Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženyrství

Diplomová práce

Řešení datové kvality v podnikové praxi

Filipp Podriadchikov

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Filipp Podriadchikov

Systémové inženýrství a informatika
Informatika

Název práce

Řešení datové kvality v podnikové praxi

Název anglicky

Data quality solutions in business practice

Cíle práce

Diplomová práce je tematicky zaměřena na problematiku řešení datové kvality v podnikové praxi. Hlavním cílem práce je návrh řešení, které povede ke zlepšení datové kvality v podnikové praxi. Za účelem dosažení hlavního cíle jsou stanoveny následující dílčí cíle:
- Prozkoumat a zmapovat oblasti a principy Business Intelligence, dat a datové kvality.
- Provést analýzu a zhodnocení nástrojů pro řízení datové kvality.
- Aplikovat vybrané metodiky/techniky zabývající se unifikací dat v podnikové praxi a demonstrovat je na případové studii.

Metodika

Pro splnění výše uvedených cílů bude použito několik postupů. Jedná se o analýzu odborné literatury (zahraniční i české) a elektronických zdrojů zabývajících se oblastí datové kvality a nástrojů pro řízení datové kvality. Praktická část práce je zaměřena na provedení čištění, unifikace a deduplikace dat v datovém skladu. Na základě poznatků získaných v teoretické a praktické části budou formulovány závěry diplomové práce.

Doporučený rozsah práce

60-80 stran

Klíčová slova

čištění dat, data, datová kvalita, business intelligence, unifikace

---

Doporučené zdroje informací

BUCUR, C. Using big data for intelligent businesses. Proceedings of the Scientific Conference AFASES. 2, 605-612, 2015.

GANDOMI, A. – HAIDER, M. Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management. 35, 137-144, 2015. ISSN:0268-4012.

MADDEN, S. From databases to big data. IEEE Internet Computing, 2012, 3: 4-6.

MANYIKA, J., CHUI, M., & BROWN, B. Big data: The next frontier for innovation, competition, and productivity [R/OL]. Las Vegas: The McKinsey Global Institute, 2012.

Margaret Rouse, 2007., OLAP (online analytical processing)

Novotný, O., Pour, J., Slánský, D. Business Intelligence – Jak využít bohatství ve vašich datech. 1. vyd. Praha: Grada Publishing, 2004. 192 s. ISBN 80-247-1094-3

Smolík, Ondřej., 2008. Datová kvalita, integrita a konsolidace dat v BI

VILLARS, Richard L.; OLOFSON, Carl W.; EASTWOOD, Matthew. Big data: What it is and why you should care. White Paper, IDC, 2011.

Předběžný termín obhajoby

2021/22 ZS – PEF

Vedoucí práce

Ing. Jan Tyrychtr, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 09. 03. 2022

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Řešení datové kvality v podnikové praxi" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.03.2023 _____

Poděkování

Rád bych touto cestou upřímně poděkoval panu Ing. Janu Tyrychtorovi, Ph.D., za jeho cenné rady, ochotu a významný přínos v práci. Jeho přítomnost a vědomosti byly klíčovými faktory naší úspěšnosti, a za to mu velmi děkuji.

Řešení datové kvality v podnikové praxi

Abstrakt

Hlavním cílem této práce je provést zhodnocení a navrhnout opatření pro zlepšení datové kvality v rámci podnikového prostředí, s důrazem na oblast business intelligence (BI). Případová studie je zvláště zaměřena na procesy čištění, sjednocení a deduplikace dat uložených v datovém skladu. Druhotným cílem je provést analýzu fungování BI, datových skladů a dalších komponent BI, s ohledem na řízení kvality dat. Celá práce je rozčleněna do dvou vzájemně propojených částí.

První část je teoretická a druhá část má praktický charakter. V úvodu teoretické části jsou představeny klíčové koncepty a zásady spojené s oblastí business intelligence, datových skladů a samotných dat. Následně jsou definovány relevantní pojmy a charakteristiky týkající se kvality dat, a to včetně úkolů a klíčových aktivit nezbytných pro úspěšné řízení datové kvality. Zvláštní pozornost je věnována faktorům, které přispívají ke vzniku nedostatečné kvality dat, opatřením vedoucím k její prevenci a důsledkům, které špatná kvalita dat s sebou nese. Poslední část teoretické části se zaměřuje na vysvětlení základních operací prováděných nad daty a jejich vztah k celkové kvalitě dat.

Praktická část studie se věnuje implementaci procesů čištění, sjednocení a deduplikace dat v rámci podnikového datového skladu. V průběhu této fáze jsou navrhována konkrétní opatření a metody, které budou aplikovány na skutečná data s cílem zvýšit jejich kvalitu. Tato část zahrnuje detailní postupy a techniky, které mají potenciál efektivně odstranit nedostatky spojené s datovou kvalitou a přispět k celkovému zlepšení podnikového informačního prostředí.

Klíčová slova: čištění dat, data, datová kvalita, business intelligence, unifikace, deduplikace

Data quality solutions in business practice

Abstract

The main goal of this study is to assess and propose measures for improving data quality within the enterprise environment, with a particular emphasis on the field of Business Intelligence (BI). The case study is specifically focused on the processes of data cleansing, standardization, and deduplication related to client data stored in the data warehouse. A secondary objective is to analyze the functioning of BI, data warehousing, and other BI components with regard to data quality management. The entire work is divided into two interrelated parts.

The first part is theoretical, while the second part takes on a practical nature. In the introduction of the theoretical section, key concepts and principles associated with the field of Business Intelligence, data warehousing, and data itself are introduced. Subsequently, relevant terms and characteristics related to data quality are defined, including the tasks and key activities essential for successful data quality management. Special attention is paid to the factors contributing to inadequate data quality, preventive measures, and the consequences of poor data quality. The last part of the theoretical section focuses on explaining the fundamental data operations and their relationship to overall data quality.

The practical section of the study is dedicated to the implementation of data cleansing, standardization, and deduplication processes within the enterprise's data warehouse. During this phase, specific measures and methods are proposed to be applied to real data with the aim of enhancing their quality. This part encompasses detailed procedures and techniques with the potential to effectively eliminate deficiencies associated with data quality and contribute to the overall improvement of the enterprise's information environment.

Keywords: data cleaning, data, data quality, business intelligence, unification, deduplication

Obsah

1 Úvod

Současný technologický průmysl, který se rychle rozvíjí, klade neustálý tlak na organizace, aby revidovaly své stávající postupy a přizpůsobily se aktuálním trendům v oblasti informačních technologií. Tato adaptace je nezbytná pro udržení konkurenceschopnosti na nasyceném trhu, ať už jde o uspokojení stávajících zákazníků nebo přilákání nových. V dnešní době, kdy informace hrají klíčovou roli v rozhodovacích procesech a strategickém plánování, je nezbytné mít k dispozici spolehlivé a kvalitní datové zdroje.

Historicky organizace uchovávaly svá data různými způsoby, ať už se jednalo o klasické kartotéky, kde údaje existovaly v podobě papírových dokumentů, nebo o jednoduché tabulky a soubory v digitální formě. Nicméně, v dnešním informačním věku se tento způsob uchovávání dat jeví jako zastaralý a neefektivní. Představa ukládání smluv, faktur, nebo transakčních záznamů v papírové podobě by dnes byla nejen neefektivní, ale také nepředstavitelná. Organizace se proto vydaly cestou digitalizace a zavedly moderní databáze a datové sklady, kde mohou uchovávat svá data virtuálně.

S tímto přechodem na digitální zpracování dat však souvisí i nové výzvy a potenciální problémy. Nejvíce se projevují při manuálním zadávání dat, kdy dochází k chybám, nekonzistencím a nesrovnalostem. Takové nedostatky mohou dramaticky ovlivnit kvalitu dat a způsobit chybné interpretace, což může mít důsledky na všech úrovních organizace, od operativního řízení až po strategické rozhodování.

Řešení těchto problémů a zajištění kvality dat se stává klíčovým faktorem pro správnou interpretaci informací, na nichž organizace staví své rozhodování a plánování budoucího vývoje. Právě tato problematika datové kvality je středem zájmu této diplomní práce, která se snaží identifikovat, analyzovat a navrhnout efektivní metody a strategie pro řešení těchto výzev v podnikovém prostředí.

Volba tématu diplomní práce "Řešení datové kvality v podnikové praxi" je relevantní a důležitá z několika důvodů:

Rostoucí význam dat v podnikání: V digitální éře hrají data stále větší roli v rozhodovacích procesech podniků. Zlepšení datové kvality je klíčové pro efektivní využití těchto dat a dosažení konkurenční výhody.

Ztráty způsobené špatnou datovou kvalitou: Chybná nebo nekvalitní data mohou mít vážné následky, jako jsou chybné rozhodnutí, finanční ztráty a poškození pověsti firmy.

Potřeba splnění regulací: V mnoha odvětvích platí přísné regulace týkající se správy dat a ochrany soukromí. Nedostatečná datová kvalita může způsobit problémy s dodržováním těchto předpisů.

Zvyšující se objem dat: S rostoucím objemem dostupných dat je obtížnější udržovat datovou kvalitu. To vyžaduje systematický a strategický přístup k této problematice.

Zlepšení rozhodování: Datová kvalita ovlivňuje rozhodovací procesy na všech úrovních podniku. Špatná data mohou vést k nepřesným analýzám a rozhodnutím. Zlepšení datové kvality může přispět k lepšímu rozhodování a strategickému plánování.

Technologický pokrok: S rozvojem technologií pro správu dat a analýzu je nyní možné efektivně řešit problémy spojené s datovou kvalitou. Práce na tomto tématu může zkoumat nové nástroje a metody pro zlepšení datové kvality.

