ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Integrace systému Portál spolupráce s průmyslem do datového skladu ČVUT
Student:	Bc. Jan Mikeš
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 Portálu spolupráce s průmyslem (SSP).
3) Navrhněte datový model databáze s integrovanými daty (tzv. integrated data layer) na základě dat SSP.
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 9. ledna 2017

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

Diplomová práce

Integrace systému Portál spolupráce s průmyslem do datového skladu ČVUT

Bc. Jan Mikeš

Vedoucí práce: Ing. Stanislav Kuznetsov

2. května 2017

Poděkování

Na tomto místě bych chtěl poděkovat vedoucímu práce Ing. Stanislavu Kuznetsovi za rady a připomínky, které mi v průběhu psaní této práce poskytoval. Dále bych chtěl poděkovat kolegům z Fakulty informačních technologií, kteří mi pomohli seznámit se jak se systémem Portál spolupráce s průmyslem, tak s datovým skladem ČVUT. V neposlední řadě bych chtěl poděkovat i všem, co mě při psaní této práce jakýmkoliv způsobem podporovali.

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 2. května 2017 .....

Odkaz na tuto práci

Mikeš, Jan. *Integrace systému Portál spolupráce s průmyslem do datového skladu ČVUT.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato práce se zabývá integrací dat Portálu spolupráce s průmyslem do datového skladu ČVUT. Nad oběma systémy byla nejdříve provedena analýza. Z ní vychází návrh všech vrstev, které odpovídají architektuře existujícího datového skladu. Konkrétně se jedná o stage, integrovanou vrstvu, přístupovou vrstvu skládající se ze sémantické vrstvy a datových tržišť. Dále je popsána implementace vytvořených ETL procesů. Na závěr jsou uvedeny demonstrativní analytické reporty postavené nad celkovým řešením.

Klíčová slova spolupráce s průmyslem, datový sklad, datová integrace, normalizovaný model, dimenzionální model, datové tržiště, ETL, SCD, report

Abstract

This thesis deals with data integration of Cooperation with Industry system into CTU Data Warehouse. Firstly, analysis was conducted for both systems. Secondly, design was proposed for all the layers based on the architecture of existing data warehouse. The layers are: stage, integrated data layer and access layer which consists of semantic layer and data marts. Thirdly, implementation of ETL processes was described. Finally, analytic reports were created to demonstrate functional solution.

Keywords cooperation with industry, data warehouse, DWH, data integration, normalized model, dimensional model, data mart, ETL, SCD, report

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Počátky datového skladu na ČVUT sahají do doby, kdy se na Fakultě informačních technologií začal vytvářet projekt *Otevřený fakultní informační systém pro spolupráci s průmyslem* [8]. Od té doby prošel datový sklad několika iteracemi a změnami, podobně i systém pro spolupráci s průmyslem vyspěl do dnešní podoby a označuje se jako *Portál spolupráce s průmyslem (SSP)*.

Datový sklad v průběhu svého vývoje změnil celou svou strukturu a začal využívat jinou architekturu. Po ustálení největších změn se logicky začaly objevovat požadavky na integraci dalších zdrojů. Hlavním účelem skladu je totiž shromažďování dat z mnoha různých univerzitních systémů, mezi které patří např. studijní informační systém KOS. Díky tomu je možné udržovat konzistentní, agregovaná data na jednom místě. Datový sklad navíc umožňuje uchovávat historické hodnoty a díky tomu dává prostor pro vytváření analýz nad těmito daty.

Vývoj skladu není u konce, probíhá jeho neustálé zlepšování a zdokonalování. Příkladem jsou paralelně vznikající diplomové práce zaměřené na datovou integraci [9] a jednotlivé vrstvy datového skladu [10]. Současně probíhá snaha o integrování dalších systémů, jako je např. V3S [11].

Cílem této práce je navrhnout integraci dat z Portálu spolupráce s průmyslem do datového skladu ČVUT, implementovat základní části návrhu a následně demonstrovat fungující řešení pomocí několika základních analytických reportů.

Práce je členěna na dvě části. První má za účel seznámit čtenáře s problematikou datových skladů obecně. Předkládá teorii obklopující datové sklady, popisuje dvě základní architektury a jejich možná rozšíření a tyto architektury stručně porovnává. Ukazuje různé struktury dat a jejich použití v různých architekturách a vrstvách datového skladu. Dále zmiňuje možnosti a způsoby integrace dat a uchovávání historických hodnot.

Druhá část má praktický význam a ukazuje autorovu vlastní tvorbu. Analyzuje existující stav Portálu spolupráce s průmyslem a datového skladu ČVUT.

Na základě této analýzy navrhuje struktury jednotlivých vrstev skladu, vytváří a popisuje jejich implementaci. Jako demonstraci řešení dále ukazuje výsledky integrace ve formě jednoduchých reportů. Ty jsou vytvořeny jak nad daty, které pochází pouze z SSP, tak nad jejich kombinací s daty, jejichž zdroje jsou v datovém skladu již integrovány.

Část I

Teoretická část

Kapitola 1

Architektura datového skladu

Zrod datových skladů spadá do přelomu osmdesátých a devadesátých let dvacátého století [12]. Důvodem pro vznik datových skladů byly nedostatky v architektuře pro systémy na podporu rozhodování (angl. Decision Support Systems – DSS). Jedná se o informační systémy, které pomáhají při realizaci řídících a rozhodovacích činností v podnikání. Vznikly dva hlavní pohledy na datové sklady, dvě architektury, které se zachovaly dodnes. Jejími autory jsou William H. Inmon a Ralph Kimball. Následující odstavce shrnují a porovnávají tyto dvě architektury.

1.1 Architektura dle Williama H. Inmona

Inmon je považován za „otce datových skladů“ a je jedním z nejznámějších světových autorů publikací ohledně datových skladů a business intelligence. Pojem datový sklad definoval v roce 1992 ve své knize *Building the Data Warehouse* [13].

1.1.1 Historické důvody

Podle Inmona [2] byly nedostatky v tzv. procesech extrakce, které sloužily ke kopírování celých nebo jen části dat z jednoho úložiště do jiného. Tyto procesy byly používány kvůli následné analýze dat. Mezi jejich výhody patřil vysoký výkon a vlastnictví dat. Pokud byla data přesunuta do jiného systému, analýza nijak nezatěžovala původní systém. Uživatel po zkopírování data vlastnil a měl nad nimi úplnou kontrolu. Díky těmto dvěma důvodům byly procesy extrakce velice rozšířené. Tento způsob analýzy však přinášel i problémy, kterými byly hlavně nízká důvěryhodnost dat a produktivita a nemožnost získání informací z dat.

Příkladem mohou být dvě firemní oddělení, které vytváří analytický report. Jedno oddělení tvrdí, že firma prodala určité množství zboží, druhé oddělení však ukazuje množství jiné. Chyba byla dána právě procesy extrakce. Jedno

oddělení si např. zkopírovalo data jeden den, druhé jiný. Neexistoval tedy žádný časový kontext, obě oddělení mohla čerpat data z jiných zdrojů, např. už vytvořených pomocí procesů extrakce, nebo odlišnými dotazy.

Firma navíc nemusí mít jeden datový zdroj. Data mohou být uložena na různých místech. Při tvorbě reportu se všechna související data musela najít, zkopírovat a zpracovat. Každý zdroj tak přidal komplexitu a snížil tím produktivitu.

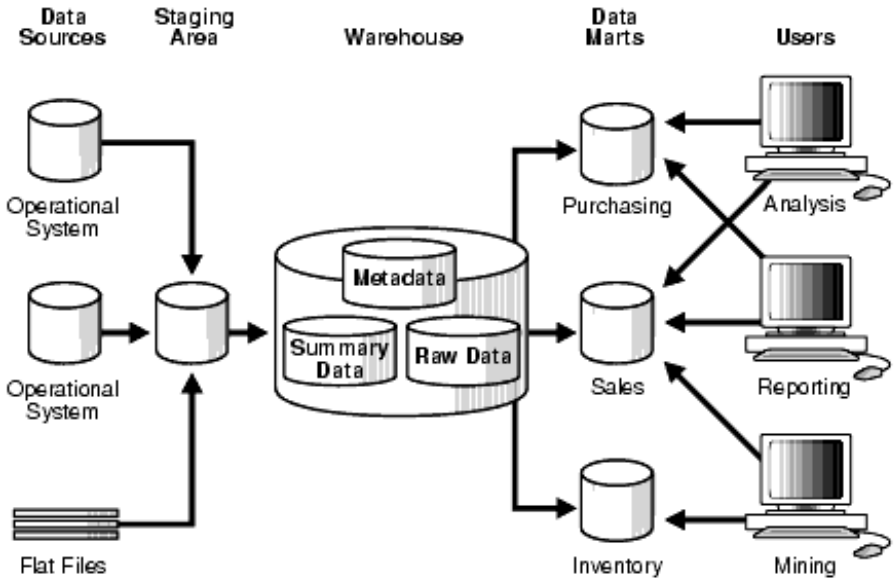
Transakční zpracování dat neuvažuje historii. Další problém tedy nastal, pokud se měla zjistit odlišnost ve stavu firmy za určité časové období (např. změna ve výplatách zaměstnanců za poslední rok). Pro takovou informaci nemusela data vůbec existovat.

1.1.2 Architektura

Uvedené problémy se snaží Inmon řešit svou koncepcí datového skladu. Definuje ho následovně:

> *„Data warehouse is a subject oriented, integrated, non volatile, time variant collection of data for management's decision making [2].“*

Existuje jedna verze pravdy, kterou je právě datový sklad [14]. Ten je součástí širší architektury, kterou nazývá Corporate Information Factory - CIF. Ukázka architektury podle Inmona je na obrázku 1.1.



Obrázek 1.1: Ukázka architektury dle Inmona, převzato z [1]

Data Sources V první úrovni se nachází zdrojové systémy, které se dělí na dva druhy – interní a externí. Příkladem interních jsou personální a mzdové systémy. Využívají se pro aktuální provoz a tomu odpovídá návrh úložiště. Data jsou často ukládána do databází, které jsou označovány jako OLTP (Online Transaction Processing) databáze. Jedná se o technologii, která slouží pro transakční zpracování dat, ale není přizpůsobena k využití pro tvorbu analytických reportů. Zdroji však nejsou jen databáze, ale mohou jimi být i soubory v různých formátech.

Staging Area Data ze zdrojů jsou zkopírována v nezměněné formě do dočasného úložiště dat (angl. Stage, Staging Area). Jedná se o přechodnou vrstvu, ze které se data dále integrují do datového skladu. Účelem této vrstvy je snížení vytížení zdrojových systémů pro analytické činnosti. Další výhodou této vrstvy je zafixování dat. Ta se mohou zálohovat pro případ havárie, navíc se nemění v průběhu transformací do datového skladu jako data, která jsou ve zdrojových systémech.

Warehouse Datový sklad je podle Inmona základem pro veškerý DSS processing. Vytváří jednu verzi pravdy a uchovává granulární data v normalizované formě (normalizovanou formou se zabývá sekce 2.1). Inmon tuto část nazývá Enterprise Data Warehouse (EDW).

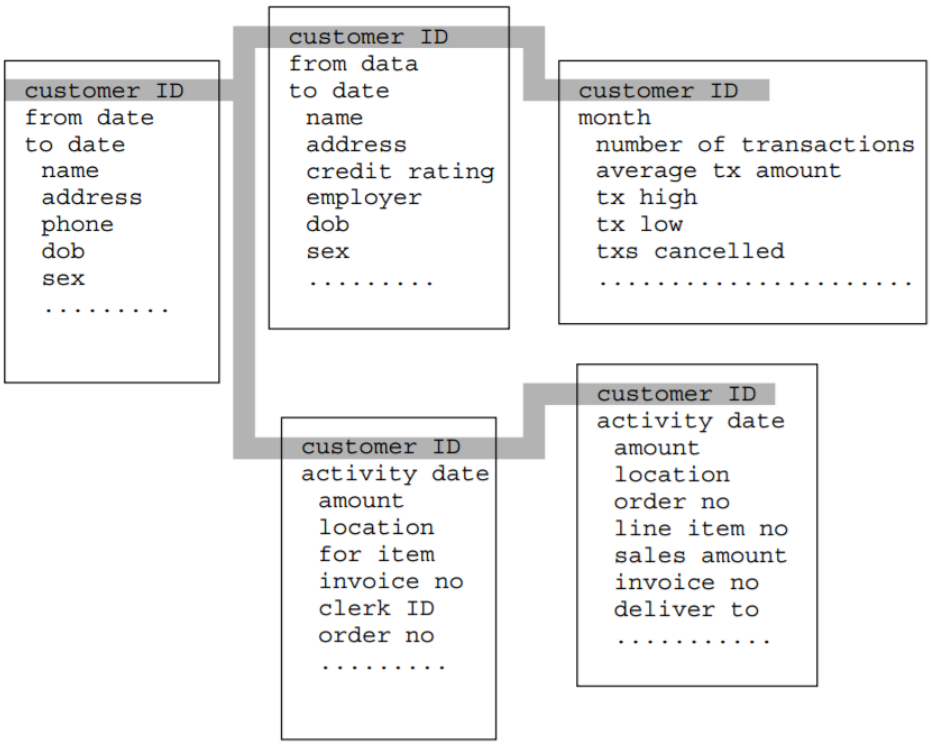
Data Marts Další vrstva se nazývá přístupová (angl. Access Layer) a tvoří ji datová tržiště (angl. Data Marts). Datové tržiště je kolekce oblastí zájmu, která jsou důležitá pro potřeby jednotlivých oddělení. Jednotlivá oddělení vlastní prostředky, které tvoří datové tržiště (hardware, software, data a technologie). Každé oddělení díky tomu může s daty v tržišti nakládat po svém. Tržiště jsou vytvořena pomocí dimenzionálního přístupu (více v sekci 2.2) a data jsou zde typicky uchována v hvězdicových schématech. Tato schémata jsou optimálně vytvořena pro analytické potřeby uživatelů v daném oddělení [15].

Users Poslední vrstvou v Inmonově architektuře jsou samotní uživatelé. Ti různými analytickými nástroji vytváří dočasné dotazy nad datovými tržišti. Pomocí těchto nástrojů tak vznikají analytické reporty, které podporují rozhodování v organizaci.

1.1.3 Charakteristika datového skladu

Podle Inmona je datový sklad subjektově orientovaný, integrovaný, nepodléhající změnám a časově proměnný. Tyto pojmy jsou rozvedeny v následujících odstavcích.

Subjektová orientace Typické operační systémy jsou zaměřeny na funkční celky, které souvisejí s aplikacemi dané společnosti. Příkladem podle Inmona může být pojišťovací firma s aplikacemi pro zpracování dat o autech, životních pojištěních a proběhlých nehodách. Datový sklad se má však zaměřovat na data orientovaná na subjekt, tím může být např. zákazník, pohledávka nebo pojistka. Data pro daný subjekt nemusí tvořit jediná tabulka. Mohou pocházet z různých zdrojů nebo z jednoho zdroje, kde se během času upravovala struktura. Ukázka vývoje struktury je na obrázku 1.2.



Obrázek 1.2: Ukázka různých struktur pro subjekt zákazníka, převzato z [2]

Integrovanost Nejdůležitější vlastností je integrovanost. Data jsou zpracovávána z různých zdrojů, které mohou být nesourodé. Při nahrávání do datového skladu jsou konvertována, formátována a jinak upravována. Zajišťuje se tím datová kvalita, podstata dat se nemění. Příkladem takových úprav je převod veličin na stejnou jednotku a sjednocení flagů. Výsledná data jsou pak konzistentní a jednotná bez ohledu na to, odkud pochází.

Neměnnost dat Další důležitou charakteristikou je neměnnost dat. Jednotlivé záznamy operačních dat jsou často upravovány. To je pro transakční zpracování typické. Do datového skladu jsou data nahrána, poté se však jako taková nemění. Změněná data jsou do skladu nahrána znovu a historická data zůstávají. Výjimku tvoří interní změny nezbytné pro funkci datového skladu, jako je např. udržení historie a návaznosti jednotlivých záznamů.

Časová proměnlivost Poslední charakteristikou je časová proměnlivost. Každá hodnota v datovém skladu je přesná vzhledem k určitému momentu v čase. Každá položka má určitým způsobem uchovaný časový údaj, ke kterému patří. Tomuto se také říká historizace dat.

1.1.4 Granularita dat v datovém skladu

Důležitým faktorem při návrhu datového skladu je granularita, která ovlivňuje veškerou architekturu obklopující datový sklad. Jedná se o úroveň detailu záznamů v datovém skladu. Čím detailnější, tím nižší granularita a naopak. Příkladem mohou být transakce v bance. Samotná transakce představuje nízkou (jemnou) granularitu, naopak souhrn transakcí za měsíc představuje vysokou (hrubou).

Granularita ovlivňuje množství dat, které se v datovém skladu nachází, a následné dotazy, které mohou být nad daty zodpovězeny. Při návrhu skladu se tak musí zvolit kompromis mezi množstvím dat ukládaných do skladu a univerzálností dotazů. Čím jemnější granularita, tím univerzálnější dotazy mohou být vytvořeny. Ve většině případů do datových skladů přichází data v hrubé granularitě a musí být rozpadnuta před uložením.

Výhodou jemné granularity je různorodost pohledů na data ve skladu. Každé oddělení totiž může mít jiné analytické požadavky. Jemná granularita přináší vyšší míru historie. Další výhodou je flexibilita. Oddělení se totiž může náhle rozhodnout pro změnu pohledu a nízká granularita toto umožní. Vzhledem k náročnosti vytvoření infrastruktury datového skladu je jemná granularita dobrá, při změně požadavků je na ně díky tomu datový sklad připraven. Na druhou stranu vyžaduje nízká granularita více zdrojů, jakými jsou např. paměť pro úložiště a výpočetní síla daného hardware.

Pro snížení náročnosti na výkon, která je spjata s obrovským množstvím dat, se používá dvojí úroveň granularity. Data s granularitou v nízké úrovni jsou uložena po určitou dobu. Po uplynutí této doby se data sumarizují do vyšší úrovně granularity a jsou uložena jinam. Nad nimi probíhá většina dotazů. Pokud je potřeba analýza nad daty jemné granularity, použije se archiv, kam byla tato data uložena.

1.1.5 Tvorba datového skladu

Inmon navrhuje iterativní cyklus vytváření datového skladu. Tvrdí, že pojem „návrh“ datového skladu není přesný, jelikož naznačuje, že mohou být jednotlivé prvky naplánované dopředu. To nemusí být pravda, všechny požadavky nejsou vždy známé předem, ale mohou se objevit až při používání. Navrhuje tedy tvorbu skladu ve fázích, kde jedna fáze vývoje závisí na výsledcích z předchozí. Nejdříve je vytvořena část skladu, DSS analytik tuto část zkontroluje a začne používat. Na základě zpětné vazby jsou data upravena či přidána. Dále je vybudována další část skladu. Tento proces pokračuje v průběhu celého životního cyklu datového skladu.

Datový sklad tak nemůže být vybudován na základě způsobů vývoje, který je řízený požadavky. Předvídání požadavků by však nemělo být ignorováno, ale bráno v potaz pro snazší budoucí rozšíření.

Návrh datového skladu se podle Inmona řídí tzv. top-down přístupem. Nejdříve je vytvořen normalizovaný datový model, který tvoří centralizovaný datový sklad. Shromažďuje tak data ze všech operačních systémů a vytváří jednu verzi pravdy. Až po vytvoření datového skladu je na řadě tvorba dimenzionálních datových tržišť pro jednotlivá oddělení případně jednotlivé analytické požadavky [16].

1.1.6 Výhody a nevýhody

Architektura podle Inmona má následující výhody [1]:

- Datový sklad je jediným zdrojem pravdy, obsahuje integrovaná data a slouží jako zdroj pro všechna datová tržiště.
- Jsou odstraněny anomálie při aktualizacích dat díky normalizované formě.
- Obchodní procesy mohou být snadno pochopeny díky modelu zaměřenému na subjekty.
- Jedná se o flexibilní architekturu, jelikož je poměrně jednoduché upravit datový sklad, který obsahuje všechna data pohromadě.
- Umožňuje vytváření analytických reportů napříč celou organizací.

Existují však i nevýhody této architektury [1]:

- Model a implementace se může stát poměrně komplexní v průběhu vývoje datového skladu.
- Pro kvalitní datový sklad je potřeba zkušených datových analytiků.
- Prvotní vytvoření a nasazení zabere více času.
- Je potřeba vytvořit více transformací, jelikož se data integrují nejdříve do datového skladu a až poté do datových tržišť.

1.2 Architektura dle Ralpha Kimballa

Ralph Kimball jako první představil koncept dimenzionálního modelování. Učinil tak v roce 1996 ve své knize *The Data Warehouse Toolkit* [17].

1.2.1 Požadavky na datový sklad

Podobně jako Inmon viděl Kimball [4] nedostatky, které nastávaly při analýze dat. Jednou z nejdůležitějších věcí pro organizaci jsou podle něj informace. Operační systémy, které s těmito informacemi pracují, byly poměrně úzce optimalizované pro svůj účel. Tyto systémy však z podstaty jejich fungování neudržovaly žádnou historii. Ta je ale pro analytická rozhodnutí nezbytná. Operační systémy také nebyly přizpůsobeny výkonově pro podporu analytických rozhodnutí. To přimělo Kimballa definovat požadavky na datový sklad.

Datový sklad musí udržovat informace jednoduše poskytnutelné Obsah datového skladu musí být snadno pochopitelný a intuitivní. Struktura dat by měla odpovídat používaným firemním entitám. Nástroje, které přistupují k datům, musí být jednoduché a snadno použitelné. Musí také vracet výsledky dotazů s minimální dobou odezvy.

Datový sklad musí poskytovat informace konzistentně Data v datovém skladu musí být důvěryhodná. Musí být shromážděna z různých zdrojů, vyčištěna, musí u nich být zajištěna datová kvalita. Konzistence se také týká názvosloví a definic.

Datový sklad musí reagovat na změny Potřeby uživatelů, data a technologie se neustále mění. Datový sklad musí být navržen tak, aby byl schopný tyto změny podpořit. Existující data by takovými změnami neměla být nijak porušena.

Datový sklad musí poskytovat informace v rozumném čase Informace z datových skladů mohou ovlivňovat rozhodnutí, která jsou potřeba udělat v určitém časovém horizontu. Uživatelé datového skladu musí mít realistický odhad, v jakém časovém horizontu je možné důležité informace z dat poskytnout s minimem času na jejich validaci.

Datový sklad musí udržovat informace v bezpečné formě Datový sklad obsahuje data, která jsou pro organizaci důležitá. Přístup k těmto citlivým datům tedy musí být kontrolovaný.

Datový sklad musí sloužit jako autoritativní a důvěryhodný zdroj informací pro zlepšení rozhodování Datový sklad musí obsahovat

správná data. Rozhodnutí, která jsou ovlivněna právě výstupy z datových skladů, mohou mít dopad na celou firmu. Data tedy musí být korektní ve všech případech.

Business sféra musí datový sklad přijmout, aby byl úspěšný I když je datový sklad velice efektivní a splňuje všechny požadavky na něj kladené, je to zbytečné, pokud nebude přijat v business sféře. Datový sklad oproti operačnímu systému není nutnou podmínkou pro chod firmy. Musí tvořit jednoduchý a rychlý zdroj informací.

1.2.2 Architektura

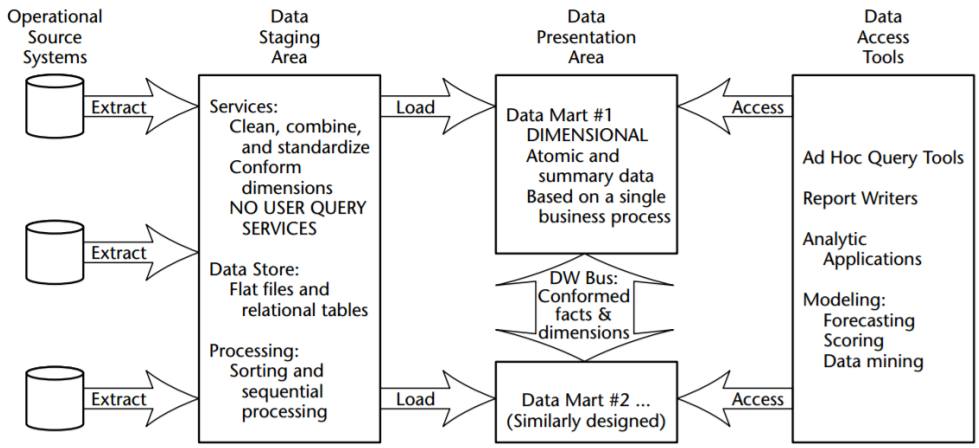
Kimball v jedné ze svých prvních publikací definoval datový sklad následovně:

> *„A data warehouse is nothing more than the union of the data marts [17].“*

Svoji definici později upravil do následujícího tvaru:

> *„Data warehouse is a copy of the data specifically structured for query and analysis [14].“*

Architektura podle Kimballa se skládá z několika vrstev. Některé z nich odpovídají vrstvám v architektuře podle Inmona. Základní struktura je na obrázku 1.3 a jednotlivé vrstvy jsou stručně popsány v následujících odstavcích.



Obrázek 1.3: Architektura dle Kimballa, převzato z [3]

Operational Source Systems Data se nachází v operačních systémech. Tyto systémy žijí mimo datový sklad a slouží k běžnému chodu firmy. Hlavní úlohou je zpracování transakcí a nejsou nad nimi dělány analytické dotazy. Neobsahují žádná, nebo velice omezená historická data. Jejich hlavní prioritou je výkon a dostupnost.

Data Staging Area Jedná se současně o úložiště dat a množinu ETL procesů (o ETL hovoří sekce 3.2). Stage, podobně jako vrstvu zdrojových systémů, má i architektura dle Inmona. Klíčovým prvkem této vrstvy je, že není přístupná běžným uživatelům a neprobíhají nad ní žádné dotazy prezentační vrstvy. Úložiště je dočasné a slouží k separaci operačních systémů a prezentační vrstvy.

Data Presentation Area V této vrstvě jsou data organizována, uložena a zpřístupněna pro přímé dotazy od uživatelů analytických aplikací. Jelikož je předchozí vrstva pro uživatele skryta, prezentační vrstva je pro ně synonymem datového skladu. Všechny analytické nástroje využívají právě tuto vrstvu. Skládá se z jednotlivých datových tržišť, ve kterých jsou data uložena v dimenzionální podobě.

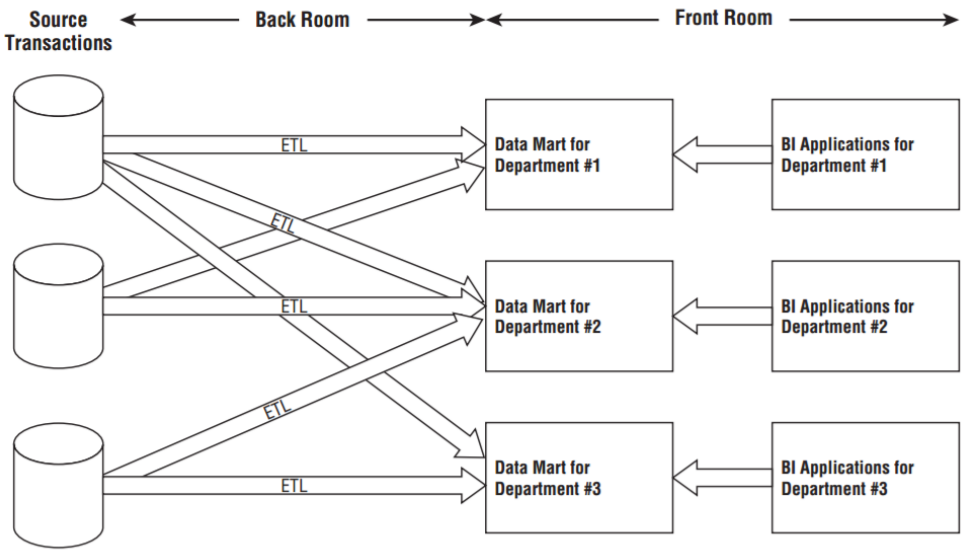
Data Access Tools Poslední komponentou architektury datového skladu jsou analytické nástroje, které přistupují k datům z předchozí vrstvy. Mezi ně se řadí jednoduché dotazovací nástroje, ale i složité nástroje pro dolování znalostí dat (angl. Data Mining).

