Bakalářská práce

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Analýza výsledků závěrečných prací pomocí business intelligence

Švec Petr

Vedoucí práce: Ing. Lukáš Zoubek
Obor: Softwarové inženýrství a technologie
Květen 2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Švec** Jméno: **Petr** Osobní číslo: **474713**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Analýza výsledků závěrečných prací pomocí business intelligence

Název bakalářské práce anglicky:

Analysis of final thesis results using business intelligence

Pokyny pro vypracování:

1) Definujte základní pojmy týkající se reportingu a analýzy dat
2) Definujte požadavky na nástroj pro reporting a analýzu dat o studiu na FEL
3) Vyberte vhodný nástroj či implementujte vlastní řešení
4) Vhodnost ověřte na datech o závěrečných pracích na FEL za posledních pět let (vytíženost vedoucích a oponentů, jejich hodnocení, porovnání mezi hodnoceními vedoucích, oponentů a komise u obhajoby práce)
5) Zhodnoťte vhodnost nástroje

Seznam doporučené literatury:

[1] Robert Laberge. Datové sklady Agilní metody a business Inteligence. COMPUTER PRESS, 2012.
[2] Zuzana äedivá Jan Pour Iva Stanovská, Maryöka Miloö. Self Service Business Intelligence: Jak si vytvo□it vlastní analytické, plánovací a reportingové aplikace. Grada Publishing a.s., 2018.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Lukáš Zoubek, Centrum znalostního managementu FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **14.02.2020** Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce: **30.09.2021**

Ing. Lukáš Zoubek podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych poděkovat vedoucímu bakalářské práce Ing. Lukáši Zoubkovi za ochotu a cenné rady.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 21. května 2020

Abstrakt

Tato bakalářská práce se zabývá implementací systému pro analýzu výsledků závěrečných prací. Na základě této analýzy lze například určit vytíženost jednotlivých vedoucích či oponentů, porovnávat hodnocení závěrečných prací napříč studijními obory atp. Tyto informace zlepší informovanost vedení fakulty.

Klíčovou částí systému je business intelligence nástroj Metabase, který umožňuje uživatelům jednoduše vytvářet nové dotazy a zobrazovat výsledky pomocí přehledných vizualizací. Během návrhu řešení byl kladen důraz na architekturu tak, aby byla použitelná i na projektech obdobného charakteru.

Vytvořený systém byl úspěšně nasazen do pilotního provozu na FEL ČVUT.

Klíčová slova: business intelligence, datové tržiště, Metabase

Vedoucí práce: Ing. Lukáš Zoubek

Abstract

This bachelor thesis deals with the implementation of a system for analyzing the results of final theses. Based on this analysis, it is possible, for example, to determine the workload of individual supervisors and opponents or to compare the evaluation of final theses across study fields, etc. This information will improve the knowledge of faculty management.

A key part of the system is the business intelligence tool Metabase, which allows users to easily create new queries and display results using clear visualizations. During the design of the solution, the emphasis was placed on the architecture so this architecture can be used on similar projects.

The created system was successfully deployed into pilot operation at FEE CTU.

Keywords: business intelligence, data market, Metabase

Title translation: Study data analysis using BI software

Obsah

Obrázky

Výpisky

Tabulky

Kapitola 1
Úvod

1.1 Motivace

Tato bakalářská práce je reakcí na chybějící reporting dat, týkajících se bakalářských a diplomových prací na fakultě elektrotechnické ČVUT. Zejména vedoucí kateder a garanti programů vyžadují přehledy vytíženosti jednotlivých vedoucích, oponentů, jejich hodnocení, výsledky jejich svěřenců atp. Studijní informační systém KOS tato data obsahuje, ale zobrazuje je nedostatečně. Vedoucí kateder a garanti programů nemají k datům přístup. Tato bakalářská práce má přispět ke změně této situace: vypovídající přehledy a přístupy k datům zlepší informovanost vedoucích pracovníků fakulty.

1.2 Cíle práce

Cíle práce jsou:

- Definovat základní pojmy týkající se reportingu a analýzy dat.
- Definovat požadavky na nástroj pro reporting a analýzu dat o studiu na FEL.
- Vybrat vhodný nástroj či implementovat vlastní řešení.
- Vhodnost ověřit na datech o závěrečných pracích na FEL za posledních pět let (vytíženost vedoucích a oponentů, jejich hodnocení, porovnání mezi hodnoceními vedoucích, oponentů a komise u obhajoby práce).
- Zhodnotit vhodnost nástroje.

1.3 Struktura práce

Struktura práce se odvíjí od jejích cílů, skládá se ze čtyř propojených a na sebe navazujících částí.

- Teorie: cílem je seznámení se s problematikou datových skladů a business inteligence.

- Analýza: Sběr požadavků na nástroj pro reporting a analýzu dat. Přehled existujících BI nástrojů. Vyběr nejlepšího řešení s ohledem na požadavky. Analýza zdrojů dat.
- Popis postupu Implementace.
- Zhodnocení zvolených nástrojů a závěr.

Kapitola 2
Teorie

Tato část pojednává o základních principech business intelligence a datových skladů. Teorie obsahuje zavedení pojmů a „best practices“, které si bylo třeba ujasnit před vhledem do samotné problematiky a tomu odpovídá adekvátně navržené řešení. Pro ujasnění pojmů byly nápomocny především tyto dvě publikace Datové sklady Agilní metody a business inteligence [1] a novější publikace Self Service Business Intelligence [2]. Publikace podávají ucelený pohled na business inteligence systémy.

2.1 Základní pojmy

Terminologie pojmů, na kterých jsou BI systémy postavené.

Informace

Informace jsou data s určitým kontextem, který nám pomáhá datům porozumět. Data bez kontextu nejsou informací.

Informační systém

Definice informačního systému se různí.
Definice podle knihy Podnikové informační systémy udává:

> *„Informační systém je soubor lidí, technických prostředků a metod (programů), zabezpečujících sběr, přenos, zpracování, uchování dat, za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení.“ [3]*

Obecně lze říci, že informační systémy jsou systémy, které pomocí procesů zpracují vstupní informace a převedou je na informace výstupní. Informační systém může, ale nemusí být podporován počítačem. Správně implementovaný informační systém může mít za následek podporu postupů, zvýšení efektivity, optimalizaci procesů a snížení nákladů. [8]

Analýza dat

Analýza dat je proces vyčištění, transformování a modelování dat s cílem objevit užitečné informace pro formování závěrů a podporu rozhodování. Analýza dat se užívá v různých oblastech podnikání i vědy. [5]
S rostoucí digitalizací roste také počet dat, jejichž pochopení poskytne uživateli přehled nad danou situací, data se tedy stanou informací, což může mít za následek například konkurenční výhodu. Firmy si situaci uvědomují a analýza dat v korporacích se stává čím dál tím důležitějším pojmem.

Normálové formy

Normálové formy se využívají u relačního databázového modelu, hlavním účelem takzvané normalizace je zmenšení redundance dat v tabulkách. Každá normálová forma musí splňovat podmínky předchozí normálové formy. Pro účely bakalářské práce se uvádějí pouze první tři normálové formy, jelikož vyšší normálové formy se v práci nevyskytují. První normálová forma obsahuje atomická, dále nedělitelná data. Sloupec v tabulce tedy neobsahuje například seznam hodnot. Druhá normálová forma musí splňovat podmínku, kdy neklíčový atribut nesmí být závislý pouze na částečném klíči, ale na klíči celém. Třetí normálová forma (3NF) odstraňuje transitivní závislosti v tabulkách. [9]

2.1.1 LDAP

LDAP (Lightweight Directory Access Protocol) je protokol pro adresářovou informační službu, využívající stromovou strukturu. Protokol umožňuje k jednotlivým osobám přiřadit role. Příkladem využití protokolu je určení struktury zaměstnanců ve firmě. Pomocí této struktury a autentizace uživatele lze v jiných aplikacích určovat jeho práva, například na základě toho, zda má uživatel přiřazenou danou roli. [17]

2.2 Business inteligence

Definování základních pojmů v oblasti BI, pojmy Business intelligence a datový sklad.

- Business intelligence

  > *„Business intelligence je zastřešující termín, který se vztahuje ke znalostem, procesům, technologiím, aplikacím a postupům, které usnadňují podnikové rozhodování. Technologie business intelligence pracuje s použitými (historickými) daty v požadovaném kontextu a pomáhá přijímat podniková rozhodnutí pro budoucnost.“[1]*

- Datový sklad

> *„Datový sklad (data warehouse) je systém, který umožňuje shromažďovat, organizovat a sdílet historická data. Datový sklad může být zaměřen na celý podnik i pouze na určitý obor činnosti.“ [1]*

2.2.1 Motivace pro BI

Hlavní motivací je využití dat rozmístěných po informačních systémech. Data v systémech jsou často duplicitní, neúplná, nebo obsahují pouze jeden úhel pohledu. Pro management může být proto obtížné dělat ať už strategická rozhodnutí, operativní rozhodnutí, či vyhodnocení aktuálního stavu pomocí sestav založených na datech pouze z jednoho ze systémů. Naopak, pokud se data využijí/zobrazí správně, mohou managementu poskytnout pomocný nástroj při rozhodování. Jejich rozhodnutí jsou následně podpořena úplnými, kvalitními daty.

2.2.2 Přínosy BI

Podle knihy Datové sklady [1] patří mezi hlavní přínosy BI následující body:

- vysoké pokrytí sestavami, například díky možné změně atributů uživatelem, či agregovací metrik
- jednotná pravda dat, všechna data jsou na jednom místě
- lepší kvalita dat, data jsou vyčištěna a zbavena duplicit
- lze dělat daty podložená strategická rozhodnutích
- lze sledovat efekt rozhodnutí učiněných v minulosti

2.2.3 Nevhodná tvorba BI

Výše jsou udané důvody, proč budovat datový sklad. V knize Datové sklady [1] je podáván i opačný pohled na situaci, kdy se naopak budovat BI řešení nedoporučuje, jelikož by jeho tvorba byla velice obtížná nebo by nepřinesla tížený efekt.

- *Zdrojové systémy neobsahují dostatečnou kvalitu dat.* Tento nedostatek se v některých případech nedá vyřešit pomocí transformace, či filtrace. Pokud se rozhodnete v takové situaci vytvářet BI, je možné, že projekt spotřebuje mnoho zdrojů a nepřinese tížený efekt.
- *Management není BI řešení nakloněn.* Pokud BI řešení není nakloněno vedení společnosti, ale například jen vedoucí oddělení, je nejspíše pouze otázkou času, kdy se zastaví financování projektu. Rychlý přínos BI je poměrně těžké ukázat. Projekty minimálně trvají kolem šesti měsíců a během tvorby je velice podstatná spolupráce právě s vedením společnosti.

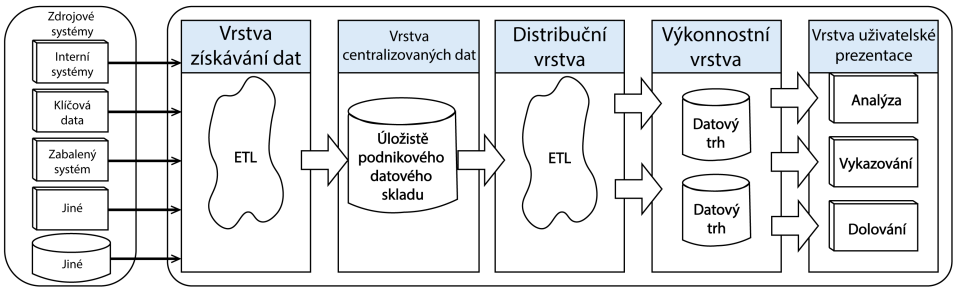
- *Cíl není jasný.* Pokud cíl projektu není jasný a BI řešení se zaměřuje na sledování a reporting nedůležitých atributů, poté řešení neposkytuje potřebnou přidanou hodnotu. BI, stejně jako každý projekt, potřebuje návratnost investice a bez stanoveného cíle není možné cíle dosáhnout.
- *Rychle se měnící prostředí.* Pokud firma expanduje do odlišných podnikových sfér, musel by se s tím měnit i model datového skladu a další komponenty, které si ukážeme dále. Z toho důvodu se v tomto případě tvorba BI nedoporučuje
- *Přidaná hodnota nebude vysoká.* Pokud BI vlastně nikdo nevyžaduje a uživatelům stačí, co poskytuje jejich aktuální informační systém, je zbytečné vytvářet nové BI.

2.2.4 Architektura Datového skladu

Existuje několik druhů architektury datového skladu. V této práci bude vysvětlena pouze nejobsáhlejší architektura, architektura centrálního úložiště (viz obrázek 2.1). Ostatní architektury se liší pouze absencí některé z vrstev.

Architektura centrálního úložiště

V architektuře centrálního úložiště se nejprve data čerpají ze zdrojových systémů. Následuje jejich transformace, čištění, odstranění duplicit (proces ETL). Poté se uloží do datového skladu a pošlou do jednotlivých datových trhů. Datové trhy již nejsou obecné, ale mají nějaký předem daný účel, oblast zájmu. Jelikož se datové modely a způsoby ukládání informací mezi datovým skladem a datovými trhy liší, je opět potřeba data před uložením přetransformovat. Datové trhy jsou úložištěmi, ze kterých se čerpá pro následnou prezentaci dat.



Obrázek 2.1: Architektura centrálního úložiště, zdroj [1] (překresleno)

2.3 Komponenty BI

2.3.1 Zdrojové systémy

Zdrojové systémy jsou systémy, ze kterých se čerpají data. Data z těchto systémů mají různou kvalitu a jsou v různých formátech od CSV, přes data

zasílána přes REST rozhraní ve formátu JSON, až po data z relační databáze.

2.3.2 ETL

Problémy s daty ze zdrojových systémů řeší vrstva ETL. Zkratka ETL je složenina tří slov Extract (= získat data ze zdrojových systémů),Transform (= převést data do vhodné podoby), Load (= načíst data do datového skladu). Tato vrstva zajistí kvalitu, převedení a případné obohacení dat. Obohacení dat znamená, že pokud data obsahují například IČO, lze z veřejně dostupného obchodního rejstříku načíst informace o firmě a s těmito daty dále pracovat.

2.3.3 Centrální úložiště

Centrální úložiště je používané v Inmonově přístupu budování datových skladů (viz 2.4).

Data převedena vrstvou ETL jsou nahrána do centrálního úložiště. Centrální úložiště obsahuje všechna potřebná data pro podnik. Tato struktura je jádrem celé architektury a bez ní není možné datový sklad udržovat. [6]

Datový sklad je tvořen, aby existovalo jednotné „místo pravdy“, ze kterého následně čerpají datová tržiště.

Podle [1] poskytne vytvoření centrálního úložiště tyto výhody:

- Konzistentní vykazování,
- opakované použití nahraných dat z externích systémů,
- jednotná verze pravdy pro celý podnik.
- V datovém skladu lze udržovat historii dat (zdrojový systém bývá optimalizován z hlediska použití a historii dat neudržuje).
- Datový sklad si aktualizuje data v pravidelných intervalech, poté se čerpá z jeho databáze. Tím pádem zdrojový systém není zatížen náporem častého dotazování na reporty, což by nejspíše znamenalo jeho zpomalení.

