

# Tvorba interní analytiky prostřednictvím SQL a BI softwaru GoodData

## Bakalářská práce

*Studijní program:*

B6209 Systémové inženýrství a informatika

*Studijní obor:*

Manažerská informatika

*Autor práce:*

**Petra Nedvědová**

*Vedoucí práce:*

Mgr. Tomáš Žížka, Ph.D.

Katedra informatiky







## Zadání bakalářské práce

# Tvorba interní analytiky prostřednictvím SQL a BI softwaru GoodData

*Jméno a příjmení:* **Petra Nedvědová**  
*Osobní číslo:* E19000227  
*Studijní program:* B6209 Systémové inženýrství a informatika  
*Studijní obor:* Manažerská informatika  
*Zadávací katedra:* Katedra informatiky  
*Akademický rok:* **2021/2022**

### Zásady pro vypracování:

1. Charakteristika pojmů v oblasti analytiky, BI a SQL
2. Analýza současného stavu interní analytiky
3. Proces zpracování dat pomocí SQL
4. Vizualizace dat platformou GoodData
5. Zhodnocení implementovaného řešení

Rozsah grafických prací:  
Rozsah pracovní zprávy:  
Forma zpracování práce:  
Jazyk práce:

30 normostran  
tištěná/elektronická  
Čeština



### Seznam odborné literatury:

- HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK, 2015. *Big Data a NoSQL databáze*. Praha: Grada Publishing. ISBN 978-80-247-5466-6.
- JACKSON, Peter a Caroline CARRUTHERS, 2019. *Data Driven Business Transformation: How to Disrupt, Innovate and Stay Ahead of the Competition*. Chichester: John Wiley & Sons. ISBN 978-1-119-54315-2.
- LAURENČÍK, Marek, 2018. *SQL: Podrobný průvodce uživatele*. Praha: Grada Publishing. ISBN 978-80-271-0774-2.
- POUR, Jan, Miloš MARYŠKA, Iva STANOVSKÁ a Zuzana ŠEDIVÁ, 2018. *Self Service Business Intelligence: Jak si vytvořit vlastní analytické, plánovací a reportingové aplikace*. Praha: Grada Publishing. ISBN 978-80-271-0616-5.
- WEXLER, Steve, Jeffrey SHAFFER a Andy COTGREAVE, 2017. *The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios*. New Jersey: John Wiley & Sons. ISBN 978-1-119-28271-6.
- PROQUEST, 2021. *Databáze článků ProQuest* [online]. Ann Arbor, MI, USA: ProQuest. [cit. 2021-09-26]. Dostupné z: <http://knihovna.tul.cz>

Konzultant: Bc. Jakub Šebek – Datový Analytik, BizzTreat s.r.o.

Vedoucí práce:

Mgr. Tomáš Žižka, Ph.D.  
Katedra informatiky

Datum zadání práce:

1. listopadu 2021

Předpokládaný termín odevzdání:

31. srpna 2023

doc. Ing. Aleš Kocourek, Ph.D.  
děkan

L.S.

Ing. Petr Weinlich, Ph.D.  
vedoucí katedry

V Liberci dne 1. listopadu 2021

## Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědoma toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

3. května 2022

Petra Nedvědová



## **Anotace**

Tato bakalářská práce se teoreticky zabývá problematikou z oblasti databází, SQL a datové vizualizace. Tyto teoretické znalosti následně využívá při praktickém zpracování dat v jazyce SQL a vizualizacích dat v platformě GoodData. Cílem je zanalyzovat data získaná z interních aplikací do prezentovatelné podoby. Výsledkem této práce je návrh a zpracování dashboardu pro zaměstnance firmy, ve kterém si mohou na denní bázi kontrolovat ukazatele své výkonnosti.

## **Klíčová slova**

Datová analytika, datová vizualizace, SQL, databáze, datové modelování, GoodData, Business intelligence

## **Annotation**

This Bachelor thesis focuses on the theory behind database systems, SQL and data visualization. The theory part of this thesis is then used in practice processing, analysing, and visualising of data on the GoodData platform. The aim of this thesis is to analyse data from internal applications and to visualise them on the GoodData platform. The result of this thesis is a mockup and a creation of a dashboard for employees. The employees can check on daily basis their performance indicators.

## **Keywords**

Data analytics, data visualization, SQL, database, data modeling, GoodData, Business intelligence



## **Poděkování**

Ráda bych poděkovala Mgr. Tomáši Žižkovi, Ph.D. za odborné vedení, cenné rady a vstřícnost při psaní závěrečné práce.

Dále mé poděkování patří kolegům z firmy Bizztreat za praktické rady, ochotu a další cenné zkušenosti.



# Obsah

Seznam ilustrací.....	13
Seznam použitých zkratk.....	14
Úvod.....	15
1 Charakteristika pojmů v oblasti databází, SQL, BI a analytiky.....	17
1.1 Databáze a databázové informační systémy.....	17
1.2 Databázové modely.....	17
1.2.1 NoSQL databáze.....	18
1.3 Struktura databáze.....	20
1.3.1 Datové modelování.....	20
1.3.2 Konceptuální model.....	20
1.3.3 Logický model.....	21
1.3.4 Fyzický model.....	21
1.3.5 Tabulky a datové typy.....	23
1.3.6 Primární klíč.....	25
1.3.7 Cizí klíč.....	25
1.3.8 Referenční integrita.....	26
1.4 SQL (Structured Query Language).....	27
1.4.1 Příkazy v SQL.....	27
1.5 Využití dat v ekonomické praxi.....	29
1.5.1 Business Intelligence (BI).....	29
1.5.2 Presentování a vizualizace dat.....	30
1.5.3 Typy dat a grafů.....	31
1.5.4 Report.....	32
1.5.5 Dashboard.....	33
1.6 Shrnutí teoretické části.....	34
2 Analýza současného stavu interní analytiky.....	35
2.1 Návrh řešení.....	35
2.2 Datové zdroje.....	38
3 Proces zpracování dat pomocí SQL.....	39
3.1 Očištění dat.....	39
3.2 Transformace dat.....	40
4 Vizualizace dat platformou GoodData.....	43
4.1 Datové modelování v platformě GoodData.....	43

4.2	Tvorba metrik.....	45
4.3	Insights.....	47
4.3.1	KPI.....	49
4.4	Dashboard.....	49
4.5	Zhodnocení implementovaného řešení.....	52
	Závěr.....	53
	Seznam použité literatury.....	55

## Seznam ilustrací

Obrázek 1: Nahrazení vazby M:N vazbami 1:M zpracované v aplikaci Lucidchart.....	23
Obrázek 2: Vizualní návrh pro řešení dashboardu zpracovaný v aplikaci Lucidchart.....	37
Obrázek 3: Ukázka zjednodušeného LDM vytvořeného v platformě GoodData.....	44
Obrázek 4: Rozložený dataset Team Members vytvořený v platformě GoodData.....	44
Obrázek 5: Záložka Load Configuration pro mapování z platformy GoodData.....	45
Obrázek 6: Složka metrik v záložce Manage z platformy GoodData .....	46
Obrázek 7: Prostředí pro tvorbu insightů z platformy GoodData .....	48
Obrázek 8: Výstupní dashboard vytvořený v platformě GoodData .....	51

## Seznam použitých zkratk

BI	Business intelligence
DDL	Data Definition Language
DML	Data Modification Language
FK	Foreign key (cizí klíč)
ID	Identity/Identification (jednoznačný identifikátor)
IT	Informační Technologie
KPI	Key performance indicators (klíčové ukazatele výkonnosti)
LDM	Logický Datový Model
MAQL	Multi-Dimension Analytical Query Language
MTD	Month-to-date (perioda od prvního dne měsíce po dnešní den)
NoSQL	not only SQL/non SQL
PK	Primary key (primární klíč)
RDM	Relational Data Model
SEQUEL	Structured English Query Language (původní název pro SQL)
SQL	Structured Query Language
SSBI	Self Service Business Intelligence (typ BI)

# Úvod

V dnešní době už není pro jednotlivce možné zpracovat všechna data, se kterými se každý den setkává. Může se proto stát, že člověk v tom velkém množství dat přehlédne důležitou informaci, která by mu mohla přinést užitek. Tohle neplatí pouze pro osoby, ale i pro každou firmu. Analýza dat je tedy v dnešní době pro podniky velmi důležitou součástí. Kdo data nedokáže zanalyzovat ztrácí přehled o interních i externích podnikových faktorech a stává se méně konkurenceschopným. Data a byznys k sobě neodmyslitelně patří, a proto se vyplatí na ně spoléhat při řízení firmy. Bez dat by byla firma slepá a svá strategická rozhodnutí by neměla o co opřít.

Kam data ukládat? Jak se dají využít? Jak je správně vizualizovat? Na tyto a další otázky má bakalářská práce postupně odpovídat. Jejím cílem je teoreticky vysvětlit pojmy z oblastí databází, SQL, analytiky, prezentování dat a použít je i prakticky. Mezi praktické cíle této práce patří zpracování dat pomocí jazyka SQL, prozkoumat platformu GoodData a s ní spojené datové modelování či tvorbu vizualizací. Tyto znalosti jsou následně aplikovány ve společnosti Bizztreat s.r.o. při návrhu řešení, zpracování interních dat a tvorbě dashboardu v platformě GoodData. Tato firma je jednou z předních firem v České republice, která se více než dvacet let věnuje datové analytice pro klienty, což bakalářské práci poskytuje skvělé výchozí prostředí pro zpracovávání právě tohoto tématu.

Bakalářská práce je rozdělena do dvou částí. První z nich je část teoretická. V této části je na základě literární rešerše rozebírána daná problematika. Čtenář se zde setká s tématy týkajícími se databází, databázových struktur, datového modelování, jazyka SQL, BI, prezentování a vizualizace dat. Teoretická část práce také přináší informace o využití dat v podniku. Jsou zde rozebírány i datové reporty, dashboardy a typy grafů, které se na nich mohou objevit.

Druhá část bakalářské práce přináší praktické poznatky získané při zkoumání BI platformy GoodData. Obsahuje analýzu současného stavu interní analytiky a návrh pro její rozšíření. Následně je tento návrh postupně zpracován pomocí očištění dat, datových transformací v jazyce SQL, nahrání konzistentních dat do platformy GoodData a práce s daty i zde. V této části jsou rozebrány postupy týkající se datové vizualizace a to od datového modelování v platformě, přes tvorbu metrik a grafů, až po výsledný dashboard.





# 1 Charakteristika pojmů v oblasti databází, SQL, BI a analytiky

Aby bylo možné správně porozumět tvorbě interní analytiky, je nutné si nejprve vysvětlit některé pojmy a principy, které s touto problematikou souvisejí. Rozebrání pojmů bude řazeno podle logické posloupnosti tak, jak data procházejí jednotlivými oblastmi, tedy od databází po analytiku.

## 1.1 Databáze a databázové informační systémy

Aby bylo možné data zpracovat, musí být uložena v databázi. Databáze je definovaná jako uspořádaná soustava dat. Je podstatné rozlišovat mezi databází a databázovým systémem. Databázový systém je software pro práci s daty, kde je možné data upravovat, přidávat data nová, stávající doplňovat o další informace, případně data mazat a také je snadno vyhledat (Laurenčík 2018).

## 1.2 Databázové modely

Podle Laurenčíka (2018) databázové modely, jsou modely, díky kterým získává databáze svoji hierarchii. Databázový model popisuje vztahy mezi uloženými daty. Existuje velké množství databázových modelů, které se historicky vyvíjely a často jeden model vychází z druhého. Rozdělení na modely je teoretická záležitost, která pomáhá rozlišit mezi modely podle způsobů, kterými se do databází ukládají data. V praxi je možné se setkat s následujícími typy modelů:

- hierarchický model,
- síťový model,
- relační model.

Historicky první databázový model byl hierarchický model, který používá uspořádání rodič – potomek, tento model ale nedokázal vykreslit všechny typy vztahů mezi daty. Později byl vytvořen model síťový, který navíc obsahoval typ vztahu více ku více, což v praxi znamenalo, že jeden potomek mohl mít více rodičů (Laurenčík 2018).