1.2.3 Udržení konzistence

Podobnou architekturou, jako je uvedena v předchozích odstavcích, je architektura nezávislých datových tržišť. Její ukázka je na obrázku 1.4 a odpovídá první definici datového skladu podle Kimballa uvedenou na začátku předchozí sekce, a tedy že *„datový sklad je pouhým sjednocením datamartů“*. Tuto architekturu označuje Inmon jako první fázi Kimballovy architektury a nazývá ji jednoduchým dimenzionálním modelem [14].

Architektura nezávislých datových tržišť však přináší problémy. Při růstu množství zdrojových systémů a datových tržišť roste i komplexita tohoto modelu. Čím dál hůře se udržuje konzistence, přibývá stále více redundantních dat. Tento problém Kimball řeší tzv. architekturou sběrnice (angl. Enterprise Data Warehouse Bus Architecture), která využívá tzv. odpovídající dimenze (angl. Conformed Dimensions).

Odpovídající dimenze jsou standardizované dimenze, které slouží k použití v různých datových tržištích a jsou sdílené napříč faktovými tabulkami (dimenzionální a faktové tabulky popisuje sekce 2.2). Na identifikaci a tvorbu odpovídajících dimenzí Kimball doporučuje tzv. matici sběrnice (angl. Bus Matrix), jejíž řádky odpovídají obchodním procesům a sloupce dimenzím.



Obrázek 1.4: Architektura nezávislých datových tržišť, převzato z [4]

Buňka v matici je označena, pokud spolu odpovídající proces a dimenze souvisí. Ukázka je na obrázku 1.5.

1.2.4 Tvorba datového skladu

Kimball je zastáncem opačného přístupu při budování datového skladu oproti Inmonovi. Doporučuje, aby byl vytvořen pomocí tzv. bottom-up přístupu [16]. Nejdříve se začne s nejvíce kritickými datovými tržišti, které slouží analytickým potřebám jednotlivých oddělení. Poté nastává integrace těchto tržišť tak, aby byly konzistentní, pomocí architektury sběrnice. Data v tržištích jsou udržována pomocí dimenzionálního přístupu.

1.2.5 Výhody a nevýhody

Výhody architektury podle Kimballa jsou [1]:

- Rychlé vytvoření a dodání první části datového skladu.
- Hvězdicové schéma je snadno pochopitelné pro uživatele.
- Menší velikost prostředí datového skladu umožňuje jeho snazší správu.
- Pro efektivní chod datového skladu je potřeba malý tým vývojářů a architektů.
- Výkon dimenzionálního přístupu je velmi dobrý.

BUSINESS PROCESSES	COMMON DIMENSIONS						
	Date	Product	Warehouse	Store	Promotion	Customer	Employee
Issue Purchase Orders	X	X	X				
Receive Warehouse Deliveries	X	X	X				X
Warehouse Inventory	X	X	X				
Receive Store Deliveries	X	X	X	X			X
Store Inventory	X	X		X			
Retail Sales	X	X		X	X	X	X
Retail Sales Forecast	X	X		X			
Retail Promotion Tracking	X	X		X	X		
Customer Returns	X	X		X	X	X	X
Returns to Vendor	X	X		X			X
Frequent Shopper Sign-Ups	X			X		X	X

Obrázek 1.5: Ukázka matice sběrnice, převzato z [4]

- Je dobře použitelný pro metriky jednotlivých oddělení, na která se orientují datová tržiště.

Architektura má ale i následující nevýhody [1]:

- Neexistuje jedna verze pravdy a data nejsou plně integrována.
- Existují redundantní data, která mohou vést k anomáliím při úpravě dat.
- Přidání sloupců do faktové tabulky může snížit výkon, jelikož tím vzroste její velikost.
- Integrace starších dat může být poměrně komplexní.

1.3 Srovnání architektur

Obě architektury získávají data ze zdrojových systémů, obsahují stage a dimenzionální datová tržiště. U Kimballa však neexistuje jedno centrální úložiště, jedna verze pravdy, což na jeho architektuře kritizuje i samotný Inmon [14].

Díky neexistenci jedné verze pravdy je znatelně rychlejší vytvořit prvotní část datového skladu dle Kimballa. Pokud však roste množství datových tržišť,

integrace dat a jejich udržitelnost může být z dlouhodobého hlediska složitější než u Inmona. Roste také množství duplicitních dat. Další porovnání je uvedeno v tabulce 1.1.

Tabulka 1.1: Porovnání architektury dle Inmona a Kimballa, převzate z [7]

Hledisko	**Inmon**	**Kimball**
Vytvoření	časově náročné	méně časově náročné
Údržba	jednoduchá	obtížná
Cena	vysoká počáteční, nižší při rozvoji	nízká počáteční, stejná při rozvoji
Požadavky na zkušenosti	specialisté	generalisté
Požadavky na integraci	přes celou organizaci	jednotlivá oddělení
Požadavky na rozhodování	strategické	taktické

1.4 Rozšíření architektury

Architektury podle Inmona a Kimballa nejsou jediné. Ukázkou mohla být architektura nezávislých datových tržišť uvedená v sekci 1.2.3. Architektury jako takové mohou být rozšířené o další vrstvy nebo jiné součásti. Některé z nich jsou popsány v následujících odstavcích.

1.4.1 Landing Layer

Jedná se o vrstvu, která vznikla pro jasně vymezené oddělení zodpovědnosti mezi dodáním dat ze zdrojových systémů a jejich následným zpracováním. Tato vrstva se nachází mezi zdrojovými systémy a dočasným úložištěm dat (Staging Area). Obsahuje data extrahovaná ze zdrojových systémů v nezměněné podobě nebo ODS databáze (více o ODS v sekci 1.4.2).

Zdrojová data jsou v této vrstvě uložena a archivována. Vrstva zajišťuje, že jsou data dostupná v krátkém čase. Je zde také kontrolována správnost dat ze zdrojových systémů a dochází k prvotnímu pročištění [18]. Zdrojové systémy mohou být na rozdíl od dat ve Staging Area nestrukturované. Transformace na strukturovaná data nastává právě mezi Landing a Staging Area.

Do Staging Area proudí už jen data s obchodním významem, ne technické struktury. Landing Area kvůli tomu může obsahovat více dat, než je ve výsledku integrováno do datového skladu. Staging Area pak zajišťuje vytvoření inkrementu pro integraci.

1.4.2 Operational Data Store

Operativní datové sklady (angl Operational Data Store - ODS) se mohou nacházet mezi zdrojovými systémy a datovým skladem. Některá data putují ze

zdrojových systémů do datového skladu, jiná do ODS a teprve poté do datového skladu. Zdrojem dat pro ODS mohou však být i některá data z datového skladu. Definice ODS podle Inmona je:

> *„An ODS is an integrated, subject oriented, volatile (including update), current-valued structure designed to serve operational users as they do high performance integrated processing.[19]“*

Subjektová orientace a integrovanost byla vysvětlena v sekci 1.1.3. Oproti datovému skladu je však ODS proměnlivý a obsahuje pouze aktuální informace, nebo jen velmi omezenou, krátkodobou historii. Dále se v něm mohou nacházet agregovaná data. To vše slouží k podpoře rozhodování. Používá se i jako jednotný integrovaný zdroj pro aktuální data a transakční zpracování, jelikož je viditelný pro uživatele, což datový sklad být nemusí.

1.4.3 Semantic Database

Jak již bylo zmíněno v definici EDW, data jsou v něm uložená v normalizované formě. Může se stát, že jednu business entitu, ke které se vztahuje analýza, tvoří více tabulek. Datová tržiště pak pracují s touto entitou, tudíž vnitřně pracují s těmito tabulkami. Každé datové tržiště však může použít pro svou analýzu jinou tabulku a kvůli tomu může vznikat nekonzistence. Konzistence je jedním ze základních podmínek pro tvorbu správné analýzy, proto musí být řešena.

Datová tržiště nemusí nutně pracovat s daty ve formátu, v jakém je dostane, ale data pro svoji analýzu z nich může odvozovat. To může být časově a výkonově náročné. Navíc se stejnou strukturou může pracovat více datových tržišť a výpočet probíhá zbytečně vícekrát.

Oba problémy řeší právě sémantická databáze (angl. Semantic Database). Tato databáze obsahuje datové struktury, které přímo odpovídají entitám. To je rozdíl oproti datovému skladu, který udržuje subjektově orientovaná data. Může sdružit několik tabulek do jedné, čímž odstraňuje nekonzistence, nebo obsahuje již zpracovaná data, čímž zvyšuje výkon a snižuje redundanci výpočtů. Struktura dat může být z tohoto důvodu denormalizovaná. Data nejsou specifická pro konkrétní reporty, slouží jako obecný vstup do datových tržišť, které s nimi dále pracují. Sémantická databáze jako rozšíření architektury dle Inmona plní stejnou funkci jako odpovídající dimenze v architektuře dle Kimballa.

Kapitola 2

Struktura dat

V jednotlivých vrstvách datového skladu jsou data uložena v různých strukturách. Tyto struktury jsou přizpůsobeny požadavkům, které jsou k daným vrstvám vztažené. V EDW jsou data uložena v normalizované podobě, v datových tržištích naopak v denormalizované. Oba přístupy jsou popsané v následujících sekcích.

2.1 Normalizovaný přístup

Normalizovaná struktura dat vzniká tzv. normalizací. Jedná se o proces, kde je struktura dat upravena tak, aby došlo k zvýšení koheze typů entit. Tento efekt je dosažen pomocí dekompozice tabulek. Jinými slovy, cílem normalizace je zredukovat, v nejlepším případě zcela odstranit redundanci dat. Minimalizování redundance je důležitý faktor při vývoji databázových systémů pro transakční zpracování, jelikož je složité správně uchovávat stejné informace na několika různých místech.

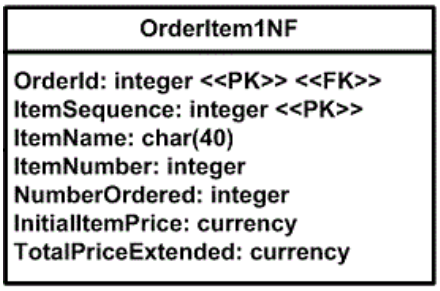
2.1.1 Normální formy

Proces normalizace probíhá postupně. Dekompozicí se získává stále nižší úroveň redundance. Existuje více různých stupňů normalizace, tzv. normální formy. Nejčastější normální formou je třetí normální forma (3NF). Databáze se často označuje jako normalizovaná, právě pokud splňuje 3NF. Ta vychází z předchozích, jejich popis je uveden v následujících odstavcích.

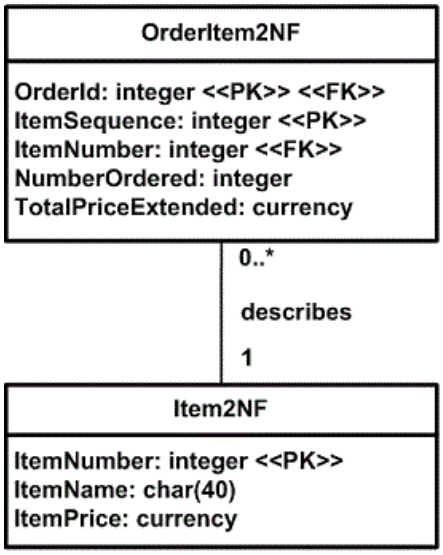
První normální forma (1NF) První normální forma požaduje, aby byla data atomická a neobsahovala opakující se skupiny [20]. Atomická data jsou taková, která mají pouze jednu hodnotu v každé buňce tabulky. Příkladem mohou být adresy. Pokud jsou celé adresy uložené v jednom sloupci, nejedná se o atomická data. Ta vzniknou rozdělením takového sloupce na jednotlivé položky adresy, jakými jsou např. ulice, číslo domu,

město. Opakující se skupiny odpovídají polím v programovacích jazycích. Pokud je v jedné buňce výčet věcí (např. výčet předmětů, které učí jeden člověk), jedná se o opakující se skupinu a opět není splněna 1NF.

Druhá normální forma (2NF) Druhá normální forma vychází z první. Tabulka je ve 2NF, pokud je v 1NF a současně všechny atributy, které nejsou součástí klíče, jsou plně závislé na primárním klíči [20]. Příkladem porušení 2NF může být tabulka, která jako primární klíč obsahuje číslo objednávky a jako další atribut cenu objednávané položky (ukázka na obrázku 2.1a). Cena však nezávisí na objednávce, ale na položce samotné. Řešením by tedy bylo vytvoření nové tabulky, která by obsahovala informace o položce a byla identifikována číslem položky (ukázka na obrázku 2.1b).



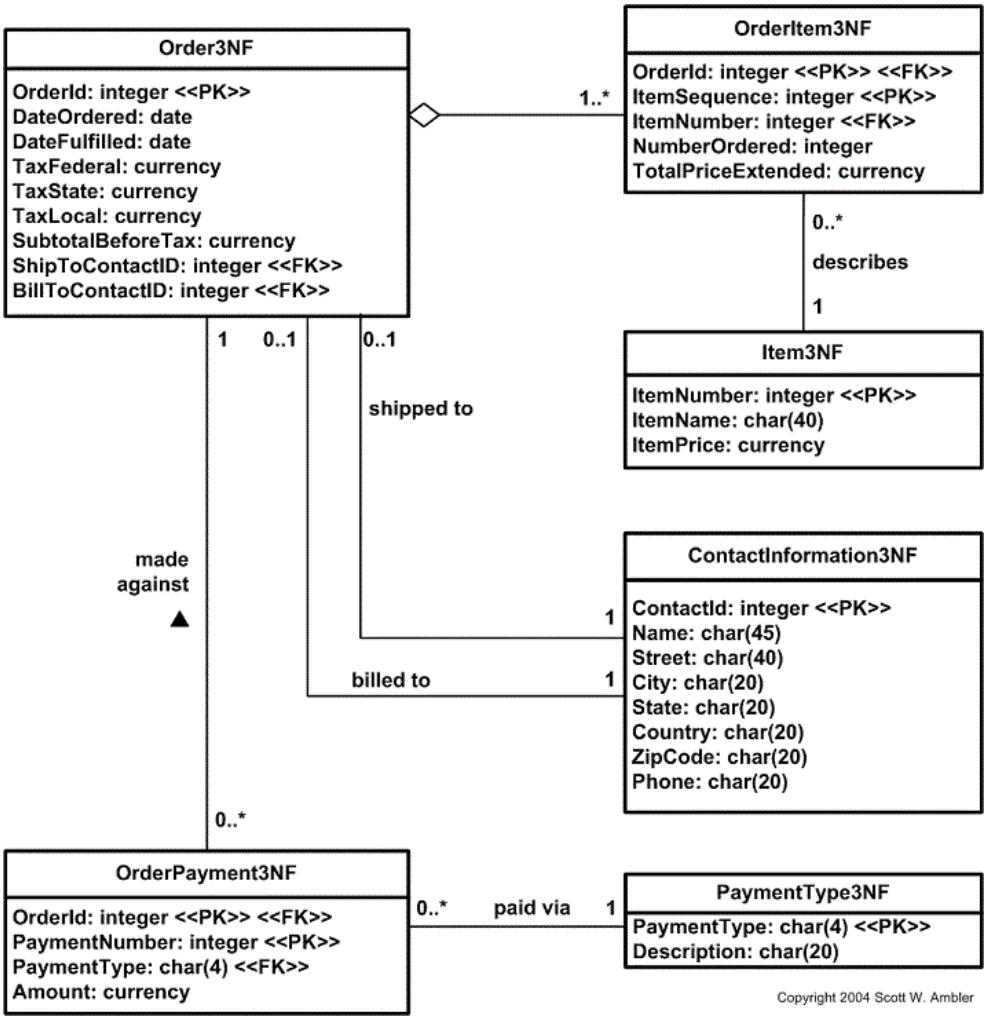
(a) Porušení 2NF, převzato z [5]



(b) Schéma ve 2NF, převzato z [5]

Obrázek 2.1: Ukázka 2NF

Třetí normální forma (3NF) Třetí normální forma řeší tranzitivní závislosti. Opět vychází z předchozí, tedy 2NF, současně musí být všechny atributy přímo závislé na primárním klíči. Jinak řečeno, atribut entity musí záviset na všech částech primárního klíče. Pokud by např. popis typu platby byl v tabulce, kde část primárního klíče tvoří číslo objednávky, není splněna 3NF, jelikož popis typu platby vůbec nezávisí na objednávce. Tento popis by se musel vyextrahovat do další tabulky, kde by klíčem byl např. právě typ platby. Ukázka 3NF je na obrázku 2.2.



Obrázek 2.2: Schéma ve 3NF, převzato z [5]

2.1.2 Výhody a nevýhody

Mezi výhody patří menší velikost databáze, která je způsobena eliminací duplicitních dat. V některých případech dochází ke zvýšení výkonu. To je způsobeno tabulkami s menším počtem sloupců, což umožňuje umístit více záznamů do jedné stránky. Menší tabulky obsahují i méně indexů a je proto jednodušší a rychlejší je přeindexovat. Při spojování tabulek lze využít pouze ty, které s dotazem nutně souvisí.

Normalizovaná struktura také odstraňuje anomálie. Pokud se vkládá do nenormalizované tabulky záznam o entitě, zbytek atributů, které s touto entitou přímo nesouvisí, nabývá hodnoty NULL. Při úpravě záznamu může dojít k úpravě pouze na určitých místech (např. jen na jednom řádku) a tím vzniká nekonzistence. Problém nastává i při mazání záznamu, kdy se přijde o data,

která by se v normalizované formě uchovala.

Normalizace však přináší i nevýhody. Mezi ně patří velké množství tabulek a s tím související zvýšená potřeba jejich spojování. Tabulky jako takové většinou obsahují umělé klíče, které nejsou na první pohled srozumitelné a ztěžují čtení dat. V některých případech je složité vytvořit dotaz, datový model totiž není přizpůsoben ad hoc dotazům.

2.2 Dimenzionální přístup

Dimenzionální modelování je široce akceptovaná a preferovaná technika pro poskytování analytických dat [4]. Má dvě jednoznačné výhody, první je jednoduchost pochopení pro uživatele a druhou je rychlost provedení analytických dotazů. K dosažení tohoto cíle však dimenzionální model přináší redundanci v datech.

2.2.1 Tabulky

Dimenzionální model tvoří dva základní druhy tabulek. Jedná se o faktové a dimenzionální tabulky, oba druhy jsou rozebrány v následujících odstavcích.

2.2.1.1 Faktové tabulky

Faktová tabulka v dimenzionálním modelu uchovává naměřené hodnoty z událostí, které vznikají z business procesů organizace. Jedná se o míry (fakta), na kterých se provádí analýza. Příkladem může být počet a cena nakoupených položek v transakci. Základním principem dimenzionálního modelování je, že hodnoty ve všech řádcích faktové tabulky mají stejnou úroveň detailu, tedy stejnou granularitu. Úroveň detailu bývá co nejvyšší.

Nejužitečnější jsou míry, které mají číselnou hodnotu a jsou aditivní. Zřídka se pro analýzu načítá pouze jeden řádek faktové tabulky. Většinou se načítá velké množství takových řádků a nejčastější operací na načtených hodnotách je právě jejich součet. U některých hodnot však nemá smysl slepě sčítat všechny (např. součet zůstatků na různých účtech za určité časové období), v tomto případě se jedná o semiaditivní míry. Existují i neaditivní míry (např. jednotková cena), u kterých má smysl udávat jejich počet nebo průměr.

Fakta často nabývají reálných hodnot, což pomáhá k jejich identifikaci. Je možné, i když velmi výjimečné, že naměřená hodnota bude textového charakteru. Většinou se jedná o výčet diskrétních hodnot. Tyto hodnoty lze převést do číselné reprezentace a ušetřit tak místo. Ve faktových tabulkách není doporučeno ukládat redundantní textové informace. Pokud se nejedná o unikátní text pro každý řádek, patří tato informace do dimenzionální tabulky. Skutečná textová míra je velmi vzácná, protože je velmi složité, v některých případech i nemožné, takovou míru analyzovat.

Důležitým pravidlem je, že se netvoří zbytečné řádky. Nemá smysl uložit řádek s nulovými hodnotami, které nereprezentují žádnou aktivitu. Uchováním pouze skutečných aktivit zůstávají faktové tabulky řídké. I přesto zabírají obvykle až 90 % prostoru dimenzionálního modelu [4]. Typické tabulky obsahují málo sloupců, ale bývají rozsáhlé z hlediska počtu řádků.

Faktové tabulky obsahují cizí klíče, které odkazují do dimenzionálních tabulek. K faktům se pak dá přistoupit pomocí dimenzí, které jsou s nimi propojeny. Ve většině případů stačí pouze několik dimenzí, které dohromady jednoznačně identifikují řádek ve faktové tabulce. Primární klíč je pak často tvořen právě cizími klíči těchto dimenzí.

Existují tří základní druhy faktových tabulek: transakční, periodické snímky a akumulační snímky. Nejpoužívanějším typem je transakční faktová tabulka.

Transakční faktové tabulky Nejzákladnějším pohledem na operace v podniku je pohled z hlediska jednotlivých transakcí. Řádek v tabulce reprezentuje událost, která nastala v konkrétním časovém bodě, a existuje pouze, pokud transakce opravdu nastala. Atomické transakce jsou jedny z nejčastějších dimenzionálních dat a umožňují analýzu ve vysokém detailu.

Periodické snímky Tento druh tabulek uchovává kumulativní statistiku pro daný časový úsek. Často se jedná o jediný zdroj, ze kterého je snadné získat pravidelný pohled na trendy firmy. V některých případech se jedná o pouhou kumulaci transakcí, která vznikla za účelem zvýšení výkonu. V jiných situacích může být poměrně složité dostat všechny související transakce a vytvořit z nich požadovanou statistiku. V takovém případě pomohou periodické snímky, kde se tento složitý výpočet provede jen jednou a výsledky se uloží do tabulky.

Akumulační snímky Jedná se o nejméně častý typ tabulek. Reprezentují procesy s jasným počátkem, koncem a standardní množinou kroků. Jsou vhodné pro analýzu různých workflow. Obsahují několik odkazů do dimenze času, které udávají, kdy proběhl jaký milník.

2.2.1.2 Dimenzionální tabulky

Dimenzionální tabulky jsou nedílným doplňkem tabulek faktových. Obsahují textový kontext spjatý s naměřenými hodnotami. Slouží jako odpovědi na otázky typu *kdo, co, kde, kdy, jak a proč*, které souvisí s měřením.

Oproti faktovým tabulkám mají více sloupců. Není neobvyklé, že dimenzionální tabulky obsahují mezi 50 a 100 sloupci [4]. Existují samozřejmě i tabulky s menším počtem atributů, i tak jich bývá více než ve faktových tabulkách. Další odlišností od faktových tabulek je menší počet řádků.

Dimenze slouží jako primární zdroj omezení a slučování dat pro analytické dotazy. Atributy by měly pro snadné porozumění obsahovat skutečná jména a ne uměle vytvořené zkratky. Pravidlem je co nejvíce omezit použití kódů v dimenzích. Výjimkou jsou kódy nebo identifikátory s důležitostí pro uživatele, kteří pomocí nich dále komunikují. Takové dimenze by však měly obsahovat i atribut s popisem.

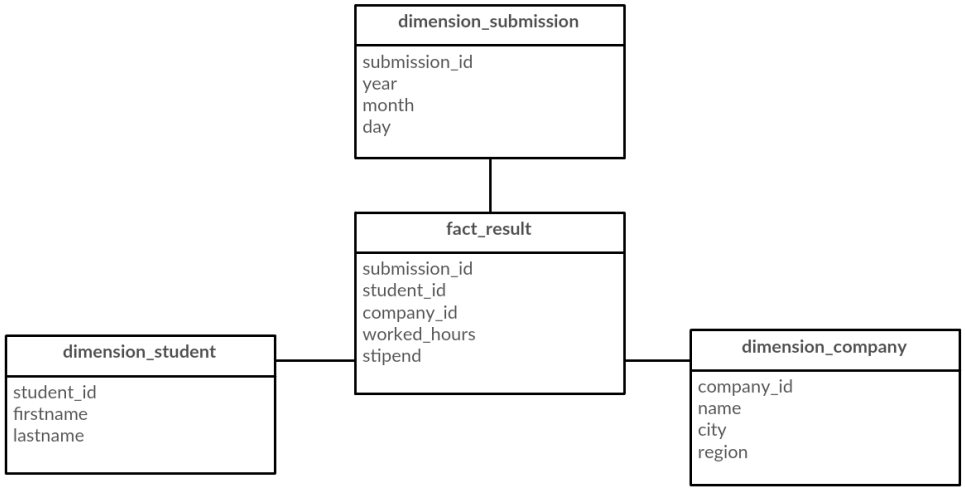
Může se stát, že při třídění faktů a dimenzí není jasné, kam atribut patří. Pokud nabývá velkého množství hodnot a podílí se na výpočtu, jedná se o fakt. Atribut s menším počtem diskrétních hodnot, který se podílí na omezení, patří do dimenzí.

2.2.2 Schémata

Jak bylo zmíněno výše, dva druhy tabulek v dimenzionálním modelu jsou faktové a dimenzionální tabulky. Ty dohromady vytváří strukturu, která je základem pro datová tržiště.

2.2.2.1 Hvězdicové schéma

Každý model je tvořen fakty, která jsou spojena s dimenzemi pomocí cizích klíčů. Této struktuře se říká hvězdicové schéma (angl. Star Schema), protože nápadně připomíná strukturu hvězdy. Samotné spojení tabulek se nazývá hvězdicové (angl. Star Join), jedná se o ustálený termín, který spadá do počátků relačních databází. Ukázka hvězdicového schématu je na obrázku 2.3.



Obrázek 2.3: Hvězdicové schéma

První zřejmou věcí, která stojí za povšimnutí, je jednoduchost takového schématu. To ocení obzvláště business uživatelé, pro které z toho vyplývá

snadné pochopení a navigace v datech. Uživatelé často ve schématu poznají klíčové prvky, které tvoří jejich business.

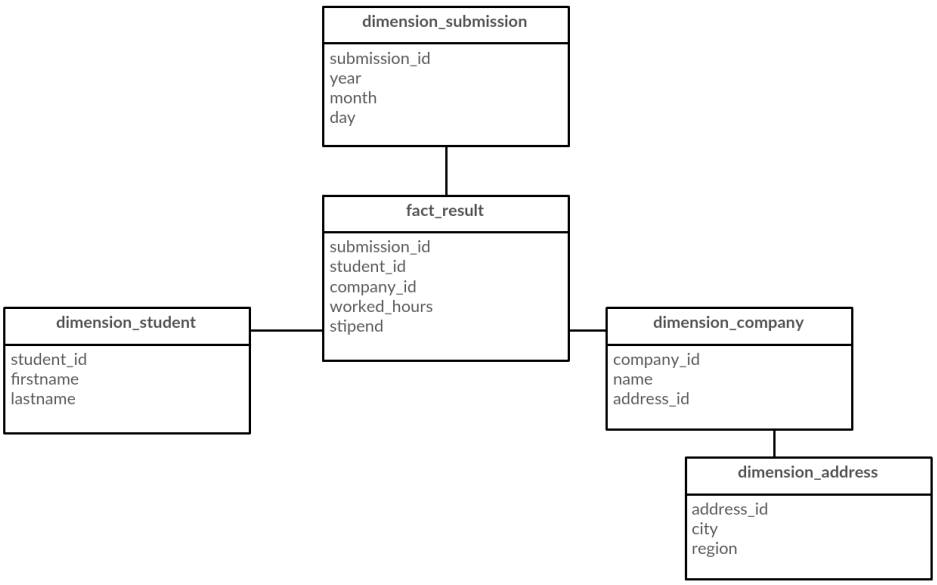
Jednoduchost modelu přináší i výkonnostní výhody. Databázový optimalizátor zpracuje tato schémata s menším počtem spojení než v normalizované formě, tedy rychleji a efektivněji. Nejprve může proběhnout značné omezení dimenzí a až jako druhý krok se vytvoří spojení dimenzí a faktové tabulky.

Další výhodou je i snadná rozšiřitelnost a reakce na změny. Všechny dimenze jsou rovnocenné vstupní body do faktové tabulky. Model je obecný a není přizpůsobený pro konkrétní dotazy. Nedochází tak k úpravě schématu, pokud se v budoucnu objeví nový analytický dotaz.