2.3.4 Operační datový sklad

Operační datový sklad slouží k uchovávání aktuálních dat bez důrazu na uchovávání historie. Díky tomu je možné se nad daty rychle dotazovat. Zároveň operační sklad slouží jako „cache“ zdrojových systémů a lze tuto vrstvu zařadit mimo systém datového skladu, nebo po ETL před datový sklad, který z operačního datového skladu může jednou za čas přečerpat informace. [6]

2.3.5 Datový trh

Datový trh je oproti datovému skladu modelován a používán pro specifický účel. Pokud datový sklad je celopodnikové řešení, datové tržiště by byla vytvořena pro jednotlivá oddělení a jejich potřeby. Decentralizovaný přístup

postupného budování datových tržišť razí Kimbal 2.4 . Podle publikace Datové sklady [1] se přístup vybudování několika tržišť „na zkušenou“ doporučuje před započetím budování celopodnikového řešení. Účelu vybudování je podřízen i návrh modelu databáze (viz 2.4.1).

2.3.6 Prezentační vrstva

Definice prezentační vrstvy podle knihy Self Service Business Intelligence:

> „Prezentační vrstva představuje komplexní ukazatele informací a činností společnosti, které poskytuje včas a ve vhodné formě. Informace jsou prezentovány pomocí reportů, případně interaktivních manažerských dashboardů. Reporting je jedním z hlavních výstupů BI řešení.“ [2]

2.4 Přístupy návrhu BI

V návrhu architektur BI existují dva hlavní přístupy. Zatímco Ralph Kimball tvrdí, že datový sklad není nic jiného než uskupení všech datových tržišť a zastává teorii zdola nahoru, tedy účelové tvoření aplikací a budování jejich sítě, Inmon má jiný přístup, když píše: „*Můžete chytit všechen plankton na světě, ale stále nemáte velrybu.*“ Inmon je tedy pro sjednocení všech dat a až jejich následné využívání. [6] [1]

2.4.1 Dimenzionální modelování

Výstižné shrnutí výhod a nevýhod Dimenzionálních modelů poskytuje publikace Reconsidering Multi-Dimensional Schemas [4]. Uvádí užití tří možných návrhů databází, které jsou uvedeny v následujících pododstavcích.

Třetí normálová forma (3NF)

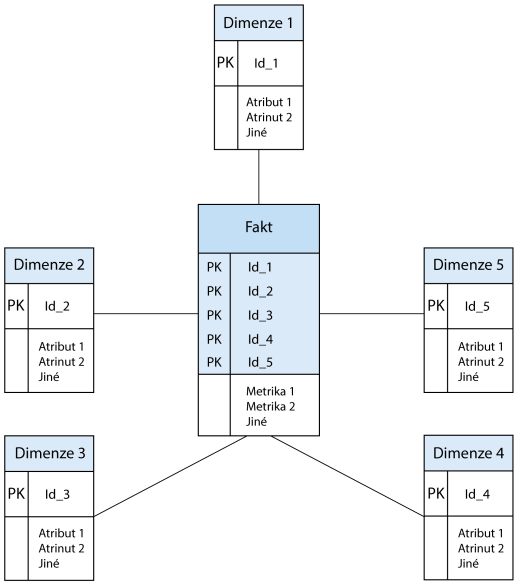
Třetí normálová forma je nejkomplexnější schéma - nevýhodou tohoto návrhu je čitelnost, případně rychlost komplikovaných dotazů.

Schéma hvězdy (star schema)

Schéma hvězdy je nejjednodušší. Výhody jsou například méně JOIN operací, z toho vyplývá větší rychlost a jednoduché upravování schématu. Nevýhodou je redundance dat v dimenzích. Ve schématu hvězdy je jedna tabulka faktů obklopena dimenzemi (viz obr. 2.2).

Schéma vločky (snowflake schema)

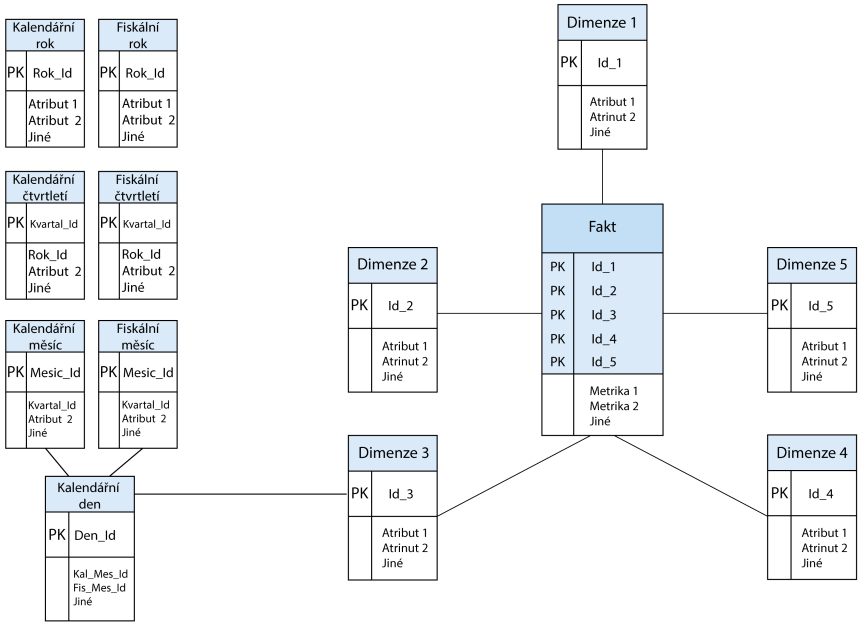
Schéma vločky je určitým kompromisem mezi komplexním 3NF a příliš jednoduchým hvězdicovým schématem. Výhodou tohoto schématu je, že lze oproti schématu hvězdy zobrazit hierarchickou strukturu.



Obrázek 2.2: Star schema, zdroj [1] (překresleno)

Tabulky faktů a dimenze

Tabulka faktů je střed multidimenzionálního návrhu, jde o centrální tabulku, která obsahuje metriky. Je obklopena a cizími klíči navázána na tabulky dimenze, které dále definují centrální tabulku. Ideálně nad dimenzemi lze dělat agregace, či například filtrování. Typickou tabulkou dimenze je například tabulka „Kategorie produktů“, která je navázána na tabulku faktů „Prodané produkty“.



Obrázek 2.3: Snowflake schema, zdroj [1] (překresleno)

Kapitola 3

Analýza

3.1 Popis aktuální situace

Aktuální situace úzce souvisí s motivací (viz 1.1). Hlavním problémem je chybějící reporting ohledně bakalářských a diplomových prací na fakultě elektrotechnické ČVUT. Složitý přístup k přehledům nad vytížeností, výsledky a hodnocení jednotlivých prací a osob. Chybí jejich jednoduchý reporting příslušným osobám, jako jsou vedoucí kateder nebo garanti programů.

3.1.1 Analýza zdrojů

Zdroj, ze kterého se budou čerpat data, je informační systém KOS. V průběhu analýzy nebyl nalezen jiný zdroj dat, kde by byly zadávány informace po vyhodnocení závěrečné práce. Toto tvrzení bylo ověřeno konzultací se studijním proděkanem. Všechny procesy týkající se zadávání informací ohledně závěrečných prací mají jen jeden systém, kam ukládají data, a to právě systém KOS.

3.1.2 Čerpání dat ze systému KOS

Čerpání dat je možné dvěma způsoby. První způsob je přímo ze systému KOS. Druhý způsob čerpání dat je přes datový sklad, který je provozován Fakultou informačních technologií.

S ohledem na čerpání dat z jednoho zdroje, neznámou strukturu datového skladu a následné možnosti, že datový sklad neobsahuje všechna potřebná data, bylo po konzultaci s vedoucím rozhodnuto pro první variantu, čerpání dat přímo z dostupných pohledů do systému KOS. Výhodou zvolené varianty je také, že komunikace se třetí stranou a spoléhání na další zdroj by mohla zásadně ovlivnit časovou délku zpracování implementační části závěrečné práce.

Zvolená varianta, čerpání dat přes pohledy do systému KOS. Pohledy neobsahují záznamy o známkách posudků vedoucích a oponentů prací, informace o těchto záznamech budou poskytnuty v podobě souborů ve formátu XLSX.

3.2 Požadavky na software

Požadavky na výsledný software jsou následující:

- Užívaný software musí být zdarma.
- Přístupnost bude z webového prohlížeče.
- Reporting dat bude poskytnut pomocí interaktivních dashboardů.
- Software umožňuje provádění analýzy dat uživatelem.
- Reporty lze uložit a nakládat s nimi dále (například ve formátu PDF).
- Přístup k přihlášení je omezen (neregistrovaný uživatel nebude mít k datům přístup).
- Přihlašování za využití protokolu LDAP (viz 2.1.1).
- Data a připojení k nim budou uložena na serverech ČVUT.

Požadavky musí splňovat celý systém, lze jich docílit kombinací několika softwarových řešení.

V případě tohoto projektu je identifikována skupina aplikaci méně často používajících uživatelů s vysokými právy přístupu k datům. To je důvodem, proč je zapotřebí pro zobrazování výstupů pouze BI nástroj, který musí být ale zároveň maximálně uživatelsky přívětivý, aby byl uživatel schopný se v něm rychle zorientovat.

3.3 Přehled existujících BI řešení splňujících požadavky

Business intelligence nástrojů existuje velké množství. Nástroje byly rozděleny do dvou kategorií: komerční a open source.

Podle zdrojů citovaných dále byly vybrány BI nástroje a u nich udělána analýza, zda splňují výše zmíněné požadavky. Pokud nástroje vzhledem k našim požadavkům využívat nelze, bude uvedeno zdůvodnění. Struktura hodnocení se skládá z popsání výhod řešení, nevýhod řešení a nakonec obsahuje souhrn, popisující případné zásadní překážky.

3.3.1 Komerční nástroje

Byly zvažovány komerční nástroje, pohybující se na špičce žebříčku BI nástrojů od Gartneru [10]. Jelikož plné licence těchto nástrojů jsou poměrně drahé, byly brány v potaz pouze dostupné verze zdarma i s jejich omezeními. Mimo žebříček Gartner byl přidán nástroj google data studio, na který bylo upozorňováno v jiném přehledu ([12]) a jenž je v plné verzi zcela zdarma.

Power BI

Microsoft business inteligence řešení nabízí desktopový nástroj zdarma. Jedná se o nejlépe hodnocený nástroj. Power BI je komplexní end-to-end řešení s výbornou vizualizací a intuitivním ovládáním. Ve free verzi je k dispozici desktop aplikace.

Mezi výhody patří:

- poskytuje real-time reporting,
- lze jednoduše vytvořit reporty.
- Ideální a používané řešení v praxi pro analytika, který se jen připojí na data a reporty si jednoduše tvoří.

Mezi nevýhody patří:

- Pokud bychom chtěli vystavit řešení na web, museli by být data přístupná veřejnosti.

Souhrn: Jediné možné řešení, jak lze powerBI využívat, by bylo sdílení společného souboru a každý uživatel by si musel u sebe na počítači soubor stáhnout a otevřít, což by bylo nepraktické. Druhou možností by bylo sdílení výstupů do vlastních webových stránek. K těmto grafům, ale nelze omezit přístup na přihlášení a mohl by je vidět kdokoli. Ani jedna varianta nesplňuje zadáné požadavky na software.

Tableau

Tableau nabízí studentskou licenci, která dává přístup k desktopové aplikaci, kde lze jednoduše tvořit a zobrazovat reporty. Zároveň zdarma nabízí aplikaci Tableau reader, která slouží pro zobrazování a je zdarma.

Mezi výhody patří:

- možnosti zobrazení,
- výkon a jednoduchost použití,
- responzivní design.

Mezi nevýhody patří:

- Data online lze zobrazit pouze všem (Tableau public).
- Tableau reader slouží pro zobrazování, je zdarma, distribuce těchto souborů vytvořených v dektopové verzi by byl problém.
- Ve sdíleném souboru se nedají automaticky aktualizovat data.

Souhrn: Bylo by třeba aktualizovat soubor pravidelně a u každého uživatele poté zvlášť stahovat. Tableau tedy nelze pro větší počet uživatelů ve free verzi používat.

Google data studio

Odlehčená verze BI nástroje, který je kompletně zdarma.

Mezi výhody patří:

- plná verze zdarma,
- snadná a intuitivní tvorba dashboardů,
- známé prostředí a sdílení pomocí google účtů.
- Lze si vizuály dotvořit, nebo přizpůsobit.

Mezi nevýhody patří:

- Nabízí menší variaci zobrazovacích prostředků.
- Připojení k databázím uloženo na cloudu googlu.
- V jeden okamžik je možné zobrazovat data pouze z jednoho zdroje, to může být v budoucnu problém

Souhrn: Vzhledem k požadavku připojení k databázi pouze na serveru ČVUT toto řešení nepřipadá v úvahu.

3.3.2 Open source

Jelikož open source řešení je obrovské množství, v prvotní analýze byly brány v potaz žebříčky, kde se porovnávaly plusy a mínusy jednotlivých řešení pro zvolení užšího výběru. Podle užitých přehledů a žebříčků ([11] ,[13],[14]), lze obecně říci, že Open source Business intelligence řešení se aktuálně dělí do dvou podkategorií komplexní (end-to-end,zavedená, starší) řešení a moderní řešení zaměřená převážně na vizualizaci dat. Komplexní řešení obsahují načítání, uchování i analýzu dat, ale často s omezeným a nepříliš atraktivním vizuálem, težko převeditelným a customizovatelným do webového prostředí, nebo obtížným pro uživatele (Pentaho, Birt, SpagoBi, Jaspersoft) [12]. Hledáno bylo uživatelsky přívětivější webové řešení, umožňující jednoduchou tvorbu reportů přímo uživateli.

Podle zdrojů ([15], [19]) byla vybrána a porovnána tři řešení, která jsou popsána v následujících podsekcích.

Metabase

Metabase je open-source nástroj pro business intelligence. Hlavní předností Metabase je, že povoluje uživateli rychle a snadno definovat otázky v podobě zjednodušených SQL dotazů.

Mezi výhody patří:

- intuitivnost,

- aktivní komunita,
- jednoduchá instalace,
- jednoduchá tvorba dashboardů i pro netechnické uživatele.
- Uživatel si může vytvořit dotaz a nastavit připomínky, kdy mu pravidelně chodí upozornění („pulses“) na email či Slack.

Mezi nevýhody patří:

- Limitovaná interaktivita mezi dvěmi widgety. To znamená, že kliknutí na jednom elementu v jedné tabulce nezvýrazní automaticky element v jiné.

Souhrn: Neobsahuje žádnou zásadní chybu, připadá proto v úvahu.