Konkurenční výhoda: Firma, která se efektivně vypořádá s datovou kvalitou, může získat konkurenční výhodu v rámci svého odvětví. Zlepšení datové kvality může vést ke zvýšení spokojenosti zákazníků a efektivnějšímu podnikání.

Celkově lze konstatovat, že problematika datové kvality má zásadní dopad na úspěšnost a udržitelnost podniků v dnešním digitálním světě, a proto je toto téma pro diplomní práci mimořádně důležité.

2 Cíl práce a metodika

2.1 Cíl práce

Diplomová práce je tematicky zaměřena na problematiku řešení datové kvality v podnikové praxi.

Hlavním cílem práce je návrh řešení, které povede ke zlepšení datové kvality v podnikové praxi.

Za účelem dosažení hlavního cíle jsou stanoveny následující dílčí cíle:

- Prozkoumat a zmapovat oblasti a principy Business Intelligence, dat a datové kvality.
- Provést analýzu a zhodnocení nástrojů pro řízení datové kvality.
- Aplikovat vybrané metodiky/techniky zabývající se unifikací dat v podnikové praxi a demonstrovat je na případové studii.

2.2 Metodika

Pro dosažení stanovených cílů se budeme řídit několika důležitými kroky a postupy. Naše práce bude zahrnovat pečlivou analýzu odborné literatury, jak zahraniční, tak české, a také elektronických zdrojů, které se týkají oblasti datové kvality a nástrojů pro její řízení. Tato teoretická část nám poskytne hlubší porozumění problematice datové kvality a přístupů k jejímu zlepšení.

Náš další významný krok zahrnuje praktickou fázi naší výzkumné práce, která je zaměřena na provedení několika klíčových procesů. Konkrétně se jedná o procesy čištění dat, standardizace a deduplikace. K řešení těchto úkolů jsme se rozhodli využít metodu vytvoření vlastní aplikace s využitím programovacího jazyka Python. Tato aplikace bude schopna provádět důkladnou validaci dat a následně korigovat identifikované chyby. Pro spuštění aplikace využijeme prostředí Jupyter Notebook, a tato aplikace bude konfigurována tak, aby prováděla operace přímo nad MySQL databází umístěnou v rámci podnikové infrastruktury.

Vývoj vlastní aplikace v jazyce Python nám umožňuje přizpůsobit proces zlepšování kvality dat konkrétním potřebám a složitostem datového souboru podniku. Tato aplikace bude poskytovat flexibilní a přizpůsobitelné řešení pro identifikaci a následnou opravu problémů spojených s kvalitou dat, což nakonec přispěje k zvýšení celkových standardů kvality dat v organizaci. Použití Jupyter Notebook nabízí uživatelsky přívětivé a interaktivní prostředí pro provádění a dokumentaci procedur zlepšování kvality dat.

Dále je důležité zdůraznit, že provádění těchto operací přímo v MySQL databázi organizace zajišťuje, že vylepšení dat jsou implementována přímo v jádru datového repozitáře, což přináší prospěch všem následným procesům a analýzám závislým na tomto datovém zdroji.

3 Teoretická východiska

3.1. Data

Data jsou lidmi vnímaná fakta, události, zprávy, měřitelné charakteristiky, zaznamenané signály. Specifikum dat spočívá v tom, že jednak existují nezávisle na pozorovateli, jednak se stávají řádnými „daty“ teprve tehdy, když existuje účelově sbírající subjekt. Z celého souboru odehrávajících se událostí, z množství vlastností skutečných objektů, badatel vyčleňuje pouze konkrétní data, malou část obrovského, potenciálně existujícího materiálu, který je podle jeho názoru nezbytný ke zpracování problému, který chce vyřešit. Data se tak stávají základem, na kterém jsou pak postaveny závěry a rozhodnutí. Jsou sekundární ve vztahu k účelu studie a předmětové oblasti, ale primární k metodám jejich zpracování a analýzy, které z dat vyčleňují pouze informace v rámci vybraného materiálu potenciálně dostupné.[1]

Pokud jsou data orientována na pochopení člověkem přímo při jejich vnímání nebo po nějaké transformaci, pak obsahují informaci. Je možné, že data neobsahují informace, které člověk aktuálně potřebuje, není schopen získat informace ze všech údajů, které má k dispozici. Šifrování informací znepřístupní data každému, kdo nemá dešifrovací klíč (kód). Šifrovaný text obsahuje informace, které ale nejsou dostupné.

Cichy a Rass se domnívají, že data jsou definována jako reprezentace faktů, konceptů nebo instrukcí formalizovaným způsobem vhodným pro komunikaci, interpretaci nebo zpracování lidmi nebo automatickými prostředky. Na základě této definice je možné domnívat se, že data představují stavy objektů nebo procesů, které probíhají v reálném světě. Pokud k datům člověk přistoupí odpovídajícím způsobem, mohou tato data poskytnout důležité informace, například o nákupech nemovitostí či o úvěrech.[2]

V ekonomii jsou data výsledkem měření (pozorování, registrace, popisu atd.) vlastností zkoumaných objektů. To znamená, že v tomto případě data charakterizují povahu a strukturu skutečně analyzovaných informací. V ekonomickém výzkumu se při analýze dat nejčastěji používá statistický přístup k interpretaci výchozích informací, což představuje nutnost vypočítat takové statistické charakteristiky, jako je průměr, rozptyl, kovariance atd.

[1] MACURA, Marek. Integration of Data from Heterogeneous Sources using ETL Technology. *Computer Science*. 2014, 15(2), s. 109–132.
[2] CICHY, Corinna, RASS, Stefan. An Overview of Data Quality Frameworks. *IEEE Access 7.* 2019, pp. 24634–24648.

Se statistickým přístupem k interpretaci výchozích informací jsou pojmy data, pozorování, implementace synonyma. Pozorování slouží jako realizace náhodné veličiny a poskytuje data pro studovaný problém.[3]

V moderní interpretaci je „analýza dat“ oborem matematiky a informatiky, který se zabývá konstrukcí a studiem nejobecnějších matematických metod a výpočetních algoritmů pro získávání znalostí z experimentálních dat, dále je to proces zkoumání, filtrování, transformace a modelování dat za účelem extrahování užitečných informací a rozhodování.

Při analýze dat se člověk obvykle snaží získat informace, které umožňují odhalit strukturu základny, na které jsou data tvořena. V tomto případě není model datové struktury zpravidla plně definován. Cílem analýzy dat je průnik do datové struktury, její informační zpřístupnění.

3.1.1 Informace

Informace jsou výsledkem transformace a analýzy dat. Rozdíl mezi informacemi a daty je v tom, že data jsou pevné informace o událostech a jevech, které jsou uloženy na určitých médiích, a informace se objevují jako výsledek zpracování dat při řešení konkrétních problémů. V databázích jsou například uložena různá data a na určitý požadavek systém správy databází vydá požadované informace.

Existují i další definice informací, např.: informace jsou informace o objektech a jevech prostředí, jejich parametrech, vlastnostech a stavu, které snižují míru nejistoty a neúplnosti znalostí o nich. Informace je podmnožinou poznatků, která je aplikována pro konkrétní řešení problémů v závislosti na konkrétní situaci, dá se tedy říci, že informace jsou data v určitém kontextu. Informace mají užitnou hodnotu, což znamená, že hodnota určité informace není pro každého stejná, liší se podle toho, kdo ji využívá. Další vlastností informací je jejich kumulativnost, tedy možnost informace hromadit. Po získání nové informace stará informace nezmizí. Informace má určitou subjektivní hodnotu, protože vzniká transformací dat na informaci a záleží přitom na tom, jaký má přínos pro toho, kdo informaci potřebuje. V případě, že data nejsou dostatečně kvalitní nebo neobsahují požadované údaje, informace má nulovou hodnotu.

[3] MAHANTI, R. Data Quality: Dimensions, Measurement, Strategy, Management, and Governance. Quality Press, 2019. ISBN 978-0-87389-977-2.

Pokud data obsahují požadované údaje, například o zákaznících, lze z nich získat informace, které mají hodnotu a v tomto případě je hodnota informací vysoká.[4]

3.1.2 Znalosti

Znalosti jsou zpracované informace zaznamenané a ověřené praxí, které byly použity a mohou být znovu použity pro rozhodování. Znalosti jsou typem informací, které jsou uloženy ve znalostní bázi a odrážejí znalosti specialisty v určité oblasti. Znalosti jsou intelektuální kapitál. Formální znalosti mohou mít formu dokumentů, norem či předpisů, které upravují rozhodování nebo učebnic. Neformální znalosti jsou znalosti a zkušenosti odborníků v určité oblasti.[5]

[4] TOUS, Rubén. Data integration with XML and semantic web technologies: novel approaches in the design of modern data integration systems. PhD thesis. Pompeu Fabra University, 2008. ISBN 978-3-8364-7138-1.
[5] MACURA, Marek. Integration of Data from Heterogeneous Sources using ETL Technology. *Computer Science*. 2014, 15(2), s. 109–132.

3.2. Datová kvalita

Jedním z hlavních problémů podnikových analytických systémů a systémů Business Intelligence je zajištění kvality dat, která jsou konsolidována pro analýzu z různých zdrojových systémů. Pokud se tomuto problému nevěnuje dostatečná pozornost, hrozí anulování výhod nejmodernějších analytických nástrojů a úsilí specialistů při vytváření analytických řešení. Samotná rozhodnutí mohou být vzdálená realitě a zkreslovat skutečný stav věcí v organizaci. Zároveň mohou rozvinutá manažerská rozhodnutí poškodit podnikání. Proto je nutné sledovat kvalitu dat a jejich transformaci ve všech fázích analytického procesu: od získávání dat ze zdrojů až po jejich zpracování v analytických systémech.[6]

Podle DMBOK je datová kvalita definována jako „*plánování, zavádění a kontrolu činností, které aplikují techniky zaměřené na řízení kvality dat tak, aby s nimi bylo možné pracovat a splňovaly potřeby konzumentů dat*“[7].

