



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

## ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

# NÁVRH DATOVÉHO SKLADU PRO KONKRÉTNÍ SUBJEKT

DATA WAREHOUSE DESIGN FOR SPECIFIC SUBJECT

## BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

## AUTOR PRÁCE

AUTHOR

Mikuláš Paraska

## VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Luhan, Ph.D., MSc

BRNO 2023

# Zadání bakalářské práce

Ústav: Ústav informatiky  
Student: **Mikuláš Paraska**  
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**  
Akademický rok: 2022/23  
Studijní program: Manažerská informatika

Garant studijního programu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

## Návrh datového skladu pro konkrétní subjekt

### Charakteristika problematiky úkolu:

Úvod  
Cíle práce, metody a postupy zpracování  
Teoretická východiska práce  
Analýza současného stavu  
Vlastní návrhy řešení  
Závěr  
Seznam použité literatury  
Přílohy

### Cíle, kterých má být dosaženo:

Cílem závěrečné práce je návrh datového skladu na základě potřeb daného subjektu. Výstupem práce bude datový sklad zaměřený na nově vytvořenou mobilní hru včetně následné analytiky dle potřeb společnosti.

### Základní literární prameny:

KIMBALL, R. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. New Jersey: John Wiley & Sons Inc, 2013. 600 p. ISBN 978-1-118-53080-1.

LABERGE, R. Datové sklady - Agilní metody a business intelligence. 1. vyd. Praha: Computer Press, 2012. 352 s. ISBN 978-80-251-3729-1.

NOVOTNÝ, O., J. POUR a D. SLÁNSKÝ. Business Intelligence: Jak využít bohatství ve vašich datech. 1. vyd. Praha: Grada, 2005. 256 s. ISBN 80-247-1094-3.

POUR, J., M. MARYŠKA, I. STANOVSKÁ a Z. ŠEDIVÁ. Self Service Business Intelligence: Jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. 1. vyd. Praha: Grada Publishing, 2018. 352 s. ISBN 978-80-271-0616-5.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2022/23

V Brně dne 5.2.2023

L. S.

---

Ing. Jiří Kříž, Ph.D.  
garant

---

doc. Ing. Vojtěch Bartoš, Ph.D.  
děkan

## **Abstrakt**

Předmětem bakalářské práce je navrhnout a vytvořit datový sklad pro startup pohybující se v herním průmyslu. Datový sklad bude navržen pro nově vytvořenou mobilní hru. Hlavní částí práce je zpracování jak teoretického, tak i praktického návrhu datového skladu podle konkrétních požadavků firmy. Jeho funkcionality bude následně otestována pomocí reportovacích a vizualizačních nástrojů.

## **Abstract**

The subject of the bachelor thesis is to design and create a data warehouse for a startup operating in the gaming industry. The data warehouse will be designed for a newly developed mobile game. The main parts of the thesis include both theoretical and practical design of the data warehouse according to specific company requirements. Its functionality will be subsequently tested using reporting and visualization tools.

## **Klíčové slova**

Datový sklad, Business intelligence, Google BigQuery, reporting, ETL, Tableau, databáze, SQL

## **Keywords**

**Data warehouse, Business intelligence, Google BigQuery, reporting, ETL, Tableau, database, SQL**

### **Bibliografická citace**

PARASKA, Mikuláš. *Návrh datového skladu pro konkrétní subjekt* [online]. Brno, 2023 [cit. 2023-05-11]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/152133>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Ing. Jan Luhan, Ph.D., MSc.

### **Čestné prohlášení**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 11. 5. 2023

---

Mikuláš Paraska

autor

## **Poděkování**

Rád bych touto cestou poděkoval vedoucímu práce, panu Ing. Janu Luhanovi, Ph.D., za jeho cenné rady, doporučení a poskytování zpětné vazby během vypracování práce. Dále bych chtěl poděkovat hernímu studiu za poskytnutí všech potřebných informací a prostředí ke zpracování této práce. A hlavně děkuji své rodině a přátelům za jejich projevenou důvěru a podporu během studia.

# Obsah

Úvod.....	11
Cíl a metodika práce .....	12
<b>1. Teoretická východiska práce.....</b>	<b>13</b>
1.1 Datový sklad.....	13
1.1.1 Systém datového skladu.....	13
1.1.2 Architektura datového skladu .....	14
1.1.3 Terminologie toku dat.....	15
1.1.4 Účel datového skladu .....	15
1.2 Datové tržiště.....	17
1.3 OLAP databáze.....	18
1.4 Business intelligence .....	19
1.5 Dolování dat .....	21
1.6 Datové Formáty .....	22
1.6.1 JSON .....	22
1.6.2 XML.....	23
1.6.3 YAML.....	23
1.6.4 Formáty Linked Data .....	24
1.6.5 CSV .....	24
1.7 Datové modelování.....	25
1.8 Dimenzionální modelování .....	25
1.8.1 Tabulky faktů .....	26
1.8.2 Tabulky dimenzí.....	27
1.9 Logický model.....	27
1.10 ETL – Extract, Transform, Load .....	29
1.11 SQL.....	30
1.12 Reporting .....	31
<b>2. Analýza současného stavu .....</b>	<b>32</b>
2.1 Informace o společnosti.....	32
2.1.1 Organizační struktura .....	32
2.2 Informace o hře.....	33
2.3 Unity .....	34
2.3.1 IronSource .....	34

2.4	Firestore.....	34
2.4.1	Firestore Cloud hosting.....	34
2.4.2	Firestore ceník.....	35
2.4.3	Firestore integrace .....	36
2.5	Zdroje dat.....	37
2.5.1	Firestore Události.....	37
2.5.2	Ironsource Události .....	38
2.5.3	AdWords Události.....	38
2.6	Bigquery .....	38
2.6.1	Bigquery ceník .....	39
2.7	Tableau .....	40
<b>3.</b>	<b>Vlastní návrh řešení .....</b>	<b>42</b>
3.1	Plán projektu.....	42
3.1.1	Cíl projektu.....	42
3.1.2	Požadavky projektu.....	42
3.1.3	sekvence úloh.....	43
3.2	Plán budování datového skladu .....	43
3.3	Návrh datové architektury .....	43
3.4	Nakonfigurování nástroje Firestore.....	44
3.4.1	Postup pro propojení Firestore s Unity .....	44
3.4.2	Definování událostí: .....	46
3.5	Nakonfigurování nástroje Ironsource .....	51
3.5.1	Napojení Ironsource na Unity projekt.....	51
3.5.2	Definování událostí Ironsource: .....	52
3.6	Analýza zdrojových systémů.....	53
3.6.1	Google AdWords .....	53
3.6.2	Firestore .....	54
3.7	Popis datové struktury .....	54
3.7.1	Vlastní návrh datové struktury .....	54
3.8	Vytvoření datového skladu.....	55
3.8.1	Propojení zdrojových systémů s datovým skladem .....	56
3.8.2	Definování tabulek .....	58
3.8.3	ELT .....	59
3.8.4	Záloha.....	62

3.9	Tvorba reportu .....	63
3.10	Zhodnocení projektu.....	64
	<b>Závěr.....</b>	<b>66</b>
	<b>Seznam použitých zdrojů .....</b>	<b>67</b>
	<b>Seznam použitých zkratk a symbolů .....</b>	<b>69</b>
	<b>Seznam použitých obrázků a tabulek.....</b>	<b>71</b>

## Úvod

S nástupem internetu se podnikání stalo mnohem rozmanitějším a ziskovějším. V dnešní době podle statistik vlastní necelých sedm miliard lidí chytrý telefon. Díky tomuto trendu se zvětšuje poptávka po mobilních hrách. Herní studia jsou motivována zvýšením zisku, a proto potřebují provádět informovaná rozhodnutí, která musí být podložena daty.

V současné době jsou data klíčovým faktorem pro rozvoj podnikání. S nárůstem digitalizace se vytváří a ukládá obrovské množství dat. Podle EXPLODING TOPICS se množství dat vytvořených v roce 2022 odhaduje na přibližně 97 zettabajtů, což je více než dvojnásobek množství dat v roce 2019. Toto množství dat se každým rokem zvyšuje a organizace potřebují tyto data shromažďovat.

Pro shromažďování dat se využívají Datové sklady. Datové sklady jsou obecně využívány v mnoha odvětvích, například v obchodních společnostech, vládních organizacích a vzdělávacích institucích. Herní průmysl však také shromažďuje obrovská množství dat a má stále větší potřebu pro efektivní správu a analýzu těchto dat. Herní studia v současné době využívají datové sklady pro ukládání a analýzu dat z her, které se hrají online.

Díky datovým skladům jsou herní studia schopna získávat cenné informace o hráčském chování, preferencích a výkonnosti. Tyto informace umožňují studiím lépe porozumět svým uživatelům, personalizovat hry, optimalizovat herní zážitky a provádět strategická rozhodnutí pro další rozvoj. Vytvoření a efektivní správa datového skladu je proto pro herní studia klíčová pro využití potenciálu dat a získání konkurenční výhody v rychle se rozvíjejícím prostředí herního průmyslu.

## Cíl a metodika práce

Hlavním cílem této bakalářské práce je vytvořit návrh a implementaci datového skladu pro mobilní hru, která byla nově vytvořena v herním studiu. Vzhledem k tomu, že firma, pro kterou je datový sklad navrhován, již používá datové sklady v jiných projektech, je nezbytné zajistit, aby datová struktura tohoto projektu byla konzistentní s ostatními datovými sklady, a aby koncoví uživatelé nemuseli procházet dalším školením. Bude se tedy klást důraz na navržení datového skladu tak, aby se jeho architektura a schéma dat shodovalo s existujícími datovými sklady, což usnadní přechod a používání pro stávající analytický tým.

Práce bude členěna do 3 hlavních kapitol. V první části budou vysvětleny teoretická východiska, které jsou úzce spjaté s návrhem a implementací datového skladu. V analytické části bude nejdříve představena herní společnost, pro kterou je datový sklad navrhován. Následně se analýza bude zabývat popisem jednotlivých nástrojů a softwaru, které budou v návrhové části využívat. V poslední části toho projektu proběhne samotné budování datového skladu. Budování datového skladu bude rozvrhnuto do jednotlivých částí. Do těchto částí se bude řadit konfigurování zdrojových systémů, modelování datového skladu a proces čištění pomocí ELT metody. V poslední fázi tohoto projektu bude otestována funkčnost datového skladu ve vizualizačním nástroji pomocí reportů.

# 1. Teoretická východiska práce

První část této bakalářské práce se zaměřuje na teoretická východiska, včetně podrobného vysvětlení jednotlivých pojmů a metodik, které jsou úzce spojeny s tvorbou datového skladu. Na základě těchto teoretických znalostí bude vypracovaný návrh a implementace datové skladu.

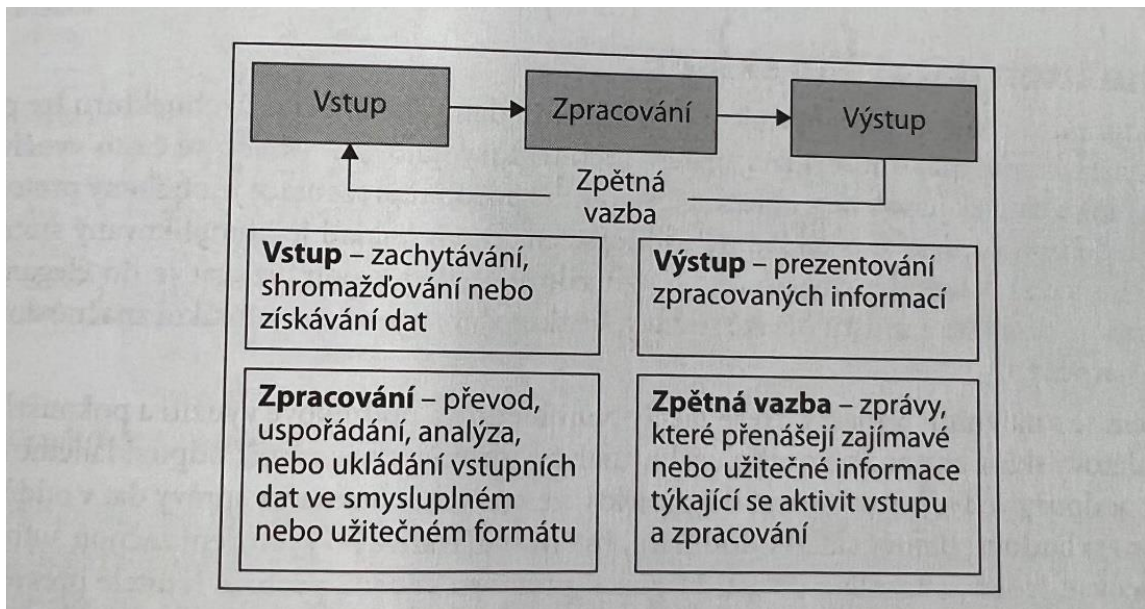
## 1.1 Datový sklad

Datový sklad představuje důležitý nástroj pro shromažďování, správu a uchovávání historických dat. Tyto údaje pocházejí převážně z provozních systémů, které je sbírají a využívají v rámci svého určení. Tyto systémy mohou být i jiného druhu, avšak v kontextu datového skladu se nejčastěji jedná o provozní systémy. Tyto sklady mohou mít více zdrojů a být zaměřeny na celý podnik, nebo na specifický obor činnosti. Tyto údaje jsou organizovány a uspořádány tak, aby bylo možné snadno získat potřebné informace pro různé účely, jako je například analýza nebo reporting. Datový sklad tak umožňuje společností udržovat a sdílet data, která jsou pro ně důležitá, a umožňuje jim využít své historické údaje k dosažení lepšího rozhodování a řízení. [1.]

### 1.1.1 Systém datového skladu

Stejně jako u jiných systémů, i datový sklad se skládá z klíčových komponentů, jako jsou vstup, zpracování, výstup a zpětná vazba. Vstup do datového skladu se týká identifikace a záznamu dat. Kvalita dat je v této fázi klíčovým faktorem, protože nesprávná data mohou vést k nepřesným výstupům a ovlivnit tak závislé procesy, subsystémy a následné analýzy a rozhodnutí. Transformace a načítání dat do centrálního prostředí datového skladu může zahrnovat fázi vstupu i zpracování. Tyto kroky jsou nezbytné pro udržení kvality dat a pro zajištění správných výstupů. Transformace a nahrávání dat do centrálního prostředí může zahrnovat proces vstupu i zpracování. Jádro celého prostředí obvykle obsahuje jednu databázi, ale může se jednat o sbírku databází. Bez ohledu na konfiguraci by však měly být centrálně řízeny. Toto centrální prostředí plní důležitou roli při transformaci a uchování dat v řádně strukturovaném a uspořádaném formátu. Struktura dat je založena na datové architektuře, přesněji řečeno na datovém modelu. Výstupní část systému zajišťuje přenos dat uživatelům. Data se obvykle rozdělují

do kombinace datových trhů, které mohou v závislosti na své definici přesahovat výkonnostní vrstvu a oblast nebo vrstvu uživatelské prezentace. Část zpětné vazby systému je založena na vstupu a výstupu. Výsledky odvození nebo agregace dat mohou být v budoucnosti nutné. [1.]

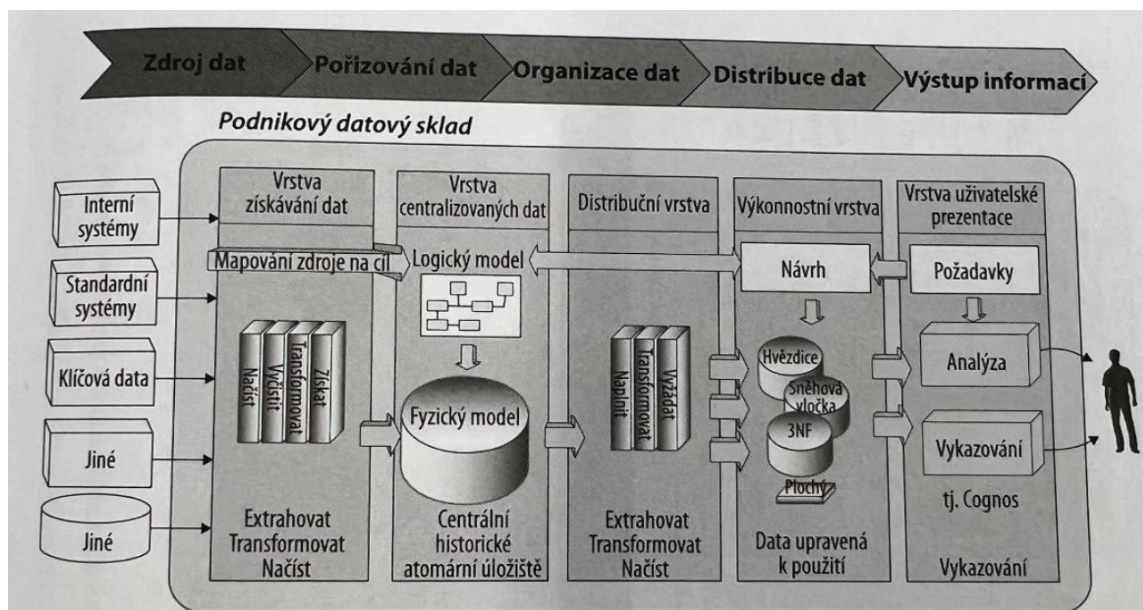


Obrázek 1: Základní komponenty systému  
(Zdroj: 1)

### 1.1.2 Architektura datového skladu

Architektura datového skladu popisuje návrh systému datového skladu. Architekturu lze přirovnat ke stavebním plánům. Při znázornění architektury se často využívají diagramy toku dat. Architekturu datového skladu členíme na jednotlivé fáze, které se dělí na [1.]:

- Zdroj dat – vnitřní systémy, standardní systémy, klíčová data a další
- Získání dat – vrstva získávání dat, mapování zdrojů na cíl
- Organizace dat – vrstva centralizace dat, logický model, fyzický model
- Distribuce dat – distribuční a výkonnostní vrstva, data připravená k použití
- Výstup informací – vrstva uživatelské prezentace, výkaz dat



Obrázek 2: grafické znázornění architektury datového skladu

(Zdroj:1)