Největší dimenzionalitu tvoří atomická data s nejnižší granularitou. Neagregovaná data, která mají nejjemnější granularitu, mají nejvyšší expresivitu. Atomická data jsou základem pro návrh každého dimenzionálního modelu. Přidání dimenze je snadné, pokud jsou její hodnoty definované pro každou míru. Přidání dalších měření do faktové tabulky je také jednoduché, podmínkou je stejná úroveň detailu, jakou mají ostatní měření v tabulce.

2.2.2.2 Schéma sněhové vločky

Dimenzionální tabulky často uchovávají vztahy, které mají určitou hierarchii. Příkladem hierarchie může být lokace. Města spadají do okresů, ty do krajů apod. Tendence normalizovat takovou strukturu vede ke schématu, které se nazývá schéma sněhové vločky (angl. Snowflake Schema). Vychází z obyčejného hvězdicového schématu, dimenze však mohou být napojeny na další dimenze, ne pouze na faktové tabulky. Ukázka takového schématu je na obrázku 2.4.



Obrázek 2.4: Schéma sněhové vločky

Kimball tento přístup nedoporučuje [4]. Redundance dat, která nastane při použití hvězdicového schématu, nemá téměř žádný vliv na výkonnost. Je to z toho důvodu, že dimenzionální tabulky zabírají mnohem menší prostor než tabulky faktové. Tato normalizace snižuje přehlednost, jelikož zavádí zvláštní tabulky a nutnost napojení pomocí dalších umělých klíčů. Preferovaným způsobem je se této normalizaci vyhnout a využít tak jednoduchost, kterou představují data uložená v hvězdicové struktuře.

Kapitola 3

Integrace dat

Ať už se jedná o jakoukoliv architekturu, ve všech musí být provedena integrace dat mezi jednotlivými vrstvami dané architektury. Procesy, které data integrují, musí zajistit i uchování historických hodnot a souvisejících metadat. Těmito principy se zabývá právě tato kapitola.

3.1 Stage

Prvním krokem při integraci dat je jejich nahrání do prostředí datového skladu, konkrétně do stage. Data se mohou získávat pomocí ETL procesů přímo ze zdrojového systému, nebo vytvořením exportu dat, který se následně nahraje do stage. Za první způsob nese odpovědnost správce ETL procesů, tedy správce datového skladu, druhý nechává odpovědnost na zdrojovém systému. Z tohoto důvodu je pro tvůrce datového skladu lepší využít export dat s předem domluveným formátem. Existují tři typy exportu dat:

Full export Nejzákladnějším a nejjednodušším způsobem exportu je full export. Jak již název naznačuje, exportují se všechna data v nezměněné formě. Výhodou je, že je zachována kompletní kopie aktuálních dat a tím pádem je snadná jejich obnova. Nevýhodou je delší doba prováděné operace, požadavky na výkon a velikost úložiště.

Increment Výsledkem inkrementálního exportu jsou pouze data, která byla změněna od posledního exportu. Výhodou je rychlejší zpracování a menší množství kopírovaných dat. Nevýhodou je jejich složitější rekonstrukce.

Delta Diferenciální export probíhá velice podobně jako inkrementální. Opět se jedná o data, která byla změněna. Tentokrát se však změna váže na poslední full export. S opakováním diferenciálních exportů narůstá oproti inkrementálním jejich velikost.

Z výše uvedeného vyplývá, že základem je full export, který musí proběhnout alespoň jednou jako prvotní export. Pokud se full export bude provádět jednou týdně, nebo alespoň měsíčně, zmenší se tím velikost delta exportu a dříve se objeví případné chyby. Exporty mohou být brány z existujících databázových záloh. Hlavní zálohovací strategie jsou [21]:

- Full export denně
- Full export týdně + Delta každý den
- Full export týdně + Increment každý den

3.2 ETL

Základem integrace v datových skladech jsou ETL systémy. Ty se starají o extrakci dat ze zdrojového úložiště (E - extract), transformaci těchto dat (T - transform) a jejich nahrání do cílového úložiště (L - load). Zajišťují přesun z jednoho místa do jiného, vynucují kvalitu dat, jejich konzistenci a sdružují data z různých zdrojů.

ETL procesy nepatří do jedné vrstvy datového skladu, ale souvisí s manipulací dat mezi všemi vrstvami. V každé vrstvě proto mohou být různé. Liší se ve zdroji zpracovávaných dat, které mohou být strukturované nebo nestrukturované. Uložení může probíhat do normalizované či dimenzionální podoby. Dále mohou uchovávat historické hodnoty, nebo je nahrazovat nejnovějšími.

Kimball tvrdí [22], že ačkoliv tvorba ETL procesů není pro koncového uživatele viditelná, může spotřebovat i 70 % zdrojů potřebných nejen pro vytvoření, ale i pro údržbu datového skladu. Dále ukazuje, že ETL procesy přidávají datům další významné hodnoty, konkrétně:

- Odstraňují chyby a opravují chybějící data.
- Poskytují zdokumentovaná měřítka kvality dat.
- Zaznamenávají tok transakčních dat.
- Přizpůsobují a sdružují data z různých zdrojů.
- Strukturují data k využití v dalších vrstvách datového skladu.

Extract V tradičním ETL systému je extrakce prvním krokem integračního procesu. Proces extrakce nejdříve prozkoumá zdroje dat a rozhodne, která jsou vhodná pro zařazení do dalšího procesu. Pro snížení náročnosti celkové integrace izoluje extrakce pouze změny, které ve zdrojových systémech nastaly. Poslední fází je přesun dat do prostředí datového skladu, kde se uskuteční další zpracování.

Transform Dalším krokem je transformace získaných dat. V průběhu transformace vznikají metadata, která slouží k diagnostikování problémů s daty ze zdrojových systémů. Součástí procesu je i ošetření kvality dat,

zachycení chyb, jejichž analýza může vést ke zlepšení kvality dat. Transformace eliminuje zbytečná data, sdružuje je z několika zdrojových systémů a odstraňuje možné duplicity, které vznikají právě odpovídajícími daty z různých zdrojů.

Load Poslední krok slouží k uložení dat do cílového systému. Tato fáze se stará o fyzickou strukturalizaci dat. Vytváří umělé klíče, které slouží k identifikaci a spojení jednotlivých tabulek, určují tím i jejich hierarchii. Dále zajišťuje logiku pro pomalu se měnící dimenze (více v sekci 3.3).

Datový sklad musí být spolehlivým zdrojem pro podporu rozhodování. K dosažení tohoto cíle jsou na ETL systém kladeny následující požadavky [23]:

- **Spolehlivost** ETL procesy musí běžet konzistentně, aby poskytovaly důvěryhodná a detailní data včas.
- **Dostupnost** Datový sklad by měl být dostupný dle určených požadavků. ETL procesy tedy nesmí nijak narušit jeho chod.
- **Reakce na změny** Datový sklad se neustále mění a spolu s ním se vyvíjí i ETL procesy.

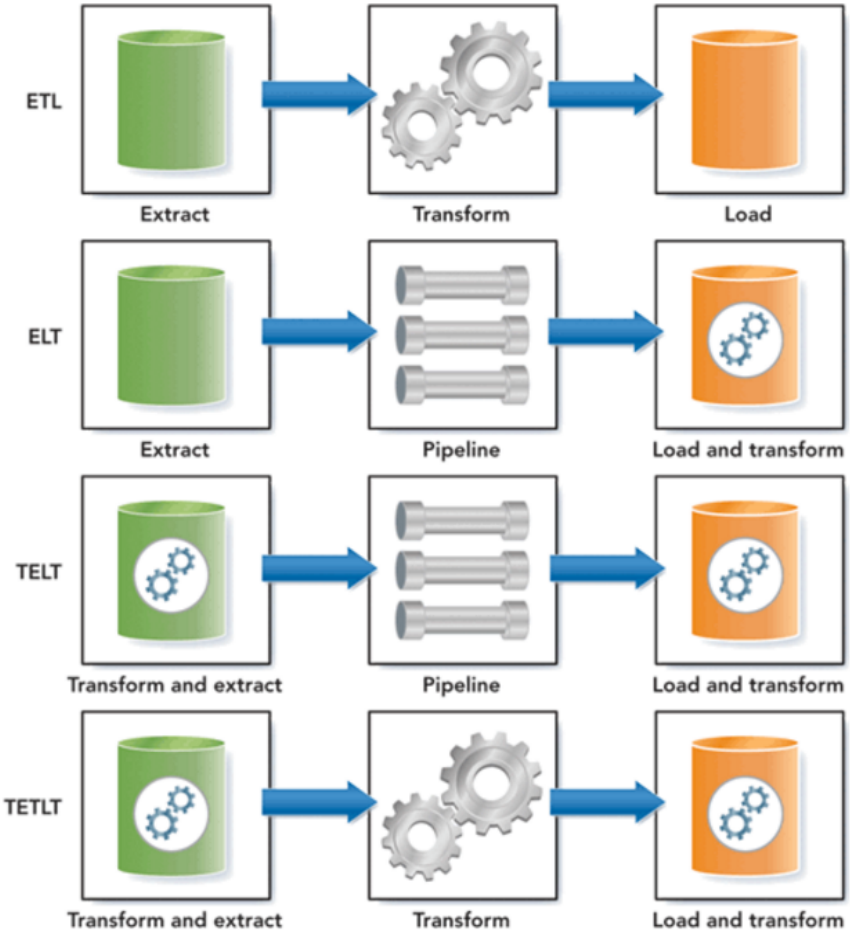
3.2.1 Další varianty

ETL nejsou jedinou možností pro integraci dat. Dalším používaným způsobem jsou ELT procesy. Data se nejdříve ze zdroje načtou, následně se uloží do cílového úložiště. Až v tomto úložišti dochází k jejich transformaci a souvisejícímu zpracování.

Výhody ELT oproti ETL mohou být např. v ceně. Software licence a vývoj mohou výrazně překročit cenu infrastruktury. Tento rozdíl je navýšen dostupností levných vícejádrových databázových serverů a spolu s virtualizací a službami v cloudu, které se dají využít k procesům transformace, může být výhodnější použít právě ELT. ETL často požaduje investice mimo databázové systémy, ELT na druhou stranu předpokládá silnou znalost databázových technik.

Rozhodujícími faktory však nemusí být pouze cena, ale požadavky na funkcionalitu, úložný prostor a výkonnost. V potaz musí být brána celková organizace související se soukromím a bezpečností dat a neméně podstatnou složkou je kapacita síťového provozu. V některých případech tyto faktory zvýhodňují zaběhlejší ETL systémy [24].

Existují i hybridní systémy, které mohou používat několik transformací. Příklady jsou TELT, ETLT a TETLT. Některé z těchto metod jsou ilustrovány na obrázku 3.1.



Obrázek 3.1: ETL a další alternativy, převzato z [6]

3.3 Pomalu se měnící dimenze

Data, která jsou integrována do datového skladu, nejsou stálá a časem se mění. Historické hodnoty se mohou využívat pro různé analýzy, a proto by měl datový sklad na změny v datech reagovat. Mechanizmus, který se v datových skladech nejčastěji využívá pro zaznamenávání historie, se nazývá pomalu se měnící dimenze (angl. Slowly Changing Dimensions - SCD) [25].

Existují různé druhy SCD, zažitým označením pro ně je zkratka SCD a číslo typu. Nejčastějšími typy jsou podle Kimballa SCD1, SCD2 a SCD3 [26], [27]. Existují však i další typy, které jsou odvozené ze zmíněných základních typů. Jedná se o typy SCD4 až SCD7 [28].

Každý typ udává, co se děje s daty, pokud nastane jedna z operací vložení, úpravy a smazání. Podle Kimballa [4] by měl mít každý atribut definovanou

strategii pro řešení historizace. Od toho se odvíjí, že v jedné tabulce může existovat více různých druhů historizace, pro každý atribut jiná.

3.3.1 SCD1

Typ 1 využívá přepsání související hodnoty či smazání záznamu. Atribut udržuje vždy pouze nejnovější hodnotu. Jedná se o nejjednodušší přístup ke změně atributu. Tento typ najde využití např. při úpravě chybně zapsaných údajů. Dalším důvodem pro SCD1 může být neexistence původní hodnoty. Ukázka je v tabulce 3.1.

Tabulka 3.1: Ukázka SCD1

Id	**Název předmětu**	**Počet kreditů**
123	MI-EDW	4

Změna počtu kreditů ze 4 na 5.

Id	**Název předmětu**	**Počet kreditů**
123	MI-EDW	5

Problémem je, že jsou ztracena veškerá historická data. Analytické nástroje, které předtím odkazovaly na staré hodnoty, nyní vrací jiné výsledky. Pokud existují agregované údaje pro starou hodnotu, tyto agregace musí být vytvořeny znovu. Jestliže je implementována historizace pomocí SCD1, je složité v budoucnu změnit tento typ historizace na jiný.

3.3.2 SCD2

SCD2 je nejvíce rozšířený typ pomalu se měnících dimenzí. Zároveň je také nejbezpečnější, pokud není jisté, který typ vybrat [4]. Jeho principem je přidání nového řádku pro každou změnu.

Pro SCD2 není vhodné používat přirozené primární klíče, protože je může mít každý zdrojový systém v jiném formátu. Spolu s umělým klíčem přibývají i další hodnoty. Jedná se o počátek a konec platnosti záznamu. Z důvodu efektivity se může přidat atribut, který ukazuje, jestli je záznam platný, či ne. Dalším přidaným atributem může být popis změny. Ukázka změny pomocí SCD2 je v tabulce 3.2.

Jelikož se v databázi může vyskytovat několik záznamů, které ve skutečnosti tvoří jeden s historickými hodnotami, musí se upravit primární klíč. Stačí, když se jako primární klíč zvolí kombinace původního klíče a atribut počátku platnosti záznamu. Pro aktivní záznamy, které nejsou ukončené, se nastavuje datum konce platnosti na dohodnuté datum ve vzdálené budoucnosti. Pro každý typ operace se provádí historizace různě.

Tabulka 3.2: Ukázka SCD2

Id	**Název předmětu**	**Počet kreditů**	**BEGIN_-EFF_-DATE**	**END_-EFF_-DATE**	**ACTUAL_-FLAG**
123	MI-EDW	4	1. 9. 2015	1. 1. 2999	1

Změna počtu kreditů ze 4 na 5 dne 1. 9. 2016.

Id	**Název předmětu**	**Počet kreditů**	**BEGIN_-EFF_-DATE**	**END_-EFF_-DATE**	**ACTUAL_-FLAG**
123	MI-EDW	4	1. 9. 2015	1. 9. 2016	0
123	MI-EDW	5	1. 9. 2016	1. 1. 2999	1

Přidání záznamu

- vytvoření nového záznamu se složeným primárním klíčem z původního a hodnoty BEGIN_EFF_DATE
- BEGIN_EFF_DATE se nastaví na aktuální datum
- END_EFF_DATE se nastaví na vzdálené datum v budoucnu (často 1. 1. 2999 nebo 31. 12. 2999)
- ACTUAL_FLAG se nastaví na 1

Úprava záznamu

- úprava END_EFF_DATE u posledního řádku původního záznamu na aktuální datum
- úprava ACTUAL_FLAG u posledního řádku původního záznamu na 0
- vytvoření nového záznamu se složeným primárním klíčem z původního a hodnoty BEGIN_EFF_DATE
- BEGIN_EFF_DATE se nastaví na aktuální datum
- END_EFF_DATE se nastaví na vzdálené datum v budoucnu (často 1. 1. 2999 nebo 31. 12. 2999)
- ACTUAL_FLAG se nastaví na 1

Smazání záznamu

- END_EFF_DATE se nastaví na aktuální datum
- ACTUAL_FLAG se nastaví na 0

3.3.3 SCD3

Ačkoliv SCD2 uchovává plnou historii změn, neumožňuje přidružit novou hodnotu ke staré a naopak. Občas je však potřeba vidět obě hodnoty. To zajišťuje

SCD3. V porovnání s předchozími dvěma technikami se jedná o tu nejméně používanou.

Podobně jako SCD2 přidává atribut k záznamu. Tento atribut obsahuje starou hodnotu, nová se napíše do originálního sloupce. Takto se může uchovávat hodnota předchozí, nebo počáteční. Alternativně se může přidat více sloupců, čímž je možné uchovávat více historických hodnot. Tabulka 3.3 demonstruje použití SCD3.

Tabulka 3.3: Ukázka SCD3

Id	**Název předmětu**	**Počet kreditů**	**Předchozí počet kreditů**
123	MI-EDW	4	NULL

Změna počtu kreditů ze 4 na 5.

Id	**Název předmětu**	**Počet kreditů**	**Předchozí počet kreditů**
123	MI-EDW	5	4

3.4 Metadata

V souvislosti s ETL procesy, o kterých pojednávala sekce 3.2, byla zmíněna metadata. Častou definicí metadat je: *„Metadata jsou data o datech.“*. To však není zcela přesné. Do části metadat samozřejmě spadají data, která popisují business data, ať už se jedná o jejich business souvislosti, či technické vlastnosti. V prostředí datových skladů sem ale patří i data o procesech, jejichž součástí jsou chybové hlášky nebo informace o výkonnosti těchto procesů. Metadata se tedy dají dělit na business, technická a procesní [22].

Business metadata Jedná se o metadata, která datům dávají business význam. Patří sem obchodní definice, které přiřazují datům obchodní termíny společnosti. Tyto definice se nachází v datových slovnících. Dalším typem jsou informace o zdrojových systémech. Ty obsahují business popis zdrojových systémů a pravidel business procesů. Slouží k tvorbě ETL procesů. Spadají sem i logické datové mapy (angl. Logical Data Maps), které se často označují anglickým termínem Data Lineage. Skládají se z mapování, které logicky vysvětluje, co se stalo s daty od doby, kdy byla extrahovaná ze zdrojového systému, až do jejich integrace v poslední vrstvě datového skladu.

Technická metadata Technická metadata popisují fyzické atributy dat, jakými jsou např. struktura, formát a umístění. Patří sem systémový inventář, který obsahuje technická metadata o všech systémech, které jsou součástí datového skladu. Uchovává tabulky, sloupce, datové typy

a vztahy mezi tabulkami. Další součástí jsou různé datové modely, které mohou mít grafickou podobu. Business pravidla by mohla být součástí business metadat, jelikož jsou ale esenciální pro ETL procesy z technického hlediska, spadají do technických metadat. Business pravidlem může být sada povolených hodnot nebo výpočet hodnot odvozených.

Procesní metadata Při ETL procesech vznikají procesní metadata. Popisují běh a výsledky těchto procesů pomocí různých statistik. Uchovávají, kolik řádků bylo zpracováno v průběhu jobu, kdy tento job začal, kdy skončil, rychlost načítání apod. Dále sem patří informace o výjimkách, které v průběhu vykonávání jobu nastaly, a způsobu jejich řešení.

Část II

Praktická část

Kapitola 4

Analýza

Na Fakultě informačních technologií ČVUT existuje Portál spolupráce s průmyslem. Umožňuje studentům pracovat na reálných projektech, díky tomu je vhodným kandidátem pro poskytování zajímavých dat, jejichž analýza může pomoci ke zkvalitnění výuky. Integrováním do již fungujícího datového skladu se otevře možnost vytváření těchto analýz. Jelikož jsou oba systémy navržené a fungují, prvním krokem k integraci SSP do datového skladu je analýza obou systémů a identifikace potřebných dat.

4.1 Portál spolupráce s průmyslem

SSP je webová aplikace, která umožňuje propojení akademické a komerční sféry. Portál umožňuje průmyslovým partnerům vkládat zadání, která mají praktický základ. Zadání jsou následně schvalována pedagogy a vypsané úlohy jsou řešeny studenty, ať už se jedná o jednotlivce nebo celý tým.

Mezi přednosti systému patří chytré algoritmy, jež zajišťují doporučení vhodných kandidátů pro řešení. Posuzování je založeno na ohodnocení studentů, které vzniká ze studijních výsledků a z předešlých spoluprací. Ohodnocení studenta motivuje k získání lepších výsledků a současně vylepšení jeho profilu, což spadá do techniky gamifikace [29].

Cílem je tedy umožnit efektivní a prospěšnou spolupráci mezi průmyslovými partnery a univerzitou. Hlavním zaměřením jsou bezesporu studenti. Mnoho studentů fakulty v průběhu studia pracuje na částečný úvazek, tato práce však ve většině případů není provázána se samotným studiem. To se snaží řešit SPP tím, že umožňuje studentům pracovat na reálných projektech, které mohou být uznány jako bakalářské, diplomové, či semestrální práce do vyučovaných předmětů a jsou finančně ohodnoceny. Tato spolupráce je výhodná i pro fakultu, která získává zpětnou vazbu důležitou pro inovaci předmětů [30].

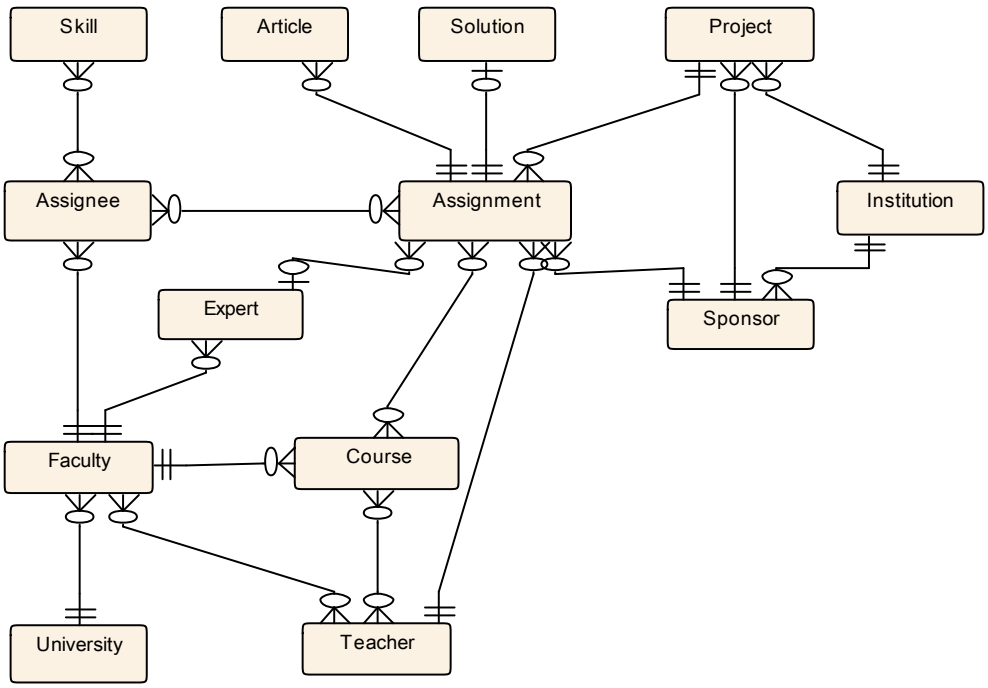
4.1.1 Zdrojové systémy

Data pro SSP pocházejí ze dvou zdrojových databází. První z nich je základní OLTP databáze, nad kterou probíhají operace jako je registrování partnerů a studentů, vkládání nových zadání, nahrávání výsledků, hodnocení studentů za spolupráci apod. Všechny uvedené operace se ihned po jejich realizaci projeví v databázi. Hlavním cílem je udržení pravidelného chodu aplikace.

Druhý zdroj pak tvoří podpůrná databáze pro business intelligence funkce v rámci SSP. Tato databáze slouží hlavně k propočítávání dovedností studenta. Udržuje jeho rozšířený profil, kde je poznat, jakou dovednost a v jaké kvalitě ovládá. Tyto hodnoty zadavatelé nevidí, slouží vnitřně pro vyhledávání podobností mezi studenty a pro doporučení studentů na konkrétní zadání. Pro průmyslové partnery je pak díky těmto doporučením jednodušší nalézt správného řešitele pro danou úlohu. Konkrétními způsoby doporučování v SSP se zabývá [31] a [32].

4.1.2 Entity v SSP

Tato sekce se zabývá entitami, se kterými pracuje SSP. Dané entity jsou pak potencionálními kandidáty pro integrování do datového skladu. Entitní model je načrtnut na obrázku 4.1. Popis entit je shrnut v následujících odstavcích.



Obrázek 4.1: Entity v SSP

Assignee Jedná se o základní entitu, která v určité formě existuje i v datovém skladu. Assignee je student, který řeší nebo chce řešit nějaké zadaní z portálu. Identifikován je umělým klíčem, uchovává však uživatelské jméno, které je jednoznačné v rámci celého ČVUT. Jeho profil může být veřejný pro ostatní řešitele nebo pro zadavatele. Uživatel může zadat svůj popis nebo k profilu připojit štítky, to vše ho blíže charakterizuje. Pro zadavatele jsou zajisté atraktivní jeho dovednosti, které jsou číselně ohodnocené, a absolvované předměty spolu s jejich klasifikací. Z těch se mimo jiné dovednosti vypočítávají. Autor se však může ohodnotit i subjektivně. S řešitelem jsou spjatá i ohodnocení od průmyslových partnerů. Uchazeč před samotným řešením zadání prochází různými stavy, sponzor ho totiž musí nejdříve schválit jako řešitele. Teprve poté může začít pracovat na úloze.

Assignment Hlavním účelem portálu je pomoci studentům získat zkušenosti z reálných projektů. Entita Assignment obsahuje informace o zadáních pro takové projekty. Pro potencionální řešitele jsou důležité informace jako popis zadání spolu s cíli a požadovanými výstupy, termín odevzdání, odhadovaná náročnost a odměna za projekt. Hotové řešení pak lze odevzdat do předmětů, které se zadáním souvisí. Jednodušší vyhledávání odpovídajících druhů zadání umožňují i tagy. K zadání patří požadavky na typy řešitelů, jelikož se nemusí jednat o programátory, ale např. o datové analytiky. Zadání u sebe uchovává i odkaz na licenci, do které spadá vytvořené řešení. Pro ujasnění požadavků je možné pokládát dotazy, na které zadavatelé odpovídají. Podobně jako řešitel i zadání má stavy, kterými musí projít pro jeho úspěšné zakončení.

Article Pro zvýšení atraktivity spolupráce studentů s průmyslem slouží články. Patří k určitým zadáním, mohou mít několik verzí. Články se mohou nacházet v jednom z předem definovaných stavů a je možné k nim vkládat komentáře.

Course Projekty, které se nachází v systému, mohou být uznány jako semestrální práce do předmětů vyučovaných na fakultě. Schvalování předmětů k zadáním provádí učitel, který daný předmět učí. Informace o předmětech poskytuje právě entita Course. Není to však její jediný cíl. Jak bylo uvedeno v souvislosti s řešitelem, z absolvovaných předmětů se vypočítávají dovednosti. Předměty se ukazují např. i u univerzitních expertů, kteří je vyučují. Podobně jako student prochází různými stavy, než může pracovat na projektu, i vztah předmětu a zadání se nachází v jednom z několika definovaných stavů. Entita uchovává i základní informace o předmětu, jakými jsou kód předmětu, počet kreditů, jméno, anotace a sylabus. K předmětu se mohou vztahovat i štítky.

Expert Průmysloví partneři mohou delegovat komunikaci se studentem na univerzitního experta. Ve většině případů se jedná o odborné pomocníky z fakulty, kde vyučují různé předměty, které jsou ukázány na jejich profilu. Mají často bližší vztah se studenty a ví, jaké jsou požadavky v jednotlivých předmětech. Pomáhají díky tomu vytvořit a spravovat zadání tak, aby bylo pro studenty atraktivní. Mohou nominovat související předměty, vybírat vhodné kandidáty, sestavovat řešitelský tým a kontrolovat kvalitu řešení. Finální akceptování řešení je však na zadavateli. Profil experta může být opět blíže charakterizován štítky a textovým popisem.

Institution Průmysloví partneři jsou ti, co zadání tvoří a umožňují tak studentům získávat praxi a zkušenosti z reálných projektů. Institution je firma, která pro studenty tyto projekty realizuje. Díky jejich zadáním je možné, že student bude s firmou spolupracovat i po skončení projektu. Průmyslová instituce může být podobně jako jiné entity opatřená tagy.

Faculty Aktuálně je podporovaná pouze Fakulta informačních technologií ČVUT. SSP má však ambice rozšířit svoji působnost i na další fakulty. K tomu je přizpůsobený datový model, kde se mohou uchovávat informace o dalších fakultách. Nabízí jméno fakulty, její zkratku, popis, štítky a kontakty.

University SSP nemusí zůstat pouze u různých fakult ČVUT. Datový model počítá i s možným rozšířením do dalších univerzit. To by pomohlo jiným univerzitám k začlenění studentů do reálných projektů, stejně jako to umožňuje FIT.