Dnes se za nejvíce používaný model považuje koncept relačního modelu. Tento model vytváří datovou tabulku se sloupci a řádky. Pokud jsou data ze sloupců společná pro více tabulek, dochází k používání tzv. relací. Relace jsou vztahy mezi tabulkami a více budou rozebírány později (Laurenčík 2018).

Výhodou databázových systémů na úrovni relačního konceptu je to, že umožňují přístup více uživatelům naráz a podporují použití dotazovacího jazyka, kterým může být například SQL (Structured Query Language). Data jsou typicky ukládána na server, kde mohou uživatelé za pomoci k tomu určených programů data prohlížet (Holubová et al. 2015).

### **1.2.1 NoSQL databáze**

V dnešní digitální době, kdy je možné sbírat ohromné množství dat každou vteřinu, se ukazuje, že standardní databázové systémy už nestačí. Za účelem zpracování takto velkého množství dat, označovaných jako Big Data, vznikají NoSQL databáze. Pod zkratkou NoSQL se skrývá not only SQL, tedy databáze, která využívá nejen SQL. NoSQL databáze nepotřebují datové schéma, toto schéma je často vytvářeno až v aplikaci, která data finálně zpracovává či vizualizuje. Díky tomu se při zpětných změnách schématu nevyvíjí zátěž na databázi a práce s daty se stává flexibilnější (Holubová et al. 2015).

NoSQL databáze mohou být vyvíjeny v rámci open source komunit, kde si mezi sebou sdílí svoji práci jedinci nebo menší firmy. Druhou možností vývoje je vývoj ve velkých společnostech, mezi které se řadí Google, Amazon či Facebook, tyto a jim podobné firmy si NoSQL databáze většinou vytváří interně. Výhodou interního vývoje je vyvíjení na míru společnosti (Holubová et al. 2015).

Existuje více než dvě stě prodejců nabízejících NoSQL databáze. Každý používá své vlastní technologické postupy tak, aby usnadnil práci s databázemi svým vlastním klientům. Přesto se dají NoSQL databáze rozdělit do několika základních skupin, podle vzájemných podobností. V odborné literatuře se uvádí rozdělení na struktury: klíč-hodnota, dokumentové, sloupcové a grafové (Abirami et al. 2021).

## **Struktura klíč-hodnota**

Nejčastější strukturou NoSQL databází je struktura klíč-hodnota. Data v této struktuře nejsou jako v tradičních relačních databázích v řádcích, sloupcích či tabulkách, ale jak už název napovídá, ukládají se po dvojích. Každá dvojice obsahuje klíč, který má většinou datový typ string a hodnotu, která obsahuje data. Jedná se o nejlehčí typ struktury databází, která umožňuje skladovat dvojice dat, ale neumožňuje mezi dvojicemi vytvářet vztahy pomocí primárních a cizích klíčů. Praktickým využitím této struktury může být košík při online nákupu, kde se skladují data způsobem: ID – 123456, jméno – Jan, cena nákupu – 1000 Kč, datum nákupu – 05-04-2022 (Abirami et al. 2021).

## **Dokumentová struktura**

Tato databázová struktura je tvořena dokumenty ve formátech JSON, XML, YAML či binární formě například ve formátu PDF nebo MS Word. Tato struktura je horizontálně škálovaná a struktura vychází podle názvu ze struktury dokumentů. Hodnoty v této struktuře mohou být dotazována pomocí primárních i cizích klíčů. Podobně jako v relační databázi, je zde možné data vytvářet, číst, mazat nebo upravovat (Abirami et al. 2021).

## **Sloupcová struktura**

NoSQL databáze využívající sloupcovou strukturu ukládají data ve sloupcích, n rozdíl od relačních databází, které ukládají data v řádcích. Díky tomu je možné dosáhnout rychlejší odezvy databáze i při velkém množství dotazovaných dat. Tento typ struktury je vhodný pro big data. Mimo jiné dostupné databáze nabízí uložení dat tímto způsobem i databáze BigQuery, která je hlavním zdrojem dat v teoretické části této práce (Abirami et al. 2021).

## **Grafová struktura**

Grafově orientovaná struktura ukládá entity, zde nazývané uzly a vztahy mezi nimi, které se nazývají hranami (Edges). Každý uzel má také své hrany, které jsou občas také označovány jako vlastnosti (Properties). Tyto vlastnosti reprezentují vzdálenost a směr mezi uzly. Z každého uzlu vycházejí přímé spojnice do uzlů sousedních. Tento typ struktury je vhodný, pro datové zpracování, jehož výstupem by mělo být nalezení co nejkratších cest z bodu A do bodu B (Abirami et al. 2021).

## **1.3 Struktura databáze**

Strukturu relační databáze tvoří několik typů objektů. Mezi hlavní z nich řadíme Tabulky, Queries a Diagramy. Tabulky jsou tvořeny zpracovanými daty. Jednotlivé tabulky jsou objekty s vlastní předem definovanou strukturou. Queries (dotazy), v odborné literatuře také uváděny pod pojmem Pohledy (Views), jsou příkazy jazyka SQL. Tyto příkazy se jednou vytvoří a poté se dají opakovaně spouštět a používat. Tabulky v databázi mohou mít vzájemné vztahy nebo na sobě mohou být závislé. Tyto vztahy se určují pomocí společných údajů, které se objevují ve dvou či více tabulkách. Objekt, který vztahy zaznamenává a ukládá, se nazývá diagram nebo také často používaným výrazem relace. Detailněji bude tato problematika rozebrána v kapitole o datovém modelování (Laurenčík 2018).

### **1.3.1 Datové modelování**

Tabulky se dají považovat za kostru databází, ale to zdaleka nestačí. Datové modelování je nedílnou součástí všech informačních systémů, databází nevyjímaje. Datový model slouží jako formální struktura, která zobrazuje vztahy v databázi. Dobře navržený datový model by měl odpovídat reálným vlastnostem skutečností. Modely tedy tvoří abstraktní obraz skutečnosti více či méně zjednodušený. Existuje více typů datových modelů (Drobíková et al. 2018).

### **1.3.2 Konceptuální model**

Konceptuální model je nejvíce obecným modelem, jde o prvotní návrh databáze, který řeší věcnou podstatu dat. Nezohledňuje se u něj závislost dat na konkrétním softwaru a tedy ani technologické charakteristiky databáze. Účelem tohoto typu modelu je specifikovat entity, jejich atributy, přirozenou identifikaci a vazby. V tomto typu modelu se nevytváří cizí klíče, vazby jsou pouze naznačeny graficky (Pour et al. 2018).

Základními prvky konceptuálního modelu jsou entity. Entity jsou chápány jako množiny věcí podobného druhu. Entity mají také vlastnosti, kterým se říká atributy entit. Každý atribut nabývá určité hodnoty. Například v entitě Student nabývá atribut Jméno hodnoty Petra a atribut Příjmení hodnoty Nedvědová. Prvky entit často figurují i ve vztazích k prvkům jiných entity. Tyto vztahy se odborně nazývají relace a dělí se na specifické druhy (Drobíková et al. 2018).

### **1.3.3 Logický model**

Logický model vychází z modelu konceptuálního a řeší strukturu databáze detailněji a na již předem vymezeném logickém základě, tím může být například relační schéma dat. Jde zde tedy hlavně o definování struktury tabulek, atributy, primární klíče a také o tvorbu relací prostřednictvím cizích klíčů (Pour et al. 2018).

### **1.3.4 Fyzický model**

Fyzický model se vygeneruje na základě modelu logického. Tento typ návrhu specifikuje implementační charakteristiky ve vztahu k příslušnému databázovému prostředí. Tím se v tomto kroku specifikují nejdetailejší charakteristiky, kterými jsou formáty datových položek, délka položek, definice cizích klíčů nebo vztahy mezi tabulkami (Pour et al. 2018).

#### **Relace**

Pokud jsou jasně specifikované entity je potřeba definovat vztahy mezi nimi. Těmto vztahům se říká relace (odvozeno od anglického slova relationships – vztahy). Existují tři základní typy relací: jedna ku jedné (1:1), jedna ku více (1:M) a více ku více (M:N nebo také M:M) (Harrington 2016).

#### **Relace 1:1**

Relace 1:1 se používá v případě, kdy k jedné entitě přísluší jedna a právě jedna entita a opačně. Lépe se tyto vztahy chápou na příkladech, proto je zde uveden hypotetický příklad malého města s letištěm. Ve městě je právě jedno letiště a letiště leží na území právě jednoho města (Harrington 2016).

V byznysové praxi je ale tento typ vztahu velmi vzácný, protože se málokdy stane, že by existovaly entity, které mají relaci právě k jedné entitě a opačně. I kdyby tato situace nastala, např. by každý dodavatel měl právě jednu e-mailovou adresu, je stejně vhodnější počítat s možností, že v budoucnosti existuje situace, ve které se e-mailových adres jednoho dodavatele objeví více. Dodavatel by mohl změnit název společnosti, nebo začít využívat jinou doménu. Proto je dobré používat relaci 1:1 velmi obezřetně. (Harrington 2016).

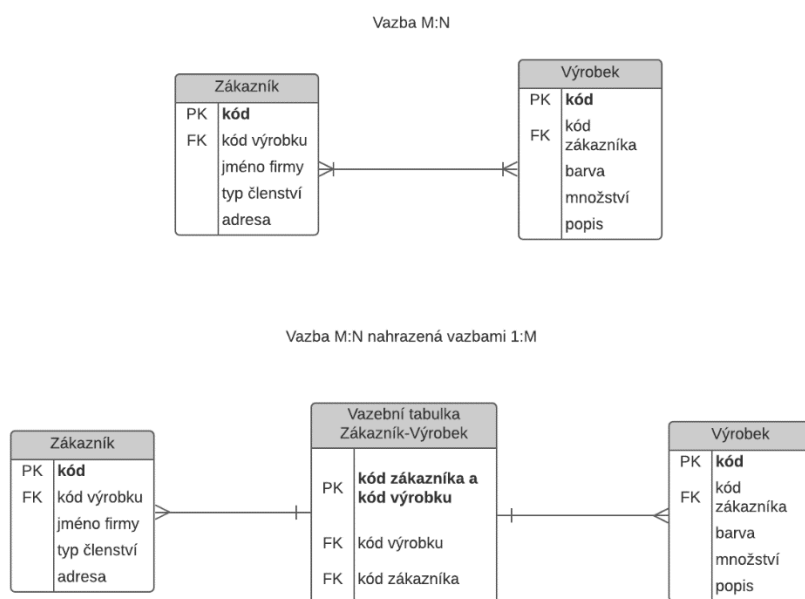
### **Relace 1:M**

Relace 1:M je nejčastěji používanou relací. Je to hlavně z důvodu, že často nejlépe popisují byznysovou realitu. Příkladem 1:M relace může být jeden dodavatel, který má více poboček a více poboček, které spadají pod jednoho dodavatele. Je nutné podotknout i to, že entity mohou mezi sebou mít relace i přestože jejich hodnoty atributů jsou nulové. Matka může mít více dcer, jednu i žádnou dceru a dcery mají právě jednu biologickou matku. Na tomto příkladu je hezky vidět, proč je vhodnější používat relaci typu 1:M před relacemi typu 1:1. Přestože je jedna matka a jedna dcera, zdánlivě relace typu 1:1, je na ně vhodně použitá relace typu 1:M (Harrington 2016).

### **Relace M:N**

Relace M:N jsou také běžné, ale nejsou nejčastější. Příkladem pro tuto relaci je vztah zákazník:výrobek. Více zákazníků kupuje několik produktů a každý produkt se může prodat více zákazníkům. I zde je možné, aby se relace týkala žádného, jednoho či více atributů entit (Harrington 2016).

U relací M:N může docházet k problému při návrhu databáze. Problém spočívá v tom, že RDM (relational data model) není schopný vytvořit přímou vazbu M:N. Je limitován pouze na dva předchozí typy relací, proto se v modelech používá jako nahrazení relace 1:M, která se vytvoří dvakrát, tím pádem se z 1:M a 1:M stane M:N. Mezi těmito relacemi vzniká tzv. vazební tabulka. Tím se celý problém vyřeší (Harrington 2016).