Superset

BI nástroj vyvinut Airbnb. Oproti ostatním nástrojům má rozdílné stupně přístupu, podle kterých se uživatelům nastavují práva (Admin, Alpha, Gamma a Public). Superset je napsán v Pythonu.

Mezi výhody patří:

- hezký vizuál komponent,
- jednoduché nasazení.
- Obsahuje mnoho vizualizačních komponent.

Mezi nevýhody patří:

- problematické filtrování napříč widgety.
- Vybírání a agregování dat je méně intuitivní než v Metabase.
- Zobrazení na mobilních zařízeních není optimální.

Souhrn: Neobsahuje žádnou zásadní chybu, připadá proto v úvahu.

Redash

Open source nástroj pro dashboardy a vizualizaci dat. Redash je aplikace psaná v Pythonu a Javascriptu.

Mezi výhody patří:

- Podporuje velké množství datových zdrojů.
- Aktivní komunita

- Jednoduchá instalace a tvorba dotazů
- Lze nastavit automatický refresh (1 minuta - 24 hodin).
- Dobrá responsivita na mobilních platformách

Mezi nevýhody patří:

- Není tolik přístupný netechnickým uživatelům jako Metabase.
- Nelze vytvářet agregace ve vizualizaci. (Je třeba agregace dělat pomocí SQL dotazů, což způsobuje redundanci a předběžné agregace v DB)

Souhrn: Neobsahuje žádnou zásadní chybu, připadá proto v úvahu.

3.4 Zvolený postup

Po zvážení plusů a mínusů vzhledem k požadavkům definovaným v kapitole 3.2 byly nejlépe hodnoceny systémy Metabase, SuperSet a Redash. Po konzultaci s vedoucím byl zvolen nástroj Metabase. V potaz byla brána především jednoduchá tvorba dotazů uživateli a přívětivost prostředí.

Metabase umožňuje připojení k databázovému zdroji, ale neobsahuje ETL nástroj. Pro převedení dat se používá nástroj Talend Open Studio for Data Integration (dále jen Talend), který byl zvolen v „leaders“ kvadrantu společností gartner pro ETL nástroje v roce 2019 [16]. Převedená data se uloží do open source databáze PostgreSQL, ze které bude Metabase čerpat. V databázi bude řádově tisíce záznamů, není proto třeba uvažovat nad databázemi pro OLAP, či big data. Výhodou PostgreSQL je výborná škálovatelnost pro, co do počtu záznamů, větší projekty. [21]

Výhodou tohoto přístupu je, že pokud by se zvolený nástroj Metabase na počátku implementace neukázal jako optimální, lze ho vyměnit a zapotřebí bude předělat pouze reporty do jiného prostředí.

Až bude řešení nasazené na serveru, data se budou aktualizovat pomocí ETL procesu v programu Talend.

3.5 Nalezené obecné vzory po analýze BI nástrojů

Výběr optimálního programu je pevně spojen s požadavky, a to nejen s uvedenými požadavkami na software (viz 3.2), ale esenciální otázkou je také složení uživatelů. Uživatelé se mohou lišit technickou zdatností a intenzitou používání nástroje, což ovlivňuje nároky na výsledný software. Jiný nástroj by byl nejvhodnějším pro dva uživatele tvořící všechny reporty na fakultě, poskytující je dále, a jiný nástroj je nejvhodnějším pro větší skupinu příležitostných uživatelů.

BI nástroj umožňuje uživatelům přehled nad daty, vytváření dotazů a hledaní pro ně zajímavostí v datech. Uživatelská přívětivost a předem vytvořené

prostředí na druhou stranu omezuje možnosti nestandartní vizualizace a přizpůsobení prostředí pro uživatele obecně.

Limitace BI nástrojů by mohla způsobit snahu do těchto nástrojů začlenit dodatečně funkcionalitu. Před možnými zásahy je důležité si uvědomit, o jak velké projekty se jedná, kolik času by zabralo seznámení se s projektem či jak by zásahy mohli zkomplikovat následné aktualizování nástroje.

Pro řídící skupiny uživatelů je zásadní intuitivní dotazování a zobrazování, zatímco pro uživatele, kteří mají omezená práva, dává smysl mít na tyto účely samostatnou reportovací aplikaci. Tvořit tedy aplikace dvě. V jedné aplikaci umožňovat libovolné vytváření dotazů. A ve druhé poskytovat pouze reporting. Nevýhodou takovéhoto typu reportingu je, že je zapotřebí zásah programátora. K reportingu dává smysl užívat embedded dotazů z BI Nástroje pomocí API, případně javascriptových knihoven, například chart.js, D3.js, Cube.js. [24]

Kapitola 4

Zvolené nástroje

4.1 Talend Open Studio for Data Integration

Talend je nástroj určený pro tvorbu ETL procesů. Jedná se o software zdarma (Free open source Apache licence). Poskytuje tvorbu ETL procesů a to pomocí tvorby diagramu procesu tvořeného z jednotlivých komponent.

Stavební prvky jsou v následujícím odstavci hierarchicky popsány od nejkomplexnějších k základním. ETL proces se nazývá „Job“. Job se skládá z jednotlivých, předem vytvořených komponent. Komponenty jsou v diagramu propojeny spojeními („Connection“). Jednotlivé prvky jsou detailněji popsány níže.

4.1.1 Komponenty

Komponenty jsou funkčními jednotkami, jejichž skládáním za sebe vzniká logika ETL procesu. Jednotlivé komponenty jsou, jako celý Talend, psány v Javě. Komponenty mají určené schéma řádku se kterým pracují (schéma viz 4.1.3) a připravená textová políčka pro vstupy, do kterých se zadávají jednotlivé proměnné. Příkladem pro vyjasnění může být komponenta `tFileInputExcel`, pro načtení dat z excelu, se vstupem pro „File name/stream“.

Často užívanými komponentami jsou například:

- `tDBInput`, komponenta pro načtení dat z databáze. Lze použít připojení na databáze uložené v „Db Connections“ v metadatech (metadata viz 4.1.4).
- `tMap`, komponenta, která slouží k převedení dat z jednoho schématu do druhého, komponenta umí i join dvou schémat, nebo provedení libovolné funkce nad daným atributem.
- `tJava`, komponenta pro dopsání vlastního Java Kódu. Touto komponentou Talend poskytuje větší flexibilitu v tvorbě procesu.
- `tRunJob`, komponenta pro spuštění Jobu. Talend tedy umožňuje přepoužití Jobů.

Pro rozšíření repertoáru dostupných komponent v programu slouží stránka exchange.talend.com, kde si lze najít a stáhnout požadovanou komponentu. Posléze je třeba komponentu vložit do složky „MyComponents“, čímž se nahraje do prostředí a lze ji používat. Komponenty jsou tvořeny komunitou a většina je jich k dispozici zdarma, což poskytuje obrovskou variabilitu komponent a plejádu zdrojových/cílových konektorů, které Talend nabízí. Komponenty si lze, v případě potřeby, vytvořit i vlastní. [22]

4.1.2 Spojení

Spojnice neboli „connection“, slouží ke spojení dvou komponent.

Spojnice může být dvou typů:

- `Row`
- `Trigger`

Row je spojnice, kterou postupně „protékají“ jednotlivé řádky. Spojnice předává schéma a data záznamu (řádky) z jedné komponenty do druhé. Row spojnice se tedy vykonávají iteračně, řádku po řádce. Row může mít typy určené dle komponenty, ze které vychází. Příkladem je komponenta `tFilterRow` s Rows typy Filter a Reject.

Trigger reaguje na stav celé komponenty. Typické jsou Triggery typu `OnComponentOk`, `OnComponentError`, či `RunIf`.

4.1.3 Schéma

Schéma udává atributy, které bude daný řádek mít. Každá komponenta a spojení má dané schéma (obyčejně se schéma převezme z předchozí komponenty). Schéma se takto použije v celém procesu. Vyjímkou je například komponenta `tMap`, která mapuje jedno schéma na druhé.

Za zmínku stojí funkce `guess schema`, díky které Talend automaticky určí schéma. Funkce je zakomponována například u komponenty `tDBInput`, kde určí schéma podle zadaného SQL dotazu.

Column	K...	Type	Nullable	Date Patter...	Length	Precision	Default	Comment
number		int			3		0	
txt		String			20			
date		String			10			
flag		String			5			

Obrázek 4.1: Talend Schéma, demo příklad

4.1.4 Ostatní prvky zásadní pro vývoj

- `Rutiny` („Routines“) slouží pro tvorbu vlastních tříd a funkcí, které lze kdekoli použít (umožňuje-li to komponenta).
- `Metadata` v této sekci v programu Talend lze uložit metadata schémat tabulek a atributů. To jak pro jednotlivé připojení do databáze, tak

například i pro XLS soubory. Tato schémata pak lze v programu libovolně používat. Díky tomu je možné změny dělat centralizovaně, na jednom místě.

- Kontextové proměnné („Context Variables“), tyto proměnné lze využívat a přistupvat k nim v jednotlivých komponentách. To při vhodném použití umožňuje měnit následně například cesty k souborům pouze na jednom místě. Kontextové proměnné lze definovat i v rodiči Jobu (rodič je takový job, který volá další job), a tak předávat proměnné v rámci úrovní.

4.2 Metabase

V této sekci jsou popsány licenční podmínky užívání Metabase, seznámení s aplikací a to jak z pohledu uživatele, tak administrátora a bližší vhled nabídne sekce s popisem možností nástroje pro budoucí rozvoj, těmito možnostmi jsou například lokalizace, přizpůsobení prostředí, popis API a vysvětlení embeddingu obsahu.

4.2.1 Licence

Používaná verze Metabase je pod APGL licencí. APGL licence zjednodušeně řečeno udává, že pokud upravíte zdrojový kód programu, musí být také přístupný.

Ve verzi zdarma je upřesněná licence pro embedding dotazy, tyto dotazy musí se zachovat logo Metabase a URL v `divu`, ale je možno embedded dotazy libovolně používat. [25] [29]

AGPL licence

> *„Filosofie licence AGPL vychází z toho, že i nepřímý uživatel (užívající software vzdáleně na serveru) má mít stejná práva, jako kdyby software užíval přímo. Proto AGPL přidává požadavek, že kdo poskytuje uživatelům k dálkovému použití upravenou verzi, musí jim její zdrojové kódy poskytnout stejně, jako kdyby jim poskytoval k místnímu užívání software pod GPL.“ [28]*

4.2.2 Seznámení s aplikací

Během seznámení úvodními kroky provede dobře zpracovaná dokumentace, která je aktualizována s každou novou verzí Metabase. Pro instalaci Metabase si lze vybrat mezi JAR souborem, spuštěním aplikace v Dockeru nebo nasazením aplikace na heroku a další cloudové služby. Pro náš projekt s požadavkem, aby vše bylo na vlastním serveru, je použita první varianta, JAR soubor. Spuštění a nastavení Metabase je otázkou několika minut. JAR soubor je ve složce spolu s pluginy a databázovými soubory Metabase, do kterých se zapisuje vytvořený obsah. Díky těmto souborům je Metabase přenosný, soubory

se pouze převezmou a aplikace se nasadí jinde. Implicitně, již při prvním spuštění je připojená databáze „sample dataset“, nad kterou si lze ozkoušet funkcionality programu.

Stejně jako v oficiální dokumentaci i zde je seznámení s programem rozděleno do tří sekcí:

- produkční část (Operations Guide),
- uživatelská část (Users Guide),
- administrátorská část (Admin Guide).

Produkční část (Operations Guide)

Po instalaci prostředí je z této sekce nejdůležitější postup aktualizace softwaru, které se pro systém běžící s JAR souborem provádí vypnutím aplikace, stažením nové verze z webových stránek Metabase. Nový soubor se ve složce nahradí za starý a aplikace se spustí. Před aktualizací je silně doporučeno zazálohovat databázové soubory Metabase. Pro provoz je důležité brát v úvahu, že nové verze mohou obsahovat chyby a je doporučeno přecházet na verze, které mají již vydaných několik opravných verzí (patchů). Z této sekce jsou pro bezpečnost a běh programu důležité, možnost enkryptování připojení k databázi, možný přechod na HTTPS, či možná změna levelu logování. Lze nastavit i JMX monitoring pro sledování a diagnostiku výkonu.

Obšírnější informace viz oficiální dokumentace Operations Guide (https://www.metabase.com/docs/latest/operations-guide/start.html).

Uživatelská část (Users Guide)

Hlavními znalostmi pro práci uživatele je orientace v kolekcích, tvorba dotazů a tvorba dashboardů.

Kolekce jsou složky ve kterých jsou uloženy dotazy, či dashboardy. Každý uživatel má kromě jemu přístupných kolekcích navíc svou vlastní kolekci, kam si může ukládat libovolné dotazy.

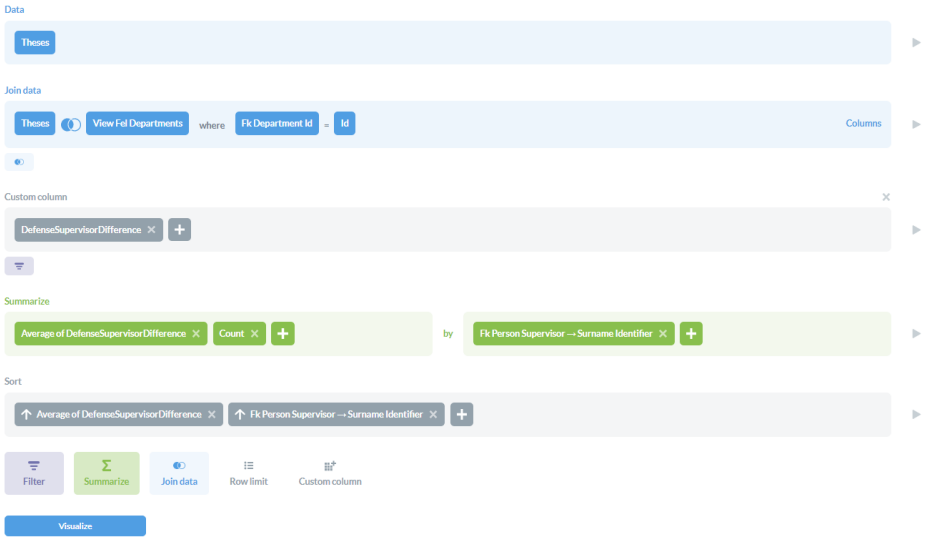
Dotaz lze vytvořit po stisknutí na tlačítko ”Ask a question”, které se nachází v liště vpravo nahoře. Zde se nacházejí tři dlaždice:

- simple question,
- custom question,
- native query.