Project Projekty sdružují jednotlivá zadání. Mohou tvořit hierarchickou strukturu. Obsahují popis a název, samozřejmostí je přítomnost štítků. Projekty patří k určité instituci, která má účet v systému. Manažer projektu je konkrétní zaměstnanec firmy.

Skill Uchovává informace o dovednostech studenta. Mohou mít hierarchickou strukturu, obsahují název a popis. Existují dva typy ohodnocení dovedností. Prvním je garantované ohodnocení, to se vypočítává z absolvovaných předmětů a hodnocení od zadavatele a učitele za odevzdaná řešení. Druhým je subjektivní ohodnocení, které student vytváří podle svého svědomí. Dovednost je jediná entita, jejíž podstatné části jsou uloženy v obou zdrojových databázích. Část v hlavní databázi slouží pro zadavatele, nastavují jimi požadavky na řešitele a následně je jimi hodnotí. Dovednosti v podpůrné databázi slouží k doporučování studentů na konkrétní zadání.

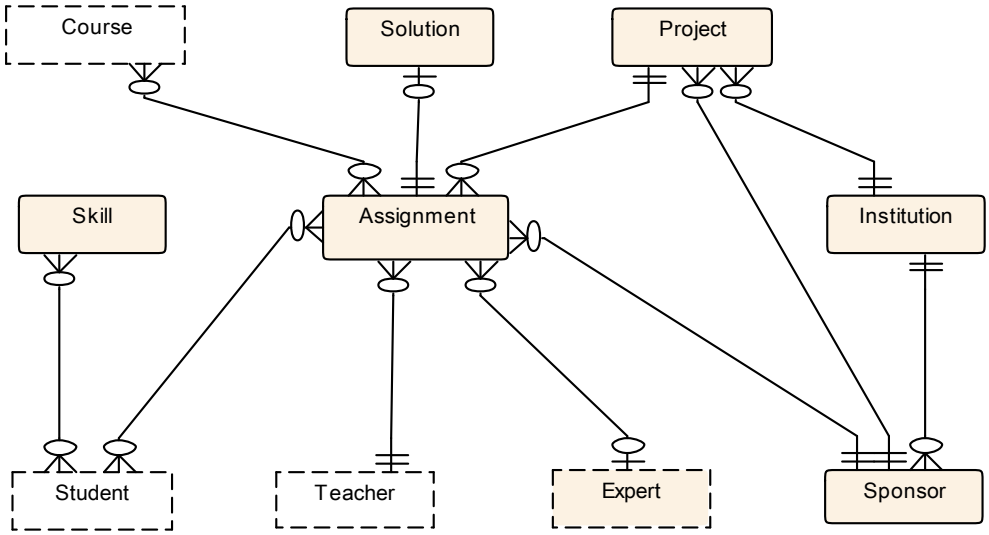
Solution Výstupem studenta nebo týmu studentů je řešení, které odpovídá požadavkům v zadání. Zadavatel musí výstup schválit, prochází tedy předem definovanými stavy. Řešení, podobně jako spolupráce, má svoje hodnocení.

Sponsor Ačkoliv hlavním poskytovatelem zadání jsou firmy, průběžné konzultace a dotazy jsou realizovány se zaměstnanci dané firmy. Sponzor je tedy konkrétní člověk, který za danou instituci komunikuje, zakládá a spravuje zadání. Instituce může mít několik sponzorů, kde každý má svá zadání. Zadavatel hodnotí spolupráci s řešitelem a za svou spolupráci je ohodnocen i on od studenta. Není překvapující, že i ke sponzorovi patří různé tagy.

Teacher Entita existuje hlavně kvůli schvalování nominovaných předmětů, do kterých je možné odevzdat řešení konkrétního zadání. Jelikož často uznává práci do svého předmětu, konzultuje průběžný postup se studentem.

4.1.3 Entity pro datový sklad

Ne všechny entity v předchozí sekci má smysl uchovávat v datovém skladu. Byly identifikovány ty nejdůležitější, které se do skladu dostanou. Některé navíc v datovém skladu již existují (více v sekci 4.2) a entity z SSP se na ně musí pouze napojit. Ukázka podstatných entit a jejich napojení na již existující je na obrázku 4.2.



Obrázek 4.2: Entity SSP pro datový sklad

Student, učitel i předmět jsou bez výplně ohraničeni čárkovaně, jelikož jsou již do skladu integrovaní z jiných zdrojových systémů. Návaznost ostatních entit na ně ale musí být zachována, proto jsou v diagramu načrtnuty. Článek je entita, která se nebude integrovat vůbec. Klíčové entity a vztahy mezi nimi jsou student a jeho dovednosti, projekty a jejich zadání spolu s řešiteli, souvisejícími předměty a učiteli, které takové zadání do předmětu schvalují. Současně je podstatné uchovávat experta, tedy osobu pomáhající řešiteli se zadáním. K uchování kontextu pro projekty a zadání jsou důležití i průmysloví partneři a jejich zaměstnanci. Jelikož mohou existovat i zadání, která nebyla úspěšná, informace o řešení byly také identifikované jako vhodné pro uchování.

4.2 Datový sklad ČVUT

Na ČVUT existuje datový sklad, který sdružuje data z několika různých univerzitních systémů, mezi které patří např. KOS, přihlášky, závěrečné práce a anketa. Architektura skladu odpovídá Inmonově architektuře. To znamená, že existuje centralizované úložiště, které je v normalizované formě. Pro integraci dat je nezbytné analyzovat struktury v této integrované vrstvě.

4.2.1 Vrstvy datového skladu

Jak již bylo zmíněno, datový sklad vychází z Inmonovy architektury a obsahuje odpovídající vrstvy [9]. Stage, neboli dočasné úložiště dat, je první vrstvou skladu. Integrovaná vrstva, které se v prostředí datového skladu ČVUT neformálně říká target a odpovídá EDW v Inmonově architektuře, je ve 3NF. Další vrstvou je přístupová vrstva, která má dvě podvrstvy. První je sémantická vrstva s tzv. stavebními bloky (angl. Building Blocks), které mají funkci jako sémantická databáze (sekce 1.4.3) nebo Conformed Dimensions (sekce 1.2.3). Stavební bloky jsou tvořeny obyčejnými databázovými pohledy. Tato vrstva ve skladu ještě fyzicky není, její tvorba probíhá v paralelně vznikající diplomové práci [10]. Druhou část přístupové vrstvy tvoří datová tržiště, která čerpají ze stavebních bloků a oproti nim je vytvořena pomocí materializovaných pohledů. Jako databáze je aktuálně využit PostgreSQL.

4.2.2 Entity v datovém skladu

Datový sklad integruje několik zdrojů, s čímž souvisí i entity z nich pocházející. V době provádění analýzy byly základní entity v datovém skladu následující:

- Organizační jednotka
- Osoba
- Učitel
- Předmět
- Zapsaný předmět

- Studijní plán
- Studium
- Akademický rok
- Přihláška
- Státní závěrečná zkouška
- Akce
- Paralelka
- Místnost

Organizační jednotka, osoba, učitel a předmět jsou entity, které mají SSP a datový sklad alespoň částečně společné. Proto jsou více rozvedeny pouze tyto entity.

Organizační jednotka Organizační jednotka v kontextu SSP odpovídá fakultě a univerzitě. V datovém skladu jsou tyto entity namodelovány obecněji a tvoří hierarchickou strukturu. Společnými prvky jsou jméno a zkratka jednotky.

Předmět Entita předmět je v datovém skladu poněkud rozsáhlá. Je to logické, předmět je základní stavební jednotkou celého studia. Má návaznost na studijní plány, paralelku, klasifikaci, související předměty apod. Oba systémy uchovávají kód předmětu a jeho návaznost na organizační jednotku, což stačí na jednoznačné určení předmětu. Uchovávají i počet kreditů, popis předmětu a studentovu klasifikaci. Ačkoliv má SSP struktury pro tyto informace také, není jejich zdrojem.

Osoba Odpovídající entity v prostředí SSP mohou být řešitel, učitel, univerzitní expert a sponzor. Návaznost může proběhnout podle uživatelského jména. Teoreticky je možné, že jedna osoba má roli všech entit, tedy studuje a pracuje na projektu, učí nějaký předmět, slouží jako expert, který radí nějakým sponzorům a současně může vypisovat zadání a je tedy sponzorem. Nemusí to být pouze v rámci jednoho semestru, ale i v rozmezí několika let.

Učitel Učitel je navázán na osobu, identifikace bude probíhat stejně, tedy podle unikátního uživatelského jména. Relativní entitou v SSP je učitel. Expert a sponzor sice mohou být učitelé, není to však podmínka.

4.2.3 Jmenná konvence datového skladu

Datový sklad používá pro jednoduchost stejnou jmennou konvenci pro všechny struktury [10]. Popsány byly prvky, které ovlivňují integraci SSP. Obecně se v modelech používají české názvy, jednotné číslo a velká písmena. Pro oddělení slov slouží podtržítko.

4.2.3.1 Tabulka

Název tabulky je ve formě *[prefix]_[domena]_NAZEV_TABULKY_[suffix]*. Prefix může být:

- T - klasická tabulka
- MV - materializovaný pohled
- V - pohled

Suffix slouží pro dimenzionální schéma a nabývá následujících hodnot:

- DIM - označení dimenzionální tabulky
- FACT - označení faktové tabulky

4.2.3.2 Atribut

Struktura názvu atributu je *[prefix]_NAZEV_ATRIBUTU_[suffix]*. Jako prefix se uvádí FK, pokud se jedná o cizí klíč. Možná suffixy jsou následující:

- BK - businessový klíč používaný ve zdrojovém systému (business PK tabulky)
- TK - technický klíč vygenerovaný v rámci datového skladu sekvencí (PK tabulky)
- ID - jednoznačný identifikátor, který identifikuje řádek v tabulce (nejedná se o business klíč používaný v rámci businessové domény)

4.2.3.3 Vztah

Vztahy mezi tabulkami jsou řešeny pouze pomocí businessových klíčů. To je důležité z hlediska historizace. Cizí klíč má tvar *FK_TABULKAODKUD_[suffix]*, kde suffix může být BK, pokud je cizí klíč zároveň businessovým klíčem.

Vazební tabulka pro vztahy M:N je pojmenována podle specifikace *[prefix]_TABULKAODKUD_TABULKAKAM_REL*. Prefix je stejný jako pro ostatní tabulky. Atributy tabulky se skládají ze dvou cizích klíčů, které jsou současně i BK.

Pro ostatní není určena zvláštní jmenná konvence. Používá se pouze zavedená konvence pro cizí klíče.

4.2.3.4 Technické atributy

Technické atributy skladu jsou strojově přidány do každé tabulky schématu.

- DATE_TO (TIMESTAMP) - platnost záznamu do
- DATE_FROM (TIMESTAMP)- platnost záznamu od
- LAST_UPDATE (TIMESTAMP) - datum poslední změny
- NAZEV_TABULKY_TK (BIGINT) - technický klíč záznamu (sekvence)
- VERSION (BIGINT) - verze záznamu

Kapitola 5

Návrh

Jak bylo zmíněno v analýze, datový sklad na ČVUT již existuje. Cílem je do něj SSP integrovat, odpadá nutnost vytváření skladu od začátku. Nemusí být diskutovány výhody jednotlivých architektur, které byly popsány v teoretické části, rozhodnutí o použité architektuře bylo vykonáno v minulosti.

Analýza 4.2 ukázala, že sklad vychází z Inmonovy architektury (původní verze odpovídala Kimballově architektuře) a přístupová vrstva obsahuje sémantickou podvrstvu. Návrh se tedy skládá ze tří hlavních částí. První je návrh stage, druhou je návrh normalizovaných struktur integrované vrstvy, třetí tvoří přístupová vrstva skládající se ze sémantické vrstvy se stavebními bloky a z datových tržišť.

5.1 Stage

V sekci 3.1 byly uvedeny tři základní typy databázového exportu, který lze využít pro tvorbu stage. V tomto případě byl zvolen dump celé databáze, tedy full export.

Výhoda spočívá v jednoduchosti při tvorbě exportů, nemusí se vyhledávat pouze změny. Použití databázového exportu místo ETL je navíc výhodnější v tom, že v případě budoucích požadavků na integraci dalších dat, jejichž struktury v databázi již jsou nebo přibudou, nebudou ovlivněny transformace mezi zdrojovým systémem a stage. Pouze se rozšíří struktura integrované vrstvy a vytvoří odpovídající ETL mezi ní a stage. Pokud by totiž data byla nahrávána do stage z SSP pomocí ETL procesů, musely by se při jakémkoliv požadavku upravovat i tyto procesy.

Problém však nastane, pokud se výrazně změní struktura zdrojového systému. Existující integrace může přestat fungovat. V takovém případě je však možné, že se tyto změny budou chtít zpropagovat do datového skladu, ETL by se tedy upravovaly v každém případě. Oba systémy jsou navíc v současné době spravovány na Fakultě informačních technologií, změna struktury může být díky tomu vyřešena v relativně krátkém časovém intervalu.

5.2 Integrovaná vrstva

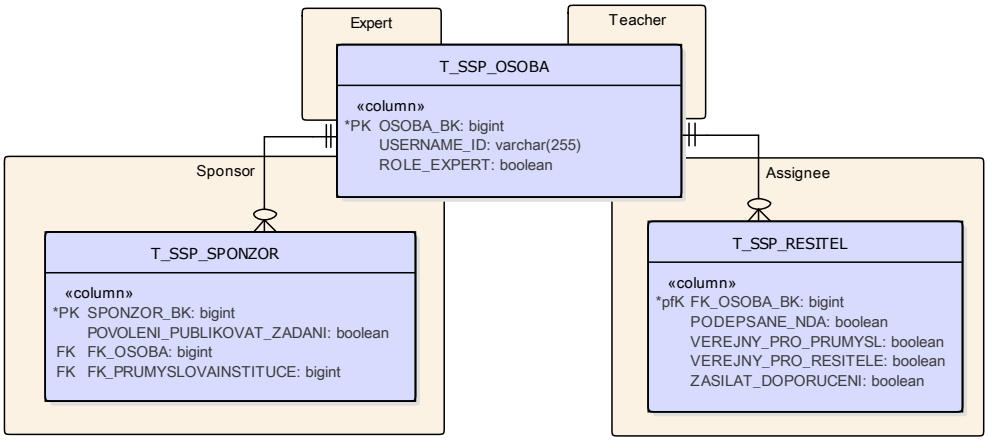
Dalším krokem návrhu v architektuře dle Inmona, tedy v architektuře, která má vrstvu s jednou verzí pravdy, je navržení právě takové vrstvy. Jedná se o integrovanou vrstvu, která v existujícím skladu již sdružuje některé zdrojové systémy.

Analýza 4.1.3 identifikovala entity ze zdrojového systému SSP, které jsou vhodné pro uchování v datovém skladu. Podstatou datového skladu ČVUT je udržovat informace související s univerzitou jako takovou. Ať už se jedná o informace důležité pro její chod, či informace, které mohou zlepšit kvalitu výuky. Hlavními entitami tedy jsou Assignee, Assignment, Skill, Expert a jejich návaznosti na studenty, učitele a předměty. V následujících sekcích jsou detailně rozebrány struktury těchto entit a jejich souvisejících vazeb. Návrh používá jmennou konvenci, která byla ukázána v sekci 4.2.3. Diagram celého schématu integrované vrstvy je v příloze C. Mapování mezi zdrojovým systémem a integrovanou vrstvou ukazuje příloha D.

5.2.1 SSP user

Všechny osoby v SSP, konkrétně řešitele, sponzora, učitele a experta, v datovém skladu sjednocuje jedna tabulka. V ní je uvedeno uživatelské jméno osoby. Pokud se jedná o studenta nebo učitele, jejich uživatelské jméno je to, které se používá v rámci ČVUT. Napojení na existující studenty a učitele může být realizováno právě přes tato jména.

Napojení přes uživatelské jméno není fyzické pomocí referenční integrity, ale logické. Alternativní spojení by mohlo probíhat přes business klíč osoby v datovém skladu. Jelikož ale někteří uživatelé SSP nemusí být existující osoby v datovém skladu, musel by být tento atribut nepovinný. Ve skladu se navíc všechny cizí klíče zahazují, rozdíl může nastat pouze ve výkonnosti vytváře-

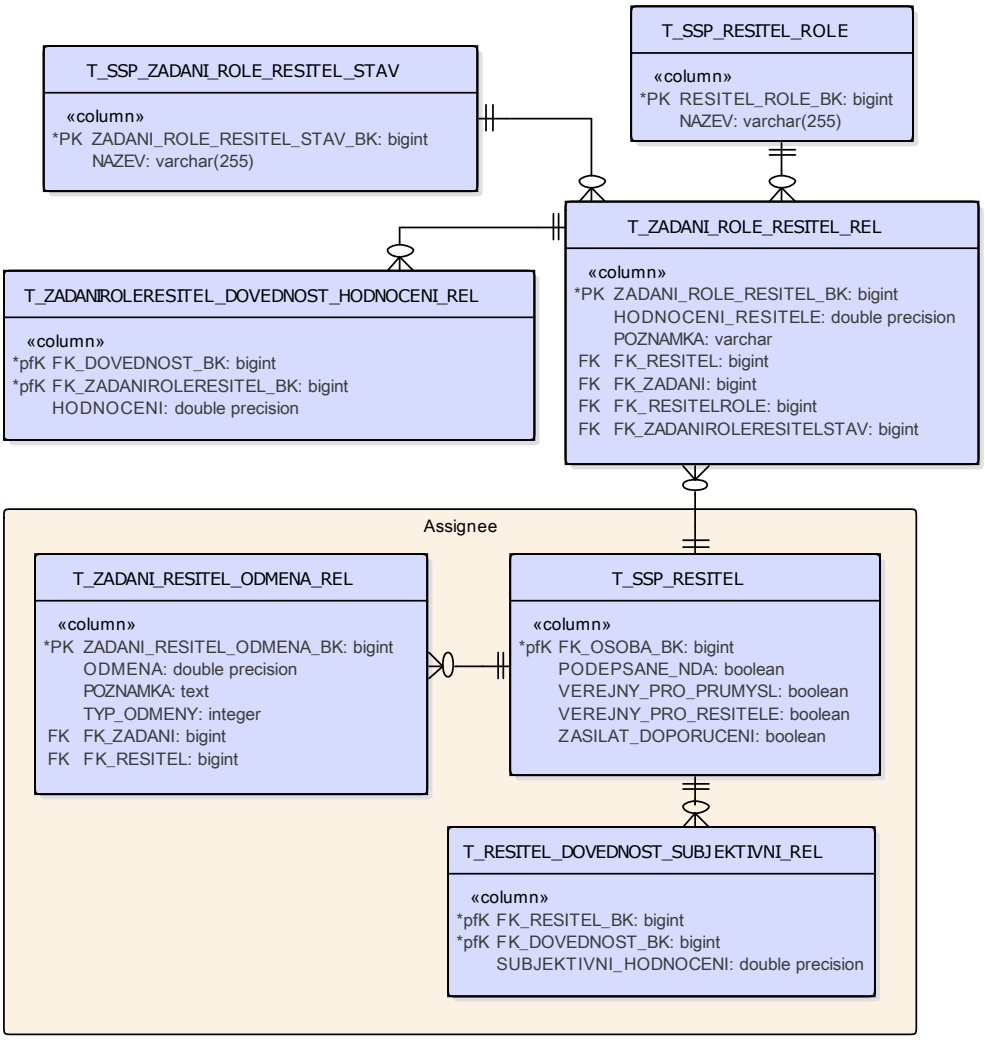


Obrázek 5.1: SSP user

ného databázového spojení. Výhodou napojení přes uživatelské jméno je tzv. pozdní napojení. Pokud by např. nebyly integrovány změny ze zdrojového systému studentů, vyhledávání business klíče by se nezdařilo a napojení řešitele by neproběhlo. Díky uživatelskému jménu dochází ke spojení, až když je požadováno, při vkládání tedy entity nemusí vůbec existovat.

Kromě uživatelského jména si tabulka nese ještě údaj, jestli má osoba roli experta. Osobu a související entity ukazuje diagram 5.1.

5.2.2 Assignee



Obrázek 5.2: Entita Assignee v datovém skladu

Tabulka 5.1: Entita Assignee v datovém skladu

Název tabulky	**Popis**
T_RESITEL_DOVEDNOST_-SUBJEKTIVNI_REL	Subjektivní ohodnocení řešitelových dovedností.
T_SSP_RESITEL	Informace o řešiteli.
T_SSP_RESITEL_ROLE	Číselník možných rolí.
T_SSP_ZADANI_ROLE_RESI-TEL_STAV	Číselník stavů pro vztahy mezi řešitelem, zadáním a rolí.
T_ZADANI_RESITEL_OD-MENA_REL	Informace o finančním ohodnocení za zadání.
T_ZADANI_ROLE_RESI-TEL_REL	Vazba mezi řešitelem, zadáním a řešitelovou rolí.
T_ZADANIROLERESITEL_-DOVEDNOST_HODNOCENI_-REL	Hodnocení řešitelových dovedností za konkrétní zadání.

Řešitel v SSP odpovídá studentům v existujícím skladu. Spojení probíhá přes uživatelské jméno uvedené u osoby. Z tohoto důvodu nejsou integrované žádné osobní údaje, ty totiž pochází z jiného zdrojového systému. Uchovávají se však vlastnosti, jestli je jeho profil veřejný, jestli má podepsanou dohodu o mlčenlivosti (NDA) a jestli si přeje zasílat doporučení.

Podstatné pro řešitele jsou jeho dovednosti spolu s jejich ohodnocením. Další důležitou vlastností je jeho vztah s konkrétními zadáními současně s rolí, v jaké k zadání přistupoval. Spolu s tím souvisí i finanční ohodnocení za daná zadání. Model ukazuje obrázek 5.2, strukturu tabulka 5.1.

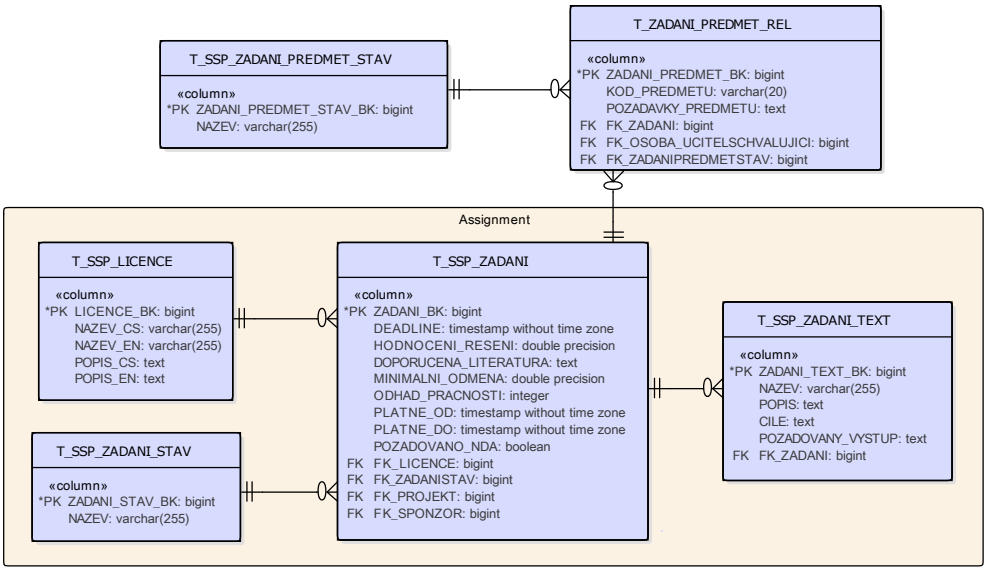
5.2.3 Assignment

Zadání je jeden ze základních stavebních kamenů SSP. Oproti řešiteli se jedná o úplně novou entitu, která ve skladu doposud neexistovala. Nelze ji tedy na nic napojit. Uchovává název s popisem, cíle a požadované výstupy, lhůtu pro odevzdání, předpokládanou časovou náročnost, minimální finanční ohodnocení a platnost zadání. Jelikož textové atributy mohou být uvedeny v různých jazykových mutacích, slouží pro jejich uchování zvláštní tabulka.

Entita je spjatá s různými předměty, tuto vazbu schvalují vyučující daného předmětu. Napojení na předmět je podobné jako napojení osoby z SSP na osobu v datovém skladu. Místo uživatelského jména zde jako atribut pro vazbu slouží kód předmětu. Opět je tímto způsobem umožněno pozdní spojení a nemusí se dohledávat business klíč předmětu.

Se zadáním dále souvisí stav a licenční podmínky, které mají český a anglický název a popis. Tabulky související s řešitelem (řešitelova role a jeho následná odměna) byly uvedeny v předchozí sekci, tabulka spjatá s expertem

bude popsána v sekci následující. Ukázka je na obrázku 5.3, význam tabulek je v tabulce 5.2.



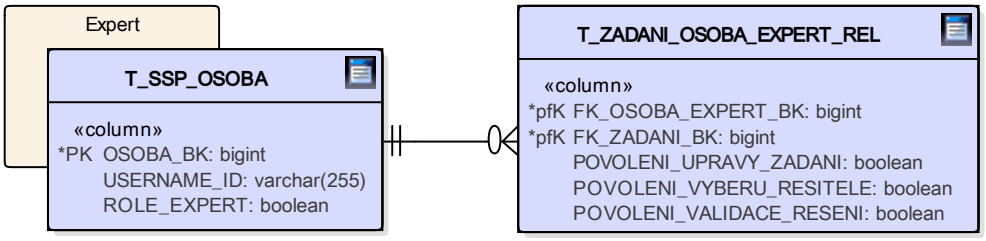
Obrázek 5.3: Entita Assignment v datovém skladu

Tabulka 5.2: Entita Assignment v datovém skladu

Název tabulky	**Popis**
T_SSP_LICENCE	Číselník licencí.
T_SSP_ZADANI	Informace o zadání.
T_SSP_ZADANI_PREDMET_-STAV	Číselník stavů pro vztah mezi zadáním a předmětem.
T_SSP_ZADANI_STAV	Číselník stavů zadání.
T_SSP_ZADANI_TEXT	Textové informace o zadání.
T_ZADANI_PREDMET_REL	Vazba mezi zadáním, předmětem a schvalujícím učitelem.

5.2.4 Expert

Expert se jako samostatná tabulka nezachovala. Existuje vazebná tabulka mezi osobou a zadáním, která udává roli osoby jako experta. U osoby jsou pomocí příznaku reflektováni všichni registrovaní experti. To je z toho důvodu, aby se zachovala informace i o těch expertech, kteří ještě na žádném zadání nespolupracovali. Opět je přiložen diagram 5.4 a popisující tabulka 5.3.



Obrázek 5.4: Entita Expert v datovém skladu

Tabulka 5.3: Entita Expert v datovém skladu

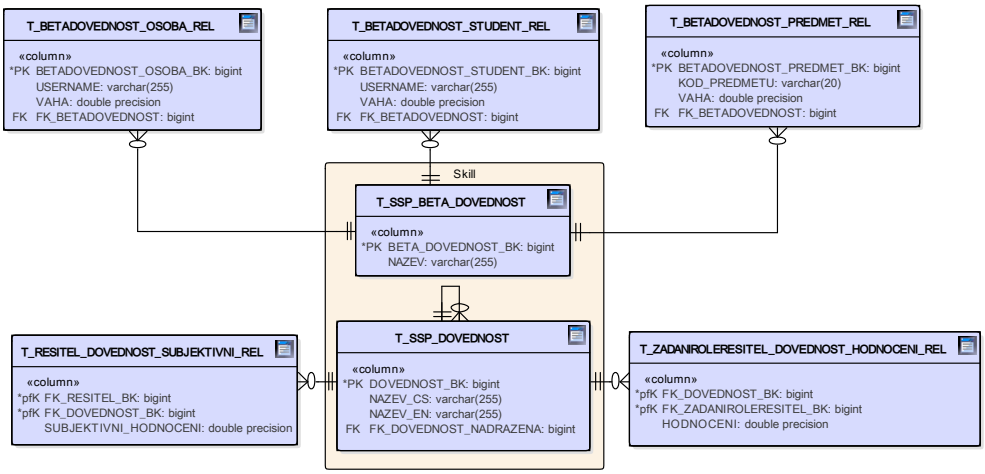
Název tabulky	**Popis**
T_SSP_OSOBA	Informace o osobě s příznakem, jestli je expert.
T_ZADANI_OSOBA_EXPERT_REL	Vazba mezi zadáním a expertem.

5.2.5 Skill

Student nabývá dovedností, jejichž hodnocení vzniká z předmětů a předchozích hodnocení za splněná zadání. Jelikož existují dva druhy dovedností, kde první slouží pro určování požadavků na řešitele zadání a pomocí druhého se studenti porovnávají a automaticky doporučují na nově vložená zadání, bylo rozhodnuto toto oddělení zachovat.

V budoucnu je možné, že se oba typy hodnocení sjednotí, aktuálně však ani jeden druh není podmnožinou druhého. Pokud by byla vytvořena jedna struktura, neexistoval by jednotný business klíč pro identifikaci daného hodnocení. Může být namítnuto, že samotný název je dostatečným identifikátorem pro danou dovednost. Ten se však může změnit např. z důvodu překlepu a historizace by pak změnu názvu považovala za úplně novou dovednost. Proto je aktuálně správným způsobem nechat dovednosti oddělené. Pokud se v některých dovednostech oba druhy protínají, spojení těchto dovedností může být realizováno přes název podobně jako navázání studentů a řešitelů pomocí uživatelského jména.

Dovednost ze základní databáze SSP obsahuje český a anglický název a tvoří hierarchickou strukturu. Řešitel ji může sám ohodnotit a současně je hodnocena pro konkrétního člověka zadavatelem za splněné zadání. Podpůrná databáze pro BI nabízí název dovednosti a jejich hodnocení vzhledem ke studentům a ostatním osobám, do kterých spadají učitelé. Zároveň se z této databáze zachovává, jaké minimální hodnocení dovednosti by měl mít člověk po absolvování určitého předmětu. Diagram struktury je na obrázku 5.5, její popis v tabulce 5.4.



Obrázek 5.5: Entita Skill v datovém skladu

Tabulka 5.4: Entita Skill v datovém skladu

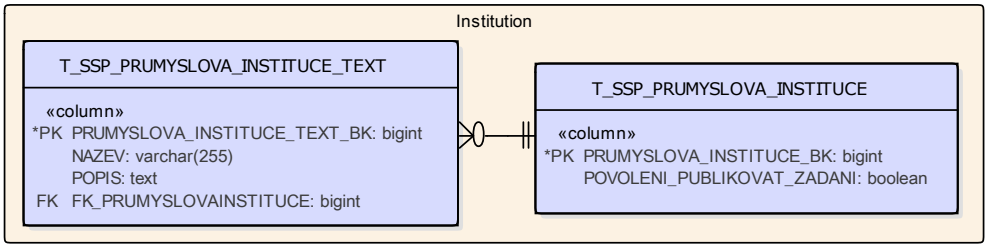
Název tabulky	**Popis**
T_BETADOVEDNOST_-OSOBA_REL	Vztah mezi dovedností z BI databáze a osobou (učitelem) v datovém skladu.
T_BETADOVEDNOST_PRED-MET_REL	Vztah mezi dovedností z BI databáze a předmětem.
T_BETADOVEDNOST_STU-DENT_REL	Vztah mezi dovedností z BI databáze a studentem.
T_RESITEL_DOVEDNOST_-SUBJEKTIVNI_REL	Subjektivní ohodnocení řešitelových dovedností.
T_SSP_BETA_DOVEDNOST	Informace o dovednosti pocházející z podpůrné BI databáze.
T_SSP_DOVEDNOST	Informace o dovednosti pocházející z hlavní databáze.
T_ZADANIROLERESITEL_-DOVEDNOST_HODNOCENI_-REL	Hodnocení řešitelových dovedností za konkrétní zadání.

5.2.6 Ostatní entity

Předchozí entity nejsou jediné, které byly při analýze navrženy pro uchování v datovém skladu. Tato sekce se zabývá těmi, které hlavním entitám sice přidávají kontext, ale nejsou nezbytné pro integraci do datového skladu. Kvůli jednoduchosti navržených struktur není uveden popis jednotlivých tabulek.

5.2.6.1 Institution

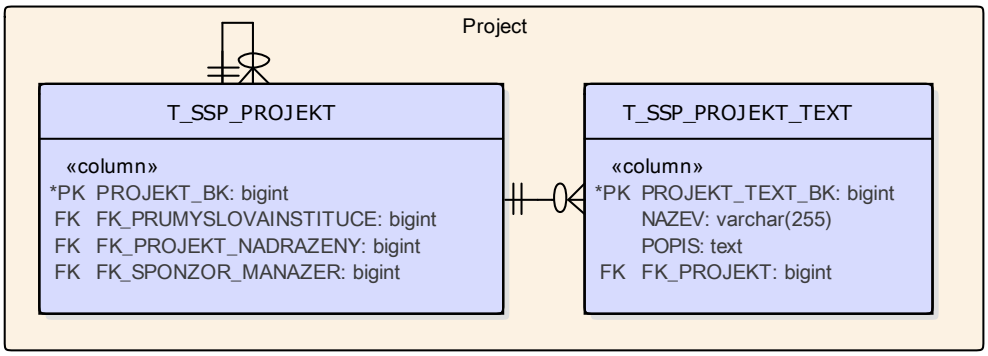
U průmyslové instituce se uchovává její textový popis a příznak, jestli může publikovat zadání. Konkrétní tabulky jsou na diagramu 5.6.



Obrázek 5.6: Entita Institution v datovém skladu

5.2.6.2 Project

Pro projekt se udržuje informace o jeho hierarchii a textovém popisu. Strukturu ukazuje obrázek 5.7.



Obrázek 5.7: Entita Project v datovém skladu

5.2.6.3 Solution

Tato entita se samostatně nezachovala. Důležité je pouze hodnocení daného řešení, které je uvedeno jako atribut u samotného zadání.

5.2.6.4 Sponsor

U sponzora se ukládají informace o jeho návaznosti k průmyslové instituci. Dále je u něj uvedeno, jestli může publikovat zadání. Jedna osoba může figurovat jako více sponzorů pro různé průmyslové instituce, proto primární klíč sponzora a osoby není shodný. Tabulka je na diagramu 5.8.

Obrázek 5.8: Entita Sponsor v datovém skladu

5.3 Přístupová vrstva

Jelikož se přístupová vrstva skládá ze dvou částí, její návrh je členěn do dvou sekcí. První se zaměřuje na struktury v sémantické vrstvě, druhá navrhuje jednotlivá datová tržiště.

5.3.1 Sémantická vrstva

Sémantická vrstva se skládá z tzv. stavebních bloků, které odpovídají různým entitám. V normalizované formě může být entita rozdělena do více tabulek. Pro odstranění nekonzistencí souvisejících se spojováním entitních tabulek slouží právě tato vrstva, kde spojování probíhá jednou.

Datový sklad ČVUT zatím sémantickou vrstvu nemá. Jak bylo zmíněno v sekci 4.2.1, je vytvářena v rámci jiné, paralelně vznikající diplomové práce. Stavební bloky, které jsou potřebné pro další vrstvu, ale které nejsou součástí SSP, zde nebudou uvedeny, ačkoliv budou implementovány. Překračují totiž rámec této práce a budou sloužit pouze pro možnost vzniku datových tržišť. Tyto bloky se po dokončení obou prací spojí dohromady.

Návrh je podobně jako návrh integrované vrstvy rozdělen podle jednotlivých entit. Pokud není uvedeno jinak, tabulky mají stejnou podobu jako v integrované vrstvě (diagram C.1). Oproti ní je zde jednodušší vytvořit jiný pohled na data. Z tohoto důvodu jsou navrženy základní stavební bloky a v některých případech i ukázka jejich rozšíření. Uvedený výčet není konečný, další bloky mohou být postupně přidány dle budoucích požadavků.

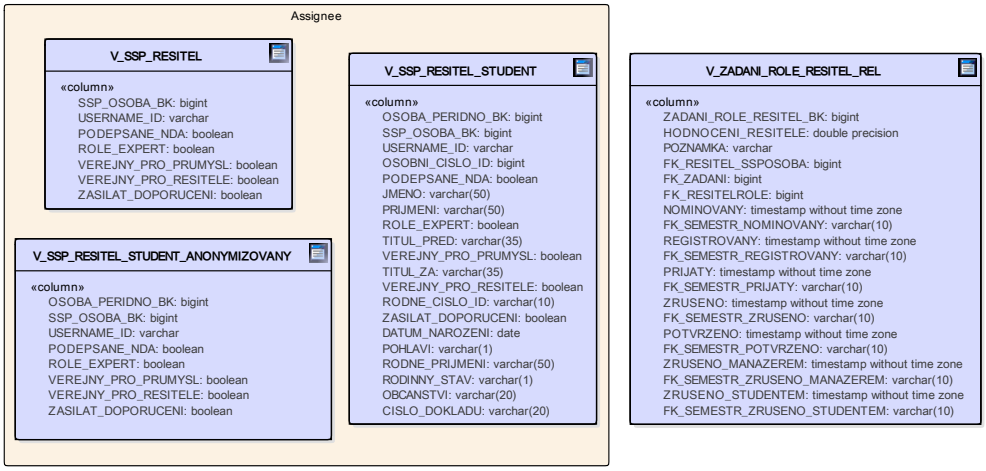
5.3.1.1 Assignee

Pro řešitele byly vytvořeny tři základní stavební bloky. V_SSP_RESITEL vznikl spojením tabulek T_SSP_RESITEL a T_SSP_OSOBA a obsahuje všechny řešitele, kteří jsou registrovaní v SSP. V_SSP_RESITEL_STUDENT uchovává pouze řešitele, kteří jsou současně v datovém skladu již uvedeni jako studenti. Oproti přechozímu navíc obsahuje informace, které se nachází v ta-

bulce T_OSOB_OSOBA. V_SSP_RESITEL_STUDENT_ANONYMIZOVANY je vytvořen stejně jako předchozí, místo identifikačních údajů obsahuje pouze hash uživatelského jména.

První pohled může být použit v situacích, kdy není podstatné, jestli osoba ve skladu existuje z jiných zdrojů. Druhý pohled poslouží v případě, kdy jsou hlavním cílem studenti. Poslední má stejné použití jako druhý, může být využit v situacích, kdy není žádané danou osobu přesně identifikovat. Příkladem může být analýza hodnocení dovedností v sekci 7.1.

Dále se změnila struktura vazebné tabulky T_ZADANI_ROLE_RESITEL_REL. Ta v pohledu V_ZADANI_ROLE_RESITEL_REL obsahuje jednotlivé stavy vztahu. Pro stavy jsou uvedeny jak odkazy na semestr, kdy změna proběhla, tak i přesný časový údaj. Tabulka, která původně číselník stavů obsahovala, se nezachovala. Vše ukazuje diagram 5.9.

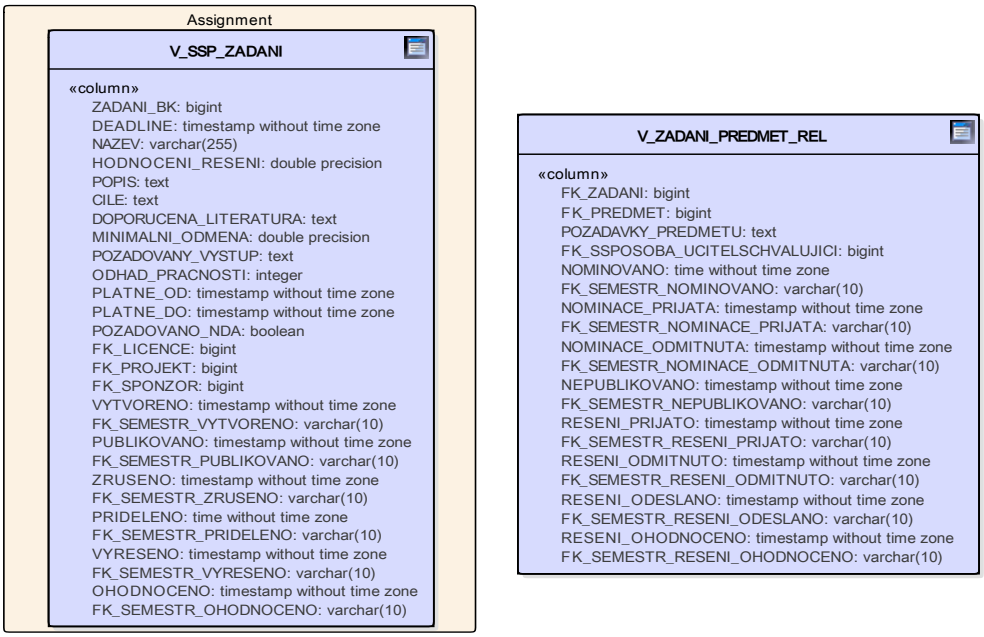


Obrázek 5.9: Entita Assignee v sémantické vrstvě

5.3.1.2 Assignment

V_SSP_ZADANI sjednocuje tabulky spjaté se zadáním. Konkrétně obsahuje textové hodnoty, jako je název apod. Ty jsou v integrované vrstvě ve zvláštní tabulce kvůli jazykovým mutacím. Tento pohled proto obsahuje pouze textové hodnoty, které byly vloženy jako poslední. Dále podobně jako u stavu mezi řešitelem, zadáním a rolí v předchozí sekci uchovává jednotlivé stavy jako samostatné položky. Znovu se jedná o konkrétní čas změny i související semestr. Tabulka s výčtem stavů se v sémantické vrstvě nezachovala.

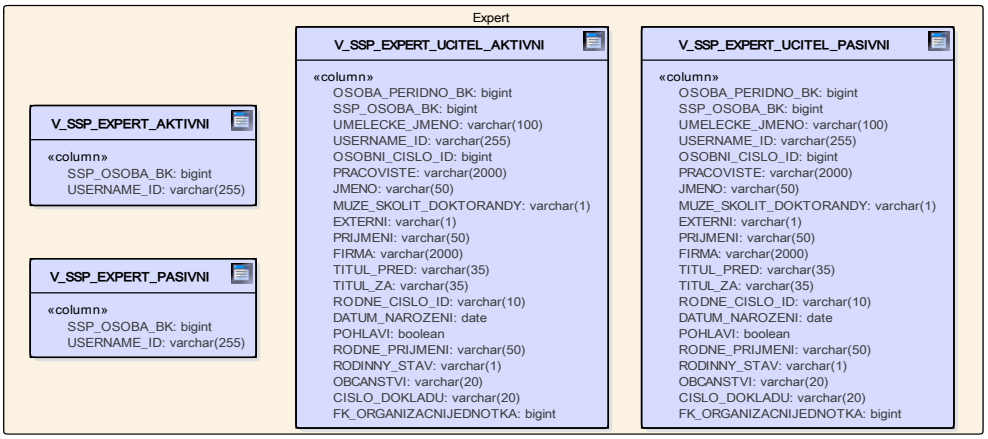
To samé se se stavy stalo i u V_ZADANI_PREDMET_REL. Navíc došlo ke spojení s předmětem z datového skladu. Místo kódu předmětu je zde jeho business klíč. Ukázka na obrázku 5.10.



Obrázek 5.10: Entita Assignment v sémantické vrstvě

5.3.1.3 Expert

Oproti integrované vrstvě, kde expert nemá samostatnou tabulku, jsou zde uvedeny čtyři stavební bloky pro tuto entitu. Dvě obsahují všechny experty, kteří jsou uvedeni v SSP. Další dvě tvoří jen ti, co jsou současně uvedení ve skladu jako učitelé. Rozdíl mezi těmito dvojicemi je v aktivitě expertů. Jeden pohled vždy zobrazuje pouze aktivní experty. To jsou ti, kteří se někdy na nějakém zadání jako experti podíleli. Druhá z dvojice zobrazuje i pasivní, tedy ty, kteří jsou jako experti registrovaní. Bloky jsou na obrázku 5.11.

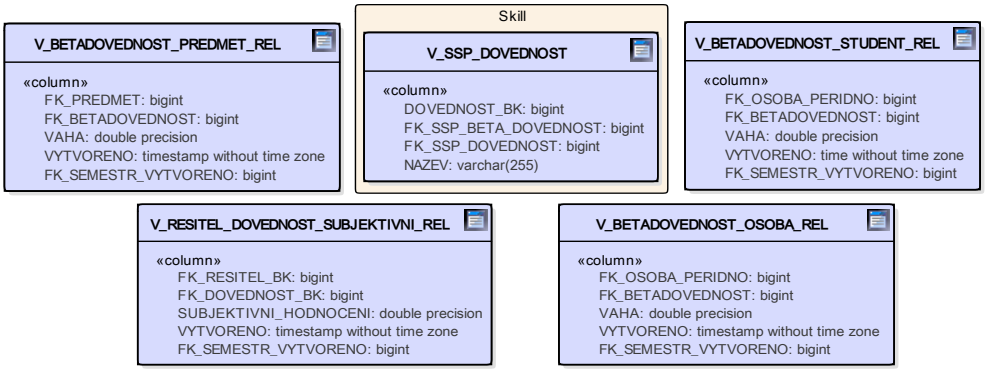


Obrázek 5.11: Entita Expert v sémantické vrstvě

5.3.1.4 Skill

Integrovaná vrstva uchovává dva druhy dovedností, které by měly být v budoucnu sjednoceny. Proto je v sémantické vrstvě pouze jeden pohled V_SSP_DOVEDNOST, který oba druhy spojuje dohromady. Množinově se jedná o sjednocení obou typů dovedností.

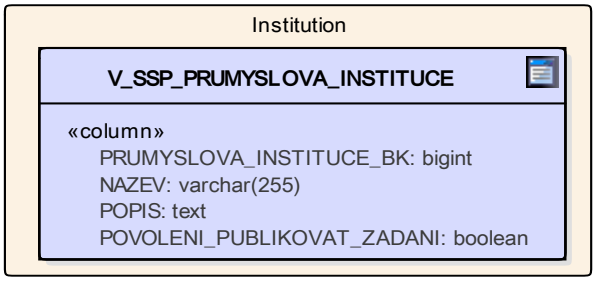
Všechny vazby, které existovaly pro betadovednost, nyní obsahují business klíče entit, ke kterým patří. Místo uživatelského jména je použit OSOBA_PERIDNO_BK a místo kódu předmětu je také jeho business klíč. Dále je u každého hodnocení uvedeno, kdy bylo vloženo do skladu. Zdrojová databáze bohužel nemá hodnotu vytvoření, nejbližší hodnotou je vložení do skladu. Toto se týká i subjektivního hodnocení v pohledu V_RESITEL_DOVEDNOST_SUBJEKTIVNI_REL. Ukázka je na obrázku 5.12.



Obrázek 5.12: Entita Skill v sémantické vrstvě

5.3.1.5 Institution

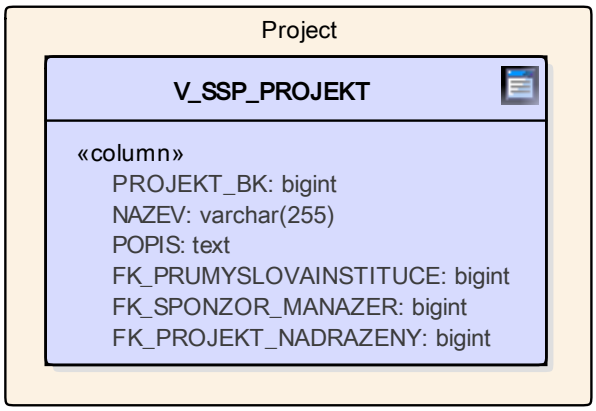
Průmyslová instituce se v integrované vrstvě skládá ze dvou tabulek, jedna hlavní a druhá s textovými hodnotami. Tyto dvě tabulky spojuje V_SSP_PRUMYSLOVA_INSTITUCE. Jako textové hodnoty bere ty, které byly vloženy jako poslední. Jedná se o stejný přístup jako u zadání. Struktura je na obrázku 5.13.



Obrázek 5.13: Entita Institution v sémantické vrstvě

5.3.1.6 Project

Stejně jako u průmyslové instituce, V_SSP_PROJEKT sjednocuje hlavní a textovou tabulku pro projekt. Sjednocení je ukázáno na obrázku 5.14.



Obrázek 5.14: Entita Project v sémantické vrstvě

5.3.1.7 Sponsor

Údaje o sponzorovi jsou v bloku V_SSP_SPONZOR. Uchovány jsou zde informace jak z tabulky T_SSP_SPONZOR, tak z tabulky T_SSP_OSOBA. Vše je vidět na diagramu 5.15.

Obrázek 5.15: Entita Sponsor v sémantické vrstvě

5.3.2 Datová tržiště

Přístupová vrstva obsahuje kromě sémantické vrstvy také datová tržiště (datamarty), nad kterými se následně provádí analýza. Struktura byla denormalizována a ve všech případech se používá hvězdicové schéma. Kvůli zjednodušení

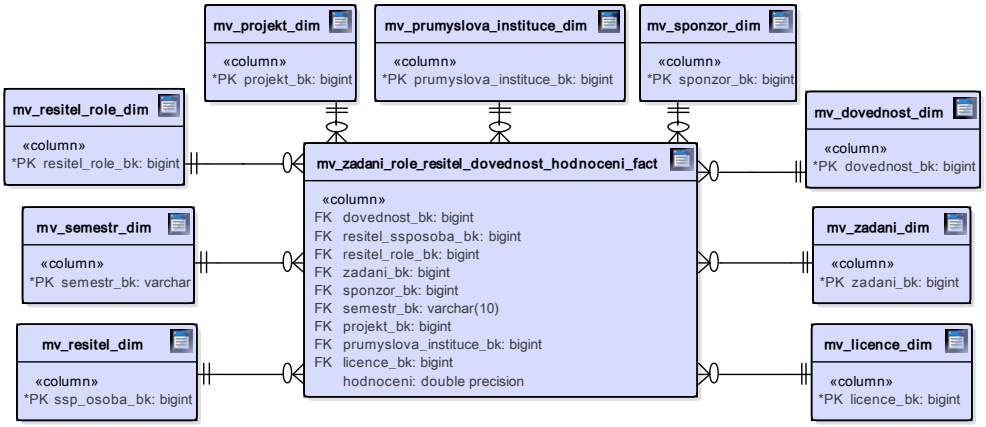
nejsou v žádném diagramu ukazovány atributy dimenzí. Ty odpovídají dané entitě ze sémantické vrstvy, pro analýzu lze vybrat pouze ty potřebné.

Data z SSP mohou sloužit jako jeden z pilířů pro hodnocení studentů. Dalším pilířem jsou známky za studium a výsledky vědecké činnost v případě doktorandů a učitelů. Proto jsou v první sekci navrženy datamarty související s řešitelem, tedy studentem. Druhá se zaměřuje na učitele, konkrétně experty. Třetí je spjatá s předměty. Ostatní entity jsou diskutovány v poslední sekci.

5.3.2.1 Řešitel

Díky stavebním blokům není podstatné, jestli je řešitel student či nikoliv. V realitě by sice nemělo nastat, že řešitel je někdo jiný než student, na datamart využívající data z SSP by to ale nemělo mít významný dopad. Na základě požadavku stačí použít vhodný blok, z kterého tržiště vzniká.

V datovém skladu je uchováno hodnocení dovedností řešitele za dané zadání a související roli. Pro tento vztah se nabízí tvorba prvního datamartu. Základními dimenzemi jsou zmiňovaný řešitel, dovednost, role, zadání a časová dimenze, na kterou je nejvhodnější vzhledem ke studentům použít semestr. Ten odpovídá semestru, kdy bylo zadání vyřešeno. Denormalizací vznikají další možné dimenze, kterými jsou sponzor, který vytvářel zadání, projekt, do kterého zadání patří, průmyslová instituce, pod kterou spadá sponzor, a licence výsledného řešení zadání. Ukázka struktury takového datamartu je na obrázku 5.16.



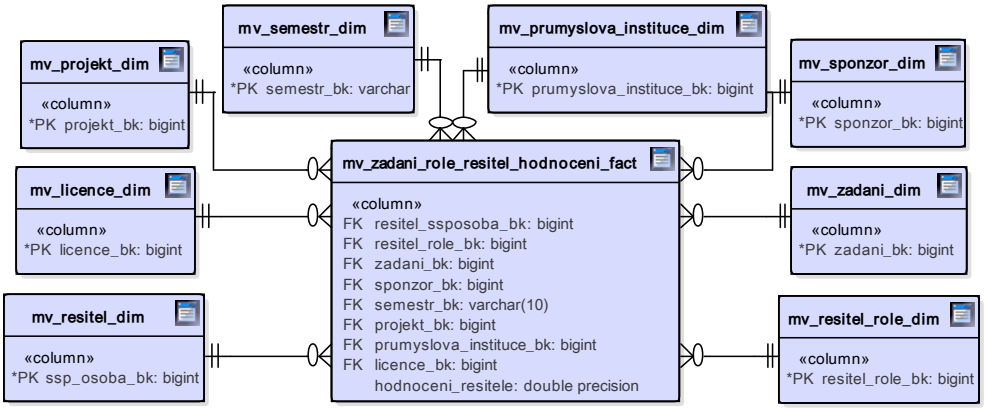
Obrázek 5.16: Datamart hodnocení dovednosti za vyřešená zadání

Jako další dimenze se nabízí ty, které je možné vytvořit z dat existujících v datovém skladu. Příkladem může být studijní obor. Přidání oboru jako nové dimenze do výše navrženého tržiště však není přímočaré. Student totiž může studovat dva obory najednou jako dvě souběžná studia. V takovém případě by pro jednoho studenta vzniklo více řádků ve faktové tabulce. Pokud by se následně dělala analýza, pro kterou je obor nezajímavý, vypočítaná hodnota

by nemusela udávat skutečný výsledek. Špatně vypočítanou hodnotou může být např. průměrné hodnocení role za semestr podle sponzorů. Hodnocení u studentů s více souběžnými studii by bylo v tabulce obsaženo několikrát a průměr by neodpovídal skutečnosti. Jako řešení lze považovat vytvoření dvou datamartů. První by neobsahoval obor jako dimenzi, jednalo by se o výše navržený datamart. Druhý by tuto dimenzi obsahoval, muselo by se však uvést do dokumentace, že rozpad podle dimenzí je možný, pouze pokud bude vytvořen i rozpad podle oboru. Čistším řešením této nekonzistence je využití pouze druhého datového tržiště, vytvoření počítané metriky, která by byla průměrem hodnocení pro různá zadání, a smazání metriky původní. Je zřejmé, že tvorba počítaných metrik závisí na použitém BI nástroji.

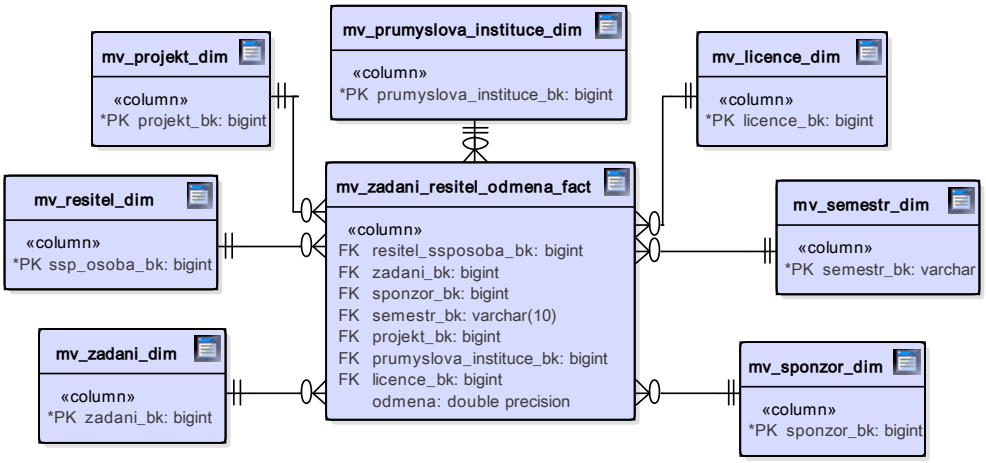
U zadání je uvedeno hodnocení řešení. Tato metrika se může zdát být vhodná pro analýzu, proto se nabízí její přidání do faktové tabulky. Jednalo by se o špatný návrh způsobený různou granularitou měřítka. Faktová tabulka totiž musí obsahovat všechna měřítka ve stejné granularitě. Řešením je vytvoření jednoho nebo několika nových datových tržišť, kde by měřítkem bylo hodnocení řešení zadání a dimenze by byly vhodně vybrané tak, aby se předešlo problému popsaném v předchozím odstavci. Na řešení může pracovat více řešitelů, jako další metrika do nového datamartu by připadala v úvahu i poměrová část bodů na řešitele, pokud by jedna z dimenzí byla právě řešitel.

Kromě hodnocení dovedností může být zajímavou metrikou i hodnocení rolí. Kromě dovednosti jsou zde stejné dimenze jako v prvním navrhovaném tržišti. Konkrétně se jedná o dimenze řešitel, role, zadání, sponzor, projekt, průmyslová instituce, licence a semestr, kdy bylo zadání vyřešeno, jako časová dimenze. Datové tržiště je na obrázku 5.17.



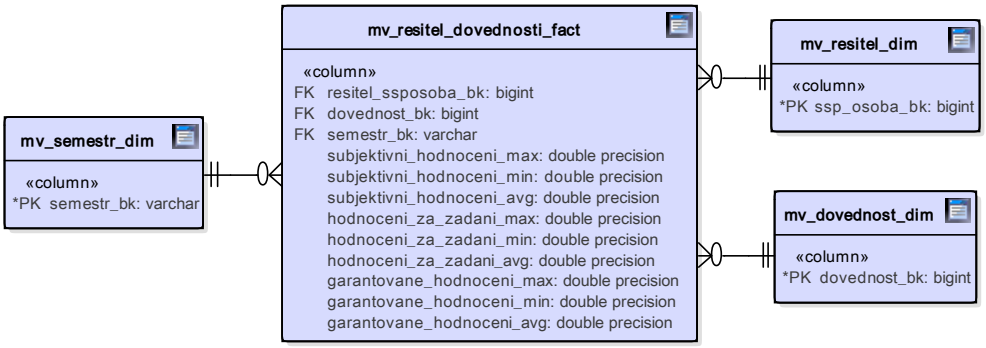
Obrázek 5.17: Datamart hodnocení role za vyřešená zadání

Řešitelé jsou hodnoceni stipendiem. Nejen pro finanční oddělení by mohla být analýza nad těmito hodnotami zajímavá. Proto vznikl návrh na další tržiště, který je uveden na obrázku 5.18. Dimenze jsou řešitel, zadání, sponzor, projekt, průmyslová instituce, licence a semestr jako časová dimenze.



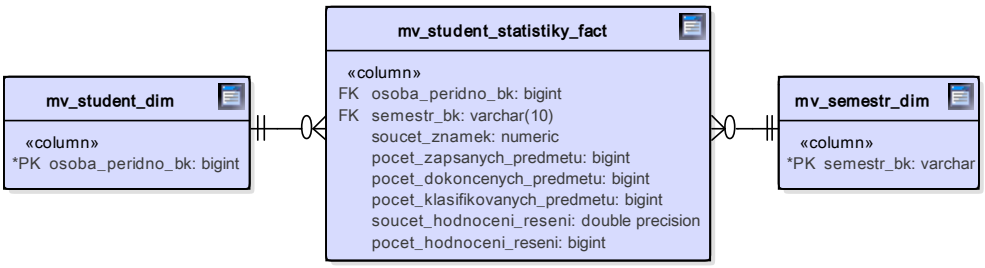
Obrázek 5.18: Datamart odměny za vyřešená zadání

Kromě hodnocení dovedností za zadání existují data i pro subjektivní hodnocení a napočítané garantované hodnocení. Pro subjektivní a garantované hodnocení je časovým kontextem datum vložení do databáze. Platí od doby, kdy byly vloženy, až do doby, kdy byly vloženy další. Pokud se za jeden semestr vloží více hodnot, může se uvádět minimální, maximální a průměrná. Pro hodnocení za zadání lze uvádět hodnoty stejné. Ukázka datamartu je na obrázku 5.19. Problémem může být, že není možné získat přesné datum vytvoření subjektivního a garantovaného hodnocení. Používá se datum vložení do skladu, což je závislé na frekvenci importu dat. Počátek platnosti prvního záznamu navíc způsobí, že se hodnocení k entitě přidělí až od prvního spuštění procesu integrace dat. To není správné, bohužel zdrojový systém neposkytuje adekvátní data. Alternativou může být přiřazení vytvoření prvního hodnocení jako počátek prvního semestru ve skladu. Ani to by však nebylo správné.



Obrázek 5.19: Datamart vývoj hodnocení dovedností

Pro všechna předchozí tržiště stačila data z SSP. Výjimkou bylo navrhované doplnění oboru. Zajímavým porovnáním můžou být statistiky za odstudované předměty a vyřešená zadání. Zde nastává menší problém, data v SSP jsou spjatá s člověkem, statistiky za studium jsou spjaté se studiem. Navrhovaným řešením je agregace studijních výsledků přes osobu. Datamart neobsahuje průměry, jelikož by po rozpadu pouze přes semestr neudával skutečné výsledky. Místo toho jsou uvedeny součty, počítanou metrikou je pak lze vydělit počty, což zajistí správnou hodnotu průměru. Oproti předchozím je zde omezující podmínka, že řešitel musí nebo musel být současně student. Datové tržiště je možné vidět na obrázku 5.20. Použití studia a souvisejícího oboru jako dimenze opět přináší problém. Řešení zadání může být spjato s více řešiteli, správné statistiky by mohly vyřešit součty poměrových hodnocení a součty poměrových počtů zadání na studenta. Průměr hodnocení za zadání v oboru by se mohl počítat z nich. Jak ale bylo zmíněno, jeden student může mít více studií, proto by se poměrové hodnoty musely přizpůsobit i této skutečnosti. V tomto případě by bylo lepším řešením vytvoření samostatného datmartu pro obory, kde by byly statistiky správně agregované. Pokud by byl požadavek na přidání oboru do existujícího tržiště, muselo by být zdokumentováno, že rozpad musí být vždy tvořen alespoň dle studenta.



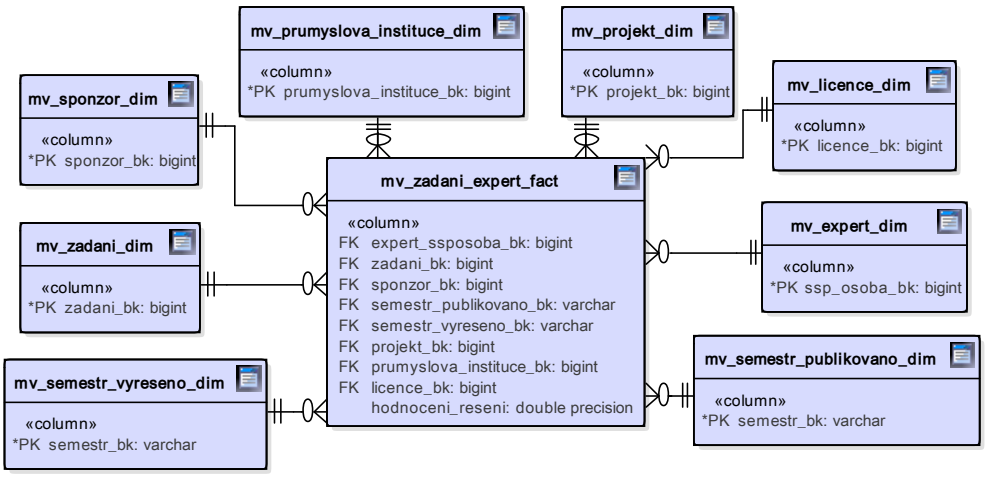
Obrázek 5.20: Datamart statistiky studenta

5.3.2.2 Expert

Ve většině případů je expertem učitel. Díky tomu má oproti sponzorovi z externí firmy výhodu v tom, že ví, jak pracovat se studenty, a zná požadavky na předměty. Nemusí to být pravidlem, proto jsou na úrovni stavebních bloků opět namodelovány různé pohledy na experta. Podobně jako u řešitele by to nemělo mít zásadní dopad na datamarty související s daty z SSP.

Pro experta mohou být důležité počty zadání a to ze dvou hledisek. Prvním jsou publikovaná zadání, druhým zadání vyřešená. Tomu odpovídají i dvě časové dimenze, které tvoří semestry. Dalšími dimenzemi jsou expert, zadání, sponzor, projekt, průmyslová instituce a licence. Podobně jako u řešitele by mohla být zajímavá metrika hodnocení řešení zadání. Jelikož se na zadání přiřazuje maximálně jeden expert, může být tato hodnota ve faktové tabulce

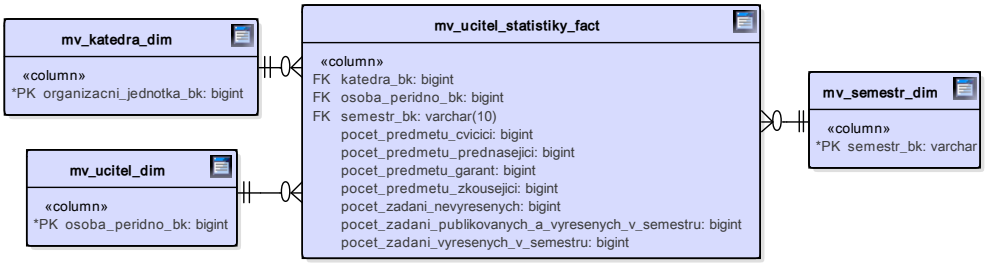
navrhovaného tržiště, aniž by musela být tvořena počítaná metrika. Ukázka je na obrázku 5.21.



Obrázek 5.21: Datamart zadání s experty

Pokud je expert učitel, patří pod nějakou katedru. Ta by mohla figurovat jako další dimenze. Použití katedry jako další dimenze ilustruje i plynulé spojení mezi daty z SSP a daty z jiného zdroje v datovém skladu.

Ukázkou většího propojení mezi daty z SSP a jiných zdrojů může být i tržiště, které obsahuje počty předmětů a zadání. Konkrétně by se mohlo jednat o počty předmětů, kde byl učitel jako cvičící, přednášející, garant a zkoušející v daném semestru, a počty zadáních ještě nevyřešených, vyřešených a současně publikovaných a vyřešených v daném semestru. Ukázka takového tržiště je na obrázku 5.22.

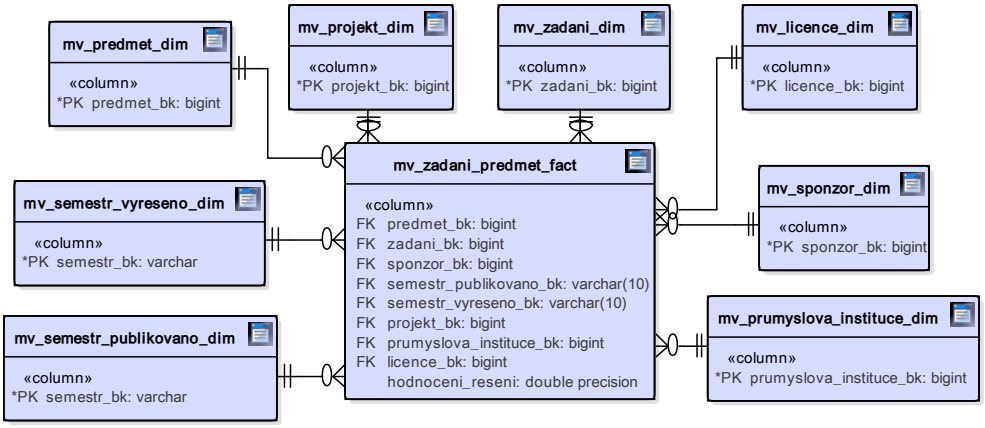


Obrázek 5.22: Datamart statistiky učitele

5.3.2.3 Předmět

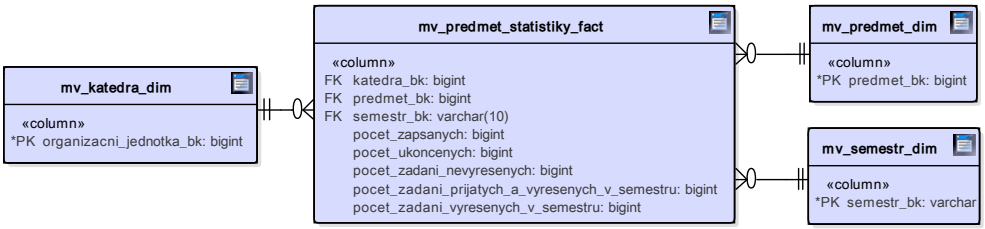
Zadání můžou být odevzdávána do různých předmětů. Proto se jako zajímavé jeví i počty odevzdaných zadání do různých předmětů. Navrhovaný datamart obsahuje klasické dimenze, jakými jsou předmět, zadání, sponzor, projekt, prů-

myslová instituce a licence. Podobně jako u experta se přidávají dvě časové dimenze, jedna pro publikovaný semestr, druhá pro odevzdaný semestr. Znovu by se mohla použít metrika hodnocení řešení. Podobně jako u experta může být zahrnuta do navrhované faktové tabulky. Zadání může být sice odevzdané do jednoho předmětu více studenty, jelikož zde ale dimenze řešitel není, struktura je v pořádku. Tržiště je ukázáno na obrázku 5.23.



Obrázek 5.23: Datamart zadání nominovaná k předmětům

Pro spojení s daty v datovém skladu lze použít jako dimenzi katedru. Dále může být vytvořeno tržiště, které obsahuje počty vyřešených a nevyřešených zadání, současně připojuje počty studentů, kteří měli daný předmět zapsaný v požadovaném semestru a kteří ho úspěšně ukončili. Struktura datamartu je na diagramu 5.24.



Obrázek 5.24: Datamart statistiky předmětu

5.3.2.4 Ostatní

Pro ostatní entity, jako je sponzor, průmyslová instituce a projekt, může být analýza provedena na již navržených datamartech. Je možné získat počty zadání, za která byl nějaký řešitel ohodnocen, za která byla udělena finanční odměna, která patřila do nějakého předmětu nebo ve kterých figuroval nějaký expert.

Pokud by se tyto statistiky často kombinovaly, nebyl by výše uvedený přístup uživatelsky přívětivý. Řešením by byla nová faktová tabulka, kde by jako atributy byly příznaky odpovídající různým požadavkům.

Business požadavky na analýzu dat se časem vyvíjejí, proto výčet navržených datových tržišť není definitivní a může se v budoucnu rozvíjet. Datamarty z přechozích sekcí byly navrženy s ohledem na základní požadavky, které na data z SSP mohou být kladena. Z tohoto důvodu byly uvedeny i možnosti špatného návrhu, tato práce tak může sloužit jako podklad pro další návrhy. Např. po dokončení práce zabývající se integrací vědeckých výsledků můžou být napojeny do jedné analýzy o učitelích informace o vědecké činnosti, statistiky o počtech vyučovaných předmětů a počty zadání, na kterých učitel spolupracoval jako expert.

Kapitola 6

Implementace

Jakmile jsou navrženy struktury pro ukládání dat, přichází čas pro přesun těchto dat mezi jednotlivými vrstvami. Do stage se data dostanou pomocí exportu z databáze. Transformaci mezi dočasným úložištěm a integrovanou vrstvou zajišťují ETL procesy. Sémantická vrstva je tvořena z jednoduchých databázových pohledů nad integrovanou vrstvou, datová tržiště vznikají ze sémantické pohledy materializovanými.

6.1 Stage

Pro převod dat ze zdrojového systému do stage byl vytvořen skript. Jeho prvním krokem je vyexportování dat z obou SSP databází. Dump obsahuje i příkazy pro vytvoření odpovídajících struktur. Druhým krokem je nahrání těchto dat do stage a následná kontrola existence všech tabulek. Poslední krok spustí ETL procesy, které slouží pro transformaci dat do integrované vrstvy.

6.2 Integrovaná vrstva

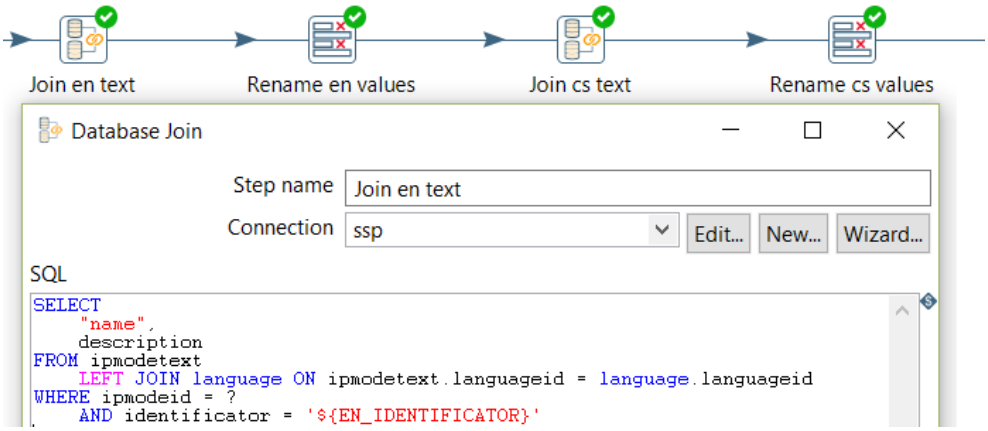
Následující odstavce se věnují ETL transformacím do integrované vrstvy, tedy převodu dat mezi stage a integrovanou vrstvou. Nejdříve jsou popsány důležité prvky, které jsou většinou specifické kvůli formátu zdrojové databáze. Poté následuje přehled jednotlivých transformací a jejich sjednocení do jobů. Pro realizaci ETL procesů byl zvolen opensource nástroj Pentaho Data Integration (PDI, znám i pod názvem Kettle), ten je v datovém skladu ČVUT pro ETL již využíván.

6.2.1 Specifické prvky

Tato část popisuje specifika struktur v SSP a způsob řešení těchto vlastností v rámci ETL transformací.

6.2.1.1 Jazyková mutace

Zdrojová databáze SSP je připravena pro různé jazykové mutace. V současnosti jsou vícejazyčné pouze některé systémové informace, jako jsou např. licence, které se vztahují k řešení zadání. U takových entit se uchovává buď anglický, nebo anglický a český text. Načtení těchto jazykových mutací bylo realizováno pomocí joinu entitní tabulky a tabulky s textem, který byl vyfiltrován na požadovaný jazyk. Konkrétní realizace proběhla pomocí stepu *Database join*. Ukázka takového spojení je na obrázku 6.1.



Obrázek 6.1: Stepy pro načtení anglické a české mutace

Entity, které vkládá uživatel, mohou mít různé jazyky. V současné verzi SSP je možné vkládat pouze jeden. Z tohoto důvodu byly v první iteraci návrhu textové informace součástí hlavních tabulek. ETL transformace tuto skutečnost musely reflektovat a join s textovou hodnotou proběhl už v rámci vstupního stepu. Jelikož data pocházela z PostgreSQL, byla využita funkce DISTINCT ON. Načetly se všechny texty, seřadily se podle data vložení a nakonec se vzal první výskyt pro každou instanci, což zaručovalo použití DISTINCT ON vzhledem k primárnímu klíči entity. Pokud by tedy přibyl text v dalším jazyce, neuchoval by se, zároveň by to ale neovlivnilo aktuální historii uchovávaných entit.

Jelikož se ale do budoucna může SSP rozšířit a využít potenciál databázového modelu s různými jazykovými mutacemi, od předchozího způsobu se upustilo. Nebylo by totiž jasné, jestli první vložený text je ten nejdůležitější, který se má ve skladu zachovat. V některém jazyce totiž může být napsáno více informací než v jiném. Z toho důvodu se uchovávají textové informace ve zvláštních tabulkách, což umožňuje uchovat je všechny.

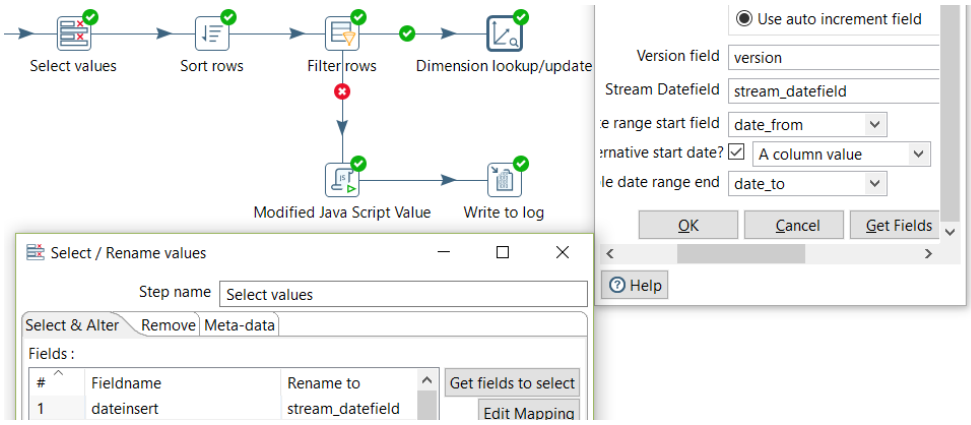
6.2.1.2 Historizace stavů

Zadání a jiné entity prochází různými stavy v průběhu svého životního cyklu. SSP umožňuje stavy uchovávat, aby mohlo tuto historii následně ukázat uživateli. Tomu odpovídá i struktura databáze zdrojového systému, kde existují vazebné tabulky mezi entitami a stavy, v nichž jsou tyto informace uložené.

V datovém skladu byl zvolen jiný přístup. Neexistuje vazebná tabulka a stav se přenesl přímo do entitní tabulky. Neznamená to však, že by se neuchovávala historie stavů. Předešlé stavy jsou uchovány díky historizaci, konkrétně díky použitému typu SCD2.

Technický postup je následovný. Entita se načte pomocí databázového spojení se všemi stavy, každý řádek dané instance odpovídá jednomu stavu. Spolu s tím se u každého stavu načte i jeho datum vložení. Stav se sám o sobě neupravuje, pokud se změní na jiný, vytvoří se nový řádek. Step *Dimension lookup/update* je pak schopný vložit pouze stavy, které byly nové a nejsou ještě ve skladu uchovány, a to se správnými daty určujícími platnost záznamu. Toto je možné, když se do položky *Stream Datefield* dá právě hodnota data, kdy byl stav vložen. Samozřejmostí je, že se záznamy nejdříve podle tohoto data seřadí. Pokud by řazení neproběhlo, mohl by se nejdříve vložit poslední stav, což by zamezilo vložení dřívějších stavů, které ale ve skladu ještě nejsou.

Obrázek 6.2 ilustruje předchozí postup. Datum vložení je nejdříve přejmenováno, poté následuje seřazení záznamů podle něj. Dalším krokem je vyfiltrování pouze těch záznamů, kde tato hodnota není prázdná. V datech z SSP totiž kvůli chybě opravdu existoval záznam s chybějící hodnotou. U něj není jisté, jaké pořadí by se takovému stavu mělo přiřadit. Účelem integrace není opravovat chyby, proto se takové záznamy jen logují. Oprava pak může proběhnout po domluvě se správci systému SSP. V posledním kroku se vyplní položka *Stream Datefield* s odpovídajícím jménem sloupce.



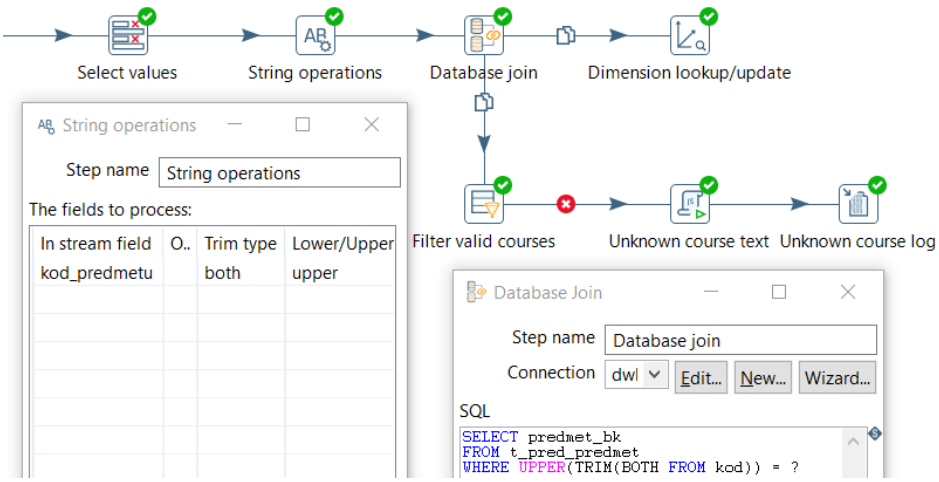
Obrázek 6.2: Ukázka použití Stream Datefield

6.2.1.3 Vazby mezi entitami

Datový sklad a SSP měly společné některé entity. Proto bylo nutné vyřešit jejich napojení. Jelikož primární klíče v SSP se liší od business klíčů ve skladu, tedy od primárních klíčů ve zdrojových systémech, napojení muselo proběhnout přes jiné identifikátory. V návrhu bylo stanoveno, že vazba mezi osobami a předměty, které existují v datovém skladu, a jejich odpovídajícími entitami v SSP bude probíhat pomocí uživatelských jmen a kódů předmětů.

Ukládání jmen a kódů však muselo být standardizováno. Od textových řetězců se odebraly počáteční a koncové mezery a byly převedeny na velká písmena. Ačkoliv napojení pomocí těchto hodnot nepotřebuje zjišťovat, jestli daná entita ve skladu existuje či nikoliv, může být vhodné jejich neexistenci alespoň vypsat do logu. Může se jednat pouze o překlep, který se může nahlásit správcům SSP.

Pro nalezení správné entity by mohl posloužit step *Database lookup*. U něj však nelze nastavit case-insensitive porovnávání a další podobné vlastnosti. Proto byl využit *Database join*. Od řetězce ze skladu jsou stejně jako od řetězce z SSP nejprve odebrané počáteční a koncové mezery a text je převeden na velká písmena. Tento postup byl použit pouze u kódů předmětů. Ukázka je na obrázku 6.3. Při vkládání osob z SSP by se v logu objevilo plno jmen, které ve skutečnosti ani ve skladu být z jiného zdroje nemají. Jedná se např. o sponzory. Proto vyhledávání jmen u osob nebylo použito, není však těžké ho implementovat podobně jako u kódů předmětů.



Obrázek 6.3: Vyhledání a logování neexistující zkratky předmětu

Vazby jsou i mezi entitami ze zdrojového systému. Jejich konzistence není řešena. I když entita pro vazbu neexistuje, vazba se stejně vytvoří. Pokud by toto nemělo nastat, muselo by se zajistit správné pořadí vkládání záznamů. Ve zdrojovém systému však existují referenční integrity a při tvorbě ETL transformací s tímto předpokladem bylo počítáno.

6.2.1.4 Smazané entity

SSP udržuje i některé historické hodnoty. Místo mazání záznamů jim jen nastavuje příznak, že je položka smazaná. U stavů pak nastavuje, že je stav zastaralý. Toto se musí řešit při načítání dat, jinak by se zpropagovaly i smazané záznamy. Stačí tedy obyčejná SQL WHERE podmínka, která bere jen záznamy s hodnotou flagu nastavenou na false. Tuto podmínku nestačí používat pouze u hlavních tabulek, ale také u tabulek vztahových a u tabulek, které se smazaným záznamem souvisí. Příkladem jsou tabulky s vícejazyčnými textovými hodnotami. U stavů se tato podmínka nevyužívá, zachovávají se totiž historizované všechny.

6.2.2 Transformace a joby

Tato sekce ze zabývá konkrétními transformacemi pro nahrání dat do integrované vrstvy a nadřazenými joby, které tyto transformace využívají.

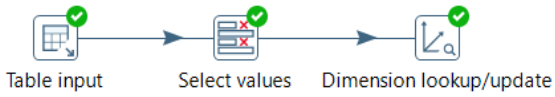
6.2.2.1 Transformace

Každá transformace se skládá z načtení dat, kde se spojí všechny potřebné tabulky, pomocí stepu *Table input*, přejmenování atributů, což zajišťuje step *Select values*, a vložení dat do cílové tabulky se zachováním historizace typu SCD2 díky *Dimension lookup/update* stepu. Mezi to mohou být přidány specifické prvky, které byly popsány v předcházejících sekcích. Z tohoto důvodu zde nebudou rozebrány všechny transformace krok po kroku, uvedeny budou pouze různé typy se souvisejícími tabulkami.

Nejjednodušší transformace se skládají z načtení dat, přejmenování atributů a vložení do cílové tabulky. Ukázka je na obrázku 6.4. Seznam tabulek, které takto vznikají, je následující:

- T_RESITEL_DOVEDNOST_SUBJEKTIVNI_REL
- T_SSP_BETA_DOVEDNOST
- T_SSP_PROJEKT
- T_SSP_PROJEKT_TEXT
- T_SSP_PRUMYSLOVA_INSTITUCE
- T_SSP_PRUMYSLOVA_INSTITUCE_TEXT
- T_SSP_RESITEL
- T_SSP_SPONZOR
- T_SSP_ZADANI_TEXT
- T_ZADANI_OSOBA_EXPERT_REL
- T_ZADANI_RESITEL_ODMENA_REL
- T_ZADANIROLERESITEL_DOVEDNOST_HODNOCENI_REL

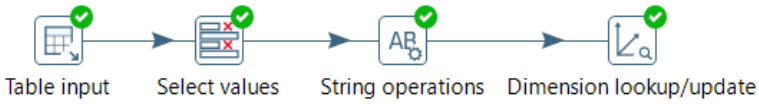
Transformace související s osobami upravují uživatelská jména. Step *String operations* zajistí, že se vloží uživatelské jméno bez počátečních a koncových



Obrázek 6.4: Nejjednodušší transformace

mezer a současně se převede na velká písmena. Transformaci ukazuje obrázek 6.5, používá se pro:

- T_BETADOVEDNOST_OSOBA_REL
- T_BETADOVEDNOST_STUDENT_REL
- T_SSP_OSOBA



Obrázek 6.5: Transformace s úpravou textu

Do některých tabulek se přidává anglický text. Transformace je tedy rozšířena o dva stepy, kde první slouží k nalezení daného textu, druhá k jeho přejmenování. Transformace je na obrázku 6.6, odpovídající tabulky jsou:

- T_SSP_RESITEL_ROLE
- T_SSP_ZADANI_PREDMET_STAV
- T_SSP_ZADANI_ROLE_RESITEL_STAV
- T_SSP_ZADANI_STAV

Obrázek 6.6: Transformace s anglickým textem

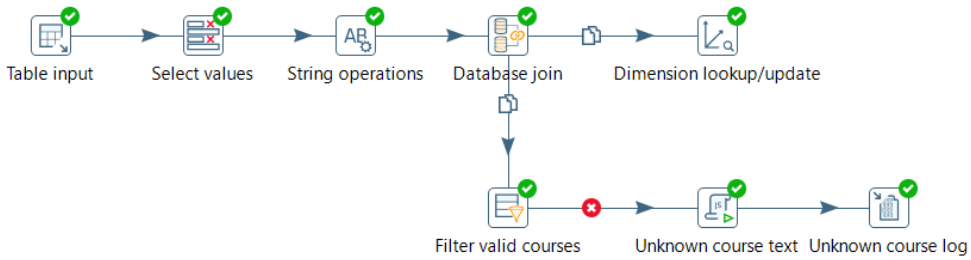
Další rozšíření obsahuje načtení i českého textu, což je stejné jako pro text anglický, a sjednocení formátu názvů. Z nich se odstraní počáteční a koncové mezery a převedou se na malá písmena. Obrázek 6.7 ilustruje takovou transformaci, která se týká následujících tabulek:

- T_SSP_DOVEDNOST
- T_SSP_LICENCE

Tabulka T_BETADOVEDNOST_PREDMET_REL v sobě obsahuje kód předmětu. Nejdříve se z něj odstraní počáteční a koncové mezery a převede

Obrázek 6.7: Transformace s anglickým a českým text a sjednocením názvů

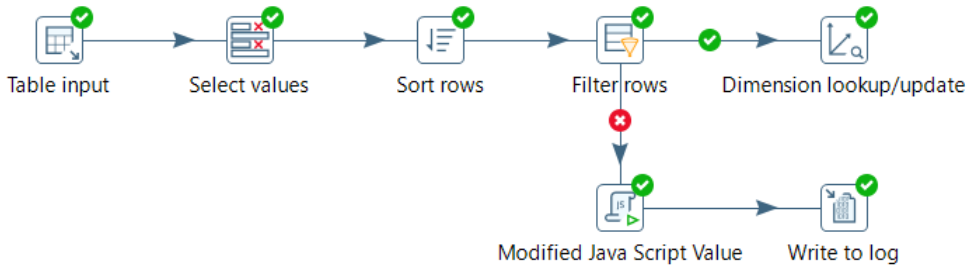
se na velká písmena. To vše pomocí stepu *String operations*. Dále následuje vyhledání takového kódu již v existujících předmětech v datovém skladu díky *Database join*. Nenalezené předměty se poté vyfiltrují a následně zalogují. Transformace je na obrázku 6.8.



Obrázek 6.8: Transformace s kódem předmětu

Další typ souvisí se stavy. Entita se načte se všemi stavy a seřadí se podle data vložení stavu díky *Sort rows*. Pokud je však hodnota takového data prázdná, nastává chyba v transformaci. Proto se tyto hodnoty filtrují stepem *Filter rows*. Stavy s prázdným datem se zalogují, ostatní se vloží do databáze. Vše je ukázáno na obrázku 6.9. Touto transformaci se tvoří následující tabulky:

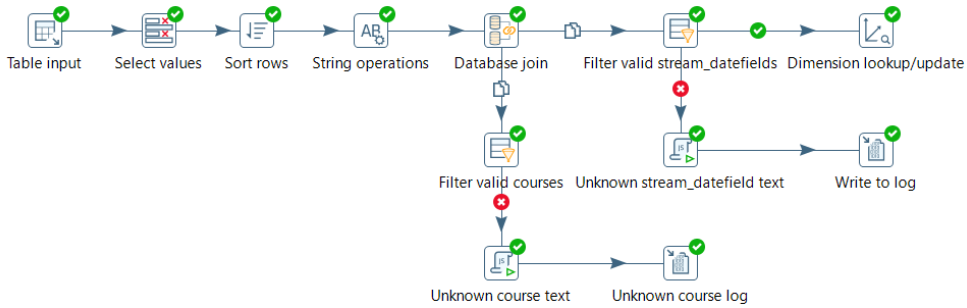
- T_SSP_ZADANI
- T_ZADANI_ROLE_RESITEL_REL



Obrázek 6.9: Transformace s datem vložení záznamu

Zbývající tabulka T_ZADANI_PREDMET_REL je vytvářena kombinací předchozích dvou transformací. Obsahuje totiž kód předmětu, který je nejprve standardizován a následně zalogován, pokud ve skladu takový předmět neexistuje. Současně ale u sebe tabulka uchovává stavy, které jsou řazeny a vkládány

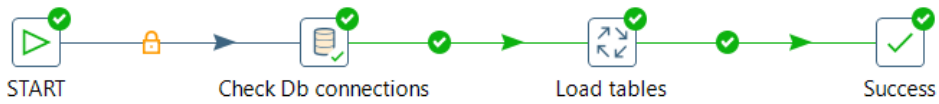
podle data vložení, kde se neexistující datum opět zaloguje. Výslednou transformaci ilustruje obrázek 6.10.



Obrázek 6.10: Transformace s kódem předmětu a datem vložení záznamu

6.2.2.2 Joby

Samotné transformace jsou důležité pro nahrání dat do jednotlivých tabulek. Spouštění těchto transformací postupně ručně není reálné. Proto se používají joby, které mohou mít hierarchickou strukturu. Na nejvyšší úrovni se provádí zkontrolování, jestli jsou všechny databáze přístupné a je možné se k nim připojit. Test je zajištěn pomocí *Check Db connections*. Pokud tento test projde, následuje job, který sdružuje všechny transformace, konkrétně se jedná o job *Load tables*. Hlavní job je na obrázku 6.11.

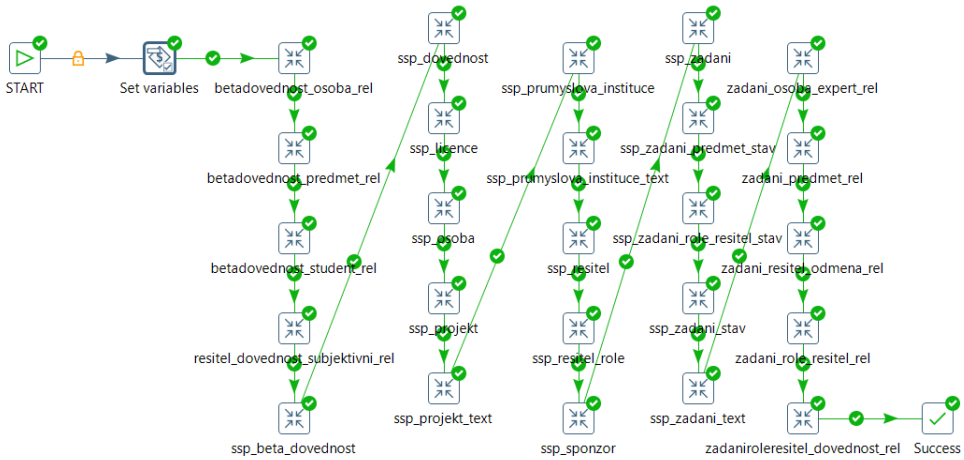


Obrázek 6.11: Hlavní job

Job, který v sobě obsahuje všechny transformace a zajišťuje tím nahrání všech dat, je na obrázku 6.12. Nejdříve step *Set variables* nastaví proměnné prostředí, kterými jsou české a anglické zkratky sloužící pro nahrání odpovídajících textů. Poté následují konkrétní transformace. Jelikož nezáleží na jejich pořadí, jsou seřazené dle abecedy.

6.3 Přístupová vrstva

Implementace přístupové vrstvy kopíruje návrh a skládá se ze dvou částí. V první části je ve stručnosti charakterizována implementace sémantické vrstvy, druhá část se zabývá datovými tržišti.



Obrázek 6.12: Job se všemi transformacemi

6.3.1 Sémantická vrstva

Sémantická vrstva je tvořena jednoduchými databázovými pohledy. Naimplementovány byly všechny entity a jejich vztahy, které byly zmíněny v návrhu. Z tohoto důvodu bude vynechán jejich seznam. Dále bylo nutné vytvořit další vhodné pohledy nad entitami, které ve skladu již existují a jsou potřeba pro vytvoření další vrstvy. Jak ale bylo řečeno v návrhu, tyto pohledy se sjednotí s těmi, které vznikají v rámci paralelně vznikající diplomové práce. Proto je zde uveden pouze výčet těchto pohledů:

- V_BEHPREDMETU_UCITEL_REL
- V_KATEDRA
- V_PREDMET
- V_SEMESTR
- V_STUDENT
- V_STUDIUM
- V_UCITEL
- V_ZAPSANY_PREDMET_S_HODNOCENIM

6.3.2 Datová tržiště

Datamarty přístupové vrstvy tvoří materializované pohledy. Celkem bylo vytvořeno deset datových tržišť. Jelikož jejich základ tvoří faktové tabulky, bude uveden jejich seznam místo názvů datamartů. Názvy odpovídají návrhu.

Následující tržiště čerpají data pouze z SSP. Výjimkou jsou dimenze semestr a předmět, jejich zdrojem jsou entity již existující v datovém skladu. Pokud se jedná o řešitele nebo experty, jsou postaveni nad pohledy, které nejsou spojeny se studenty ani učiteli. Do zmíněných tržišť patří:

- MV_ZADANI_ROLE_RESITEL_DOVEDNOST_HODNOCENI_FACT
- MV_ZADANI_ROLE_RESITEL_HODNOCENI_FACT
- MV_ZADANI_RESITEL_ODMENA_FACT
- MV_ZADANI_RESITEL_HODNOCENI_FACT
- MV_ZADANI_EXPERT_FACT
- MV_ZADANI_PREDMET_FACT

Jelikož betadovednost je spjata se studentem, MV_RESITEL_DOVEDNOSTI_FACT používá spojení studenta a řešitele. Podobně MV_STUDENT_-STATISTIKY_FACT využívá pohled, kde je současně řešitel a student. Pro MV_UCITEL_STATISTIKY_FACT je použito napojení experta a učitele. MV_PREDMET_STATISTIKY_FACT bere v úvahu pouze předměty, které byly někdy nominované k nějakému zadání.

Kapitola 7

Testování

Hlavním cílem práce bylo navržení integrace SSP do datového skladu ČVUT. Jako demonstrace, že navržené řešení funguje, zde bude uvedeno několik reportů z vytvořených datamartů. Reporty slouží pouze pro ilustraci analýz, které jsou nyní díky integraci SSP do skladu možné.

Jako technologické řešení byl použit Pentaho Business Analytics Server, jehož součástí je OLAP server Mondrian. Pro tvorbu reportů byl ve formě pluginu do Pentaho BA Serveru využit nástroj Saiku.

7.1 Reporty

První report slouží pro analýzu vyřešených zadání. Semestry v řádcích znamenají, kdy bylo zadání publikováno, ve sloupcích jsou semestry, kdy bylo zadání vyřešeno. V buňkách tabulky jsou uvedeny počty zadání odpovídající dané kombinaci semestrů. Výsledek je na obrázku 7.1.

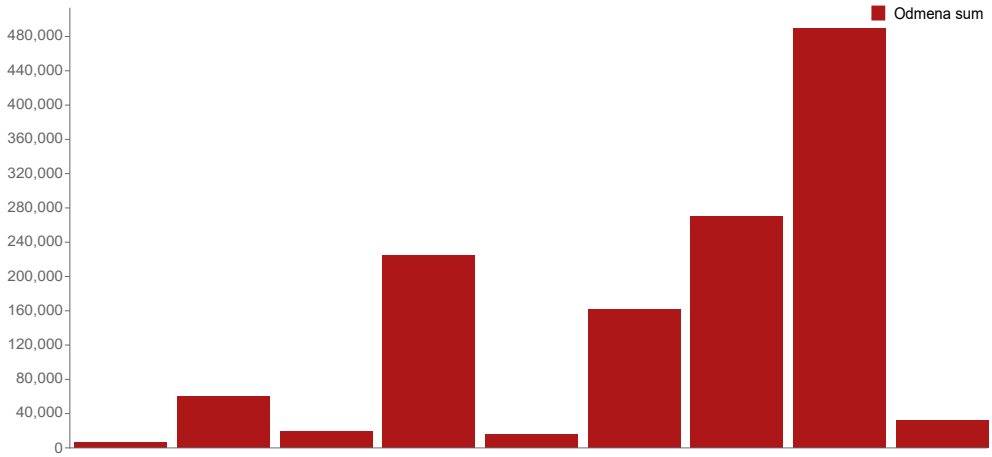
		B131	B141	B142	B151	B152	B161
		Zimní 2013/2014	Zimní 2014/2015	Letní 2014/2015	Zimní 2015/2016	Letní 2015/2016	Zimní 2016/2017
Semestr bk	Oznaceni cs	Zadani count	Zadani count	Zadani count	Zadani count	Zadani count	Zadani count
B131	Zimní 2013/2014	1		1			
B141	Zimní 2014/2015		2	3			1
B142	Letní 2014/2015			1	1	3	
B151	Zimní 2015/2016				1	6	
B152	Letní 2015/2016					8	8

Obrázek 7.1: Vyřešená zadání podle semestrů publikace a vyřešení

Předchozí report se zaměřoval pouze na zadání bez žádného dalšího kontextu a bez jiných statistik. Zajímavějším zdrojem dat jsou vyplacené odměny za zadání. Ty jsou navíc v reportu rozpadnuty přes jednotlivé průmyslové instituce, jejichž zaměstnanci zadání vytvořily. Jelikož se jedná o finanční částku, což je ve velké míře citlivý údaj, byly výše těchto částek náhodně pozměněny. Dále byla vybrána pouze podmnožina institucí a jejich jména byla zanony-

mizována. Jedná se pouze o demonstraci použití. Graf celkové sumy je na obrázku 7.2, kvůli přehlednosti neobsahuje zanonymizovaná jména, report na obrázku 7.3 je podrobnější a ukazuje:

- **Odmena sum** Celkový součet vyplacených odměn
- **Odmena min** Minimální vyplacená odměna
- **Odmena max** Maximální vyplacená odměna
- **Resitel count** Počet řešitelů, kterým byly odměny vyplaceny
- **Odmena resitel avg** Průměrná hodnota odměny za řešitele
- **Zadani count** Počet zadání, za která byly odměny vyplaceny
- **Odmena zadani avg** Průměrná hodnota odměny za zadání



Obrázek 7.2: Graf sumy vyplacených odměn

Nazev	Odmena sum	Odmena min	Odmena max	Resitel count	Odmena resitel avg	Zadani count	Odmena zadani avg
0ea6c47b7ec846f7b8bbab87a04fef39	6 250,00	1 250,00	5 000,00	2	3 125,00	2	3 125,00
156dfb9473bb57d8d25f9d92aa590635	60 000,00	60 000,00	60 000,00	1	60 000,00	1	60 000,00
6175faa9e99cb6404b897d0be311d8ab	19 500,00	2 000,00	15 000,00	3	6 500,00	3	6 500,00
68d7f23be5798f9d112e64551c787264	225 000,00	,00	130 000,00	6	37 500,00	3	75 000,00
69468d4b1df50ecae20a84733a48bff8	16 000,00	16 000,00	16 000,00	1	16 000,00	1	16 000,00
8db52718db510429d75b376b7bab6744	161 500,00	500,00	130 000,00	5	32 300,00	5	32 300,00
98a3339decbba24eeb33ac3ba6344ed9	270 250,00	,00	140 000,00	10	27 025,00	11	24 568,18
9db3f8f197fb4b95725ecae0c0eb318f	490 000,00	20 000,00	180 000,00	4	122 500,00	4	122 500,00
eb2b775b01eb073d473ab1e3ce1d660e	32 000,00	32 000,00	32 000,00	1	32 000,00	1	32 000,00

Obrázek 7.3: Statistiky vyplacených odměn

Při návrhu datamartů bylo řečeno, že jejich hlavním zaměřením jsou studenti, učitelé a předměty. Report rozpadlý dle průmyslových institucí demonstroval, že tyto entity nejsou jedinými, pro které lze vytvářet analýzy. Další reporty se ale budou zabývat už jen výše zmíněnými a s nimi souvisejícími entitami. Proto jsou na obrázku 7.4 ukázána hodnocení dovedností za zadání. Hodnocení je pro autora a pro jednoho dalšího řešitele, vzhledem k údajům

byly uživatelská jména v řádcích anonymizována. Sloupce znázorňují semestry a dovednosti. Hodnoty znamenají:

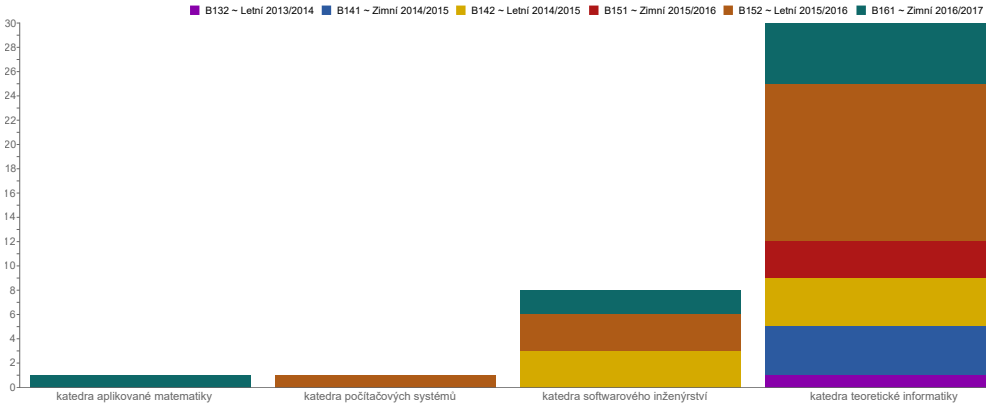
- **Hodnoceni** Průměrné hodnocení dovednosti (v tomto případě se jedná pouze o jedno)
- **Zadani count** Počet zadání, za která bylo alespoň jedno hodnocení dovednosti uděleno

	Letní 2014/2015				Letní 2015/2016					
	B142				B152					
	datamining		programming		datamining		pattern recognition		programming	
Username id	Hodnoceni	Zadani count	Hodnoceni	Zadani count	Hodnoceni	Zadani count	Hodnoceni	Zadani count	Hodnoceni	Zadani count
a72692d54e7ae7f761a6e117216ff931	,60	1	1,00	1	,80	1	1,00	1		
82bab2626262aff36b96c1ec120544a7					,80	1			,80	1

Obrázek 7.4: Hodnocení dovedností za zadání

Zadání jsou nominovaná do předmětů. Jelikož není jisté, jestli bylo zadání do nominovaného předmětu odevzdáno, graf 7.5 ukazuje počty vyřešených zadání, která byla do předmětu pouze úspěšně nominována, za všechny semestry. Rozpad je tvořen přes katedry, do kterých předmět spadá. Obrázek 7.6 ukazuje více statistik takových zadání za poslední dva semestry, konkrétně:

- **Zadani count** Počet vyřešených zadání, jejichž nominace byla do předmětu v katedře přijata
- **Hodnoceni min** Minimální hodnocení řešení zadání, jejichž nominace byla do předmětu v katedře přijata
- **Hodnoceni max** Maximální hodnocení řešení zadání, jejichž nominace byla do předmětu v katedře přijata
- **Hodnoceni avg** Průměrné hodnocení řešení zadání, jejichž nominace byla do předmětu v katedře přijata



Obrázek 7.5: Graf počtů vyřešených zadání nominovaných do předmětů dle kateder

	B152				B161			
	Letní 2015/2016				Zimní 2016/2017			
Nazev cs	Zadani count	Hodnoceni min	Hodnoceni max	Hodnoceni avg	Zadani count	Hodnoceni min	Hodnoceni max	Hodnoceni avg
katedra aplikované matematiky					1	1,00	1,00	1,00
katedra počítačových systémů	1	1,00	1,00	1,00				
katedra softwarového inženýrství	3	0,80	1,00	0,93	2	1,00	1,00	1,00
katedra teoretické informatiky	13	0,00	1,00	0,89	5	0,80	1,00	0,96

Obrázek 7.6: Statistiky vyřešených zadání nominovaných do předmětů dle kateder za poslední dva semestry

V rámci práce byly navrženy a vytvořeny datamarty, které sdružují statistiky i z jiných datových zdrojů. Následující report, který ukazuje obrázek 7.7, obsahuje studijní výsledky autora této práce za ukončené semestry magisterského studia spolu se statistikami za vyřešená zadání. Vypracováno bylo pouze jediné zadání, ale i to pro ilustrování kombinace statistik postačuje. Navíc bylo velmi snadné ověřit pravdivost těchto údajů. Konkrétní statistiky jsou:

- **Zapsane predmety count** Počet zapsaných předmětů
- **Dokoncene predmety count** Počet absolvovaných předmětů
- **Klasifikovane predmety count** Počet absolvovaných předmětů zakončených zkouškou
- **Absolvovane / zapsane** Poměr absolvovaných vůči zapsaným předmětům
- **Prumer** Průměr za absolvované předměty zakončené zkouškou
- **Reseni count** Počet vyřešených zadání
- **Hodnoceni reseni avg** Průměrné hodnocení řešení zadání

Semestr bk	Oznaceni cs	Zapsane predmety count	Dokoncene predmety count	Klasifikovane predmety count	Absolvovane / zapsane	Prumer	Reseni count	Hodnoceni reseni avg
B141	Zimní 2014/2015	7	7	7	1,00	1,00		
B142	Letní 2014/2015	8	8	8	1,00	1,13		
B151	Zimní 2015/2016	7	7	6	1,00	1,17		
B152	Letní 2015/2016	5	4	4	0,80	1,00	1	1,00
B161	Zimní 2016/2017	5	5	4	1,00	1,25		

Obrázek 7.7: Statistiky autora za ukončené semestry magisterského studia

Podobné statistiky byly na reportu 7.8 vytvořeny i pro učitele. Nejedná se o studijní výsledky, ale o počty předmětů, který daný člověk učil. Z dat z SSP jsou uvedeny počty zadání, na kterých spolupracoval jako expert a která byla úspěšně vyřešena. Tyto informace mohou být považovány za citlivé, proto není uvedeno jméno učitele. Seznam konkrétních statistik je následující:

- **Cvicici** Počet předmětů, kde figuroval jako cvičící
- **Prednasejici** Počet předmětů, kde figuroval jako přednášející
- **Zkousejici** Počet předmětů, kde figuroval jako zkoušející
- **Garant** Počet předmětů, kde figuroval jako garant

- **Nevyresena zadani** Počet nevyřešených zadání zbývajících v daném semestru, u kterých figuroval jako expert
- **Vyresena zadani** Počet vyřešených zadání v daném semestru, u kterých figuroval jako expert

Semestr bk	Oznaceni cs	Cvicici	Prednasejici	Zkousejici	Garant	Nevyresena zadani	Vyresena zadani
B141	Zimní 2014/2015	6	3	3	4	4	0.00
B142	Letní 2014/2015	3	2	2	5	1	3
B151	Zimní 2015/2016	5	4	4	5	1	0.00
B152	Letní 2015/2016	3	2	2	5	4	0.00
B161	Zimní 2016/2017	7	6	6	8	3	1

Obrázek 7.8: Statistiky anonymního učitele

Poslední report opět sdružuje statistiky z několika systémů, tentokrát se však jedná o předmět. Bylo zvoleno Vytěžování znalostí z dat jako bakalářský a Metody výpočetní inteligence jako magisterský předmět. Jedná se o předměty související se znalostním inženýrstvím a spadající pod katedru teoretické informatiky, která měla v jedné z předchozích analýz nejvíce vyřešených zadání nominovaných do jejich předmětů. Z SSP opět pochází statistiky za zadání, která byla nominována k předmětu a zároveň byla vyřešena. Dalšími informacemi jsou počty studentů k předmětu. Celkově report 7.9 zobrazuje:

- **Zapsani** Počet studentů, kteří si zapsali předmět
- **Ukonceni** Počet studentů, kteří úspěšně absolvovali předmět
- **Nevyresena zadani** Počet nevyřešených zadání zbývajících v daném semestru, jejichž nominace byla do předmětu přijata
- **Vyresena zadani** Počet vyřešených zadání v daném semestru, jejichž nominace byla do předmětu přijata

Nazev cs	Oznaceni cs	Zapsani	Ukonceni	Nevyresena zadani	Vyresena zadani
Algoritmy data miningu	Letní 2013/2014	34	10	7	0.00
	Letní 2014/2015	45	22	26	3
	Letní 2015/2016	67	24	50	6
Vytěžování znalostí z dat	Letní 2013/2014	111	78	3	0.00
	Letní 2014/2015	95	55	18	2
	Letní 2015/2016	92	73	31	3

Obrázek 7.9: Statistiky BI-VZD a MI-ADM

7.2 Zhodnocení výsledků

Reporty v předchozí sekci ilustrovaly funkčnost celého řešení. Výhodou použité technologie OLAP a dimenzionálního modelování je zpřístupnění dat pro analytické dotazy business uživatelům. Ti k tomu nepotřebují technické vzdělání, maximálně proškolení v daném BI nástroji. Uživatelé jsou schopni pracovat s daty z různých úhlů pohledu, mohou vytvářet podmínky a data různě agregovat. Jednoduchost použití se odvíjí od zvoleného nástroje. V případě analýzy nad existujícími datamarty nejsou uživatelé závislí na týmu IT specialistů a interaktivní práce s daty může probíhat individuálně bez jakékoliv pomoci. Data jsou navíc získávána z jednoho centralizovaného úložiště, čímž je zajištěna konzistence získávaných informací. Umožněno je také provádění analýz nad daty pocházejícími z několika různých systémů současně, což by bez správné architektury datového skladu bylo velmi těžko proveditelné. Uchování dat v samostatném úložišti navíc nijak nezatěžuje zdrojové systémy při vykonávání analytických dotazů.

Největší nevýhodou je náročnost vytvoření návrhu a implementace takového řešení. Datový sklad je komplexní a skládá se z několika vrstev. Z toho vyplývá, že jakákoliv změna se může promítnout do všech vrstev. Úprava je kvůli tomu náročná na lidské zdroje a datový sklad obecně pomalu reaguje na změny.

Závěr

V úvodu byl zmíněn cíl práce, kterým bylo navrhnout a implementovat integraci dat Portálu spolupráce s průmyslem do datového skladu ČVUT. Jelikož již datový sklad na ČVUT existuje, byla brána v potaz jeho architektura a existující integrace dalších zdrojových systémů. Příkladem je dodržování jmenné konvence a použití integračního nástroje Pentaho Data Integration. Návrh a implementace využívaly existující vrstvy tam, kde to bylo možné. V ostatních případech bylo explicitně zmíněno, že sjednocení bude muset nastat po dokončení paralelně vyvíjených prací věnujících se dané problematice. Konkrétním příkladem jsou některé pohledy v sémantické vrstvě.