Obrázek 1: Nahrazení vazby M:N vazbami 1:M zpracované v aplikaci Lucidchart

Zdroj: vlastní tvorba

### 1.3.5 Tabulky a datové typy

Základem databáze jsou již zmíněné datové tabulky, ty obsahují data, se kterými se dá dále pracovat. Každá tabulka se dělí do řádků (rows) a sloupců (columns). V řádcích se objevují údaje a sloupce tyto údaje rozdělují podle obsahu a typu dat (Laurenčík 2018).

Rozlišuje se několik typů sloupců a to podle údajů ukládaných v databázi. Jsou to sloupce textové, číselné, „datumové“ a sloupce pro logické hodnoty. Toto je základní rozdělení, pokud je žádoucí uložit neobvyklejší údaje např. o geografické poloze nebo obrázky, je to v relační databázi také možné. Pokud už je určen typ sloupce, je možné ukládat pouze hodnoty stejného typu. Což znamená, že do sloupce s číselným typem se nedají uložit data v textové podobě a naopak. Toto pravidlo mnohonásobně zvyšuje výkonnost při práci s daty, protože není nutné, aby databázový systém zjišťoval datové typy každého záznamu (Laurenčík 2018).

Protože však dělení pouze na textové, číselné, „datumové“ a logické typy není dostačující, používá se v relačních databázích detailnější dělení. Je vhodné zmínit, že datové typy se v různých databázových systémech mohou lišit. Zde jsou uvedeny pro zjednodušení pouze vybrané datové typy z databázového systému MySQL (Laurenčík 2018).

Číselné typy se dělí na celočíselné typy a typy s desetinnými čísly. V MySQL se mimo jiné používá číselný typ INT – celé číslo v intervalu od -2 147 483 647 do +2 147 483 647. Pro čísla, která obsahují desetinná místa, se nejčastěji používá datový typ FLOAT, který má také větší rozsah než INT (Laurenčík 2018).

V textových datových typech se rozlišují datové typy TEXT, VARCHAR a CHAR. CHAR je typ s pevně danou délkou. VARCHAR si zaznamenává zadané znaky, délka tedy není předem pevně daná. Typ TEXT slouží pro rozsáhlá textová data (Laurenčík 2018).

„Datumové“ typy MySQL jsou čtyři: DATETIME datový typ pro uložení „datumu“ a času ve formátu YYYY-MM-DD HH:MI:SS, DATE pro uložení pouze „datumu“ ve formátu YYYY-MM-DD, TIME pro uložení pouze času ve formátu HH:MI:SS a TIMESTAMP neboli časové razítko. TIMESTAMP při novém záznamu vloží do pole aktuální „datum“ a čas (Laurenčík 2018).

Poslední z často využívaných datových typů jsou logické hodnoty. Jejich hodnoty jsou v MySQL zapisovány jako 1 (True) nebo 0 (False). Pro jejich zápis je určen datový typ BIT (Laurenčík 2018).



### 1.3.6 Primární klíč

Aby bylo možné identifikovat každý řádek v tabulce jednotlivě, je zapotřebí mít unikátní identifikátor tohoto řádku. Takovému identifikátoru se říká primární klíč (primary key). Pokud je dosaženo unikátnosti primárních klíčů na každém řádku v tabulce, znamená to, že je možné přesně definovat, že je zacházeno s hledaným řádkem. Pro konkrétní hodnotu, tedy stačí znát tři věci: název tabulky, název sloupce a primární klíč řádku. Název primárního klíče musí být unikátní a nenulová hodnota. Nulová hodnota NULL je v databázích hodnotou, která značí neznámou hodnotu, ale není to přesně to samé jako hodnota 0 nebo prázdná (empty)/ nevyplněná hodnota. Primární klíč se také dá vytvořit spojením několika hodnot, například spojením rodného čísla zaměstnance a aktuálního data a času. Tímto způsobem můžeme teoreticky sledovat unikátní záznamy v čase u každého zaměstnance (Harrington 2016).

Vytvoření správného primárního klíče není jednoduché. Některé entity mají svůj primární klíč přirozeně. Příkladem přirozeného primárního klíče u osob je rodné číslo, u produktů to může být unikátní kód ve vzorkovníku nebo číslo produktu. Pokud je to možné doporučuje teoreticky zaměřená odborná literatura využívat přirozených primárních klíčů. Při praktické tvorbě databází jsou ale často i u těchto případů využívány umělé primární klíče tzv. ID. ID by se mělo využívat v případech, kdy přirozený primární klíč neexistuje. ID se většinou tvoří z číselné řady – 0001,0002,0003... (Harrington 2016).

### 1.3.7 Cizí klíč

Cizí klíč je možností, jak mezi sebou dvě tabulky se společnými prvky propojit. Vychází tedy z relací mezi tabulkami. Cizí klíč je sloupec v tabulce B, který odkazuje na primární klíče v tabulce A. Vychází-li se ze vztahu majitel – obchod a majitelé mají jako id své rodné číslo a každý majitel má k sobě přiřazené své obchody, je cizím klíčem v tabulce obchod rodné číslo majitele. Takto se snadno dá zjistit, které obchody jsou majetkem, kterého majitele. Cizí klíč je v databázích značen zkratkou FK z anglického foreign key (Drobíková et al. 2018).

Zajímavým úkazem je možnost využití cizího klíče, který odkazuje na primární klíč ve stejné tabulce. Cizí klíč nemusí odkazovat vždy na primární klíč tabulky jiné, ale může odkazovat i na tabulku vlastní. Příkladem toho může být situace, kdy zaměstnanec v tabulce zaměstnanci, je zároveň manažerem i zaměstnancem, cizí klíč `manager_id` tedy odkazuje na primární klíč `employee_id` ve stejné tabulce (Harrington 2016).

### **1.3.8 Referenční integrita**

Součástí mezitabulkových relací v databázi jsou většinou i omezovací podmínky pro práci se záznamy v těchto tabulkách. Tyto podmínky slouží k zamezení výskytu hodnot cizích klíčů, které neodpovídají hodnotě žádného primárního klíče v tabulce hlavní. Je možné, aby v tabulce firem, určité firmě neodpovídala žádná faktura, ale není možné, aby faktura neodpovídala žádné firmě. Plyne to z logického předpokladu, že firma mohla dosud nevystavit žádnou fakturu, ale pokud už faktura existuje, musela být vystavená konkrétní firmou. Tyto omezovací podmínky se nazývají referenční integrita (Laurenčík 2018).

Rozlišují se dva typy referenční integrity a to kaskádové a restriktivní. Kaskádová pravidla umožňují změny kaskádově, což znamená, že pokud vznikne úprava primárního klíče v hlavní tabulce, upraví se stejně hodnoty cizích klíčů v tabulce vedlejší. Stejně jako úprava funguje v tomto případě i mazání záznamů. Oproti tomu restriktivní pravidla zamezují úpravám či mazání záznamů v hlavní tabulce, pokud se hodnota primárního klíče vyskytuje v propojené tabulce. Tyto typy omezovacích podmínek se nastavují podle logiky ukládaných dat (Laurenčík 2018).

## 1.4 SQL (Structured Query Language)

Tento dotazovací jazyk byl vytvořen pro práci s relačními databázemi. Začátek jeho vývoje se datuje do druhé poloviny sedmdesátých let 20. století. Nejprve byl pojmenován zkratkou SEQUEL, ale později byl přejmenován na dnes již známější SQL. Protože se začalo SQL používat při dotazování do několika databázových systémů, například při uvedení databázového systému Oracle v roce 1977, bylo zapotřebí SQL standardizovat. V roce 1982 byla vytvořena první standardní verze jazyka SQL institucí American National Standards Institute, tato verze byla od té doby ještě několikrát přepracovaná do podoby, jaká je dnes známá. Dnes je jazyk SQL implementován v mnoha databázových programech a méně či více se od standardní verze liší (Laurenčík 2018).

### 1.4.1 Příkazy v SQL

SQL je dotazovací jazyk, který není tzv. case sensitive, což znamená, že nerozeznává rozdíl mezi velkými a malými písmeny. Toho se v praxi využívá a je dobrým zvykem psát klíčová slova v SQL velkými písmeny a názvy sloupců a tabulek písmeny malými. Toto se využívá pro dosažení přehlednosti kódu, není to však nutností pro jeho funkčnost. Stejně tak příkazy v SQL ignorují přechody na novou řádku. Pro lepší přehlednost je opět vhodné zvolit oddělení logických částí a novou část začít psát vždy na nový řádek. Příkazy se ukončují středníkem, který většina databázových systémů vyžaduje (Laurenčík 2018).

#### SELECT

Nejpoužívanějším příkazem je příkaz SELECT, ten se používá k zobrazení dat. Jeho povinnou součástí je klauzule FROM. Při psaní dotazu (query) je nejprve nutné znát tabulku nebo tabulky, ze kterých se data berou. Poté se klauzulí WHERE vyfiltrují specifická data, pokud je to třeba a nakonec se vše zabalí do klauzule SELECT, ve které se definuje, které sloupce jsou k zobrazení. Při použití znaku \* se vyberou všechny definované sloupce. Jednoduchým příkladem je následující příkaz, který vybere všechna data o zvířatech typu kočka z databáze (Beaulieu 2020):

```
SELECT * FROM animals
WHERE animal_type = 'cat'
;
```

Mimo povinné klauzule SELECT a FROM, má příkaz SELECT i spoustu nepovinných klauzulí. Nejčastěji používanou je právě filtrovací klauzule WHERE uvedená výše, která obsahuje logický výraz, jehož výsledkem je buď hodnota True, pro kterou se řádky zobrazí, nebo hodnota False, pro kterou se řádky nezobrazí. Klauzulí pro počítání částečného součtu je klauzule GROUP BY. Touto klauzulí je možné počítat například zisky po jednotlivých odděleních. HAVING je klauzule používaná dohromady s klauzulí GROUP BY a využívá se k omezení souhrnů či součtů podle definované podmínky, dalo by se tedy říci, že plní podobnou funkci pro GROUP BY jako klauzule WHERE pro SELECT. Poslední nepovinnou klauzulí je ORDER BY, která seřadí výsledek podle jednoho či několika kritérií (Laurenčík 2018).

### **Příkazy DML**

Tyto příkazy slouží pro práci s daty, pod zkratkou DML se skrývá Data Modification Language. Přestože nejsou tyto příkazy, tak používané jako příkaz SELECT, mohou být užitečné. Pomocí těchto příkazů se dají do tabulek zapisovat nová data do již existujících řádků, upravovat původní data, přidávat řádky nebo řádky odstraňovat. Mezi tyto příkazy se řadí příkaz INSERT – příkaz pro vložení, DELETE – příkaz pro smazání či UPDATE – příkaz pro úpravy. Tyto příkazy se také dají kombinovat s příkazem SELECT (Laurenčík 2018).

### **Příkazy DDL**

Jazyk Data Definition Language slouží pro definici dat. Pomocí DDL se dají vytvářet nebo upravovat některé specifické databázové objekty. Pomocí DDL se dají vytvořit tabulky, měnit struktury již existujících tabulek nebo měnit relace a s nimi související klíče. Opět je možné nakombinovat tento typ příkazů s příkazem SELECT. Příkladem příkazů DDL může být tento jednoduchý skript (Burian 2014):

```
Create Table Student
(Id Char(15) Not Null,
Kresni_jmeno Char(30),
Prijmeni Char (50))
;
```

## 1.5 Využití dat v ekonomické praxi

Okolo roku 1970 byl svět svědkem třetí industriální revoluce, která přinesla nové technologie, internet a velké množství dat. S příchodem moderních technologií se naskytla příležitost zapojit informační technologie i do řízení podniků. V konkurenčním prostředí trhu se časem stalo začlenění informačních technologií do běžného chodu firmy nutností. Přeorientování firem na „IT-driven“ organizaci, neboli firmu řízenou pomocí IT, s sebou často neslo jistá úskalí, ale v dlouhodobém časovém horizontu se vyplatilo (Jackson a Carruthers 2019).

Ruku v ruce s informačními technologiemi šla data. Umět správně firemní i konkurenční data zpracovat je chápáno jako velká konkurenční výhoda. Hodně firem rozumí, že jim tato schopnost poskytne kvalitnější a efektivnější klientský servis. Rychlejší získávání zákazníků, větší šanci na jejich udržení i více individuální přístup k odlišným skupinám zákazníků. Rozhodnutí zaměřit se na zpracování dat často probíhalo za účelem růstu, ale v některých případech, třeba když začal Netflix konkurovat klasickým televizním kanálům v Americe, to mohl být jediný způsob jak firmu zachránit před krachem (Jackson a Carruthers 2019).