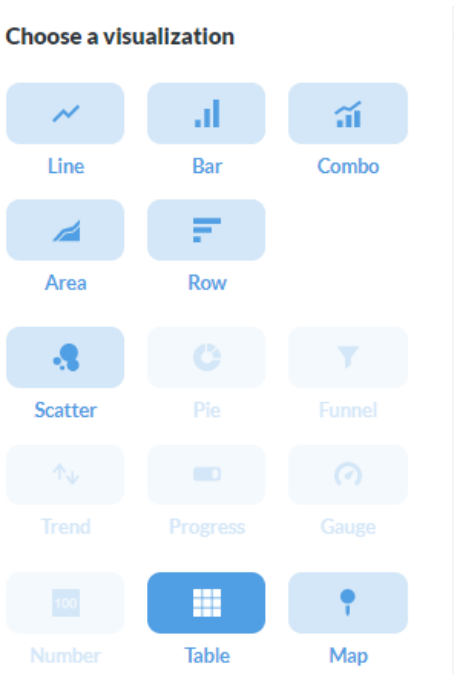
Simple question umožňuje zobrazení jednotlivých tabulek. Zvolíme databázi připojenou na Metabase, označovanou dále jako dataset, a tabulku v této databázi. Tuto tabulku lze filtrovat, či sumarizovat podle zvoleného atributu (volí se metrika, podle jakého atributu sumarizovat). Poté kliknutím na tlačítko „Visualization“, umístěno vlevo dole, lze dotaz vizualizovat zvolením vhodného typu grafu (viz obr. 4.3) a jeho parametrů (os, měřítka atd.).

Custom question slouží pro složitější dotazy, pokud je třeba propojení (JOIN) více tabulek, vytvoření vlastního sloupce (dle nějakého výpočtu), atp.

Native query umožňuje psaní nativních dotazů. Dokonce i s použitím proměnných.



Obrázek 4.2: Metabase tvorba dotazu („custom question“)



Obrázek 4.3: Metabase typy vizualizací dotazů (doporučené jsou zvýrazněny)

Dashboardy lze tvořit kliknutím na tlačítko „+“ v panelu s výběrem create dashboard. Dashboard je přehled, který je složen z několika dotazů.

Na dashboard lze vyskládat předem vytvořené dotazy, či textová pole a to v různých velikostech a poměrech stran. Na dashboard lze přidávat filtry. Filtrům se určí atributy a dotazy, kterým jsou poplatné. Filtry fungují jako restrikce ve WHERE v SQL. Pro více hodnot v jednom filtru je dotaz filtrován s logickým operátorem OR pro tyto hodnoty. V případě více filtrů na dashboardu je mezi jednotlivými filtry logický operátor AND.

Metabase kromě výše uvedených také nabízí například:

- propojení se Slackem
- upozorňování na docílení předem nastavených met přes email
- tisk otázek a dashboardů
- pravidelné zasílání grafů na mail

Pro uživatelskou část aplikace je výborně sepsaná dokumentace, která obšírněji popisuje možnosti metabase (https://www.metabase.com/docs/latest/users-guide/start.html).

Administrátorská část (Admin Guide)

Do administrátorského portálu se lze přepnout po kliknutí na nastavení a zvolení volby „Admin“. V textu budou vysvětleny základní operace jako jsou přidání databáze, správa databázového modelu, správa uživatelů, správa skupin a jejich práv. Nakonec jsou výčtem zmíněny další možnosti programu.

Správa databází je intuitivní, po kliknutí v panelu na „Databases“ se ukáže seznam připojených databází. Databáze lze jednoduše smazat, či po kliknutí na tlačitko „Add database“ a vyplnění odpovídajících políček přidat novou.

Přidanou databázi lze upravit po kliknutí na „Data model“. Je možno zde upravovat:

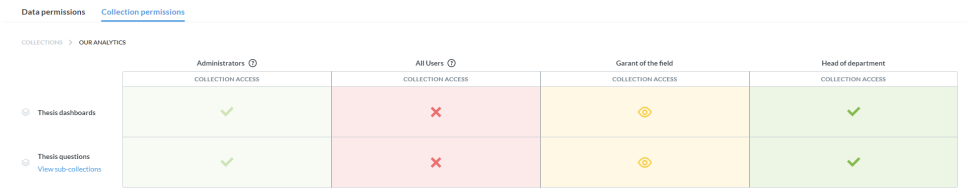
- jednotlivé názvy (tabulek, či sloupců)
- typ sloupce (cizí klíč, jméno entity, číslo, měna, enum atd...), volba typu sloupce ovlivní jeho zobrazování v grafech, či tabulkách
- viditelnosti sloupců (skrýt úplně, viditelné všude, či viditelné pouze v detailu tabulky)

Přidávání uživatelů a skupin je naprosto intuitivní a po kliknutí v panelu na „People“, lze přidávat jednotlivé uživatele a k nim vytvořené skupiny. Zde je důležité zmínit, že při napojení na LDAP se podle propojení dotazu a skupiny v LDAPu po prvním přihlášení uživateli účet v Metabase vytvoří.

Nastavení a správa práv se nastaví po kliknutí v panelu na „Permissions“. Přístupy se nastavují v rámci skupin, pro jednotlivé databáze a kolekce. U databází lze nastavovat přístupy i v rámci tabulek.

Typy přístupů jsou (viz 4.4):

- přístup i s možností editace (zelená fajfka)
- přístup v režimu read-only (žluté oko)
- přístup odepřen (červený křížek)



Obrázek 4.4: Metabase Permissions

Další možná nastavení:

- nastavení integrace se Slackem,
- přihlašování pomocí google účtu,
- přihlašování pomocí LDAP,
- formátování, čísel, času a měn,
- nastavení cachování dat pro složitější dotazy,
- povolení veřejného sdílení dotazů,
- vkládání dotazů do jiné aplikace (embedding),
- Nastavení SMTP emailu, který posílá resetovaná hesla a tak podobně.
- Další konfigurační nastavení (URL stránky, jméno stránky, jazyk, povolení vnořených dotazů atp.).

4.2.3 Další možnosti nástroje

V této sekci jsou detailněji popsány možnosti nástroje s důrazem pro možný budoucí rozvoj.

Lokalizace

Lokalizace Metabase je možná. Oficiální distribuce přidává do svých verzí jazyky pouze za určitých podmínek. Podmínkami jsou 100% pokrytí výrazů jazykem, nenarušování designu (neobsahuje příliš dlouhé překlady), překladatel, firma či osoba, která udržuje překlady akttuální. Lokalizace do ostatních jazyků není snahou core týmu Metabase, ale zainteresovaných osob. Psát překlady může každý. Aktuálně si je možno vybrat ze 14ti jazyků, jedním z nich je i slovenština.

Přizpůsobení prostředí

Pokud je třeba uzpůsobovat prostředí ve větší míře, je záhodno zvážit, zda bylo rozhodnuto pro správný BI nástroj. Při uzpůsobeném kódu aplikace to v případě aktualizace znamená, že se stáhne zdrojový kód z GIT repozitáře Metabase, v případě konfliktů je správce vyřeší, aplikaci zbuilduje a nasadí. Tento postup je tedy o něco komplikovanější než pouhá výměna Jar souborů.

Metabase se skládá ze dvou částí. První je backend, který je psaný v Clojure, a data poskytuje dále pomocí svého REST API, s tím komunikuje část druhá, frontend, která je psána v Javascriptu.

Nejobvyklejším přizpůsobením bude nejspíše přidání, či úprava nějakého typu vizualizace. Momentálně je toto issue v backlogu na GITu Metabase, po jeho splnění by mělo být umožněno lehké přidávání komponent, které bude podporováno týmem Metabase. Aktuální přidání komponenty „na vlastní nebezpečí“ je možné vytvořením komponenty v reactu, přidáním k vizualizacím, přidání cesty do souboru index.js u vizualizací a zbuildováním projektu.

API

Komunikace s REST rozhraním probíhá ve formátu JSON. Při custom dotazech se užívá vlastní jazyk MBQL. [27] Díky tomu Metabase umí s výsledkem následně pracovat (příkladem může být kliknutí na sloupec a tím zobrazení odpovídajících záznamů a tak podobně). Metabase umožuje psát a přes REST posílat i dotazy v nativních jazycích.

```
"type": "query",
        "query": {
            "source-table": 9,
            "filter": [
                "=",
                [
                    "fk->",
                    [
                        "field-id",
                        80
                    ],
                    [
                        "field-id",
                        141
                    ]
                ],
                10011104
            ],
            "aggregation": [
                [
                    "count"
                ]
```

```
        ],
        "breakout": [
            [
                "field-id",
                132
            ]
        ]
    }
```

Výpis 4.1: příklad dotazu v jazyce MBLQL

Aktuální dokumentaci je možné vygenerovat pomocí příkazu *java -jar metabase.jar api-documentation*, ve složce kde je metabase umístěno.

Příkladem jak využít REST API Metabase může být dotaz `/api/card/:id/query/json`, který v JSON formátu vrátí data Metabase dotazu s daným :id. Tyto data lze následně zobrazit pomocí jiného vizuálu, pokud z nějakého důvodu nestačí možnosti Metabase. Dalším příkladem může být generování otázek pomocí skriptu.

Embedding obsahu

Metabase nabízí jednoduchý způsob embeddingu dotazu či dashboardů do jiné aplikace. Nevýhodou open source licence je logo Metabase v levém dolním rohu, které musí být z licenčních důvodů zachováno. Ověření externí aplikace probíhá pomocí vygenerovaného tokenu z atributů dotazu v Metabase a tajného klíče, který zná jak Metabase, tak externí aplikace. Příklad kódu můžeme vidět na obrázku 4.5 v horní polovině se nachází kód pro vygenerování URL adresy požadovaného dotazu v backend aplikaci v jazyce Node.js. Otázku lze vložit i pomocí jiných jazyků, než nabízí přímo metabase, například v Javě.

Při embeddingu u dotazů či dashboardů je nutné určit parametrům jeden ze tří typů:

- Editabled – umožňuje upravovat filtry přímo v zobrazeném dashbordu/-dotazu.
- Disabled – paramemetry se neberou v úvahu.
- Locked – parametry se určují atributem na backendu.

Díky této parametrizaci je umožněno, aby se externí aplikace starala o to, jaký uživatel má práva vidět dotazy a s jakými parametry.

Insert this code snippet in your server code to generate the signed embedding URL

```
// you will need to install via 'npm install jsonwebtoken' or in your package.json

var jwt = require("jsonwebtoken");

var METABASE_SITE_URL = "http://localhost:3000";
var METABASE_SECRET_KEY = "d114d264853c352e8f38d3616d4d496916f4160abbf45cd554e60d825b14b450";

var payload = {
  resource: { dashboard: 33 },
  params: {
    "semester": "B181"
  },
  exp: Math.round(Date.now() / 1000) + (10 * 60) // 10 minute expiration
};
var token = jwt.sign(payload, METABASE_SECRET_KEY);

var iframeUrl = METABASE_SITE_URL + "/embed/dashboard/" + token + "#bordered=true&titled=true";
```

Then insert this code snippet in your HTML template or single page app.

```
<iframe
    src="{{iframeUrl}}"
    frameborder="0"
    width="800"
    height="600"
    allowtransparency
></iframe>
```

Obrázek 4.5: Příklad vložení dashboardu do externí aplikace

V případě použití embeddingu pro více externích aplikací by bylo vhodné udělat aplikaci, která by se v architektuře nacházela mezi Metabase a externí aplikací a fungovala by jako vrstva proxy. Tato aplikace by spravovala práva externích aplikací na dané datasety. Výhodou tohoto přístupu je, že jednotlivé externí aplikace by neznaly tajný klíč a například zákaz zobrazování pro jednu aplikaci by neznamenal regenerování klíče. Aplikace by měla vyžadovat minimální režii.

Další možnosti analýzy dat

Analýza dat pouze pomocí SQL dotazů může být v některých případech omezující. Pro komplexnější výpočty a například větší transformace dat v dotazu jsme nalezli dvě řešení pro další možnosti analýzy dat v Metabase:

- První řešení využívá možnosti psaní procedur v PostgreSQL, ať už „nativním“ jazyce PL/PgSQL, nebo v jazyce Python, který procedury v PostgreSQL také podporují. Pro méně náročné procedury je možné pro tvorbu skriptu použít databázový server a v Metabase volat proceduru, případně i s parametry. Výhodou procedur v Pythonu je i to, že umožňují libovolný import knihoven třetích stran. [26] Spouštění procedur umožňuje okamžitou odpověď s nejaktuálnějšími daty. Nevýhodou je zatěžování databázového serveru, který je hardwarově uzpůsoben spíše pro rychlé odpovědi, než pro vysoký výkon. To lze částečně obejít, pokud se v proceduře pouze zavolá, například přes REST rozhraní, skript, který se

vykoná na aplikačním serveru. Toto řešení lze použít, nicméně nejedná se o příliš elegantní přístup.

- Druhou možností je pravidelné spouštění skriptů, například vždy při aktualizaci datového tržiště, případně při předem určeným spouštěním úloh pomocí plánovače úloh (např. CRONu). Výsledky by se mohli ukládat buď do databáze PostgreSQL, nebo do NoSQL databáze, příkladem může být MongoDB pro kterou má Metabase konektor pro připojení. Dotazy ukládané do NoSQL by umožnily rychlou tvorbu nových dotazů, díky tomu, že není potřeba tvořit pro každý dotaz novou tabulku, a zanechání vysoké flexibility, kdy při změně dotazu není nutné schéma tabulky měnit.

Kapitola 5

Implementace

Implementační část začíná tvorbou datového modelu. Po vytvoření je model naplněn daty pomocí ETL procesů v nástroji Talend. Následně je řešeno vytvoření přehledů a dobře navrženého, udržitelného prostředí v bussiness intelligence nástroji Metabase.

5.1 Tvorba datového modelu

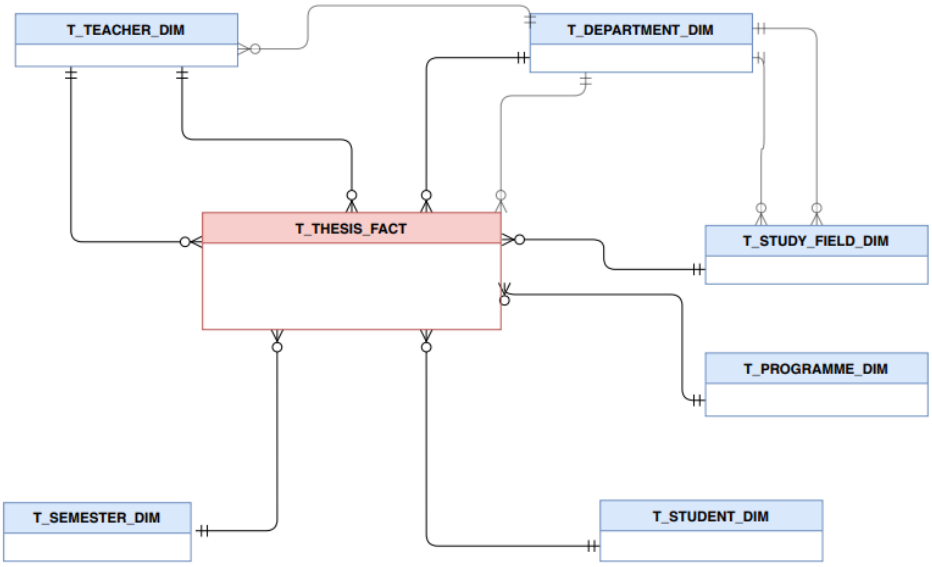
Po analýze dostupných pohledů do systému KOS byly, podle účelu a požadavků na výstupy, určeny jednotlivé entity a k nim relevantní atributy. Z těchto entit bylo vytvořeno schéma datového tržiště. Tržiště bylo zpočátku navrhováno ve schématu vločky (viz 2.4.1), iterativně se pomocí denormalizace databáze dospělo k jednoduššímu modelu ve schématu hvězdy (viz 2.4.1). Model byl tvořen v programu Visual paradigm, který umožňuje z modelu vygenerovat skripty pro CREATE a DROP databáze. Za pomoci těchto skriptů byla vygenerována databáze v PostgreSQL. Nejaktuálnější CREATE a DROP skripty jsou současně uložené v projektu mimo model, kvůli použitým View a Triggerům, které v modelu nejsou zahrnuty.