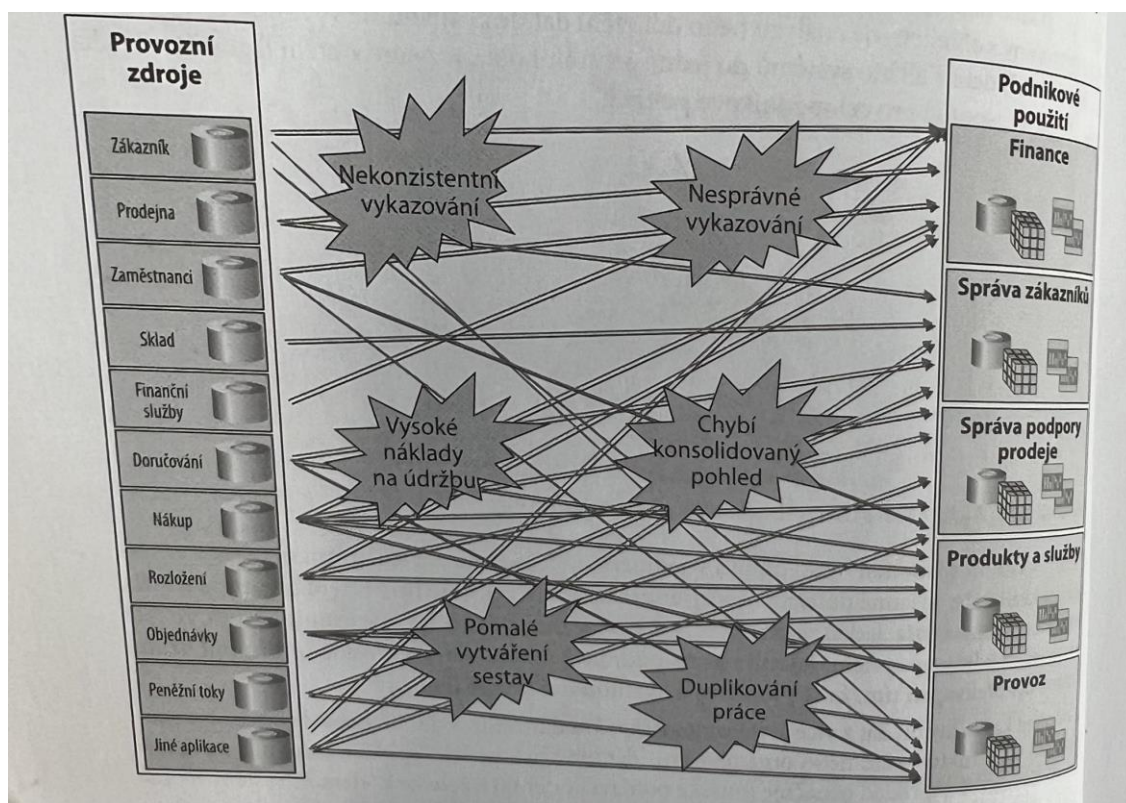
### 1.1.3 Terminologie toku dat

V oboru datových skladů se často používají termíny shora dolů (angl. top-down) a zdola nahoru (angl. bottom-up). Tyto fráze vycházejí z architektury toku dat. Datový tok lze rovněž zkoumat v kontextu toho, zda data vstupují do systému nebo z něj vystupují. Metodika Billa Inmona se k datovým skladům přibližuje hlavně „shora dolů“, což znamená, že se zaměřuje na data, ale nechybí zde ani podnikový účel. Naopak, metodika Ralpa Kimballa vede „zdola nahoru“. Tento přístup klade důraz na podnikový cíl a data slouží k jeho podpoře. Obě metody vedou ve skutečnosti ke stejnému konečnému cíli, to znamená k tomu, aby podnik na základě uspořádání a strukturování základních dat mohl přijímat informovaná rozhodnutí. [1.]

### 1.1.4 Účel datového skladu

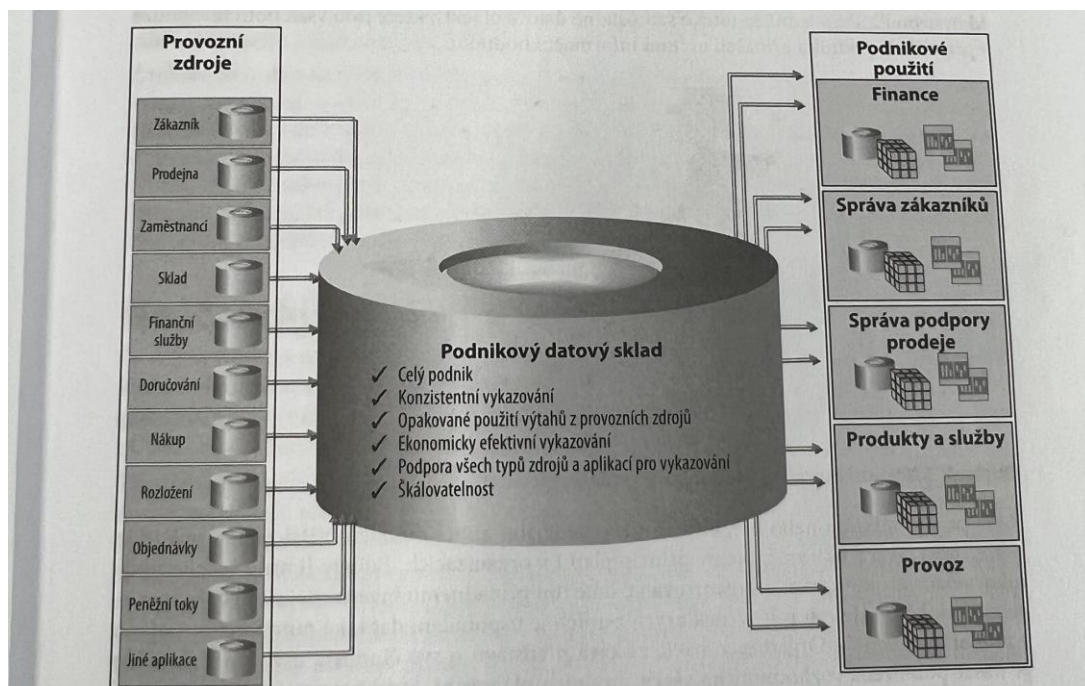
Prostředí datového skladu má takovou strukturu, aby mohlo organizovaným způsobem uchovávat historická data, které jsou integrována z jiných zdrojových systémů. Provozní systémy bývají často navrženy pro specifické funkce. Jedná se o například o řízení pokladních terminálů, fakturační systémy, skladové hospodářství a další. Některé z těchto systémů nemusí patřit do podnikové třídy a nejsou navrženy s ohledem na analýzu nebo dolování dat. Je proto nutné vytvořit nové prostředí, které slučuje data z těchto systémů

do jedné centrální oblasti zvané datový sklad, kde budou data k dispozici pro celopodnikové využití. Potíže s kvalitou dat se mohou vyskytnout i pouze u jednoho provozního systému, ale při spojení dvou a více nezávislých systémů je kvalita dat nesmírně důležitá. Vzhledem ke slučování dat z více systému musí správa datového skladu věnovat pozornost strukturování nebo organizování dat, aby byl zajištěn stejný slovník. Například můžeme mít subjekt A, která označuje mužské pohlaví kódem 1 a ženské kódem 2, druhý subjekt B pracuje s kódy M (mužské) Ž (ženské) a N (neznámé). Hlavním důvodem pro existenci datového skladu spočívá v tom, že odstraňuje „špagetový“ zmatek, který vzniká, při již zmíněné kombinaci odlišných systémů [1.]



Obrázek 3: grafické znázornění špagetového systému  
(Zdroj: 1)

Centralizované "atomární" úložiště datového skladu řeší většinu těchto problémů. Pokud centralizujete data celého podniku, vznikne společný slovník, který uživatelům poskytuje jednotnou terminologii. Všechna data také mohou být vysoké kvality. Každý objekt, který je importován do úložiště centrálního datového skladu, musí totiž zpravidla projít analýzou a profilováním. Při vykazování z centrálního prostředí je možné se spolehnout na data každé sestavy. Jestliže vzniknou jakékoliv problémy, lze všechny datové komponenty zpětně sledovat až k jejich zdrojům. [1.]



Obrázek 4: grafické znázornění centralizovaného systému  
(Zdroj: 1)

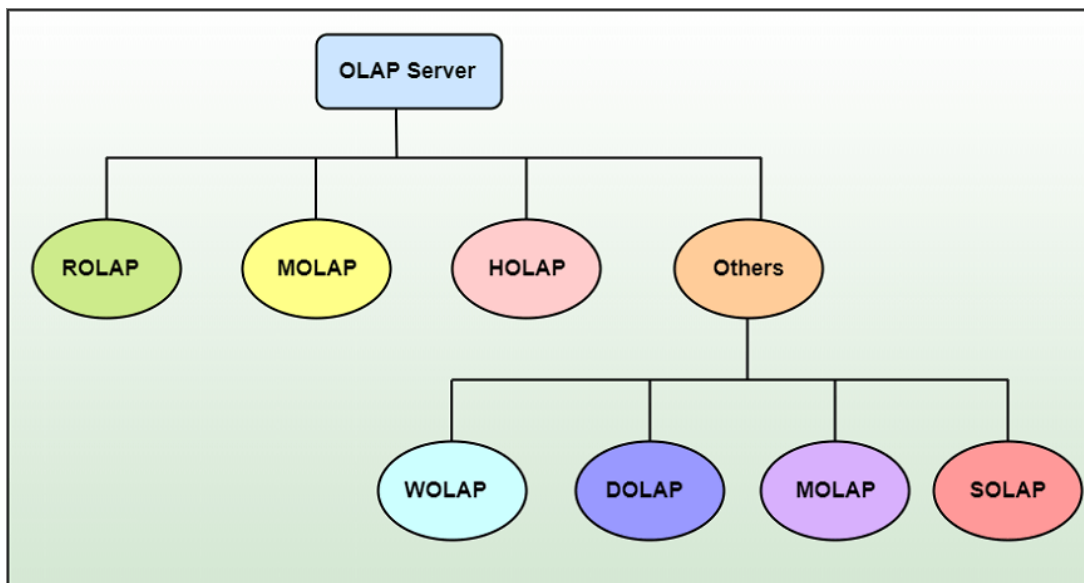
## 1.2 Datové tržiště

Princip datových tržišť (angl. Data Marts) je velmi podobný principu datových skladů. Rozdíl je v tom, že datové tržiště jsou určena jen pro specifický okruh uživatelů například to mohou být oddělení, divize, pobočka, závod a podobně. Datové tržiště je pak interpretované jako decentralizovaný datový sklad, který se budou postupně integrovat do celopodnikového řešení. V některých případech slouží Data Marts, i po vytvoření celopodnikového datového skladu, jako přechod při transformaci dat z produkčních databází. Data Mart je tak problémově orientovaný datový sklad, určený pro pokrytí konkrétní problematiky daného okruhu uživatelů a umožňující flexibilní jednorázovou analýzu. Vytváření datových trhů má za následek zkrácení doby návratnosti investic do těchto projektů. Díky tomu, že se snižují náklady na získávání a analýzu dat, snižuje se také riziko při zavádění nových projektů, které se na těchto datech zakládají. To znamená, že firmy a organizace mohou s větší jistotou investovat do inovací a nových řešení, protože mají k dispozici kvalitní a relevantní data, na kterých mohou stavět. [2.]

### 1.3 OLAP databáze

OLAP (Online Analytical Processing) je typ databáze která je určena pro analýzu dat a umožňuje provádět komplexní analýzy dat v reálném čase. Skládají se v jednu nebo několika souvisejících OLAP kostek. Ty většinou, na rozdíl od datových skladů, již zahrnují předzpracované agregace dat podle definovaných hierarchických struktur dimenzí a jejich kombinací. Technologie OLAP si realizuje v několika různých variantách [1.]:

- MOLAP (Multidimensional Online Analytical Processing): Tyto databáze ukládají data ve formě multidimenzionálního pole
- ROLAP (Relational Online Analytical Processing) - Tyto databáze ukládají data v relační databázi
- HOLAP (Hybrid Online Analytical Processing) - Tyto databáze kombinují výhody MOLAP a ROLAP databází a používají kombinaci multidimenzionálního pole a relačního modelu dat
- DOLAP (Desktop Online Analytical Processing) se vztahuje k analýze dat, která se provádí na desktopových počítačích a kde data jsou uložena lokálně na disku [5.]
- SOLAP (Spatial Online Analytical Processing) se zaměřuje na analýzu prostorových dat, kde se kromě tradičních číselných dat ukládají také geografické informace, jako jsou například souřadnice, adresy a mapové podklady [5.]
- WOLAP (web-based Online Analytical Processing) je typ OLAP databáze, který se zaměřuje na webovou technologii a umožňuje uživatelům analyzovat data prostřednictvím webového prohlížeče. [5.]



Obrázek 5: typy OLAP databáze

(Zdroj: 5)

## 1.4 Business intelligence

Definice business intelligence se týká všech aspektů, které umožňují efektivní rozhodování ve firmě. Tyto aspekty zahrnují znalosti, procesy, technologie, aplikace a postupy. Tyto nástroje pracují s daty a informacemi z minulosti, aby pomohly při plánování a rozhodování pro budoucnost. Business intelligence se zaměřuje na interní informace o podnikových procesech a fungování, které jsou nutné pro taktické a strategické plánování. Tyto informace se zpravidla zpracovávají a získávají z aktuálních podnikových procesů. Tyto základní data se využívají hlavně pro interní analýzu, ale mohou se také kombinovat s externími analýzami, jako je například analýza SWOT a PEST. Analýza SWOT poskytuje informace, které pomáhají přiřadit podnikové prostředky a možnosti k parametrům konkurenčního prostředí, v němž společnost působí. Analýza pest je přehled hlavních faktorů externího prostředí, ve kterém společnost funguje. [2.]

Cílem technologie business intelligence je prostřednictvím počátečních klíčových indikátorů ukazatelů výkonu stanovit cíle podniku. Tyto klíčové ukazatele výkonu se dají rozdělit na metriky nebo fakta, které přímo odpovídají provozním systémům organizace, nebo jsou shromažďovány z těchto základních provozních metrik. Fakta bývají často v číselných a kvantifikovatelných podobách. Lze je tedy počítat. V rámci business

intelligence se analytické zkoumání často uskutečňuje na agregované úrovni nežli na úrovni jednotlivých instancí. Společnost například potřebuje zjistit, kolik zákazníku pochází z jednotlivé země, a kolik z těchto zákazníku pochází z určitého regionu, nikoli kde bydlí konkrétní osoba a kolik toho nakoupila. Když nastane druhá varianta, tak ji můžeme odvodit ze stávajícího provozního systému. Stejnou informace lze získat ze systému business intelligence za předpokladu, že systém BI uchovává data na stejné úrovni podrobnosti. Některé podniky využívají pro uložení dat systém ODS (angl. Operational data store). Tato uložení však mívají limitovanou historii, takže nejsou tak často používána. [2.]

Dobré systémy BI se vyznačují jejich přesností a aktuálností dále také mají vysokou hodnotu a umožňují rozhodování. Přesnost ve smyslu, že lze datům důvěřovat, aktuální znázorňuje, že dostupnost dat se řídí pravidelným rozvrhem, vysoká hodnota se týká užitečnosti pro podnikové uživatele a možnost rozhodování souvisí s uplatněním informace v podnikovém rozhodovacím procesu.

Současné systémy BI podnikovým uživatelům umožňují, aby se na data dotazovali přímo. Díky tomuto opatření se podařilo rozhodovací proces zkrátit z několika hodin, či dokonce dnů na pouhé minuty. [2.]



**Obrázek 6: Business Intelligence**

(Zdroj: 7)

## 1.5 Dolování dat

Data mining pomocí speciálních algoritmů umožňuje automaticky odhalovat strategické informace v datech. Tato analytická technika je pevně spojena s datovými sklady, které představují kvalitní zdroj dat pro tyto speciální analýzy. [2.]

Data mining se vyznačuje jako proces extrakce významných, neznámých nebo nedefinovaných informací z velkých databází. Hlavním rysem data miningu je, že se jedná o analýzy získané z obsahu dat, nikoliv předem určené uživatelem nebo implementátorem, a zaměřuje se především na odvození prediktivních informací, nikoliv pouze deskriptivních. Data mining slouží cílovým uživatelům k objevování nových zjištění, umožňuje testování hypotéz a odhaluje skryté vztahy mezi ekonomickými proměnnými. [2.]

Existuje několik typů nástrojů pro dolování dat. Řada z nich je určena specialistům, kteří se zabývají statistikou, některé nástroje jsou určeny pro řídicí pracovníky. Cíle data miningu jsou stejné jako cíle BI, mají tedy poskytovat strategické informace, které umožňují efektivní rozhodování ve firmě. Dolování dat se odlišuje od statistických nástrojů tím, že cílí na odlišné uživatele. Statistické úlohy dolování dat jsou naprogramovány podle předdefinovaných algoritmů a jejich proces je zcela automatizován, a tak jejich cílovým uživatelem může být manažer bez jakýchkoliv znalostí statistiky, nikoliv pouze specialista, který vytváří analýzy pro manažera. [2.]

Dolování dat využívá množství matematických a statistických metod. Zde jsou ty nejpoužívatelnější [2.]:

- Rozhodovací stromy: je prediktivní model, kde jsou data reprezentována jako strom s kořenem a větvemi, které vedou ke koncovým uzlům. V každém uzlu rozhodovacího stromu je testována jedna z příznaků dat a na základě výsledku testu jsou data přesunuta do dalšího uzlu, dokud není dosaženo koncového uzlu, kde jsou data klasifikována jako jedna z předem definovaných kategorií. Rozhodovací stromy mohou být založeny na množství algoritmů. Například ID3, CART a CHAID. Rozhodovací stromy jsou velmi častou technikou především díky své snadné implementaci.

- Neuronové sítě: jsou nejčastější metodou pro tvorbu prediktivních modelů. Fungují na podobných principech, které jsou inspirovány biologickým chováním mozku. Tyto sítě se skládají z mnoha umělých neuronů, které spolu komunikují prostřednictvím propojení, a tím umožňují učení, klasifikaci a predikci na základě předchozích zkušeností.
- Genetické algoritmy: je technika simulující evoluci pro dedikování, jak by měly být atributy formovány, vyvíjeny, modifikovány atd.
- Clustering a klasifikace: Clustering (shlukování) je proces rozdělení datové sady na skupiny, tzv. shluky, tak, aby data v každém shluku byla si podobná a data v různých shlucích si byla odlišná. To znamená, že objekty v jednom shluku by měly být si podobné, zatímco objekty v jiných shlucích by měly být si odlišné. Klasifikace zase definuje podstatné atributy skupin v podobě klasifikačních kritérií.