- Konzistence. V případě, že podnik má data v konzistentní podobě, neexistují rozpory napříč databázemi a potom by měly být porovnávané hodnoty z několika zdrojů dat totožné. Každý jednotlivý zákazník by měl mít jednu adresu ve všech databázích podniku. Ke standardním ukazatelům konzistence patří rozptyl, odchylka, popřípadě směrodatná odchylka.

- Přesnost. Tento pojem znamená, že měřená hodnota odpovídá reálné hodnotě neobsahující chyby, přičemž chybou může být překlep, nadbytečnost informace nebo její neaktuálnost. Důležitý je kupříkladu poměr chybovosti.

- Úplnost. Pokud mají být data úplná, musejí mít určitou logiku. Co se týká této charakteristiky dat, hledají se pole, která obsahují neúplné hodnoty, popřípadě ta pole, kde hodnoty chybějí. Ukazatelem je v tomto případě poměr úplných záznamů.

- Auditovatelnost. Provedené změny musejí být sledovatelné i v historii, je tedy nezbytné, aby data byla přístupná. Metriku zde představuje procentuální podíl polí, u kterých nelze určit, kdo a kdy je upravil a jaké úpravy to byly. Ukazatelem je rovněž procento dat, která byla změněna, časové rozestupy mezi daty a podobně.

[6] MAHANTI, R. Data Quality: Dimensions, Measurement, Strategy, Management, and Governance. Quality Press, 2019. ISBN 978-0-87389-977-2.
[7] DAMA International, DAMA – DMBOK, Data Management Body Of Knowledge Second Edition. Technics publications, 2017. ISBN 978-16-346-2236-3

- Jedinečnost. Pokud z nějakého důvodu není nutné, aby byla data zaznamenána opakovaně, mají být zaznamenána pouze jednou. Především je nutné, aby data neobsahovala chybné duplicitní údaje vztahující se k jedné entitě. Metriku zde představuje procento duplicitních hodnot.

- Včasnost. Je nezbytné, aby data byla přístupná ve správný čas, tedy tehdy, když je potřebuje ten, kdo je oprávněn s nimi nakládat. Metrikou je v tomto případě časový rozptyl.[8]

Zvyšování úrovně kvality dat není jednorázovou záležitostí, je to činnost prováděná průběžně s cílem dosažení co nejvyšší úrovně datové kvality.

3.2.1 Řízení datové kvality

Řízení kvality dat lze definovat jako „kvalitně orientovaný přístup ke správě dat“. To znamená „aplikaci konceptu celkového managementu kvality (TQM), za účelem zlepšení kvality dat a informací, včetně definice principů kvality dat, měření kvality dat (její audit a certifikace), analýzy kvality dat, čištění dat a opravy a další zlepšování procesů řízení kvality dat“. Aby byl proces řízení kvality dat efektivnější, neměl by se soustředit pouze na opravu špatných dat, ale měl by se také snažit předcházet problémům v průběhu celého životního cyklu dat v rámci organizace, aby byly splněny informační potřeby uživatelů a zainteresovaných stran. Navíc je nutné zajistit efektivní interakci mezi obchodními jednotkami a IT infrastrukturou společnosti, aby byly zohledněny jak obchodní aspekty správy dat, tak technické.[9]

Podle Mahanti lze náklady na použití nekvalitních dat klasifikovat následovně:

• Náklady na chyby a selhání procesů, které jsou důsledkem nesprávného fungování procesů, z důvodu nízké kvality dat;

• Náklady na přepracování a odstraňování problémů, za účelem dosažení požadované úrovně kvality;

• Náklady na příležitosti, spojené s promeškanými přínosy a příležitostmi.[10]

[8] SMITH, Rachel. Seven Important Characteristics Of Data Quality & Metrics To Track. [online]. [cit. 01.10.2022]. Dostupné z: https://www.clearpointstrategy.com/data-quality-metrics/
[9] AGRAWAL, Parag et al. Foundations of Uncertain-Data Integration. *Proc. VLDB Endow*. 2010, 3(1), s. 1080–1090.
[10] MAHANTI, R. Data Quality: Dimensions, Measurement, Strategy, Management, and Governance. Quality Press, 2019. ISBN 978-0-87389-977-2.

V následujícím textu budou diskutovány nejdůležitější aspekty konceptu kvality dat a také některé přístupy a metodiky používané organizacemi k hodnocení a zlepšování kvality dat.

Podle DAMA „správa dat“ nejčastěji znamená:

- Obchodní funkce, která vyvíjí strategie, plány a projekty pro řízení, ochranu, údržbu a přidávání hodnoty dat;
- Softwarová aplikace pro implementaci a hodnocení účinnosti funkcí správy dat;
- Úředníci vykonávající funkce správy dat.[11]

3.2.2 Vstupy procesu řízení datové kvality

Existuje mnoho vstupů, na základě kterých se stanoví datová kvalita. Prvním krokem je vytvoření plánu zajištění kvality dat, v němž jsou jednotlivé fáze, které povedou ke zlepšení. Je nutné mít na paměti datovou architekturu společnosti a její procesní dokumentaci. Katalogy datových zdrojů a IT služeb jsou rovněž důležité, společnost by měla mít dokumentovány všechny využívané datové zdroje[12].

3.2.3 Přístupy k řízení datové kvality

Petr Šmejkal je toho názoru, že existuje několik přístupů k řízení datové kvality. Je to přístup proaktivní, restriktivní, reaktivní a represivní[13].

Proaktivní přístup

Tento pojem znamená předcházení vzniku nekvalitních dat a informací. Dá se říci, že je jakousi prevencí, předpokládá využití metodiky řízení kvality informací již při návrhu procesů pro řízení informačního systému, tedy aby tento přístup byl účinný, je nezbytné mít odpovídajícím způsobem navržený datový model, protože nevhodné navržení modelu se stane příčinou nízké datové kvality (absence klíčů tabulky).

[11] DAMA International, DAMA – DMBOK, Data Management Body Of Knowledge Second Edition. Technics publications, 2017. ISBN 978-16-346-2236-3
[12] DENNIS, Amber. Data Architecture with Data Governance [online]. [cit. 11.10.2022]. Dostupné z: https://www.dataversity.net/data-a rchitecture-with-data-governance-a-proactive-approach/
[13] MARR, Bernard. How To Define A Data Use Case – With Handy Template. [online]. [cit. 09.10.2022]. Dostupné z: https://bernardmarr.com/default.asp?contentI D=1837

Je tedy nezbytné, aby každá společnost, která pracuje s informacemi a daty, měla datový model, který je odpovídajícím způsobem navržený a správně řízený. Předpokladem úspěchu v tomto případě je určení pouze jednoho datového architekta, protože potom nebude problém udržet jednotný styl návrhu databáze. Také je možné využít CASE nástroje pro řízení dokumentace a jako nezbytné se ukazuje implementace metadat, zejména těch, která jsou využívána pro mapování transformačních pohledů na plnění tabulek, definici atributů, vazeb a omezení pro server a podobně.

Restriktivní přístup

V tomto případě se jedná o vytvoření striktních pravidel a omezení, která mají zamezit vzniku nekvalitních informací. To však může být problematické, protože existují obchodní pravidla, která jsou velice komplikovaná, a je příliš náročné zavádět je prostřednictvím těchto omezení. Kromě toho se může stát, že nové omezení, jestliže nebude implementováno vhodným způsobem, se obrátí v chybu ve stávajícím systému. Tato omezení jsou zaváděna pouze ve stanovené úrovni a za stanovených podmínek[14].

Reaktivní přístup

Jak už napovídá název tohoto přístupu, v tomto případě se jedná o reakci na vzniklý problém. Tyto přístupy jsou tedy opakem přístupů proaktivních, jejichž cílem je vzniku problémů předcházet. Příkladem může být unifikace dat, která je předmětem praktické části práce, přičemž tento přístup neřeší příčinu vzniku vadných dat, nýbrž pouze hledá způsob, jak tuto situaci napravit. Dále je to například kontrolní report, korekce dat, data profiling, konsolidace, unifikace a čištění dat[15].

Represivní přístup

Tento přístup využívá smluvní závazky a zákonné normy. Jsou to především dokumenty související s utajováním informací, s jejich zabezpečením nebo omezením při jejich využívání (GDPR) a také to mohou být smluvní dohody, jejichž hlavní součástí je kvalita dat[16].

[14] DENNIS, Amber. Data Architecture with Data Governance [online]. [cit. 11.10.2022]. Dostupné z: https://www.dataversity.net/data-a rchitecture-with-data-governance-a-proactive-approach/
[15] MAHANTI, R. Data Quality: Dimensions, Measurement, Strategy, Management, and Governance. Quality Press, 2019. ISBN 978-0-87389-977-2.
[16] MARR, Bernard. How To Define A Data Use Case – With Handy Template. [online]. [cit. 09.10.2022]. Dostupné z: https://bernardmarr.com/default.asp?contentI D=1837

3.2.4 Role a zodpovědnosti

Samostatně stojí za to věnovat pozornost odpovědnosti za data, protože proces řízení kvality dat je komplexní proces v rámci provozních procesů podniku. Měla by být vytvořena instituce vlastníků dat (Data Stuard), jejichž úkolem je zajišťovat procesy pro řízení kvality a bezpečnosti dat v oblasti jejich odpovědnosti. Je důležité si uvědomit, že oblast odpovědnosti je určena pouze povahou dat a nezávisí na IT systémech, kde se tato data používají. Chief Information Officer (CIO) je zodpovědný za celkový proces řízení kvality. Jeho povinností je zajišťovat fungování a provádění postupů na straně IT služby, kontrolovat úroveň služeb a přispívat ke zlepšování efektivity používaných praktik.