5.1.1 Popis datového modelu

Datový model je tvořen dimenzemi okolo jedné tabulky faktů (T_THESIS_FACT). Ostatní entity jsou popisné svým názvem (viz 5.1).

Zvolená jmenná konvence databázového modelu, se drží konvencí vytvořených pro datový sklad ČVUT podle Diplomové práce Ing. Jakuba Krejčího [18] (konvence je v citované práci popsána v příloze B). Výhodou konkrétní použité konvence je zanechání zdrojového názvu tabulky a atributu u technických klíčů. Konvence slouží pro systematický způsob pojmenovávání, které má za následek jednodušší porozumění databázovému modelu.

Na obrázku 5.1 je zjednodušený databázový model tabulek, bez atributů. Kompletní databázový model je obsažen v příloze D.



Obrázek 5.1: Databázový model tabulek a relací datového tržiště

5.2 Tvorba ETL tabulek

Pro přehled nad vazbami mezi sloupcemi cílové a zdrojové databáze, se jako nejlepší volba ukázala tabulka v excelu (viz přiložené CD soubor `ETLTables.xls`).

Každá tabulka obsahuje sloupce:

- název atributu (udává sloupec v cílové DB),
- název tabulky zdroje,
- název atributu zdroje (udává sloupec ve zdrojové DB).

Tabulka:	T_STUDENT_DIM	
Název atributu	Název tabulky zdroje	Název atributu zdroje
ID		
VFF_OSOBY_OSOBNI_CISLO_BK	vff_osoby	osobni_cislo
VFF_OSOBY_PERIDNO_TK	vff_osoby	peridno
USERNAME	vff_osoby	username
NAME	vff_osoby	jmeno
SURNAME	vff_osoby	prijmeni
TITLE_BEFORE	vff_osoby	titul
TITLE_AFTER	vff_osoby	titul_za
SEX	vff_osoby	pohlavi
BIRTHDATE	vff_osoby	datum_nar
EMAIL	vfstudent	email
HIGH_SCHOOL_IZOCODE	vfstudent	odkud_skola_kod
MATURITY_YEAR	vfstudent	rmat
NATIONALITY	vfstudent	narodnost

Obrázek 5.2: Příklad ETL tabulky T_STUDENT_DIM

Při větším počtu zdrojů lze přidat sloupec Zdroj. Tento sloupec zatím nebyl potřeba.

5.3 Tvorba ETL procesů

Pro naplnění databáze daty je použit již výše zmíněný program Talend Open Studio for Data Integration. V této sekci jsou užívány pojmy, které byly vysvětleny během seznámení s programem Talend (viz 4.1).

5.3.1 Popis hlavního ETL procesu

Hlavní ETL proces je složen z několika dílčich částí (části jsou červeně vyznačeny na obrázku 5.3):

1. Před spuštěním samotného Jobu. V této části je přesměrováno logování z konzole do souboru.

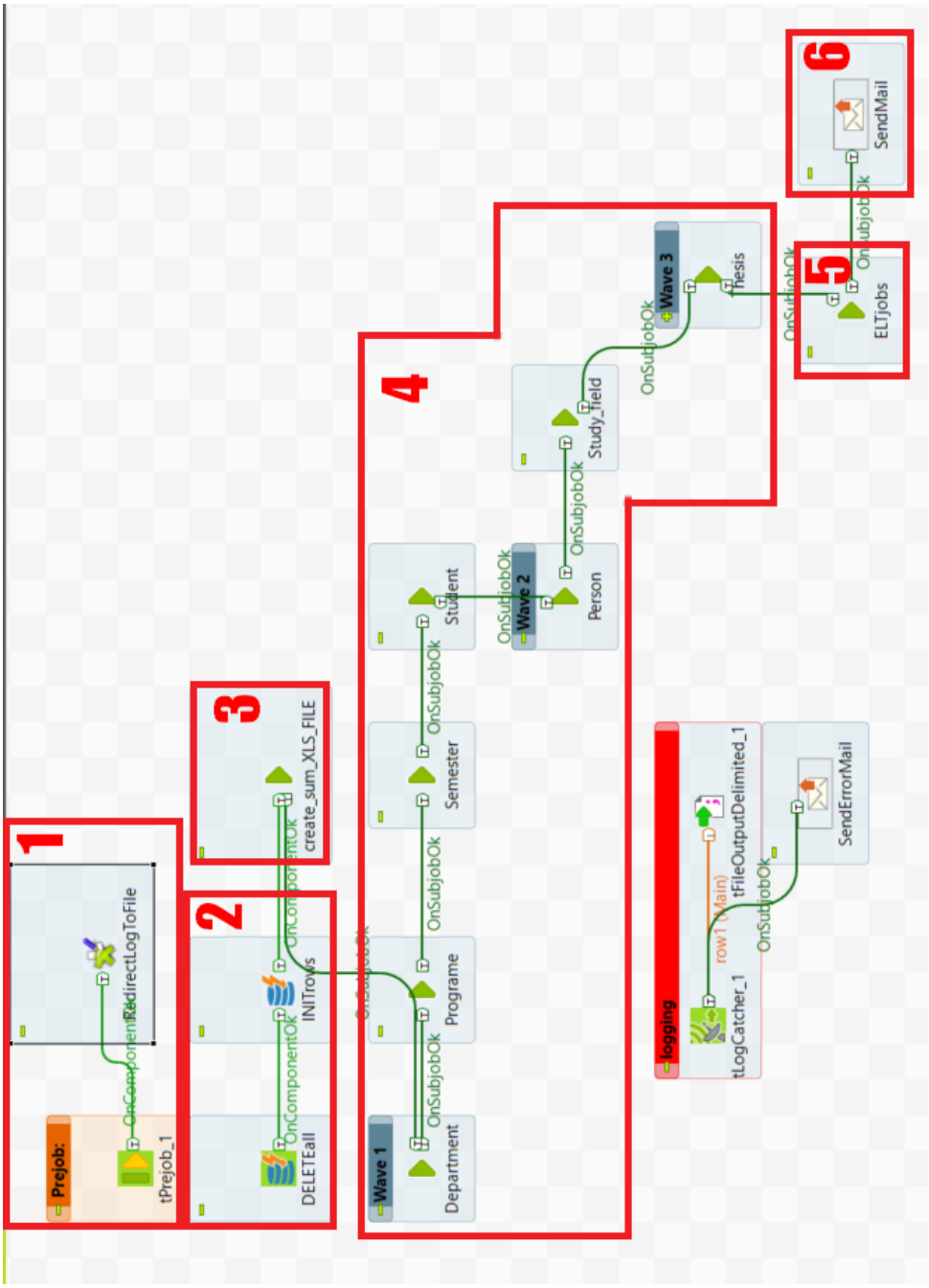
2. Vymazání řádků z databáze. Jelikož se aktualizace bude dělat jednou za delší časový úsek, přibližně půl roku, data nejsou neaktualizována, ale pokaždé se všechny záznamy smažou a znovu nahrají.

3. Vytvoření XLS souboru se všemi známkami posudků od vedoucích a oponentů. Jelikož tyto známky nejsou obsaženy v pohledech do databáze, je třeba je načítat z jiného zdroje. Tímto zdrojem jsou poskytnuté XLS soubory.

4. Plnění databáze probíhá ve „vlnách“ podle závislosti referenční integrity v databázi. První je třeba postupně naplnit tabulky dimenzí a až poté tabulku faktů. Na obrázku (viz 5.3) je vidět pořadí spouštění ETL procesů, pro naplnění jednotlivých tabulek.

5. ELT job. Job, který maže záznamy na které není nikde odkazováno. Semestry, které nejsou nikde použity. učitelé, kteří nevedli, ani neoponovali žádnou práci a tak podobně.

6. Zaslání emailu o průběhu procesu s cestou k Logovým souborům.



Obrázek 5.3: Hlavní ETL proces

5.3.2 Řešení nekonzistence dat

Během plnění dat bylo objeveno několik problémů s konzistencí dat. Nekonzistence v rámci cizích klíčů byla řešena jednou z níže popsaných tří možností, z kterých bylo vybráno podle smyslu dat:

- Povolení hodnoty null u cizího klíče.

- Vytvoření záznamu v tabulce, kam má cizí klíč odkazovat, jenž má v popisu uvedeno, že ve zdrojové databázi cizí klíč nebyl nalezen.
- Smazání záznamu, pokud cizí klíč neexistoval například jen u jednoho záznamu. Záznam často nedává vůbec smysl a místo změny schématu databáze se záznam maže.

Nekonzistence v rámci datových typů zdrojového a cílového schématu se řeší převážně v komponentě `tMap`. Pro jednoduché situace ternárním operátorem, pro komplikovanější případy napsání java funkce v „Routines“.

5.3.3 Popis jednotlivých vln

Plnění databáze probíhá ve „vlnách“, podle závislosti referenční integrity v databázi. Proces se obecně skládá z výběru ze zdrojové databáze, přemapování na schéma cílové databáze a vložení záznamů do cílové databáze. V této sekci tomuto procesu budeme říkat základní ETL proces.

5.3.4 Vlna 1

V této vlně jsou tabulky, které v sobě neobsahují žádný cizí klíč.

Konkrétně se jedná o tabulky:

- T_DEPARTMENT_DIM
- T_PROGRAME_DIM
- T_STUDENT_DIM
- T_SEMESTER_DIM

Převod dat do těchto tabulek je poměrně jednoduchý. U většiny tabulek ve vlně 1 jde o základní ETL proces. Situace je odlišná pouze u tabulky T_SEMESTER_DIM, kde se před uložením záznamů do databáze profiltrují semestry. Odfiltrované jsou speciální semestry pro uznané předměty, semestr pro doktorské studium a podobně.

5.3.5 Vlna 2

Zde jsou tabulky, které v sobě obsahují cizí klíč z tabulek vlny 1.

U vlny 2 se konkrétně jedná o tabulky:

- T_TEACHER_DIM
- T_STUDY_FIELD_DIM

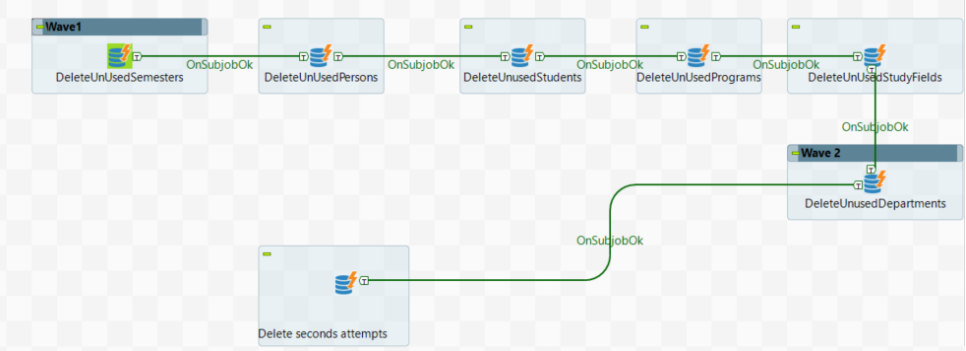
Skládají se opět pouze ze základního ETL procesu. V této sekci se narazilo na nekonzistenci cizích klíčů, kdy u T_STUDY_FIELD_DIM bylo během mapování třeba povolit hodnotu null pro atribut katedry, kam cizí klíč ukazoval na záznam 0, který v katedrách neexistuje. Obdobná změna byla třeba i u cizích klíčů T_TEACHER_DIM odkazujících na T_DEPARTMENT_DIM.

5.3.6 Vlna 3 (T_THESIS_FACT)

Ve vlně tři se plní tabulka faktů. Jedná se o nejkomplexnější převod. Při převodu je zapotřebí kontrola existence vedoucích a oponentů v databázi a odfiltrování záznamů u kterých osoby v databázi neexistují. Výpočet posledního akceptovatelného datumu (5 let dozadu). Načtení semestrů, podle jejichž datumu začátku a konce se určí semestr SZZ. Namapování známek posudků z excelu ke správnému záznamu SZZ, či výpočet a namapování váženého průměru studenta k záznamu o SZZ.

5.3.7 ELT Jobs

Zde probíhá mazání záznamů dimenzí. Jde o záznamy, které nikam neodkazují. To znamená, že jejich Id není použito jako Id cizího klíče u žádného záznamu. Mazání probíhá opět ve vlnách určené podle referenční integrity. V první vlně jsou smazány záznamy, na něž není odkazováno, všech tabulek dimenzí, kromě T_DEPARTMENT_DIM. Ve druhé vlně jsou smazány záznamy v tabulce T_DEPARTMENT_DIM, jelikož jejich cizí klíč je obsažen i v jiných tabulkách dimenzí. Mimo vlny je dotaz, který maže druhé pokusy závěrečných zkoušek, pokud nebyly tyto pokusy způsobeny závěrečnou prací, ale známkou z předmětu při prvním pokusu závěrečné zkoušky.



Obrázek 5.4: ELT job

Příklady SQL ELT dotazů

```
-- smazání nepoužitých Study fields
DELETE FROM t_study_field_dim WHERE vff_toboryst_id_tk IN
(SELECT vff_toboryst_id_tk FROM t_study_field_dim
EXCEPT
 (SELECT t_thesis_fact.fk_study_field_id_tk
  FROM t_thesis_fact ));

-- smazání druhého pokusu SZZ,
      pokud na prvním byla závěrečná práce úspěšně obhájena
DELETE FROM t_thesis_fact
WHERE attempt_number = 2 AND vff_tstudenti_id_tk NOT IN (
```

```
SELECT vff_tstudenti_id_tk
FROM t_thesis_fact
WHERE attempt_number = 1 AND thesis_defense_grade = 'F'
);
```

5.4 Vytvoření business inteligence prostředí

Jako BI nástroj je použit již výše zmíněný program Metabase. V této sekci jsou užívány pojmy a možnosti nástroje, které byly vysvětleny během seznámení s programem Metabase (viz 4.2.2).

5.4.1 Přizpůsobení prostředí

Po instalaci prostředí a přidání připojení databáze datového tržiště, bylo primárním úkolem přizpůsobení aplikace pro potřeby projektu a vytvoření uchopitelného uživatelského prostředí. Prostředí samo o sobě je intuitivní, ale obsah v něm jednoduše a rychle bobtná. Bez předem určených pravidel by obsah rychle způsobil nepřehlednost a zmatek. Přizpůsobení prostředí se podařilo doladit, díky několika iteračním krokům. Přizpůsobení poskytla větší uživatelskou přívetivost.