## 1.6 Datové Formáty

Většina aplikací dnes funguje na webových prostředích, proto je znalost datových formátů velmi důležitá. Aplikace je strukturována tak, že klíčové části jsou implementovány jako služby, které jsou využívány dalšími částmi aplikace, jako například klientskou aplikací napsanou v JavaScriptu, která běží přímo ve webovém prohlížeči. Mezi jednotlivými částmi aplikace dochází k výměně dat v přesně definovaném formátu, V ideálním případě jsou datové formáty pro komunikaci a ukládání dat shodné, což usnadňuje integraci a zpracování dat v celé aplikaci. [4.]

### 1.6.1 JSON

Datový formát JSON (Javascript Object Notation) původně vznikl pro zasílání dat mezi serverovou a klientskou částí webové aplikace. Patří do podmnožiny jazyka Javascript, tato technologie poskytuje možnost reprezentovat základní datové struktury a usnadňuje jejich použití přímo v prohlížeči. Postupem času se stal JSON rozšířeným datovým formátem, a je možné ho používat ve většině programovacích jazycích pomocí knihoven. Díky své struktuře je JSON snadno mapovatelný na objekty v daném jazyce, což zjednodušuje práci s daty v různých aplikacích a umožňuje efektivní přenos dat mezi různými systémy. JSON umožňuje reprezentovat čtyři jednoduché datové typy (řetězec,

číslo, logická hodnota a null) a dva strukturované (objekty a pole). Pole a objekty pak mohou jako své indexy obsahovat libovolný další datový typ. JSON je velmi flexibilní formát a umožňuje reprezentaci téměř jakéhokoliv druhu datové struktury. [4.]

### 1.6.2 XML

Značkovací jazyk XML neboli extensible markup language vychází jako zjednodušení z komplexního značkovacího jazyka SGML. Značkovací jazyky doplňují prostý text o strukturu elementů, která mu dá význam. Původně se tyto značkovací jazyky využívali na elektronické publikování pro reprezentování velkých dokumentů s upevněnou strukturou a potřebou lepšího propojení pomocí odkazů. Nejvíce se XML využíval jako reprezentaci právních textů nebo technické dokumentace. [4.]

V minulosti byla velké poptávka po formátu, který by uměl snadno propojovat systémy a zároveň mezi nimi vyměňovat data. XML tyto nároky splňovalo, a tím se stal nejpoužívanějším formátem pro výměnu dat. Vznikly i složitější nadstavby, jako jsou webové služby SOAP nebo WSDL, které samostatné XML doplnili o další vrstvy jako je například autentizace, rozhraní atd. Před nástupem datového formátu JSON se používala technika dnes známá jako AJAX. Její funkcionalita fungovala, že webová aplikace byla napsána v Javascriptu a běžela v prohlížeči. A právě pro přenos dat mezi serverem a prohlížečem se používal formát XML, jediný problém byl ten, že XML podporuje jen rozhraní DOM, které není uživatelsky přívětivé. Místo XML se tak začal používat JSON. [4.]

### 1.6.3 YAML

Datový formát YAML (aint markup language) není zdaleka tak rozšířeným datovým formátem jako je XML nebo JSON, ale i tak je možné se setkat se systémy, které YAML využívají. Tento formát byl navržen pro ukládání dat, tak aby byla pro člověka dobře srozumitelná a čitelná. YAML lze mapovat na datové běžně používané v programovacích jazycích. Jednodušší dokumenty se podobají formátu JSON a odpovídají tomu i základní datové typy. YAML má tu výhodu, že se snaží uživateli zaručit lehčí ruční zápis souborů, jeho syntaxe nabízí velmi široké spektrum jednotlivých zkratk. Kromě jednoduchého zápisu může jazyk YAML nabídnout pokročilé funkce, které v XML nebo JSON

nenajdeme, avšak tyto funkce komplikují syntaxi a implementaci jazyka. Pro zpracování dat se, tak jazyk YAML moc nerozšířil, tak jako XML a JSON. [4.]

## **1.6.4 Formáty Linked Data**

Linked data je iniciativa, která vychází ze sémantického webu a má za cíl propojit různé zdroje dat a umožnit nad nimi provádět dotazy. Tento koncept umožňuje reprezentovat data pomocí datového modelu RDF, který se skládá ze tří částí: subjektu, predikátu a objektu. Základem tohoto modelu jsou URI, která se používají k jednoznačnému identifikování subjektů a predikátů. Díky tomu lze sdílet identifikátory napříč doménami a využívat jednoho identifikátoru pro jednu věc. [4.]

### **1.6.4.1 RDF/XML**

RDF/XML je jedním z nejstarších formátů pro ukládání RDF datového modelu v podobě XML dokumentu. Tento formát není považován za optimální, jelikož je méně přehledný a úsporný v porovnání s novějšími formáty. V dnešní době se proto RDF/XML příliš nepoužívá a nahrazují jej modernější a účinnější alternativy, jako je například formát Turtle nebo JSON-LD. [4.]

### **1.6.4.2 JSON-LD**

Formát JSON-LD využil vzrůstající potenciál formátu JSON ve webovém prostředí, a tím navrhl metodu, jak RDF graf přenášet v podobě dokumentu JSON. Většina dokumentů JSON-LD vypadá celkem přirozeně, používá se jen několik vlastností se speciálními klíči. Strukturovaná data lze zařazovat přímo do webových stránek, kde je mohou využít vyhledávače. Lze využít i kódovací jazyk HTML, kde lze do kódu přidat několik speciálních atributů, které do stránky doplní potřebnou sémantiku. [4.]

## **1.6.5 CSV**

Formát CSV (comma-separated Values) jedná se o jednoduchý formát pro ukládání dat, která mají podobu tabulky. Každý řádek textového souboru obsahuje jeden záznam, který bývá často ohraničen pomocí oddělovačem, který má znázorňovat sloupce. Ohraničující znak bývá ve většině případů středník, čárka nebo dvojtečka. První řádek

většinou značí názvy položek. Formát CSV je ale až příliš primitivní a nemá jasně definovanou formu zápisu. Mohou se používat různé oddělovače položek, velmi špatně se definují položky, které obsahují stejnou hodnotu jako má oddělovač, a v neposlední řadě neexistuje ani způsob, jak do souboru uložit informaci o použitém kódování. Kvůli těmto nedostatkům je dobré se formátu CSV vyhnout. Každopádně z historických důvodů je CSV stále velmi používaným formátem. [4.]

## **1.7 Datové modelování**

Datové modelování je proces, který se využívá k navrhování a dokumentaci struktury dat v databázích. Jeho účelem je přispívat k vytvoření kvalitního návrhu databází, které jsou konstruovány efektivně s minimalizací redundance dat a zajištěním požadovaného výkonu a doby odezvy při provádění dotazovacích operací. Databáze jsou navrhovány tak, aby poskytovaly uživatelům data, která odpovídají jejich aktuálním a budoucím požadavkům. To znamená, že databáze by měla být schopna v budoucnu rozšiřovat, upravovat a udržovat. Integrita dat je zajištěna v relačních databázích na úrovni položek, tabulek a vazeb mezi tabulkami. Relační databáze jsou navrženy tak, aby udržovaly přesnost uložených dat po celou dobu provozu. Tento koncept se opírá o tzv. cizí klíče, které definují vazby mezi tabulkami a zajišťují konzistenci dat v celé databázi. Tyto klíče jsou jedním z důležitých prvků zajištění integrity a spolehlivosti relačních databází. [3.]

## **1.8 Dimenzionální modelování**

Dimenzionální modelování se využívá pro naplnění účelu analytických aplikací business intelligence. Které mají za úkol prezentovat uživatelům potřebné informace co nejjednodušším způsobem. Dále by měli poskytovat odpovědi na dotazy s minimální dobou odezvy a v neposlední řadě zajišťovat relevantní informace podnikovým procesům. [3.]

Dimenzionální modely a na nich postavené aplikace jsou pro uživatele mnohem pochopitelnější než normalizované modely pro transakční aplikace. Uložení dat a operace s nimi jsou zde založeny na multidimenzionalitě, což umožňuje efektivnější a rychlejší analýzu dat. V centru schématu je tabulka faktů, tedy tabulka sledovaných hodnot.

Dimenzionální tabulky slouží jako uložení vlastností, tedy většinou jako popisné informace. [3.]

### 1.8.1 Tabulky faktů

Jak už bylo zmíněno tabulka faktů je tabulka sledovaných hodnot ekonomických a dalších ukazatelů identifikovaných klíčem složeným z cizích klíčů dimenzionálních tabulek, nebo vlastním umělým primárním klíčem. Cizí klíče potom slouží pouze pro vazby dimenzionálních tabulek. Tabulky faktů se tak vážou na tabulky dimenzí, ale tabulky faktů nelze vázat do další tabulky faktů. [3.]

#### 1.8.1.1 Granularita v tabulce faktu

Granularita tabulky faktu se vztahuje k úrovni detailu nebo zrnitosti dat, které jsou uloženy v centrální tabulce datové skladu. Granularita určuje, jak detailní jsou faktická data v tabulce faktu a jak jsou agregována. [3.]

**Vysoká granularita** znamená, že centrální tabulka obsahuje podrobné informace o událostech, transakcích nebo jiných událostech, což umožňuje velmi detailní analýzu dat. Například tabulka faktu s vysokou granularitou by mohla obsahovat každou prodejní transakci, která zahrnuje informace o produktech, cenách, časech a zákaznících. [3.]

Na druhé straně **nízká granularita** znamená, že tabulka faktu obsahuje vyšší úroveň agregovaných dat, což umožňuje rychlejší analýzu dat na úrovni vyššího přehledu. Například tabulka faktu s nízkou granularitou by mohla obsahovat pouze celkové prodeje za každý den, což by umožnilo rychlé srovnání výkonnosti prodeje v průběhu času. [3.]

#### 1.8.1.2 Typy tabulek faktů

Tabulky faktů rozdělujeme na tři hlavní typy vzhledem k jejich granularitě dat.

- **Transakční tabulky faktů:** jsou struktury v datovém skladu, které slouží k ukládání dat o transakcích nebo událostech a pohybují se na nejvyšší možné granularitě dat. Transakční tabulky bývají v praxi často využívány.
- **Periodické snímované tabulky faktů:** jsou struktury, které ukládají agregované informace o transakcích a událostech v pravidelných intervalech (například ve dnech) a vyjadřují souhrnné hodnoty ukazatelů za celý časový snímek (například

celkový objem transakcí za určitý časový usek) tento typ tabulek se používá pro predikci trendů vybraných ukazatelů, a je v praxi nejpoužívatelnější.

- **Akumulované snímkové tabulky faktů:** Jsou rovněž závislé na výskytu transakcí, ale jejich hodnoty se v čase postupně aktualizují. Tyto tabulky se v praxi moc nevyužívají.

## 1.8.2 Tabulky dimenzí

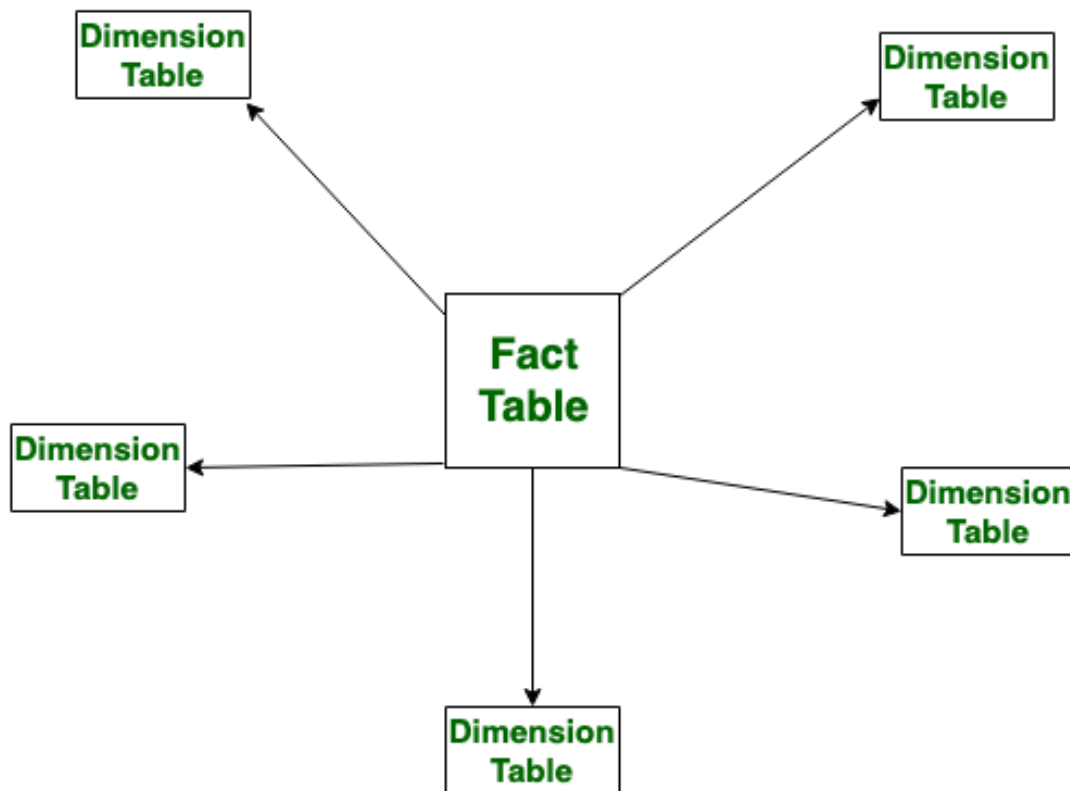
V dimenzionálních modelech se používají tabulky nazývané dimenzionální tabulky. Tyto tabulky obsahují atributy, které popisují vlastnosti hodnot uložených v tabulce faktů. Dimenzionální tabulky tedy slouží jako úložiště popisných a doplňujících informací o hodnotách v tabulce faktů. Můžeme si je představit jako podnikové číselníky, například dimenze zboží, zákazníků, dodavatelů atd. [3.]

## 1.9 Logický model

Logický model datového skladu je abstraktní popis struktury datového skladu, který se zaměřuje na vztahy mezi tabulkami a atributy. Logický model datového skladu zahrnuje schéma tabulek faktů a dimenzí, které popisují faktická data a metadata, jako jsou popisy, definice a vztahy mezi tabulkami. Logický model datového skladu je obvykle vytvořen v nástrojích pro modelování dat, jako jsou ER diagramy, UML diagramy a další. [3.]

### 1.9.1.1 Schéma STAR

Je architektura, kdy centrální tabulka faktů obsahuje všechna faktická data a je spojena s dimenzionálními tabulkami pomocí jednoduchých vztahů. Schéma Star je jednoduchá, snadno srozumitelná a rychlá při dotazování. Tato architektura je vhodná pro prosté a stabilní datové sklady. [3.]

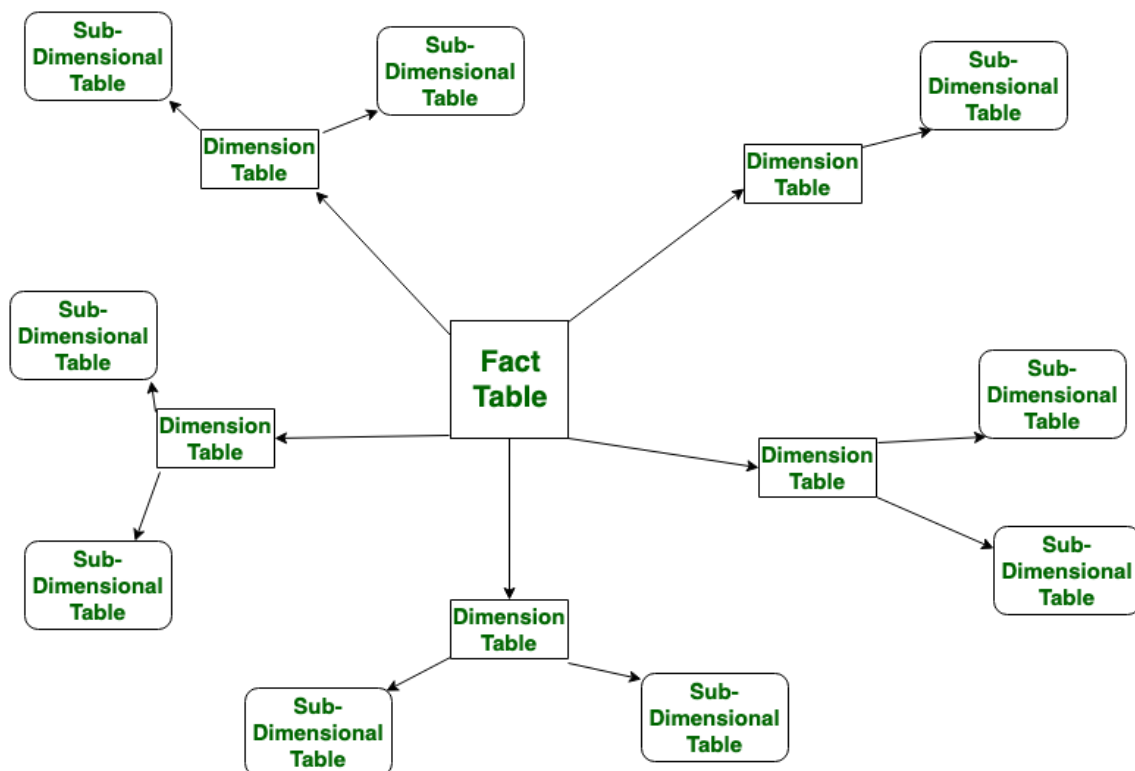


Obrázek 7: Schéma STAR

(Zdroj: 6)

### 1.9.1.2 Schéma SNOWFLAKE

Na druhé straně, Snowflake schéma je složitější architektura, kde dimenzionální tabulky obsahují hierarchické vztahy mezi atributy a jsou propojeny pomocí složitých vazeb. Tato architektura umožňuje větší flexibilitu a složitější analýzy, ale může být pomalejší při dotazování a složitější při správě. V Snowflake schematu jsou dimenzionální tabulky často normalizovány do více tabulek, což vede k větší složitosti a nižšímu výkonu při dotazování. [3.]



