroj – export obrázku z programu DataGrip

Na následujících obrázcích (Obrázek č. 16 a 17) můžeme vidět zdroje pěti tuků, které jeden konkrétní uživatel (žena) s konkrétním identifikátorem (guid_user = **7e808d70275db61c**) přijal dle zapsaných potravin k večeři prvním pěti zapsaným záznamům podle tohoto kódu:

```
SELECT TOP 5 measurement_value AS 'Měřená hodnota', mtfd.measurement_title AS 'Typ
hodnoty', fnd.title AS 'Název jídla', ftd.name AS 'Druh jídla', ud.sex AS
'Pohlaví', ud.birth_year AS 'Rok narození', dtd.name AS 'Doba jídla', created_date
FROM star_fact_table.diary_food_fact_table AS dft
LEFT JOIN star_fact_table.measurement_type_food_dimension AS mtfd ON
dft.measurement_type_food = mtfd.guid_measurement_type
LEFT JOIN star_fact_table.food_name_dimension AS fnd ON dft.guid_foodstuff =
fnd.guid_foodstuff
LEFT JOIN star_fact_table.food_type_dimension AS ftd ON dft.guid_foodstuff =
ftd.guid_foodstuff
LEFT JOIN star_fact_table.user_dimension AS ud ON dft.guid_user = ud.guid_user
LEFT JOIN star_fact_table.diary_time_dimension AS dtd ON dft.guid_diary_time =
dtd.guid_diary_time
WHERE dft.measurement_type_food IN ('4') AND dft.guid_user IN ('7e808d70275db61c')
AND dft.guid_diary_time IN ('5') AND dft.measurement_value NOT IN ('0') AND
dft.guid_foodstuff IS NOT NULL;
```

Obrázky jsou rozděleny na dvě části, jelikož dohromady by byl obrázek moc malý a nečitelný.

	[Měřená hodnota]	[Typ hodnoty]	[Název jídla]	[Druh jídla]
1	11.1	fat	lehká bílková omeleta se šunkou	Hotová jídla
2	1.17	fat	semínka sezamová	Ořechy a semena
3	1.17	fat	semínka sezamová	Ořechy a semena
4	6.958	fat	olej olivový	Tuky
5	4.98	fat	olej řepkový	Tuky

Obrázek 16 - SELECT prvních 5 záznamů dle kódu, 1. část

Zdroj: vlastní tvorba, obrázek z program DataGrip

Pohlaví	[Rok narození]	[Doba jídla]	created_date
F	1976	vecere	2018-02-05
F	1976	vecere	2018-01-07
F	1976	vecere	2018-03-02
F	1976	vecere	2018-01-11
F	1976	vecere	2018-01-13

Obrázek 17 - SELECT prvních 5 záznamů dle kódu, 2. část

Zdroj: vlastní tvorba, obrázek z program DataGrip

9 Shrnutí výsledků

Cílem práce bylo seznámení se s teoretickou problematikou Business Intelligence, konkrétně s jednou z jejích částí, s datovými sklady, následné navržení a implementace datového skladu na základě dat z databáze firmy Dine4Fit.

V počátku se práce zabývá popisem Business Intelligence jako takové a poté se zaměřuje na jednotlivé její podkategorie. Databáze a jejich dělení na transakční a analytické, problematika datový skladů i s jejich komponentami, ETL procesy. Následně jsou probrána schémata datových skladů, také metody jejich budování a historizace dat v nich. Zmínka je zde i o OLAP kostce, která poté s daty z datových skladů následně pracuje.

Další fází bylo navržení datového skladu pro aplikaci Kalorické tabulky. Data pro tuto část bakalářské práce byla stažena pomocí softwaru DataGrip. Během stahování dat se vyskytly problémy s omezením doby stahování, a tak data musela být stahována v postupných dávkách. Dalším problém byla také malá kapacita lokálního SSD disku, kvůli kterému bylo nutné omezit rozpětí dat pouze za období prvního půlroku roku 2018. Následně byla vytvořena lokální databáze, kam byla nahrána potřebná data z vybraných tabulek. Některé tabulky však obsahovaly neúplná data a byla potřeba je doplnit. V této lokální Microsoft SQL databázi byla data očištěna a transformována – doplnění některých prázdných sloupců, které by datový sklad neměl obsahovat. Dalším krokem bylo vybudování druhé databáze, která splňovala hvězdicovitý tvar a obsahovala jednu tabulku faktů a šest tabulek různých dimenzí. Tato databáze byla vybudována tak, aby umožňovala sledování stravovacích návyků jednotlivých uživatelů. Data do ní byly nahrány pomocí SQL skriptů z předem vytvořené a upravené lokální databáze.