Přejmenování tabulek a sloupců

Pro usnadnění přehledu v Metabase byly přejmenovány tabulky a sloupce. Názvy se nyní nedrží databázové jmenné konvence, ale jsou upraveny, aby byli maximálně popisné. S touto úpravou souvisí i změna viditelností technických sloupců a nastavení typů sloupců a jejich formátování při zobrazení. U změněných atributů je v popisu uveden jejich původní název. Pro případ, že by uživatelé chtěli například psát nativní SQL dotazy.

Struktura kolekcí

U struktury kolekcí nastal problém s přepoužitelností dotazů. Smysl dotazu bývá často definován spíše filtry na dashboardu, než dotazem samotným. Zároveň Metabase umožňuje vložit dotaz pouze do jedné kolekce. Bez dobré struktury kolekcí by hrozila nepřehlednost nebo duplikace dotazů.

V kořenové kolekci jsou dvě základní kolekce, Thesis questions a Thesis Dashboards.

Thesis questions se dělí na:

- Basic questions obsahuje základní dotazy.
- Complex questions obsahuje náročnější dotazy (například vnořené dotazy psané v SQL).

- Grouped by questions – dotazy, které jsou podle něčeho seskupeny. Pokud je mnoho dotazů seskupeno jednou metrikou, vnoříme do této kolekce další (např. „Grouped by supervisor“).
- Temporary questions – pomocné dotazy, tato kolekce se uživalům nezobrazuje.

Pokud je otázka odvozena od jiné, to jest liší se pouze restrikcí, poté je tato otázka přidána do subkolekce „Derived questions“. Tato subkolekce se nachází v kolekci, kde je uložen původní dotaz.

Thesis dashboards je, jak název napovídá, kolekce pro dashboardy.

Tato struktura dává otázce jednoznačné místo a díky tomu lze otázky nalézt a přepoužívat.

Jmenná konvence dotazů

Nevhodná jmenná konvence u dotazů má podobné vedlejší efekty, jako nevhodná struktura kolekcí. Hrozí nepřehlednost, název nic neříkající o obsahu dotazu způsobí špatnou dohledatelnost dotazu a z toho vyplývající tvorbu duplicitních dotazů. Je-li dotaz odvozen, poté platí rozšířená konvence.

- Konvence: sumarizace a smysl dotazu (group by), dataset
  Jinak řečeno: co dotaz znázorňuje (čím je shlukováno), dataset
  Příklad: Average of Defense grade (Department, Semester), Thesis
- Rozšířená konvence: sumarizace a smysl dotazu (group by)[restrikce], dataset

  Jinak řečeno: pokud je dotaz odvozen a je v něm filter, poté za závorku píšeme do hranatých závorek podmínku.

  Např: Average of Defense grade (Department, Semester) [Defense grade = A] , Thesis

Dataset pro „THESIS_MARKET“ se označuje pro zkrácení „Thesis“.

Jmenná konvence je určena pro snadnější vyhledávání a udržitelnou přehlednost i přes rostoucí počet dotazů.

Úvodní stránka

Pro lepší přehlednost byly na hlavní obrazovku („obrazovku, na kterou uživatel vstoupí po přihlášení“) přidány dva dashboardy.

První dashboard slouží jako rozcestník, ve kterém lze nalézt odkazy na dashboardy, kde je největší agregace (top level dashboardy). Z těchto dashboardů se lze cestou proklikat k většímu detailu záznamů.

Druhý dashboard je rychlým úvodem do prostředí. Tento dashboard je určený pro nové uživatele, obsahuje základní orientaci v prostředí, představuje možnosti Metabase a vysvětluje základní strukturu projektu.

5.4.2 Tvorba analytického obsahu

Na výše popsaných základech byla založena tvorba samotného obsahu. Obsah je tvořen dvěma hlavními částmi. Hlavními částmi jsou dotazy a dashboardy, kde jsou jednotlivé dotazy seskupeny.

Tvorba Dotazů

Při tvorbě dotazů se osvědčilo denormalizované a tím pádem i co do počtu tabulek menší schéma datového modelu, jelikož Metabase nabízí do filtrů kromě atributů vlastní tabulky i atributy z tabulek sousedních. To při použitém schématu znamená, že z tabulky faktů dotaz „dosáhne“ na libovolnou tabulku, bez nutnosti použití JOINů během psaní dotazu, což umožňuje výbornou přepoužitelnost dotazů na dashboardech.

Jednou z hlavních výhod Metabase je intuitivita vytváření dotazů. Nicméně daní za tuto vlastnost je předem určené chování, které ne vždy vyhovuje našemu záměru.

Hlavní problémy, na které bylo v rámci tvorby dotazů nutné reagovat:

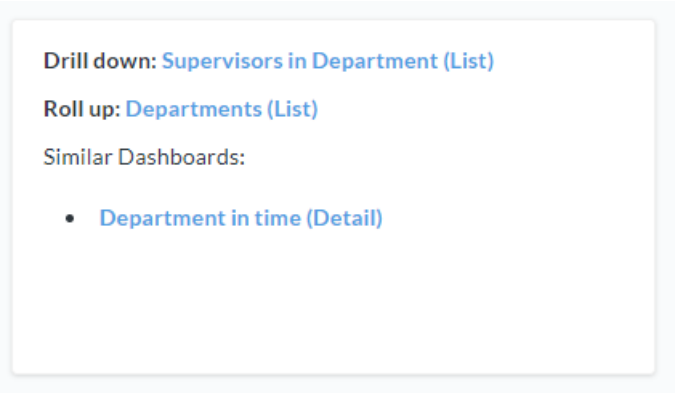
- Stejné názvy u sloupcového grafu na ose X se sumarizují. Problém byl Vyřešen přidáním sloupce, který je tvořen příjmením a identifikátorem v závorce.
- Zobrazování hodnoty libovolného atributu místo id cizího klíče. Nefunguje pro 2 sloupce ukazující na stejnou tabulku. Vyřešeno vytvořením View pro druhou hodnotu, hodnota je v Metabase napojená cizím klíčem na toto View, nikoli na původní tabulku. Poté vše funguje korektně.
- U sloupcového grafu, kde máme odděleny jednotné známky a tyto známky jsou v grafu seskupeny na sobě (například podle kateder), nelze řadit podle celkové velikosti sloupce. Očekává se, že problém bude v nejbližší době vyřešen.
- Metabase umožňuje namapovat na čísla libovolné texty, které následně v grafech či tabulkách zobrazuje místo původního čísla. Tuto vlastnost bylo v úmyslu použít pro známky, nicméně pokud se udělal například průměr, neukazovalo se mapování, ale číslo. Toto chování je logické, nicméně, důsledkem toho je zobrazování osy Y u známek ve formátu čísel, nikoli písmen.

Tvorba Dashboardů

Při tvorbě dahboardů bylo cílem udržet celkovou konzistenci vzhledu mezi dashboardy, aby dashboardy byly pro uživatele rychle přehledné.

Problémem Metabase je, že ve verzi zdarma neobsahuje rozhraní pro pohyb skrz dashboardy. Tato funkcionalita je pro BI systém důležitá a v projektu byl tento pohyb řešen pomocí odkazů v menu dashboardu (viz obr. 5.5).

- Drill down – sekce obsahující odkazy na dashboardy s větším detailem záznamů.
- Roll up – sekce obsahující odkazy na dashboardy s více agregovanými dotazy.
- Similar dashboards – sekce obsahující odkazy na dashboardy podobné právě otevřenému dashboardu.



Obrázek 5.5: Metabase menu dashboardu Department(Detail)

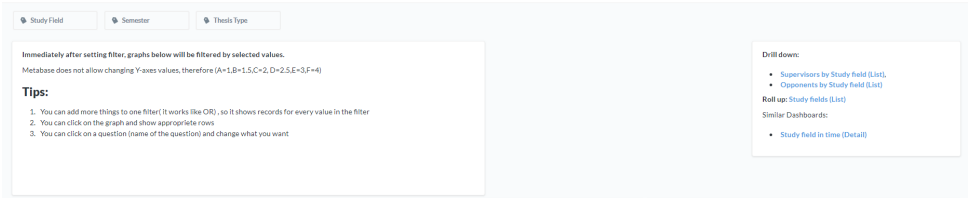
Dalším možným přístupem je pomocí SQL příkazu, například funkce `concat`, poskládat odkaz a ten následně zobrazit v tabulce. Metabase dokonce umožňuje místo samotného odkazu zobrazit v tabulce libovolný text, který je ale stále odkazem a po kliknutí na něj se otevře původní odkaz.

Objevené problémy spjaté s dashboardy:

- Do filtrů nelze napsat SQL dotaz, zobrazují se tedy všechny hodnoty zvoleného atributu v tabulce. Toto chování lze obejít pomocí vytvoření View, které zobrazuje daný dotaz a s kterým již Metabase umí pracovat.
- Dotaz psaný nativním SQL nelze použít na dashboardu s filtrem. Řešením by bylo složitější dotazy dělat přes View.

Struktura Dashboardů

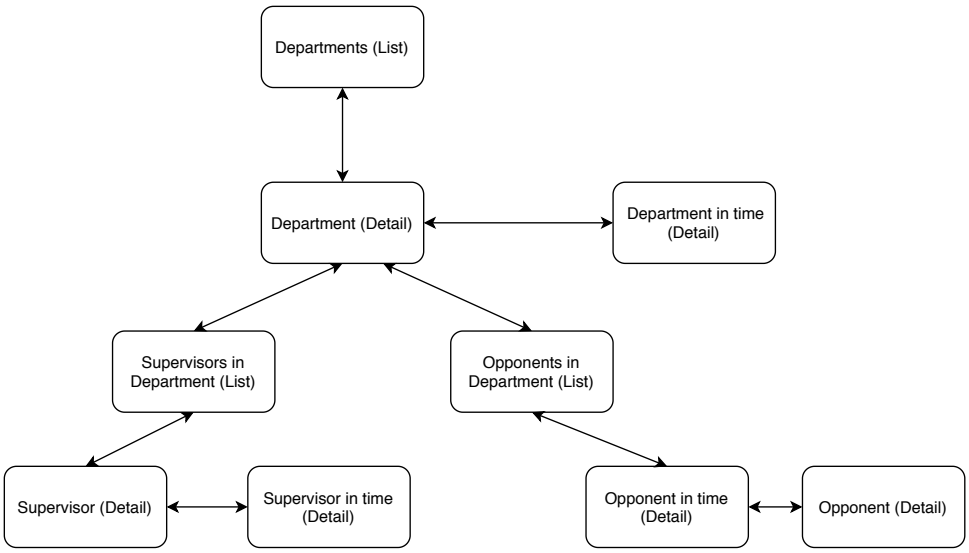
Dashboardy mají jednotnou strukturu. Každý dashboard obsahuje v záhlaví filtry, které lze nastavit. Pod filtry jsou nalevo vysvětlení a Tipy, napravo se nachází menu pro pohyb po dashboardech (viz obr. 5.6). Následuje samotný obsah dashboardu (již není součástí obrázku). Text s nadpisem „Interesting questions concerning this Topic:“ odděluje dashboard a zmenšené otázky, které by mohly určité uživatele zaujmout, ale na dashboardu samotném by zabíraly místo.



Obrázek 5.6: Metabase dashboard záhlaví Study field (Detail)

V projektu jsou dashboardy dvou typů `list`, nebo `detail`. Na dashboardu typu List se porovnávají jednotlivé entity náležící stupni agregace (například známky kateder). Dashboardy typu Detail ukazují výsledky právě jedné této entity. Dashboardy Detailu mají vždy stejnou strukturu a liší se pouze použitými filtry. Příklad dashboardu detailu lze najít v příloze E. Dashboardy Listu se otázkami liší. Jejich odlišení je nutné především kvůli tomu, že otázky listu jsou seskupované (GROUP BY) vždy jiným atributem.

Takto vytvořené dashboardy jsou pospojované do logických cest, kterými se lze pohybovat. Aktuálně naimplementované cesty jsou z pohledu kateder a studijních oborů.



Obrázek 5.7: Mapa cesty dashboardů z pohledu kateder

5.4.3 Nastavení skupin a práv

V aplikaci jsou vytvořené tři skupiny uživatelů. Jedná se o administrátory, garanty programů a vedoucí kateder. V aplikaci je nastavené připojení na LDAP, oproti kterému se budou uživatelé a jejich přihlašovací údaje ověřovat. Práva garantů programů a vedoucích kateder jsou aktuálně nastavena tak, že umožňují zobrazení všech kolekcí a libovolný přístup k datům. Uživatelé mohou tvořit ve vlastní kolekci, přičemž na požádání budou jimi vytvořené dashboardy přesunuty do společné kolekce.

5.5 Podpora v produkci

Podpora produkčního běhu je konkrétně popsána v administrátorské příručce na Gitlabu. Obecně se jedná o aktualizace dat, spouštění ETL procesů a povyšování Metabase na nové verze.

Kapitola 6

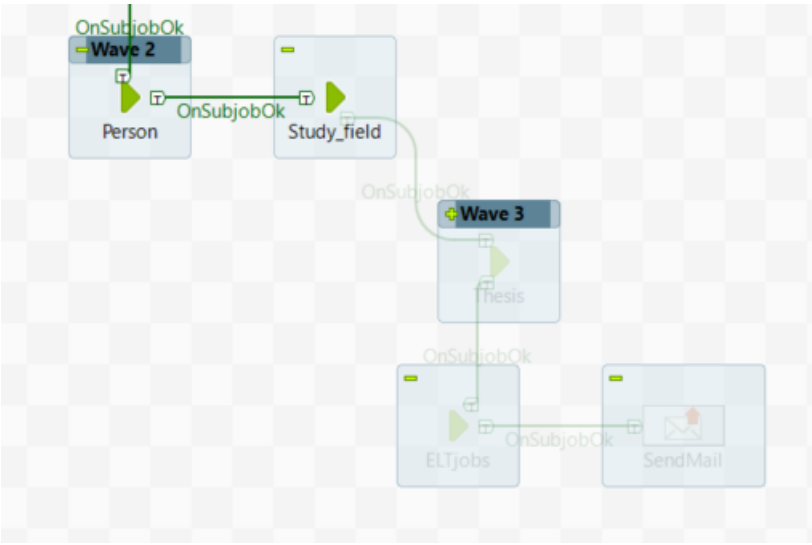
Vyhodnocení užitých nástrojů

6.1 Talend Open Studio for Data Integration

V této kapitole je popsáno zhodnocení aplikace Talend Open Studio for Data Integration po fázi ETL převodu dat. Jsou zde popsány uživatelská přívětivost, výhody a problémy, na které se během tvorby převodu dat narazilo.