Obrázek 8: Schéma SNOWFLAKE

(Zdroj: 6)

## 1.10 ETL – Extract, Transform, Load

ETL je velmi důležitou součástí celého komplexu business intelligence. Název ETL bývá v praxi označován jako datová pumpa. Jak si můžeme povšimnout ze zkratky, tak ETL se dělí na tři části. První část se nazývá Extrakce a jejím úkolem je data ze zdrojového systému vybrat. Druhou částí je transformace, který má následná data upravit do požadované formy a uspořádat. Poslední fází je nahrávání kdy se mají data nahrát do specifických datových struktur, respektive do datového skladu. [2.]

ETL nástroje jsou často využívány pro přenos dat mezi různými databázovými systémy nebo formáty dat, jako je například CSV nebo XML. Data mohou být přenášena najednou v předem definovaných intervalech, například denně, týdně nebo měsíčně, v závislosti na potřebě nových dat. Během procesu ETL je vynakládána značná pracovní kapacita na transformaci dat, což zahrnuje jejich čištění a přípravu pro nahrání do cílové databáze. Tato úloha obvykle zabírá až 60% celkové pracovní kapacity. [2.]

Aby proces ETL fungoval správně je nutno dodržovat určité podmínky [2.]:

- Ze zdrojových systému musí být vybrána jenom ty data, která budou využívána ve prospěch analytického, plánovacího a rozhodovacího procesu firmy, tedy nikoliv veškerá data společnosti.
- Data jsou převedena na nové datové struktury pro databáze, které musí být navrženy předem tak, aby nejefektivněji odpovídaly potřebám podnikového řízení.
- Jelikož jsou většinou data tahána z více zdrojových systému, tak je nutné tyto data vyčistit od případných duplicit či multipllicit, jelikož může nastat situace, kdy budeme mít stejná data a k tomu v různé kvalitě v různých zdrojových systémech.
- Je nutné zajistit konsolidaci dat, aby byla zaručena potřebná kvalita, což zahrnuje eliminaci chyb a nepřesností.

## 1.11 SQL

SQL – Structured Query Language (Strukturovaný dotazovací jazyk) je programovací jazyk používaný pro práci s relačními databázemi. Je v první řadě určen uživatelům, avšak v dnešní době ho využívají i tvůrci aplikací. SQL se dá využívat v jakémkoliv prostředí. [8.]

SQL není pouze jazyk pro dotazování, ale také umožňuje definovat strukturu tabulky a naplnit sloupce daty, definovat vztahy a organizaci dat. SQL také umožňuje řízení přístupu k datům, tedy udělování a odebrání přístupových oprávnění na různých úrovních, což chrání data před náhodným nebo úmyslným zničením, neautorizovaným čtením nebo manipulací s nimi. Dále SQL umožňuje sdílené využívání dat a zajišťuje plynulý průběh činností při přístupu k datům více uživatelů současně. [8.]

SQL se skládá z různých příkazů, které provádějí různé úkoly v relační databázi například [8.]:

SELECT: používá se pro vyhledání dat z tabulky nebo více tabulek v databázi.

INSERT: používá se pro vkládání nových záznamů do tabulky.

UPDATE: používá se pro aktualizaci existujících záznamů v tabulce.

DELETE: používá se pro odstraňování záznamů z tabulky.

Díky jednoduché syntaxi se SQL stal nejpoužívanějším nástrojem pro práci s relačními databázemi a je využíván v mnoha aplikacích.

## 1.12 Reporting

Reporting je proces, kdy se data z různých zdrojů shromažďují a organizují do srozumitelných a užitečných informací. Cílem reportingu je prezentovat klíčové ukazatele výkonnosti a poskytnout podrobný pohled na aktuální stav podniku. [2.]

V BI reportingu se používají různé nástroje a technologie pro vizualizaci dat, včetně tabulek, grafů, dashboardů, a dalších interaktivních prvků. Reporting může být realizován v reálném čase, nebo s určitým zpožděním v závislosti na potřebách uživatelů a dostupnosti dat. [2.]

V rámci reportingu lze identifikovat [2.]:

- **Standardní reporting:** kdy jsou v určitých časových periodách spouštěny předdefinované dotazy
- **Ad hoc reporting:** je vytváření přizpůsobených reportů na základě aktuálních potřeb uživatele, tyto reporty bývají jednorázové.

## 2. Analýza současného stavu

V této části si rozeberme společnost, ve které pracuji a pro kterou datový sklad navrhuji. V analýze budou popsány veškeré informace k mobilní hře, pro kterou se datový sklad vytváří. Dále se bude analýza zabývat popisem veškerých nástrojů a využívaného softwaru. Poslední částí bude výběr vhodného datové skladu pro navržený projekt.

### 2.1 Informace o společnosti

Jak už bylo zmíněno, tak společnost, ve které datový sklad vytvářím se zabývá vývojem her na mobilní zařízení. Svoji kariéru začala již v roce 2005, kdy zakladatelé začali vyvíjet hry pro webové platformy. Postupně se rozrostla a začala vyvíjet hry pro chytré mobilní telefony a tablety. V roce 2023 už má na svém kontě desítky úspěšných her v různých kategoriích. Díky své dlouhodobé přítomnosti na trhu má firma zavedené principy a metodiky, které bude potřeba při tvorbě datového skladu dodržet.

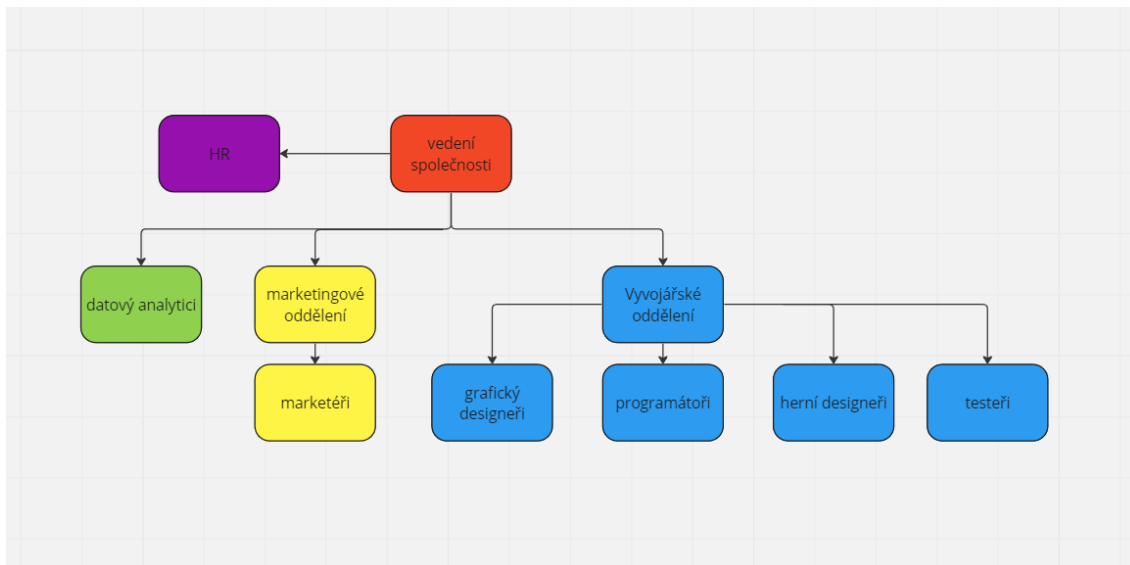
#### 2.1.1 Organizační struktura

Organizační struktura hraje klíčovou roli pro správný chod a směr společnosti. Tato struktura se skládá z oddělení zaměřených na různé aspekty vývoje her. Každé oddělení má své specifické úkoly, zodpovědnosti a hierarchii a spolupracuje s ostatními odděleními na vytvoření a úspěšném dokončení projektů. Organizační struktura se dělí na 3 hlavní oddělení, a každá z těchto částí potřebuje dostávat klíčové informace pro rozhodování. Tyto klíčové informace podávají datoví analytici, kteří jsou nezbytnou součástí celé společnosti. Zde jsou rozděleny jednotlivé části:

**Vedení společnosti:** řídí celkovou strategii firmy, zahrnující rozhodnutí ohledně nových projektů, zaměstnanců a zázemí. Oddělení potřebuje dostávat klíčové informace o zisku z produktů.

**Marketingové oddělení:** zaměřuje se na propagaci her, tvorbu reklamních kampaní a vytváření marketingových materiálů. Oddělení potřebuje dostávat klíčové informace ohledně výsledků kampaní a celkové propagace.

**Vývojářské oddělení:** Tvoří hlavní práci celého vývoje mobilních her, skládá se z několika profesí jako jsou herní designeři, programátoři, testeři. Oddělení potřebuje dostávat klíčové informace ohledně uživatelských interakcí.



**Obrázek 9: Organizační struktura**

(Zdroj: Vlastní zpracování)

## 2.2 Informace o hře

Projekt, pro který je datový sklad vyvíjen je akční mobilní hra, ve které hráči sestavují vlastní bojové tanky a účastní se soubojů proti tankům jiných hráčů v aréně. Hráči mohou své stroje vylepšovat a upravovat, aby byly úspěšnější v soubojích. Cílem hry je získávat body za vítězství a postupovat do vyšších arén. Hra má jednoduché ovládání, které umožňuje hráčům snadno sestavit své vlastní stroje a přizpůsobit je podle svých preferencí. Souboje mezi stroji jsou rychlé a intenzivní, což zaručuje zábavu a napětí. Hráči mohou také soutěžit v různých ligách a turnajích s různými výzvami a odměnami.

V současné době je tato hra spuštěna v beta testování pouze pro určité regiony a zároveň se neustále iteruje a vyvíjí. Jednotlivá oddělení potřebují klíčové informace pro další iterace, což znamená, že výstavba datového skladu musí být začleněna již v rámci beta testování.

## 2.3 Unity

Unity je herní engine, který byl založen roku 2005. Herní engine můžeme chápat jako software určený pro tvorbu a vývoj interaktivních her pro různé platformy jako jsou například počítače, mobilní zařízení, herní konzole nebo virtuální realitu.

Tento software je postavený na programovacím jazyce C#. Unity nabízí možnost vyvíjet hry ve 2D nebo 3D grafice, takže si vývojář může svobodně vybrat výtvarný styl, který chce pro svůj projekt využít. Tento nástroj nabízí mnoho funkcí pro tvorbu her včetně vizuálního editoru, správy animací, správy fyzikálního chování objektů a mnoho dalšího.

Díky svému výkonu integraci a příjemnému uživatelskému prostředí je Unity jeden z nejpoužívanějších herních nástrojů na světě. Pro většinu vývojářů je Unity zdarma, pakliže se vývojář nedostane na zisk ve výši 100 000 dolarů. [10.]

### 2.3.1 IronSource

IronSource je společnost, která byla v roce 2022 koupena společností Unity. Tato platforma slouží pro správu a monetizaci mobilních aplikací. Pomocí svých nástrojů vkládá do těchto aplikací reklamu, díky kterým získává nové uživatele, pomáhá zvyšovat zisky a analyzovat data. [9.]

## 2.4 Firebase

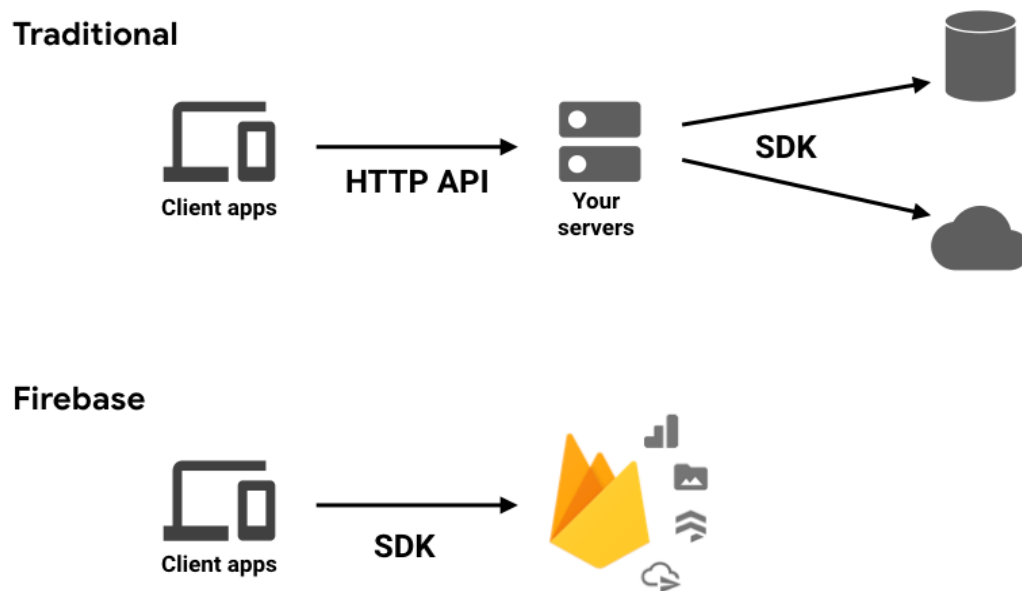
Je platforma poskytující nástroje pro vývoj mobilních a webových aplikací. Tyto nástroje vám pokrývají velkou část služeb, které by vývojáři museli normálně tvořit jako jsou například databáze v reálném čase, cloudové úložiště, autentizace uživatelů, zasílání zpráv a mnoho dalšího. Všechny tyto služby jsou hostovány v cloudu a škálovatelné s malým nebo žádným úsilím ze strany vývojáře. [11.]

### 2.4.1 Firebase Cloud hosting

Hostováno v cloudu znamená to, že backendové komponenty produktů jsou plně spravovány a provozovány společností Google. Backendové služby jsou přímo interagovány s klienty pomocí SDK poskytovaných Firebase, aniž by bylo nutné zřizovat

jakýkoli middleware mezi aplikací a službou. Pokud je použita jedna z možností Firebase databáze, obvykle je kód pro dotazování se na databázi napsán v klientské aplikaci. Tradiční vývoj aplikací obvykle zahrnuje psaní jak frontendového, tak backendového softwaru, kde frontendový kód jen volá API koncové body vystavené backendem, a backendový kód ve skutečnosti provádí práci. S produkty Firebase je však tradiční backend obejít, což umožňuje klientovi provádět práci. Administrativní přístup k jednotlivým produktům je poskytován konzolí Firebase. [11.]

Ve zkratce, Firebase nabízí plně spravované backendové služby, které komunikují přímo s klientem bez nutnosti psát vlastní backendový kód. Firebase poskytuje také nástroje pro administraci těchto služeb prostřednictvím konzole Firebase.



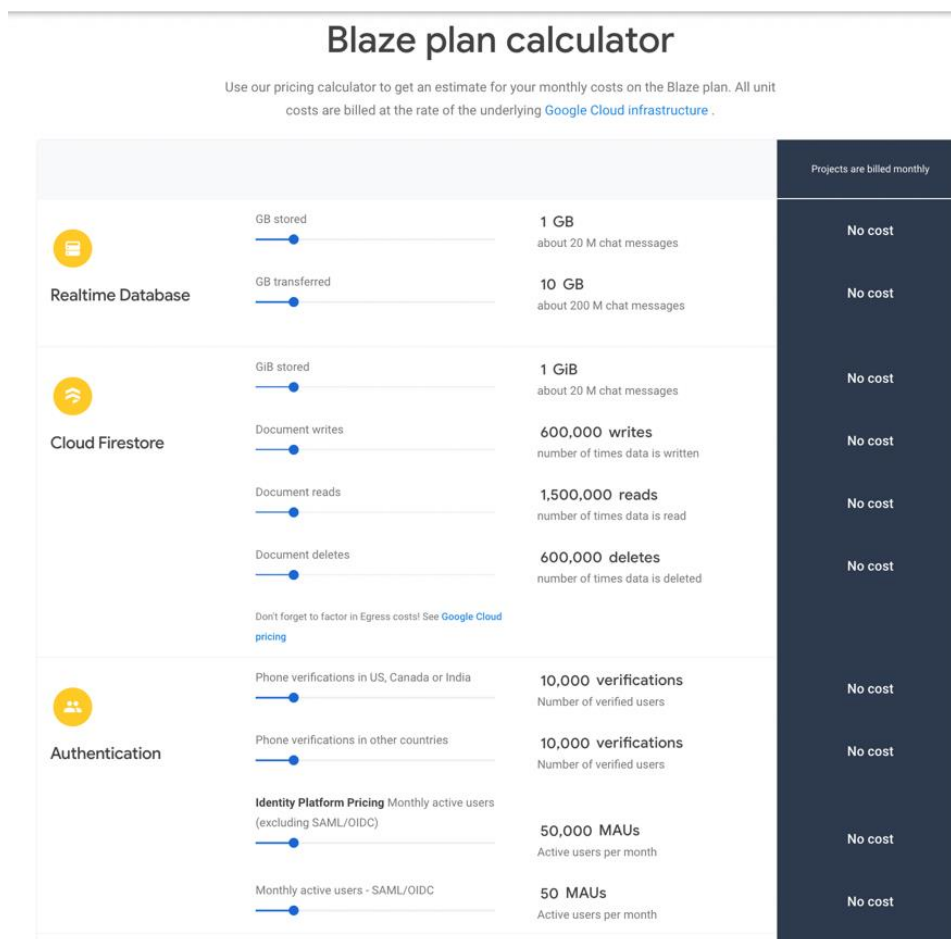
Obrázek 10: Výhoda Firebase cloud hosting  
(Zdroj:11)

## 2.4.2 Firebase ceník

Firestore nabízí dva hlavní typy předplatných, které se liší podle úrovně funkčnosti a limitů využití služeb [12.]:

**Spark:** Základní plán je zdarma a obsahuje základní funkce jako Realtime Database, Authentication a Storage, s omezením počtu uživatelů, datového objemu a přenosu dat.

**Blaze:** Tento plán je placený a umožňuje využívat Firebase služeb s neomezeným datovým objemem a přenosem dat. Cena se liší podle počtu aktivních uživatelů, počtu operací v cloudové Firestore databázi a dalších faktorů. Na oficiálních webových stránkách Firebase je k dispozici kalkulačka, která umožňuje zákazníkům propočítat, kolik bude stát využití této služby.



Obrázek 11: Blaze plan kalkulačka

(Zdroj: 12)

Herní studio se rozhodlo využít pro tento projekt předplatné Blaze.

### 2.4.3 firebase integrace

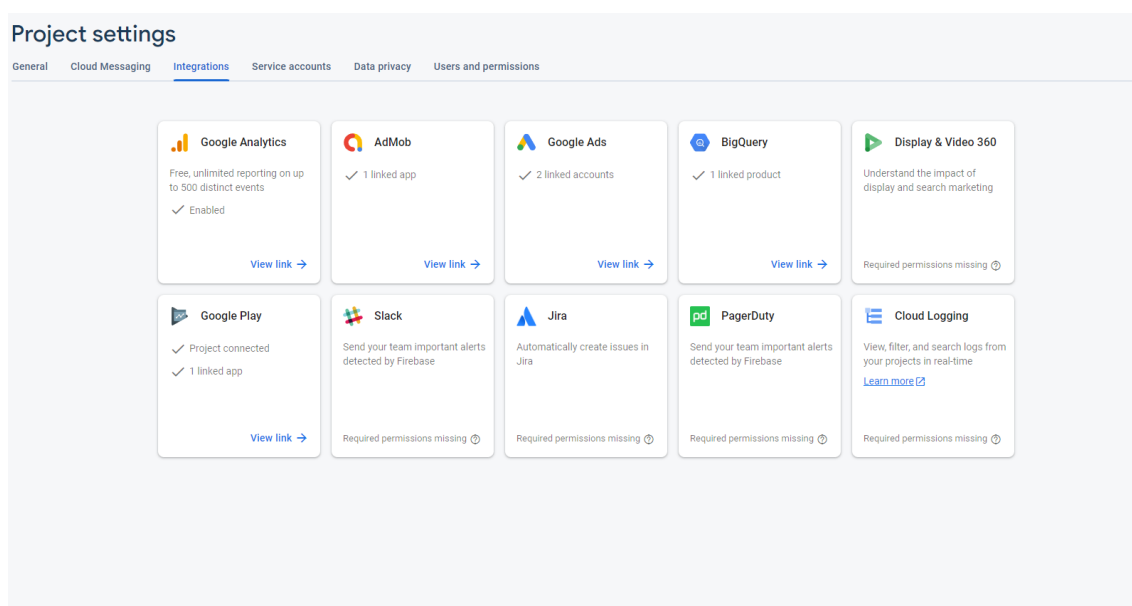
Nástroj Firebase umožňuje propojení s jinými službami a nástroji, díky tomuto propojení se usnadní práce s daty a uživatelskými interakcemi. Například:

**Google Analytics:** Firebase nabízí plnou integraci s Google Analytics, což umožňuje vývojářům sledovat návštěvnost a chování uživatelů ve svých aplikacích.

**AdMob:** AdMob je reklamní platforma společnosti Google, která umožňuje vývojářům monetizovat své aplikace prostřednictvím inzerce. Firebase nabízí plnou integraci s AdMobem.

**Google Ads:** Firebase nabízí integraci s která umožňuje vývojářům vytvářet, spravovat a měřit své reklamy přímo z Firebase konzole.

**Bigquery:** Toto propojení s BigQuery umožňuje vývojářům exportovat data ze svých Firebase projektů do BigQuery a provádět sofistikované analýzy a vizualizace.



**Obrázek 12: Firebase typy integrace**

(Zdroj: 13)

## 2.5 Zdroje dat

V této pod kapitole si popíšeme jednotlivé nástroje, které společnost využívá pro sběr dat z jednotlivých služeb pro marketing, vrcholový management a herní vývojáře.

### 2.5.1 Firebase Události

Jsou události, které slouží pro sledování uživatelského chování v mobilních aplikacích. Firebase události jsou definovány jako akce, které uživatelé vykonávají v aplikaci, například na jaké tlačítka klikají nebo celkově jak s aplikací interagují. Díky těmto

událostem můžeme sledovat klíčové ukazatele výkonosti. Pomocí těchto událostí můžeme získat informace o uživatelském chování v reálném čase.

Firestore události se dělí na dvě základní kategorie [14]:

**Automatické události:** Kde nástroj definuje určité události, které jsou v aplikaci vyvolány automaticky, jako například otevření aplikace, návštěva stránky, dokončení transakce a podobně. Automatické události nelze upravovat ani měnit jejich datové typy.

**Uživatelské události:** Jsou události, které jsou relevantní pro konkrétní aplikaci. Tyto události si vytváří vývojář sám, pomocí nástroje Firestore Console. Uživatelské události mohou být například kliknutí na specifické tlačítko. Vytvoření vlastních událostí v aplikaci umožňuje poskytnout detailnější analýzu uživatelského chování.

### 2.5.2 Ironsource Události

Ironsource události jsou navrženy s cílem pomoci sledovat monetizační fáze a měřit efektivitu nákupu. Tyto události se zaměřují na sledování konkrétních akcí, jako jsou kliknutí na reklamy, instalace aplikací a dokončení nákupu, což pomáhá vývojářům mobilních her optimalizovat své reklamní kampaně a maximalizovat své příjmy z reklam.

### 2.5.3 AdWords Události

AdWords události slouží k sledování akcí uživatelů v rámci reklamních kampaní, které jsou spuštěny pomocí platformy Google AdWords. Tyto události napomáhají měřit účinnost a efektivitu svých reklamních kampaní. Mezi typické AdWords události patří například kliknutí na reklamu nebo její zobrazení. Sledování těchto dat může vést k lepší optimalizaci reklamních strategií pro zlepšení firemních výsledků.

## 2.6 Bigquery

BigQuery je plně spravovaný podnikový datový sklad. tento nástroj spadá pod Google cloud platform, BigQuery umožňuje ukládat spravovat a analyzovat velké množství dat v reálném čase. Díky jeho vysoké škálovatelnosti a výkonnosti dokáže BigQuery zpracovávat obrovské objemy dat s vysokou rychlostí a efektivitou. Obrovské objemy dat se pohybují až do řád petabytů. [15.]

BigQuery umožňuje ukládání dat ve formátu tabulek a podporuje mnoho formátů dat, včetně CSV, JSON, Parquet a Avro. Data lze načíst pomocí souborů, streamingu nebo pomocí přímého přístupu k datovým zdrojům jako například Google Analytics nebo Google Sheets. Přístup do BigQuery je k dispozici přes cloudovou konzoli, klasické webové rozhraní, pomocí příkazového řádku anebo pomocí volání Rest API, kde je potřeba využít jednotlivé knihovny (Python, Java). Pro vizualizaci dat jsou k dispozici nástroje třetích stran (Tableau, Power BI). [15]

## 2.6.1 Bigquery ceník

Cenový model Google BigQuery se skládá ze dvou hlavních složek [16]:

**Výpočetní (analytické) ceny:** jsou náklady na zpracování dotazů, včetně dotazů SQL, uživatelsky definovaných funkcí, skriptů a určitých příkazů jazyka pro manipulaci s daty (DML) a jazyka pro definici dat (DDL). Výpočetní složka nabízí ze dvou modelů výpočtu cen pro spouštění dotazů:

**Cena na vyžádání:** znamená, že uživatel platí jen za počet bajtů zpracovaných každým dotazem. Tento způsob platby je vhodný pro zákazníky, kteří potřebují flexibilitu a škálovatelnost, protože mohou rychle a efektivně využívat výpočetní zdroje v závislosti na potřebách jejich podnikání. Prvních 1 TB zpracovaných dat dotazů za měsíc je zdarma.

*Tabulka 1: Google BigQuery ceník 1*

(Zdroj: 16)

Úkon	Ceny	Podrobnosti
Dotazy (na vyžádání)	5 USD za TB	První 1 TB měsíčně je zdarma.

**Cena za kapacitu:** Tento cenový model spočívá v účtování výpočetní kapacity, kterou použijete při spouštění dotazů, přičemž se měří pomocí virtuálních CPU (slotech) v průběhu času. Tento model využívá edice Bigquery (Standard, Enterprise a Enterprise Plus.) Zákazník může využít automatický škálovač BigQuery nebo si zakoupit závazky slotů, což je vyhrazená kapacita, která je vždy k dispozici pro vaše úlohy, za nižší cenu. Cena může být různá v závislosti na regionu, ve kterém je datový sklad registrován.

Oba způsoby platby mají své výhody a nevýhody a zákazníci si musí vybrat ten, který nejlépe vyhovuje jejich potřebám a způsobu využívání služby BigQuery. Společnost využívá v rámci cenové úspory model ceny na vyžádání.

**Cena úložiště:** je cena za uložení dat, která jsou načtena do BigQuery. Zde uživatel platí za aktivní úložiště a dlouhodobé úložiště. **Aktivní úložiště** zahrnuje jakoukoli tabulku nebo oddíl tabulky, které byly změněny za posledních 90 dní. **Dlouhodobé úložiště** zahrnuje jakoukoli tabulku nebo oddíl tabulky, které nebyly změněny po dobu 90 po sobě jdoucích dnů. Cena úložiště automaticky klesne přibližně o 50 %. Prvních 10 GB úložiště je zdarma.

*Tabulka 2: Google BigQuery ceník 2*

(Zdroj: 16)

Úkon	Ceny	Podrobnosti
Aktivní logické úložiště	0,02 USD za GB	Prvních 10 GB je každý měsíc zdarma.
Dlouhodobé logické úložiště	0,01 USD za GB	Prvních 10 GB je každý měsíc zdarma.
Aktivní fyzické úložiště – Náhled	0,044 USD za GB	Prvních 10 GB je každý měsíc zdarma.
Dlouhodobé fyzické úložiště – Náhled	0,022 \$ za GB	Prvních 10 GB je každý měsíc zdarma.

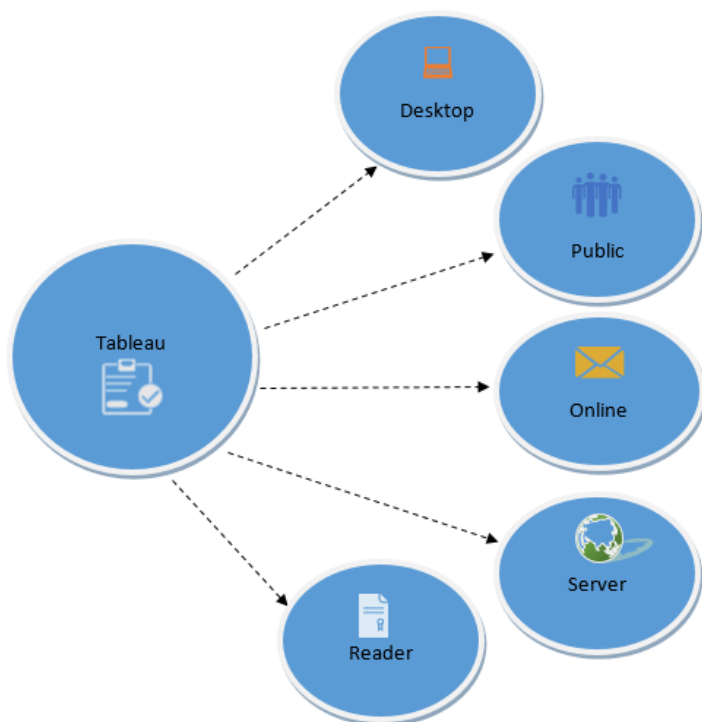
## 2.7 Tableau

Tableau je nástroj který umožňuje snadno a rychle vizualizovat data v BI. Pomáhá při zjednodušování nezpracovaných dat ve velmi snadno srozumitelném formátu. Tento nástroj umožňuje vytvářet data, která mohou pochopit odborníci na jakékoli úrovni organizace. Uživatelé mohou data propojovat a zobrazovat z různých datových zdrojů, kde tyto data lze vizualizovat do vlastních Dashboardů. Hlavní výhodou Tableau je, že nevyžaduje žádné technické nebo jakékoli programovací dovednosti. Program nabízí

vysokou škálu funkcí pro vizualizaci jako jsou například grafy, diagramy, mapy a další.  
[17]

Tableau nabízí sadu produktů které uživatel může využít [17]:

- **Tableau Desktop:** Je sada, kde se tvoří veškerá potřebná vizualizační práce. Pro analýzu dat v reálném čase poskytuje Tableau desktop propojení k datovému skladu a dalším různým typům souborů.
- **Tableau Public:** Tato verze je pro nové uživatele, kteří si chtějí nástroj vyzkoušet. Hlavní nevýhodou je, že soubory nelze uložit lokálně jen na cloudový server, kde si tyto soubory může prohlížet kdokoliv.
- **Server Tableau:** Software se specificky používá ke sdílení sešitů, vizualizací, které jsou vytvořeny v aplikaci Tableau Desktop v celé organizaci.
- **Tableau online:** Jeho funkce jsou podobné jako Tableau Server, ale data jsou uložena na serverech hostovaných v cloudu. Výsledky jsou sdíleny na webovém prostředí.



Obrázek 13: Tableau rozlišení

(Zdroj: 16)

### **3. Vlastní návrh řešení**

Poslední kapitolou této bakalářské práce je vlastní návrh řešení datového skladu pro herní společnost. V této kapitole se budu zabývat kompletním návrhem datového skladu, kde budu vycházet z vypracované analýzy v předchozí kapitole.

#### **3.1 Plán projektu**

Před budováním datového skladu je nutné nejdříve naplánovat celý projekt. Plán projektu definuje cíle projektu, jejich požadavky a následně sekvenci úloh pro vytvoření datového skladu. Požadavky a cíle si definuje vedení společnosti a uživatelé, kteří budou s datovým skladem kooperovat. Na základě těchto požadavků je vytvořen plán pro vytvoření datového skladu, který zajistí, že finální výstupy budou odpovídat cílům projektu.

##### **3.1.1 Cíl projektu**

Cílem projektu je kompletní vytvoření datového skladu na základě požadavků vedení a uživatelů, kteří budou v budoucnu s datovým skladem pracovat. Hlavním cílem je umožnění efektivnějšího zpracování dat pro podporu rozhodování a řízení jednotlivých oddělení. Projekt musí být navržen tak, aby jej bylo možné v budoucnu rozšiřovat a upravovat. Správnost dat v datovém skladu bude ověřována prostřednictvím reportů.

##### **3.1.2 Požadavky projektu**

Požadavky datového skladu jsou klíčovým krokem v procesu jeho vývoje. Požadavky projektu si určují klíčoví zaměstnanci společnosti a koncový uživatelé. Datový sklad bude sbírat data ze dvou zdrojových systémů. Prvním zdrojovým systémem je AdWords, který ukazuje úspěšnost jednotlivých marketingových kampaní. Výstup tohoto zdroje je určen pro marketingové oddělení. Dalším zdrojovým systémem je Firebase, který shromažďuje data o aktivitách jednotlivých uživatelů ve hře. Výstup z Firebase je určen pro vývojáře.

### 3.1.3 sekvence úloh

Při budování datového skladu je důležité vytvořit podrobný plán. Na začátku projektu je nutné propojit Firebase účet s Unity pomocí Firebase SDK. V Unity je třeba definovat Firebase události, které chceme zaznamenávat pro další analýzu. Další částí je vytvoření BigQuery projektu, kde napojíme všechny zdrojové systémy. V BigQuery je potřeba vytvořit procedury pro úpravu dat pomocí ELT metody. V poslední fázi se vytvoří report sloužící jako kontrola funkčnosti datového skladu.

## 3.2 Plán budování datového skladu

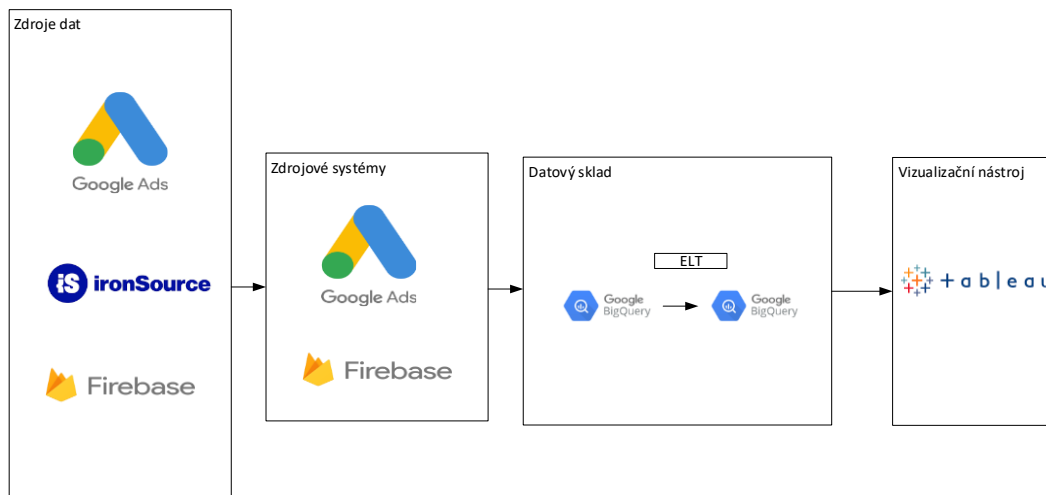
Pro plnění stanovených cílů a požadavků se bude budování datového skladu dělit na několik fází. Každá fáze má svou vlastní časovou náročnost, tudíž je klíčové sledovat úspěšnost jednotlivých fází. Celý proces výstavby datového skladu je rozdělen do následujících etap:

- Návrh datové architektury
- Nakonfigurování nástroje Firebase
- Nakonfigurování nástroje Ironsource
- Vytvoření datového skladu v Google BigQuery
- Propojení jednotlivých zdrojových systémů s Google BigQuery
- Vytvoření ELT procesu v Google BigQuery
- Vytvoření zálohy
- Tvorba reportu

## 3.3 Návrh datové architektury

Následující graf znázorňuje toky dat mezi jednotlivými komponentami datového skladu. Zdrojové systémy se skládají z nástroje Firebase a AdWords a dále se zde vyskytuje nástroj Ironsource, o kterém bylo zmíněno v předchozích kapitolách. Ironsource je považován za zdroj dat, ale je propojen s nástrojem Firebase, takže se nepovažuje za samostatný zdrojový systém. Popis propojení zdrojových systémů s datovým skaldem bude popsán v při vytváření samotného datového skladu. Pro zpracování dat byla použita metoda ELT s využitím procedurálních dotazů v datovém

skladu BigQuery. Posledním komponentem je vizualizační nástroj Tableau, který bude sloužit k reportování dat.



Obrázek 14: Datový architektura

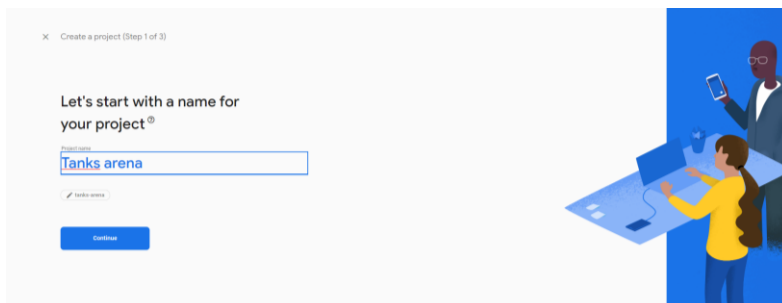
(Zdroj: Vlastní zpracování)

### 3.4 Nakonfigurování nástroje Firebase

V této části se zaměříme na proces vytvoření Firebase projektu a následné propojení s Unity za pomoci Firebase SDK. K tomu využijeme JSON souboru, který slouží k inicializaci a konfiguraci Firebase SDK. Nakonec nadefinujeme některé události, které chceme v naší hře zaznamenávat do datového skladu.

#### 3.4.1 Postup pro propojení Firebase s Unity

**Registrace účtu:** Pro začátek je nutné vytvořit nový projekt v konzoli Firebase. K tomuto úkonu je třeba se přihlásit pomocí Google účtu a poté vyplnit název projektu.

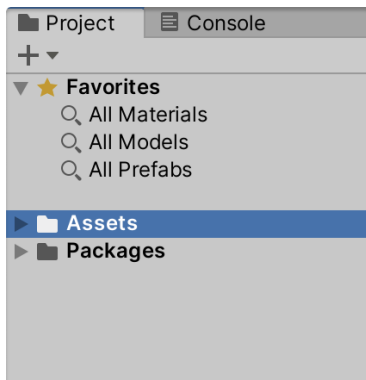


Obrázek 15: Vytvoření projektu ve Firebase

(Zdroj: 13)

**Přidání aplikace:** Dalším krokem je přidání Unity aplikace do našeho projektu ve Firebase. Je třeba vyplnit informace o platformě, pro kterou bude hra vydávána na výběr je buď Android nebo iOS.

**Stažení konfiguračního souboru:** V této fázi je nutné si stáhnout konfigurační soubor JSON pro naši aplikaci. Tento soubor obsahuje klíče a tokeny, které jsou potřeba pro propojení unity s Firebase. Soubor následně nahrajeme do našeho unity projektu.



**Obrázek 16: Přidání konfiguračního souboru**

(Zdroj: 13)

**Integrace s Unity:** Po stažení konfiguračního souboru je třeba stáhnout Firebase SDK a importovat ho do našeho projektu. K tomuto účelu využijeme soubor "FirebaseDatabase.unitypackage", který poté nakonfigurujeme do vytvořeného unity projektu a importujeme potřebné knihovny.

```
using Firebase;
using Firebase.Database;

public class MyScript: MonoBehaviour {
    void Start() {
        // Get the root reference location of the database.
        DatabaseReference reference = FirebaseDatabase.DefaultInstance.RootReference;
    }
}
```

**Obrázek 17: kód k uložení Firebase událostí do databáze**

(Zdroj: 18)

V tento moment je propojení Firebase s Unity dokončeno. Následně si uživatel může definovat vlastní události, které chce zaznamenávat do Firebase databáze.

### 3.4.2 Definování událostí:

Základní analýza událostí firebase byla už nastíněna v rámci analýzy současného stavu. V této podkapitole bude částečně znázorněna a popsáno vytvoření všech vlastních a automatických událostí. V práci nejsou zveřejněny jednotlivé událostní kódy. V obrázku níže si můžeme všimnout všech nadefinovaných událostí. Ke každé události je přiřazeno několik parametrů a jejich datové typy.

```
// Events
private const string EVENT_NEW_USER = "new_user";
private const string EVENT_LOAD_COMPLETED = "load_completed";
private const string EVENT_VIDEO_AD_WATCH = "video_ad_watched";
private const string EVENT_VIDEO_AD_CANCEL = "video_ad_cancel";
private const string EVENT_PURCHASE_COMPLETED = "purchase_completed";
private const string EVENT_CHAMPIONSHIP_RANKUP = "championship_rankup";
private const string EVENT_ITEM_FUSED = "items_fused";
private const string EVENT_ITEMS_SCRAPPED = "items_scrapped";
private const string EVENT_QUICK_FIGHT_STARTED = "quick_fight_started";
private const string EVENT_QUICK_FIGHT_COMPLETED = "quick_fight_completed";
private const string EVENT_CHAMPIONSHIP_FIGHT_STARTED = "championship_fight_started";
private const string EVENT_CHAMPIONSHIP_FIGHT_COMPLETED = "championship_fight_completed";
private const string EVENT_TUTORIAL_STEP = "tutorial_step";
private const string EVENT_TUTORIAL_COMPLETED = "tutorial_completed";
```

Obrázek 18: Všechny nadefinované události

(Zdroj: Vlastní zpracování)

Firestore má již předem definované některé vlastní parametry, které se automaticky odesílají pro každou událost. U těchto parametrů není možné změnit datový typ.

Parameter	Description	Type
event_name	The name of the event.	STRING
event_date	The date on which the event was logged (YYYYMMDD format in the registered timezone of your app).	DATE
event_timestamp	The time (in microseconds, UTC) at which the event was logged on the client.	INTEGER
user_pseudo_id	The pseudonymous id (e.g., app instance ID) for the user.	STRING
category	The device category (mobile, tablet, desktop).	STRING
mobile_model_name	The device brand name.	STRING
mobile_brand_name	The device model name.	STRING
id	The package name or bundle ID of the app.	STRING
version	The app's versionName (Android) or short bundle version.	STRING
platform	The platform on which the app was built.	STRING
source	Name of the network that first acquired the user. This field is not populated in intraday tables.	STRING
name	Name of the marketing campaign that first acquired the user. This field is not populated in intraday tables.	STRING
medium	Name of the medium (paid search, organic search, email, etc.) that first acquired the user. This field is not populated in intraday tables. In our case we have categories: organic (values: none, (direct)), campaigns (value cpc), cross-promo ('app' + 'WindowAd' + 'VideoAd' + 'EventAd'), rest can be some minor media that includes few users.	STRING
country	The country from which events were reported, based on IP address.	STRING

Obrázek 19: automatické události

(Zdroj: Vlastní zpracování)

Následně budou popsány jednotlivé události a k nim přiřazené parametry, ke každému parametru byli nedefinované příslušné datové typy, které jsou znázorněny v příložených obrázcích.

**quick\_fight\_started:** je událost, která se zaznamená, když uživatel klikne na tlačítko quickfight. K této události se zaznamenávají jednotlivé parametry, jako je například ID souboje nebo vybavení hráčů a další.

Tabulka 3: Definování událostí 1

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
fight_id	Unique Fight Identifier	STRING
challenger_gear	Challenger gear	STRING
opponent_gear	Oponent gear	STRING

**quick\_fight\_completed:** je událost, která se zaznamená, když uživatel dokončí svůj souboj v rámci quickfightu. K této události se zaznamenávají jednotlivé parametry například délka souboje nebo výsledek toho souboje a další.

**Tabulka 4: Definování události 2**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
fight_id	Unique Fight Identifier	STRING
challenger_gear	Challenger gear	STRING
oponent_gear	Oponent gear	STRING
result	Result of the match	STRING
duration	Duration of the match in milliseconds	INTEGER

**championship\_fight\_started:** je událost, která se zaznamená, když uživatel klikne na tlačítko championshipfightu. K této události se zaznamenávají jednotlivé parametry například ID souboje nebo vybavení hráčů.

**Tabulka 5: Definování události 3**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
fight_id	Unique Fight Identifier	STRING
challenger_gear	Challenger gear	STRING
oponent_gear	Oponent gear	STRING

**championship\_fight\_completed:** je událost, která se zaznamená, když uživatel dokončí svůj souboj v rámci championshipfightu. K této události se zaznamenávají jednotlivé parametry, například délka souboje nebo výsledek toho souboje a další.

**Tabulka 6: Definování události 4**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
fight_id	Unique Fight Identifier	STRING
challenger_gear	Challenger gear	STRING
oponent_gear	Oponent gear	STRING
result	Result of the match	STRING
duration	Duration of the match in milliseconds	INTEGER
sc_gained	SC gained from championhip fight	INTEGER

**championship\_rankup:** je událost, která se zaznamená, když uživatel zlepší svoji pozici v rámci šampionátu. K této události se zaznamenávají jednotlivé parametry například jestli uživatel postoupil přímo nebo v jakém šampionátu se uživatel nacházel a další.

**Tabulka 7: Definování události 5**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
championship_stage	What phase of the game the player was	INTEGER
championship_prestige	What prestige the player has	INTEGER
position		INTEGER
was_promote	If user promote or not	INTEGER
was_instant	if user promote instant or from timer	INTEGER
sc_gained	SC gained from rank up	INTEGER

**Item\_upgrade:** je událost, která se zaznamená, když uživatel vylepší svůj předmět. K této události se zaznamenávají jednotlivé parametry, například jaký předmět uživatel vylepšil a kolik za něj utratil herní měny.

**Tabulka 8: Definování události 6**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
item	Upgraded item	STRING
sc_spent	Amount of coins player spent	INTEGER

**items\_scrapped:** je událost, která se zaznamená, když uživatel prodá svůj předmět. K této události se zaznamenávají jednotlivé parametry, například kolik předmětu v rámci jedné akce prodal a kolik za něj dostal herní měny.

**Tabulka 9: Definování události 7**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
count_items_scrapped	count of items scrapped	INTEGER
sc_gained	Original target of fusing	INTEGER

**tutorial\_step:** je událost, která se zaznamená, jak uživatel postupuje v rámci tutoriálu. K této události se zaznamenávají jednotlivé parametry, například jednotlivé kroky v rámci dané části tutoriálu. Parametr preset, pak znázorňuje název jednotlivých tutoriálu ve hře.

**Tabulka 10: Definování události 8**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
preset	Id of the tutorial preset	STRING
step_index	Index of step	INTEGER

**tutorial\_completed:** je událost, která se zaznamená, když uživatel dokončí tutoriál. K této události se zaznamenávají jednotlivé parametry, například jak dlouho uživateli trvalo tutoriál projít a další.

**Tabulka 11: Definování události 9**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
preset	Id of the tutorial preset	STRING
step_index	Index of step	INTEGER
duration	Duration of the match in milliseconds	INTEGER

**New\_user:** Je událost, která se vyvolává tehdy, když uživatel přijde do hry poprvé.

K této události se zaznamenávají jen automatické parametry.

**load\_completed:** Je událost, která se vyvolává tehdy, když uživatel spustil aplikaci a je v procesu načítající se hry. Tato událost zahrnuje parametr, který zaznamenává délku času potřebného k načtení hry.

**Tabulka 12: Definování události 10**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
loading_time	Game loading time in milliseconds	INTEGER

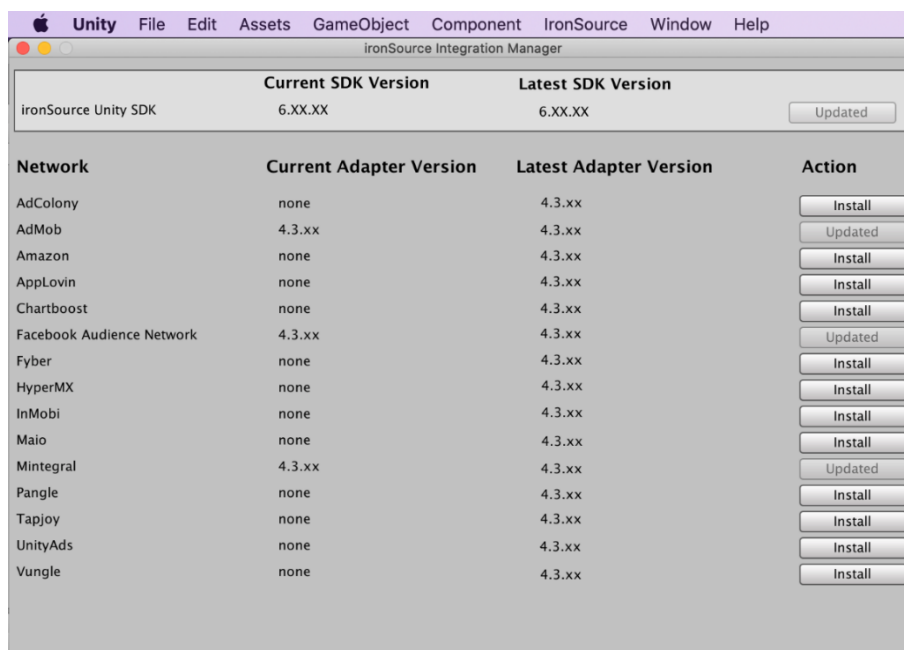
## 3.5 Nakonfigurování nástroje Ironsource

Ironsource slouží jako platforma pro monetizaci mobilních her nebo aplikací. Díky tomuto nástroji může společnost zvýšit svůj zisk pomocí reklam a dalších zdrojů příjmů. V této kapitole bude popsáno propojení Ironsource mezi unity pomocí SDK. Následně je potřeba propojit nástroj Ironsource s nástrojem Firebase. V poslední části je potřeba nadefinovat monetizační události pro tento projekt.

### 3.5.1 Napojení Ironsource na Unity projekt

**Stáhnutí Unity plugin:** V prvním kroku je nutné si stáhnout unity plugin SDK. Tento soubor je třeba importovat do našeho unity projektu a nainstalovat potřebné moduly a knihovny.

**Správa adaptérů:** V dalším kroku je potřebná správa adaptérů v uživatelském rozhraní. Správa adaptérů v Ironsource se týká správy a konfigurace jednotlivých reklamních sítí nebo poskytovatelů reklamních platforem, kteří jsou připojeni k Ironsource. Adaptéry jsou potřebné k tomu, aby se reklamy z různých sítí mohly zobrazovat v mobilní aplikaci. Při konfiguraci adaptérů je důležité si uvědomit, že každá reklamní síť má svá pravidla a omezení.



Obrázek 20: Správa adaptérů

(Zdroj:20)

V tento moment jsou Ironsource a Unity úspěšně propojeny a nyní je třeba vytvořit skripty pro jednotlivé události, které chceme zaznamenávat v naší Firebase databázi.

**Vytvoření skriptů:** V poslední části našeho napojení je potřeba vytvoření událostí, které budou uloženy do databáze Firebase. Tento kód implementuje událost sledování úspěšného zobrazení reklamy pomocí IronSource a následně je uložen do databáze Firebase.

```
private void ImpressionSuccessEvent(IronSourceImpressionData impressionData) {
    if (impressionData != null) {
        Firebase.Analytics.Parameter[] AdParameters = {
            new Firebase.Analytics.Parameter("ad_platform", "ironSource"),
            new Firebase.Analytics.Parameter("ad_source", impressionData.adNetwork),
            new Firebase.Analytics.Parameter("ad_unit_name", impressionData.getInstanceName),
            new Firebase.Analytics.Parameter("ad_format", impressionData.getAdUnit),
            new Firebase.Analytics.Parameter("currency", "USD"),
            new Firebase.Analytics.Parameter("value", impressionData.getRevenue)
        };
        Firebase.Analytics.FirebaseAnalytics.LogEvent("ad_impression", AdParameters);
    }
}
```

Obrázek 21:Napojení Ironsource události do Firebase databáze

(Zdroj:19)

### 3.5.2 Definování událostí Ironsource:

IronSource umožňuje vývojářům definovat vlastní události, které se vyskytují v aplikaci během zobrazení reklam. Definování těchto událostí umožňuje vývojářům sledovat, jak uživatelé interagují s monetizačními prvky dané hry. V práci nejsou zveřejněny jednotlivé událostní kódy. V předchozí kapitole byli ukázány veškeré nadefinované události, které se ukládají do databáze Firebase z těchto událostí jsou právě tři, které spadají do skupiny Ironsource událostí. Ke každé události je přiřazeno několik parametrů a jejich datové typy.

**video\_ad\_watched:** je událost, která se zaznamená, když uživatel klikne na tlačítko reklamy. K této události se zaznamenává parametr context, který slouží jak popis odměny, za kterou byla reklama spuštěna.

Tabulka 13: Definování událostí 11

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
context	Video context like timeskip, lucky wheel etc...	STRING

**video\_ad\_cancel:** je událost, která se zaznamená, když uživatel zruší spuštěnou reklamu. K této události se zaznamenává parametr context, který slouží jak popis odměny, za kterou byla reklama spuštěna.

**Tabulka 14: Definování událostí 12**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
context	Video context like timeskip, lucky wheel etc...	STRING

**Purchase\_completed:** je událost, která se zaznamená tehdy, když si uživatel něco ve hře zakoupil. K této události se zaznamenává parametr context který slouží jak popis odměny, za kterou byl nákup proveden a parametr price který ukazuje cenu za kterou zákazník zaplatil.

**Tabulka 15: Definování událostí 13**

(Zdroj: Vlastní zpracování)

Parameter	Description	Type
context	Purchase context box,gems etc...	STRING
price	price in USD	INTEGER

## 3.6 Analýza zdrojových systémů

Základní analýza zdrojových dat byl provedena už v analytické části. Pro fungování datové skladu jsou využívány dva zdrojové systémy Firebase a Google AdWords.

### 3.6.1 Google AdWords

V rámci tvorby online reklamy je pro marketingové oddělení klíčovým prvkem využití AdWords. Největší část reklamního rozpočtu tvoří PPC kampaně, které jsou spuštěny na platformě Google. Pro každou zemi je využívána specifická PPC kampaň. Data z těchto kampaní se aktualizují na denní bázi a výkon kampaní je určován pomocí různých parametrů, které se ukládají přímo do nástroje. Sledují se konkrétní ukazatele, jako je například cílová země, počet zobrazení reklamy, interakce s reklamou, náklady na

konkrétní kampaň a měna, ve které jsou náklady vyjádřeny. Propojení Google AdWords s datovým skladem bude popsáno při tvorbě tohoto skladu.

### **3.6.2 Firebase**

V naší aplikaci je nástroj Firebase používán k sledování uživatelských akcí, přičemž tyto zdroje dat jsou považovány za důležité oddělením specializujícím se na vývoj her. Kombinace zdrojových dat z nástroje Ironsource a Firebase je realizována pomocí SDK klíčů a tokenu v Unity. Tyto data jsou následně ukládána do Firebase databáze. Z této databáze Firebase vytváří již předdefinované reporty přímo v nástroji, avšak tyto reporty jsou nedostačující pro rozhodování vývojářského oddělení, proto tato databáze bude následně propojena s BigQuery pomocí Firebase rozšíření.

## **3.7 Popis datové struktury**

Jak již bylo zmíněno, firma vlastní i další datové sklady, s nimiž pracují datový analytici. Vzhledem k tomu je nezbytné, aby byl návrh datové struktury vytvořen přesně tak, jak je to běžné v ostatních projektech.

Datová struktura v projektech je navrhována tak, že se ze zdrojového systému Firebase nahrají veškerá data do jedné velké tabulky v BigQuery z nichž se pomocí ELT procedur exportují ty události a její parametry, které chce firma k danému projektu sledovat. Pro tyto události jsou vytvořeny dvě cílové reportující tabulky, které jsou propojeny skrze identifikátory uživatelů (userID). Pro marketingová data se ze zdrojového systému nahrávají veškeré informace do desítek tabulek, procedury poté tyto tabulky sloučí do jedné reportující tabulky, která poskytuje celkový přehled a analýzu dat pro potřeby společnosti.

### **3.7.1 Vlastní návrh datové struktury**

Společnost využívá tedy tři tabulky, ze kterých vytváří následné reporty. Jelikož tyto tabulky obsahují desítky parametrů a zároveň jejich datová velikost může být v budoucnu mnohonásobně větší, tak by měla společnost zvážit změnu datové struktury. Doporučil bych firmě rozdělit datovou strukturu na tabulky faktů a dimenzí podle metody dimenzionálního modelování. Každá událost by charakterizovala jednu dimenzi

s popisnými informacemi. Tabulky faktů by zaznamenávali jejich číselné metriky, tyto tabulky by byly propojeny skrze identifikátory uživatelů (userID). Další dimenzí by mohla být časová dimenze, která by se propojila s tabulkou faktů přes vlastní indikátor. Díky tomuto procesu by firma mohla ušetřit na nákladech příliš velkého dotazování, anebo snížit složitost jednotlivých dotazů.

### **3.8 Vytvoření datového skladu**

Prvním krokem k vytvoření datového skladu je zaregistrování se do nástroje Google cloud platform. Tato platforma je cloudová služba, která spojuje všechny dostupné cloudové služby od Googlu. Tento nástroj nabízí širokou škálu služeb a produktů pro řešení firemních potřeb. Nabízí široký výběr uložišť pro ukládání dat, širokou škálu relačních a nerelačních databází jako je například Firestore, nástroje pro práci s umělou inteligencí nebo také poskytuje různé zabezpečení a mnoho dalších nástrojů. Google big query patří mezi nástroje z kategorie Big Data. Google cloud platform je také známí svoji vysokou škálovatelností a flexibilitou.

V následujícím kroku je nutné využít konzoli Google cloud a v ní vytvořit nový projekt. Tento krok vyžaduje zadání názvu projektu a zvolení organizace. Organizace projektu v BigQuery slouží k logickému oddělení datových zdrojů, uživatelů, rolí a přístupových práv. Po vytvoření projektu se zobrazí úvodní strana celého projektu, kde jsou popsány jednotlivé informace o projektu, datové zdroje, vyúčtování a jeho fakturační údaje. Jednotlivé tutoriály a mnoho dalšího.

V tento moment je potřeba se přesunout do nástroje BigQuery, kde můžeme začít definovat jednotlivé komponenty pro tvorbu datového skladu. Automaticky se přiřadil projekt, který funguje jako logickou jednotkou s nejvyšší entitou, projekt sdružuje datové sety a určuje jednotlivé výpočetní zdroje. Každý projekt má své jedinečné ID a název. Následující krok je vytvoření datového setu, kde je opět potřeba zadání názvu a nastavení lokality, kde budou data skladované. Ve vyskakovacím okně máme již předem nadefinovaný region US multiregion, avšak tento region je pro náš projekt nevyhovující. Jelikož naše data jsou skladované na centrálním serveru v Belgii. Pro co nejmenší latenci byl zvolen region europe-west1. Datový set představuje v Bigquery alternativu ke klasické relační databázi. V následujícím kroku můžeme nastavit automatické mazání

tabulek po určité době, nicméně kvůli uchování všech historických dat plánujeme tuto funkci neaktivovat. V pokročilých možnostech nastavení datového setu necháme předdefinované nastavení, které se týká šifrování, třídění a filtrování názvů. Po vytvoření datového setu bude nutné propojit zdrojové systémy s datovým skladem.

The screenshot shows the 'Create dataset' configuration page. It includes the following fields and options:

- Project ID:** tanks-arena (with a 'CHANGE' link)
- Dataset ID \*:** Events (with a note: 'Letters, numbers, and underscores allowed')
- Location type:** Region (selected with a radio button). Description: 'Specifying a region provides dataset colocation with other GCP services'. Alternative: Multi-region (radio button). Description: 'Letting BigQuery select a region within a group of regions provides higher quota limits'.
- Region \*:** europe-west1 (Belgium) (dropdown menu)
- Default table expiration:** 'Enable table expiration' checkbox is unchecked. Input field: 'Default maximum table age' (Days).
- Advanced options:** (indicated by an upward arrow)
  - Encryption:** 'Google-managed encryption key' (selected with a radio button, 'No configuration required'). Alternative: 'Customer-managed encryption key (CMEK)' (radio button, 'Manage via Google Cloud Key Management Service').
  - Case Insensitive:** 'Enable case insensitive table names' checkbox is unchecked.
  - Default Collation:** 'Enable default collation' checkbox is unchecked. Input field: 'Default Collation' (dropdown menu).
- Buttons:** 'CREATE DATASET' (blue) and 'CANCEL' (grey).

Obrázek 22: Vytvoření datového setu

(Zdroj: 21)

### 3.8.1 Propojení zdrojových systémů s datovým skladem

V předchozí podkapitole byl vytvořen datový sklad a jeho komponenty. V tento moment je nutné připojit zdrojové systémy Firebase a Google AdWords. Oba tyto nástroje spadají pod Google Cloud platform, tudíž následné propojení s datovým skladem je velmi prosté.

#### 3.8.1.1 Google Adwords

Pro integrování zdroje dat z Google AdWords do BigQuery je nezbytné provést následující kroky. Nejprve je třeba vstoupit do transformačního editoru dat a vytvořit novou transformaci, která bude sloužit jako zdroj dat pro BigQuery. V tomto procesu je

vybrán zdroj dat Google AdWords. Následujícím krokem je zadání názvu této transformace, který bude použit k její identifikaci. Poté si nadefinujeme rozvrh přenosu dat. Jelikož chceme, aby data byla aktuální zvolíme frekvenci přenosu dat na každý den. V datové transformaci si můžeme nastavit, jestli chceme data začít přenášet od určitého data nebo ihned. V poslední části toho nastavení je nutné vybrat patřičnou datovou sadu, do které chceme následující data posílat a vyplnit uživatelské ID z nástroje Google Ads.

The screenshot shows the 'Up to BigQuery transfer details' configuration page. It is divided into several sections: 'Source type' where 'Google Ads' is selected; 'Transfer config name' with a display name 'AdWords\_Transfer\_TArena\_converted'; 'Schedule options' where the repeat frequency is set to 'Days' at '05:30 UTC', with 'Start at set time' selected; 'Start date and run time' set to '2/24/23, 6:30 AM CET'; and 'Destination settings' where the dataset is 'AdWords'. A notification at the bottom of the schedule section states: 'Scheduled to run every day at 05:30 UTC starting on February 24, 2023 at 6:30:00 AM UTC+1.'

**Obrázek 23: Napojení Google AdWords do Google Bigquery**

(Zdroj: 21)

### 3.8.1.2 Firebase

Pro úspěšnou integraci dat ze zdrojového systému Firebase je nutné provést následující kroky. Prvním krokem je přihlášení do konzole Firebase a následný výběr projektu, který bude integrován. V dalším kroku je nutné přejít do záložky integrace v nastavení projektu. Integrace Firebase disponuje mnoha různými nástroji, které lze s ní propojit, jako například AdMob, Slack nebo Google Analytics. V tomto případě je vybrán BigQuery pro integraci dat.

Následně se otevře vyskakovací okno, kde je potřeba nakonfigurovat nastavení pro import dat. V dalším kroku je třeba zvolit, jaké služby budou reportovány do datového skladu a příslušný region. V posledním kroku se opět definuje časová frekvence posílání dat do datového skladu. V tento moment se automaticky vytvoří nový datový set s názvem analytics\_345895714, kde se nahrávají uživatelská data z naší mobilní aplikace.

### **3.8.2 Definování tabulek**

V předchozí podkapitole byla popsána integrace zdrojových systémů do datového skladu. V rámci integrace s Google AdWords byl vytvořen rozsáhlý datový soubor obsahující mnoho různých metrik. Nicméně pro účely tohoto projektu nejsou všechny tyto metriky relevantní a nebudou využity. V případě Firebase byl vytvořen datový soubor obsahující jedinou tabulku, která zahrnuje veškeré události z Firebase, včetně automatických událostí. Pro tento projekt však nebudou využity všechny automatické události, proto je nutné nadefinovat si vlastní tabulky, kterou budou plněny pomocí ELT procesu, který bude popsán v následující podkapitole.

Pro vytvoření tabulky je potřeba definovat název projektu a datový set, ke kterému bude datová tabulka přiřazena. Následně je potřeba zvolit název tabulky a navrhnout její atributy, které budou zobrazeny jako sloupce. K těmto atributům je potřeba přiřadit jejich datové typy, popřípadě širší specifikace jako je například délka textového pole nebo popis. Plnění tabulek je možné vytvářet dvěma způsoby: vytvářecím dotazem anebo manuálním definováním jednotlivých atributů. Atributy znázorňují parametry událostí, které byly nastíněny v předchozí kapitole. Jakmile budou atributy vnořeny do tabulky, tak už není jakýmkoliv způsobem tyto atributy modifikovat. V projektu se budou definovat pouze tři tabulky. V obrázku, níž je vidět příklad definování tabulky Gameplay-events.

Filter Enter property name or value ?

<input type="checkbox"/> Field name	Type	Mode	Collation	Default Value <span style="font-size: small;">?</span>	Policy Tags <span style="font-size: small;">?</span>	Description
<input type="checkbox"/> event_name	STRING	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> event_date	DATE	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> event_timestamp	INTEGER	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> user_pseudo_id	STRING	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> category	STRING	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> mobile_model_name	STRING	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> mobile_brand_name	STRING	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> operating_system_version	STRING	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> id	STRING	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>
<input type="checkbox"/> version	STRING	NULLABLE		<input type="text" value="Default Value"/>		<input type="text" value="Description"/>

Rows per page: 10 | 1 - 10 of 64 | < >

#### New fields

Edit as text

Field name \*

Type \*

Mode

Description

sc\_gained

INTEGER

NULLABLE

SAVE
CLOSE

Obrázek 24: Definování tabulek

(Zdroj: 21)

### 3.8.3 ELT

V této podkapitole bude popsán proces zpracování dat v datovém skladu pomocí metody ELT, která zahrnuje postupný přenos a přípravu dat pro analýzu. V předchozí podkapitole jsme definovali jednotlivé tabulky a jejich datové typy. V současné době je nutné provést pročištění zdrojových tabulek a nahrání dat do nově vytvořených tabulek. K tomuto účelu využíváme funkci "plánované dotazy" (scheduled queries). Tyto dotazy jsou procedury, které umožňují automatické spouštění dotazů v pravidelných intervalech. Kritéria pro naplánování dotazů mohou zahrnovat datum a čas spuštění, frekvenci spouštění nebo výsledky předchozích dotazů. Každé tabulce náleží jedna procedura a všechny procedury jsou aktualizovány na denní bázi. Všechny plánované dotazy se spouští v jiný čas, kvůli případnému přerušení.

#### Gameplay\_events\_table\_refresh:

Tato procedura slouží k naplnění dat do tabulky **Events.Gameplay\_events\_par\_c** ze zdrojové tabulky **tanksarena-83c53.analytics\_345895714.events\_\***, která obsahuje záznamy o událostech, které byly definovány v předchozí kapitole. Procedura nejprve

definuje datum, ze kterého budou data načítána. Následně pomocí SELECT dotazu se vybere řada sloupců, které se následně uloží do tabulky. Použitím WHERE klauzule se filtrují pouze záznamy, které splňují určité podmínky, například název události.

```

DECLARE begin_date STRING
DEFAULT (FORMAT_DATE("%Y%m%d", DATE_ADD((SELECT MAX(event_date) FROM Events.Gameplay_events_par_c TABLE), INTERVAL 1 DAY)));

INSERT INTO
  Events.Gameplay_events_par_c

WITH base AS (
  SELECT
    event_name,
    PARSE_DATE("%Y%m%d", event_date) AS event_date,
    event_timestamp,
    user_pseudo_id,
    device.category,
    device.mobile_model_name,
    device.mobile_brand_name,
    device.operating_system_version,
    app_info.id,
    app_info.version,
    app_info.install_source,
    platform,
    device.language,
    traffic_source.source,
    traffic_source.name,
    traffic_source.medium,
    geo.country,
    (SELECT value.int_value FROM UNNEST(event_params) WHERE key = 'ga_session_number') AS session_number,
    (SELECT value.int_value FROM UNNEST(event_params) WHERE key = 'ga_session_id') AS session_id,
    (SELECT value.string_value FROM UNNEST(user_properties) WHERE key = 'inv_gems') AS inv_gems,
    (SELECT value.string_value FROM UNNEST(user_properties) WHERE key = 'inv_coins') AS inv_coins,
    (SELECT value.string_value FROM UNNEST(user_properties) WHERE key = 'rew_ads_watched') AS rew_ads_watched,
    (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'fight_id') AS fight_id,
    (SELECT value.string_value FROM UNNEST(user_properties) WHERE key = 'championship_stage') AS championship_stage,
    (SELECT value.string_value FROM UNNEST(user_properties) WHERE key = 'championship_prestige') AS championship_prestige,
    (SELECT value.int_value FROM UNNEST(event_params) WHERE key = 'position') AS position,
    (SELECT value.int_value FROM UNNEST(event_params) WHERE key = 'was_instant') AS was_instant,
    (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'challenger_gear') AS challenger_gear,

```

```

        (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'oppo-
nent_gear') AS opponent_gear,
        (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 're-
sult') AS result,
        CAST((SELECT value.double_value FROM UNNEST(event_params) WHERE key = 'du-
ration') AS INT64) AS duration,
        (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'origi-
nal_item') AS original_item,
        (SELECT value.string_value FROM UNNEST(event_pa-
rams) WHERE key = 'new_item') AS new_item,
        (SELECT value.string_value FROM UNNEST(event_pa-
rams) WHERE key = 'items_used') AS items_used,
        (SELECT value.int_value FROM UNNEST(event_pa-
rams) WHERE key = 'sc_spent') AS sc_spent,
        (SELECT value.string_value FROM UNNEST(event_params) WHERE key = 'scrap-
ped_items') AS scrapped_items,
        (SELECT value.int_value FROM UNNEST(event_params) WHERE key = 'sc_ga-
ined') AS sc_gained,
        (SELECT value.int_value FROM UNNEST(event_params) WHERE key = 'loadingTi-
me') AS loadingTime,
        (SELECT value.string_value FROM UNNEST(user_proper-
ties) WHERE key = CONCAT('firebase_exp_', (SELECT MAX(SPLIT(key, '_')[OFF-
SET(2)]) FROM UNNEST(user_properties) WHERE key LIKE 'fire-
base_exp_%')) AS experiment_group,
FROM
    `tanksarena-83c53.analytics_345895714.events_*`
WHERE
    _TABLE_SUFFIX BETWEEN begin_date AND FOR-
MAT_DATE('%Y%m%d', DATE_SUB(CURRENT_DATE(), INTERVAL 1 DAY))
    AND event_name IN ('quick_fight_started', 'quick_fight_completed', 'cham-
pionship_fight_started', 'championship_fight_completed', 'championship_ran-
kup', 'items_fused', 'items_scrapped', 'load_completed')
    AND app_info.id NOT LIKE '%mod%'
)
-- Odstranění duplicitních řádků --
SELECT
    * EXCEPT(row_number)
FROM (
    SELECT
        *,
        ROW_NUMBER() OVER (PARTITION BY event_time-
stamp, event_name, user_pseudo_id) row_number
    FROM
        base
)
WHERE
    row_number = 1

```

**Automatic\_events\_table\_refresh:** Tato procedura dělá úplně stejnou činnost jako ta předcházející, jediný rozdíl je v tom, že tato procedura plní tabulku s automatickými událostmi ze zdrojové tabulky firebase do **Events.Automatic\_events\_par\_c**.

**AdWords\_spend\_transfer:** Tato procedura spojuje data z různých tabulek v marketingových zdrojích a vypočítává klíčové metriky, jako jsou náklady, konverze, zobrazení a interakce pro různé kampaně. Výsledné informace jsou uloženy v nové tabulce s názvem **AdWords\_Spend**. Procedura používá několik SQL dotazů k agregaci dat z tabulek **tanksarena-83c53.AdWords.pCampaignBasicStats\_9812237111** a **tanksarena-83c53.AdWords.p\_Campaign\_9812237111**.

```
SELECT
  a.date,
  a.campaign_id,
  b.campaign_name,
  a.expenses,
  a.conversions,
  a.impressions,
  a.interactions
FROM(
  SELECT
    CampaignId AS campaign_id,
    SUM(Cost) / 1000000 AS expenses,
    SUM(Conversions) AS conversions,
    SUM(Impressions) AS impressions,
    SUM(Interactions) AS interactions,
    Date AS date
  FROM
    `tanksarena-83c53.AdWords.p_CampaignBasicStats_9812237111`
  GROUP BY
    campaign_id,
    date
) a
RIGHT JOIN(
  SELECT
    campaignId AS campaign_id,
    campaignName AS campaign_name
  FROM
    `tanksarena-83c53.AdWords.p_Campaign_9812237111`
  WHERE
    CampaignName LIKE 'TA -%'
  GROUP BY
    campaign_id,
    campaign_name
) b
ON a.campaign_id = b.campaign_id
```

### 3.8.4 Záloha

V tento moment je datový sklad plně funkční, v poslední podkapitole této části je zálohování dat. Zálohování dat je zásadní pro udržení datové integrity a pro zajištění obnovy dat v případě výpadku systému nebo poškození dat. Pokud by došlo ke ztrátě dat

a nebyly by dostupné zálohy, mohlo by to mít pro společnost vážné důsledky při rozhodování.