Ve výsledné hvězdnicové databázi tak lze třídit záznamy dle uživatelů, jež je vytvářejí, datumu, denního období zápisu, hledaného názvu nebo typu jídla či dle konkrétní makroživiny, mikroživiny nebo přijaté energie.

10 Závěr

V dnešním světě, kdy informace hrají významnou roli na poli obchodu, marketingu i managementu, je nutné držet krok s dobou. Proto spousta firem investuje do vlastních datových skladů a Business Intelligence se rozvíjí. Tato práce mi ukázala, jakým stylem se orientovat a řídit při tvorbě jednoduššího datového skladu na konkrétních datech získaných z praxe. V práci jsem sám zjistil, jak datový sklad vytvořit a jak poslouží i následně tvorbám různorodých analýz. Tato práce může pomoci čtenáři nastínit, jakým způsobem lze vytvořit datový sklad a napomůže mu s ujasněním pojmů z oblasti BI. Myslím, že každá firma, jež shromažďuje data o svých obchodech, zaměstnancích i zákaznících, by měla mít nejen databázi plnou dat, ale kvalitně vybudovaný datový sklad, který jí díky dataminingovým a jiným operacím přinese mnoho užitečných informací.

11 Seznam použité literatury

[1] L. GÁLA, J. POUR a Z. ŠEDIVÁ, *Podniková informatika. 2., přeprac. a aktualiz. vyd.*, Praha: Expert (Grada), 2009., ISBN 978-80-247-2615-1.

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

[3] ORACLE CORPORATION, *Co je relační databáze.* Oracle [online]. 2020 [cit. 2020-01-25]. Dostupné z: https://www.oracle.com/cz/database/what-is-a-relational-database/

[4] ORACLE CORPORATION , *Data Warehousing Logical Design.* Oracle [online]. 2017 [cit. 2020-01-25]. Dostupné z: https://docs.oracle.com/database/121/DWHSG/ch2logdes.htm#DWHSG9311

[5] TERADATA CORPORATION , *Data warehousing - schemas.* Teradatapoint [online]. 2020 [cit. 2020-02-03]. Dostupné z: https://www.teradatapoint.com/data-warehousing-schemas

[6] COMPARE BUSINESS PRODUCTS , *Top 10 Largest Databases in the World.* Compare business products [online]. 17.3. 2010 [cit. 2020-02-03]. Dostupné z: https://www.comparebusinessproducts.com/fyi/10-largest-databases-in-the-world

[7] PEDERSEN T. *Multidimensional Modeling*. Encyclopedia of Database Systems. Springer. [online]. 2009 [cit. 2020-02-03]. Dostupné z: https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_229. Boston, MA

[8] SANSU GEORGE , *Inmon or Kimball: Which approach is suitable for your data warehouse?. Computerweekly* [online]. 14.2. 2012 [cit. 2020-02-03]. Dostupné z: https://www.computerweekly.com/tip/Inmon-or-Kimball-Which-approach-is-suitable-for-your-data-warehouse

[9] GURU99 , *ETL (Extract, Transform, and Load) Process. Guru99* [online]. 2020 [cit. 2020-02-06]. Dostupné z: https://www.guru99.com/etl-extract-load-process.html

[10] R. KIMBALL, M. ROSS, *The Data Warehouse Toolkit Second Edition,* New York: John Wiley & Sons, Inc., 2002., ISBN 0-471-20024-7

[11] HUMPRIES a kol., *Data warehousing návrh a implementace*, Praha: Computer Press, 2002,. ISBN 80-7226-560-1

[12] MICROSOFT, *Přehled datových krychlí OLAP Service Manageru pro pokročilou analýzu. Microsoft Docs* [online]. 6.5. 2019 [cit. 2020-02-06]. Dostupné z: https://docs.microsoft.com/cs-cz/system-center/scsm/olap-cubes-overview?view=sc-sm-2019