Je třeba zdůraznit, že datová kvalita určitým způsobem ovlivňuje všechny uživatele systému. Je ovšem důležité určit, kdo bude za datovou kvalitu odpovídat. Jednotlivé role související s řízením datové kvality jsou definované prostřednictvím RACI matice. Je to metoda, která se používá pro předání odpovědnosti buďto konkrétním osobám nebo pracovním pozicím v průběhu plnění nějakého úkolu, například v projektu nebo v rámci pracovní organizace. RACI jsou počáteční písmena slov, jejichž význam je popsán níže.

- R: Responsible je osoba odpovědná za vykonání nějakého úkolu,
- A: Accountable je osoba odpovědná za celý úkol, za vše, co bylo v souvislosti s tím uděláno,
- C: Consulted je osoba předávající rady týkající se určitého úkolu, obvykle to bývá seniorní zaměstnanec
- I: Informed je osoba informovaná o tom, jak probíhá plnění úkolu a o rozhodnutích, která byla uskutečněna v souvislosti s tímto úkolem, například projektový manažer.

Vedení společnosti by mělo pochopit, že proces řízení kvality dat nelze řešit pouze na úkor IT oddělení, jehož úkolem je plnit servisní funkci pro obchod a reagovat v rámci plnění zadaných úkolů. IT specialisté nemohou zcela samostatně určovat strukturu dat, logiku jejich zpracování, pravidla plnění adresářů a reportovací algoritmy. Zákazníky IT oddělení jsou podnikové divize, které mají zájem na zefektivnění své práce.

3.2.5 Klíčové aktivity procesu řízení datové kvality

Existuje několik hlavních aktivit, jejichž prostřednictvím jsou řízena externí data. V prvé řadě je to „analýza potřeb sdílení dat s partnery společnosti“, která je vhodná pro vyhodnocení obchodního partnera vzhledem k jeho významu pro podnik, určení míry jeho důvěryhodnosti, stanovení způsobu dodávky dat a řízení[17].

Jednou za dalších těchto aktivit je „analýza business požadavků na externí zdroje dat“. Jde o to, že každý jednotlivý požadavek je třeba vyhodnotit z mnoha hledisek. Například z hlediska obchodního, zda má v úmyslu podporovat podnikové procesy, zda existuje nějaká přidaná hodnota, popřípadě jestli daný zdroj už v rámci firmy existuje, dále je hodnocena náročnost požadavku a podobně. Každému z požadavků je přiřazena určitá priorita, popřípadě mohou být stejně strukturované. Informace související s daným požadavkem mají vliv na úkoly analýzy a na plánování rozvoje datových zdrojů.

Další aktivitou je „technicko-organizační řešení sdílení dat“. Je nutné, aby zaměstnanci, kteří mají oprávnění, byli vybaveni potřebnými přístupovými právy. Data musejí být zabezpečena a vybrané řešení musí splňovat určité požadavky. Data je možné ukládat různými způsoby, například přímo na serverech, které má ve vlastnictví společnost, kde je ovšem nutné počítat s vysokými náklady a horší škálovatelností, v hostingových centrech, která jsou spravována externě, tedy mimo společnost, popřípadě v cloudu, což je řešení, které má nižší počáteční náklady a možnost vyšší škálovatelnosti.[18]

Dále je to „analýza dostupných externích zdrojů dat“, která je zaměřena na zdroje dostupné online i na placené komerční soukromé databáze, což mohou být specializované technologické zdroje či marketingová data. Toto řešení může společnosti rozšířit databázi. Analýza v tomto případě obsahuje informace o „původci dat, dostupnost, míra datové kvality a validace dat, nákladovost daných dat“.

Na to navazuje „volba externích zdrojů dat“. To probíhá tak, že oddělení odpovědné za data vybere z analýzy několik datových zdrojů, které nejvíce ze všech odpovídají požadavkům. Poté je provedeno srovnání vybraných zdrojů vzhledem k datové kvalitě, předpokládané výši nákladů, dodavatelské společnosti, aby mohla být vybrána vhodná varianta.

[17] MBI. Řízení datových zdrojů a jejich kvality. [online]. [cit. 15.10.2022] Dostupné z: https://mbi.vse.cz/
[18] MBI. Řízení datových zdrojů a jejich kvality. [online]. [cit. 15.10.2022] Dostupné z: https://mbi.vse.cz/

Je dobré využít možnost testování poskytovaných služeb v případě, že dodavatel tuto možnost nabídne a výsledek konzultovat s příslušným obchodním manažerem společnosti, který má o data zájem.[19]

Poslední z důležitých aktivit je „obchodně-technologické zajištění“. Toto je fáze, v níž jsou připravovány smlouvy s dodavateli a po jejich podepsání a po následné úpravě infrastruktury je externí datový zdroj zaveden do datové základny.[20]

3.2.6 Stárnutí dat

Informační stárnutí je často spojováno s pojmem hodnoty. Například informace, které již nemají pro spotřebitele žádnou hodnotu, jsou často považovány za zastaralé. Tato myšlenka informačního stárnutí má určitý význam: vlastnosti opačné ke stárnutí jsou novost a praktická užitečnost informací pro řešení naléhavých vědeckých, technických a jiných problémů. Zda jsou informace zastaralé nebo ne, lze pochopit pouze ve srovnání s jinými - novými, relevantními, užitečnými. N. Wiener napsal, že hlavním důvodem stárnutí informací není samotný čas, ale vznik nových informací.

Informační stárnutí může být absolutní i relativní. Za *absolutně zastaralé* informace jsou považovány informace, které se s příchodem nových informací ukázaly jako nespolehlivé. O relativní povaze informačního stárnutí lze uvažovat z hlediska jeho novosti nejen na základě časových parametrů, ale i ve vztahu k souhrnným či individuálním znalostem. Pokud se například ve fyzice objeví nová částice, pak informace o ní budou nové jak pro fyziku, tak pro každého, kdo o ní čte nebo slyší. Přitom sdělení o dříve známých částicích ve fyzice (ve vztahu k celkovým znalostem této vědy) bude považováno za poměrně zastaralé a ve vztahu k odděleným individuálním znalostem konkrétního člověka, např. pro školáka, může to být nové.

Při zvažování problému stárnutí informací by se tedy neměla opomíjet souvislost mezi informací samotnou a jejím konzumentem, která často v praxi působí jako jediný pozorovatelný typ spojení.

[19] MBI. Řízení datových zdrojů a jejich kvality. [online]. [cit. 15.10.2022] Dostupné z: https://mbi.vse.cz/
[20] MBI. Řízení datových zdrojů a jejich kvality. [online]. [cit. 15.10.2022] Dostupné z: https://mbi.vse.cz/

3.2.7 Přemisťování a reorganizace dat

Data, s nimiž se pracuje při vytváření nového systémového řešení, jsou během vývoje převedena do požadované podoby. V této fázi se pracuje s velkým množstvím dat a není vyloučené, že dojde k chybě, která je sem často zanesena při procesech ETL. Zdrojem dat jsou zdrojové systémy a během tohoto procesu se přenášejí data i z dalších zdrojů, které existovaly již dříve. V nich však mohou být chyby, které se mohou přenést také do databáze příjemce, pokud nejsou vstupní data dobře zkontrolována. Ne vždy je vlastní jediná společnost, vlastníkem může být další společnost, která ručí za jejich kvalitu. Pokud se ukáže, že data jsou nekvalitní, je nutný kontakt mezi spotřebitelem a dodavatelem, aby mohla být provedena oprava nebo vyhodnoceny náklady, které bude nutné na nápravu vynaložit.

3.2.8 Základní operace s daty

Během informačního procesu se data transformují z jedné formy do druhé pomocí různých metod. Zpracování dat zahrnuje mnoho operací. S rozvojem vědeckého a technologického pokroku a obecnými komplikacemi komunikace v lidské společnosti neustále rostou mzdové náklady na zpracování dat. Především je to dáno neustálým komplikováním podmínek pro řízení výroby a společnosti. Druhý faktor, který také způsobuje obecný nárůst objemu zpracovávaných dat, je spojen s vědeckotechnickým pokrokem, a to s rychlým vznikem a zaváděním nových datových nosičů, způsobů jejich uchovávání a dodávání.

Možné operace s daty zahrnují následující:[21]

1. *Získávání* – nebo shromažďování informací, za účelem zajištění dostatečné úplnosti pro rozhodování.

2. *Formalizace* - uvedení dat do jedné formy pro zvýšení úrovně dostupnosti k nim.

3. *Třídění* - řazení dat podle daného atributu za účelem jejich pohodlného použití a dalšího zpracování.

4. *Filtrování* – vyřazení nepotřebných dat, která nenesou žádné cenné informace.

[21] MAHANTI, R. Data Quality: Dimensions, Measurement, Strategy, Management, and Governance. Quality Press, 2019. ISBN 978-0-87389-977-2.

5. *Ochrana* - opatření zaměřená na zákaz neoprávněné úpravy, mazání a přidávání dat v systému.

6. Archivace - organizace ukládání dat, slouží ke snížení ekonomických nákladů a zvyšuje celkovou spolehlivost informačního procesu jako celku.

7. Pohyb - příjem a přenos dat mezi vzdálenými oblastmi informačního procesu.

8. Transformace - přenos dat z jedné formy do druhé, což je poměrně často spojeno se změnou typu média, např. videosekvence může být uložena jak ve formě kazety, tak v elektronické podobě v paměti počítače.

3.2.9 Výstupy procesu řízení datové kvality

Výsledkem procesu jsou tyto výstupy:

- Analýza a plán rozvoje datových zdrojů
- Aktualizovaná datová architektura
- Aktualizovaný katalog datových zdrojů

3.3 Data Governance

Tento pojem je dosti obtížné definovat, protože existuje mnoho definic a tedy i mnoho způsobů, jak jej pochopit. Podle názoru Ladleyho se jedná o soubor tří základních konceptů: data managementu, datové architektury a Enterprise Information Managementu (EIM), což je management podnikových informací. EIM je program, který spravuje podniková informační aktiva a jeho cílem je podporovat obchod a zvyšovat jeho hodnotu. Podle Ladleyho jsou výstupem zavedení DG soubor principů a závazných pravidel a doporučuje zavedení DG v roli pozvolného, průběžného vývojového procesu, který má dopad na organizační strukturu firmy.[22]

Také lze DG definovat jako „procesní a organizační zastřešení vybraných oblastí za využití specifických nástrojů – například pro správu metadatových úložišť". Cílem činností v souvislosti s DG je zavedení prostředí, které je zcela kontrolované v rámci všech operací v organizaci a v organizační struktuře společnosti se DG realizuje prostřednictvím role datového vlastníka. Aby bylo možno dosáhnout cíle, je nezbytné provést změny ve struktuře organizace, v podnikových procesech i směrnicích, není to tedy zcela snadné a je k tomu nutná podpora podnikového managementu.[23]

Obsah této diplomové práce však nejvíce souzní s definicí, která říká, že Data Governace se rovná strategické úrovni řízení dat (potažmo informatiky), kterou je stejně tak jako v případě ostatních klíčových aktiv nutné sladit jednak se strategickým řízením těchto aktiv, jednak se strategií celé firmy. V závislosti na této definici jsou v dalších podkapitolách uvedeny úlohy řízení datových zdrojů a jejich kvality.

Bez ohledu na definici DG existuje mnoho důvodů, proč se tímto pojmem zabývat. Je to například zvyšování objemu dat v organizacích, což mnohdy přináší určitou nekonzistenci, kterou je nutné rozpoznat včas, tedy dříve, než na základě existence nekvalitních dat padnou chybná rozhodnutí. Zvyšující se množství reportů vytváří potřebu pochopit data a porozumět jim, což se netýká pouze jednotlivců, ale organizace jako celku.

[22] LADLEY, John. Data Governance: How to Design, Deploy and Sustain an Effective Data Governance Program. 2., vydání. Cambridge, Massachusetts: Academic Press, 2019. ISBN: 9780128158326
[23] HÁJEK, Petr. Jak „mít pořádek" v datech. [online]. [cit. 07.10.2022]. Dostupné z: https://profinit.eu/blog/jak-mit-poradek-v-datech/

Mohou to být některé pojmy, zejména z oblasti obchodu, které jsou často v jednotlivých odděleních pochopeny odlišně a DG může být prostředkem k včasnému odhalení problémů tohoto typu a k definování standardů, tedy k tomu, aby důležité pojmy byly chápány v celé organizaci stejně[24].

3.3.1 Oblasti související s Data Governance

Data Strategy má počátek v podnikové strategii. DG z datové strategie vychází a také ji podporuje s DM. Je však důležité vědět, že na základě inovativního způsobu využití dat, která již existují, může vzniknout nová podniková strategie. Rovněž je důležité vědět, že podniková a datová strategie musejí být v souladu.

Data Architecture je určitým náhledem na datové zdroje, na databáze a jejich charakteristiky a další aspekty a zprostředkovává jejich rozvoj. Pokud je architektura řádně zdokumentovaná, je východiskem pro DG a DM a dává možnost využívat moderní technologie v souvislosti s naplněním cílů podnikové strategie (MBI portál). Také je důležité vědět, že Data Architecture není součást DG, že je to spíše disciplína, která umožňuje nahlédnout na tentýž problém z pohledu, který je více technologického rázu, přičemž obě disciplíny by se měly doplňovat.[25]

3.3.2 Oblasti Data Governance

K typickým rolím a odpovědnostem při zavedení DG patří Chief Data Officer (CDO, datový ředitel), Data Owner (vlastník dat), Data Steward (datová obsluha). Klíčová je role datového ředitele, který může delegovat strategické řízení na jednotlivé vlastníky dat. Datový ředitel je obvykle součástí vrcholového managementu organizace.[26]

Master and Reference Data Management, čili Správa kmenových a referenčních dat, má v kompetenci datovou kvalitu a aktuálnost dat v tabulkách kmenových dat, která se

[24] LABERGE, R. Datové sklady - Agilní metody a business intelligence. Praha: Computer Press, Albatros Media a.s. 2016. ISBN 978-80-251-3729-1
[25] DENNIS, Amber. Data Architecture with Data Governance: A Proactive Approach. [online]. [cit. 18.10.2022]. Dostupné z: https://www.dataversity.net/data-a rchitecture-with-data-governance-a-proactive-approach/
[26] SLÁNSKÝ, D. Data a analytika pro 21. století. Praha: Professional Publishing, 2018. ISBN 978-80-88260-25-7.

týkají například zákazníků, produktů, určitého regionu a také integritu veškerých vzájemných vazeb. V této souvislosti se také hovoří o Golden Record (zlatý záznam).

Data Security Management představuje soubor nástrojů, které umožňují dostupnost a celistvost dat. Procesy zálohování a obnovování dat, řízení přístupu na úrovni datové integrace, vizualizační a analytické vrstvy a monitorování mimořádných a neobvyklých aktivit patří do oblastí, které mají těsnou souvislost s činností datových týmů. Tento soubor, tady Data Security Management, nemá v rámci modelu DG, který je navržen níže, svou vlastní kapitolu, protože toto téma je vzhledem ke svému rozsahu a komplexnosti v rámci této závěrečné práce neuchopitelné, stejně tak jako v této práci nemá samostatný prostor problematika GDPR.[27]

Data Quality Management má hlavní roli v pochopení analytických výstupů jako celku. Pokud jsou data chybná, nepřesná, neodpovídají skutečnosti, popřípadě jsou dodána se zpožděním, ztrácejí na důležitosti, stejně jako se ztrácí důvěra uživatelů. Čím větší je objem dat, čím častěji probíhají změny a čím kvalitnější je self-sefvice, tím vyšší jsou nároky na to, aby byla zajištěna co nejvyšší možná datová kvalita napříč Data Pipeline[28].

Analytici a vývojáři potřebují zejména tzv. Data Discovery nástroje, které jim umožňují objevovat datové zdroje a pochopit jejich strukturu a Metadata Management nástroje jim právě toto umožňují, protože poskytují slovníky a katalogy umožňující pochopení podnikových pojmů a výrazů a podporují vyhledávání informací. Různé datové a procesní modely mohou pomoci definovat a podchytit komplexitu řešení. Data Lineage, tedy mapa původu dat, prostřednictvím metadat například zachycuje informace sdělující, kdy a jak byl určitý záznam vytvořen, kolik času na to bylo potřeba, která data figurovala na vstupu a která na výstupu.

Knowledge and Collaboration Management je obvykle portálem shromažďujícím výše uvedené komponenty, což mohou být schémata, vyhledávače, slovníky, katalogy a ty potom doplňuje o další materiály, které podporují efektivní práci týmu (praktické ukázky kódů nebo tutoriály). Spadají sem také nástroje zprostředkující komunikaci jako například Slack a Trello a nástroje pro vývoj aplikací, například Gitlab.[29]

[27] SLÁNSKÝ, D. Data a analytika pro 21. století. Praha: Professional Publishing, 2018. ISBN 978-80-88260-25-7.
[28] LABERGE, R. Datové sklady - Agilní metody a business intelligence. Praha: Computer Press, Albatros Media a.s. 2016. ISBN 978-80-251-3729-1
[29] SLÁNSKÝ, D. Data a analytika pro 21. století. Praha: Professional Publishing, 2018. ISBN 978-80-88260-25-7.

3.3.3 Hybné síly

Hybnými silami, které vedou k zavádění DG v organizacích, jsou především stále narůstající objemy dostupných dat, která jsou začleněna do podnikových aplikací. Jak už bylo zmíněno, čím více dat a čím rozsáhlejší integrace, tím více prostoru pro chyby a kromě toho je potom velmi náročné pochopit všechny souvislosti. V současné době jsou v této souvislosti často využívané specializované nástroje určené ke specifickým úkonům. K dalším hybným silám patří regulace a opatření, z nichž nejvíce známé je patrně GDRP, podle nichž organizace mají povinnost mít v datech určitý systém a musejí mít tedy zavedené zákonné procesy opravňující k nakládání s těmito daty. Také není výjimkou, že zejména ve velkých organizacích, ale nejen v nich, existuje mezi jednotlivci i mezi odděleními odlišná interpretace dat a pojmů, což organizace přivádí k zavádění podnikových slovníků.[30]

3.3.4 Rámcový model Data Governance v podnicích

Základním krokem k vytvoření praxe Data Governance je navrhnout vyspělou IT infrastrukturu. Práce s daty, která se stala strategickým digitálním aktivem, vyžaduje od výpočetní infrastruktury maximální flexibilitu. Hlavním úkolem každého CIO je proto vytvořit infrastrukturu pro produktivní práci analytiků a platforem.

Jedním z příkladů práce s infrastrukturou v rámci strategie Data Governance je vytvoření jediného virtuálního prostředí pro práci s daty. To vyžaduje správně vybudované kapacity, které kombinují produktivní výpočetní platformy pro velké množství informací, úložiště pro archivy, disky pro ukládání horkých dat a rychlý přístup k nim kdykoli.

Po vytvoření výpočetní infrastruktury připravené na přechod na Data Governance se můžeme rozhodnout, kam data uložit. To lze provést v rámci společnosti, nebo může být využito cloudu, externího datového centra, nebo soukromého datového centra. V datovém katalogu se zároveň shromažďují všechna technická metadata (například údaje o informačních systémech, ve kterých jsou informace uloženy) a pro každý obchodní termín je v obchodním glosáři zaznamenána jediná definice pro celou společnost.[31]

[30] COUTURE, Nancy. Why data governance? [online]. [cit. 22.10.2022]. Dostupné z: https://www.cio.com/article/3245588/why-data-governance.html
[31] GÁLA, L., POUR, J., ŠEDIVÁ, Z. Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi. 3., aktualizované vydání. Praha: Grada Publishing, 2015. ISBN 978-80-247-5457-4

Dále by měla být upravena správa dat ve stávajících produkčních systémech: zavést vzor a rozdělit odpovědnosti a pravomoci za zřizování a ověřování dat tak, aby je zadávali a ověřovali zkušení odborníci, a ne všichni uživatelé systémů. Kromě toho můžete přidat postup pro křížovou kontrolu dat mezi systémy, například v podobných systémech, zkontrolovat přítomnost již zadaných dat a zadat je do svého systému podobným způsobem, aniž by došlo k rozporu se zadanými. Pokud jsou rozpory nevyhnutelné, zahajte postup pro opravu neplatných údajů.

Dalším krokem je nastavení procesů pro extrakci dat z transformačních systémů a jejich načtení do požadovaných pohledů (ETL, Extract, Transform & Load). To znamená, že pro správný přenos musí být data přenesena do jediného systému hodnot a detailů. Z toho plynoucí výhody jsou hotové postupy pro vykládání a nezbytné transformace dat, které lze na požádání znovu použít. Nástroje ETL zpravidla umožňují rychle přidávat a měnit nastavení transformace, což dále snižuje náklady.

Současně s ETL je žádoucí implementovat Enterprise Service Bus (ESB), protože to zautomatizuje proces získávání správných dat na správná místa ve správný čas, zaručí takové doručení a centralizuje správu integrace. Některé společnosti s tím končí, protože další akce pro organizace určitých odvětví a velikostí budou vyžadovat velké investice do změn pracovních procesů.

Po vytvoření základních potřeb pro Data Governance se můžeme bavit o plnohodnotné kontrole kvality dat. To se řeší pomocí datového profilování. Jsou identifikovány parametry, které mají být kontrolovány, a je zaveden pojem „kvalitativní údaje“. Všechny tyto iniciativy umožňují zahájit plnohodnotnou práci na zlepšování kvality dat a jejich udržování v tomto stavu.

Kvalitu dat lze udržovat pomocí postupů pro kvalitu dat. Jedná se o poměrně složitý mechanismus, který vyžaduje značné propracování detailů: jmenování odpovědných osob (správce dat), vývoj metodiky, použití automatických systémů. Ale při použití těchto mechanismů můžeme mluvit o kvalitních, konzistentních, spolehlivých, neopakujících se datech.

Dalším velkým blokem je zavedení nástroje Master Data Management (MDM). MDM je potřeba ke správě procesu shromažďování dat, jejich ověřování, deduplikace a přeměny běžného úložiště na čisté úložiště dat. Realizace této části s sebou nese významnou změnu v práci podniku s daty.

Body jejich zřizování jsou kontrolovány, upravovány případné změny, zaváděny nové role, pro které jsou vyvíjeny postupy a nástroje, které pomáhají při rozhodování v případě rozporů.

A samozřejmě před celým procesem tvorby Data Governance je velmi důležité implementovat pravidla pro zajištění kvality dat a provést příslušné organizační změny. To vše umožní přesněji a rychleji analyzovat data, generovat zprávy a činit správná obchodní rozhodnutí. A konečně odpovědět na hlavní výzvu – vytvořit adekvátní model a nástroje pro správu dat.[32]

Pro lepší pochopení je na obrázku uveden následující příklad:

Obr. 1 – Platforma pro správu Master Data

Správa dat podle Data Governance ovlivňuje strategickou, operační a taktickou úroveň organizace. Proto musí být správa dat prováděna v nepřetržité integraci, aby byla data efektivně organizována a využívána.

Hlavní charakteristiky Data Governance budou: dostupnost, pohodlí, bezúhonnost, bezpečnost. Jako každý systém Data Governance obsahuje: management, soubor norem a pokynů, plán implementace standardů a pokynů.

Klíčovými funkcemi programů Data Governance je především vytváření procesů zadávání dat, v souladu se zavedenými standardy. Příklady zahrnují správnost termínů, konzistentnost obchodních procesů, srovnatelnost s podobnými nebo souvisejícími údaji.

Jmenování osob odpovědných za zadávání, správnost a bezpečnost informací.

[32] SLÁNSKÝ, D. Data a analytika pro 21. století. Praha: Professional Publishing, 2018. ISBN 978-80-88260-25-7.

Jednotný systém správy dat, tzn. stejné standardy platí pro všechny obchodní procesy.

Mezi oblíbené programy pro správu dat v současnosti patří:

Axon Data Governance

Obsahuje sadu funkcí, včetně obchodního glosáře a datových nástrojů.

Základní funkce:

• údržba koncepčního a logického datového modelu

• vyhledávání objektů všech kategorií metadat

• vytváření a udržování dalších kategorií metadat, seznamů a rolí zúčastněných stran

• řízení životního cyklu dat

• nastavení schvalovacích procesů (například obchodních podmínek)

• vytváření vazeb mezi logickými a fyzickými datovými modely

• sledování a vizualizace aktuálních ukazatelů kvality dat

Egnyte

Jediná platforma pro správu dat. Egnyte je vhodný pro všechny velikosti podniků a má nekonečnou škálovatelnost, díky čemuž je perfektní volbou pro rostoucí organizace. Program se dobře integruje s produkty třetích stran. To zahrnuje nativní podporu pro Microsoft 365 a Google Workspace a také řadu externích integrací, jako je Zoom a Trello. Servery Egnyte jsou uloženy v datovém centru SSAE-16 Level II. Každý soubor je chráněn jak při přenosu, tak v klidu, pomocí 256bitového šifrování AES.

Hlavní výhodou tohoto systému bude:

• Jediný datový prostor

• Automatické ověřování správnosti a srovnatelnosti

• Bezpečnost

• Práce na dálku

• Každodenní podpora

• Schopnost spolupracovat

• Pravidelné aktualizace

SAP Master Data Governance

Systém pro konsolidaci a centrální správu životního cyklu kmenových dat.

Program konsoliduje kmenová data v celé organizaci a poskytuje centralizovanou správu pro jednotný pohled na podnikání, stejně jako předdefinované doménově specifické aplikace a strukturu pro zákaznicky definovaná kmenová data.

Tento systém poskytuje různé formy přístupu, a to jak lokální, tak cloudový.

Školení v dovednostech nezbytných k optimalizaci používání SAP Master Data Governance.

Existují hotové modely pro prezentaci informací.

OvalEdge

Program umožňuje kombinovat databáze do jednoho inteligentního katalogu skenováním systémů a dat.

Tento systém je vhodný jak pro zkušené uživatele, tak pro začátečníky, protože uživatelsky přívětivé rozhraní usnadňuje používání.

Poskytuje spolupráci pro analytické, informační a obchodní týmy, která vám pomůže rychle a snadno vyhledávat a sdílet data.

Spolupracuje s různými platformami pro správu dat, business intelligence a analýzu. Některé z nich zahrnují Amazon S3, Salesforce, MySQL, MongoDB atd.

Všechna data lze organizovat pomocí značek, statistik využití a dalších značek.

StealthAUDIT

Platforma pro správu nestrukturovaných dat. Možnost integrace s IAM, HR systémy a dalšími aplikacemi, což usnadňuje ochranu soukromí dat. Poskytuje desítky již vygenerovaných šablon výkazů a také umožňuje vytvářet potřebné výkazy pro zkušené uživatele. Jednou z výhod programu je automatická detekce zastaralých a opakujících se dat a jejich vyčištění ze systému.

Umožňuje sledovat zastaralé seznamy adresátů, abyste bylo vidět, které z nich se nepoužívají, a také určuje, kdo vlastní DL. Pokud nebyl přiřazen žádný vlastník, určuje, kdo je „nejpravděpodobnějším vlastníkem“.

3.4 Business Intelligence

Vznik termínu Business Intelligence se datuje do roku 1958, kdy americký vědec Hans Peter Lun publikoval článek „A Business Intelligence System“ v IBM System Journal. O třicet let později, na konci osmdesátých let, analytici společnosti Gartner podali širší výklad pojmu Business Intelligence: „proces zaměřený na uživatele, který zahrnuje přístup k informacím a jejich zkoumání, jejich analýzu, rozvoj intuice a porozumění, které vedou ke zlepšení a neformálnímu rozhodování“. Později v roce 1996 se objevilo vysvětlení: „Business Intelligence je nástroj pro analýzu dat, vytváření sestav a dotazování, který může pomoci podnikovým uživatelům orientovat se v množství dat, aby z nich mohli syntetizovat smysluplné informace.“[33]

Nezávislá výzkumná firma Forrester Research definuje Business Intelligence jako „soubor metodologií, procesů, struktur a technologií, pro přeměnu nezpracovaných dat na smysluplné a použitelné informace, používané k efektivnímu pochopení obchodních procesů a přijímání informovaných rozhodnutí na strategické, taktické a operační úrovni“.