Datově orientovaná firma se také dá lépe organizovat. Velké množství společností má šanci zlepšit svou organizaci, pokud se bude věnovat zkoumání dat. Datoví analytici mohou poskytnout informace jako podklady pro velká rozhodnutí. Pohledy na náklady nebo výnosy v různém časovém období za jednotlivé pobočky a oddělení, jejich výdělečnost, počet nově získaných klientů a mnoho dalších faktorů se může stát dobrým základem pro řízení úspěšné firmy (Jackson a Carruthers 2019).

### 1.5.1 Business Intelligence (BI)

Přesná definice pro BI se hledá těžko hlavně kvůli komplexnosti pojmu. Obecně se dá říci, že BI jsou technologie a metody, jimiž se získávají nové informace využívané pro chod byznysu (Holubová et al. 2015).

Business Intelligence je považována za jednu z nejdůležitějších aplikací IT v rámci podniku. Její implementace může totiž ve firmách ovlivnit mnohé ekonomické aktivity. Mezi ně se dá zařadit řízení podniku, obchodní aktivity, plánování a rozhodování. Zároveň se zavedení BI může promítnout i do celkové prosperity firmy a konkurenceschopnosti na trhu. Proto představují investice do BI v dnešní době pro firmy jednu v nejčastějších investic v IT sféře (Pour et al. 2018).

Jedním z trendů v rámci BI je Self Service Business Intelligence (SSBI), což se dá přeložit jako samoobslužná Business Intelligence. Oproti klasickému BI, které se zaměřuje na celopodnikové systémy, velké množství dat a složité principy, SSBI se zaměřuje na tvorbou řešení pro individuální uživatele a menší celky napříč firmou. Zároveň SSBI řešení používají jiné produkty např. Power BI, Tableau nebo GoodData (Pour et al. 2018).

## **1.5.2 Prezentování a vizualizace dat**

Data jsou chápána jako jevy, které se dají nějakým způsobem zachytit, například zapsat. Data můžeme analyzovat a výsledkem takovéto analýzy by měl být datový výstup, tedy zpracovaná data, kterým se dá snadno porozumět. Výstupem sice může být i nezpracovaný CSV soubor obsahující spoustu čísel, ale pro lidské chápání je mnohem vhodnější vidět data zpracovaná nejlépe ve formě interaktivních vizualizací. K finálním reportům se dá využít například program Microsoft Excel nebo další pokročilejší tooly jako PowerBI nebo GoodData (Gemignani 2015).

Při tvorbě takovýchto datových výstupů je nutné myslet v první řadě na datového konzumenta, tedy člověka, který data bude přijímat, a přizpůsobit tomu výstup. Než se začne s datovou vizualizací je dobré si zodpovědět několik otázek. Pro koho se tato data zpracovávají? Jak podrobná data jsou potřeba? Která data jsou důležitá a která nikoli? Bude toto řešení srozumitelné pro datové konzumenty? Je tedy vhodné se nejprve zaměřit na vytyčený cíl a až poté se pustit do práce. Zkušenější datový analytik zvolí správný typ grafu, vhodné popisky i barvy. Neméně důležité je i správné rozložení jednotlivých grafů v rámci reportu nebo dashboardu (Gemignani 2015).

### 1.5.3 Typy dat a grafů

Při tvorbě jednotlivých vizualizací je nutné znát typ dat, který je zpracováván. Existují tři základní typy dat a to nominální, ordinální a kvantitativní. Nominální nebo také kategoriální data reprezentují věci nebo vlastnosti, které se nedají vzájemně porovnávat, ani průměrovat. Jsou to například data typu křestní jméno a příjmení, povolání nebo barva očí. Ordinální data jsou podobná datům nominálním, ale umožňují porovnávání. Je to například platová třída, dosažené vzdělání nebo míra spokojenosti zákazníka. K ordinálním datům je tedy možné přiřadit jistou škálu. Kvantitativní data jsou data numerická, dají se s nimi provádět matematické operace jako je sčítání, odčítání, aritmetický průměr a další. Příkladem kvantitativních dat jsou zisk, počet zákazníků, bodový výsledek testu nebo počet stránek knihy (Wexler et al. 2017).

#### **Tabulka**

Základní forma prezentace dat je tabulka. Ta se volí v případech, kdy je vyžadováno hodně detailních informací, které by se ve vizualizacích ztratily. Předpoklad pro dobrou tabulku je její jednoduché uspořádání, usnadňuje se tím orientace v datech. Tabulka pro operativní a taktické účely by měla zdůrazňovat odchylky, limity nebo rozdíly mezi sledovanými veličinami (Pour et al. 2018).

#### **Sloupcový graf**

Sloupcový graf je oblíbeným typem grafu. Sloučí pro porovnání kvantitativních ale i ordinálních nebo nominálních hodnot. Tento graf existuje ve dvou základních podobách a to jak vertikálně, tak horizontálně. Pro horizontální sloupcový graf se také používá termín pruhový graf. Tyto grafy jsou vhodné, protože poskytují snadné a přehledné porovnání hodnot. Hodnoty ve sloupcovém grafu je nutné seřadit sestupně nebo vzestupně podle logického významu vizualizované proměnné. Přestože je sloupcový graf velmi srozumitelný, existují nevhodná použití tohoto grafu, a to zejména v případech, kdy dojde k zobrazení velkého objemu dat a jednotlivé sloupce jsou od sebe pro čtenáře nerozeznatelné (Pour et al. 2018).

### **Výsečový graf**

Výsečový neboli koláčový graf slouží k zobrazení vztahu části pozorovaných dat vůči celku. Celý kruh tvoří 100 %, šířka výseče určuje podíl z celku. Přestože je výsečový graf často používaný, není nejvhodnějším grafem. Je často chybně interpretován. Přestože je možná vizuálně hezký, podle odborníků je velmi nepřesný, protože lidské oko nemá schopnost rozpoznat a porovnat všechny úhly výseče a udělat si jasnou představu o prezentovaných datech (Pour et al. 2018).

### **Bullet graf**

Bullet graf je jedna z nejlepších možností, pokud je vhodné porovnávat skutečnou hodnotu oproti té cílové. Skládá se ze tří hlavních částí. První částí je barevný sloupec naležato uprostřed, který značí skutečnou hodnotu. Druhou částí bullet chartu je čára, která značí, cílovou hodnotu. Poslední část je obdélníkový podklad, často světle šedé barvy, ten slouží pro měření výkonu a naznačuje, kdy je plnění plánu nedostatečné a kdy uspokojivé (Wexler et al. 2017).

### **Spojnicový graf**

Spojnicový graf slouží pro vizualizaci kvantitativních hodnot v čase. Má tvar křivky, která spojuje jednotlivé body. Tento graf se hodí pro znázornění časové řady nebo korelace hodnot. Dá se na něm znázornit více proměnných a jejich porovnání. Tento typ grafu se dá kombinovat se sloupcovým grafem (Pour et al. 2018).

## **1.5.4 Report**

Při tvorbě výstupů pro byznys a komunikaci BI k cílovým uživatelům je nejběžnější cestou report. Obecně by se dalo říci, že jsou reporty v praxi využívány skoro všude, aniž by si to jejich uživatelé uvědomovali. Report je totiž dokument, který obsahuje data. Report může být jednoduchá tabulka nebo komplexní soubor tabulek a grafů, které jsou interaktivní a poskytují uživateli i takzvané drillly, což jsou prokliky k detailnějším nebo souvisejícím částem reportu, které umožňují zaměřit se více na případný problém. Report nevede čtenáře k předem definovaným závěrům, ale spíše poskytuje data, ze kterých musí čtenář vyvodit závěry sám (Alexander et al. 2013).



### 1.5.5 Dashboard

Definice dashboardu by mohla znít nějak takto: dashboard je typ vizuálního zobrazení dat a používá se ke sledování důležitých informací, které jsou prezentovány co nejsrozumitelnější formou. V překladu do češtiny znamená slovo dashboard „přístrojová deska“. To právě z důvodu, že přináší klíčové ukazatele v podobě metrik a základní informace v reálném čase, stejně tak jako palubní deska například v letadle. Zároveň by měl být dashboard lehce aktualizovatelný a filtrovatelný. Protože pro každého člověka je nejsrozumitelnější lehce odlišné zobrazení dat, něco jako ideální dashboard neexistuje. Existují však jisté postupy, kterými se dá k ideálu dashboardu přiblížit (Wexler et al. 2017).

Na Dashboardu by nikdy neměly chybět klíčové hodnoty relevantní pro chod podniku nebo obecně pro cíl se kterým je dashboard tvořen. Dashboard zobrazuje data graficky, pomocí výše zmíněných grafů. Tyto grafy jsou často používány v kombinaci s jednotlivými klíčovými hodnotami nebo detailnějšími tabulkami. Vizualizace by měly vést čtenáře k důležitým závěrům a pomáhat mu soustředit se na klíčové trendy nebo znázornit výkyvy od normálu. Každý dashboard by měl zobrazovat pouze ta data, která jsou relevantní pro konkrétní dashboard. Je tedy vhodné vytvořit více dashboardů pro více problematik. Dashboard pro prodeje se bude lišit od dashboardu pro řízení firmy, a to i přestože mohou mít jisté klíčové hodnoty stejné. Dashboady obsahují předem definované závěry a nevyžadují tedy od čtenáře jeho vlastní analýzu dat, která může být časově velmi náročná (Alexander et al. 2013).

## 1.6 Shrnutí teoretické části

V oblasti datové analytiky je pracováno s nemalým množstvím odborných pojmů a principů. Ty nejrelevantnější v nich jsou zde nastíněny a rozebrány. Je tedy jasné, že data se ukládají v databázích, které jsou budovány podle stanovených pravidel a postupů. Data se mohou díky databázovým systémům dotazovat pomocí jazyka SQL a upravovat podle požadavků. Upravená, tedy očištěná a transformovaná data, se dále využívají k praktickému fungování podniku. Data mohou být prezentována pomocí grafů, tak aby předávala jejich konzumentům informace. Tento teoretický úvod byl nezbytný pro druhou část této práce, kterou je tvorba interní analytiky ve vybrané firmě. Vytváření následujícího řešení bude souviset s postupy probranými v první části této práce teoreticky.

## **2 Analýza současného stavu interní analytiky**

V současné době má firma Bizztreat, pro kterou je tento projekt zpracováván, v interní analytice zpracované již velké množství dat. Firma tato data využívá pro strategické rozhodování a pro přehled nad firemní situací. Data, která však zatím zpracovaná nejsou, jsou data týkající se jednotlivých členů firmy. Projekt, kterým se tato práce zabývá, spočívá ve zpracování těchto dat. Jedná se konkrétně o data o plánech odpracovaných hodin alokovaných na měsíc, a jejich plnění. Zadáním je tedy zpracování dat do podoby, ve které bude možné je nahrát do BI platformy GoodData a vytvořit z nich dashboard.

Tento dashboard bude využíván dvěma hlavními způsoby. Primárně bude fungovat jako osobní přehled pro každého člena firmy. Díky interně využívané platformě Bizzflow se budou data na dashboardu automaticky aktualizovat na denní bázi, což umožňuje jednotlivým uživatelům mít přístup k aktuálním datům. Uživatelé tak mohou mít přehled o svém pracovním výkonu a efektivněji plánovat svoji práci v průběhu každého měsíce. Sekundární využití je pro vedoucí jednotlivých týmů. Ti získají detailnější přehled o plnění plánů jednotlivými členy firmy a mohou agilněji reagovat při problémech s tím spojených.

### **2.1 Návrh řešení**

Aby se docílilo jasnější představě o výsledku již před začátkem práce na projektu a zamezilo se pozdějšímu nedorozumění, je vhodné při tvorbě jakéhokoliv řešení nejprve vytvořit mockup. Mockup je co nejdětalněji provedený návrh, který popisuje, jak by finální řešení mělo vypadat. Je vytvořen na základě požadavků zadavatele a může být několikrát pozměněn tak, aby byl zadavatel nakonec spokojený a zároveň tak, aby bylo navrhované řešení proveditelné.