[13] ZENTUT , *Fact Table. ZenTut* [online]. 2020 [cit. 2020-02-06]. Dostupné z: https://www.zentut.com/data-warehouse/fact-table/

[14] GURU99 , *What is OLAP (Online Analytical Processing): Cube, Operations & Types. Guru99* [online]. 2020 [cit. 2020-02-06]. Dostupné z: https://www.guru99.com/online-analytical-processing.html

[15] JETBRAINS , *DataGrip.* JetBrains [online]. 2020 [cit. 2020-03-30]. Dostupné z: https://www.jetbrains.com/datagrip/

[16] ŠEDA JAN , *Úvod do JDBC. Interval* [online]. 4.3.2003 [cit. 2020-03-30]. Dostupné z: https://www.interval.cz/clanky/uvod-do-jdbc/

[17] JAVATPOINT , *Java JDBC Tutorial.Java T Point* [online]. 2018 [cit. 2020-03-30]. Dostupné z: https://www.javatpoint.com/java-jdbc

[18] KALORICKÉ TABULKY. Kalorické Tabulky [online]. 2020 [cit. 2020-03-30]. Dostupné z: https://www.kaloricketabulky.cz/user/diary

[19] ROZENBERG T., *Is Data Warehousing Dead?* [online]. 25.1.2018 [cit. 2020-02-03] Dostupné z: https://medium.com/@tamiro/is-data-warehousing-dead-727757b0c424 nahoře

[20] CHAFIK A., *What does a Data Mart contain?* [online]. 31.7.2016 [cit. 2020-02-03] Dostupné z: https://www.quora.com/What-does-a-Data-Mart-contain

[21] PANOPLY, *Data Mart vs. Data Warehouse* [online]. 2012 [cit. 2020-02-08] Dostupné z: https://panoply.io/data-warehouse-guide/data-mart-vs-data-warehouse/

[22] VITHAL S., *Different Extraction Methods in Data Warehouse* [online]. 28.2.2018 [cit. 2020-02-08] Dostupné z: https://dwgeek.com/different-extraction-methods-data-warehouse.html/

[23] GURU99, *What is Multidimensional schema?* [online]. 2020 [cit. 2020-03-03] Dostupné z: https://www.guru99.com/star-snowflake-data-warehousing.html

[24] OLAP, *WHAT IS THE DEFINITION OF OLAP?* [online]. 2020 [cit. 2020-03-03] Dostupné z: https://olap.com/olap-definition/

[25] GALAKTIKASOFT, *OLAP Operations in Data Mining* [online]. 25.1.2018 [cit. 2020-03-03] Dostupné z: http://wappreview.de/tino/projekt_handel_III/datawrhs/slice.html

UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Akademický rok: 2018/2019

Studijní program: Systémové inženýrství a informatika
Forma studia: Prezenční
Obor/kombinace: Informační management (im3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení:	**Juraj Jankovič**
Osobní číslo:	**I1700631**
Adresa:	Hornoměstská 373, Trutnov – Dolní Předměstí, 54101 Trutnov 1, Česká republika
Téma práce:	Návrh datového skladu pro analýzy databázové aplikace Kalorické tabulky
Téma práce anglicky:	Data Warehouse concept for analyses of database application Kalorické tabulky
Vedoucí práce:	Ing. Barbora Tesařová, Ph.D. Katedra informatiky a kvantitativních metod

Zásady pro vypracování:

Cílem bakalářské práce je navržení datového skladu pro analýzy nejvýznamnějších trendů v konzumaci potravin.

Úvod

- Cíl práce

Teoretická část

- Business Intelligence
- Datový sklad
- ETL procesy
- Historizae dat
- Dimenzionální modelování v datovém skladu
- Technologie OLAP

Praktická část

- Návrh a implementace datového skladu

Seznam doporučené literatury:

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.
HUMPHRIES, Mark. *Data warehousing: návrh a implementace*. Praha: Computer Press, 2002. Databáze. ISBN 80-7226-560-1.

Podpis studenta: Datum:

Podpis vedoucího práce: Datum: