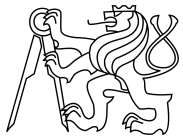
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Integrace systému V3S do datového skladu ČVUT
Student:	Bc. Michal Štádler
Vedoucí:	Ing. Stanislav Kuznetsov
Studijní program:	Informatika
Studijní obor:	Znalostní inženýrství
Katedra:	Katedra teoretické informatiky
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

1) Seznamte se s problematikou datového skladu a proveďte rešerši používaných architektur.
2) Seznamte se s daty ze zdrojového systému v3s.cvut.cz .
3) Navrhněte datový model databáze s integrovanými daty (tzv. integrated data layer) na základě dat zdrojového systému.
4) Tento datový model využijte pro vytvoření vrstvy s integrovanými daty.
5) Pro účely analýzy pomocí technologie OLAP navrhněte přístupovou vrstvu (tzv. access layer) datového skladu ČVUT s vhodnými datovými tržišti (tzv. data marts).
6) Vytvořte přístupovou vrstvu a alespoň na některých datových tržištích demonstrujte využití technologie OLAP a shrňte výhody a nevýhody takto prezentovaných dat.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 24. ledna 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra teoretické informatiky

Diplomová práce

Integrace systému V3S do Datového skladu ČVUT

Bc. Michal Štádler

Vedoucí práce: Ing. Stanislav Kuznetsov

9. května 2017

Poděkování

Rád bych poděkoval vedoucímu mé diplomové práce Stanislavu Kuznetsovi za podporu, pozitivní přístup a podnětné nápady. Dále bych rád poděkoval Janu Dvořákovi za konzultace kolem V3S, Robertovi Kotlářovi a Jakubu Krejčímu za konzultace kolem Datového skladu ČVUT, Michalu Valentovi za získání přístupu k V3S a Karlovi Kloudovi za uvedení do problematiky vědeckých výsledků. Také bych rád poděkoval rodičům a přátelům za obrovskou podporu během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 9. května 2017 .....

Odkaz na tuto práci

Štádler, Michal. *Integrace systému V3S do Datového skladu ČVUT.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Práce obsahuje analýzu systému V3S pro účel jeho integrace do Datového skladu ČVUT. Dále popisuje návrh a realizaci stage, integrované, sémantické, přístupové a prezentační vrstvy nově integrovaných dat z V3S do Datového skladu ČVUT. Závěr práce obsahuje několik příkladů reportů založených na datech z V3S, které kromě svého vlastního významu i ověřují funkčnost integrace.

Klíčová slova Datový sklad, datové tržiště, normalizovaný model, dimenzionální, model, ETL, SCD, reporty, Kettle, V3S, OLAP, Inmon, Kimball

Abstract

This thesis contains analysis of the System V3S for the purpose of its integration into the CTU Data Warehouse. The thesis also describes a solution concept and realization of stage, integrated, semantic, data access and data presentation layers of newly integrated data from V3S into the CTU Data Warehouse. The conclusion of the thesis includes several examples of reports, based on the data from V3S, which besides their own analytical meaning test the functionality of the V3S integration into the CTU Data Warehouse.

Keywords Data warehouse, data mart, normalized model, dimensional model, ETL, SCD, reports, Kettle, V3S, OLAP, Inmon, Kimball

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Myšlenku datového skladu na ČVUT poprvé otevřel v roce 2013 Stanislav Kuznetsov svou diplomovou prací s názvem Datový sklad fakulty[1]. Od té doby prošel projekt zásadním vývojem čítajícím tisíce hodin práce a zároveň byl předmětem několika závěrečných prací. Soustavná práce tak přinesla konkrétní výsledky a v současné chvíli je již v Datovém skladu ČVUT integrováno několik systémů, například KOS, Anketa ČVUT a Systém závěrečných prací (ZP). Další systémy jsou navíc v různých fázích procesu integrace (Portál spolupráce s průmyslem, Edux a Progtest). Integrace Aplikace na evidenci výsledků vědy a výzkumu (V3S) tak přirozeně zapadá do systematického rozvoje Datového skladu ČVUT.

Práce si klade za cíl integrovat V3S do Datového skladu ČVUT a rozšířit ho tak o další datový zdroj. Ambice je provést integraci v celé délce datového cyklu, tedy od prvotní analýzy zdroje dat, přes integrovanou vrstvu, datová tržiště, až po otestování řešení skrz vygenerované reporty.

Důvod pro integraci V3S do Datového skladu ČVUT je relativně přímočarý: činnost ČVUT je obecně rozdělena na vzdělávací a výzkumnou, v současné verzi ale Datový sklad reflektuje pouze vzdělávací část poslání ČVUT. Data o výzkumné činnosti na ČVUT v Datovém skladu úplně chybí. Integrace V3S má ambici tento stav napravit.

Po zaintegrování V3S do Datového skladu ČVUT bude možné nad daty z V3S provádět daleko podrobnější analýzy, než které nabízí současné webové rozhraní dostupné z `https://v3s.cvut.cz`. Díky technologii OLAP bude možné nad daty provádět analytické operace typu „data drilling“ a „slice and dice“. Dalším významným benefitem pak bude možnost analýzy dat z různých kombinací zdrojových systémů. Příkladem takové kombinace je spojení V3S dat o vědecké činnosti vědců na ČVUT s jejich lektorskou činností, které umožní získat komplexní metriku pro hodnocení akademických pracovníků na ČVUT.

Práce je rozdělena na teoretickou a praktickou část. V teoretické části se čtenář seznámí s problematikou datového skladování, vyjasní si termino-

logii a dozví se vše potřebné o základních architekturách datových skladů. Praktická část pak čtenáře provede analýzou zdrojového systému a všemi vrstvami Datového skladu ČVUT v části týkající se V3S. Nejzajímavější budou pro čtenáře zřejmě pasáže o integrované vrstvě a o prezentační vrstvě. Prezentační vrstva s reporty postavenými nad technologií OLAP zároveň slouží k otestování funkčnosti integrace V3S.

Část I

Teoretická část

Kapitola 1

Základní architektury datových skladů

Mnoho organizací dnes buduje své datové sklady pro ukládání velkého množství dat s časovým kontextem, které chtějí dále použít pro podporu rozhodování nebo jiné analytické úkoly. Tyto organizace mají mnoho možností technické realizace, a proto je nutné se nejdříve seznámit se základními koncepty datového skladování. Během posledních dvou až tří dekád se na poli problematiky datových skladů ustálily dva hlavní přístupy, oba pojmenované po jejich původních autorech: Inmonův datový sklad (William H. Inmon) a Kimballův datový sklad (Ralph Kimball).

Obě architektury sdílí mnoho shodných principů, včetně obecné myšlenky datového skladu: datový sklad je centrální repositář atomických dat získaných průřezově přes celou firmu nebo obecně oblast zájmu. Datový sklad tak reprezentuje kompletní a důvěryhodný pohled na informace potřebné k běhu, pochopení a rozvoji celé organizace.

William Inmon navíc prosazuje takzvaný přístup shora (top-down), který adaptuje nástroje tradiční relační databáze na potřeby vývoje celofiremního datového skladu. Z tohoto celofiremního datového skladu se pak vyvinou jednotlivé databáze oddělení, které následně slouží pro podporu rozhodování.

Ralph Kimball na druhou stranu navrhuje přístup zespoda (bottom-up), který používá princip dimenzionálního modelování. Dimenzionální modelování je unikání technika vyvinutá pro datové sklady. Místo vybudování jedné společné databáze pro celou firmu Kimball navrhuje vytvořit jednu databázi pro každý hlavní firemní proces. Celofiremní koheze je pak dosažena jinou Kimballovou inovační technikou, tzv. standardem společné datové sběrnice[1].

Oba přístupy jsou podrobněji popsány v následujících kapitolách.

[1] ang. data bus standard

1.1 Inmonův datový sklad

Inmon je v literatuře považován za „otce datových skladů“, protože v počátku devadesátých let jako první definoval pojem datový sklad ve své práci Building the Data Warehouse[2]. Immon ve vydání z roku 2011[3] definuje pojem datový sklad následovně:

> *„Datový sklad je subjektově orientovaná, integrovaná, historizovaná, neproměnlivá a granulovaná kolekce dat sloužící pro podporu rozhodování.“*

Tato definice si zaslouží další komentář, který je rozveden na následujících řádcích.

1.1.1 Subjektová orientace

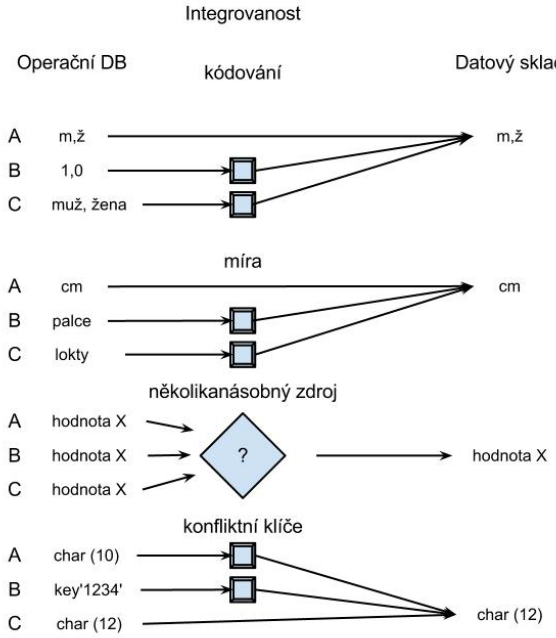
Datový sklad je orientovaný podle hlavních oblastí definovaných ve vysokoúrovňovém datovém modelu organizace a podle klíčových entit, o které se firma zajímá. Typicky může jít o oblast zákazníků, produktů, transakcí atd., přičemž každá oblast může obsahovat desítky, stovky, nebo i více spolu spjatých fyzických tabulek. Z tohoto vysokoúrovňového modelu se následně odvodí detailní logické modely pro každou klíčovou entitu. Důležité je, že struktura entit je budována v normalizované formě, což zamezuje redundanci dat a vzniku datových anomálií.

1.1.2 Historizace

Datový sklad udržuje historická data klidně i roky dozadu. Data se v datovém skladu nikdy nemažou ani nemění, čímž se zásadně liší od transakčních systémů, které typicky uchovávájí pouze aktuálně platná data. Historický kontext se v datových skladech udržuje pomocí příznaků „platnost od“ a „platnost do“, čímž vznikne sekvence obrazů dat ze zdrojového systému.

1.1.3 Integrace (homogenita)

Do datového skladu typicky vstupuje několik (nehomogenních) datových zdrojů, které datový sklad integruje a přetváří je na homogenní vrstvu přehledných a důvěryhodných informací. Tyto datové zdroje mohou být jak interní (například firemní CRM, ERP), tak i externí (například veřejně dostupné informace Českého statistického úřadu). Integrace je v datovém skladu klíčová pro zajištění správnosti a jednotnosti dat napříč datovým skladem. Příklad integrace několika entit ukazuje obrázek 1.1.



Obrázek 1.1: Příklad integrace několika entit do datového skladu

1.1.4 Neproměnlivost

Ukládání dat zpravidla probíhá v dávkách a reprezentuje kopii zdrojového systému v nějakém konkrétním čase. Datový sklad udržuje data beze změn. Jakmile jsou data nahrána, jediný přípustný mechanismus změny dat v datovém skladu je tzv. historizace: vytvoření další kopie zdrojových dat a jejich úprava podle platnosti jednotlivých záznamů. Tento mechanismus je podrobně popsán v sekci věnované historizaci dat.

1.1.5 Granularita

Otázka granularity obvykle bývá jednou z nejdůležitějších otázek při navrhování designu datového skladu, protože přímo ovlivňuje celou architekturu prostředí datového skladu. Granularita nám říká, do jakého detailu datový sklad informace uchovává: čím vyšší detail, tím nižší stupeň granularity. Čím nižší detail, tím větší stupeň granularity. Příklad si lze vypůjčit třeba z bankovního prostředí: jednotlivé finanční transakce reprezentují nízkou granularitu, měsíční souhrn transakcí naopak granularitu vysokou. Granularita tedy velmi ovlivňuje množství dat uložených v datovém skladu i otázky, které datový

sklad může zodpovědět. Cílem je tedy najít vhodnou rovnováhu mezi potřebnou úrovní detailu odpovědí a nároky na technologii.

1.1.6 Architektura

Inmon zastává tzv. „přístup shora“ (v literatuře označovaný jako „top-down“[2]). Ten spočívá ve vytvoření centralizovaného datového skladu pokrývajícího celý podnik a až následného budování jednotlivých databází, které jsou přizpůsobeny konkrétním oddělením. Takovýmto databázím Inmon říká datová tržiště[3].

Postupné budování datového skladu s datamarty Inmon[3] nazývá „den-1-den-n fenomén“ (Day 1-Day n Phenomenon). Inmon dále vysvětluje, že datový sklad by se neměl stavět najednou, ale jednotlivé subjekty by se měly přidávat postupně. Postup je ukázaný na obrázku 1.2. Začíná se pouze s operačními systémy/databázemi, následně se začnou integrovat první tabulky nějakého konkrétního subjektu a datový sklad začne mít první uživatele. Dále se přidávají nové datové zdroje, které se integrují do současných struktur datového skladu a začíná se otevírat prostor pro analytické zpracovávání dat. V další fázi začínají vzkvétat datová tržiště (databáze jednotlivých oddělení), která se postupně stanou nejefektivnější, nejrychlejší a nejjednodušší cestou k získání a analýze dat.

Inmon dále navrhuje, aby byl integrovaný datový sklad v normalizovaném datovém modelu a z něj byla odvozena datová tržiště vytvořená podle dimenzionálního přístupu, který vysvětlím v následující kapitole. Pohled z ptačí perspektivy na Inmonův integrovaný datový sklad lze nalézt na obrázku 1.3. Inmon tedy dělí datový sklad na následujích 5 vsrtev.

1.1.6.1 Zdrojové systémy (data source)

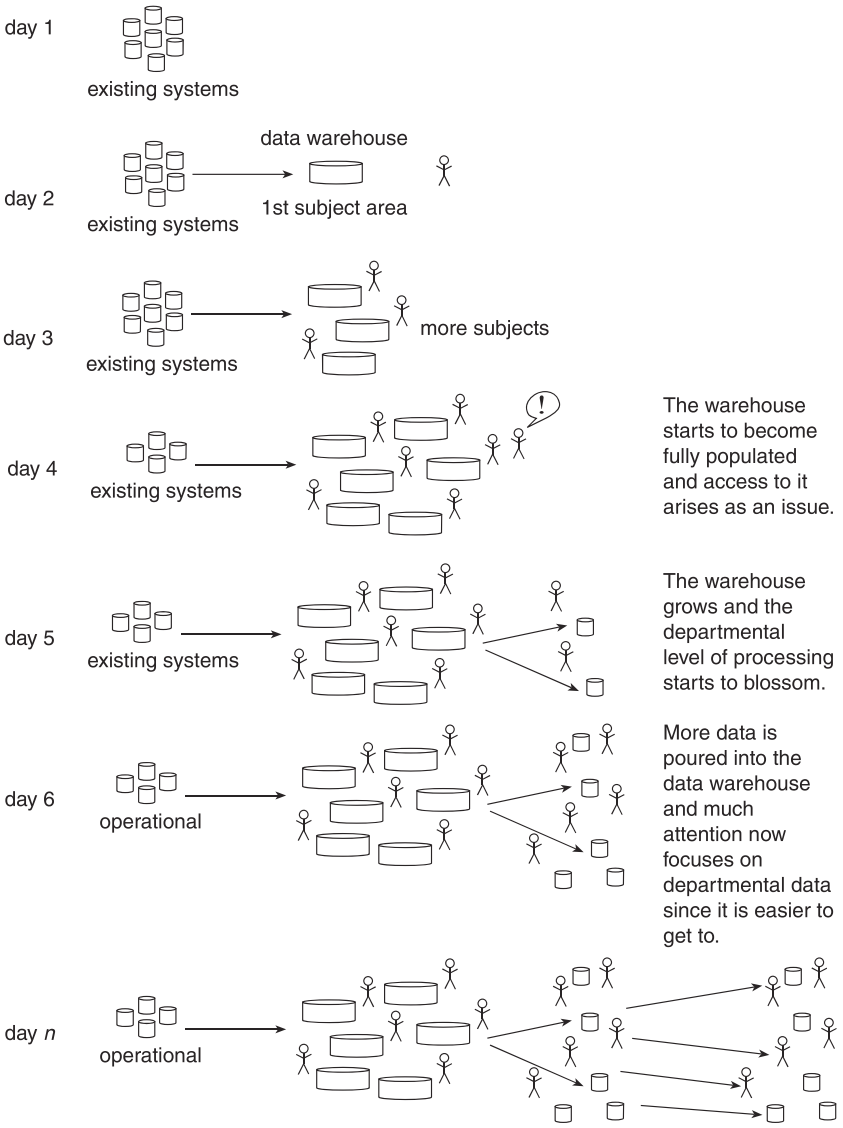
Ačkoli je vrstva zdrojových systémů zahrnuta ve schématu, součástí datového skladu není. Zdrojové systémy jsou totiž operační databáze, ze kterých datový sklad čerpá data.

1.1.6.2 Stage (staging area)

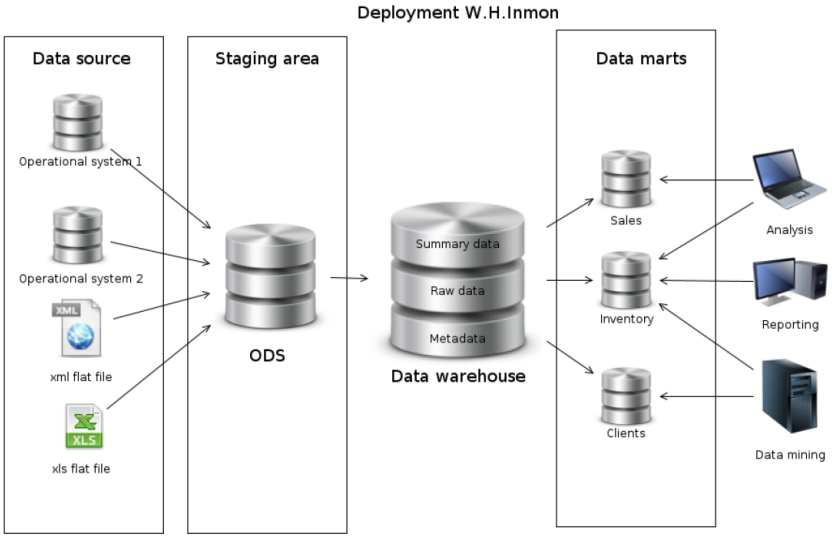
První vrstva, která je fyzicky součástí datového skladu. Do stage se kopírují data 1:1 ze zdrojových systémů (proces extrakce dat). Dále zde probíhá čištění, kombinace a standardizace dat (proces transformace dat) a jejich následné nahrání do integrované vrstvy (proces nahrání dat). Tyto tři procesy jsou souhrnně označovány zkratkou ETL a jsou dále popsány ve vlastní kapitole.

[2] https://en.wikipedia.org/wiki/Data_warehouse

[3] datovým tržištím se také říká datové marty nebo datamarty



Obrázek 1.2: Inmonův den-1-den-n fenomén



Obrázek 1.3: Integrovaný datový sklad podle W. H. Inmona

1.1.6.3 Integrovaná vrstva (data warehouse)

Úkolem této vrstvy je uložit a držet data správně organizovaná tak, aby nad nimi bylo dále možné vytvářet analýzy nebo další datové struktury s businessovým významem. Obecně se jedná o nejdůležitější část Inmonovy architektury.

Mezi základní principy integrované datové vrstvy patří především následující vlastnosti.

- Centralizovaná databáze: všechna data ze zdrojových systémů obecně mohou být uložena různými, mezi sebou nekonzistentními způsoby. Mohou mít i různé definice stejného objektu, mohou být organizovaná v jiné logické struktuře. Úkolem centralizované databáze je všechny tyto objekty sjednotit a uložit do jedné fyzické databáze.

- Detailní data s historickým kontextem. Integrovaná vrstva udržuje historický kontext dat, viz sekce o historizaci.

- Obecné datové struktury nezávislé na aplikaci nebo dotazu: Integrovaná vrstva nemodeluje business procesy, ale vztahy mezi datovými prvky, ve tvaru blízkém třetí normální formě.

- Škálovatelnost modelu: Existující datové struktury je možné rozšiřovat bez nutnosti jejich reorganizace či změn.

1.1.6.4 Přístupová vrstva (data marts)

Přístupová vrstva přebírá datové struktury z integrované vrstvy a vytváří z nich entity s businessovým významem. Dělí se na 2 části, sémantickou databázi/vrstvu a datová tržiště (datamarty). Sémantická databáze obsahuje znovu použitelné stavební bloky, které obsahují fundamentální businessové termíny nebo jejich hlavní elementy. Struktura sémantické databáze může a nemusí být denormalizovaná. Struktury sémantické databáze se pak používají pro vytváření datamartů ve hvězdicovém nebo vločkovém schématu (popsáno dále).

1.1.6.5 Prezentační vrstva

Úkolem prezentační vrstvy je abstrahovat informace z dat a předat je business uživateli v pro něj stravitelné formě, například přes reporty, analytické nástroje, nebo třeba vygenerované datasety.

1.2 Kimballův datový sklad

Další významnou osobou v prostředí datových skladů je Ralph Kimball, který definoval koncept datových tržišť spolu s pojmy „star“ (hvězdicové schéma) a „snowflake“ (vločkové schéma) pro použití v dimenzionálním modelování.

Kimball definuje datový sklad následovně[4]:

> *„Datový sklad je kopií transakčních dat specificky strukturovaných pro dotazování a reportování.“*

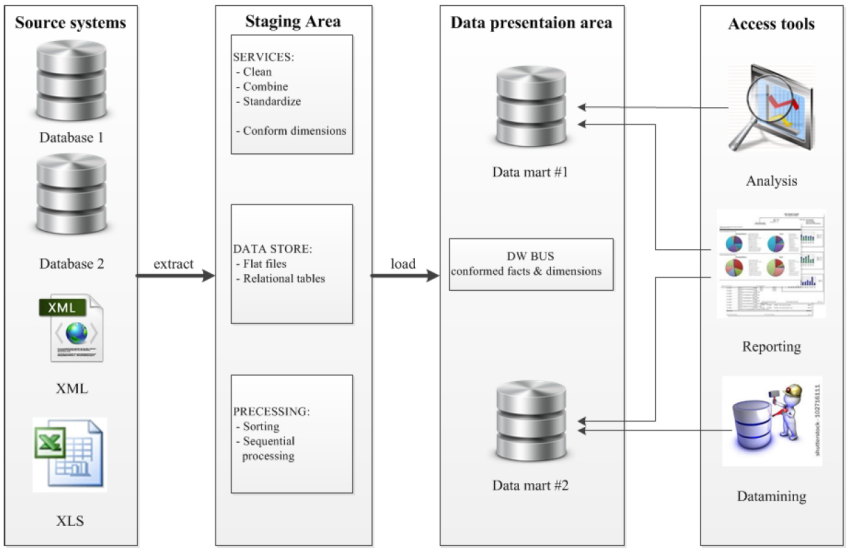
Architektura Kimballova datového skladu je zachycena na obrázku 1.4 a zahrnuje 4 vrstvy.

1.2.1 Zdrojové systémy (source systems)

Ačkoli je vrstva zdrojových systémů zahrnuta ve schématu, součástí datového skladu není. Zdrojové systémy jsou totiž operační databáze, ze kterých datový sklad čerpá data.

1.2.2 Stage (staging area)

První vrstva, která je fyzicky součástí datového skladu. Do stage se kopírují data 1:1 ze zdrojových systémů (proces extrakce dat). Dále zde probíhá čištění, kombinace a standardizace dat (proces transformace dat) a jejich následné nahrání do prezentační vrstvy (proces nahrání dat). Tyto tři procesy jsou souhrnně označovány zkratkou ETL a jsou dále popsány ve vlastní kapitole.



Obrázek 1.4: Integrovaný datový sklad podle Kimballa

1.2.3 Prezentační vrstva (data presentation area)

Oblast, kde se nacházejí datová tržiště. Datová tržiště jsou databáze přizpůsobená potřebám jednotlivých oddělení ve firmě. Data jsou v datamartech uchována atomická s respektem k dimenzionálnímu přístupu. Používá se tzv. „společná sběrnicová architektura“ (data warehouse bus architecture), která umožňuje inkrementální rozvoj datového skladu, potažmo celého BI. Společná sběrnicová architektura dekomponuje plánovací proces datového skladu do lehce zvládnutelných kousků odpovídajících businessovým procesům ve firmě a využívá tzv. „odpovídající dimenze“ (conformed dimensions). Odpovídající dimenze jsou společné, standardizované dimenze, které jsou jednou zpracovány ETL a následně znovu použity více faktovými tabulkami[4], díky čemuž poskytují konzistentní a deskriptivní atributy napříč dimenzionálními modely i možnost integrovat data z různých businessových procesů. Díky tomu odpovídající dimenze eliminují redundanci v designu, zjednodušují vývoj a ultimátně zkracují čas implementace.

1.2.4 Přístupové nástroje (access tools)

Vrstva ležící mimo datový sklad zahrnující nástroje pro analýzu, datamining a reportování dat z datových tržišť.

Kimball tedy doporučuje datový sklad stavět „odspodu“ (bottom-up): začít s budováním jednotlivých datamartů sloužících jednotlivým oddělení s tím,

[4]problematika faktových tabulek je dále vysvětlena v následující kapitole

že se na konci spojí pomocí sběrnicové architektury[5].

1.3 Srovnání Inmonovi a Kimballovy architektury

Nejprve se podíváme na srovnání základních rozdílů v přehledné tabulce [6].

	charakteristika Kimballa	**charakteristika Inmona**
požadavky datové integrace	individuální businessové požadavky	integrace přes celou firmu
struktura dat	měřitelná data (KPI, metriky výkonu firmy)	různé typy dat
perzistence dat ve zdrojových systémech	stabilní zdrojové systémy	zdrojové systémy se rychle mění
potřebné znalosti	malý tým s obecnými znalostmi	větší tým specialistů
časové omezení	urgentní potřeba pro první datový sklad	velkorysejší časový rámec pro splnění požadavků businessu
náklady na vybudování	ze začátku nízké	ze začátku vysoké

Tabulka 1.1: Tabulka základních rozdílů Kimballovy a Inmonovy architektury

Na následujících řádcích jsou popsány vybrané výhody a nevýhody obou architektur podrobněji. Srovnání kromě Inmona[3] a Kimballa[4] čerpá i z článku Mary Breslin z Business Intelligence Journal[7].

1.3.1 Hlavní výhody Inmonovi architektury

Mezi hlavní výhody Inmonovi architektury patří především níže zmíněné body.

- Datový sklad slouží jako jediný a důvěryhodný zdroj pravdy v celé firmě, protože jsou všechna data v datovém skladu integrovaná. Integrovaná data jsou zároveň jediným zdrojem pro datová tržiště.

- Netvoří se datové anomálie, protože je tu minimum redundance. To znamená i zjednodušení ETL.

- Firemní procesy mohou být jednoduše pochopeny, protože logický model reprezentuje detailní firemní entity.

- Flexibilita: když se firemní požadavky nebo zdrojový systém změní, je nutné změnu v datovém skladu udělat pouze na jednom místě.

1.3.2 Hlavní nevýhody Inmonovi architektury

Mezi hlavní nevýhody Inmonovi architektury patří především níže zmíněné body.

- Model a implementace může být časem komplexní a vyžaduje velké množství tabulek a tabulkových spojení.
- Potřeba většího množství expertních pracovníků v datovém modelování se znalostí firemních procesů. Tyto zdroje mohou být velmi drahé a náročné na nalezení.
- Prvotní náklady na vybudování jsou vysoké, takže je nutná ochota a trpělivost vedení firmy.
- ETL je složitější, protože datamarty si berou data z integrované vrstvy datového skladu.

1.3.3 Hlavní výhody Kimballovy architektury

Mezi hlavní výhody Kimballovy architektury patří především níže zmíněné body.

- Rychlé zavedení datového skladu, první výsledky jsou viditelné velice rychle.
- Hvězdicové schéma je pochopitelné i businesss uživateli a je jednoduše použitelné pro reportování.
- Správa celého systému je jednoduchá.
- Výkonnost hvězdicového schématu je velmi dobrá. Databáze bude často provádět „hvězdicové spojení“ (star join), kde se vytváří kartézský produkt za použití všech dimenzionálních hodnot, které se pak následně spojí s faktovými tabulkami. Tento způsob je v databázích obecně velice efektivní.
- Je potřeba pouze malý tým vývojářů.
- Funguje velmi dobře pro metriky v rámci oddělení a KPI, protože datamarty jsou navrhovány s ohledem na konkrétní potřeby jednotlivých oddělení.

1.3.4 Hlavní nevýhody Kimballovy architektury

Mezi hlavní nevýhody Kimballovy architektury patří především níže zmíněné body.

- Neexistence autoritativní pravdy, protože datový sklad není plně integrovaný.
- Redundantní data mohou způsobit datové anomálie.
- Přidávání sloupců do faktových tabulek může způsobit výkonnostní problémy, protože faktové tabulky jsou navržené velmi hluboce. Když se přidá nový sloupec, velikost faktové tabulky se o hodně zvýší, čímž trpí výkonnost a dimenzionální model se tak špatně přizpůsobuje změnám.
- Nezvládne všechny reportovací potřeby podniku, protože datový model je orientovaný směrem k firemním procesům místo orientace na firmu jako celek.

Kapitola 2

Popis datového skladu

Následující kapitola se zabývá základními pojmy týkajících se datového skladu, vysvětlením principu dimenzionálního modelování a také v datových skladech velmi známé zkratce ETL (extract, transform, load).

2.1 Cíle datového skladování

Jedním z nejdůležitějších aktiv každé firmy jsou informace. Tyto informace jsou téměř vždy využity ke dvěma různým účelům: jednak k operativě potřebné pro každodenní práci, jednak k analýze pro podporu rozhodování. Jednoduše řečeno, operační systémy jsou místa, kam data vkládáme, datový sklad/BI prostředí je naopak místo, kde data vybíráme. Lidé okolo operačních systémů otáčí kolo organizace, mají na starost objednávky, zákazníky nebo produkty. Jejich databáze jsou orientované pro rychlé a paralelní zpracovávání aktuálních transakčních dat. Naopak uživatelé datového skladu pozorují, jak se kola organizace otáčejí a vyhodnocují jejich kondici. Počítají, kolik nových objednávek firma má a porovnávájí je s minulostí nebo vyhodnocují stížnosti zákazníků. Zajímají se o to, jestli operativní procesy fungují správně. Jejich databáze jsou orientované na analytické dotazování, tedy na často komplikované, jednorázové, nestandardní, do historie sahající a mnoho dat zpracovávající dotazy, které odpovídají na konkrétní analytický dotaz. Hlavní důvody pro pořízení datového skladu jsou následující[4].

- Společnost sbírá mnoho dat, ale neumí je číst.
- Společnost neumí data rozkouskovat na menší části a dále analyzovat z různých úhlů pohledu (tzn. „slice and dice“[5]).
- Business uživatelé se potřebují jednoduše dostat k datům.

[5] popsáno v části o technologii OLAP

- Neexistence autoritativní pravdivé informace (každé oddělení reportuje jiné odpovědi na stejnou otázku).
- Lidé chtějí tvrdá fakta na podporu rozhodování.

Z těchto business požadavků volně vznikají požadavky na datový sklad.

- Informace v datovém skladu musí být jednoduše dostupné.
- Informace v datovém skladu musí poskytovat konzistentní informace.
- Datový sklad se musí lehce adaptovat na změny.
- Datový sklad musí poskytovat informace včas.
- Datový sklad musí být bezpečný a musí chránit informační aktiva firmy.
- Datový sklad musí sloužit jako autoritativní zdroj informací.
- Komunita business uživatelů musí datový sklad vnímat jako pomocníka.

2.2 Dimenzionální modelování

Dimenzionální modelování databáze se postupem času stalo široce přijímanou, respektovanou a preferovanou technikou pro uložení analytických dat, a to zejména z následující důvodů:

- Přináší business uživatelům data v dostatečně jednoduché, a tudíž pochopitelné formě.
- Poskytuje rychlou odezvu SQL dotazů díky spojování tabulek přes primární indexy (dotazy přes primární index jsou v databázových systémech obecně velmi efektivní).

2.2.1 Normalizovaný model

Před tím, než si vysvětlíme princip dimenzionálního modelování, je ještě nutné se zmínit o modelu normalizovaném, který se běžně používá pro operační databáze. Normalizace databáze je proces optimalizace pro vkládání, upravování a mazání dat (obecně řečeno optimalizace pro transakční zpracování), jehož cílem je odstranění redundantních dat za pomoci normalizačních pravidel. Obecným rysem normalizačního pravidla je rozkouskovat transakci do co nejvíce tabulek. Pokud následně měníme, vkládáme, nebo mažeme záznam z normalizované databáze, provádíme operaci vždy pouze na jednom místě, což se následně pozitivně projeví na výkonu celého databázového systému.

Celkem se rozlišuje 5 úrovní normalizace, o normalizované databázi už ale hovoříme při třetím stupni (třetí normální forma, 3NF). Vyšší normální formy

jsou vždy podmnožinou nižších, pro přehlednost si připomeneme pouze první tři úrovně, protože vyšší normální formy nejsou z hlediska datového skladování relevantní.

- **1NF:** každý atribut relace obsahuje jen atomické hodnoty.
- **2NF:** 1NF + každý neklíčový atribut relace je plně závislý na primárním klíči, a to na celém klíči a nejen na nějaké jeho podmnožině.
- **3NF:** 2NF + žádný z atributů není tranzitivně závislý na klíči.

Struktury v třetí normální formě jsou tedy díky optimalizaci pro transakční zpracování velmi vhodné pro operační databáze. Naopak pro analytické databáze struktury v třetí normální formě moc vhodné nejsou. Důvod je ukrytý už v samotné nosné myšlence normálních forem, existenci mnoha tabulek, které se musí pro přečtení transakce spojit. To v kombinaci s potřebou velmi komplikovaných analytických dotazů znamená velmi nízkou efektivitu, kterou uživatel pocítí skrz dlouhou odezvu analytických dotazů. Druhou nevýhodou složité struktury normalizované databáze je fakt, že je pro běžného uživatele nepřehledná.

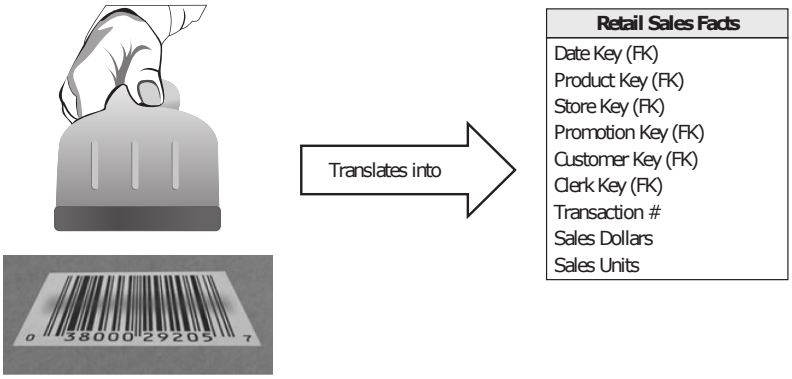
2.2.2 Dimenzionální model

Dimenzionální modelování je technika určená k optimalizaci databází orientovaných na rozsáhlé analytické dotazy. Kimball[4] dokonce tvrdí, že jde o nejlepší techniku prezentování informací uživatelům. Využívá koncept faktových a dimenzionálních tabulek v hvězdicovém nebo vločkovém schématu. Všechny 4 pojmy si teď vysvětlíme.

2.2.2.1 Faktová tabulka

Faktová tabulka v dimenzionálním modelu ukládá míru výkonnosti vycházející z událostí businessových procesů. Kimball říká, že tyto míry výkonnosti zabírají v databázi zdaleka nejvíc prostoru, a proto si zaslouží speciální zacházení. Neměly by být v rámci firmy replikovány a měly by být přístupné všem oddělením ve firmě. To má za následek dva pozitivní jevy, jednak vyšší výkonnost řešení a jednak zaručení konzistence přes celou firmu.

Termín fakt reprezentuje businessovou míru nebo hodnotu. Představme si trh, kde se prodává ovoce a kde se v transakcích zapisuje, kolik jednotek ovoce se prodalo a kolik to stálo. V takovém případě by faktová tabulka prodeje vypadala jako na obrázku 2.1. Každý řádek ve faktové tabulce odpovídá jedné hodnotě (faktu). Data na každém řádku jsou v konkrétní úrovni detailu, kterou nazýváme granularita. Jedním ze základních principů dimenzionálního modelování je, že všechna fakta ve faktové tabulce musí mít stejnou, ideálně co nejmenší granularitu. Nejužitečnější faktové tabulky jsou pak ty, které jako míru mají nějakou numerickou hodnotu.



Obrázek 2.1: Překlad míry firemního procesu do faktové tabulky

2.2.2.2 Dimenzionální tabulka

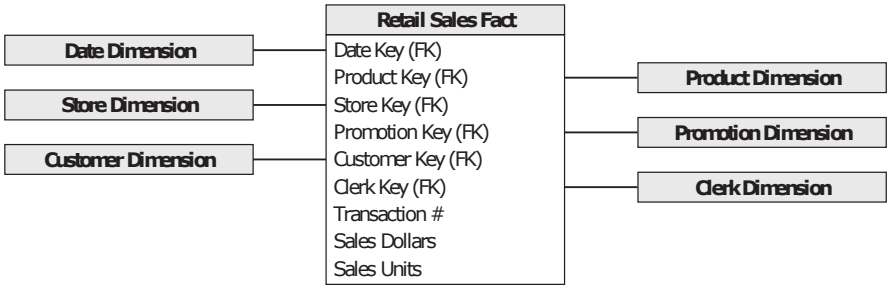
Dimenzionální tabulky obsahují textový kontext asociovaný s událostí firemního procesu a popisují firemní fakta. Dimenzionální tabulky popisují kdo, co, kde, kdy, jak nebo proč asociované s danou událostí. Jak ukazuje obrázek 2.2, dimenzionální tabulka často má velmi mnoho atributů, ani vyšší řád desítek není žádný problém. Dimenzionální tabulky většinou mají méně řádků než faktové tabulky, kromě již zmíněné šířky mohou být ale bohaté na obsah i díky objemným buňkám (například textové pole). Každá dimenze má právě jeden primární klíč, který zajišťuje referenční integritu a spojení k faktové tabulce. Dimenzionální tabulky také slouží jako primární zdroj referenční integrity a popisků pro reporty. Správné hodnoty v dimenzionálních tabulkách jsou základem pro použitelné a srozumitelné business inteligence prostředí.

Product Dimension
Product Key (PK)
SKU Number (Natural Key)
Product Description
Brand Name
Category Name
Department Name
Package Type
Package Size
Abrasive Indicator
Weight
Weight Unit of Measure
Storage Type
Shelf Life Type
Shelf Width
Shelf Height
Shelf Depth
...

Obrázek 2.2: Dimenzionální tabulka produktu

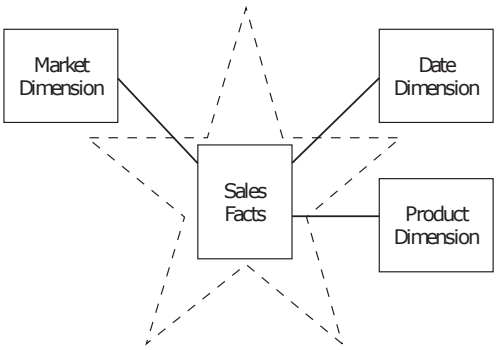
2.2.2.3 Fakta a dimenze spojené v hvězdicovém schématu

Se znalostí dimenzionálních a faktových tabulek se konečně můžeme pustit do tvorby kompletního dimenzionálního modelu. Příklad dimenzionálního modelu je možné též vidět na obrázku 2.3. Tento konkrétní příklad popisuje dříve zmíněný proces na trhu týkající se prodeje ovoce: faktová tabulka obsahující údaje o prodeji neboli míru (cenu a počet kusů) a dále cizí klíče do dimenzionálních tabulek obsahujících textový kontext. Důraz je kladen na to, aby všechny cizí klíče z faktové tabulky měly svůj unikátní primární klíč v příslušné dimenzionální tabulce.



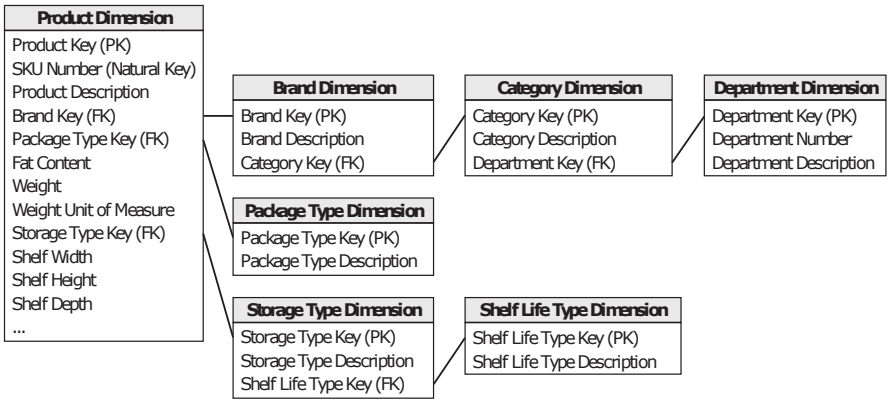
Obrázek 2.3: Faktové a dimenzionální tabulky v hvězdicovém modelu

Takováto konstrukce vypadající jako hvězda 2.4 je v literatuře běžně nazývána jako hvězdicové spojení (star join), případně hvězdicové schéma (star scheme).



Obrázek 2.4: Grafické naznačení hvězdicového schématu

První věc, které si pozorný čtenář hned všimne, je jednoduchost a přehlednost této struktury. Dalším benefitem je přímočará možnost optimalizace takové struktury. Předem totiž víme, přes jaké klíče se budou tabulky nejčastěji spojovat, a proto na ně rovnou můžeme nasadit indexy. Správně vybrané indexy jsou totiž základním požadavkem pro dobře optimalizovanou a efektivně fungující databázi. Poslední velkou výhodou dimenzionálního modelu



Obrázek 2.5: Dimenze produktu ve vločkovém schématu

v hvězdicovém schématu je schopnost dobře se adaptovat na změnu. Všechny dimenze jsou si totiž sobě rovné a zároveň všechny dimenze jsou symetrickým vstupním bodem do faktové tabulky, je tedy snadné nějakou dimenzi přidat nebo upravit. Dimenzionální model také není zaujatý směrem k nějakému konkrétnímu dotazu, což zvyšuje obecnost a flexibilitu modelu.

2.2.2.4 Fakta a dimenze spojené ve vločkovém schématu

Vločkové schéma (snowflake scheme) vychází z hvězdicového schématu a v mnoha případech může být jeho vhodnější alternativou. Rozdíl je v tom, že dimenze ve vločkovém schématu mohou obsahovat tzv. poddimenze. Tyto poddimenze jsou dimenze normalizované do dalších tabulek, takže celé schéma připomíná sněhovou vločku. Kimball říká, že takováto hierarchická struktura dimenzí sice reprezentuje data správně a přesně, ale měli bychom se jí v modelech vyhnout, protože zhoršuje přehlednost a je tedy méně pochopitelná pro business uživatele. Kimball také říká, že vločkové schéma negativně ovlivňuje výkonnost databáze z důvodu nutnosti většího počtu spojení. Mezi kladné vlastnosti vločkového schématu naopak patří nižší míra redundance v datech, užší dimenze (méně sloupců v dimenzionálních tabulkách) a často i lepší podpora ze strany analytických aplikací. Posledním velkým pozitivním vlivem může být i jednodušší transformace dat, protože data jsou ve zdrojových systémech většinou uložena v normalizované podobě a tudíž se lépe mapují na taktéž normalizované poddimenze ve vločkovém schématu. Kimball ukazuje vločkovou strukturu, která je normalizována směrem k třetí normální formě, na příkladu dimenze produktu (obrázek 2.5).

2.3 ETL

Každý datový sklad ke svému fungovaní potřebuje data ze zdrojových systémů. Zdrojový systém si v kontextu datového skladování můžeme představit jako systém mimo naše BI prostřední, který zaznamenává transakční data a nad kterým máme velmi limitovanou nebo dokonce žádnou kontrolu. Hlavními prioritami zdrojového systému jsou orientace na výkon a dostupnost. Dotazy do operační databáze zdrojového systému jsou často přímočaré a většinou pracují pouze s jedním záznamem. Tyto operace jsou součástí běžného transakčního toku, dotazy jsou předem definované, a tedy celkově nevhodné pro analytické potřeby.

Množině procesů pro extrakci dat ze zdrojového systému, jejich transformaci do podoby vhodné pro analytické použití a nahrání do datového skladu se zkráceně říká ETL (extract, transform, load).

2.3.1 Extrakce

Extrakce je prvním krokem procesu přenosu dat ze zdrojových systémů do prostředí datového skladu. Extrakce znamená nejen přečtení dat, ale i jejich porozumění a 1:1 zkopírování do první vrstvy datového skladu, které se v závislosti na zvolené terminologii říká landing nebo staging vrstva. Velmi důležité je zmínit, že od okamžiku potvrzení převzetí dat v landing/staging vrstvě přebírá za data zodpovědnost datový sklad, stává se jejich vlastníkem a má nad nimi plnou kontrolu. Data ze zdrojových systémů mohou mít mnoho podob, od klasických exportů z provozních databází, přes textové a xml soubory, logy, až po soubory pro tabulkové kalkulátory. Ze zdrojového systému typicky neextrahujeme všechna data, ale pouze ta data, která chceme dále analyzovat.

2.3.2 Transformace

Po extrakci dat následuje transformace, která zahrnuje především následující kroky.

- Čištění dat: parsování do standartních formátů, zpracování chybějících hodnot, vyřešení doménových konfliktů.
- Deduplikaci dat.
- Kombinaci dat z různých zdrojů.

Díky těmto krokům mají data vyšší hodnotu a jsou připravená pro nahrání do integrovaného úložiště. V rámci transformace je žádoucí vytvářet i diagnostická metadata, která budou později (v rámci datové kvality) sloužit jako podklad pro zvýšení kvality dat přicházejících ze zdrojového systému.

2.3.3 Nahrání

Nahrání je poslední krok ETL procesu, který zahrnuje fyzické přemapování dat pro uložení do faktových a dimenzionálních tabulek a jejich faktické nahrání do datového skladu. Protože hlavním cílem ETL procesu je předat data ve formě faktových a dimenzionálních tabulek, je tato část ETL vůbec nejkritičtější. Po skončení procesu nahrání dat je vhodné mít implementovanou i kontrolu správnosti nahraných dat.

2.4 Metadata

Jednoduše řečeno, metadata jsou data o datech a slouží jako dokumentace dat. Metadata popisují jak data ve zdrojových systémech, tak i data v datovém skladu. Přenos metadat patří do kompetence ETL procesu. Obecně lze říci, že rozlišujeme 3 druhy metadat[8].

- **Businessová metadata** definují vlastníka dat, businessové slovníky nebo změnovou politiku.
- **Technická metadata** zahrnují jména databázových systémů, tabulek, sloupečků, jejich datové typy, velikosti, nebo povolené hodnoty. Technická metadata taktéž obsahují informace o struktuře dat, jako jsou primární a cizí klíče nebo databázové indexy.
- **Operační metadata** udržují tzv. data lineage (historii odkud se data vzala a jaké transformace na ně byly použity) a stav datové transformace. V rámci stavu datové transformace uchováváme například následující záznamy[9].
  - Návratový kód jobů (doběhl, skončil s chybou/upozorněním).
  - Časové razítko začátku a konce jobu.
  - Počítač, na kterém job běžel.
  - Jakékoli další poznámky o běžícím jobu.

2.5 Datová kvalita

Datová kvalita nám říká, v jakém stavu jsou data z hlediska úplnosti, validity, konzistence, integrity nebo přesnosti[10]. Datová kvalita je proces, který se dá rozdělit do tří kroků: kontrola, analýza a návrhy řešení.
Kontrola zahrnuje:

- kontrolu správnosti hodnot dodaných ze zdrojových systémů,
- kontrolu správnosti/existence vazeb mezi daty,

- kontrolu správnosti výpočtu hodnot v přístupové vrstvě.

Analýza zahrnuje:

- analýzu příčiny/zdroje nesprávných hodnot dodaných ze zdrojových systémů,
- analýzu příčiny/zdroje nesprávných vazeb mezi daty,
- analýzu příčiny/zdroje nesprávných hodnot v přístupové vrstvě.

Návrhy řešení zahrnují:

- navrhnutí procesů pro minimalizaci nebo odstranění vzniku nesprávných hodnot dodaných ze zdrojových systémů,
- navrhnutí procesů opravy a prevenci vzniku nesprávných vazeb mezi daty,
- navrhnutí procesů/oprav algoritmů pro výpočet hodnot ukládaných v přístupové vrstvě.

Datová kvalita tedy je o hledání chyb, návrhů oprav a následné kontrole. Datová kvalita není o opravě dat, vstupní hodnoty ze zdrojových systémů do BI se v rámci datové transformace měnit nesmí.

Závěrem lze dodat, že datový sklad není zodpovědný za špatné hodnoty dat ze zdrojového systému, ale je zodpovědný za monitoring a analýzu problémů kvality dat.

2.6 Historizace a pomalu se měnící dimenze

Jak již bylo naznačeno na předcházejících řádcích, v prostředí BI je žádoucí udržovat historický kontext záznamů. V datových skladech se k tomuto účelu používá technika známá pod jménem „pomalu se měnící dimenze“ (slowly changing dimensions, SCD), která umožňuje přiřadit konkrétní dimenzionální hodnotě vlastní datum a čas platnosti. Technik pro implementaci SCD je více, v práci ke popsán dále používán Kimballův[4] přístup. Kimball rozeznává šest různých typů SCD označených „SCD typ 1“ až „SCD typ 6“. Všechny typy SCD pracují na principu přidání technických sloupců s časovým razítkem, které, na základě typu SCD, sledují logiku změn. V datovém skladu máme celkem 3 operace, které musí SCD podporovat.

- **Přidání** – záznam je nový, potřebujeme ho přidat do tabulky.
- **Změna** – záznam v tabulce existuje, ale hodnota se změnila.
- **Smazání** – záznam v tabulce existuje, ale je potřeba ho smazat.

V práci jsou popsány první 4 typy SCD (ostatní nejsou z hlediska práce relevantní). Pro lepší pochopení je uveden i příklad změny etapy studia z bakalářského na magisterské u studenta s přihlašovacím jménem „stadlmic“.

2.6.1 SCD typ 0: Zachování originálu

Jde o nejjednodušší typ SCD, žádná speciální akce se nekoná. Dimenzionální data zůstávají stejná jako při prvním vložení, dimenzionální atribut se nikdy nemění.

2.6.2 SCD typ 1: Přepsání staré hodnoty

Tato metoda neuchovává žádnou historii dimenzionálních změn. Stará dimenzionální hodnota se jednoduše přepíše hodnotou novou. Tento typ SCD je jednoduchý na údržbu a často se používá pro data, kde změny plynou z procesních úprav (například odebrání speciálních znaků).

2.6.2.1 Postupy pro statusy SCD typ 1

- **Přidání** – záznam se zapíše do tabulky s identifikátorem, který je další v řadě.
- **Změna** – záznam se změní.
- **Smazání** – záznam se ponechá beze změny.

2.6.2.2 Příklad změny záznamu SCD typ 1

Tabulky ukazující záznam studenta a jeho typu studia před 2.1 a po změně 2.2.

ID	username	phase
1	stadlmic	bc

Tabulka 2.1: Příklad SCD typ 1: Tabulka se záznamem o typu studia před změnou

ID	username	phase
1	stadlmic	mgr

Tabulka 2.2: Příklad SCD typ 1: Tabulka se záznamem o typu studia po změně

2.6.3 SCD typ 2: Vytvoření nového záznamu

Tato metodologie zachovává celou historii dimenze. Idea je taková, že starý záznam se označí jako neaktuální a přidá se nový záznam se stejným identifikátorem, označený jako platný a obsahující požadované změny. Takového chování se dosáhne zahrnutím následujících sloupců do struktury dimenzionální tabulky:

- ID – unikátní identifikátor (součást PK)
- EFC_TD – datum, od kdy je záznam platný (součást PK).
- END_DT – datum, do kdy je záznam platný (součást PK). Tady je potřeba definovat datum pro aktuálně platný záznam, běžně se používá 31.12.2199.
- IS_ACT – vlajka aktivního záznamu (1 – aktivní, 0 – neaktivní). Pro každý PK může být aktivní maximálně jeden záznam.
- IS_DEL – vlajka smazaného záznamu (1 – smazaný, 0 – nesmazaný).

2.6.3.1 Postupy pro statusy SCD typ 2

- **Přidání:**
  - ID – přidá se unikátní identifikátor, který je další v řadě.
  - EFC_TD – přiřadí se aktuální datum.
  - END_DT – přiřadí se 31.12.2199 (implicitní datum).
  - IS_ACT – vlajka aktivního záznamu se nastaví na 1 – aktivní.
  - IS_DEL – vlajka smazaného záznamu se nastaví na 0 – nesmazaný.
- **Změna:**
  - END_DT – přiřadí se aktuální datum.
  - IS_ACT – vlajka aktivního záznamu se nastaví na 0 – neaktivní.

  Dále se přidá nový záznam s atributy:
  - ID – přiřadí se stejný identifikátor, jako u měnícího se záznamu.
  - EFC_TD – přiřadí se aktuální datum.
  - END_DT – přiřadí se 31.12.2199 (implicitní datum).
  - IS_ACT – vlajka aktivního záznamu se nastaví na 1 – aktivní.
  - IS_DEL – vlajka smazaného záznamu se nastaví na 0 – nesmazaný.
- **Smazání:**

- END_DT – přiřadí se aktuální datum.
- IS_ACT – vlajka aktivního záznamu se nastaví na 0 – neaktivní.
- IS_DEL – vlajka smazaného záznamu se nastaví na 1 – smazaný.

2.6.3.2 Příklad změny záznamu SCD typ 2

Tabulky ukazující záznam studenta a jeho typu studia před 2.3 a po změně 2.4.

ID	username	phase	EFC_DT	END_DT	IS_ACT	IS_DEL
1	stadlmic	bc	07-22-2010	31-12-2199	1	0

Tabulka 2.3: Příklad SCD typ 2: Tabulka se záznamem o typu studia před změnou

ID	username	phase	EFC_DT	END_DT	IS_ACT	IS_DEL
1	stadlmic	bc	07-01-2011	06-30-2014	0	0
2	stadlmic	mgr	07-01-2014	31-12-2199	1	0

Tabulka 2.4: Příklad SCD typ 2: Tabulka se záznamem o typu studia po změně

SCD typ 2 je obecně docela nákladný na databázové operace, takže ho není doporučeno používat pro dimenze, kde by se v budoucnu mohl objevit nový atribut[11].

2.6.4 SCD typ 3: Přidání nového sloupce

Tato technika funguje na principu verzování dimenzí časovým razítkem. Většinou se uchovává jenom aktuální a předchozí hodnota dimenze. Nová hodnota je nahrána do nového/současného sloupce a stará hodnota je nahrána do starého/předcházejícího sloupce. Obecně řečeno, historie je limitována počtem sloupců vytvořených pro ukládání historických dat.

Struktura SCD typ 3 kromě vlastních sloupců obsahuje:

- VALUE_OLD – slovo „VALUE“ se nahradí příslušným názvem atributu.
- VALUE_UPD_DT – datum změny.

2.6.4.1 Postup pro statusy SCD typ 3

- **Přidání:**
  - ID – přidá se unikátní identifikátor, který je další v řadě.

  - VALUE – přidá se vkládaná hodnota.
  - VALUE_OLD – přiřadí se N/A (implicitní hodnota).
  - VALUE_UPD_DT – přiřadí se datum přidání hodnoty.

- **Změna:**
  - VALUE_OLD – přiřadí se obsah VALUE.
  - VALUE_UPD_DT – přiřadí se datum nové hodnoty.
  - VALUE – přiřazuje se nová hodnota.

- **Smazání:** záznam zůstává beze změny.

2.6.4.2 Příklad změny záznamu SCD typ 3

Tabulky ukazující záznam studenta a jeho typu studia před 2.5 a po změně 2.6.

ID	username	phase	VALUE_OLD	VALUE_UPD_DT
1	stadlmic	bc	N/A	07-01-2011

Tabulka 2.5: Příklad SCD typ 3 Tabulka se záznamem o typu studia před změnou

ID	username	phase	VALUE_OLD	VALUE_UPD_DT
1	stadlmic	bc	N/A	07-01-2011
2	stadlmic	mgr	bc	6-30-2014

Tabulka 2.6: Příklad SCD typ 3: Tabulka se záznamem o typu studia po změně

Kapitola 3

Použité nástroje

Tato kapitola stručně představuje v práci využité nástroje pro ETL (Pentaho Data Integration) a tvorbu reportů (Saiku). Důvod pro výběr těchto dvou konkrétních nástrojů je zřejmý, oba se aktuálně v Datovém skladu ČVUT používají, takže případná další diskuze o volbě nástrojů je v rámci této práce zbytečná.

3.1 Pentaho Data Integration

Pentaho Data Integration (PDI, případně slangově Kettle) je nástroj softwarové společnosti Pentaho určený k návrhu a exekuci ETL procesů. PDI jako ETL nástroj má své nejběžnější využití v prostředí datových skladů, umí ale i migrovat data mezi aplikacemi a databázemi, exportovat data z databází, čistit data nebo integrovat aplikace[12]. PDI začínal jako software vyvíjený komunitou už v roce 2006. Od té doby si drží pověst kvalitního kousku software s dobrou podporou. Mezi jeho výhody patří rozšiřitelnost, přehlednost a platformní nezávislost, protože je napsaný v Javě[13].

Pentaho je softwarová společnost pohybující se v prostředí business intelligence, která nabízí open source produkty[6] zaměřené na datovou integraci, OLAP služby, reporting, informační dashboardy, datamining a ETL[14]. Společnost Pentaho byla založena v roce 2004 pěti nadšenci a má své sídlo v Orlandu na Floridě[15].

K vytvoření ETL v rámci této práce bylo použito PDI verze 7.0.

3.1.1 Základní práce s PDI

Prostředí Kettle se dělí na dvě části: design a canvas[7].

[6] produkty s otevřeným zdrojovým kódem
[7] plátno, pracovní plocha

Část design obsahuje všechny možné komponenty/kroky, které je možné využít. Každá komponenta má unikátní funkcionalitu, rozdělenou do několika kategorií (následující výčet obsahuje pouze vybrané kategorie).

- **Input:** kategorie obsahuje kroky určené k načtení dat. Výběr možností je široký, od „CSV file input“ (načítání dat ze souboru s daty oddělenými čárkou), přes „Generate random value“ (vygeneruje náhodné hodnoty), až po „Table input“ (připojí se k databázi a provede SQL příkaz).

- **Output:** kategorie obsahuje kroky k zapsání dat. Umí zapisovat do textového souboru („Text file output“), databáze („Table output“) a mnohých dalších formátů.

- **Transform:** kategorie pro transformaci dat. Problémem nejsou různé transformace s textovými řetězci, normalizace, mapování hodnot, selekce hodnot nebo přejmenování.

- **Scripting:** kategorie uživateli umožňuje napsat si vlastní skript. Podporovány jsou například jazyky SQL, Javascript, Java. Dále je možné využít regulárních výrazů nebo matematických formulí.

- **Data warehouse:** kategorie kroků používaných speciálně v datových skladech. Kategorie obsahuje i komponentu „Dimesion lookup/update“, která zajistí uložení dat do databáze při zachování principu Slowly Changing Dimension[8].

- **Validation:** komponenty umožňují různé způsoby validace dat.

Druhá část PDI obsahuje pracovní plátno, kam se kroky umisťují. ETL proces typicky vzniká spojením několika komponent, kdy na začátku máme načtení dat, uprostřed datové transformace, skripty, čištění a na konci zápis dat. Takovémuto spojení několika komponent se běžně zkráceně říká „transformace“.

3.2 Pentaho BI Server, Mondrian a Saiku

Pentaho BI Server je platforma umožňující uživatelům přístup k businessovým datům ve formě dashboardů, reportů a OLAP kostek skrz webové rozhraní. Pentaho BI Server má 2 důležité komponenty: Mondrian a Saiku[16].

[8] princip je podrobně popsaný v předcházející kapitole

3.2.1 OLAP a Mondrian

Online Analytical Processing (OLAP) je kategorie softwarové technologie umožňující analytikům a manažerům získat pohled do dat skrz rychlý, konzistentní a interaktivní přístup k široké škále různých pohledů na informace, které byly transformovány z jejich nezpracované podoby. OLAP reflektuje dimenzionální pohled na organizaci tak, jak ho vidí uživatel. OLAP funkcionalita je charakterizovaná dynamickou multi-dimenzionální analýzou konsolidovaných firemních dat podporující analytické aktivity konečného uživatele.

Pro OLAP jsou charakteristické následující techniky.

- **Slice and dice:** technika kouskování informace na menší části za účelem jejich průzkumu z různých úhlů pohledu tak, aby jim analytik blíže porozuměl.

- **Data drilling nebo drilldown:** doslova zavrtání se do dat za účelem zjištění podrobnější a specifičtější informace.

Mondrian je open source OLAP server podporují jazyk MDX[9]. Mondrian používá jako zdroj dat relační databázi (v případě Datového skladu ČVUT PostgreSQL).

3.2.2 Saiku

Saiku přišlo na svět v roce 2008 díky svým autorům Tomu Barberovi a Paulovi Stoellbergerovi s původním názvem Pentaho Analysis Tool. Jméno Saiku tento analytický software dostal v roce 2010 po kompletním přepsání zdrojového kódu[17].

Dnes Saiku nabízí pro uživatele přívětivé webové prostředí, ve kterém může jednoduše analyzovat data a vytvářet a sdílet reporty. Navázání Saiko na OLAP server umožňuje uživatelům vybrat si míry a dimenze, které potřebují k analýze dat a provádění technik „slice and dice“ a/nebo „data drilliing“.

3.2.2.1 Postup práce se Saiku

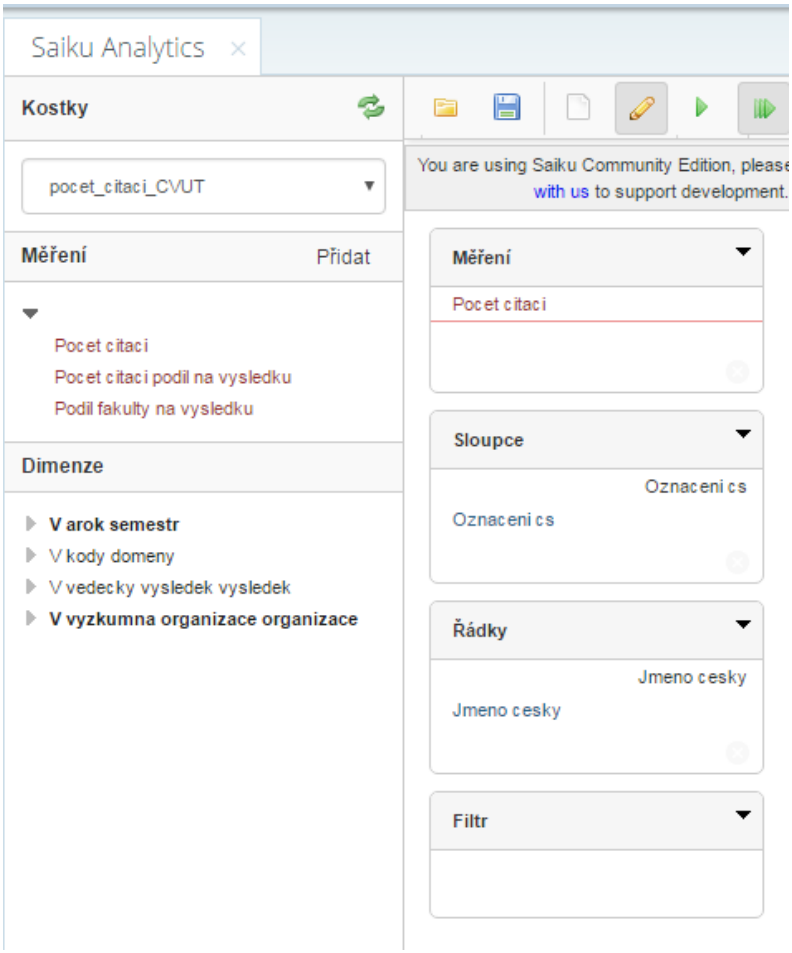
Práce v Saiku je velice intuitivní.

1. **Vytvoření datového zdroje** probíhá přes průvodce „Manage Data Sources“, kdy je nutné nastavit připojení do databáze, nastavit faktovou tabulku i její dimenze a způsob spojení.

2. **Zvolení kostky** je dostupné pod File -> New -> Saiku Analytics -> Create a new query -> [výběr kostky z drop-down menu].

[9] MDX je standartní dotazovací jazyk v prostředí OLAP

3. **Přidání metrik a dimenzí** probíhá přetažením jednotlivých položek z výběru vlevo do příslušné buňky vpravo, čímž se vytvoří report. Jednoduchost vytvoření reportu v Saiku ukazuje obrázek 3.1.



Obrázek 3.1: Ukázka vytvoření jednoduchého reportu v Saiku

Část II

Praktická část

Kapitola 4

Analýza zdrojového systému (V3S + EZOP)

Následující kapitola si klade za cíl čtenáři popsat systém V3S a vysvětlit jeho vztah k systému EZOP. V3S a EZOP jsou sice samostatné aplikace a EZOP stojí mimo rozsah této práce, oba systémy ale sdílí společnou databázi, takže by nebylo rozumné EZOP vůbec nezmínit. V3S je popsán podrobně, velká část kapitoly je věnována popisu a vysvětlení jednotlivých datových struktur. EZOP je v kapitole popsán pouze stručně tak, aby byl zřetelný jeho kontext vzhledem k V3S.

4.1 Základní popis celého systému

Systémy V3S a EZOP vznikaly na ČVUT v rámci projektu „Rozvoj EZOP a V3S“ financovaného v rámci Institucionálního plánu rozvoje ČVUT 2016[18].