Business Intelligence (nebo BI) je systém, který automaticky shromažďuje informace z různých zdrojů, spojuje je do souvislého obrazu ve vhodném formátu a umožňuje vytvářet sestavy rychle a pohodlně, analyzující velké množství dat. Analytici očekávají, že globální trh se systémy BI vzroste z 24 miliard USD v roce 2021 na 43 miliard USD v roce 2028, což naznačuje poptávku po podnikání.[34]

Řešení BI dnes nepředstavují pouze reporting, analýzy a poskytování informací, ale také takové komponenty struktury, jako je sběr dat, integrace, řízení kvality dat. Reporting je jen částí funkcí technologie Business Intelligence.[35]

Shrneme-li výše uvedené, můžeme dojít k závěru, že business intelligence v širokém slova smyslu definuje proces přeměny dat na informace a znalosti o podnikání, na podporu lepšího a neformálního rozhodování, informační technologie, metody a prostředky sběru dat, konsolidaci informací a poskytování přístupu obchodním uživatelům, znalostem, obchodním znalostem získaným z hloubkové analýzy dat.

---

[33] OSIRI, John Kalu. Entrepreneurial Marketing: Creating a Customer Base. London: Unleash Publishing, 2014. ASIN B00HPL25ZA

[34] OSIRI, John Kalu. Entrepreneurial Marketing: Creating a Customer Base. London: Unleash Publishing, 2014. ASIN B00HPL25ZA

[35] POUR, J. a kol. Self service business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. Praha: Grada Publishing, 2018. ISBN 978-80-271-0616-5

3.4.1 Komponenty Business Intelligence

Srdcem technologie BI je organizace přístupu koncových uživatelů a analýza strukturovaných kvantitativních dat a obchodních informací. To generuje interaktivní proces obchodního uživatele, tzn. projev intuice, tvoření závěrů, hledání vztahů, což umožňuje efektivně změnit podnik pozitivním směrem. Business Intelligence je určen pro širokou škálu uživatelů v podniku, včetně manažerů a analytiků.[36]

Dnešní kategorie produktů BI zahrnují: BI nástroje a BI aplikace.

Nástroje se zase dělí na:

- generátory dotazů a sestav – „desktopové“ nástroje, které uživatelům poskytují přístup k databázím, provádějí analýzy a generují sestavy;
- pokročilé nástroje BI (především nástroje pro online analytické zpracování OLAP);
- Enterprise BI suites (EBIS) – způsob poskytování nástrojů BI, které byly dříve dodávány jako různé produkty;
- BI platformy - sady nástrojů pro tvorbu, implementaci, podporu a údržbu BI aplikací.

Typické bloky moderních BI systémů

Hlavní schopnosti systémů BI jsou rozvíjeny ve čtyřech hlavních oblastech: ukládání dat, integrace dat, analýza dat a prezentace dat. Ukládání dat používaných pro obchodní analýzu je organizováno ve speciálních úložištích (datovém skladu). Tyto údaje by měly odrážet aktuální, skutečný a úplný obraz podnikání. Informace v datovém skladu, včetně historických dat, jsou shromažďovány z různých provozních (transakčních) systémů a jsou speciálně strukturovány pro efektivnější analýzu a zpracování požadavků (na rozdíl od konvenčních databází, kde jsou informace organizovány tak, aby optimalizovaly dobu zpracování aktuálních transakcí). Pro řešení užších, specifických úloh lze z obecného úložiště izolovat podmnožiny dat – tzv. data marts.[37]

[36] GÁLA, L., POUR, J., ŠEDIVÁ, Z. Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi. 3., aktualizované vydání. Praha: Grada Publishing, 2015. ISBN 978-80-247-5457-4
[37] LABERGE, R. Datové sklady - Agilní metody a business intelligence. Praha: Computer Press, Albatros Media a.s. 2016. ISBN 978-80-251-3729-1

1) Existuje několik přístupů k integraci dat. K vytváření a údržbě datových skladů se používají tzv. ETL nástroje - nástroje pro extrakci a transformaci dat, tedy jejich uvedení do požadovaného formátu, jejich zpracování podle určitých pravidel, kombinování s jinými daty, jakož i pro načítání dat, zápis dat do úložiště nebo na jinou základnu.

Kromě ETL systémy BI obsahují nástroje SQL, které uživatelům umožňují přímý přístup k datům. V poslední době se nástroje pro generování a zpracování požadavků více zaměřují na neškolené podnikové uživatele, než na zkušené IT profesionály.

2) Analýza dat.

Pro komplexní analýzu dat v moderních systémech BI se používají nástroje OLAP. Umožňují zvažovat různé části dat, včetně časových, které umožňují identifikovat různé trendy a závislosti (podle regionu, produktu, zákazníka). K prezentaci dat se používají různé grafické nástroje: sestavy, grafy, přizpůsobitelné pomocí parametrů.

Mezi nejpokročilejší BI řešení patří bloky pro dolování dat. Tyto nástroje jsou navrženy tak, aby pomohly při identifikaci skrytých vzorců, modelů a vytváření prognóz. Jsou založeny na skenování a statistickém zpracování obrovského množství dat a jsou navrženy tak, aby usnadnily přijímání správných a informovaných strategických rozhodnutí, prostřednictvím analýzy různých scénářů. Jako nástroje se používají neuronové sítě a rozhodovací stromy.[38]

3) Panely a mapy.

Běžným nástrojem vizualizace dat v moderních BI řešeních jsou dashboardy, které zobrazují výsledky v podobě škál a indikátorů, které umožňují kontrolovat aktuální hodnoty vybraných indikátorů, porovnávat je s kritickými hodnotami a identifikovat tak potenciální hrozby pro podnikání. Ovládací panely, stejně jako skórovací karty, jsou založeny na analýze klíčových ukazatelů výkonnosti. Zpravidla však ovládací panely zobrazují aktuální stav obecných ukazatelů a bodovací karty jsou navrženy tak, aby porovnávaly aktuální ukazatele s plánovanými a zobrazovaly dynamiku změn těchto ukazatelů v čase.[39]

[38] POUR, J. a kol. Self service business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. Praha: Grada Publishing, 2018. ISBN 978-80-271-0616-5
[39] TYRYCHTR, J. Business Intelligence. Praha: ČZU v Praze, 2014. ISBN 978-80-213-2516-6.

3.4.2 Podniková analytika

Systémy podnikové analýzy řeší velmi širokou škálu úloh. Mezi nejdůležitější úkoly tedy patří sledování, analýza a úprava provozních cílů, které poskytují následující výhody:[40]

- podpora rozvoje podnikových procesů a strukturálních změn podniku;
- možnost modelování různých obchodních situací v jediném informačním prostředí;
- provádění operativní analýzy nestandardních požadavků;
- snížení rutinní pracovní zátěže zaměstnanců a uvolnění času pro hlubší analytickou práci;
- stabilní provoz s nárůstem objemu zpracovávaných informací, možnost škálování.

Pokud jde o podporu strategického rozvoje podniku, systémy BI poskytují:

- hodnocení efektivity různých oblastí podnikání;
- posouzení dosažitelnosti stanovených cílů;
- posouzení účinnosti využívání zdrojů, a to i ze strany dceřiných společností;
- hodnocení efektivnosti provozní, investiční a finanční činnosti;
- obchodní modelování a hodnocení investičních projektů;
- řízení nákladů, daňové plánování, plánování kapitálových investic.

Podle analytiků z MiPro Consulting přináší zavedení nezávislého BI systému v organizaci řadu výhod, oproti používání analytických nástrojů, zabudovaných do jiných podnikových informačních systémů. Některé z výhod systému BI zahrnují:

- větší viditelnost a pohodlí při práci s informacemi pro podnikové uživatele;
- možnost využití několika analytických řešení pro různé činnosti v měřítku celého podniku, nikoli v rámci jednotlivých oddělení;
- umožňuje extrahovat, analyzovat a konsolidovat data prakticky z jakéhokoli zdroje;
- založené na průmyslové, podporované a vyvíjené platformě BI;
- poskytuje potřebnou škálovatelnost, efektivitu, výkon;
- umožňuje budovat a udržovat end-to-end postupy a procesy zpracování, jednotné centralizované analytické modely a projekty v celé organizaci;

[40] SLÁNSKÝ, D. Data a analytika pro 21. století. Praha: Professional Publishing, 2018. ISBN 978-80-88260-25-7.