Pro zajištění zálohování dat v datovém skladu byla použita metoda plánovaných dotazů. Konkrétně byly nastaveny zálohy pro tři hlavní tabulky **AdWords\_Spend**, **Events.Gameplay\_events\_par\_c**, **Events.Automatic\_events\_par\_c**. Tyto zálohy se pravidelně aktualizují každé dva týdny do samostatných tabulek. Tím je zajištěna ochrana důležitých dat a minimalizuje se riziko ztráty nebo poškození dat. Níže je popsána záloha k tabulce **Events.Gameplay\_events\_par\_c**.

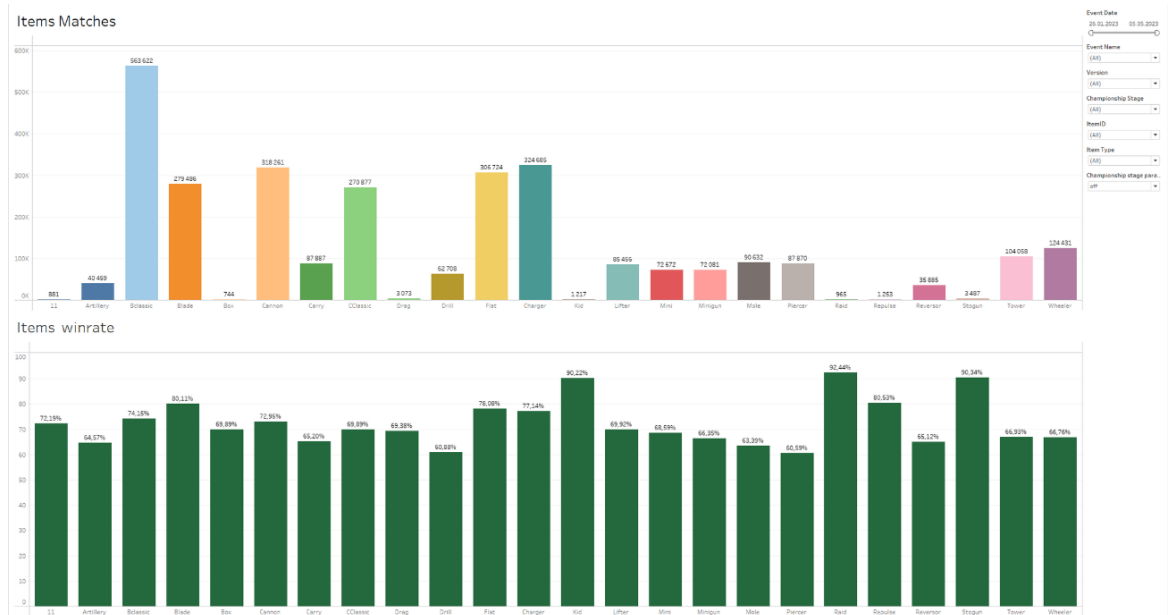
```
DECLARE snapshot_name STRING;
DECLARE expiration TIMESTAMP;
DECLARE query STRING;
SET expiration = DATE_ADD(@run_time, INTERVAL 400 DAY);
SET snapshot_name = CONCAT("backup_snapshots.Gameplay_events_par_c_",
    FORMAT_DATETIME("%Y%m%d", @run_date));
SET query = CONCAT("CREATE SNAPSHOT TABLE ", snapshot_name,
    " CLONE `tanksarena-83c53.Events.Gameplay_events_par_c` OPTIONS(expira-
    tion_timestamp=TIMESTAMP \"\",
    expiration, \"\");");
EXECUTE IMMEDIATE query;
```

### 3.9 Tvorba reportu

Poslední kapitola návrhové části se zabývá tvorbou finálního reportu, který slouží jako důkaz funkčnosti a správnosti celého systému. Společnost používá pro reportování nástroj Tableau, který umožňuje vytvářet vizuálně atraktivní a srozumitelné reporty. K demonstraci funkčnosti byl vytvořen report nad herními daty, které byly nasbírány během beta testování. Tento report poskytuje užitečné informace o chování hráčů a umožňuje nám lépe porozumět chování uživatelů naší hry.

Prvním krokem při tvorbě v nástroji Tableau je nejdříve nutné provést autentizaci do Google Cloud Platform účtu. Po úspěšné autentizaci vybereme zdrojový systém BigQuery a zvolíme datový sklad, který odpovídá projektu, s nímž pracujeme. V posledním kroku je nutné vytvořit SQL dotaz, který nám nahraje potřebná data do vizualizačního prostředí. V samotném vizualizačním prostředí poté můžeme upravovat a zpracovávat jednotlivá data a využít různé funkce a kalkulace pro vytvoření složitějších vizualizací.

Vizualizační report je prezentován pomocí dashboardu, který se skládá ze dvou sloupcových grafů. První graf ukazuje, jaké předměty jsou v rámci soubojů nejvíce využívány. Jednotlivé sloupce jsou barevně odděleny. Druhý graf ukazuje u jednotlivých předmětů výtěžností poměr. V tomto dashboardu lze využívat filtry pro podrobnější analýzu. Díky tomuto dashboardu můžou herní designeři upravovat vlastnosti každého předmětu a tím ovlivnit jeho využití a úspěšnost, aby hra byla co nejvíce zábavná a hratelná.



Obrázek 25: Finální Dashboard

(Zdroj: Vlastní zpracování)

### 3.10 Zhodnocení projektu

Cílem projektu bylo dosažení návrhu a implementace datového skladu pro data z nově vytvořené mobilní hry, přičemž tento cíl byl úspěšně dosažen. Počátečním krokem projektu bylo vytvoření podrobného plánu a určení požadavků od společnosti, kde byl datový sklad vyvíjen. Následně byly zprovozněny nástroje Firebase a IronSource, které byly sloučeny do jednoho zdrojového systému, po čemž byly definovány události pro sledování dat. Pro vytvoření datového skladu byl využit nástroj Google BigQuery, kde jsme propojili jednotlivé zdrojové systémy a tím umožnili nahrávání dat do datového skladu. Protože tato data byla v nedostačující formě, bylo nutné je pomocí metody ELT

upravit a nahrát do vlastních tabulek. V rámci bezpečnosti byla vytvořena záloha. Na závěr projektu byl vytvořen report, který demonstroval funkčnost celého datového skladu.

Jelikož se hra momentálně nachází ve fázi beta testování a je spuštěna pouze v některých vybraných zemích, počet nahraných řádků se pohyboval v milionech za měsíc. Datový sklad byl koncipován tak, aby se mohl v budoucnu rozšiřovat a modifikovat. V případě, že se hra dostane do fáze plného vydání, a bude spuštěna pro všechny země, tak se počet řádků může výrazně zvýšit, proto by měla společnost dle mého uvážení zapřemýšlet nad proškolením zaměstnanců a koncových uživatelů, kteří s datovým skladem pracují. Tím by mohla být zlepšena logická struktura datového skladu, což by mělo za následek úsporu finančních prostředků a času, které by jinak byly vynaloženy na zbytečně velké dotazování v obrovských tabulkách.

## Závěr

Cílem této bakalářské práce byl návrh a implementace datového skladu pro nově vytvořenou mobilní hru. Vzhledem k tomu, že společnost již pracuje s jinými datovými sklady, bylo nezbytné přizpůsobit tento projekt tak, aby se zachovala logická struktura a aby koncoví uživatelé nemuseli být znovu proškoleni. Datový sklad byl tvořen pro oddělení herních vývojářů, marketingu a vrcholového managementu. K tomuto účelu byly vybrány různé nástroje, jako jsou Google AdWords pro marketingové účely, Ironsource pro mediace a Firebase pro data vytvořená ve hře. Samotný datový sklad byl vytvořen v Bigquery, kde se data upravili pomocí ELT procesu. Aby se zajistila správnost dat, byl vytvořen vizualizační report v nástroji Tableau.

V prvním části bakalářské práce jsem popsal teoretická východiska celé práce, tak aby čtenářovi byli jasné veškeré výrazy a pojmy používané v ostatních částí. Teoreticky byli popsány základy datového skladu, datové tržiště, OLAP databáze, Business Intelligence, ETL proces, nejpoužívanější datové formáty a další potřebné teoretické znalosti k budování datového skladu a jeho použití.

V analytické části této práce jsem nejprve popsal společnost, pro niž byl datový sklad vytvářen, a dále jsem se zaměřil na současný stav mobilní hry. Poté jsem se věnoval obecnému popisu nástrojů, které byly využity v návrhové fázi projektu. Ačkoliv jsem se při práci na projektu nezabýval rozpočtem, v rámci analýzy jsem zahrnul ceníky za jednotlivé služby, které byly využity.

Návrhová část se zabývala implementací datového skladu. Pro lepší orientaci jsem nejdříve vytvořil plán celé návrhové části. V další části jsem vytvořil jednotlivé události z herních dat. Následně jsem nakonfiguroval zdrojové nástroje a sloučil je do dvou výsledných zdrojových systémů. Při tvorbě datového skladu v nástroji Google Bigquery byli napojeny data ze zdrojových systémů, následně jsem tyto data upravil pomocí ELT metody. V poslední části tvorbě datového skladu byla vytvořena záloha. Jako demonstraci funkčnosti celého řešení byl vypracován report v podobě dashboardu, který se týkal herních dat.

## Seznam použitých zdrojů

- (1) LABERGE, R. Datové sklady – Agilní metody a business intelligence. 1. vyd. Praha: Computer Press, 2012. 352 s. ISBN 978-80-251-3729-1
- (2) NOVOTNÝ, O., J. POUR a D. SLÁNSKÝ. Business Intelligence: Jak využít bohatství ve vašich datech. 1. vyd. Praha: Grada, 2005. 256 s. ISBN 80-247-1094-3
- (3) POUR, J., M. MARYŠKA, I. STANOVSKÁ a Z. ŠEDIVÁ. Self Service Business Intelligence: Jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. 1. vyd. Praha: Grada Publishing, 2018. 352 s. ISBN 978-80-271-0616-5.
- (4) HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK. Big Data a NoSQL databáze. Praha: Grada Publishing, 2015. ISBN 978-80-247-5466-6.
- (5) Types of OLAP. Javapoint.com [online]. [cit. 2023-05-14]. Dostupné z: <https://www.javatpoint.com/data-warehouse-types-of-olap>
- (6) Difference between Star Schema and Snowflake Schema. Geeksforgeeks.org [online]. Noida [cit. 2023-05-14]. Dostupné z: <https://www.geeksforgeeks.org/difference-between-star-schema-and-snowflake-schema/>
- (7) BUSINESS INTELLIGENCE. Ostaraward.com [online]. London [cit. 2023-05-14]. Dostupné z: <https://ostaraward.com/bi/>
- (8) [online]. [cit.2023-05-14]. Dostupné z: <http://books.fs.vsb.cz/SQLReference/Sadovski/SQL-PRVN.HTM>
- (9) About. Is.com [online]. [cit. 2023-05-14]. Dostupné z: <https://www.is.com/about/>
- (10) What is Unity? – A Guide for One of the Top Game Engines. Gamedevacademy.org [online]. [cit. 2023-05-14]. Dostupné z: <https://gamedevacademy.org/what-is-unity/>
- (11) What is Firebase? The complete story, abridged. Medium.com [online]. [cit. 2023-05-14]. Dostupné z: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>
- (12) Firebase pricing. Firebase.com [online]. [cit. 2023-05-14]. Dostupné z: <https://firebase.google.com/pricing>

- (13) Firebase console. Console.firebase.com [online]. [cit. 2023-05-14]. Dostupné z: <https://console.firebase.google.com/>
- (14) Events and properties. Support.google.com/firebase [online]. [cit. 2023-05-14]. Dostupné z: [https://support.google.com/firebase/topic/6317484?hl=en&ref\\_topic=6386699&sjid=3047854641957380968-EU](https://support.google.com/firebase/topic/6317484?hl=en&ref_topic=6386699&sjid=3047854641957380968-EU)
- (15) WHAT IS BIGQUERY. Events and properties. Cloud.google.com [online]. [cit. 2023-05-14]. Dostupné z: <https://cloud.google.com/bigquery/docs/introduction>
- (16) BigQuery pricing. Cloud.google.com [online]. [cit. 2023-05-14]. Dostupné z: <https://cloud.google.com/bigquery/pricing>
- (17) What is Tableau. Guru99.com [online]. 2023 [cit. 2023-05-14]. Dostupné z: <https://www.guru99.com/what-is-tableau.html>
- (18) Save Data. Firebase.google.com [online]. [cit. 2023-05-14]. Dostupné z: <https://firebase.google.com/docs/database/unity/save-data>
- (19) Measure ad revenue. Firebase.google.com [online]. [cit. 2023-05-14]. Dostupné z: <https://firebase.google.com/docs/analytics/measure-ad-revenue#unity>
- (20) Unity Plugin Integration. Developers.is.com [online]. [cit. 2023-05-14]. Dostupné z: <https://developers.is.com/ironsource-mobile/unity/unity-plugin/#step-2>
- (21) Bigquery console. Console.cloud.google.com [online]. [cit. 2023-05-14]. Dostupné z: <https://console.cloud.google.com/bigquery>

## Seznam použitých zkratek a symbolů

ELT – Extraction, load, transform

OLAP – On-Line Analytical Processing

MOLAP – Multidimensional Online Analytical Processing

HOLAP – Hybrid Online Analytical Processing

ROLAP – Relational Online Analytical Processing

DOLAP – Desktop Online Analytical Processing

SOLAP – Spatial Online Analytical Processing

WOLAP – web-based Online Analytical Processing

PEST – Political, Economic, Social, Technological

SWOT – Strengths, Weaknesses, Opportunities, Threats

BI – Business Intelligence

ID3 – Iterative Dichotomiser 3

CART – Classification and Regression Trees

CHAID – Chi-Square Automatic Interaction Detection

JSON – JavaScript Object Notation

XML – Extensible Markup Language

SOAP – Simple Object Access Protocol

WSDL – Web Services Description Language

AJAX – Asynchronous JavaScript and XML

YAML – YAML Ain't Markup Language

RDF – Resource Description Framework

URI – Uniform Resource Identifier

JSON-LD – JSON for Linked Data

HTML – Hypertext Markup Language

CSV– Comma-Separated Values

ER– Entity-Relationship

UML– Unified Modeling Language

ETL– Extract, Transform, Load

SQL– Structured Query Language

C#– C Sharp

2D– Two-Dimensional

3D– Three-Dimensional

SDK– Software Development Kit

API– Application Programming Interface

DML– Data Manipulation Language

DDL– Data Definition Language

TB– Terabyte

USD– United States Dollar

GB– Gigabyte

iOS– Apple's mobile operating system

PPC– Pay-Per-Click

ID– Identification

## Seznam použitých obrázků a tabulek

<b>Obrázek 1: Základní komponenty systému</b> .....	14
<b>Obrázek 2: grafické znázornění architektury datové skladu</b> .....	15
<b>Obrázek 3: grafické znázornění špagetového systému</b> .....	16
<b>Obrázek 4: grafické znázornění centralizovaného systému</b> .....	17
<b>Obrázek 5: typy OLAP databáze</b> .....	19
<b>Obrázek 6: Business Intelligence</b> .....	20
<b>Obrázek 7: Schéma STAR</b> .....	28
<b>Obrázek 8: Schéma SNOWFLAKE</b> .....	29
<b>Obrázek 9: Organizační struktura</b> .....	33
<b>Obrázek 10: Výhoda Firebase cloud hosting</b> .....	35
<b>Obrázek 11: Blaze plan kalkulátor</b> .....	36
<b>Obrázek 12: Firebase typy integrace</b> .....	37
<b>Obrázek 13: Tableau rozlišení</b> .....	41
<b>Obrázek 14: Datový architektura</b> .....	44
<b>Obrázek 15: Vytvoření projektu ve Firebase</b> .....	44
<b>Obrázek 16: Přidání konfiguračního souboru</b> .....	45
<b>Obrázek 17: kód k uložení Firebase událostí do databáze</b> .....	45
<b>Obrázek 18: Všechny nadefinované události</b> .....	46
<b>Obrázek 19: automatické události</b> .....	47
<b>Obrázek 20: Správa adaptérů</b> .....	51
<b>Obrázek 21: Napojení Ironsource události do Firebase databáze</b> .....	52
<b>Obrázek 22: Vytvoření datového setu</b> .....	56
<b>Obrázek 23: Napojení Google AdWords do Google Bigquery</b> .....	57
<b>Obrázek 24: Definování tabulek</b> .....	59

<b>Obrázek 25: Finální Dashboard .....</b>	<b>64</b>
<b>Tabulka 1: Google BigQuery ceník 1 .....</b>	<b>39</b>
<b>Tabulka 2: Google BigQuery ceník 2 .....</b>	<b>40</b>
<b>Tabulka 3: Definování událostí 1 .....</b>	<b>47</b>
<b>Tabulka 4: Definování událostí 2 .....</b>	<b>48</b>
<b>Tabulka 5: Definování událostí 3 .....</b>	<b>48</b>
<b>Tabulka 6: Definování událostí 4 .....</b>	<b>48</b>
<b>Tabulka 7: Definování událostí 5 .....</b>	<b>49</b>
<b>Tabulka 8: Definování událostí 6 .....</b>	<b>49</b>
<b>Tabulka 9: Definování událostí 7 .....</b>	<b>49</b>
<b>Tabulka 10: Definování událostí 8 .....</b>	<b>50</b>
<b>Tabulka 11: Definování událostí 9 .....</b>	<b>50</b>
<b>Tabulka 12: Definování událostí 10 .....</b>	<b>50</b>
<b>Tabulka 13: Definování událostí 11 .....</b>	<b>52</b>
<b>Tabulka 14: Definování událostí 12 .....</b>	<b>53</b>
<b>Tabulka 15: Definování událostí 13 .....</b>	<b>53</b>