EZOP je aplikace Evidence vědeckovýzkumných projektů na ČVUT (od toho zkratka EZOP). Zároveň je podkladem pro komunikaci a konzultace s řešiteli v období, kdy se chystají podat projektový návrh i dále v průběhu projektu.

Aplikace V3S eviduje výsledky vědy a výzkumu (od toho zkratka V3S) a další aktivity vědecko-výzkumných pracovníků ve vědecké komunitě. Aplikace V3S také slouží k odesílání výsledků do RIV[10], exportům pro statistické analýzy i k interním hodnocením vědecko-výzkumné činnosti. Zkratka V3S se vyskytuje ve většině následujících kapitol. Všechna následná použití zkratky V3S v této práci tedy odkazují na tuto aplikaci. V3S a EZOP se dohromady skládají ze čtyř logických částí.

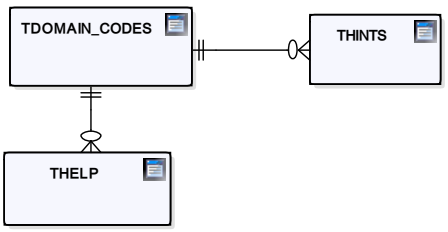
- **VVVS_EZOP:** obsahuje informace o projektech a jejich účastnících. Dále obsahuje informace o rozpočtech a jejich čerpání.

[10]Registr informací o výsledcích MŠMT ČR

- **VVVS_CORE:** obsahuje informace o osobách, organizacích, rolích, notifikacích a uživatelských právech. Ve VVVS_CORE jsou uloženy i číselníky.
- **VVVS_REST:** obsahuje vědecké výsledky, jejich autory, afiliace a návaznosti.
- **VVVS_EXT:** obsahuje data z externích zdrojů, například WOS nebo Scopus[11]. Také se zde shromažďují data pro systém RIV pro hodnocení RVVI[12].

Samotný systém V3S tvoří logické části VVVS_CORE, VVVS_REST a VVVS_EXT. VVVS_EZOP tedy nebude dále uvažován, protože není součástí systému V3S a tedy stojí mimo rámec této práce. Z VVVS_EXT nebudou použity žádné tabulky, protože projekt Datového skladu ČVUT do budoucna počítá s přímým napojením na potřebné externí zdroje. Předmětem integrace tak zůstávají části VVVS_CORE a VVVS_REST. Model VVVS_CORE je dále rozdělen do šesti částí.

- **Core:** obsahuje tabulky vztahující se k nápovědě VVVS, z pohledu datového skladu není tato část zajímavá.

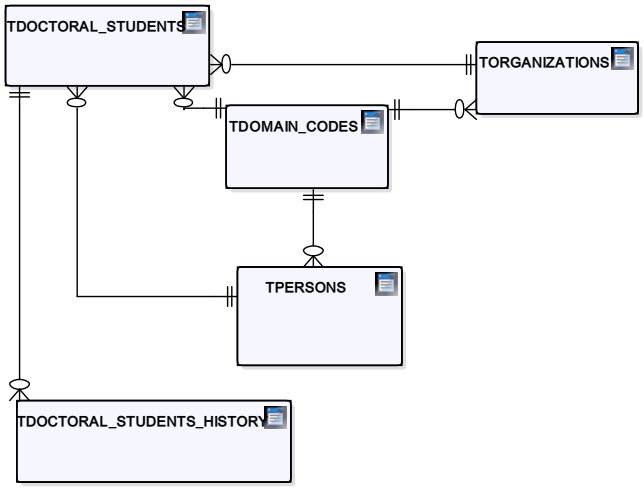


Obrázek 4.1: Schéma V3S týkající se CORE

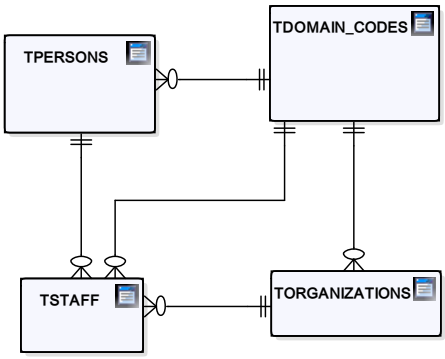
- **Doctoral_Students:** obsahuje informace o doktorském studiu. Fakticky jde o kopii pohledu VV_DOK z Komponenty studium (KOS). Data z KOSu jsou již v Datovém skladu ČVUT integrována přímo, další práce s touto částí tedy není předmětná.
- **Persons:** obsahuje tabulku osob (TPERSONS) z centrálního registru ČVUT, jejich relaci k ČVUT (TSTAFF). Dále tu lze nalézt tabulku s organizacemi (TORGANIZATIONS) a číselník (TDOMAIN_CODES).
- **Rights:** tabulka s uživatelskými oprávněními, z hlediska datového skladu neobsahuje zajímavé informace.

[11] Web of Science a Scopus jsou abstraktové a citační databáze odborné recenzované literatury

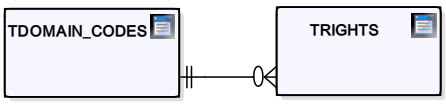
[12] Rada pro výzkum, vývoj a inovace



Obrázek 4.2: Schéma V3S týkající se DOCTORAL_STUDENTS

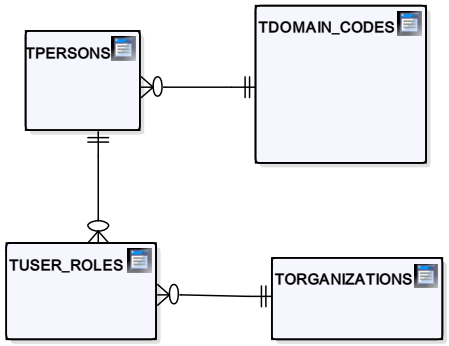


Obrázek 4.3: Schéma V3S týkající se PERSONS

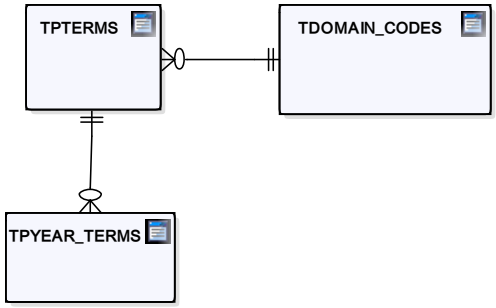


Obrázek 4.4: Schéma V3S týkající se RIGHTS

- **Roles:** obsahuje informace týkající se rolí ve zdrojovém systému, z pohledu datového skladu opět nezajímavá část.
- **Terms:** dle dokumentace obsahuje číselník termínů. Vyplněnost tohoto číselníku je ale mizivá, navíc z hlediska Datového skladu ČVUT neobsahuje žádnou zajímavou vazbu, takže nebude integrován.



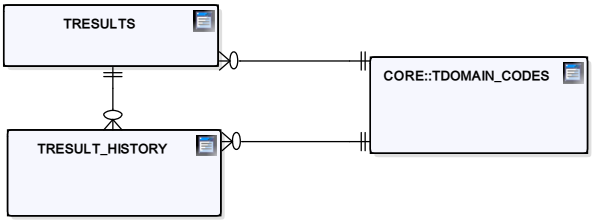
Obrázek 4.5: Schéma V3S týkající se ROLES



Obrázek 4.6: Schéma V3S týkající se TERMS

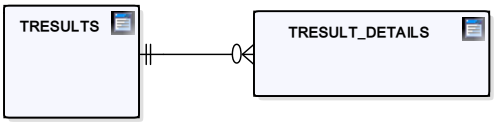
Model VVVS_REST je dále rozdělen na 9 částí.

- **Results:** tabulka vědeckých výsledků (TRESULTS) s číselníkem (TDOMAIN_CODES). Jde o nejdůležitější tabulky ve V3S.



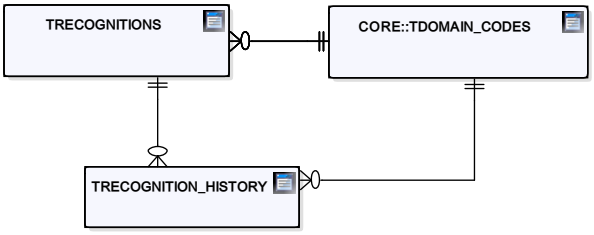
Obrázek 4.7: Schéma V3S týkající se RESULTS

- **Details:** kromě TRESULTS obsahuje i tabulku TRESULT_DETAILS s dalšími detaily vědeckých výsledků.
- **Recognitions:** obsahuje uznání vědeckou komunitou, jako např. recenzování pro odborný časopis, členství v redakčních radách, organizačních/programových výborech konferencí, členství v odborných společ-



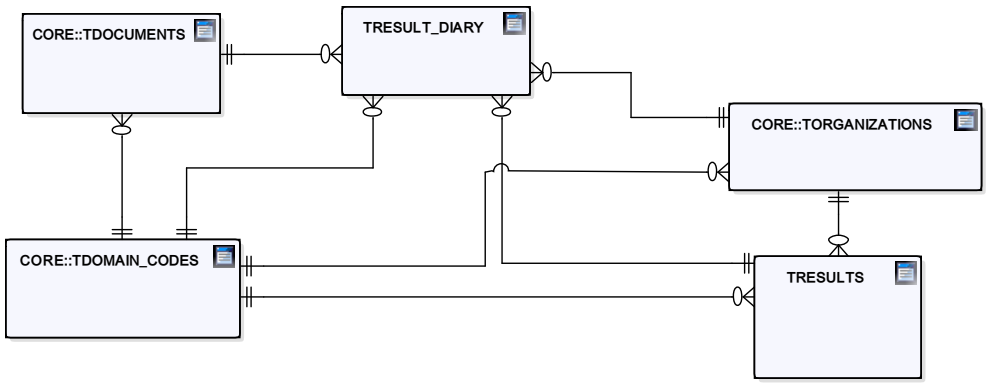
Obrázek 4.8: Schéma V3S týkající se DETAILS

nostech, různá ocenění apod. v tabulkách TRECOGNITIONS a TRECOGNITION_HISTORY.



Obrázek 4.9: Schéma V3S týkající se RECOGNITIONS

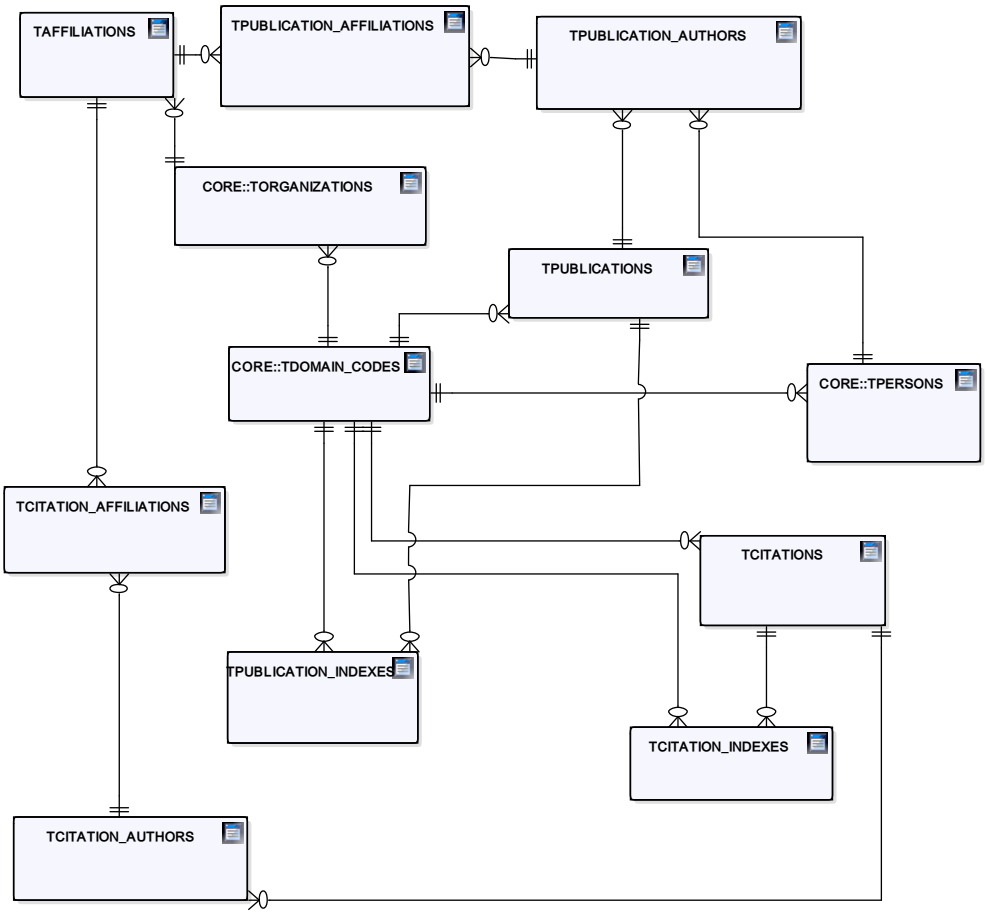
- **Diary:** obsahuje deník výsledku v tabulce TRESULT_DIARY: obecné záznamy vztahující se k výsledku. Jednak obecné poznámky, jednak informace o odeslání do RIV a zjištění přítomnosti záznamu v něm. Z hlediska datového skladu jde o informace teoreticky využitelné k pokročilým analýzám o časové délce cesty výsledku, v současné verzi ale nebude implementováno z důvodu přehlednosti schématu.



Obrázek 4.10: Schéma V3S týkající se DIARY

- **Citations:** v této části jsou kromě citací výsledků CITATIONS a CITATION_AUTHORS i afiliace publikací a citací (TAFFILIATIONS,

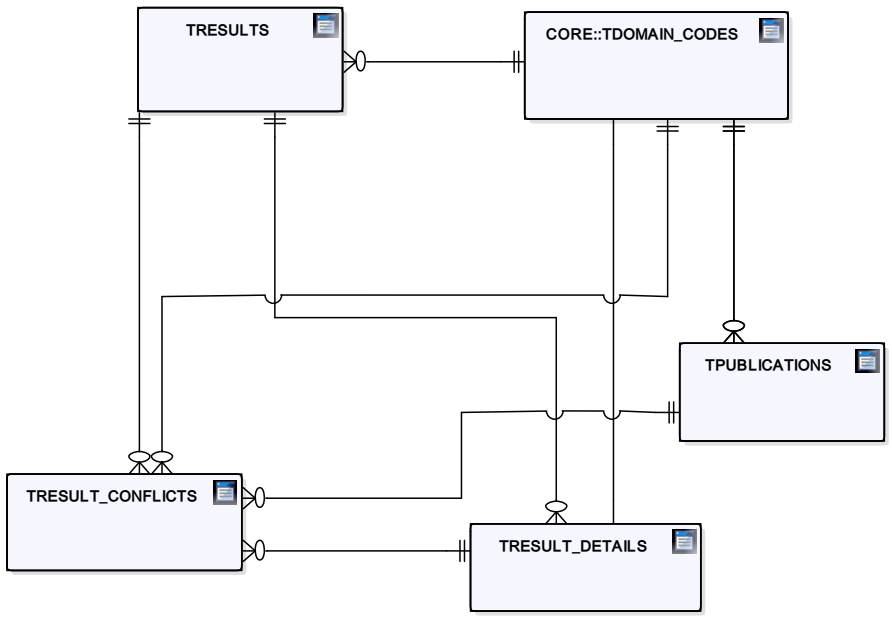
TPUBLICATION_AFFILIATIONS a TCITATION_AFFILIATIONS) i publikace samotné (TPUBLICATIONS). Tabulky s citacemi a publikacemi se stahují z externích systémů, takže by se neměly integrovat v rámci V3S, ale samostatně. V rámci diplomové práce jsou ale zaintegrované citace, aby bylo možné v závěrečné části ukázat větší rozmanitost datových tržišť a reportů.



Obrázek 4.11: Schéma V3S týkající se CITATIONS

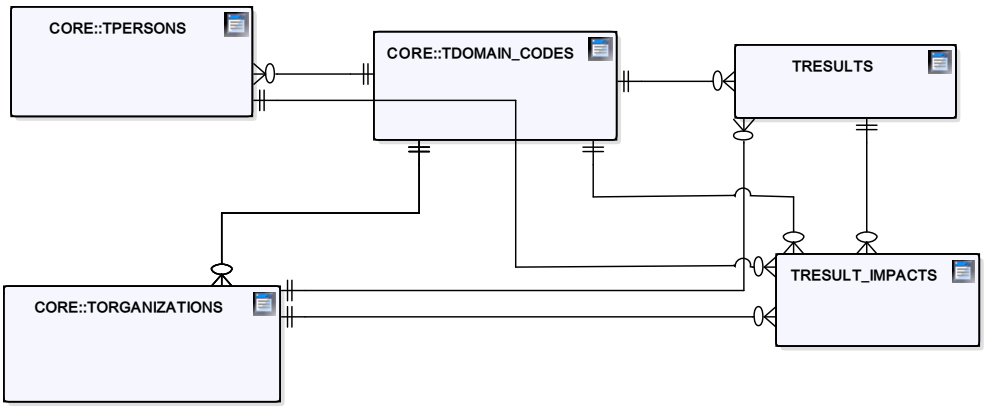
- **Conflicts:** část, ve které se strojově generují návrhy na propojení výsledků s WOS a SCOPUS, které se nepodařilo navázat přímo přes z tabulky TRESULTS. V současné verzi nebude implementováno do datového skladu kvůli vazbě na externí zdroj dat.

- **Impacts:** hlavní tabulkou této části je tabulka TRESULT_IMPACTS, která obsahuje informace o impaktech na výsledky. Tabulka má vazby



Obrázek 4.12: Schéma V3S týkající se CONFLICTS

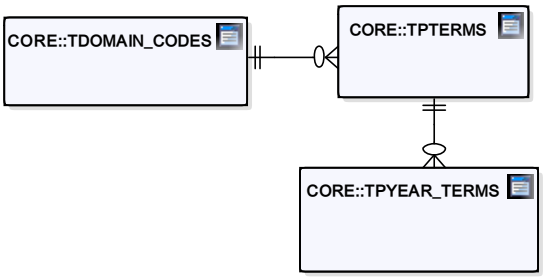
na TRESULTS, TORGANIZATIONS, TPERSONS a číselník TDOMAIN_CODES.



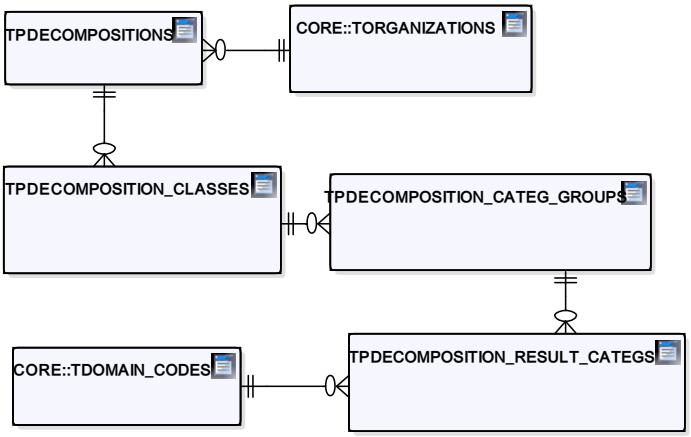
Obrázek 4.13: Schéma V3S týkající se IMPACTS

- **RIV:** obsahuje číselníky, do datového skladu se načítá pouze TDOMAIN_CODES.
- **Decompositions:** obsahuje dekompozice akademických výsledků, v současné verzi nebude implementováno do datového skladu z důvodu

nedostatečné zajímavosti.



Obrázek 4.14: Schéma V3S týkající se RIV



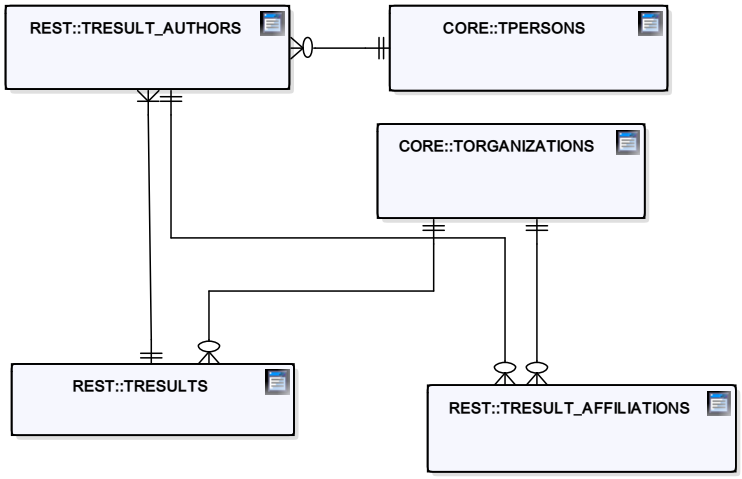
Obrázek 4.15: Schéma V3S týkající se DECOMPOSITIONS

Dále model VVVS_REST ukrývá tabulky TRESULT_AUTHORS a TRESULT_AFFILIATIONS, které sice v žádné části dokumentace uvedené nejsou, z pohledu Datového skladu ČVUT ale zajímavá jsou: obsahují totiž afiliace mezi výsledkem, autorem a organizací, které jsou klíčové k analýze vztahů mezi těmito entitami.

Dále zdrojový systém obsahuje různé pohledy pro zpřehlednění dotazů a řízení uživatelských oprávnění, které se ale do datového skladu nepromítnou: do datového skladu se stahují přímo samotné tabulky a pohledy definované zdrojovým systém tak nejsou potřeba.

4.2 Podrobnější popis vybraných částí V3S

Následující řádky obsahují podrobnější popis vybraných, typicky pro datový sklad zajímavějších, tabulek a sloupců. V žádném případě se se tedy nejedná o kompletní dokumentaci V3S. Celkový model V3S je dostupný v příloze práce.



Obrázek 4.16: Schéma V3S týkající se tabulek TRESULT_AUTHORS a TRESULT_AFFILIATIONS

4.2.1 Tabulka TRESULTS

Tabulka TRESULTS obsahuje informace o vědeckých výsledcích, což z ní dělá nejdůležitější tabulku V3S. Výčet jejích nejdůležitějších sloupců je následující.

- **Result:** unikátní identifikátor výsledku.
- **Superior_result:** unikátní identifikátor nadřazeného výsledku.
- **Organization:** identifikace univerzity, kde práce vznikla. Vyplněnost této hodnoty je ale velmi nízká, takže se musí zjistit vazbou přes příslušnou tabulku afiliací.
- **Result_category_code:** kód kategorie výsledku, viz číselník.
- **Language_code:** kód jazyka, viz číselník.
- **Name_abbr:** krátký název práce.
- **Name_orig, name_en, name_cs:** originální jméno výsledku. Name_en a name_cs obsahují originální jméno, případně jeho jazykovou mutaci.
- **Abstract_orig:** originální abstrakt výsledku. V dalších sloupcích obsahuje i případnou jazykovou mutaci, obdobně jako jméno.
- **Year:** rok vzniku výsledku.
- **Number_of_figures:** počet obrázků v práci (pokud dává tato metrika smysl).

- **Number_of_attachments:** počet příloh práce (pokud dává tato metrika smysl).
- **Event_location:** místo události (pokud dává tato metrika smysl, typicky u konference).
- **Start_date, end_date:** začátek a konec události (pokud dává tato metrika smysl, typicky u konference).
- **Number_of_participants, number_of_foreign_participants:** počet účastníků, počet zahraničních účastníků (pokud dává tato metrika smysl, typicky u konference).
- **Scopus_code, wos_code:** identifikační kódy v systémech Scopus a WoS.
- **Authors:** seznam prvních tří autorů v textovém řetězci. Všichni autoři se zjistí přes příslušnou tabulku afiliací.
- **Keywords:** klíčová slova vědeckého výsledku.

4.2.2 Tabulka TDOMAIN_CODES

Tabulka TDOMAIN_CODES je číselník společný pro celý V3S. Nejdůležitější sloupce jsou následující.

- **Domain:** doména v tabulce pro vyhledávání.
- **Code:** kód záznamu v rámci domény.
- **Value_cs, value_en:** textová hodnota v českém a anglickém jazyce.

Z hlediska datového skladu jsou nejzajímavější kódy reprezentující typ vědeckého výsledku. Tabulka 4.1 obsahuje tři desítky nejpoužívanějších typů vědeckých výsledků na FIT spolu s jejich textovou vysvětlivkou v češtině.

4.2.3 Tabulka TPERSONS

Tabulka TPERSONS obsahuje informace o osobách. Na zbytek Datového skladu ČVUT se osoby budou napojovat přes uživatelské jméno, které je v případě obou systémů jedinečným a společným identifikátorem osoby. Tabulka obsahuje následující zásadní sloupce.

- **Person:** unikátní identifikátor osoby ve V3S.
- **User_name:** unikátní uživatelské jméno osoby.
- **Birth_date:** datum narození.

kód	**vysvětlení kódů v češtině**
STAZMJ	stať ve sborníku z mezinár. konf.
STAZFJ	stať ve sborníku z prestižní konf.
STATMJ	stať ve sborníku z mezinár. konf. cizojazyčně
ASW	software splňující podmínky RIV
ASW	autorizovaný software
CLAZEJ	článek v periodiku excerpovaném SCI Expanded
FVZ	funkční vzorek
CLAZRJ	článek v odborném recenzovaném periodiku
STAZCJ	stať ve sborníku z prestižní konf. (Scopus)
CLAZCJ	článek v periodiku excerpovaném databází Scopus
STATMC	stať ve sborníku z mezinár. konf. česky
VZPT_C	výzkumná zpráva v češtině
ABSTMJ	abstrakt ve sborníku z mezinár. konf. cizojazyčně
STATLC	stať ve sborníku z lokální konf. česky
STAZLJ	stať ve sborníku z lokální konf.
TZPT_J	technická zpráva cizojazyčně
SBOTMJ	sborník z mezinár. konf. cizojazyčně
PRET	nepublikovaná vyzvaná odborná přednáška
CLATJC	článek v periodiku z pozitivního seznamu RVVI česky
KNF	pořádání konference
ABSZMJ	abstrakt ve sborníku z mezinár. konf.
PRO	prototyp
WSH	pořádání workshopu
CLX	článek, který teprve vyjde (je přijatý)
CLX	článek v periodiku, který teprve vyjde
VZPT_J	výzkumná zpráva cizojazyčně
PREZ	nepublikovaná vyzvaná odborná přednáška
KAPZNJ	kapitola v knize s nadnárodní působností
DIPT	diplomová práce

Tabulka 4.1: Tabulka ukazující textovou vysvětlivku kódů, které jsou nejpoužívanější na FIT

- **Sex_code:** kód pohlaví.
- **Name, surname:** jméno, příjmení.
- **Pre_title, post_title:** tituly před a za jménem.
- **Email, phone:** email, telefon.

4.2.4 TORGANIZATIONS

Tabulka TORGANIZATIONS obsahuje informace o organizacích (fakultách, vědeckých pracovištích a podobně). Důležité sloupce tabulky TORGANIZATIONS jsou následující.

- **Organization:** unikátní identifikátor organizace.
- **Superior_organization:** identifikátor nadřízené organizace. Pracovišti je typicky nadřízená katedra nebo fakulta.
- **Name_cs:** jméno v češtině.
- **Abbr_cs:** zkrácené jméno v češtině.

Kapitola 5

Návrh systému a implementace

Následující kapitola hovoří o čtyřech vrstvách Datového skladu ČVUT, konkrétně o stage, integrované vrstvě, sémantické vrstvě a přístupové vrstvě s datovými tržišti. Tato architektura je v souladu se současnou architekturou Datového skladu ČVUT, kterou ve své závěrečné práci podrobně popsal Robert Kotlář[19]. V kapitole záměrně chybí prezentační vrstva, které je věnována samostatná kapitola.

Ačkoli to není ve všech případech explicitně zmíněno, text se vždy týká pouze V3S části Datového skladu ČVUT, nepojednává tedy o Datového skladu ČVUT jako o celku. Čtenář v kapitole nalezne návrh a popis implementace všech čtyř zmíněných vrstev. U integrované a přístupové vrstvy jsou navíc dostupné i podrobně popsané datové modely.

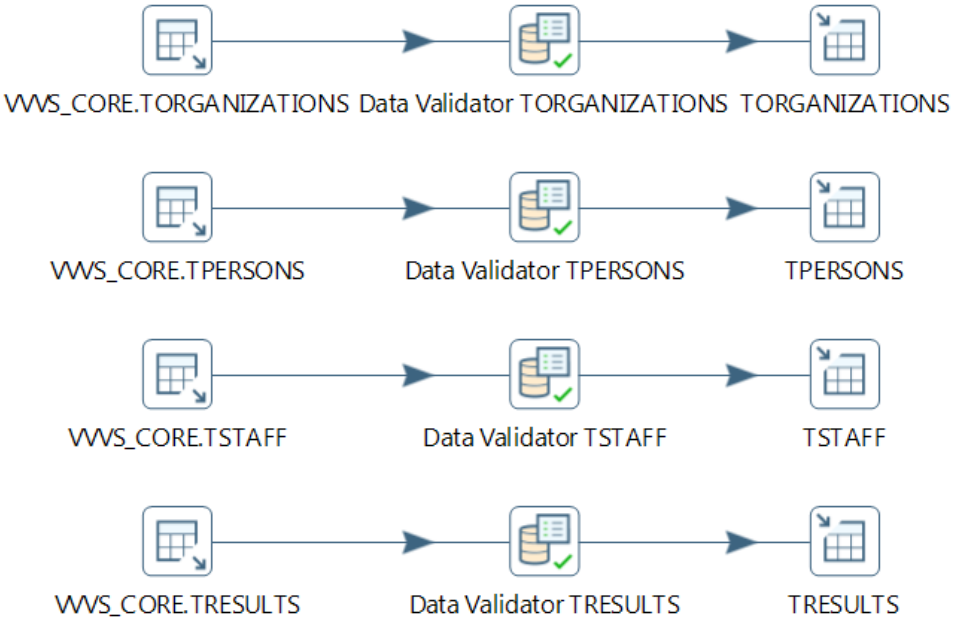
5.1 Stage

Přenos dat ze zdrojové databáze V3S do stage oblasti Datového skladu ČVUT zajišťuje transformace pojmenovaná SS_STG spouštěná v Pentaho Data Integration. Přenos validních záznamů probíhá v režimu 1:1, tedy cokoli se objeví na zdroji, je následně zkopírováno do stage oblasti.

Pro každou stahovanou tabulku je potřeba vytvořit vlastní tok s vlastními komponenty (ukázka na obrázku 5.1), přičemž postup je v tomto případě pro všechny tabulky analogický. Za všechny lze uvést příklad stáhnutí tabulky VVVS_CORE.TRESULTS.

Prvním krokem stáhnutí tabulky VVVS_CORE.TRESULTS je načtení dat ze zdrojového systému komponentou „Table input“. Uvnitř komponenty je nutné nastavit připojení do V3S a specifikovat SQL dotaz do zdrojové databáze následující způsobem:

```
SELECT
translate( RESULT , chr(0),' ') as RESULT,
translate( NAME_ORIG , chr(0),' ') as NAME_ORIG,
```



Obrázek 5.1: Transformace zajišťující přenos dat ze zdrojového systému do stage oblasti datového skladu

```
translate( NAME_EN , chr(0),' ') as NAME_EN,
translate( NAME_CS , chr(0),' ') as NAME_CS,
.......
FROM VVVS_rest.TRESULTS
where VALID=1
```

V SQL dotazu je nutné vyjmenovat všechny sloupce a nahradit znak chr(0) za znak ' ', protože znak chr(0) není podporován PostgreSQL databází, na které běží Datový sklad ČVUT. Dále nás v datovém skladu zajímají pouze validní záznamy, takže se ptáme pouze na řádky s nastavenou vlajkou VALID na 1.

Další krok validuje data. V tomto konkrétním případě pouze kontroluje vyplněnost a datový typ klíčových polí, který pro kontrolu kvality dat v aktuální chvíli stačí.

Posledním krokem je zapsání výsledků do stage oblasti Datového skladu ČVUT. V podrobnostech kroku je nutné specifikovat připojení k cílové databázi, schéma a tabulku. Commit size se osvědčila na hodnotě 1000.

Z výše popsané transformace je patrné, že data se až na minimální úpravu nutnou z důvodu kompatibility databází kopírují 1:1 přesně tak, jak ve své práci navrhuje Inmon[3].

5.2 Integrovaná vrstva

Integrovaná vrstva je centralizovaná a historizovaná databáze všech dlouhodobě uložených dat v datovém skladu. V první části této sekce se čtenář dozví vše podstatné o datovém modelu zaintegrovaného systému V3S, který dává dobrý přehled o návrhu vrstvy. Druhá část sekce je věnována ETL procesům popisujícím realizaci integrované vrstvy.

V grafickém znázornění modelů jsou naznačeny primární klíče, cizí klíče a vztahy mezi tabulkami. Tato notace je použita čistě z důvodu rychlejší orientace v modelu, v Datovém skladu jsou tato integritní omezení vypnutá. Čtenář by tak naznačení primárních klíčů, cizích klíčů i vztahů mezi tabulkami měl brát čistě informativně.

5.2.1 Datový model

Datový model části integrované vrstvy Datového skladu ČVUT týkající se V3S vychází z původní struktury a vztahů tabulek zdrojového systému. Úplně se ale změnilo logické uspořádání, z původních celků VVVS_CORE, VVVS_REST a VVVS_EXT vlastně nezbylo nic. Nové logické uspořádání respektuje ideu rozdělení entit v datovém skladu podle businessového významu a zavádí skupiny vědecký výsledek, vědec, výzkumná organizace a citace. Navíc mnoho tabulek ze zdrojového systému bude čtenář v modelu hledat marně, velké množství jich do integrované vrstvy nebylo připuštěno. Úkolem datového skladu není stáhnou celý zdrojový systém, ale pouze tabulky vhodné k dalšímu analytickému zpracování. Podrobný popis odmítnutých tabulek nalezne čtenář v kapitole o zdrojovém systému.

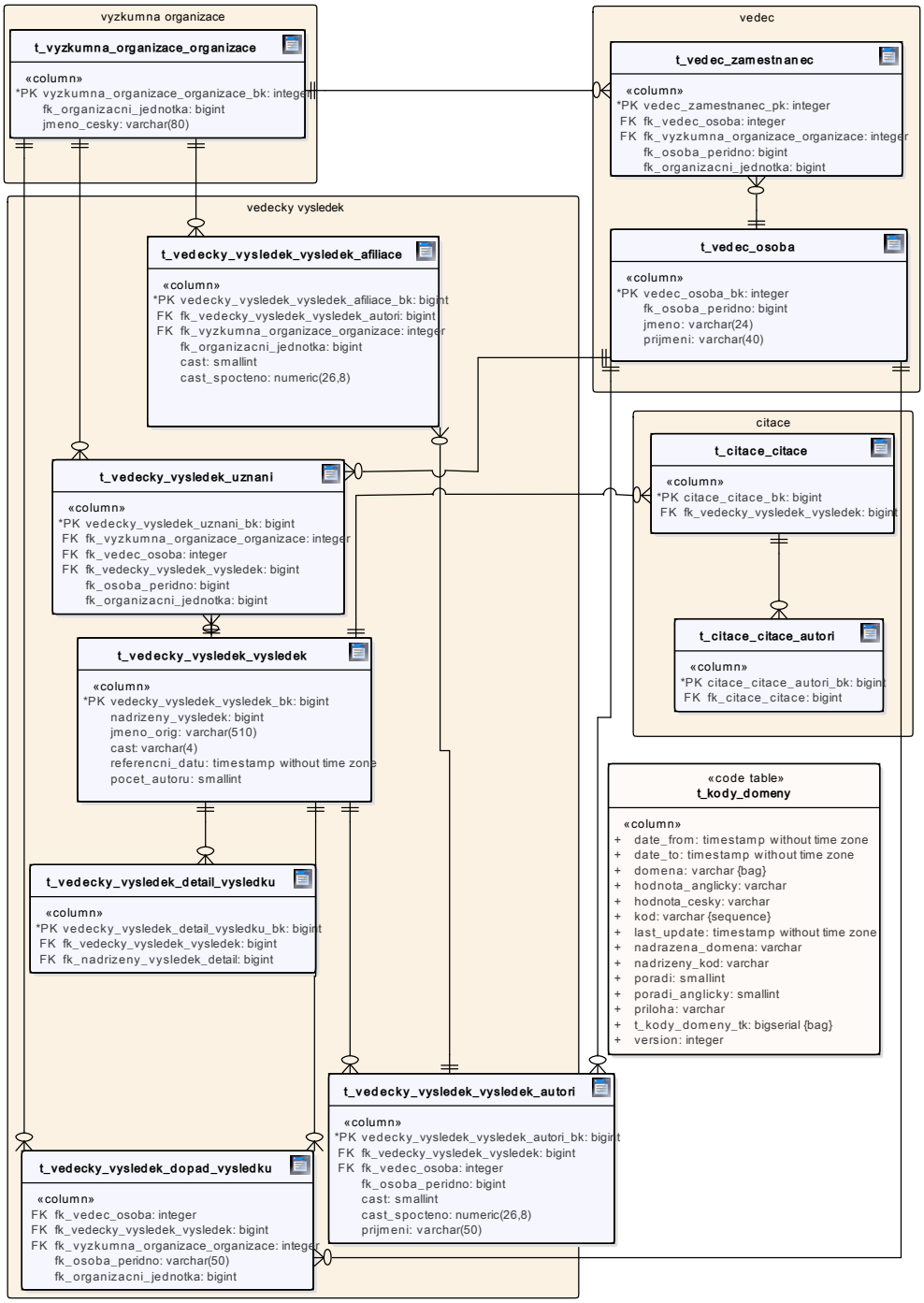
Náhled schématu integrované vrstvy ukazuje obrázek 5.2. Obrázek z důvodu malého rozměru stránky ukazuje pouze velmi zkrácenou verzi tabulek. Plná verze modelu (včetně napojení na schéma Datového skladu ČVUT platné k prvnímu dubnu 2017) je dostupná jako příloha. V modelu jsou naznačeny i cizí klíče odkazující se na tabulky mimo V3S. Konkrétně jde o „fk_osoba_peridno“ (odkaz do T_OSOB_OSOBA z Datového skladu, napojuje se přes uživatelské jméno ČVUT, které je v obou systémech jedinečným a společným identifikátorem osoby) a „fk_organizacni_jednotka“ (odkaz do T_ORGJ_ORGANIZACNI_JEDNOTKA z Datového skladu, napojuje se přes českou zkratku fakulty).

Následující sekce je věnována podrobnějšímu popisu jednotlivých oblastí.

5.2.1.1 Výsledek

Část výsledek obsahuje informace týkající se vědeckých výsledků zaznamenaných ve V3S. Celkem obsahuje 6 tabulek, konkrétně:

- **T_vedecky_ vysledek_vysledek:** nejdůležitější tabulka V3S, obsahuje informace o vědeckých výsledcích.



Obrázek 5.2: Zjednodušený model integrované části Datového skladu ČVUT týkající se V3S

- **T_vedecky_vysledek_detail_vysledku:** obsahuje další detaily o výsledku.
- **T_vedecky_vysledek_uznani:** uznání vědeckou komunitou.
- **T_vedecky_vysledek_vysledek_autori:** vztah mezi výsledkem a autorem.
- **T_vedecky_vysledek_vysledek_afiliace:** afiliace autorů vědeckých prací.
- **T_vedecky_vysledek_dopad_vysledku:** impakt na výsledky.

Obrázek 5.3 pak vyobrazuje model graficky, včetně výčtu sloupců jednotlivých tabulek.

5.2.1.2 Výzkumná organizace

Část o výzkumné organizaci obsahuje jednu tabulku, konkrétně **T_vyzkumna_organizace_organizace**. Tato tabulka obsahuje organizační jednotky (fakulty) včetně podřízených jednotek (katedry a pracoviště).

Obrázek 5.4 pak vyobrazuje model graficky, včetně výčtu sloupců jednotlivých tabulek.

5.2.1.3 Vědec

Část vědec se týká osob zaznamenaných ve V3S. Z pohledu Datového skladu jsou nazváni vědci. Část obsahuje 2 tabulky, konkrétně:

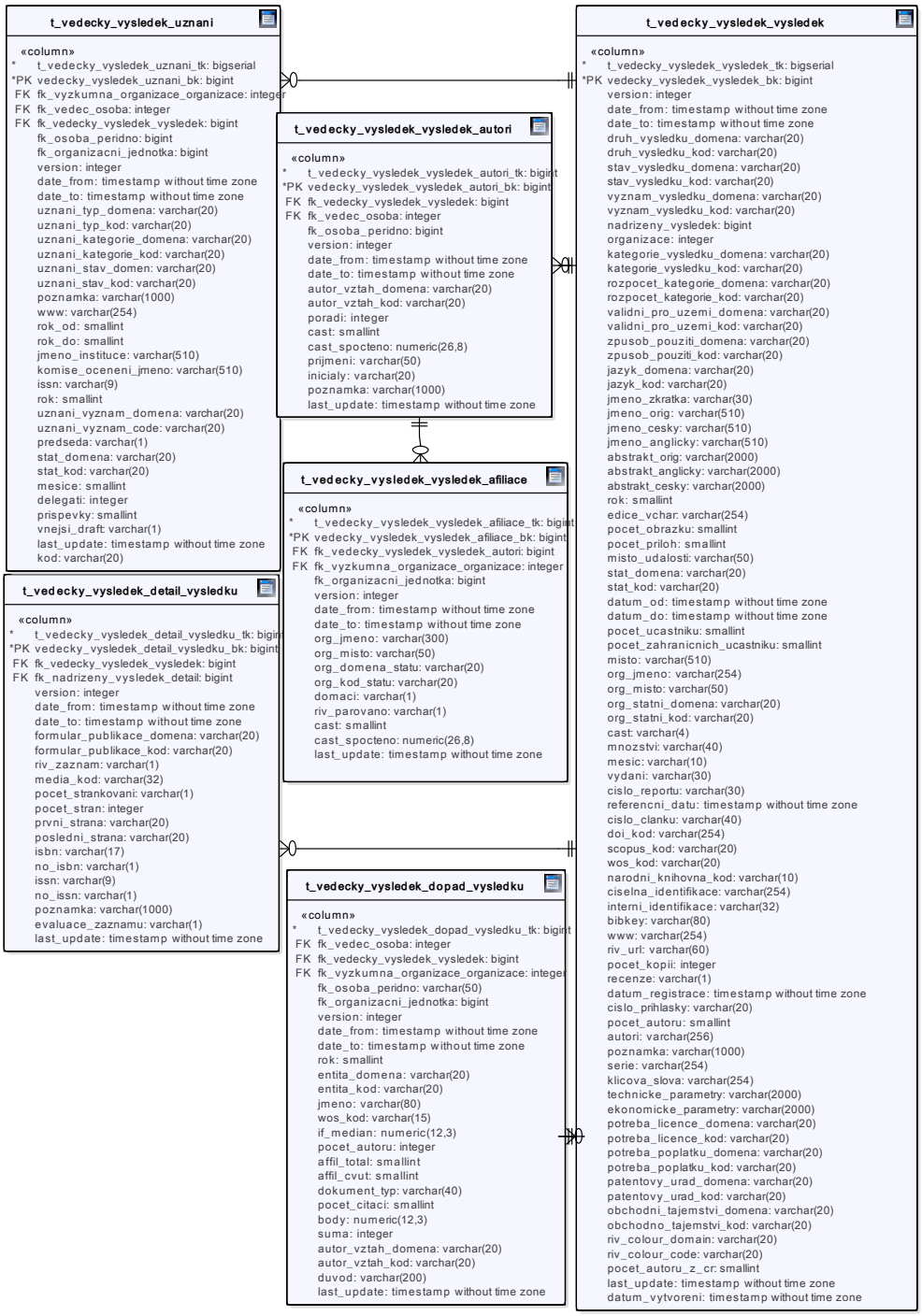
- **T_vedec_osoba:** osoba podle centrálního registru ČVUT.
- **T_vedec_zamestnanec:** relace osob na ČVUT.

Obrázek 5.5 pak vyobrazuje model graficky, včetně výčtu sloupců jednotlivých tabulek.

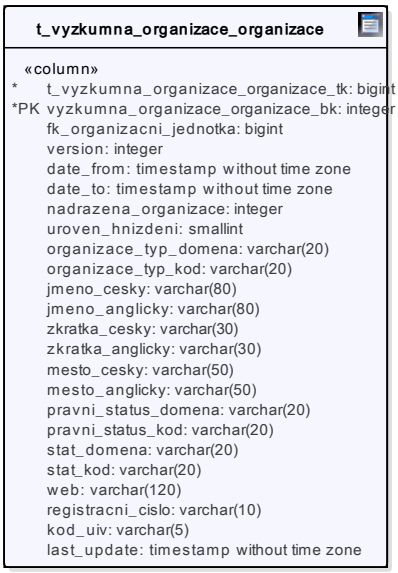
5.2.1.4 Číselník

Číselník **T_kody_domeny** v žádné skupině není, protože obsahuje informace týkající se všech předchozích skupin.

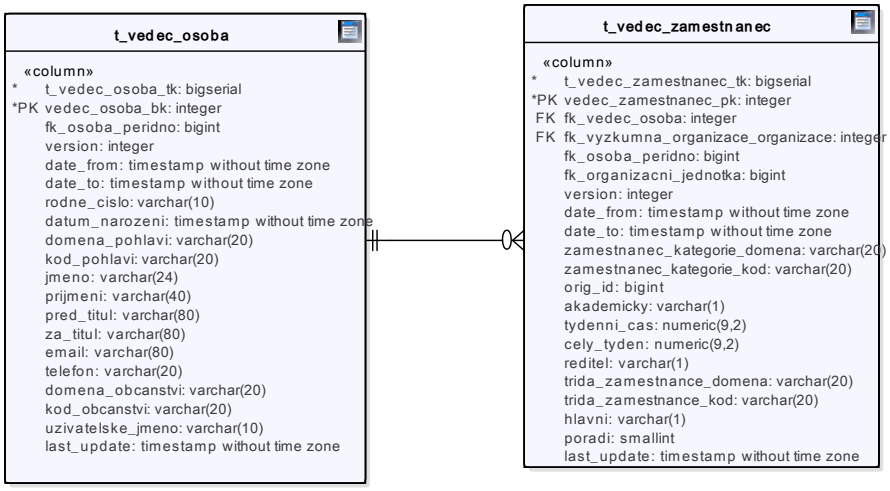
Obrázek 5.6 pak vyobrazuje model číselníku graficky, včetně výčtu sloupců jednotlivých tabulek.



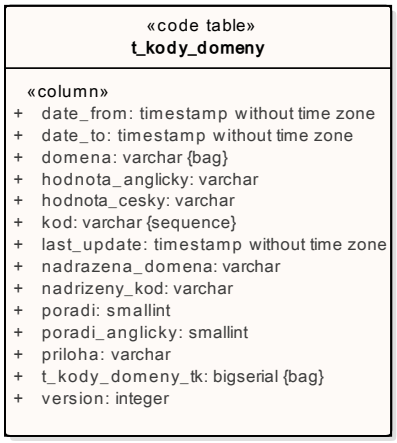
Obrázek 5.3: Model integrované části Datového skladu ČVUT týkající se oblasti vědeckého výsledku



Obrázek 5.4: Model integrované části Datového skladu ČVUT týkající se výzkumné organizace



Obrázek 5.5: Model integrované části Datového skladu ČVUT týkající se osob vědců



Obrázek 5.6: Model číselníku k V3S

5.2.1.5 Citace

Část citace obsahuje 2 tabulky týkající se citací vědeckých výsledků, konkrétně:

- **T_citace_citace:** tabulka citací publikací.
- **T_citace_citace_autori:** tabulka autorů vztažená na citace v tabulce citací.

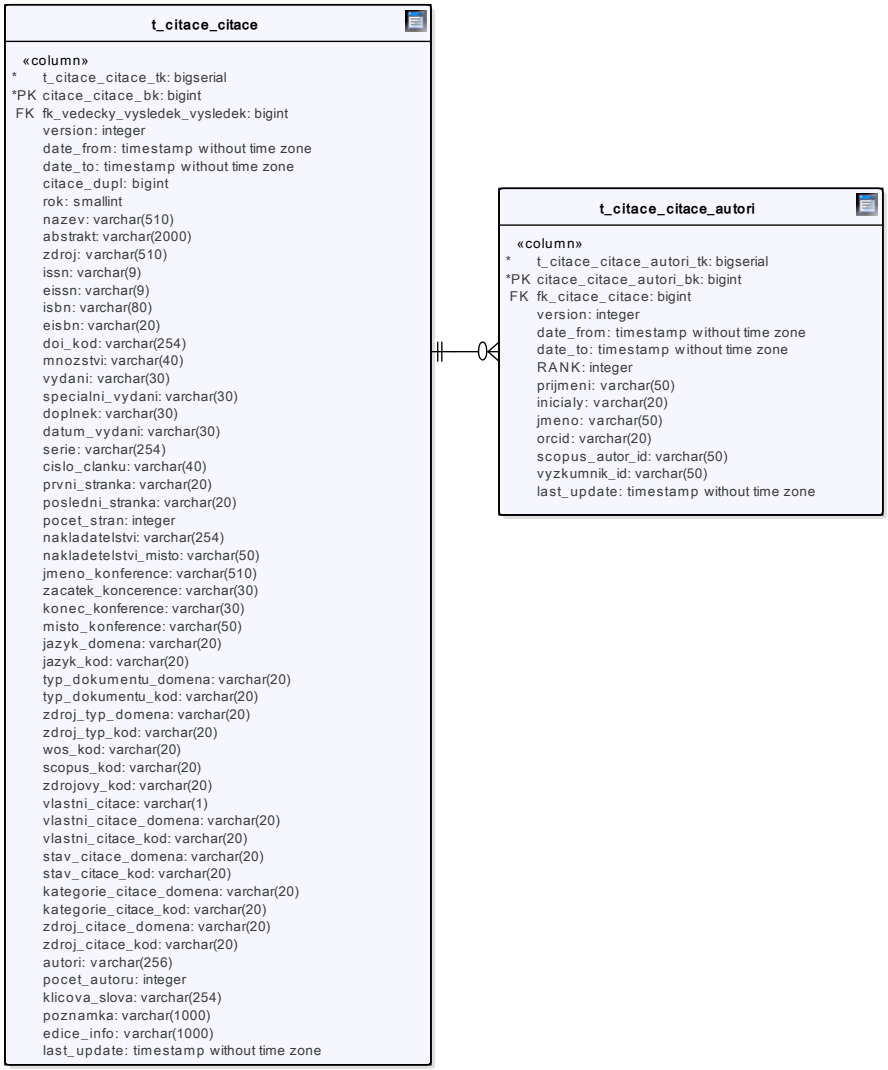
Obrázek 5.7 pak vyobrazuje model graficky, včetně výčtu sloupců jednotlivých tabulek.

5.2.2 ETL

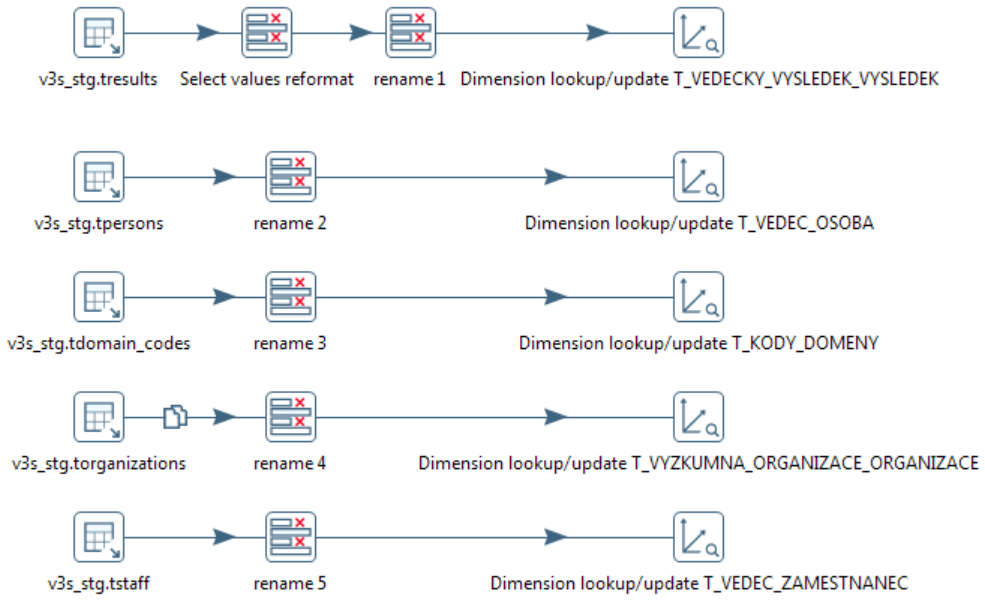
Přenos tabulek ze stage oblasti Datového skladu do integrované oblasti zajišťuje transformace pojmenovaná „STG_CORE“ spouštěná v Pentaho Data Integration.

Pro každou stahovanou tabulku je potřeba vytvořit vlastní tok s vlastními komponenty (ukázka na obrázku 5.8), avšak postup je v tomto případě pro všechny tabulky analogický. Za všechny lze uvést příklad přenosu tabulky TRESULTS.

Prvním krokem série je „Table input“. Uvnitř komponenty je nutné nastavit připojení do PostgreSQL databáze a specifikovat SQL dotaz do stage vrstvy. V SQL dotazu je vhodné funkcí CAST přetypovat sloupce do datového typu, ve kterém budou uložené v integrované vrstvě a u některých tabulek i připojit data z dalších tabulek Datového skladu ČVUT (například „osoba_peridno“, která se napojuje přes uživatelské jméno ČVUT, nebo „organizacni_jednotka“, která se napojuje přes českou zkratku fakulty). Příkladem datového typu předmětného k přetypování je DATE/TIMESTAMP.



Obrázek 5.7: Model integrované části Datového ČVUT týkající se vědeckých citací



Obrázek 5.8: Ukázka transformace ze stage do integrované vrstvy

Select / Rename values

Step name: Select values reformat

Select & Alter | Remove | Meta-data

Fields to alter the meta-data for :

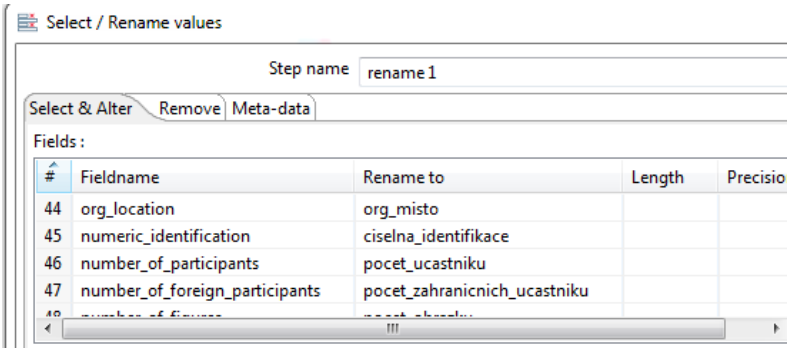
#	Fieldname	Type	L...	I	Format	Date Format
1	end_date	Timestamp	18		dd-MM-yy	N
2	reference_date	Timestamp	18		dd-MM-yy	N
3	registration_date	Timestamp	18		dd-MM-yy	N
4	start_date	Timestamp	18		dd-MM-yy	N
5	authors	String	2..			N
6	numeric_identification	String	2..			N

Obrázek 5.9: Podrobnosti kroku Select values reformat

Druhý krok sekvence specifikuje metadata přenášených dat, které Pentaho BI v některých případech neumí určit samo. Příkladem takového datového typu je datum (ukázka na obrázku 5.9), kde je nutné specifikovat datový typ spolu s jeho formátem. V případě V3S jsou data ve formátu „dd-MM-yy“.

V3S obsahuje veškeré informace v angličtině, Datový sklad ČVUT je celý česky. Názvy sloupců je tedy nutné přejmenovat podle jmenné konvence Datového skladu (ukázka na obrázku 5.10). Tento proces je realizován komponentou „Select/Rename values“ v třetím kroku sekvence.

Posledním a nejzajímavějším krokem je „Dimension lookup/update“.



Obrázek 5.10: Podrobnosti kroku přejmenování

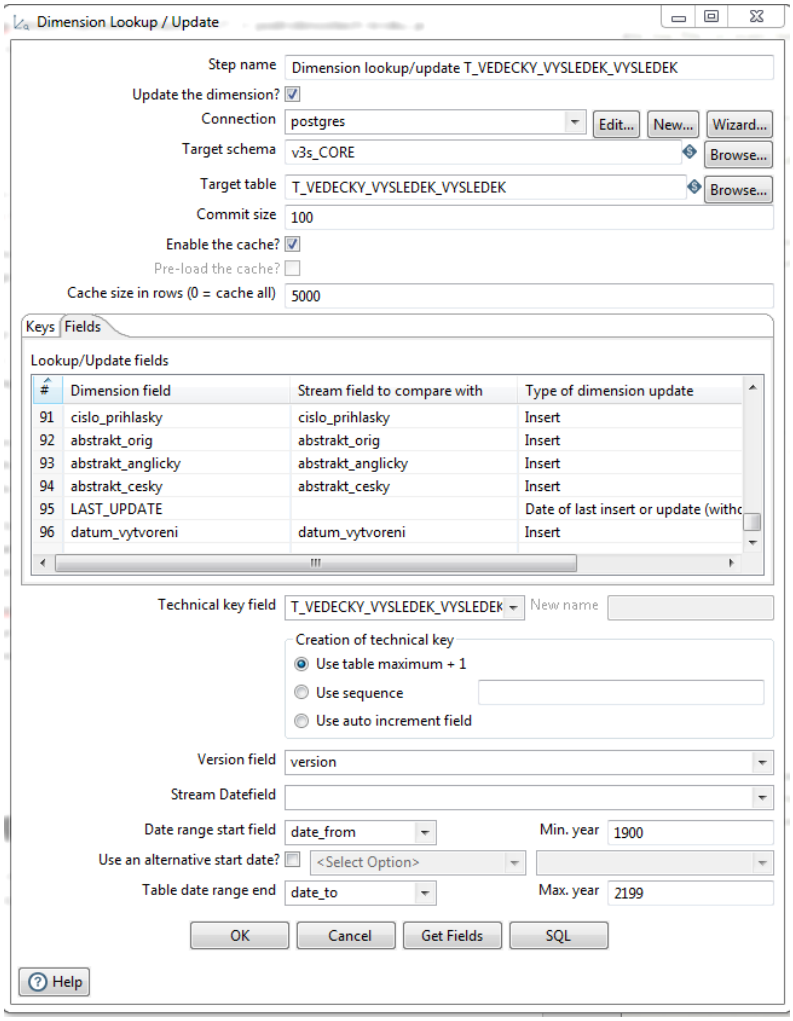
Tento krok realizuje zápis tabulky do integrované vrstvy a to včetně implementace principu pomalu se měnící dimenze typu 2 (SCD typ 2). Díky tomu jsou záznamy historizovány standardizovaným způsobem. V kroku je kromě standardního nastavení připojení k databázi a specifikace cílové tabulky nutné nastavit následující parametry.

- Klíčová pole (key fields), podle kterých se bude řádek vyhledávat v dimenzi. V tomto případě stačí sloupec „vedecky_vysledek_vysledek_bk“, protože je unikátním identifikátorem dimenze platného výsledku.
- Technický klíč, který slouží jako unikátní identifikátor řádku. Jmenná konvence je t_[jmenotabulky]_tk.
- Pole verze, nastaveno na „version“ (podle jmenné konvence).
- Nejdřívější možné datum, nastaveno na „date_from“ a rok 1900.
- Nejpozdější možné datum, nastaveno na „date_to“ a rok 2199.
- V kartě „fields“ se dále nastavuje, která dimenze se porovnává s kterou (typ aktualizace dimenze je „insert“). Pro dodržení jmenné konvence Datového skladu ČVUT je nutné přidat pole „LAST_UPDATE“ (typ aktualizace dimenze je „Date of last insert or update“).

Nastavení podrobně ukazuje obrázek 5.11.

5.3 Sémantická vrstva

Sémantická vrstva vytváří pohledy do integrované datové vrstvy tak, aby vznikla vrstva objektů reprezentující aktuálně platné a businessem definované objekty. Tyto objekty jsou pak následně využívány datamarty bez nutnosti



Obrázek 5.11: Ukázka nastavení SCD typ 2

starání se o další úkony spojené s mapováním často komplexních datových struktur nebo jejich filtrováním (například podle platnosti založené na principu SCD).

Technicky jsou entity sémantické vrstvy realizovány jako databázové pohledy do integrované vrstvy datového skladu. Tyto pohledy filtrují pro další vrstvy nezajímavé sloupce tabulek (například údaje o platnosti, technickém klíči a revizi záznamu) a vybírají z nich aktuálně platné záznamy (podmínky „date_to“ = 2199-12-31 23:59:59.999 a „jmeno_tabulky_tk>“ 0), které reflektují businessové entity. Názvy zdrojových tabulek jsou obdobné, jako názvy cílových pohledů: pro jejich získání stačí vyměnit prefix „V_“ za prefix „T_“.

Pohledy sémantické databáze jsou následující:

- **V_VEDECKY_VYSLEDEK_VYSLEDEK:** kromě již popsaných transformací vybírá datum z (sestupně podle priority): „referencni_datum“, „datum_registrace“, „datum_od“, „rok“, „datum_vytvoreni“ a přiřazuje mu semestr podle rozsahu v „t_arok_semestr“ z Datového skladu ČVUT.
- **V_VYZKUMNA_ORGANIZACE_ORGANIZACE:** žádná další transformace.
- **V_VYZKUMNA_ORGANIZACE_ORGANIZACE_CVUT:** kromě již popsaných transformací vybírá organizace s identifikátorem mezi 11000 a 84000 (organizace na ČVUT).
- **V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR:** žádná další transformace.
- **V_VEDECKY_VYSLEDEK_UZNANI:** žádná další transformace.
- **V_VEDECKY_VYSLEDEK_DOPAD_VYSLEDKU:** žádná další transformace.
- **V_VEDECKY_VYSLEDEK_DETAIL_VYSLEDKU:** žádná další transformace.
- **V_VEDEC_ZAMESTNANEC:** žádná další transformace.
- **V_VEDEC_OSOBA:** žádná další transformace.
- **V_VEDEC_OSOBA_hash:** kromě již popsaných transformací zahashuje jméno osoby funkcí MD5.
- **V_KODY_DOMENY:** žádná další transformace.
- **V_CITACE_CITACE_AUTORI:** žádná další transformace.
- **V_CITACE_CITACE:** žádná další transformace.
- **V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE:** žádná další transformace.
- **V_AROK_SEMESTR:** data ze současné verze Datového skladu ČVUT.
- **V_PRED_BEHPREDMETU_UCITEL_REL:** data ze současné verze Datového skladu ČVUT.
- **V_OSOB_OSOBA:** data ze současné verze Datového skladu ČVUT.

- **V_OSOB_OSOBA_HASH:** data ze současné verze Datového skladu ČVUT.
- **V_OSOB_UCITEL_FIT:** data ze současné verze Datového skladu ČVUT s agregací lektorských metrik.

5.4 Přístupová vrstva (datová tržiště)

Přístupová vrstva je místem, kde vznikají datová tržiště. Sekce konkrétně pojednává o třech datamartech navržených pro analýzu a prezentaci vybraných dat ze systému V3S. První se týká kombinace vědecko-výzkumné a lektorské činnosti učitelů na FIT ČVUT. Další dva poskytují metriky pro hodnocení počtu a kvality vědeckých výsledků na ČVUT.

Datamarty mají i svojí anonymizovanou verzi. Modely anonymizovaných datamartů ale v práci zobrazené z důvodu přehlednosti nejsou (anonymizované datamarty jsou kromě anonymizace totožné, tudíž by jejich popis byl značně redundantní). Technicky je anonymizace provedena zahashováním předmětné hodnoty jednosměrnou funkcí MD5. Čistě z důvodu přehlednosti výsledků jsou pak v práci vždy zobrazeny jenom první 4 znaky hashe. Anonymizace je v tomto případě velice důležitá, protože některá data jak z V3S, tak i z Datového skladu ČVUT nelze z právních důvodů poskytnout veřejně.

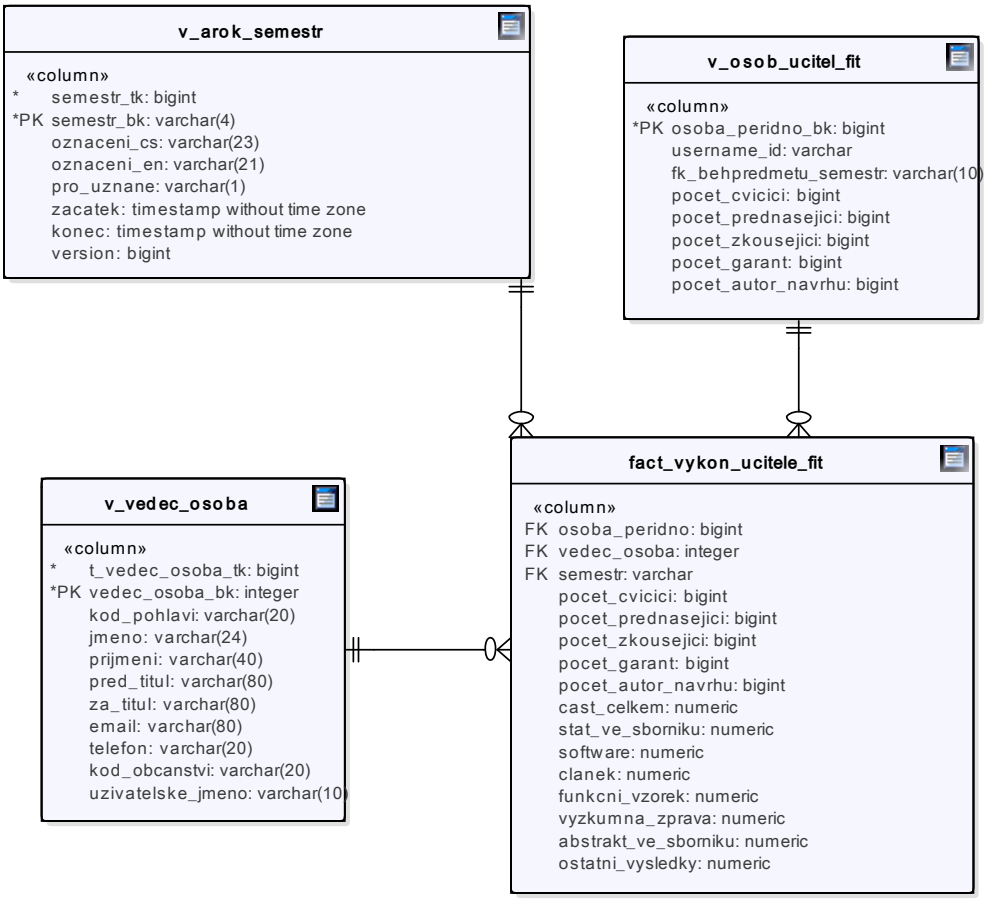
5.4.1 Datamart Výkon učitele FIT

Datamart Výkon učitele FIT si klade za cíl shromažďovat informace kombinující výsledky vědecko-výzkumné činnosti vědců z FIT ČVUT[13] s metrikami týkajících jejich lektorské činnosti.

Datamart má následující dimenze:

- **Osoba_peridno:** unikátní identifikátor osoby v současné verzi Datového skladu ČVUT, tzv. „peridno“ z tabulky V_OSOB_UCITEL_FIT. S Částí věnující se V3S se v integrované vrstvě napojuje přes ČVUT username s tabulkou V_VEDEC_OSOBA, které taktéž jednoznačně identifikuje osobu na ČVUT. Osoby jsou omezené na učitele FIT (učitel FIT je definován jako osoba, která někdy učila na FIT).
- **Vedec_osoba:** unikátní identifikátor osoby v systému V3S (tabulka V_VEDEC_OSOBA).
- **Semestr:** unikátní identifikátor semestru v současné verzi Datového skladu ČVUT. Odpovídající tabulka v Datovém skladu ČVUT obsahuje začátek a konec semestru, takže je možné přiřadit k datu výsledku

[13] Po zpřístupnění dat o lektorské činnosti na ostatních fakultách není problém datamart rozšířit na celé ČVUT.



Obrázek 5.12: Schéma datamartu s výkonem učitele FIT

z V3S unikátní semestr na ČVUT (po odfiltrování semestrů, které mají nesmyslnou platnost, protože v KOS slouží k jinému účelu).

Datamart má následující metriky:

- **Pocet_cvicici:** počet paralelek cvičení, které osoba odvedla. Počítá se agregací z tabulky V_BEHPREDMETU_UCITEL_REL.
- **Pocet_prednasejici:** počet paralelek přednášek, které osoba odvedla. Počítá se agregací z tabulky V_BEHPREDMETU_UCITEL_REL.
- **Pocet_zkousejici:** počet paralelek, které osoba odzkoušela. Počítá se agregací z tabulky V_BEHPREDMETU_UCITEL_REL.
- **Pocet_garant:** počet paralelek, které osoba odgarantovala. Počítá se agregací z tabulky V_BEHPREDMETU_UCITEL_REL.
- **Pocet_autor_navrhu:** počet paralelek, kde je osoba uvedená jako autor předmětu. Počítá se agregací z tabulky V_BEHPREDMETU_UCITEL_REL.
- **Cast_celkem:** součet níže uvedených metrik, tedy stat_ve_sborniku, software, clanek, funkcni_vzorek, vyzkumna_zprava, abstrakt_ve_sborniku a ostatni_vysledky. Součin se počítá z tabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.
- **Stat_ve_sborniku:** suma součinu míry afiliace výsledku k organizaci a k osobě, kde je kód výsledku označen „STA“ (stať ve sborníku). Součin se počítá z tabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.
- **Software:** suma součinu míry afiliace výsledku k organizaci a k osobě, kde je kód výsledku označen „ASW“ (software). Součin se počítá zvtabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.
- **Clanek:** suma součinu míry afiliace výsledku k organizaci a k osobě, kde je kód výsledku označen „CLA“ (článek). Součin se počítá z tabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.
- **Funkcni_vzorek:** suma součinu míry afiliace výsledku k organizaci a k osobě, kde je kód výsledku označen„ FVZ“ (funkční vzorek). Součin se počítá z tabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.

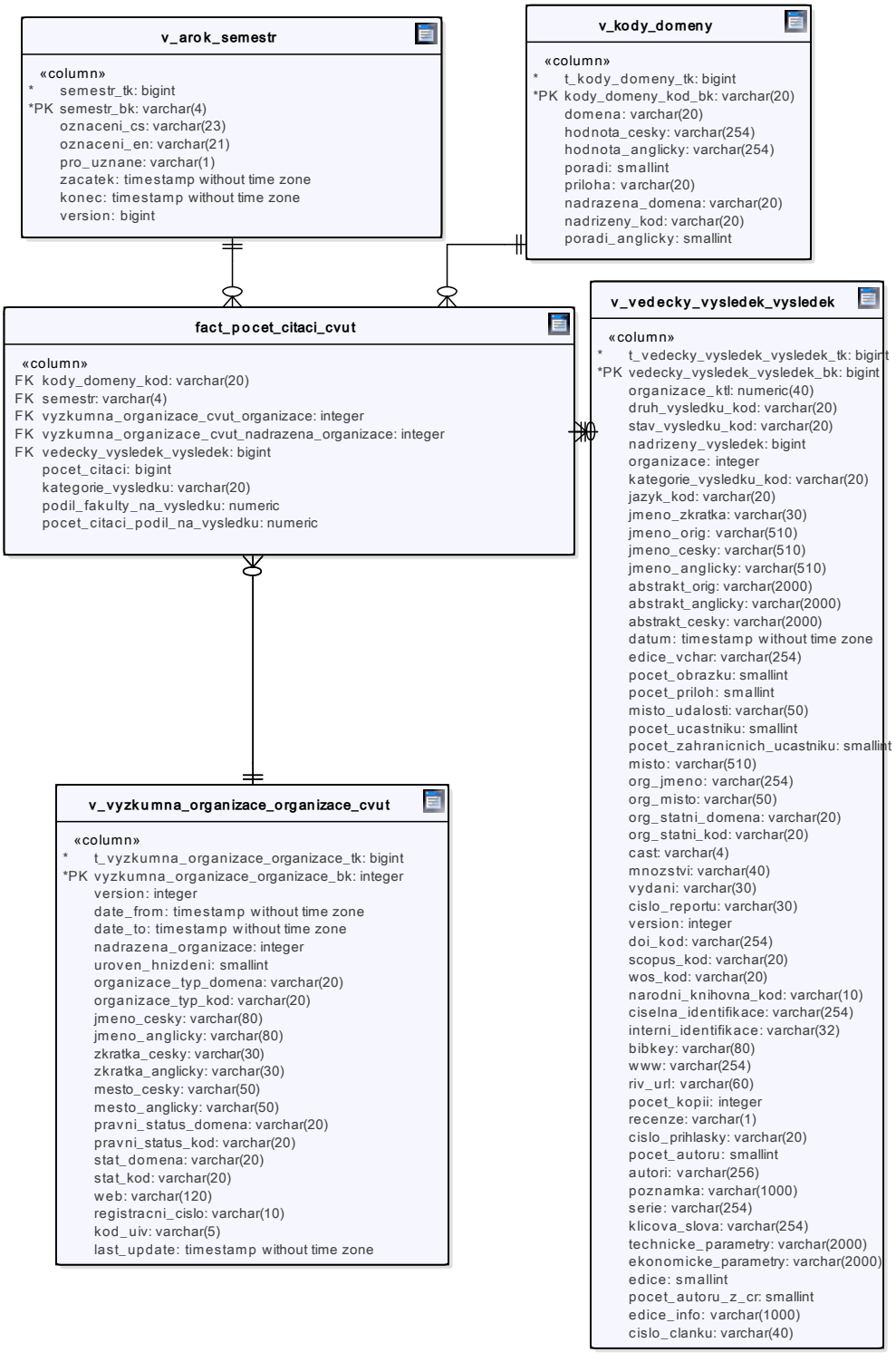
- **Vyzkumna_zprava:** suma součinu míry afiliace výsledku k organizaci a k osobě, kde je kód výsledku označen „VZP“ (výzkumná zpráva). Součin se počítá z tabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.
- **Abstrakt_ve_sborniku:** suma součinu míry afiliace výsledku k organizaci a k osobě, kde je kód výsledku označen „ABS“ (abstrakt ve sborníku). Součin se počítá z tabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.
- **Ostatni_vysledky:** suma součinu míry afiliace výsledku k organizaci a k osobě, kde kód výsledku není označen ani jedním z výše uvedených kódů. Součin se počítá z tabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.

5.4.2 Datamart Počet citací ČVUT

Datamart Počet citací ČVUT obsahuje informace o citacích jednotlivých vědecký prací ze systému V3S.

Datamart má následující Dimenze:

- **Semestr:** unikátní identifikátor semestru v současné verzi Datového skladu ČVUT. Odpovídající tabulka v Datovém skladu ČVUT obsahuje začátek a konec semestru, takže je možné přiřadit k datu výsledku z V3S unikátní semestr na ČVUT (po odfiltrování semestrů, které mají nesmyslnou platnost, protože v KOS slouží k jinému účelu).
- **Kody_domeny_kod:** označuje kategorii typu vědeckého výsledku. Popis kategorií je v číselníku V_KODY_DOMENY.
- **Vyzkumna_organizace_cvut_organizace:** unikátní identifikátor organizace/pracoviště ve V3S. Dostupný v tabulce V_VYZKUMNA_ORGANIZACE_ORGANIZACE.
- **Vyzkumna_organizace_cvut_nadrazena_organizace:** unikátní identifikátor nadřízené organizace (na úrovni fakulty) ve V3S. Dostupný v tabulce V_VYZKUMNA_ORGANIZACE_ORGANIZACE.
- **Vedecky_vysledek_vysledek:** unikátní identifikátor vědeckého výsledku ve V3S. Dostupný v tabulce V_VEDECKY_VYSLEDEK_VYSLEDEK.



Obrázek 5.13: Schéma datamartu s počtem citací vědeckých prací

Datamart má následující metriky:

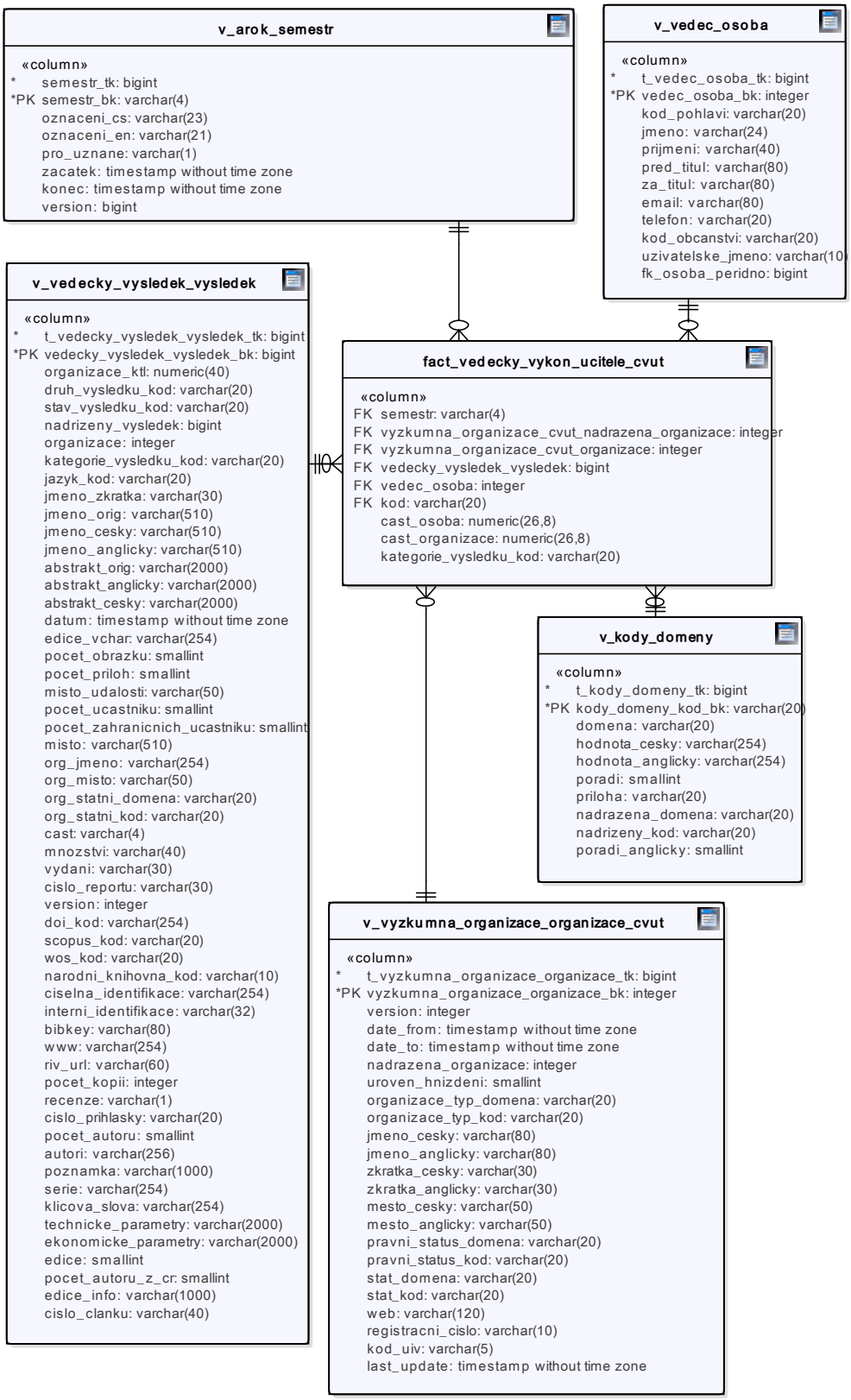
- **Pocet_citaci:** počet citací vědeckého výsledku.
- **Podil_fakulty_na_vysledku:** poměr, jaký měla fakulta na výsledku. Počítá se jako suma součinu míry afiliace výsledku k organizaci a k osobě. Součin se počítá z tabulek V_VEDECKY_VYSLEDEK_VYSLEDEK_AUTOR a V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.
- **Pocet_citaci_podil_na_vysledku:** součin předchozích metrik. Vhodný ke zjištění, jestli na hodně citovaných pracích organizace s někým dalším spolupracuje, nebo jestli bádá svépomocí.

5.4.3 Datamart Vědecký výkon učitele ČVUT

Datamart vědecký výkon učitele ČVUT ukrývá informace o míře afiliace jednotlivých vědeckých výsledků z ČVUT ke konkrétním organizacím a konkrétním osobám na ČVUT.

Datamart má následující dimenze:

- **Semestr:** unikátní identifikátor semestru v současné verzi Datového skladu ČVUT. Odpovídající tabulka v Datovém skladu ČVUT obsahuje začátek a konec semestru, takže je možné přiřadit k datu výsledku z V3S unikátní semestr na ČVUT (po odfiltrování semestrů, které mají nesmyslnou platnost, protože v KOS slouží k jinému účelu).
- **Kody_domeny_kod:** označuje kategorii typu vědeckého výsledku. Popis kategorií je v číselníku V_KODY_DOMENY.
- **Vedec_osoba:** unikátní identifikátor osoby v systému V3S (tabulka V_VEDEC_OSOBA).
- **Vedecky_vysledek_vysledek:** unikátní identifikátor vědeckého výsledku ve V3S. Dostupný v tabulce V_VEDECKY_VYSLEDEK_VYSLEDEK.
- **Vyzkumna_organizace_cvut_organizace:** unikátní identifikátor organizace/pracoviště ve V3S. Dostupný v tabulce V_VYZKUMNA_ORGANIZACE_ORGANIZACE.
- **Vyzkumna_organizace_cvut_nadrazena_organizace:** unikátní identifikátor nadřízené organizace (na úrovni fakulty) ve V3S. Dostupný v tabulce V_VYZKUMNA_ORGANIZACE_ORGANIZACE.



Obrázek 5.14: Schéma datamartu s vědeckým výkonem učitele ČVUT

Datamart má následující metriky:

- **Cast_osoba:** míra afiliace vědeckého výsledku ke konkrétní osobě. Dostupná v tabulce V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.
- **Cast_organizace:** míra afiliace vědeckého výsledku ke konkrétní organizace. Dostupná v tabulce V_VEDECKY_VYSLEDEK_VYSLEDEK_AFILIACE.

Kapitola 6

Prezentační vrstva a reporty

Následující kapitola ukazuje příklady využití technologie OLAP na v předcházející kapitole navržených datových tržištích. Navržené reporty jsou příkladem, jak mohou reporty vypadat a zároveň ověřením/otestováním funkčnosti integrace V3S do Datového skladu ČVUT.

Je nutné zdůraznit, že prezentační vrstva již je prostředím, kde se pohybují business uživatelé, do jejichž kompetence spadá tvorba reportů i prezentace dat, a proto z principu věci výčet zmíněných reportů v žádném případě nevyužívá plný potenciál navržených datových tržišť.

Z důvodu zachování anonymity výsledků jsou některé hodnoty pro účel prezentace v diplomové práci anonymizovány (typicky nahrazeny prvními čtyřmi znaky MD5 hashe ze jména/příjmení). V reálném nasazení je možné zobrazovat jak hash, tak i konkrétní identitu jednotlivých osob v závislosti na tom, jaké má uživatel reportu oprávnění.

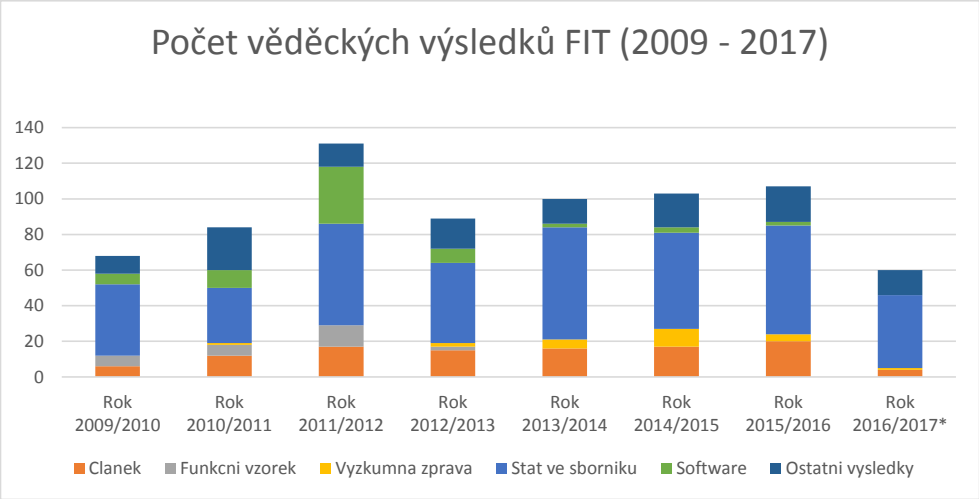
Zobrazené výsledky reprezentují reálná data z Datového skladu ČVUT a systému V3S platná k prvnímu dubnu 2017.

6.1 Reporty datamartu Výkon učitele FIT

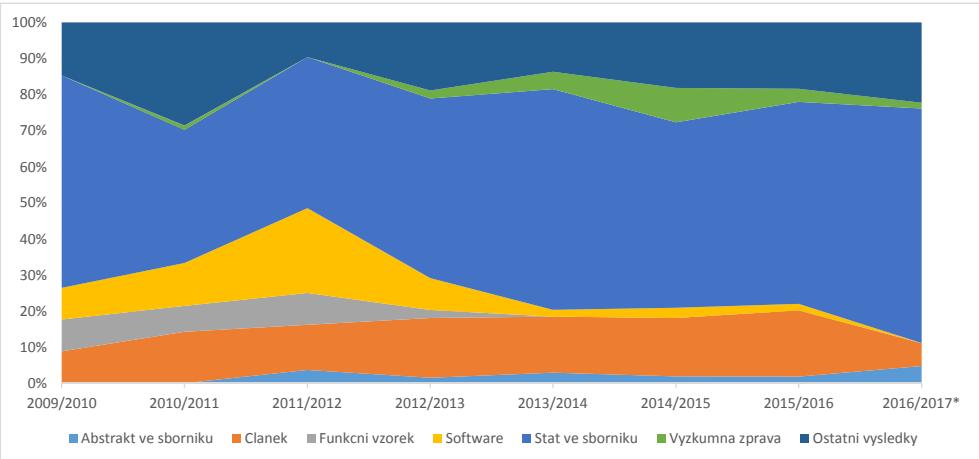
První report plynoucí z datamartu Výkon učitele FIT (učitel na FIT je osoba, která alespoň jednou učila na FIT) zachycuje počet vědeckých výsledků na FIT tak, jak je zaznamenal systém V3S. Jak je vidět na obrázku 6.1, nejúspěšnější z hlediska počtu vědeckých výsledků byl pro FIT akademický rok 2011/2012 a to zejména díky kategoriím Software, a Stať ve sborníku.

Druhý report zachycuje obdobné období, jako předcházející report. Tentokrát je zde komunikován poměr jednotlivých vědeckých výsledků normovaný na 100 %. Z grafu na obrázku 6.2 je vidět, že vědecké výsledky na FIT dlouhodobě táhne především kategorie Stať ve sborníku.

Tabulky 6.1 a 6.2 ukazují, kolik toho konkrétní osoby odučili a vyzkoumali na ČVUT a zároveň jsou učiteli na FIT. Zkratky v hlavičkách tabulek odpovídají metrikám popsaným v sekci o datamartech.



Obrázek 6.1: Počet vědeckých výsledků na FIT v letech 2009 až 2017, *data za akademický rok 2016/2017 nejsou kompletní



Obrázek 6.2: Poměr vědeckých výsledků na FIT za dobu existence FIT, *data za akademický rok 2016/2017 nejsou kompletní

osoba	výuka	věda	#cvi	#před	#zk	#gar	#aut
f95e	1 320	0,00	312	271	273	237	227
4fab	855	3,76	167	95	113	112	368
86f1	749	0,00	187	169	178	110	105
2666	737	11,21	152	124	137	151	173
ceed	704	5,19	144	105	138	157	160
eb07	688	0,00	166	147	153	113	109
98fb	631	0,00	146	128	130	113	114
91d4	590	3,05	258	100	95	69	68
dca7	587	0,40	86	43	51	88	319

Tabulka 6.1: Učitelé z FIT a jejich celková výuka a suma poměrných vědeckých výsledků bez časového ohraničení, setříděno dle celkové výuky

Tabulka 6.1 je setříděná podle celkové výukové aktivity („Výuka“ je sumou počtu odvedených cvičení, přednášek, zkoušek, garantovaných předmětů a autorství předmětu). Konkrétní identitu osoby prozradit nelze, pro zajímavost ale lze uvést, že tato osoba uvedené hodnoty sbírala přes 2 dekády na několika fakultách ČVUT.