Níže uvedený návrh byl vytvořený na webové platformě Lucidchart. Vykresluje cílový dashboard, ke kterému bude tento projekt směřovat. Mockup obsahuje tři sekce. První sekce zahrnuje klíčové ukazatele a bullet chart. Mezi klíčovými ukazateli se objevuje celkový počet plánovaných hodin, celkový počet odpracovaných hodin, utilizace a počet hodin, které je nutné každý den odpracovat, aby byl plán naplněný. Utilizace je poměr mezi hodinami odpracovanými pro zákazníka a hodinami, které jsou strávené na interních schůzkách a nefakturují se tudíž konkrétnímu zákazníkovi. Bullet chart bude obsahovat celkem tři parametry: odpracované hodiny, plánované hodiny a parametr pojmenovaný MTD planned hours, který prezentuje informaci, kolik hodin má být odpracováno právě k dnešnímu dni – tedy jestli je plněný měsíční plán k dnešnímu dni. Čísla volená v tomto návrhu jsou čistě náhodná a slouží pouze pro představu finálního celku.

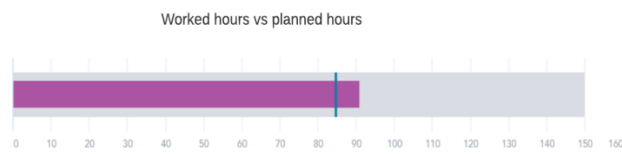
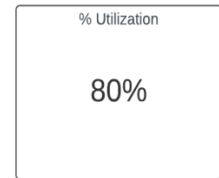
Druhá sekce s názvem Projects & Tasks se zaměřuje na projekty, na kterých byla v aktuálním měsíci odvedena práce, zároveň hodiny, které jsou zobrazovány, se budou dělit na non-billable a billable. Non-billable hodiny jsou ty, které jsou vedeny pro interní potřeby a záměry. Zatímco Billable hodiny jsou ty, které byly odpracovány v rámci projektů pro zákazníky. Definice a pochopení veškerých interních pojmů, které budou na dashboardu zobrazeny, je klíčové pro funkční dashboard. V druhé části této sekce pak bude vytvořený graf s daty týkajícími se odpracovaných hodin na jednotlivých táscích – úkolech. Vedle tohoto grafu bude tabulka s detailními informacemi a čísly tak, aby byla tyto čísla na první pohled viditelná.

Třetí a poslední sekce bude obsahovat pouze jeden sloupcový graf, který bude sloužit jako kalendář. Uživatel na něm uvidí jednotlivé dny v měsíci a sumu hodin, které v uplynulý den odpracoval. Hodiny budou rozděleny opět na billable a non-billable. Na ose x bude datum v číselném formátu a zároveň textem uvedený den v týdnu.

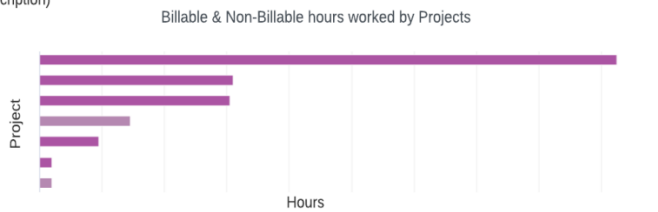
Filtr pro tento měsíc

Filtr pro konkrétního člověka

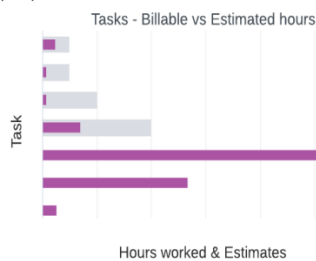
KPI's  
(Description)



Projects & Tasks  
(Description)



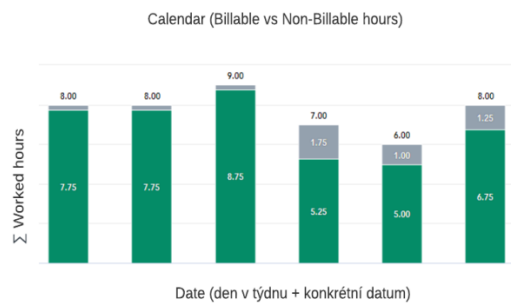
Tasks - Estimates  
(Description)



Tabulka obsahující informace z grafu "Tasks - Billable vs Estimated hours"

Project	Task	Worked Hours	Estimated hours
---------	------	--------------	-----------------

Calendar  
(Description)



Obrázek 2: Vizuální návrh pro řešení dashboardu zpracovaný v aplikaci Lucidchart

Zdroj: vlastní tvorba

## 2.2 Datové zdroje

Data, která budou zpracovávána pochází ve valné většině ze dvou zdrojů. Prvním z nich jsou Google Spreadsheets neboli česky z Google tabulky. Tento tabulkový program, podobný Microsoft Excelu, je dostupný online a umožňuje vytvářet přehledné tabulky, které mohou uživatelé mezi sebou sdílet a vzájemně doplňovat nebo upravovat. Každý tým spravuje svůj vlastní Google Spreadsheets, ve kterém mimo jiné eviduje jména členů týmu, jejich emailové adresy, měsíc, ke kterému se zápis vztahuje, a pro tento projekt stěžejní data o plánech každého zaměstnance. Aby ale bylo možné data z těchto zdrojů efektivně extrahovat, musí všechny týmy udržovat stejnou strukturu sloupců a data vpisovat pouze do kolonek, které jsou pro to určené. Mimo tyto plánovací tabulky, je v interní analytice používán také Google Spreadsheets, který funguje jako kalendář. Obsahuje na 30 let dopředu záznamy o všech datech a rozlišuje mezi všedními dny a víkendem. Také jsou v tomto dokumentu evidované státní svátky a názvy jednotlivých dní v týdnu.

Prvním datovým zdrojem používaným v interní analytice je aplikace Paymo. V této aplikaci každý zaměstnanec vykazuje své odpracované hodiny na jednotlivých úkolech. Tato aplikace umí rozlišovat i mezi jednotlivými projekty či úkoly, a je proto vhodným datovým zdrojem, díky němuž jsou všechna data přehledně na jednom místě. Z tohoto zdroje jsou čerpána data o odpracovaných hodinách, rozdělená podle jednotlivých zaměstnanců, dnů, projektů, zadání a typů vykazovaných hodin. Zároveň se zde i evidují informace o předem odhadovaných délkách jednotlivých zadání.

Všechna data jsou uchovávána v databázi BigQuery, ze které se poté nahrávají ve finální podobě do GoodData. BigQuery je NoSQL databáze fungující na principech sloupcové struktury. Než je ale možné data nahrát a dostat z nich smysluplný a vypovídající výsledek, je nutné je zpracovat způsoby, které budou detailněji rozebírány v následující kapitole.

## 3 Proces zpracování dat pomocí SQL

V projektu interní analytiky existují tři základní projektové vrstvy. Každá vrstva obsahuje soubory typu SQL, ve kterých dochází ke zpracování dat. Vrstvy jsou od sebe oddělené pro přehlednost číslováním, první vrstva je 00 vrstva, ve které dochází k čištění extrahovaných dat. Druhá vrstva je 10\_common\_data vrstva, ve které dochází k transformacím, které jsou společné pro celý projekt. Třetí vrstvou je 30 vrstva, ve které se provádějí transformace specifické pro jednotlivé tabulky.

### 3.1 Očištění dat

K čištění dat dochází v 00 vrstvě. V této vrstvě se pracuje s daty, která jsou ve své prvotní podobě a nemusí vždy být v ideální stavu k analyzování. Jedním ze základních procesů v 00 vrstvě je očištění dat od prázdných hodnot. Prázdná hodnota vzniká nevyplněním textového nebo číselného pole již v aplikaci, ze které data přicházejí. Defaultně se prázdné hodnoty objevují jako NULL, s touto hodnotou ale nelze moc dobře pracovat a proto se nahrazuje svým adekvátním ekvivalentem na základě datového typu.

Pokud je prázdná hodnota typu string, využívá se k nahrazení NULL hodnoty text '--empty--'. Následující příklad uvádí očištění takových hodnot:

```
CASE WHEN `description` IS NULL OR TRIM(`description`) = '' THEN '--empty--' ELSE TRIM(`description`) END as `description`
```

Pokud je prázdná hodnota číselného typu, v následujícím příkladě jde o integer a bude nahrazen hodnotou 0:

```
CASE WHEN `duration` IS NULL OR TRIM(`duration`) = '' THEN 0 ELSE CAST(TRIM(`duration`) as Int64) END as `duration`
```

Typickým příkladem je také čištění „datumových“ datových typů, které se nedají nahradit ani '--empty--' hodnotou ani 0, proto jsou nahrazovány datem, které je tak vzdálené od aktuálních kalendářních dní, že se při datové analýze dá snadno rozpoznat a vyfiltrovat. Je jím hodnota '1970-01-01 00:00:00'. K očištění tedy dochází takto:

```
CASE WHEN `date` IS NULL OR TRIM(`date`) = '' THEN CAST('1970-01-01' AS DATE) ELSE CAST(TRIM(`date`) AS DATE) END as `date`
```

## 3.2 Transformace dat

Pokud jsou data očistěná, je možné s nimi provádět takové transformace, které odpovídají žádanému výsledku. Jednou takovou transformací, je rozpočet měsíčních plánů na období začínající prvním dnem v měsíci a končící aktuálním dnem v měsíci. Anglicky se tato časová perioda nazývá Month-to-date (MTD). V tomto projektu bude sloužit jako ukazatel hodin, které by k aktuálnímu dni měly být odpracované. Aby se dala perioda MTD používat konzistentně na více faktů v rámci celého projektu, byl vytvořen MTD koeficient, kterým se následně mohou fakta (číselné hodnoty) v rámci projektu násobit.

Pro výpočet koeficientu, je nejprve nutné zjistit, kolik pracovních dní v měsíci už bylo. Následující skript pro tabulku kumulativně sčítá pracovní dny:

```
CREATE OR REPLACE TABLE `tr`.`tmp_workdays_in_month_cumulative` AS
SELECT
`date`,
SUM(`workday_cz_flag`) OVER(PARTITION BY `year_month` ORDER BY
`date` ) AS `workdays_in_month_cumulative`
FROM `tr`.`in_calendar`
WHERE `date` >= '2021-01-01'
;
```

Další číslo, které je potřebné pro získání koeficientu, je celkový/maximálně možný počet pracovních dní v měsíci:

```
CREATE OR REPLACE TABLE `tr`.`tmp_workdays_in_month` AS
SELECT
SUM(`workday_cz_flag`) AS `workdays_in_month`,
CAST(DATE_TRUNC(`date`, MONTH) AS DATE) AS `month`
FROM `tr`.`in_calendar`
WHERE `date` >= '2021-01-01'
GROUP BY `month`
;
```



Tyto dva získané fakty se přes společné měsíce a dny napárují na původní tabulku `tr`.`in\_calendar`, protože je tato tabulka tabulkou výchozí:

```
CREATE OR REPLACE TABLE `tr`.`tmp_workdays` AS
SELECT
`weekday_name_long` AS `weekday`,
c.`date`,
w.`month`,
`day_of_month`,
w.`workdays_in_month`,
x.`workdays_in_month_cumulative`
FROM `tr`.`in_calendar` c
LEFT JOIN `tr`.`tmp_workdays_in_month` w
    ON DATE_TRUNC(c.`date`, MONTH) = w.`month`
LEFT JOIN `tr`.`tmp_workdays_in_month_cumulative` x
    ON c.`date` = x.`date`
WHERE c.`date` >= '2021-01-01'
;
```

Výpočet koeficientu, jako podíl kumulativních dní a celkových dní v měsíci:

```
CREATE OR REPLACE TABLE `tr`.`tmp_coefficient_mtd` AS
SELECT
`weekday`,
`date`,
`month`,
`day_of_month`,
`workdays_in_month`,
`workdays_in_month_cumulative`,
CASE
    WHEN `weekday` IN ('Saturday', 'Sunday')
        THEN CAST((((`workdays_in_month_cumulative` /
`workdays_in_month` ) ) AS FLOAT64)
    ELSE CAST(((( `workdays_in_month_cumulative` - 1) /
`workdays_in_month` ) ) AS FLOAT64)
END AS `coefficient_mtd`
FROM `tr`.`tmp_workdays`
WHERE `date` >= '2021-01-01'
;
```