Jelikož má SSP potenciál se do budoucna vyvíjet, byly v práci navrženy nebo diskutovány některé budoucí úpravy. Jedná se např. o budoucí sjednocení dovedností. Ty jsou bohužel v současné době tvořeny dvěma nekonzistentními sadami, což se muselo promítnout v návrhu jejich struktur. Dále byly v některých případech uvedeny i příklady špatného návrhu či implementace, text práce díky tomu může sloužit jako podklad pro budoucí vývoj a vyvarování se chyb. V tomto případě se jedná např. o ukázku chyb u návrhů některých datových tržišť, kde byly současně uvedeny i jejich příčiny a navržena správná řešení takových problémů.

V práci byly vytvořeny návrhy všech vrstev datového skladu, jmenovitě stage, integrované vrstva a přístupová vrstva skládající se ze sémantické vrstvy a datových tržišť. Tomu odpovídá i implementace jednotlivých vrstev. Funkčnost celkového řešení demonstrovaly reporty postavené nad implementovanými datamarty. Některé z nich zobrazují data pouze z SSP, jiné k tomu přidávají informace z ostatních systémů, čímž ilustrují úspěšnost vytvořené integrace.

Zadání mělo několik podúkolů. Všechny úkoly a požadavky, některé z nich vyvstaly až v průběhu, byly splněny. Proto považuji i hlavní cíl za splněný.

Osobním přínosem byla práce na reálném projektu v prostředí univerzity. Velice oceňuji i spolupráci s kolegy, kteří spravují a vyvíjí datový sklad. Od nich jsem získal plno nových znalostí a dovedností.

Literatura

[1] Rangarajan, S.: Data Warehouse Design – Inmon vs Kimball Architecture. 2016, [cit. 2017-01-28]. Dostupné z: http://infosolblog.com/data-warehouse-design-inmon-vs-kimball-architecture/

[2] Inmon, W. H.: *Building the Data Warehouse*. Wiley Publishing, Inc., čtvrté vydání, 2005, ISBN 978-0-7645-9944-6.

[3] Kimball, Ralph a Margy Ross: *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, Inc., druhé vydání, 2002, ISBN 978-0-471-20024-6.

[4] Kimball, Ralph a Margy Ross: *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley & Sons, Inc., třetí vydání, 2013, ISBN 978-1-118-53080-1.

[5] Ambler, S. W.: Introduction to Data Normalization: A Database "Best" Practice. [cit. 2017-02-18]. Dostupné z: http://agiledata.org/essays/dataNormalization.html

[6] Janies, L.: Easy in, easy out. 2008, [cit. 2017-02-22]. Dostupné z: http://apps.teradata.com/TDMO/v08n04/FactsandFun/PartnerConnection/Easy.aspx

[7] ZenTut.com: *Kimball vs. Inmon Data Warehouse Architectures* [online]. [cit. 2017-01-29]. Dostupné z: http://www.zentut.com/data-warehouse/kimball-and-inmon-data-warehouse-architectures/

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

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

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

[11] Štádler, M.: *Integrace V3S do datového skladu ČVUT*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií, Praha, 2017.

[12] Power, D. J.: A Brief History of Decision Support Systems. *DSSResources.COM*, říjen 2007, [cit. 2017-01-26]. Dostupné z: `http://dssresources.com/history/dsshistory.html`

[13] Inmon, W. H.: *Building the Data Warehouse*. John Wiley & Sons, Inc., první vydání, 1992, ISBN 978-0-471569-60-2.

[14] Inmon, W. H.: A Tale of Two Architectures. *InmonCif.com*, 2010, [cit. 2017-01-27]. Dostupné z: `http://www.inmoncif.com/products/ATALEOFTWOARCHITECTURES.pdf`

[15] Inmon, W. H.: Data Mart Does Not Equal Data Warehouse. 1999, [cit. 2017-01-28]. Dostupné z: `http://www.information-management.com/infodirect/19991120/1675-1.html`

[16] George, S.: Inmon vs. Kimball: Which approach is suitable for your data warehouse? 2012, [cit. 2017-01-27]. Dostupné z: `http://www.computerweekly.com/tip/Inmon-vs-Kimball-Which-approach-is-suitable-for-your-data-warehouse`

[17] Kimball, R.: *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*. John Wiley & Sons, Inc., první vydání, 1996, ISBN 978-0-471-15337-5.

[18] Russo, E.: Working With Big Data: What Is a Landing Area? [cit. 2017-02-17]. Dostupné z: `https://www.datavail.com/blog/what-is-a-landing-area-with-big-data/`

[19] Inmon, W. H.: The Operational Data Store. 1998, [cit. 2017-02-03]. Dostupné z: `http://www.information-management.com/issues/19980701/469-1.html`

[20] Poolet, M. A.: SQL by Design: Why You Need Database Normalization. 1999, [cit. 2017-02-18]. Dostupné z: `http://sqlmag.com/database-performance-tuning/sql-design-why-you-need-database-normalization`

[21] SearchDataBackup: *Full, incremental or differential: How to choose the correct backup type*. [online], srpen 2008, [cit. 2017-04-24]. Dostupné z: `http://searchdatabackup.techtarget.com/feature/`

Full-incremental-or-differential-How-to-choose-the-correct-backup-type

[22] Kimball, Ralph a Joe Caserta: *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley Publishing, Inc., první vydání, 2004, ISBN 978-0-7645-6757-5.

[23] Becker, B.: Subsystems of ETL Revisited. 2007, [cit. 2017-02-21]. Dostupné z: http://www.kimballgroup.com/2007/10/subsystems-of-etl-revisited/

[24] Kajeepeta, S.: Is it Time to Switch to ELT? 2010, [cit. 2017-02-22]. Dostupné z: https://enterpriseinformationmanagement.wordpress.com/2010/06/08/is-it-time-to-switch-to-elt/

[25] Novotný, O., J. Pour a D. Slánský: *Business Intelligence*. Grada Publishing, a.s., první vydání, 2005, ISBN 80-247-1094-3.

[26] Kimball, R.: Slowly Changing Dimensions. 2008, [cit. 2017-02-24]. Dostupné z: http://www.kimballgroup.com/2008/08/slowly-changing-dimensions/

[27] Kimball, R.: Slowly Changing Dimensions, Part 2. 2008, [cit. 2017-02-24]. Dostupné z: http://www.kimballgroup.com/2008/09/slowly-changing-dimensions-part-2/

[28] Ross, M.: Slowly Changing Dimensions Are Not Always as Easy as 1, 2, 3. 2005, [cit. 2017-02-24]. Dostupné z: http://www.kimballgroup.com/2005/03/slowly-changing-dimensions-are-not-always-as-easy-as-1-2-3/

[29] *Portál SSP* [online]. [cit. 2017-03-11]. Dostupné z: https://ssp.fit.cvut.cz/

[30] Kordík, P.: *Portál spolupráce s průmyslem (SSP)* [online]. [cit. 2017-03-11]. Dostupné z: https://wiki.cvut.cz/confluence/pages/viewpage.action?pageId=14057588

[31] Kordík, Pavel a Stanislav Kuznetsov: Mining Skills from Educational Data for Project Recommendations. In *International Joint Conference*, Springer, 2015, s. 617–627.

[32] Kuznetsov, Stanislav et al.: Reducing cold start problems in educational recommender systems. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, IEEE, 2016, s. 3143–3149.

Příloha **A**

Seznam použitých zkratek

1NF First Normal Form

2NF Second Normal Form

3NF Third Normal Form

BI Business Intelligence

CIF Corporate Information Factory

ČVUT České vysoké učení technické

DSS Decision Support Systems

DWH Data Warehouse

EDW Enterprise Data Warehouse

ETL Extract, Transform, Load

ELT Extract, Load, Transform

FIT Fakulta informačních technologií

NDA Non Disclosure Agreement

ODS Operational Data Store

OLTP Online Transaction Processing

SCD Slowly Changing Dimensions

SSP Spolupráce s průmyslem

Příloha **B**

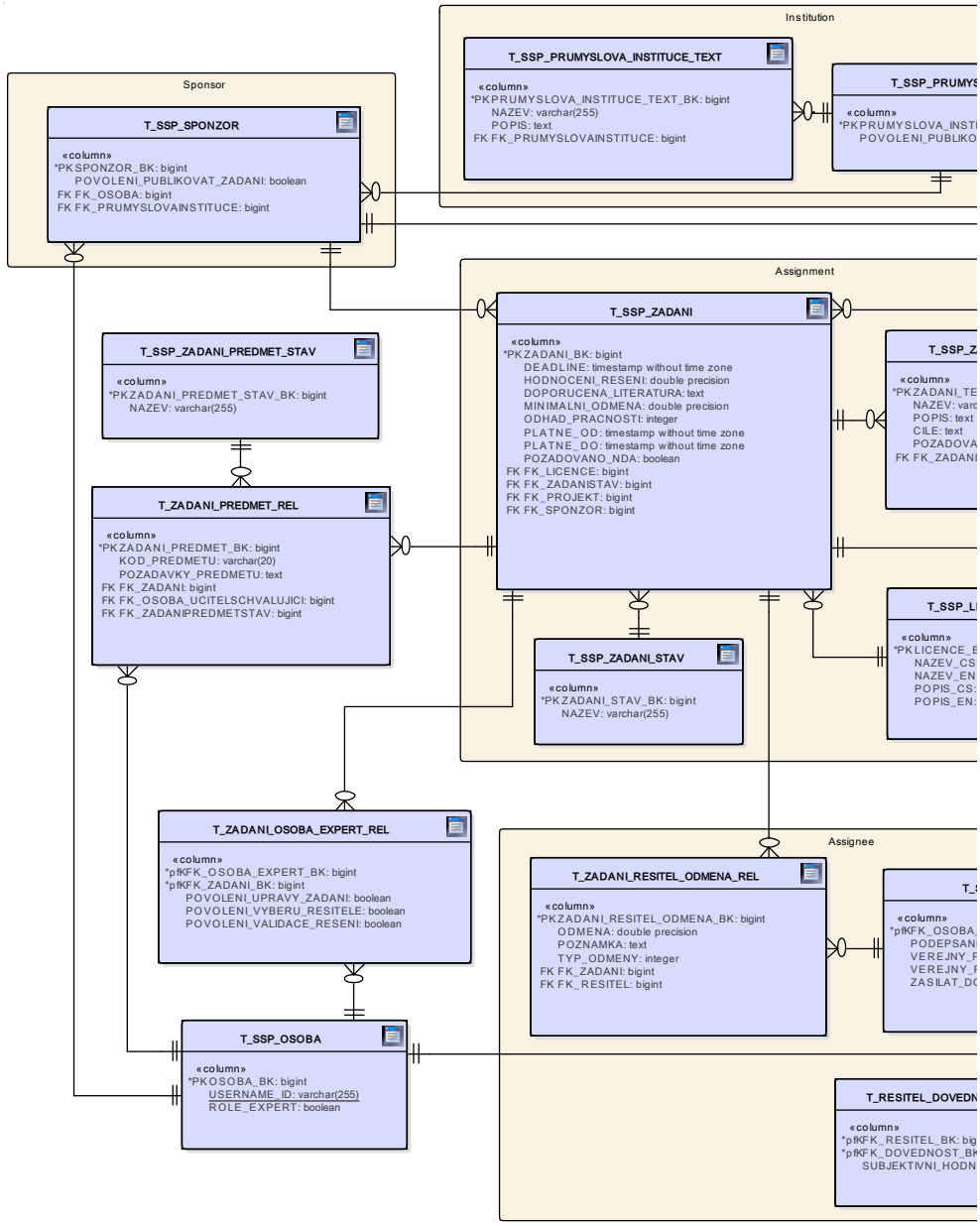
Obsah přiloženého CD

```
readme.txt ..................................stručný popis obsahu CD
src
  impl ..................................zdrojové kódy implementace
    etl ....................................soubory s ETL procesy
    sql ....................................soubory s SQL skripty
  thesis ....................... zdrojová forma práce ve formátu LaTeX
text ..................................................... text práce
  DP_Mikes_Jan_2017.pdf ................text práce ve formátu PDF
```

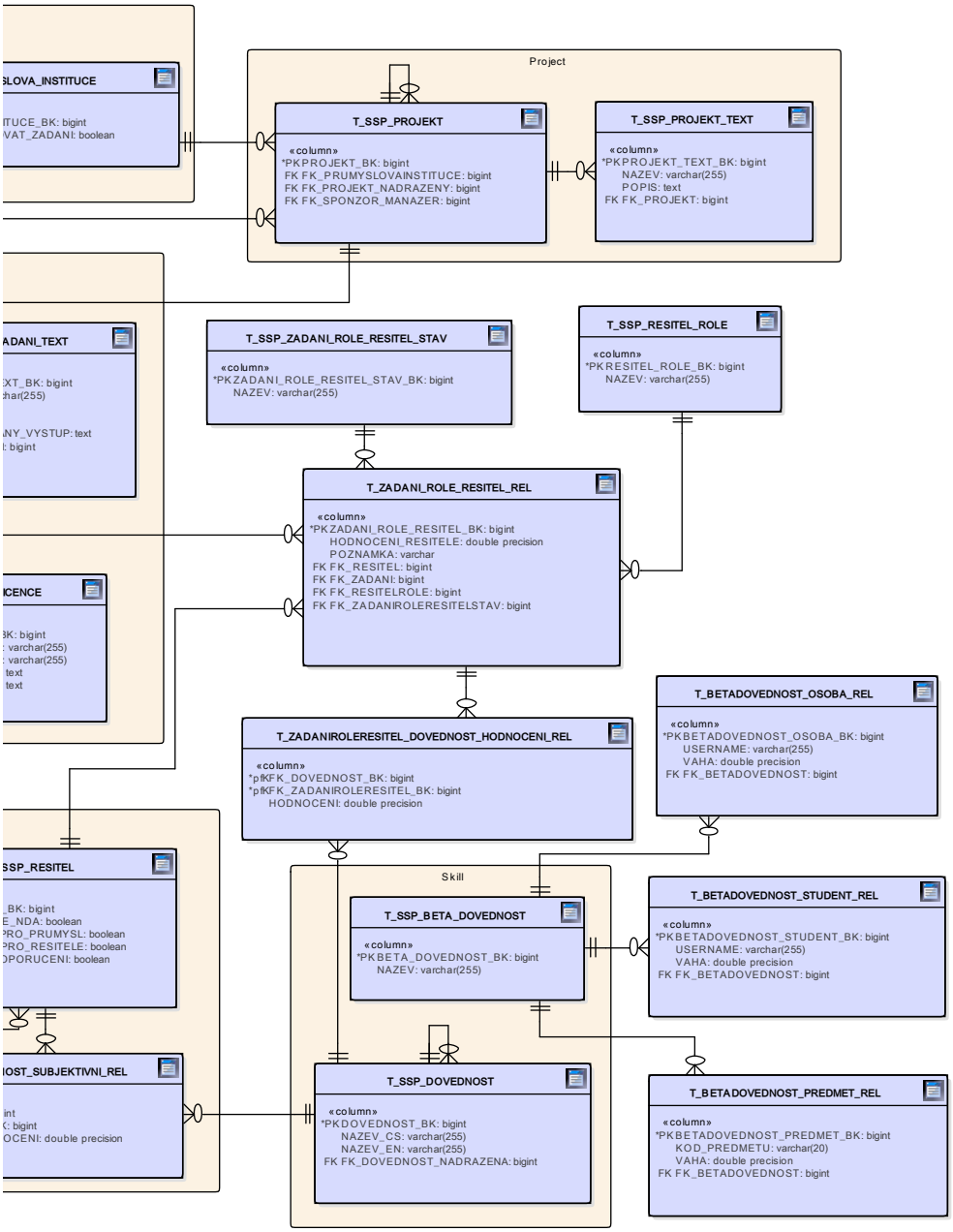
Příloha C

Schéma integrované vrstvy

Diagram je vzhledem k jeho velikosti umístěn na následující dvoustraně.



Obrázek C.1: Schéma integrované vrstvy


Project
T_SSP_PROJEKT
«column»
*PK PROJEKT_BK: bigint
FK FK_PRUMYSLOVAINSTITUCE: bigint
FK FK_PROJEKT_NADRAZENY: bigint
FK FK_SPONZOR_MANAZER: bigint
T_SSP_PROJEKT_TEXT
«column»
*PK PROJEKT_TEXT_BK: bigint
NAZEV: varchar(255)
POPIS: text
FK FK_PROJEKT: bigint
T_SSP_ZADANI_ROLE_RESITEL_STAV
«column»
*PK ZADANI_ROLE_RESITEL_STAV_BK: bigint
NAZEV: varchar(255)
T_SSP_RESITEL_ROLE
«column»
*PK RESITEL_ROLE_BK: bigint
NAZEV: varchar(255)
T_ZADANI_ROLE_RESITEL_REL
«column»
*PK ZADANI_ROLE_RESITEL_BK: bigint
HODNOCENI_RESITELE: double precision
POZNAMKA: varchar
FK FK_RESITEL: bigint
FK FK_ZADANI: bigint
FK FK_RESITELROLE: bigint
FK FK_ZADANIROLERESITELSTAV: bigint
T_BETADOVEDNOST_OSOBA_REL
«column»
*PK BETADOVEDNOST_OSOBA_BK: bigint
USERNAME: varchar(255)
VAHA: double precision
FK FK_BETADOVEDNOST: bigint
T_ZADANIROLERESITEL_DOVEDNOST_HODNOCENI_REL
«column»
*pfK FK_DOVEDNOST_BK: bigint
*pfK FK_ZADANIROLERESITEL_BK: bigint
HODNOCENI: double precision
Skill
T_SSP_BETA_DOVEDNOST
«column»
*PK BETA_DOVEDNOST_BK: bigint
NAZEV: varchar(255)
T_BETADOVEDNOST_STUDENT_REL
«column»
*PK BETADOVEDNOST_STUDENT_BK: bigint
USERNAME: varchar(255)
VAHA: double precision
FK FK_BETADOVEDNOST: bigint
T_SSP_DOVEDNOST
«column»
*PK DOVEDNOST_BK: bigint
NAZEV_CS: varchar(255)
NAZEV_EN: varchar(255)
FK FK_DOVEDNOST_NADRAZENA: bigint
T_BETADOVEDNOST_PREDMET_REL
«column»
*PK BETADOVEDNOST_PREDMET_BK: bigint
KOD_PREDMETU: varchar(20)
VAHA: double precision
FK FK_BETADOVEDNOST: bigint

Příloha D

Mapování dat integrované vrstvy

Mapování dat mezi zdrojovými databázemi a integrovanou vrstvou je uvedeno v následujících tabulkách. Tabulky beta_skill, beta_course_skill, beta_student_skill a beta_user_skill jsou součástí podpůrné BI databáze pro SSP, ostatní pochází z databáze hlavní.

Tabulka D.1: Mapování dat - T_ZADANIROLERESITEL_DOVEDNOST_-HODNOCENI_REL

Cílový atribut	**Zdrojová tabulka**	**Zdrojový atribut**
FK_DOVEDNOST_BK	assigneeskillevaluation	skillid
FK_ZADANIROLERESITEL_BK	assigneeskillevaluation	assignmentroleassigneeid
HODNOCENI	assigneeskillevaluation	rating

Tabulka D.2: Mapování dat - T_ZADANI_ROLE_RESITEL_REL

Cílový atribut	**Zdrojová tabulka**	**Zdrojový atribut**
ZADANI_ROLE_RESITEL_BK	assignmentroleassignee	assignmentroleassigneeid
HODNOCENI_RESITELE	assigneeevaluation (Vazba přes assigneeevaluationid.)	rating
POZNAMKA	assignmentroleassignee	message
FK_RESITEL	assignmentroleassignee	assigneeid
FK_ZADANI	assignmentrole (Vazba přes assignmentroleid.)	assignmentid
FK_ZADANIROLE	assignmentrole (Vazba přes assignmentroleid.)	assignmentrolenameid
FK_ZADANIROLERESITELSTAV	assigneestate (Vazba přes assignmentroleassigneeid.)	assigneestatenameid

Tabulka D.3: Mapování dat - T_ZADANI_RESITEL_ODMENA_REL

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
ZADANI_RESITEL_ODMENA_-BK	assigneereward	assigneerewardid
ODMENA	assigneereward	amount
POZNAMKA	assigneereward	note
TYP_ODMENY	assigneereward	rewardtype
FK_ZADANI	assigneereward	assignmentid
FK_RESITEL	assigneereward	assigneeid

Tabulka D.4: Mapování dat - T_ZADANI_PREDMET_REL

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
ZADANI_PREDMET_BK	courseassignment	courseassignmentid
KOD_PREDMETU	course (Vazba přes courseid.)	coursecode
POZADAVKY_PREDMETU	courseassignment	courserequirements
FK_ZADANI	courseassignment	assignmentid
FK_OSOBA_UCITELSCHVALU-JICI	teacher (Vazba přes teacherid.)	sspuserid
FK_ZADANIPREDMETSTAV	courseassignmentstate (Vazba přes courseassignmen-tid.)	courseassignmentstatenameid

Tabulka D.5: Mapování dat - T_ZADANI_OSOBA_EXPERT_REL

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
FK_OSOBA_EXPERT_BK	expert (Vazba přes expertid.)	sspuser_id
FK_ZADANI_BK	expertassignment	assignmentid
POVOLENI_UPRAVY_ZADANI	expertassignment	editdraftpermission
POVOLENI_VYBERU_RESITELE	expertassignment	choosesolverspermission
POVOLENI_VALIDACE_RESENI	expertassignment	solutionvalidationpermission

Tabulka D.6: Mapování dat - T_SSP_ZADANI_TEXT

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
ZADANI_TEXT_BK	assignmenttext	assignmenttextid
NAZEV	assignmenttext	title
POPIS	assignmenttext	description
CILE	assignmenttext	goals
POZADOVANY_VYSTUP	assignmenttext	requiredoutputs
FK_ZADANI	assignmenttext	assignmentid

Tabulka D.7: Mapování dat - T_SSP_ZADANI_STAV

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
ZADANI_STAV_BK	assignmentstatename	assignmentstatenameid
NAZEV	assignmentstatenametext (Vazba přes assignmentstatena-meid, anglický text.)	name

Tabulka D.8: Mapování dat - T_SSP_ZADANI_ROLE_RESITEL_STAV

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
ZADANI_ROLE_RESITEL_-STAV_BK	assigneestatename	assigneestatenameid
NAZEV	assigneestatenametext (Vazba přes assigneestatena-meid, anglický text.)	name

Tabulka D.9: Mapování dat - T_SSP_RESITEL_ROLE

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
RESITEL_ROLE_BK	assignmentrolename	assignmentrolenameid
NAZEV	assignmentrolenametext (Vazba přes assignmentrolenameid, anglický text.)	name

Tabulka D.10: Mapování dat - T_SSP_ZADANI_PREDMET_STAV

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
ZADANI_PREDMET_STAV_BK	courseassignmentstatename	courseassignmentstatenameid
NAZEV	courseassignmentstatenametext (Vazba přes courseassignmentstatenameid, anglický text.)	name

Tabulka D.11: Mapování dat - T_SSP_ZADANI

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
ZADANI_BK	assignment	assignmentid
DEADLINE	assignment	deadline
HODNOCENI_RESENI	solutionevaluation (Vazba do solution přes solutionid, dále přes solutionevaluationid.)	rating
DOPORUCENA_LITERATURA	assignment	literature
MINIMALNI_ODMENA	assignment	reservedpayoff
ODHAD_PRACNOSTI	assignment	timeestimate
PLATNE_OD	assignment	validfrom
PLATNE_DO	assignment	validto
POZADOVANO_NDA	assignment	assignee_must_sign_nda
FK_LICENCE	assignmentipmode (Vazba přes assignmentid.)	ipmodeid
FK_ZADANISTAV	assignmentstate (Vazba přes assignmentid.)	assignmentstatenameid
FK_PROJEKT	assignment	projectid
FK_SPONZOR	assignment	sponsorid

Tabulka D.12: Mapování dat - T_SSP_SPONZOR

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
SPONZOR_BK	sponsor	sponsorid
POVOLENI_PUBLIKOVAT_ZADANI	sponsor	publishassignmentspermission
FK_OSOBA	sponsor	sspuserid
FK_PRUMYSLOVAINSTITUCE	sponsor	institutionid

Tabulka D.13: Mapování dat - T_SSP_RESITEL

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
FK_OSOBA_BK	assignee	sspuserid
PODEPSANE_NDA	assignee	nda_signed
VEREJNY_PRO_PRUMYSL	assignee	public_for_industry
VEREJNY_PRO_RESITELE	assignee	public_for_assignees
ZASILAT_DOPORUCENI	assignee	sendrecommendations

Tabulka D.14: Mapování dat - T_SSP_PRUMYSLOVA_INSTITUCE_-TEXT

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
PRUMYSLOVA_INSTITUCE_-TEXT_BK	institutiontext	institutionidtextid
NAZEV	institutiontext	name
POPIS	institutiontext	description
FK_PRUMYSLOVAINSTITUCE	institutiontext	institutionid

Tabulka D.15: Mapování dat - T_SSP_PRUMYSLOVA_INSTITUCE

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
PRUMYSLOVA_INSTITUCE_BK	institution	institutionid
POVOLENI_PUBLIKOVAT_ZADANI	institution	publishassignmentspermission

Tabulka D.16: Mapování dat - T_SSP_PROJEKT_TEXT

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
PROJEKT_TEXT_BK	projecttext	projecttextid
NAZEV	projecttext	title
POPIS	projecttext	description
FK_PROJEKT	projecttext	projectid

Tabulka D.17: Mapování dat - T_SSP_PROJEKT

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
PROJEKT_BK	project	projectid
FK_PRUMYSLOVAINSTITUCE	project	institutionid
FK_PROJEKT_NADRAZENY	project	parentprojectid
FK_SPONZOR_MANAZER	project	projectmanagerid

Tabulka D.18: Mapování dat - T_SSP_OSOBA

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
OSOBA_BK	sspuser	sspuserid
USERNAME_ID	sspuser	username
ROLE_EXPERT	expert (True, pokud existuje expert s odpovídajícím sspuserid.)	sspuserid

Tabulka D.19: Mapování dat - T_SSP_LICENCE

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
LICENCE_BK	ipmode	ipmodeid
NAZEV_CS	ipmodetext (Vazba přes ipmodeid, český text.)	name
NAZEV_EN	ipmodetext (Vazba přes ipmodeid, anglický text.)	name
POPIS_CS	ipmodetext (Vazba přes ipmodeid, český text.)	description
POPIS_EN	ipmodetext (Vazba přes ipmodeid, anglický text.)	description

Tabulka D.20: Mapování dat - T_SSP_DOVEDNOST

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
DOVEDNOST_BK	skill	skillid
NEZEV_CS	skilltext (Vazba přes skillid, český text.)	name
NAZEV_EN	skilltext (Vazba přes skillid, anglický text.)	name
FK_DOVEDNOST_NADRAZENA	skill	parentskillid

Tabulka D.21: Mapování dat - T_SSP_BETA_DOVEDNOST

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
BETA_DOVEDNOST_BK	beta_skill	skillid
NAZEV	beta_skill	name

Tabulka D.22: Mapování dat - T_RESITEL_DOVEDNOST_SUBJEKTIVNI_REL

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
FK_RESITEL_BK	assigneesubjectiveskill	assigneeid
FK_DOVEDNOST_BK	assigneesubjectiveskill	skillid
SUBJEKTIVNI_HODNOCENI	assigneesubjectiveskill	rating

Tabulka D.23: Mapování dat - T_BETADOVEDNOST_STUDENT_REL

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
BETADOVEDNOST_STUDENT_-BK	beta_student_skill	student_skill_id
USERNAME	beta_student_skill	username
VAHA	beta_student_skill	weight
FK_BETADOVEDNOST	beta_student_skill	skill_id

Tabulka D.24: Mapování dat - T_BETADOVEDNOST_PREDMET_REL

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
BETADOVEDNOST_PREDMET_-BK	beta_course_skill	beta_course_skill_id
KOD_PREDMETU	beta_course_skill	course_code
VAHA	beta_course_skill	weight
FK_BETADOVEDNOST	beta_course_skill	skillid

Tabulka D.25: Mapování dat - T_BETADOVEDNOST_OSOBA_REL

Cílový atribut	Zdrojová tabulka	Zdrojový atribut
BETADOVEDNOST_OSOBA_BK	beta_user_skill	user_skill_id
USERNAME	beta_user_skill	username
VAHA	beta_user_skill	weight
FK_BETADOVEDNOST	beta_user_skill	skill_id