6.1.1 Uživatelská přívětivost

Po základním seznámení (viz 4.1) se nástroj stává přehledným. Pokud jsou psány procesy přehledně, lze se rychle v diagramu zorientovat a pochopit postup transformace dat procesem. Uživatelské přívětivosti přidává mód pro ladění procesu („debug mód“), kde lze přepínat mezi Designerem a vygenerovaným kódem programu. Převážně během debugování, ale nejen tehdy, vývojář ocení možnost deaktivování komponenty (viz obr. 6.1). Tím například převod dat dostane do fáze, které požaduje a až poté zkouší změny procesu na dané, problematické části.



Obrázek 6.1: Deaktivované komponenty, příklad

6.1.2 Výhody

Mezi výhody lze řadit přehlednost, jednoduchou rozšiřitelnost, díky možnostem dopsání java kódu. Ať už se jedná o kód psaný do tříd a metod pomocí routines, nebo pomocí komponenty `tJava`. Pro rozumně fungující ETL nástroj je naprosto zásadním prvkem dobře zvládnutý debug mód. Debugování v Talendu může mít několik podob, které lze volit podle situace. V procesu lze při nezdařilém pokusu vidět výpisy z konzole, debugovat si proces řádku po řádce, či přepnout do režimu Java kódu a debugovat pomocí breakpointů. U žádného z režimů nebyli pozorovány zvláštnosti v chování a fungují, jak je očekáváno. Další výhodou jsou dostupné komponenty, které jsou předem hotové pro daný problém. Výhodou je i jejich počet. Nejasnosti a problémy jsou snadno dohledatelné na fóru, kde operuje aktivní komunita. Centralizované údaje pro připojení ke zdrojům, například databázi, a kontextové proměnné umožňují rychlé změny.

6.1.3 Nevýhody a problémy

Po změně schématu cílové databáze je potřeba pro změnu schématu v procesu udělat několik za sebou jdoucích kroků, což trvá déle, než by bylo v ideálním případě třeba. Proměnné, které se využívají napříč procesem, ale nejsou součástí schématu řádku se ukládají do globální mapy, kde klíčovým atributem je zadaný název a hodnotou vložená proměnná. Toto řešení se zdálo nepříliš elegantním. Vytváření diagramu a skládání správných komponent v odpovídajícím pořadí si žádá určitý cvik. Během projektu nebyly u převodu dat řešeny principiálně delikátní a neobvyklé záležitosti, i to může být důvodem, proč se neobjevil větší počet nevýhod programu.

6.1.4 Zhodnocení

Pro používání nástroje Talend je potřeba prvotní časová investice pro pochopení principů, na kterých je postaven, poznání základních komponent, které nabízí a best practices, které usnadní práci a zminimalizují duplikace. Nástroj poskytuje dopsání libovolného Java kódu, jednotlivých tříd a funkcí. Toto v kombinaci s možným přidáním libovolné komponenty ze široké nabídky a dobře zpracovaným debug módem tvoří flexibilní a překvapivě pružný „klikací“ nástroj, který umožňuje rychlou a jednoduchou tvorbu ETL procesů.

Nástroj sám o sobě funguje adekvátně k požadavkům, je ale na zvážení, zda je tým, který má s nástrojem pracovat, nakloněn učit se tento nástroj používat. Velké procento programátorů preferuje čisté psaní kódu bez grafického rozhraní, pro tuto skupinu lze zmínit jiné možnosti ETL převodu dat, jako variantu lze uvést například psaní ETL pipelines v Apache SPARK. [23] Před začátkem projektu je na zvážení, jakou z těchto cest se vydat.

6.2 Metabase

V této kapitole jsou popsány výhody, nevýhody a celkové zhodnocení nástroje po fázi implementace.

6.2.1 Výhody

Výhody nástroje Metabase se prolínají textem (viz 3.3.2 a 4.2.2). Největší výhody jsou jednoduchá tvorba dotazů, i bez znalosti jazyka SQL, tvorba dashboardů. Uživatelské prostředí, které je rychle pochopitelné. Připojení k LDAPu a správa uživatelů v rámci skupin umožňuje jednoduše určovat přístupy. Během tvorby se jako velká výhoda ukázala aktivní komunita a zároveň aktivní vývoj, který rozšiřuje možnosti nástroje.

6.2.2 Nevýhody

Během vývoje bylo zaznamenáno několik nevýhod aplikace. Nemožnost vytvoření proměnných, například pro linky, na které se odkazuji na dashboardu, či skriptů, které by se daly volat na stisknutí tlačítka. Druhou zásadnější nevýhodou je nepodporovaná změna agregace dotazu, po kliknutí například na daný sloupec, nicméně Metabase nabízí alternativy, jak tohoto chování docílit (viz 5.4.2).

Metabase neumožňuje dotaz filtrovat podle přihlášeného uživatele, je to dáno tím, že Metabase je nástroj určen pro analýzu a nikoliv pro reporting všem skupinám uživatelů s právy na úrovni atributů, nikoliv tabulek. Této funkcionality lze docílit tvorbou více dotazů, případně dotazem s parametrem, který se zobrazuje mimo aplikaci. Parametr poté určuje aplikace externí.

V neposlední řadě nevýhodou může být paradoxně i rychlá tvorba dotazů, nedodržuje-li se jmenná koncence a struktura kolekcí, může být aplikace rychle nepřehledná. Kromě výše sepsaných se objevily i určité mírně omezující nevýhody během tvorby dotazů (viz 5.4.2) a tvorby Dashboardů (viz 5.4.2). Problémy nebyly zásadnějšího rázu a není třeba je zde znovu popisovat.

6.2.3 Zhodnocení

Během tvorby projektu se nenalezla chyba, která by zásadnějším způsobem ovlivnila celkové řešení a nebyla by možná obejít. Metabase nabízí jednoduchou a rychlou tvorbu obsahu s možností snadného embeddingu do dalších aplikací. Pro udržitelný přehled nad dotazy je třeba používat jmennou konvenci dotazů a mít pevně danou strukturu kolekcí. Metabase nabízí velký počet driverů pro připojení k různým typům databází a je rychle se rozvíjejícím projektem s aktivní komunitou. Přibližně jednou za čtvrt roku je vydána nová verze s rozšířenou funkcionalitou. Na nové verze není ideální přecházet okamžitě, ale počkat si na patch, jelikož je u nové verze větší pravděpodobnost, že bude obsahovat nějakou chybu. Jako celek funguje Metabase korektně a nástroj splnil naše požadavky a očekávání.

Kapitola 7

Závěr

Cílem práce bylo vytvořit systém pro analýzu výsledků závěrečných prací. V rámci zpracované teorie byly definovány základní pojmy a koncepty business inteligence a analýzy dat. Pro účely návrhu systému byl následně zjištěn aktuální stav a specifikovány požadavky na výsledný systém. Určené požadavky zahrnovaly, aby užívaný software byl zdarma, dostupnost z webového prohlížeče, reporting pro uživatele, který bude poskytnut pomocí interaktivních dashboardů a možnost provádění analýzy dat uživatelem. Na základě specifikovaných požadavků byla analyzována možná řešení. Po bližší analýze byla, jako nástroj pro analýzu dat, zvolena aplikace Metabase.

Podle zvoleného nástroje pro analýzu dat byl definován postup, který je složen z převodu dat do vlastní databáze pomocí nástroje Talend Open Studio for Data Integration, převedená data se ukládádají do PostgreSQL databáze, ze které Metabase zobrazuje data. Po zvolení postupu následovalo seznámení se zvolenými nástroji a tím i ověření jejich vhodnosti pro daný úkol.

Pro potřeby výsledného systému byla zjištěna dostupná data. Na základě těchto dat byla navržena databáze. Databáze byla posléze naplněna daty pomocí ETL procesu v aplikaci Talend.

Po naplnění databáze bylo možné databázi napojit na aplikaci Metabase a v ní vytvořit požadované dotazy na data a výsledky zobrazit pomocí přehledných vizualizací. Pro zobrazení dat byly vytvořeny dashboardy, které umožňují pohledy na data z různých pohledů, dashboardy jsou mezi sebou logicky pospojovány pomocí odkazů. Pro vytvoření intuitivního uživatelského prostředí, byla určena struktura kolekcí, jmenná konvence dotazů a nastavena možnost přihlášení uživatelů přes LDAP.

Po dokončení implementace byly popsány možnosti Metabase pro potenciální rozšíření systému o další funkcionality nebo použití stávajícího systému na projekty obdobného charakteru. Jednotlivé nástroje byly zhodnoceny a celkový systém byl nasazen do pilotního provozu na FEL ČVUT, předem stanovené cíle byly splněny.

Příloha A

Literatura

[1] LABERGE, Robert. *Datové sklady: agilní metody a business intelligence*. 25.7.2012. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1.

[2] POUR, Jan, Miloš MARYŠKA, Iva STANOVSKÁ a Zuzana ŠEDIVÁ. *Self service business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace*. 2018. Praha: Grada Publishing, 2018. Management v informační společnosti. ISBN 978-802-7106-165.

[3] MOLNÁR, Zdeněk. *Podnikové informační systémy*. Vyd. 2. přeprac. V Praze: Česká technika - nakladatelství ČVUT, 2009. Management v informační společnosti. ISBN 978-80-01-04380-6.

[4] MARTYN, Tim. Reconsidering Multi-Dimensional Schemas. *SIGMOD Record* [online]. 2004, **33**(1), 83-87 [cit. 2020-04-30]. Dostupné z: `http://sigmodrecord.org/publications/sigmodRecord/0403/B6.Martyn_6page.pdf`

[5] XIA, Belle a Peng GONG. Review of business intelligence through data analysis. *Benchmarking: An International Journal*. 2014, , 300—311.

[6] KOTLÁŘ, Robert. *Datový sklad ČVUT - způsoby datové integrace* [online]. Praha, 2017 [cit. 2020-04-30]. Dostupné z: https://dspace.cvut.cz/bitstream/handle/10467/70140/F8-DP-2017-Kotlar-Robert-thesis.pdf. Diplomová práce. České vysoké učení technické v Praze. Vedoucí práce Ing. Stanislav Kuznetsov.

[7] ABRAMSON, Ian. *Data Warehouse: The Choice of Inmon versus Kimball* [online]. In: . [cit. 2020-04-30]. Dostupné z: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.715.9705&rep=rep1&type=pdf`

[8] NÁPLAVA, Pavel. *B6B16INS 3.přednáška*, 56-64 [online]. [cit. 2020-04-30]. Dostupné z: `https://moodle.fel.cvut.cz/pluginfile.php/184378/mod_page/content/18/Prednaska01.pdf`

[9] SUN, Junping. *Relational Database Theory - Normalization* [online]. In: . s. 16-23 [cit. 2020-04-30]. Dostupné z: http://scis.nova.edu/ jps/teaching/phdiss/diss02s/diss750/notes/diss02-5.pdf

[10] HOWSON, Cindi, James RICHARDSON, Rita SALLAM a Austin KRONZ. *Magic Quadrant for Analytics and Business Intelligence* [online]. In: . s. 6 [cit. 2020-04-30]. Dostupné z: https://cadran-analytics.nl/wp-content/uploads/2019/02/2019-Gartner-Magic-Quadrant-for-Analytics-and-Business-Intelligence-Platforms.pdf

[11] 77 open source free and top dashboard software. *Predictiveanalyticstoday* [online]. [cit. 2020-04-30]. Dostupné z: https://www.predictiveanalyticstoday.com/open-source-dashboard-software/

[12] PSISTAKI, Giorgos. *The Ultimate List of Business Intelligence Tools and Dashboards* [online]. 2019 [cit. 2020-04-30]. Dostupné z: https://www.blendo.co/blog/the-ultimate-list-of-custom-dashboards-and-bi-tools-to-track-your-metrics-and-gather-insights/

[13] BOBRIAKOV, Igor. *11 BI Tools Pricing and Capabilities — Analysis of Legacy Modern Business Intelligence Tools* [online]. Published 15.7.2019 [cit. 2020-04-30]. Dostupné z: https://www.intermix.io/blog/bi-tools-2020-compared/

[14] OSWAL, Nilam. *10 Best Free And Open Source BI Tools* [online]. Published 4.11.2019 [cit. 2020-04-30]. Dostupné z: https://www.softwaresuggest.com/blog/best-free-open-source-bi-tools/

[15] *TESTED: SIX SOFTWARE TOOLS FOR CONSTRUCTING A DATA PLATFORM* [online]. Published 26.11.2018 [cit. 2020-04-30]. Dostupné z: https://drive.tmaas.eu/2018/11/26/six-software-tools-for-constructing-a-data-platform/

[16] *2019 Gartner Magic Quadrant for Data Integration Tools* [online]. Published 1.8.2019 [cit. 2020-04-30]. Dostupné z: https://www.talend.com/resources/2019-gartner-magic-quadrant-data-integration-tools/

[17] ZAPLETAL, Lukáš. *Lehký úvod do LDAP* [online]. Published 24.7.2010 [cit. 2020-04-30]. Dostupné z: https://www.root.cz/clanky/lehky-uvod-do-ldap/

[18] JAKUB, Krejčí. *Návrh datových vrstev pro datový sklad ČVUT*. Praha, 2017. Diplomová práce. České vysoké učení technické v Praze Fakulta informačních technologií.