Tohle je jedno z řešení, jak vypočítat MTD koeficient. Jakákoliv fakta je teď možné vynásobit tímto koeficientem. Takže pokud by bylo žádoucí, vypočítat kolik hodin je podle plánu v této chvíli nutné odpracovat, pouze by se vynásobily celkové plány MTD koeficientem. Výhodou je i to, že koeficient se každý den mění v závislosti na aktuálním dni v měsíci a řešení je tedy vždy aktuální. Pro měsíce, které již uplynuly se koeficient vždy rovná 1, protože plány by už měly být kompletně splněny. Naopak pro měsíce, které teprve nastanou, je koeficient rovný 0, protože ještě není nutné plnit plány za budoucí měsíce. Pro rozlišení měsíců se dá použít tato část skriptu:

```
CASE
WHEN `month` < DATE_TRUNC(CURRENT_DATE(),MONTH)
THEN 1
WHEN `month` = DATE_TRUNC(CURRENT_DATE(),MONTH)
THEN (SELECT `coefficient_mtd` FROM `tr`.`tmp_coefficient_mtd`
WHERE DATE = CURRENT_DATE())
ELSE 0
END AS `mtd_koef`
```

Některá data je nutné transformovat a obohatit. U jiných je možné data rovnou nahrát do platformy GoodData. Vždy záleží na požadavcích klienta a na kvalitě a struktuře dat, která jsou na začátku projektu extrahovaná. Přestože se v prostoru GoodData dají určité datové transformace ještě vytvářet, je dobrým zvykem si většinu dat předpřipravit už na úrovni transformací. Důvodem k tomu je hlavně to, že SQL transformace nabízejí více možností, jak data upravit než GoodData. Tam se dají provádět spíše jednodušší agregace dat, například sumace, aritmetický průměr nebo filtrování podle určitých kritérií.

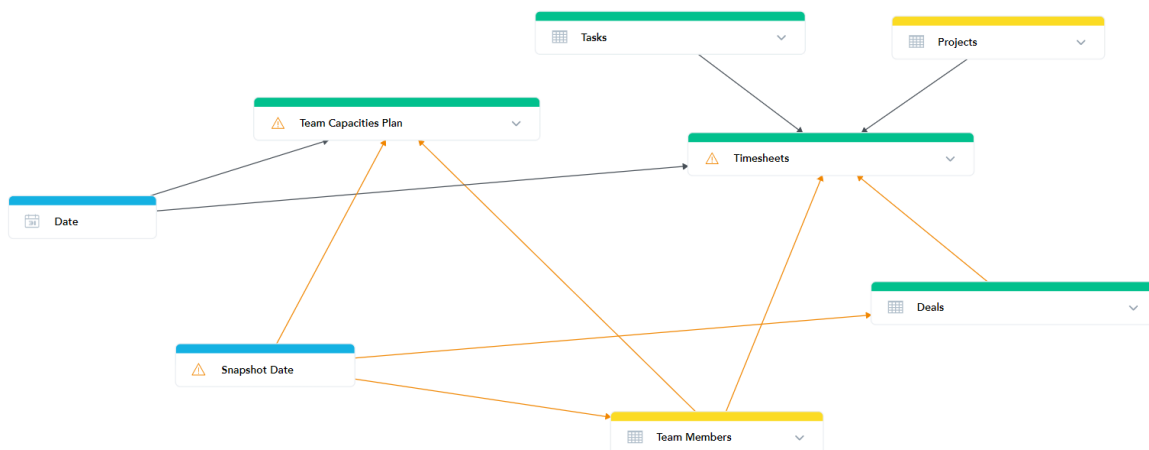
## 4 Vizualizace dat platformou GoodData

K vizualizaci dat byla předem zvolena platforma GoodData, protože nejvíce vyhovovala potřebám společnosti. Než je ale možné od nahraných dat přejít k přehledným dashboardům, je nutné v GoodData udělat několik kroků. Vše začíná vytvořením logického datového modelu nebo modifikací modelu stávajícího. Po vytvoření modelu a jeho namapování, se data můžou nahrát. S nahranými daty se již dá pracovat a vytvářet grafy. Nejprve je ale dobré si v rámci mezikroku vytvořit metriky, které nahraná data ještě více zformují, a až poté z nich vytvářet insighty. Na závěr se z insightů vytvoří dashboard.

### 4.1 Datové modelování v platformě GoodData

V platformě GoodData je možné datově modelovat pomocí datasetů a relací, kterým se zde říká connection points. Mezi datasety patří faktové tabulky a dimenze. Datasety obsahují atributy, fakty, primární a cizí klíče. Mezi speciálními komponenty v datovém modelování jsou zde datумы (Dates), které figurují odděleně od zbytku datasetů. Faktové datasety jsou zelené, dimenze jsou žluté a datумы (Dates) jsou zbarvené modře. Toto rozdělení je velmi dobré pro přehlednost v modelu, obzvláště v modelech složitějších.

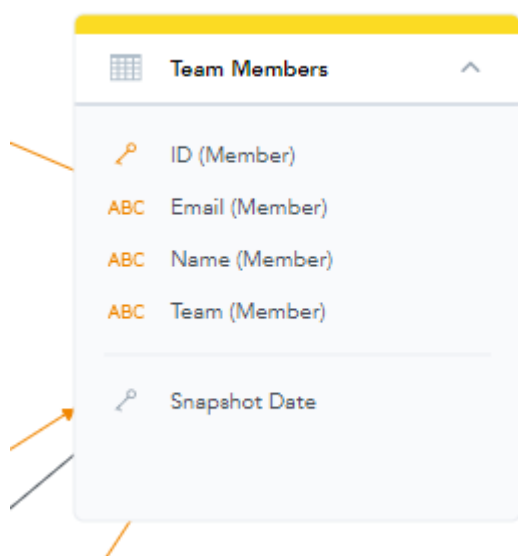
Datasety, které přímo souvisejí s dashboardem se dají také rozdělit na dimenze, faktové tabulky a „datумы“. Mezi faktové datasety, tedy datasety obsahující číselné fakty, se dají zařadit tabulky: Team Capacities Plan, Timesheets, Deals, Tasks. Dimenze v projektu jsou Projects, Team Members. U „datumových“ tabulek rozlišujeme Date a Snapshot Date. Snapshot Date funguje k zálohování dat. Je to datum, které vždy náleží ke dni, ve kterém byla data extrahována. Jeho používáním se dají zpětně dohledat například změny ve zdrojových dokumentech nebo den, kdy se poprvé objevil určitý bug. Datasetů je v celém pracovním prostoru (GoodData workspace) samozřejmě mnohem více, protože se přidávají v závislosti na množství analyzovaných dat. Pro účely této práce a s ohledem na čitelnost je model zjednodušený pouze na datasety, které přímo souvisejí s dashboardem.



Obrázek 3: Ukázka zjednodušeného LDM vytvořeného v platformě GoodData

Zdroj: vlastní tvorba

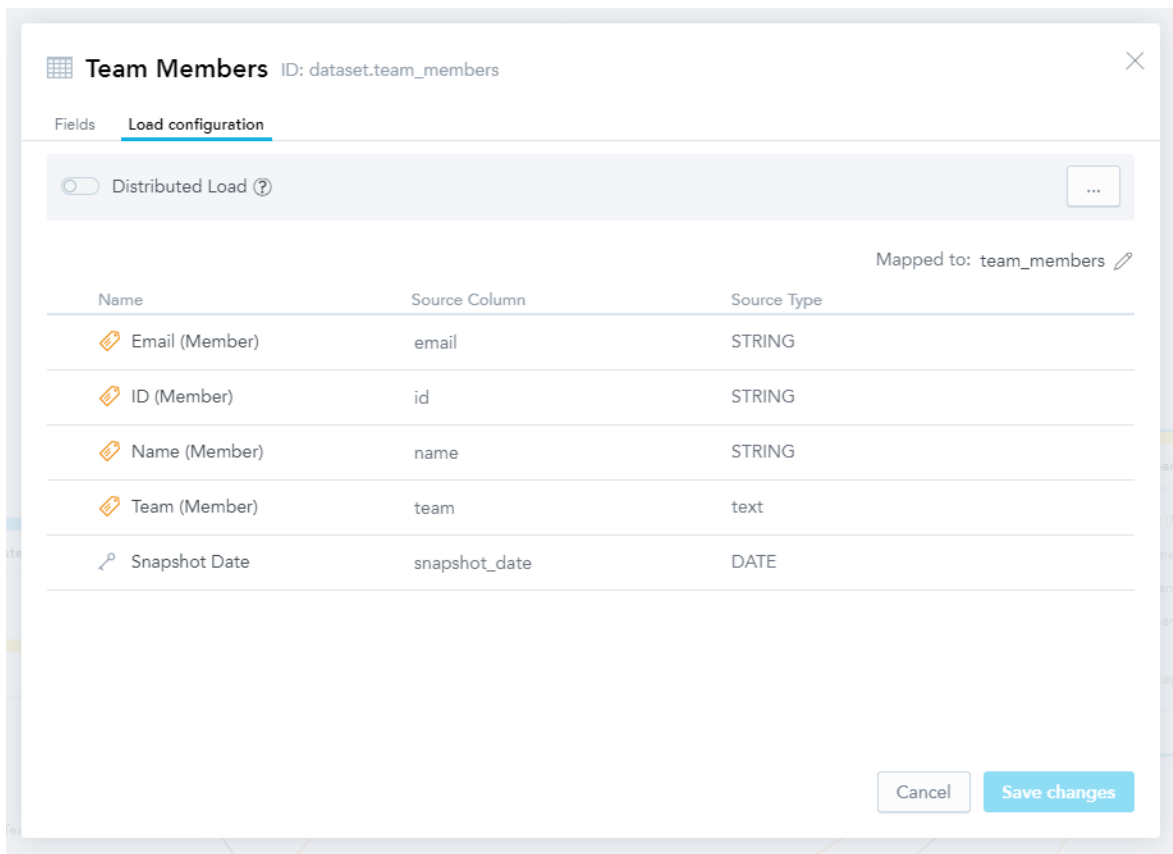
Jednotlivé datasety se dají také rozbalit šipkou vpravo a v editačním módu se do nich dají vložit pole s atributy, fakty nebo klíči. Primární klíče se řadí do prvního řádku datasetu, poté následují atributy, třetí v pořadí jsou fakty a pod šedou čarou se objevují cizí klíče.



Obrázek 4: Rozložený dataset Team Members vytvořený v platformě GoodData

Zdroj: vlastní tvorba

Každý dataset má také záložku Load Configuration, kde je nutné zadat název zdrojového sloupce a název zdrojové tabulky. Tomuto procesu se říká mapování a pokud se vše správně namapuje, mohou se data nahrát. Z nahraných dat se dají vytvářet metriky, které jsou více rozebrány v následující kapitole.