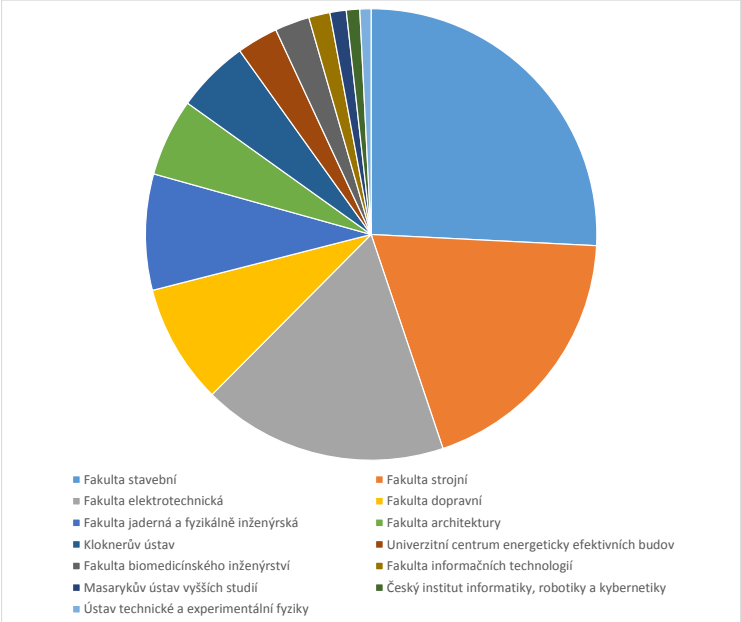
Tabulka 6.2 je setříděná podle celkových vědeckých výsledků učitelů na FIT s ohledem na signifikanci spoluautorství (pokud učitel s někým na vědecké práci spolupracoval, je v v této metrice započítána pouze jeho poměrná část uvedená ve V3S).

osoba	výuka	věda	abstr	člán	vzor	SW	stať
7f79	334	36,54	0	3,91	0	11,50	15,30
3a84	258	31,65	0,33	1,83	0	6,67	16,53
0ae6	212	30,46	0	17,70	0	0	2
3956	435	26,92	0,67	0	11,00	1,30	6,58
4525	407	24,66	1,50	0	0,50	,040	17,17
5ef8	483	22,86	0,33	5,35	0	0,60	7,74
302f	195	21,64	1	6,35	0	0	8,30
2814	342	19,87	0,67	1,01	0,50	1,17	13,17
92b5	289	18,69	0	0,63	0	1,18	11,79

Tabulka 6.2: Učitelé z FIT a jejich celková výuka a suma poměrných vědeckých výsledků bez časového ohraničení, setříděno dle celkové vědy

6.2 Reporty datamartu Počet citací ČVUT

Příkladem použití datamartu Počtu citací ČVUT je report udávající poměr citací vědeckých prací vzniklých na jednotlivých fakultách a ústavech ČVUT.

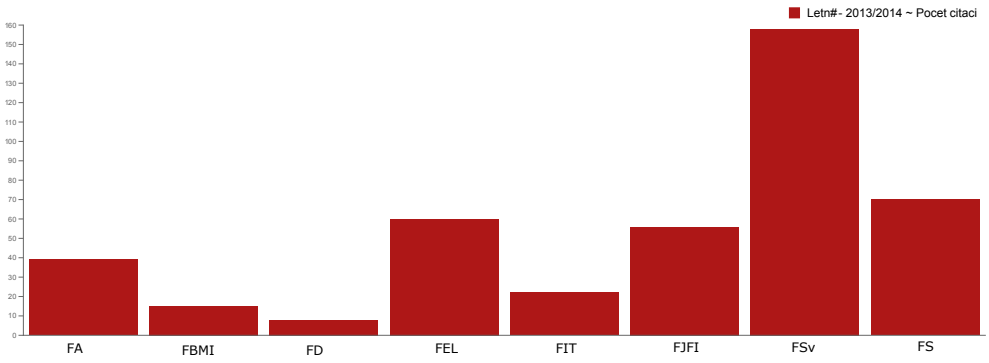


Obrázek 6.3: Poměr počtu citací vědeckých výsledků na jednotlivých fakultách za semestr B151

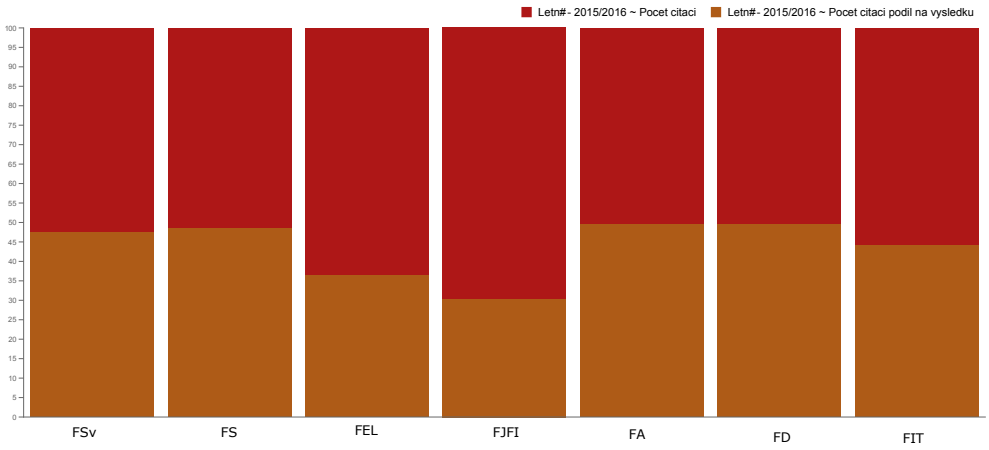
Konkrétní ukázka reportu zobrazeného na obrázku 6.3 je omezená na semestr B151, ve kterém byla, co se citací vědeckých prací týče, nejúspěšnější Fakulta stavební, následovaná Fakultou strojní a Fakultou elektrotechnickou.

Jiným příkladem použití datamartu Počtu citací ČVUT je report udávají počet citací vědeckých prací, na kterých spolupracovali akademici z ČVUT. Na obrázku 6.4 je vidět, že v letním semestru 2013/2014 byla v tomto směru první Fakulta stavební, s bezmála sto šedesáti citacemi.

Další report ukazuje poměr počtu citací a počtu citací vztažených na podíl na výsledku. Kdyby byly všechny hodnoty rozpůlené uprostřed, znamená to, že fakulta v daném roce na žádných výsledcích s nikým dalším nespolupracovala. Report na obrázku 6.5 říká, že v letním semestru 2015/2016 Fakulta jaderná a fyzikálně inženýrská na hodně citovaných vědeckých výsledcích poměrně intenzivně spolupracovala s dalšími organizacemi, naopak Fakulty architektury, dopravní a strojní měly v daném roce citované zejména čistě své vědecké výsledky.



Obrázek 6.4: Počet citací vědeckých prací, na kterých spolupracovali akademici z ČVUT, data z letního semestru 2013/2014



Obrázek 6.5: Poměr počtu citací a počtu citací vztažených na podíl na výsledku, data za letní semestr 2015/2016

6.3 Reporty datamartu Vědecký výkon učitele ČVUT

Následující report srovnává počet vědeckých výsledků jednotlivých pracovišť na ČVUT. V tabulce 6.3 je zobrazeno několik v tomto směru nejúspěšnějších pracovišť ČVUT, spolu se sumou počtu poměrných výsledků. První místo za semestr B151 patří Katedře mechaniky na Stavební fakultě.

sum	**pracoviště**	**fakulta**
178.7	Katedra mechaniky	F. stavební
164.5	Ústav letecké dopravy	F. dopravní
154.7	Katedra betonových a zděných konstrukcí	F. stavební
137.5	Ústav mechaniky, biomechaniky a mechatr.	F. strojní
136.0	Oddělení experimentálních a měřicích metod	Klokn. ústav
109.3	Ústav letadlové techniky	F. strojní
107.6	Ústav výrobních strojů a zařízení	F. strojní
100.3	Ústav strojírenské technologie	F. strojní
98.75	Katedra konstrukcí pozemních staveb	F. stavební
93.96	Katedra ocelových a dřevěných konstrukcí	F. stavební
89.37	Ústav teorie a dějin architektury	F. architektury
84.42	Katedra kybernetiky	F. elektrotechnická
83.84	Katedra počítačů	F. elektrotechnická
79.85	Katedra geomatiky	F. stavební
77.73	Katedra radioelektroniky	F. elektrotechnická

Tabulka 6.3: Suma poměrných výsledků jednotlivých pracovišť na ČVUT za semestr B151

Poslední report je obdobný jako předchozí s rozdílem granulity a jiného časového určení: Místo pracovišť jsou hodnoceni konkrétní lidé a semestr je tentokrát zvolen B091. Jak je vidět v tabulce 6.4, nejvyšší sumu poměrných výsledků má pracovník z Fakulty strojní z Ústavu mechaniky (jméno anonymizováno).

6.4 Výhody a nevýhody technologie OLAP

OLAP je pouze jednou z mnoha technik datové analýzy a jako každý přístup má své lepší i horší vlastnosti. Tato kapitola si klade za cíl čtenáře seznámit s hlavními pozitivy a negativy technologie OLAP.

osoba	sum	fakulta	pracoviště
3e06	29.2	F. strojní	Ústav mech., biomech. a mechatr.
c8f1	28.0	F. strojní	Ústav strojírenské technologie
e340	25.0	F. elektrot.	Katedra fyziky
e340	23.2	F. stavební	Katedra ocelových a dřevěných konstr.
e685	19.4	F. strojní	Ústav automob., spal. motorů a kol. voz.
f1a7	19.3	Klokn. ústav	Oddělení spolehlivosti konstrukcí
437d	18.9	F. strojní	Ústav mechaniky tekutin a energetiky
091c	18.4	F. stavební	Katedra betonových a zděných konstr.
464e	17.3	F. stavební	Katedra zdravotního a ekologického inž.
036e	17.2	FJFI	Katedra fyzikální elektroniky

Tabulka 6.4: Suma poměrných výsledků jednotlivých pracovníků na ČVUT za semestr B091

6.4.1 Výhody technologie OLAP

Hlavní výhody technologie OLAP jsou následující:

- Konzistentní informace a kalkulace: nezáleží na tom, jak rychle nebo kolik dat je skrz OLAP zpracováno, výsledky reportu jsou prezentovány konzistentně, takže analytici a manažeři v každém okamžiku ví, kterou informaci kde hledat. To je důležité především při srovnávání historických reportů a při vytváření strategií do budoucna.
- What-if analýza: možnost vytvářet podmíněné scénáře a do jisté míry predikovat do budoucna.
- Generování automatizovaných reportů čitelných pro člověka, a to i z obrovského množství dat.
- Multidimenzionální prezentace dat analytikovi umožňuje používat metody slice and dice a data drilling, díky čemuž dokáže získat jak celkový pohled na danou problematiku, tak i detailní informace o konkrétní entitě.
- Díky multidimenzionálnímu pohledu je možné objevit a pochopit dříve neznámé vztahy mezi entitami.
- OLAP vytváří jednotnou a jednoduchou platformu pro plánování, reportování, analýzu i predikci.
- Reporty mající podporu v datamartech lze vytvořit extrémně rychle.
- Agregovaná data (třeba počítané metriky) ve faktových tabulkách se předpočítávají, takže není nutné při každém dotazu počítat vše znovu.

6.4.2 Nevýhody technologie OLAP

Hlavní nevýhody technologie OLAP jsou následující:

- Na všechno je nutné mít model/datamart, který nějakou dobu trvá vyvinout.
- Velká závislost na IT znalostech: analytik, který vytváří datamarty musí mít relativně obsáhlé odborné znalosti o dimenzionálním modelování a skriptování (zejména SQL). V případě tvorby integrované vrstvy k tomu ještě přibývají znalosti týkající se ETL.
- OLAP není obecně vhodný pro jakýkoli typ interaktivní nebo ad-hoc analýzy, protože na vše je potřeba mít připravený model.
- Reakce na změnu je pomalá, protože je potřeba měnit i model.
- OLAP funguje pouze pro strukturovaná data, která se většinou musí přiohnout, aby respektovala dimenzionální model.
- Předpočítaná data ve faktové tabulce se neaktualizují v reálném čase.

Závěr

V úvodu byl popsán cíl této práce, který se týkal integrace V3S do Datového skladu ČVUT v celé délce datového cyklu. Tento cíl se podařilo splnit v plném rozsahu. V teoretické části práce se čtenář seznámil s problematikou datového skladování, vyjasnil si terminologii a dozvěděl se vše potřebné o základních architekturách datových skladů. Praktická část pak navázala na nově nabyté teoretické znalosti a provedla čtenáře jak analýzou zdrojového systému, tak i návrhem a popisem realizace stage, integrované, sémantické, přístupové i prezentační vrstvy nově vzniklé části Datového skladu ČVUT. V závěru práce si pak čtenář mohl zkontrolovat funkčnost řešení prozkoumáním prezentační vrstvy (reportů) vzniklými nad daty z V3S, nebo přečíst o výhodách a nevýhodách technologie OLAP.

Přínos integrace V3S do Datového skladu ČVUT se dá chápat ve dvou rovinách: první rovina je obecnější, V3S se integrací zařadil k řadě dalších systémů (KOS, Anketa ČVUT, Závěrečné práce, Portál spolupráce s průmyslem, Progtest nebo Edux), které v Datovém skladu ČVUT už jsou, nebo jejich integrace právě probíhá. Tuto rovinu lze chápat jako součást přirozeného rozvoje Datového skladu ČVUT a benefitem čistě z důvodu „čím víc integrovaných systémů, tím větší potenciál“. Pro druhou, konkrétní rovinu je nutné si uvědomit, že ČVUT není pouze vzdělávací institucí, ale i institucí vědeckou. Tato skutečnost je ale Datovým skladem poprvé reflektována až po integraci V3S, který nabízí právě data o výsledcích vědecké činnosti na ČVUT. Po integraci V3S z je možné provádět daleko podrobnější analýzy, než které nabízí současné webové rozhraní dostupné z `https://v3s.cvut.cz`. Zároveň je možné data z V3S kombinovat s dalšími datovými zdroji. Příkladem takové kombinace je spojení V3S dat o vědecké činnosti vědců na ČVUT s jejich lektorskou činností, které umožní získat komplexní metriku pro hodnocení akademických pracovníků na ČVUT.

Tématem případné návazné práce by mohla být implementace dalších datových tržišť sloužících specifickým potřebám konkrétních uživatelů, třeba i v kombinaci s dalšími zaintegrovanými datovými zdroji. Jiným a v této

práci minimálně prozkoumaným tématem by mohl být způsob prezentace dat, zejména v oblasti vizualizace. Z hlediska vizualizace výsledků totiž v práci použité Saiko nabízí pouze absolutní základ.

Literatura

[1] Kuznetsov, S.: *Datový sklad fakulty*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.

[2] Inmon, W. H.: *Building the data warehouse*. Wiley, 1992.

[3] Inmon, W. H.: *Building the data warehouse*. Wiley, 2011.

[4] Kimball, R.; Ross, M.: *The data warehouse toolkit: the definitive guide to dimensional modeling*. Wiley, 2013, ISBN 978-1-118-53080-1.

[5] Enterprise Data Warehouse Bus Architecture. Navštíveno 1. 3. 2017. Dostupné z: `http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/kimball-data-warehouse-bus-architecture/`

[6] Kimball vs. Inmon in Data Warehouse Architecture. Navštíveno 1. 3. 2017. Dostupné z: `http://www.zentut.com/data-warehouse/kimball-and-inmon-data-warehouse-architectures/`

[7] Breslin, M.: Data Warehousing: Battle of the Giants. *Business Intelligence Journal*, 2004, navštíveno 10. 3. 2017. Dostupné z: `https://cours.etsmtl.ca/mti820/public_docs/lectures/DWBattleOfTheGiants.pdf`

[8] Data Warehousing - Metadata Concepts. Navštíveno 11. 2. 2017. Dostupné z: `https://www.tutorialspoint.com/dwh/dwh_metadata_concepts.htm`

[9] Operational metadata. Navštíveno 11. 2. 2017. Dostupné z: `http://www.ibm.com/support/knowledgecenter/en/SSZJPZ_11.3.0/com.ibm.swg.im.iis.ds.direct.doc/topics/OperationalMetadata.html`

[10] DQ Management. Navštíveno 11. 2. 2017. Dostupné z: https://edux.fit.cvut.cz/courses/MI-EDW.16/_media/lectures/bi_dqmanagement.pdfl

[11] What are Slowly Changing Dimensions? Navštíveno 14. 2. 2017. Dostupné z: http://datawarehouse4u.info/SCD-Slowly-Changing-Dimensions.htmll

[12] Community Wiki Home. Navštíveno 11. 4. 2017. Dostupné z: http://wiki.pentaho.com/display/COM/Community+Wiki+Home

[13] Pentaho about us. Navštíveno 11. 4. 2017. Dostupné z: http://www.pentaho.com/about

[14] How Do You Make a Pentaho? Navštíveno 17. 2. 2017. Dostupné z: http://www.computerworlduk.com/it-business/how-do-you-make-a-pentaho-3568902/

[15] Hitachi to Buy Pentaho to Bolster Data-Analysis Software Tools. Navštíveno 17. 2. 2017. Dostupné z: https://www.bloomberg.com/news/articles/2015-02-10/hitachi-to-buy-pentaho-to-bolster-data-analysis-software-tools

[16] Getting Started With Pentaho BI Server 5, Mondrian and Saiku. Navštíveno 11. 4. 2017. Dostupné z: http://www.joyofdata.de/blog/getting-started-with-pentaho-bi-server-5-mondrian-and-saiku/

[17] Saiku documentation. Navštíveno 11. 4. 2017. Dostupné z: http://saiku-documentation.readthedocs.io/en/latest/

[18] Projekt Rozvoj EZOP a V3S. Navštíveno 11. 1. 2017. Dostupné z: https://portal.cvut.cz/informace-pro-zamestnance/informacni-system-cvut/vysledky-vedy-a-vyzkumu-v3s/projekt-rozvoj-ezop-a-v3s/

[19] Kotlář, R.: *Datový sklad ČVUT - způsoby datové integrace*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Příloha A

Seznam použitých zkratek

BI Business entelligence

SCD Slowly changing dimension, pomalu měnící se dimenze

ČVUT České vysoké učení technické

FA Fakulta architektury ČVUT

FBMI Fakulta biomedicínského inženýrství ČVUT

FD Fakulta dopravní ČVUT

FEL Fakulta elektrotechnická ČVUT

FIT Fakulta informačních techonologií ČVUT

FJFI Fakulta jaderná a fyzikálně inženýrská ČVUT

FS Fakulta strojní ČVUT

FSv Fakulta stavební ČVUT

ETL Extract, transfom, load

KPI Key performance indicators

OLAP Online Analytical Processing

SCD Slowly changing dimension

SQL Structured query language

V3S Aplikace na evidenci výsledků vědy a výzkumu

XML Extensible markup language

Příloha **B**

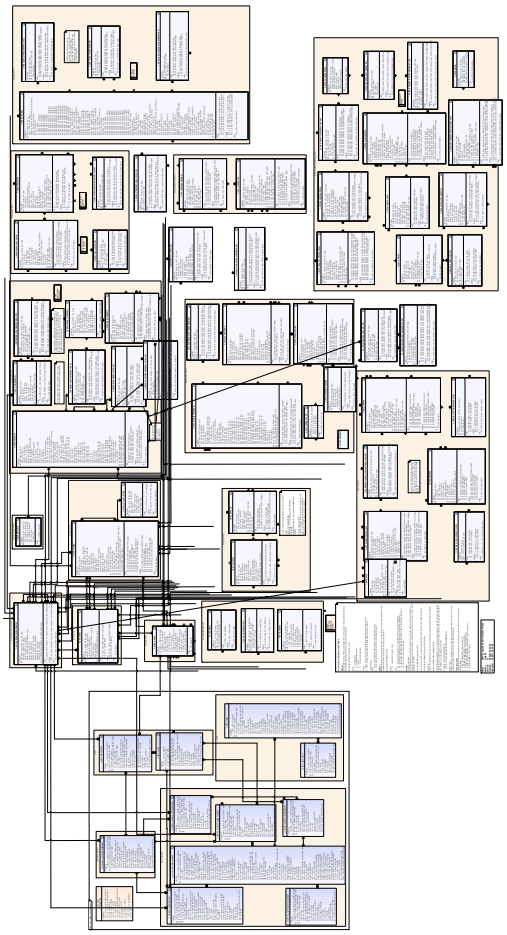
Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `impl` ..... PBI soubory, Enterprise Architect model a SQL skripty
    - `datamarts_cln.SQL` ..... datamartová vrstva
    - `datamarts_create table mock.SQL` .. vytvoření tabulek z pohledů (aby je vidělo Saiko)
    - `model.eap` ..... EA model
    - `SEMANTIC.SQL` ..... sémantická vrstva
    - `SS_STG.ktr` ..... transformace ze stage do integrované vrstvy
    - `STG_CORE.ktr` ..... transformace ze zdrojového systému do stage
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
    - `bp.tex` ..... zdrojová forma práce ve formátu LaTeX
    - `bp.bib` ..... zdrojová forma citací ve formátu LaTeX
- `text` ..... text práce
  - `dp.pdf` ..... text práce ve formátu PDF
  - `Logický model datového skladu včetně V3S.pdf` . model Datového skladu ČVUT s integrovaným V3S

Příloha C

Celkový datový model

Obrázek C.1 vyobrazuje celkový model Datového skladu ČVUT s integrovaným V3S (část vlevo). Podrobnější prozkoumání je možné pouze v elektronické verzi, kde se model dá přiblížit.



Obrázek C.1: Schématický obrázek Datového skladu s integrovaným V3S (část vlevo)