[19] OGANESYAN-PEEL, Arturs. *Business Intelligence Tools We Recommend* [online]. Published 24.5.2018 [cit. 2020-04-30]. Dostupné z: https://blog.anant.us/business-intelligence-tools-we-recommend-1-4-metabase/

[20] INSAUSTI, Sebastian. *Running a Data Warehouse on PostgreSQL* [online]. Published 1.8.2019 [cit. 2020-04-30]. Dostupné

z: https://severalnines.com/database-blog/running-data-warehouse-postgresql

[21] DUNN, Ron. *Data Warehousing on PostgreSQL: FOSSASIA Summit 2016* [online]. 18.10.2016 [cit. 2020-04-30]. Dostupné z: https://www.youtube.com/watch?v=AuuLqKPldxs

[22] How to create a custom component. *Help.talend.com* [online]. [cit. 2020-04-30]. Dostupné z: https://help.talend.com/reader/QDC7DnW3S_chYXXornFuGw/3QRqVgh0vXKx3BPeUidHZw

[23] Building Robust ETL Pipelines with Apache Spark. *Databricks.com* [online]. Published 12.6.2016 [cit. 2020-04-30]. Dostupné z: https://databricks.com/session/building-robust-etl-pipelines-with-apache-spark

[24] SARING, Jonathan. *11 Javascript Data Visualization Libraries for 2019* [online]. [cit. 2020-04-30]. Dostupné z: https://blog.bitsrc.io/11-javascript-charts-and-data-visualization-libraries-for-2018-f01a283a5727

[25] License. *Metabase.com* [online]. [cit. 2020-04-30]. Dostupné z: https://www.metabase.com/license/

[26] *On snakes and elephants Using Python inside PostgreSQL: Using Python inside PostgreSQL* [online]. Published: 26.5.2015 [cit. 2020-04-30]. Dostupné z: https://wulczer.org/pywaw-summit.pdf

[27] BELAK, Simon, ed. *(Incomplete) MBQL Reference* [online]. Last edited 28.1.2020 [cit. 2020-04-30]. Dostupné z: https://github.com/metabase/metabase/wiki/(Incomplete)-MBQL-Reference

[28] JELÍNEK, Lukáš. Copyleftové licence: GPL, LGPL, AGPL. *Linuxexpres.cz* [online]. Publikováno 7.5.2014 [cit. 2020-04-30]. Dostupné z: https://www.linuxexpres.cz/copyleftove-licence-gpl-lgpl-agpl

[29] *LICENSE-EMBEDDING: METABASE APP-EMBED.JS SOFTWARE LICENSE AGREEMENT* [online]. [cit. 2020-04-30]. Dostupné z: https://github.com/metabase/metabase/blob/master/LICENSE-EMBEDDING.txt

Příloha B

Seznam použitých zkratek

Zkratka	Popis
BI	Business inteligence
KOS	Informační systém Komponenta studium
SZZ	Státní závěrečná zkouška
API	Rozhraní pro komunikaci s aplikací (Application Programming Interface)
IČO	Identifikační číslo osoby (unikátní záznam pro právnické a fyzické osoby)
Id	Identifikační číslo záznamu
3NF	Třetí normálová forma

Tabulka B.1: Seznam použitých zkratek

Příloha C

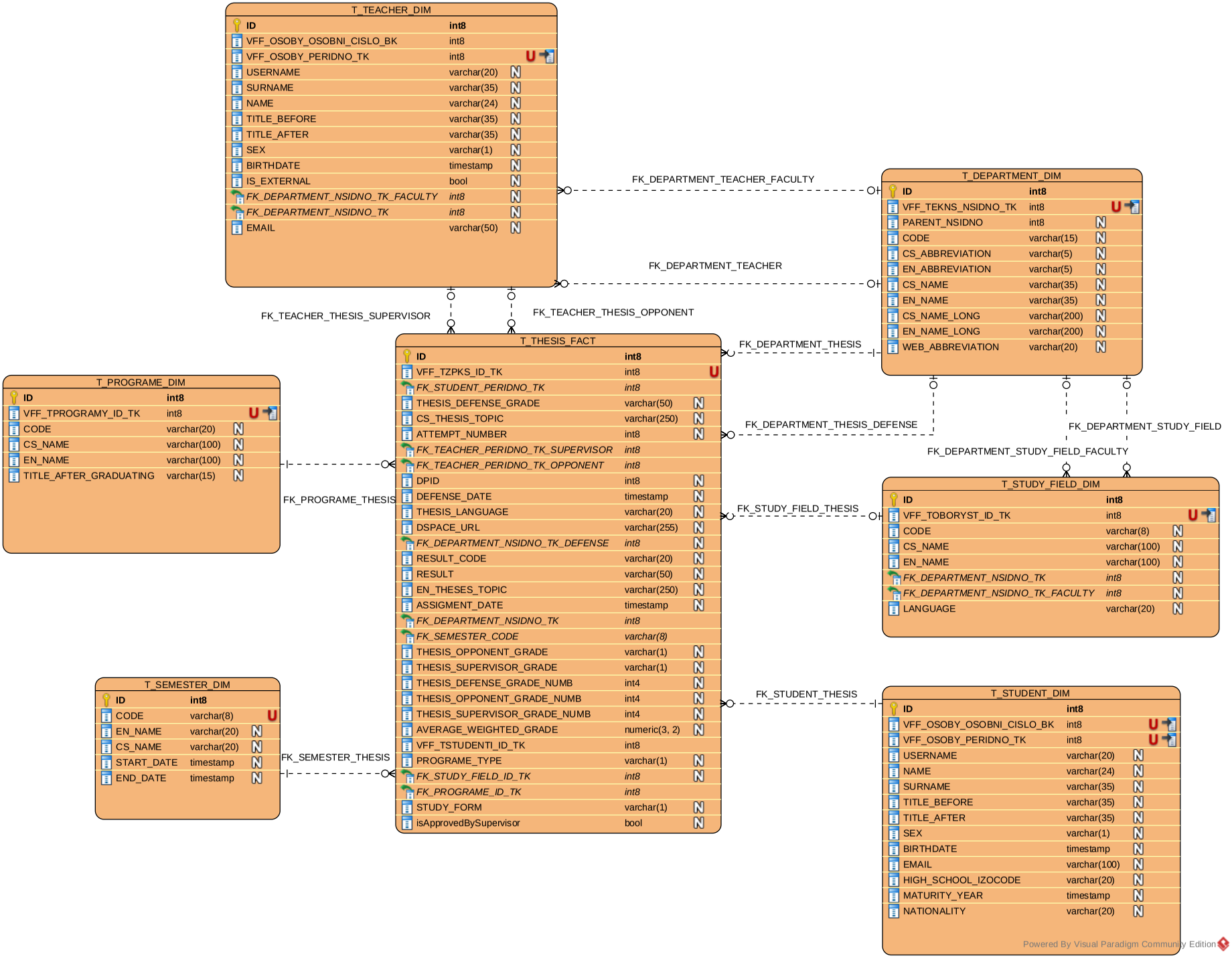
Obsah přiloženého CD

Složka	Popis obsahu
DB_model	složka obsahuje databázový model
DB_scripts	složka obsahuje skripty pro CREATE a DROP databáze
ETL_files	složka obsahuje: `ETL_copyAndFormatFile.py` - python soubor pro převod xls souboru do požadovaného formátu `ETL_tables.xlsx` - soubor s popisem zdrojových a cílových sloupců dat `ETL_THESIS_MARKET_TALEND.zip` - zdrojový kód ETL procesu
Metabase_configFiles	složka obsahující soubory Metabase

Tabulka C.1: Obsah přiloženého CD

Příloha D

Datový model


T_TEACHER_DIM
ID int8
VFF_OSOBY_OSOBNI_CISLO_BK int8
VFF_OSOBY_PERIDNO_TK int8 U
USERNAME varchar(20) N
SURNAME varchar(35) N
NAME varchar(24) N
TITLE_BEFORE varchar(35) N
TITLE_AFTER varchar(35) N
SEX varchar(1) N
BIRTHDATE timestamp N
IS_EXTERNAL bool N
FK_DEPARTMENT_NSIDNO_TK_FACULTY int8 N
FK_DEPARTMENT_NSIDNO_TK int8 N
EMAIL varchar(50) N
FK_DEPARTMENT_TEACHER_FACULTY
T_DEPARTMENT_DIM
ID int8
VFF_TEKNS_NSIDNO_TK int8 U
PARENT_NSIDNO int8 N
CODE varchar(15) N
CS_ABBREVIATION varchar(5) N
EN_ABBREVIATION varchar(5) N
CS_NAME varchar(35) N
EN_NAME varchar(35) N
CS_NAME_LONG varchar(200) N
EN_NAME_LONG varchar(200) N
WEB_ABBREVIATION varchar(20) N
FK_DEPARTMENT_TEACHER
FK_TEACHER_THESIS_SUPERVISOR
FK_TEACHER_THESIS_OPPONENT
T_THESIS_FACT
ID int8
VFF_TZPKS_ID_TK int8 U
FK_STUDENT_PERIDNO_TK int8
THESIS_DEFENSE_GRADE varchar(50) N
CS_THESIS_TOPIC varchar(250) N
ATTEMPT_NUMBER int8 N
FK_TEACHER_PERIDNO_TK_SUPERVISOR int8
FK_TEACHER_PERIDNO_TK_OPPONENT int8
DPID int8 N
DEFENSE_DATE timestamp N
THESIS_LANGUAGE varchar(20) N
DSPACE_URL varchar(255) N
FK_DEPARTMENT_NSIDNO_TK_DEFENSE int8 N
RESULT_CODE varchar(20) N
RESULT varchar(50) N
EN_THESES_TOPIC varchar(250) N
ASSIGMENT_DATE timestamp N
FK_DEPARTMENT_NSIDNO_TK int8
FK_SEMESTER_CODE varchar(8)
THESIS_OPPONENT_GRADE varchar(1) N
THESIS_SUPERVISOR_GRADE varchar(1) N
THESIS_DEFENSE_GRADE_NUMB int4 N
THESIS_OPPONENT_GRADE_NUMB int4 N
THESIS_SUPERVISOR_GRADE_NUMB int4 N
AVERAGE_WEIGHTED_GRADE numeric(3, 2) N
VFF_TSTUDENTI_ID_TK int8
PROGRAME_TYPE varchar(1) N
FK_STUDY_FIELD_ID_TK int8 N
FK_PROGRAME_ID_TK int8
STUDY_FORM varchar(1) N
isApprovedBySupervisor bool N
FK_DEPARTMENT_THESIS
FK_DEPARTMENT_THESIS_DEFENSE
FK_DEPARTMENT_STUDY_FIELD
FK_DEPARTMENT_STUDY_FIELD_FACULTY
T_PROGRAME_DIM
ID int8
VFF_TPROGRAMY_ID_TK int8 U
CODE varchar(20) N
CS_NAME varchar(100) N
EN_NAME varchar(100) N
TITLE_AFTER_GRADUATING varchar(15) N
FK_PROGRAME_THESIS
T_STUDY_FIELD_DIM
ID int8
VFF_TOBORYST_ID_TK int8 U
CODE varchar(8) N
CS_NAME varchar(100) N
EN_NAME varchar(100) N
FK_DEPARTMENT_NSIDNO_TK int8 N
FK_DEPARTMENT_NSIDNO_TK_FACULTY int8 N
LANGUAGE varchar(20) N
FK_STUDY_FIELD_THESIS
T_SEMESTER_DIM
ID int8
CODE varchar(8) U
EN_NAME varchar(20) N
CS_NAME varchar(20) N
START_DATE timestamp N
END_DATE timestamp N
FK_SEMESTER_THESIS
T_STUDENT_DIM
ID int8
VFF_OSOBY_OSOBNI_CISLO_BK int8 U
VFF_OSOBY_PERIDNO_TK int8 U
USERNAME varchar(20) N
NAME varchar(24) N
SURNAME varchar(35) N
TITLE_BEFORE varchar(35) N
TITLE_AFTER varchar(35) N
SEX varchar(1) N
BIRTHDATE timestamp N
EMAIL varchar(100) N
HIGH_SCHOOL_IZOCODE varchar(20) N
MATURITY_YEAR timestamp N
NATIONALITY varchar(20) N
FK_STUDENT_THESIS
Powered By Visual Paradigm Community Edition

Příloha E

Dashboard detailu katedry

Thesis Department | Semester | Thesis Type

Immediately after setting filter, graphs below will be filtered by selected values.

Metabase does not allow changing Y-axes values, therefore (A=1,B=1.5,C=2, D=2.5,E=3,F=4)

Tips:

1. You can add more things to one filter(it works like OR) , so it shows records for every value in the filter
2. You can click on the graph and show appropriete rows
3. You can click on a question (name of the question) and change what you want

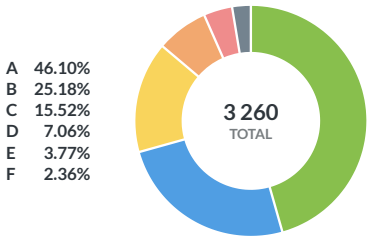
Drill down:

- Supervisors in Department (List)
- Opponents in Department (List)

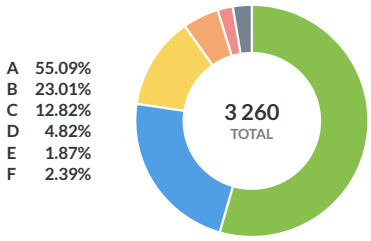
Roll up: Departments (List)

Similar Dashboards:

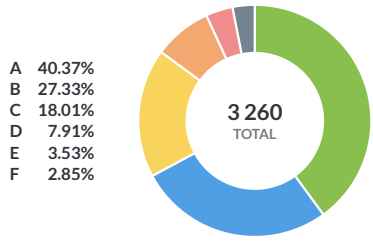
Percentage of Defense's Grades (Defense grade) [Defensed Thesis], T...



Percentage of Supervisor's Grades (Supervisor grade) [Defensed Thesis], ...



Percentage of Opponent's Grades (Opponent grade) [Defensed thesis], Th...

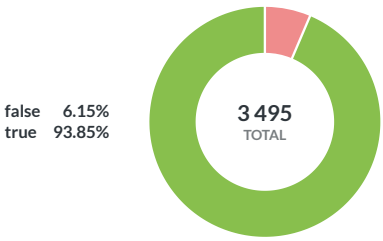


1,5
Average of Defense's Grades [Defensed Thesis], Thesis

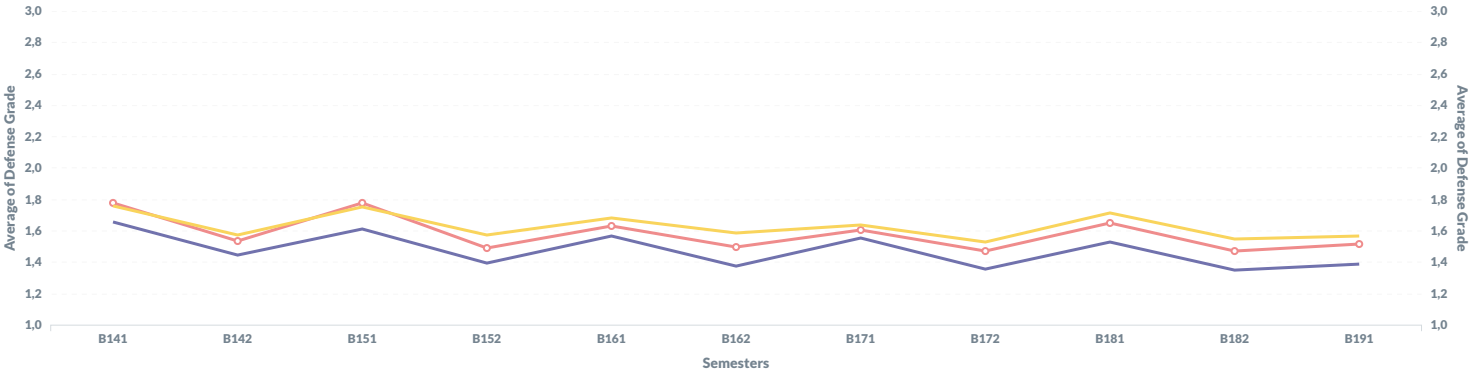
1,4
Average of Supervisor's Grades [Defensed Thesis], The...

1,6
Average of Opponent's Grades [Defensed Thesis], Thesis

Percentage of Approved Thesis (IsApprovedBySupervisor), Thesis



Average of Defense's Grades (Semester) [Defensed Thesis], Thesis | Average of Opponent's Grades (Semester) [Defensed Thesis], Thesis | Average of Supervisor's Grades (Semester) [Defensed Thesis], Thesis



Interesting questions concerning this Topic:

In this section are questions related to this dashboard.

This is just an overview of these questions

Questions:

- Average of Defense Grade
- Average of Students weighted grade

It is possible that some questions have been added, but not updated in this list. Link to all questions