Obrázek 5: Záložka Load Configuration pro mapování z platformy GoodData

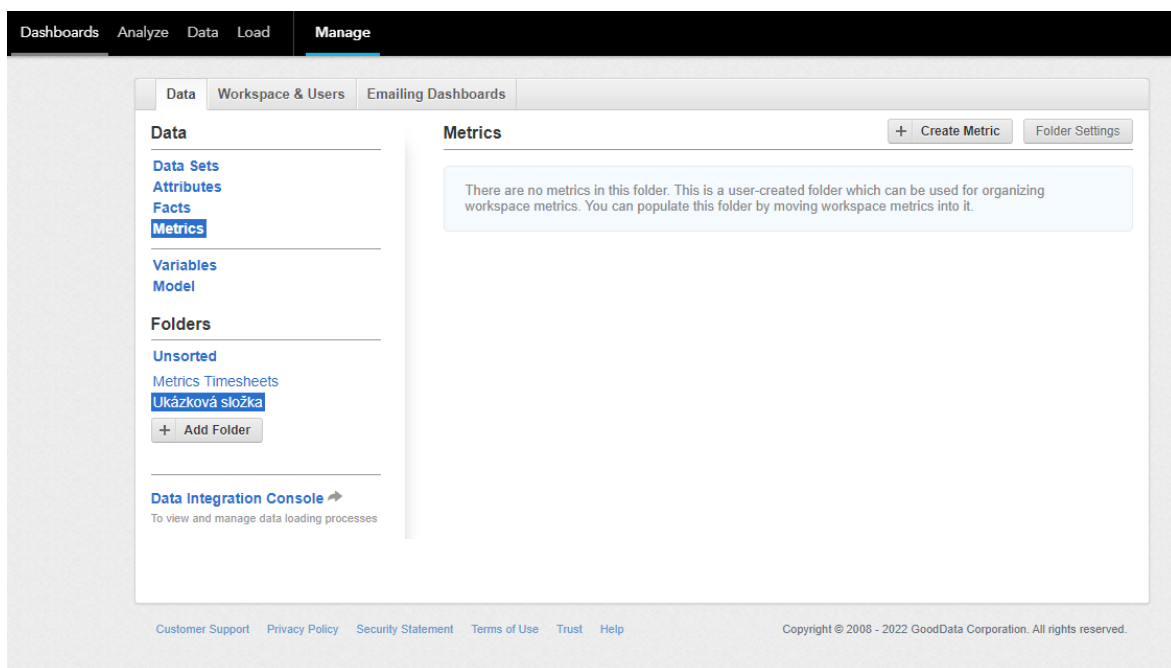
Zdroj: vlastní tvorba

## 4.2 Tvorba metrik

Metriky v platformě GoodData se dají psát nad všemi nahranými daty, samozřejmě s ohledem na jejich logické využití. Pro metriky společnost vyvinula vlastní dotazovací jazyk, který se jmenuje MAQL. Metriky vždy vrací číselný výstup. Jazyk MAQL je velmi podobný a pravděpodobně byl z velké části inspirován jazykem SQL. Uživatel se v něm totiž setká s klíčovými slovy, které již může znát z jazyka SQL. Příkladem takových podobností jsou klíčová slova SELECT, WHERE, SUM, MAX a další.

Po prozkoumání těchto metrik autorka práce zjistila, že je při jejich používání uživatel občas limitován množstvím funkcí, které jsou k dispozici, a občas i logickou strukturou modelu. GoodData stále svoji platformu vylepšuje a některé problémy mohou být tedy v budoucnosti už pouhou historií, není však ale od věci si výpočty, založené na složitějších principech, vyřešit již při transformacích dat pomocí SQL.

Ve workspacu se soubor již existujících metrik dá nalézt pod záložkou Manage. Dále uživatel po pravé straně v menu musí rozkliknout záložku Metrics. Zde se metriky dají rozdělovat do složek a složky si každý uživatel může vytvořit podle vlastních potřeb. Kliknutím na Create Metric se uživatelé dostanou do prostředí, ve kterém mohou vytvořit vlastní metriky. Metriky se tvoří kombinací psaní klíčových slov a vybíráním elementů. Mezi elementy patří: fakty, již existující metriky, atributy, hodnoty atributů a variables. Používání již napsaných metrik v metrikách nových je funkční hlavně při tvorbě složitějších metrik, protože to usnadňuje orientaci v metrice a ulehčuje to práci uživatelům, kteří nemusejí psát metriky se stejným základem několikrát.



Obrázek 6: Složka metrik v záložce Manage z platformy GoodData

Zdroj: vlastní tvorba

Na Dashboardu se nachází několik metrik, mezi ně patří například metrika s názvem „ $\Sigma$ Billable hours to work each day“. Tato metrika má za úkol vypočítat počet odpracovaných hodin, které jsou nutné každý další pracovní den odvést tak, aby byly naplněny odhady za daný měsíc. Kombinací již existujících metrik je možné takovou metriku vytvořit v MAQL například následovně:

```
SELECT(( $\Sigma$  Billable hours planned (Team)- $\Sigma$  Billable hours worked) /Remaining work days)
```

Jednotlivé metriky se také dají dále prokliknout a uživatel se díky tomu může dostat na větší detail. Následující metrika, je proklik z metriky „Remaining work days“. Díky těmto detailním pohledům se dá snadno dohledat, ze kterých faktů či atributů byla metrika vytvořena a případně ji upravit či zkopírovat. Níže uvedená metrika pracuje s hodnotou pracovních dní a s hodnotou aktuálního dne, od kterého je odečten jeden den. Tento odečet zajišťuje, aby koncoví uživatelé, kteří se v průběhu pracovního dne dívají na tuto hodnotu, viděli číslo, které nepočítá aktuální den, tedy dnešek. Je to protože dnešek teprve probíhá a není tedy vhodné, aby s ním bylo nakládáno jako s dnem, který již proběhl.

```
SELECT((SELECT MAX(Work Days Total(Team Capacities Plan))) - (SELECT MAX(Work Day Current (Team Capacities Plan))-1))
```

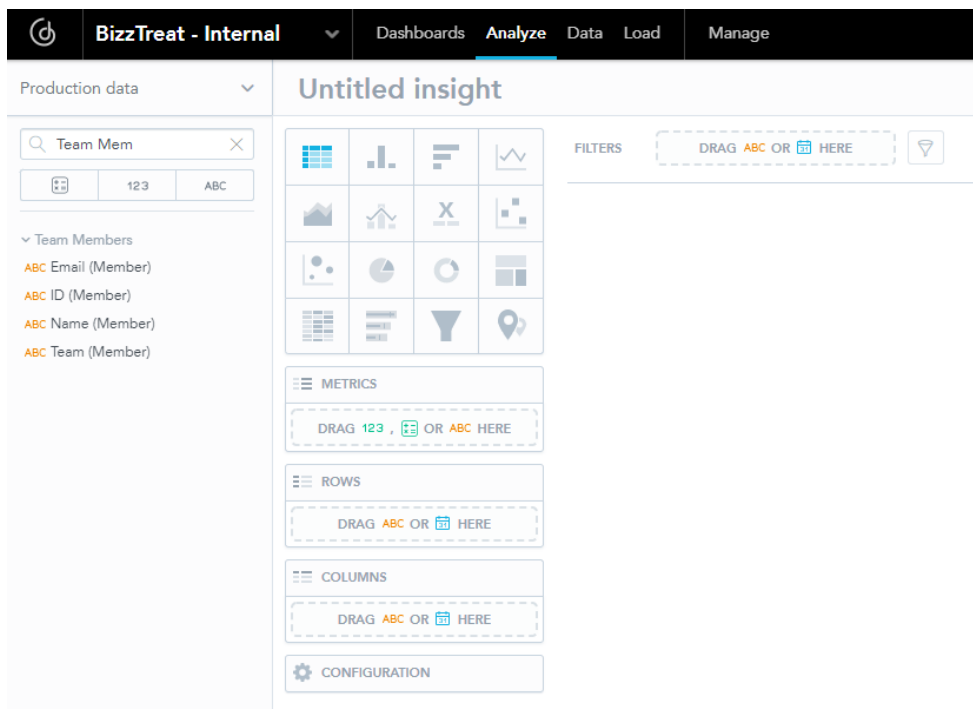
Výše zmíněný odečet jednoho dne byl také přidán do metrik s výpočtem MTD metrik. A to až dodatečně na přání koncových uživatelů. Nevyhovovalo jim totiž, že když se ráno na tento ukazatel dívají, vidí hodiny odpovídající v praxi konci pracovního dne. Což mělo za následek, že každé ráno viděli nesplněné plány na uvedený den. Tento pohled mohl pro některé z nich být nepříjemným, a proto se zavedl odečet jednoho dne od těchto metrik.

Při tvorbě interní analytiky se člověk může často setkat s podobnými situacemi. Je proto vždy nutné s koncovými uživateli aktivně diskutovat a brát jejich požadavky na změny jako klasickou součást tvorby dashboardu. Tato zpětná vazba může být cennou součástí při vytváření projektů, protože je může posunout k uživatelsky přívětivějšímu výsledku. V praxi zároveň mohou být výpočty metrik lehce posunuté od teoretického výpočtu a pro kvalitní fungování projektu je nutné na to brát ohledy.

### 4.3 Insights

Po vytvoření všech metrik, jsou data připravená na poslední fázi úpravy a tou je samotná datová vizualizace. Ta je na platformě GoodData tvořena pomocí takzvaných „insightů“, což jsou různé typy vizualizací. Mezi tyto vizualizace mohou patřit KPI, různé typy grafů či tabulky. Ve workspacu se tvoří tyto insighty v záložce se jménem Analyze. Je zde velké množství možností. Prvním krokem je, vybrat si typ insightu, který bude vytvářen. Tento typ je znázorněn miniaturami ve čtvercích a dá se vždy rozkliknout. Na níže připojeném obrázku, je příklad prostředí, pro tabulku.

Možnosti pro každý insight se liší, například pro tabulky existuje volba metriky či faktů do řádků, dále atributů také do řádků či do sloupců. V některých insightech je možné vložit více atributů či faktů, jinde je jejich počet omezený. Každý insight se musí pojmenovat, což se dá využít i k orientaci ve velkém množství insightů. Každý insight má také možnost filtrování, podle atributů či „datumů“. Všechna data, která zde uživatel chce kombinovat, spolu však musí logicky souviset a musí odpovídat logickému datovému modelu. V opačném případě není totiž možné metriky, atributy či fakta přetáhnout do stejného insightu.



Obrázek 7: Prostředí pro tvorbu insightů z platformy GoodData

Zdroj: vlastní tvorba



### 4.3.1 KPI

Specifickým typem insightu jsou KPI, v platformě GoodData pojmenované jako Headlines. KPI jsou klíčové ukazatele výkonnosti a mohou to být jak absolutně tak relativně vyjádřené ukazatele. Příkladem v tomto projektu je třeba celkový počet odpracovaných hodin v měsíci. Tyto ukazatele přináší obecnou informaci o počtu a mohou být ještě porovnávány se sekundární metrikou. Tou může být například stejná metrika, jako je ta primární, ale uvedené číslo odpovídá minulému období, například měsíci. Nebo to může být i jiná hodnota, která s primární metrikou souvisí například odpracované hodiny jako metrika primární, plány jako metrika sekundární.

Tyto klíčové ukazatele byly vytvořeny pro výsledný dashboard celkem čtyři:  $\Sigma$  Total hours planned,  $\Sigma$  Worked hours, % Utilization a  $\Sigma$  Total hours to work each day. Všechny vyjadřují specifický klíčový ukazatel výkonnosti, který je pro firmu zásadní a je tak důležité, aby tato čísla byla uvedena na prvních místech v dashboardu. Větší detail těchto hodnot se pak dá vyčíst z grafů, které tyto KPI rozšiřují. KPI jsou nedílnou součástí většiny dashboardů.

## 4.4 Dashboard

Finální dashboard se na platformě GoodData skládá celkem ze čtyř KPI, čtyř grafů a jedné doplňující tabulky. Celý dashboard je defaultně filtrován přes měsíc, který začíná prvním dnem v měsíci a končí aktuálním dnem v měsíci. Dny, které ještě neproběhly, se do výpočtů logicky nepočítají. Pokud by měl uživatel zájem se podívat i na uplynulý měsíc, týden nebo dokonce rok, je zde i možnost si nastavit specifické časové období. Ve filtrech se také objevuje filtr na projekty, díky kterému má uživatel možnost se podívat na konkrétní projekty u zákazníků, pro které pracuje. Tento filtr je vhodný zejména pro uživatele, kteří pracují souběžně na velkém množství projektů. Ve všech sekcích byl také přidán popis dat, která uživatel může v dané sekci nalézt. Tento popis přináší zejména novým uživatelům detailnější informace o datech v sekci, protože pro využití dashboardu je nutné, aby uživatelé co možná nejlépe rozuměli zobrazovaným datům. Také tyto popisky přinášejí doporučení, jak data prezentovat a co z nich lze vlastně vyčíst.

Dashboard je rozdělen do čtyř sekcí. Úvodní sekcí je sekce s názvem „Contract“. Zde jsou ty nejzákladnější informace pro uživatele. V této sekci se objevují informace o plánovaných hodinách, odpracovaných hodinách a hodinách, které je nutné odpracovat, aby byl naplněn cílový plán. Dále si zde mohou uživatelé kontrolovat svoji firemní utilizaci, tedy jak velké procento z jejich odpracovaných hodin je výdělečných. Z těchto KPI je poskládaný i bullet chart, na kterém se dají stejné hodnoty sledovat v grafickém zobrazení.

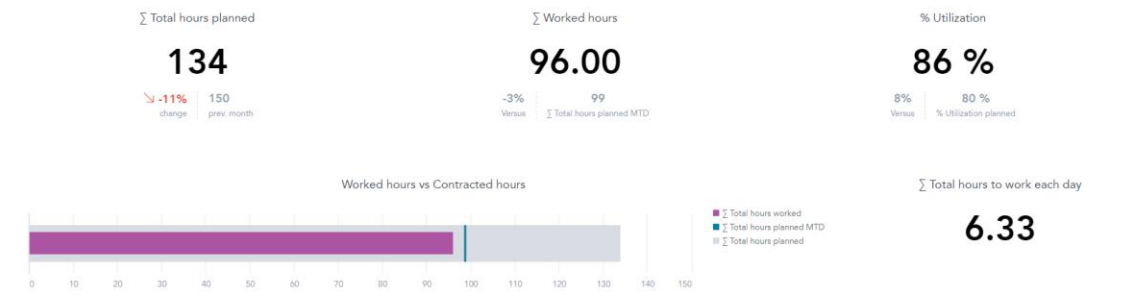
Další dvě sekce jsou více konkrétní a ukazují konkrétní projekty a úkoly, na kterých uživatel daný měsíc pracoval. Projekty se zde zobrazují v rámci časového období, které si uživatel odfiltruje, zatímco pod „Tasks – Estimates“ se naházejí data, která jsou časově neomezená. Tato sekce totiž slouží primárně ke kontrole plánů a odpracovaných hodin na jednotlivých úkolech. Nedává tedy smysl, aby byla časově filtrovaná, protože na zadání se může pracovat i několik měsíců a graf o tyto historická data nesmí být ochuzen. Zároveň pokud na jednom úkolu pracuje více zaměstnanců, promítají se odpracované hodiny všem těm, kteří na úkolu pracují.

Poslední sekcí je sekce kalendář, která slouží primárně ke kontrole vykázaných hodin. Občas se může stát, že zaměstnanec zapomene kompletně vykázat odpracované hodiny za některý den v týdnu a nevšimne si toho. Díky tomuto kalendáři si může snadno zkontrolovat, jestli má u všech všedních dnů vykázané hodiny. Zároveň také vidí, které dny pracoval pod či nad svůj stanovený denní limit a může si zde i kontrolovat, jestli nemá mezi jednotlivými pracovními dny velké výkyvy. Bonusovou vlastností tohoto kalendáře je i barevné rozdělení hodin na výdělečné a nevýdělečné, takže se zde například dá zpětně dohledat, kdyby měl uživatel velké množství nevýdělečných hodin v konkrétním období.

Před uvedením do produkce byl dashboard ještě několik dní testován a kontrolován a byli vytvořeny dodatečné změny, které by mohli usnadnit uživatelům práci s dashboardem. Mezi tyto úpravy patřilo například přidání popisků či filtru pro projekty. Bylo také provedeno hromadné představení dashboardu. Na tomto představení, byly prezentovány jednotlivé sekce dashboardu, vysvětlení uživatelům, jaká data zde najdou a jakými způsoby mohou s dashboardem pracovat. Poté byl dashboard uveden do produkce - začal být aktivně používán a testován samotnými uživateli.

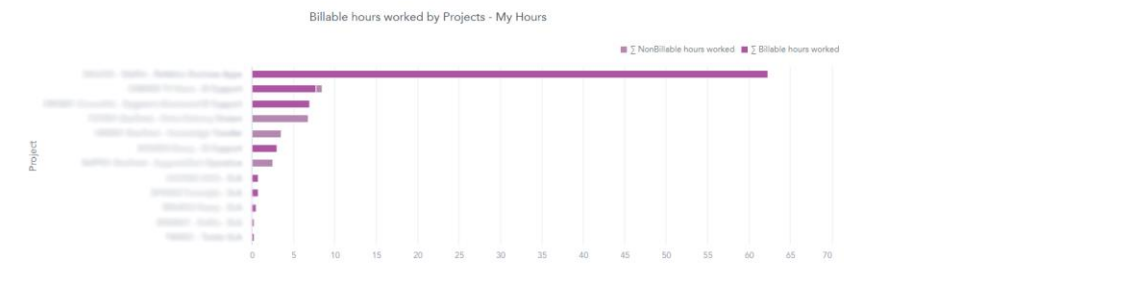
### Contract

How do I fulfill the contract? How many hours should I do this month?  $\Sigma$  Total hours planned contain the contracted hours including your planned holiday. What is my utilization? How many hours do I need to work each day to fulfill the plan?



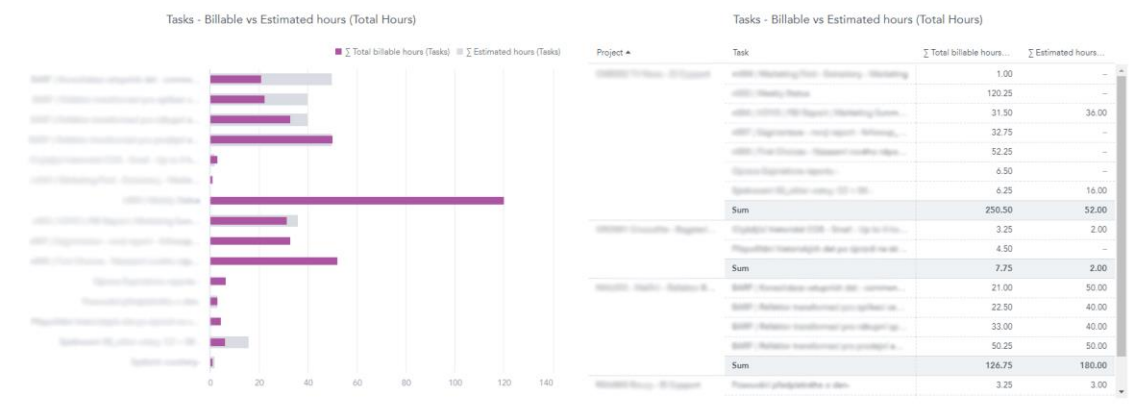
### Projects & Tasks

How many billable vs non-billable hours is in each project? + drill to entries



### Tasks - Estimates

How many billable hours is billed on each task in total? How many hours remain? These hours are from everybody, who worked on the same task as you. (Tasks without SLA and non-billable)



### Calendar

Calculation of billable vs non-billable hours worked each day. + drill to entries



Obrázek 8: Výstupní dashboard vytvořený v platformě GoodData  
Zdroj: vlastní tvorba

## 4.5 Zhodnocení implementovaného řešení

Výsledné řešení vychází z původního návrhu, přináší však i vizualizační doplňky, které v původním návrhu nebyly. Všechny vyžadované části z návrhu byly ve výsledném dashboardu vytvořeny. V mnoha ohledech je ale tento návrh doplněn a rozpracován detailněji. Například se zde objevují navíc popisky jednotlivých sekcí, popisky os grafů nebo je zde přidán filtr na projekty. Zaměstnanci společnosti ohodnotili projekt pozitivní zpětnou vazbou, o čemž vypovídá i to, že dashboard používají na denní bázi. Tento projekt by se tedy dal označit za úspěšný.

Dalšími kroky, které už jsou nadstavbou pro toto zadání, by mohlo být rozšíření dashboardu o takzvané „drill downy“. To jsou prokliky z jednotlivých insightů do ještě detailnějšího pohledu. V těchto detailnějších reportech by se mohly například objevit informace o jednotlivých poznámkách o odpracovaných hodinách. Toto rozšíření by bylo možné použít v sekci kalendář a v sekci Project & Tasks.

## Závěr

Tato bakalářská práce rozebírala kroky a postupy, které vedly k vytvoření dashboardu pro datovou analytiku ve firmě Bizztreat s.r.o. Cílem této práce bylo teoreticky vysvětlit pojmy z oblastí databází, SQL, analytiky, prezentování dat a použít tyto znalosti i prakticky. Tímto praktickým využitím byly zamýšleny jednotlivé části procesu zpracování dat.

V první části této bakalářské práce byla provedena literární rešerše z oblasti dat, datového zpracování, databází, BI, datového modelování, SQL a datové vizualizace. V druhé části práce bylo díky těmto poznatkům navrženo řešení, které odpovídalo požadavkům dashboardu v interní analytice podniku. Řešení bylo zpracováno v platformě GoodData a uvedeno do používání.

Nejprve bylo nutné zjistit v jakém stavu je aktuálně interní analytika a poté se zorientovat v aplikacích, ze kterých jsou čerpaná zdrojová data. Také bylo nutné se seznámit s procesem čištění dat a vytvořit množství SQL skriptů, které data transformují. Pro data byl upraven logický datový model, aby mohla být řádně nahraná do platformy GoodData. V této platformě docházelo ke zpracování dat pomocí metrik. Závěrem byly z těchto dat vytvořeny grafy a klíčové hodnoty, kterými se zaplnil výsledný dashboard.

Výsledkem této práce je dashboard pro zaměstnance firmy, na kterém si na denní bázi kontrolují ukazatele své výkonnosti, plnění měsíčních plánů a hodinové rezervy na projektech. Každý zaměstnanec se tak stává informovanou jednotkou, která se může flexibilně rozhodovat a podle aktuálních dat plánovat svoji každodenní práci.

Tvorba tohoto projektu vyžadovala velké množství, pro autorku projektu často nových, znalostí. Tyto znalosti byly získané z literatury či praktickou zkušeností spojenou s tvorbou výše zmíněných datových procesů. Největším přínosem tohoto projektu je fakt, že je dashboard aktivně využíván již několik měsíců a řídí se podle něj velké množství firemních zaměstnanců. Výsledkem práce je tedy prakticky využitelný projekt s přínosem pro firmu Bizztreat.



## Seznam použité literatury

ABIRAMI, R Nandhini, Seifedine KADRY, Amir H. GANDOMI a Balamurugan BALUSAMY, 2021. *Big Data: Concepts, Technology, and Architecture*. Hoboken: John Wiley. ISBN 978-1-119-70182-8.

ALEXANDER, Michael a John WALKENBACH, 2013. *Excel Dashboards and Reports*. 2. vyd. New Jersey: John Wiley. ISBN 978-1-118-49042-6.

BEAULIEU, Alan, 2020. *Learning SQL: Generate, Manipulate, and Retrieve Data*. 3. vyd. Canada: O'Reilly Media. ISBN 978-1-492-05761-1.

BURIAN, Pavel, 2014. *Internet inteligentních aktivit*. Praha: Grada ISBN 978-80-247-5137-5.

DROBÍKOVÁ, Barbora, Radka ŘÍMANOVÁ, Jiří SOUČEK a Martin SOUČEK, 2018. *Teoretická východiska informační vědy: využití konceptuálního modelování v informační vědě*. Praha: Univerzita Karlova, Karolinum. ISBN 978-80-246-3716-7.

GEMIGNANI, Zach, Chris GEMIGNANI, Richard GALENTINO a Patrick Jude SCHUERMANN, 2015. *Efektivní analýza a využití dat*. Brno: Computer Press. ISBN 978-80-251-4571-5.

HARRINGTON, Jan L., 2016. *Relational Database Design and Implementation*. 4. vyd. Cambridge: Morgan Kaufmann. ISBN 978-0-12-804399-8.

HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK, 2015. *Big Data a NoSQL databáze*. Praha: Grada Publishing. ISBN 978-80-247-5466-6.

JACKSON, Peter a Caroline CARRUTHERS, 2019. *Data Driven Business Transformation: How to Disrupt, Innovate and Stay Ahead of the Competition*. Chichester: John Wiley. ISBN 978-1-119-54315-2.

LAURENČÍK, Marek, 2018. *SQL: Podrobný průvodce uživatele*. Praha: Grada Publishing. ISBN 978-80-271-0774-2.

POUR, Jan, Miloš MARYŠKA, Iva STANOVSKÁ a Zuzana ŠEDIVÁ, 2018. *Self Service Business Intelligence: Jak si vytvořit vlastní analytické, plánovací a reportingové aplikace*. Praha: Grada Publishing. ISBN 978-80-271-0616-5.

WEXLER, Steve, Jeffrey SHAFFER a Andy COTGREAVE, 2017. *The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios*. New Jersey: John Wiley. ISBN 9781119282716.