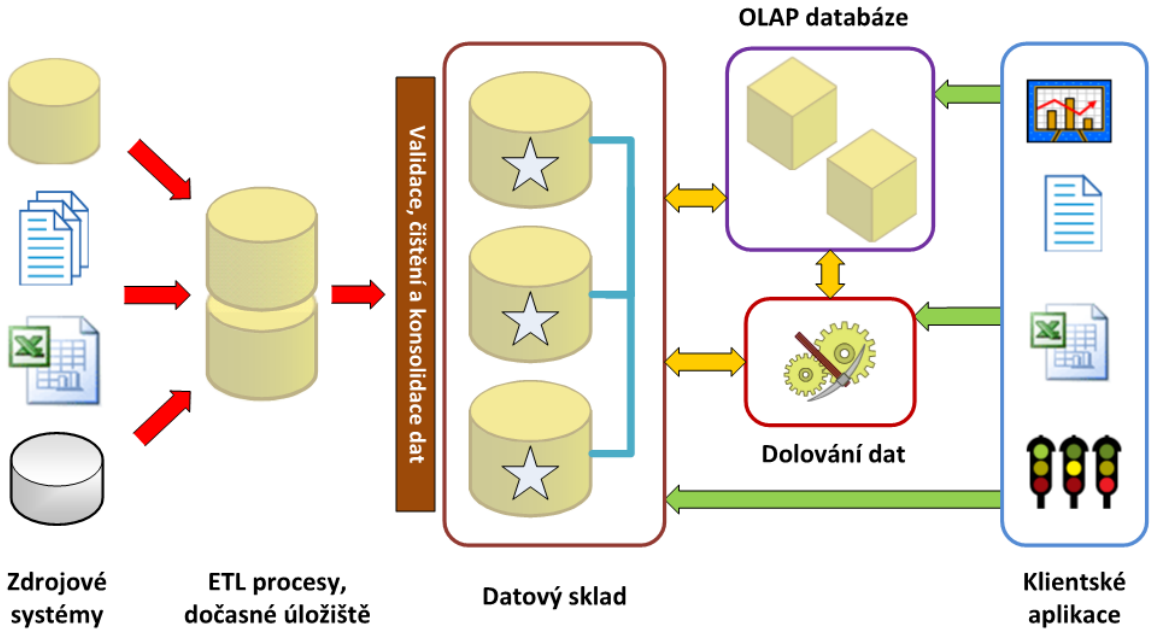
- obsahuje vestavěné nástroje pro řešení různých a různorodých analytických úloh jak z pohledu obchodního, tak z pohledu informačních technologií;
- Poskytuje přístup k datům a analytickým nástrojům pro více uživatelů.

Použití analytických nástrojů zabudovaných do jiných podnikových informačních systémů, například třídy ERP nebo CRM, má zpravidla následující omezení:[41]

- omezený soubor implementovaných analytických nástrojů, které jsou stejné pro všechny uživatele bez ohledu na jejich role a úkoly;
- možnost využívat k analýze pouze interní data, zatímco informace z jiných systémů zůstávají nepřístupné;
- nedostatek vyvinutých vestavěných nástrojů pro analýzu vede k tomu, že systém slouží pouze k extrakci dat v něm uložených, která jsou následně exportována a analyzována v Excelu;
- ERP a CRM systémy mají zpravidla omezený počet uživatelů, což „odřízne“ velké množství zaměstnanců společnosti od analytiků, pro které by tyto informace byly užitečné a zajímavé (výrazný nárůst počtu uživatelů snižuje výkon transakčních systémů);
- transakční systémy obvykle neobsahují všechny ukazatele potřebné pro analýzu, např. informační panely;
- omezená možnost vytváření flexibilních uživatelských požadavků;
- použití velkých objemů nashromážděných historických informací je omezené.

[41] TYRYCHTR, J. Business Intelligence. Praha: ČZU v Praze, 2014. ISBN 978-80-213-2516-6.

Při implementaci technologie BI v podniku je prvním úkolem zorganizovat jeden informační prostor, jedinou databázi. Úložiště Business Intelligence může přijímat informace z jakýchkoli dostupných zdrojů, včetně externích zákazníků, prostřednictvím firemního webového portálu společnosti (Obr. 2).



Obr. 2 – Schéma budování informačního systému založeného na technologii BI

Reporting je jen malá část schopností systémů vybudovaných na bázi BI technologie, která umožňuje postupné rozšiřování systému na plnohodnotný analytický nástroj, spíše než zavádění nových nezávislých programů pro řešení jednotlivých úloh analýzy činností a řízení podniku.

Hlavním rizikem při používání systémů BI je, že technologie business intelligence se mění příliš rychle a je třeba je sledovat. Další riziko souvisí s kvalitou dat: pokud nebudou správně transformována, vyčištěna a konsolidována, pak žádné nástroje nebo aplikace BI nebudou schopny zvýšit spolehlivost dat. Samotná technologie BI není schopna tyto problémy komplexně řešit a jejich zanedbávání se vrací k informační anarchii.[42]

[42] GÁLA, L., POUR, J., ŠEDIVÁ, Z. Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi. 3., aktualizované vydání. Praha: Grada Publishing, 2015. ISBN 978-80-247-5457-4

3.4.4 Specifika použití Business Intelligence v podnicích

Systémy třídy BI poskytují podporu pro analýzu relevantních informací pro společnost agregovaných z databází různých formátů a zlepšují rozhodovací proces tím, že prezentují informace v různých sekcích a vhodnou formou.

Abychom pochopili, proč jsou nástroje BI nyní tak žádané velkými a středními společnostmi, měli bychom zvážit nedostatky tradičních informačních prostředí.

První nevýhodou je poskytování takových informací o stavu podnikání, které nelze použít k určení hybatelů změny ukazatelů, faktorů ovlivňujících podnikání a také nemožnost řešit otázky strategického rozvoje firmy bez dalších analytiků. To vede k druhému nedostatku – nedostatečné integraci potřebných dat ze všech zdrojů a navázání spojení mezi nimi. Třetí nevýhodou je nízká rychlost reportingu, z důvodu práce s velkými daty a nízká interaktivita při komplexní analýze dat. Nevýhodou konvenčních systémů na operační a taktické úrovni jsou také omezené možnosti vizualizace dat a nutnost využití práce IT specialisty k úpravám stávajících formulářů výkazů, nebo k vytvoření nového formuláře výkazu.[43]

Shrneme-li to, můžeme zdůraznit následující požadavky na systémy BI, které řeší výše uvedené problémy:

1. Vysoká rychlost zpracování a výstupu velkých dat.

2. Nezávislá konfigurace systému podnikovými uživateli: možnost použití metody Drag & Drop, možnost flexibilně definovat dimenze, hierarchie, skupiny a různé datové sady.

3. Bohatá funkčnost pro vizualizaci dat. Kromě vytváření různých nástrojů pro vytváření grafů, dashboardů a dalších vlastních objektů pro analytiku to znamená možnost interakce mezi daty a jejich grafickou reprezentací a naopak. To znamená, že je realizován vysoký stupeň interaktivity mezi uživateli a daty.

4. Včasnost a relevance aktualizace dat a zpráv.

5. Možnost integrace dat z různých zdrojů, dostupnost API pro připojení k dalším službám a webové připojení.

[43] LABERGE, R. Datové sklady - Agilní metody a business intelligence. Praha: Computer Press, Albatros Media a.s. 2016. ISBN 978-80-251-3729-1

Je třeba zdůraznit, že pro maximalizaci návratnosti zavádění BI technologií potřebují společnosti mít systémy pro měření podnikových procesů, protože bez znalosti ukazatelů, kterými by se management měl při rozhodování řídit, není možné provádět odhady, prognózy a vypracování efektivní strategie rozvoje společnosti.[44]

3.5 Představení vybraných metodik pro řešení datové kvality

Výběr konkrétní aplikace nebo nástroje závisí na potřebách organizace, velikosti datového skladu a technologickém prostředí. V našem případě bylo rozhodnuto použití Pythonu pro ověření datové kvality v MySQL databázi. Zde je několik důvodů, proč jsme preferovali Python:

Flexibilita a přizpůsobitelnost: Python je obecný programovací jazyk s bohatými knihovnami pro analýzu dat, což znamená, že můžete psát vlastní skripty a provádět zcela vlastní kontrolu kvality dat. Tím získáte větší kontrolu nad procesem a můžete ho přizpůsobit konkrétním potřebám organizace.

Zdarma a open-source: Python je zdarma a open-source jazyk, což znamená, že nemusíte platit za licenci. Tím ušetříme náklady, které by jinak byly spojeny s komerčními nástroji a aplikacemi.

Široká komunita a ekosystém: Python má rozsáhlou komunitu uživatelů a bohatý ekosystém knihoven pro práci s daty, což znamená, že máte přístup k mnoha nástrojům a knihovnám pro analýzu, vizualizaci a čištění dat.

Rychlý vývoj a prototypování: Python je vhodný pro rychlý vývoj a prototypování. Můžete rychle vytvářet a testovat skripty a scénáře pro kontrolu kvality dat.

[44] POUR, J. a kol. Self service business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. Praha: Grada Publishing, 2018. ISBN 978-80-271-0616-5

Integrace s existujícími systémy: Python lze snadno integrovat s dalšími systémy a nástroji. Můžete například provádět kontrolu kvality dat a následně importovat data do jiných aplikací.

Pro vytvoření návrhu řešení, které povede k zlepšení datové kvality v podnikové praxi v našem případě, je nutná znalost jazyka SQL a programovacího jazyka Python. Tyto klíčové komponenty jsou nezbytné pro dosažení hlavního cíle této diplomové práce.

3.5.1 SQL

SQL (Structured Query Language) je speciální programovací jazyk, který se používá pro práci s relačními databázemi. Jeho hlavním účelem je manipulace s daty uloženými v databázových systémech. SQL umožňuje uživatelům definovat, dotazovat se, aktualizovat a spravovat data v databázích.

Základní operace, které SQL umožňuje provádět:

SELECT: Používá se pro vyhledávání dat z databáze. Pomocí SELECT lze získat určité sloupce z tabulky, provádět filtry, řadit data a provádět agregační operace, jako je například SUM, AVG, COUNT, atd.

INSERT: Slouží k vkládání nových záznamů (řádků) do databáze.

UPDATE: Umožňuje aktualizovat existující záznamy v databázi.

DELETE: Používá se pro odstranění záznamů z databáze.

CREATE: Slouží k vytváření nových tabulek, databází a dalších objektů v databázovém systému.

ALTER: Umožňuje měnit strukturu existujících objektů, jako jsou tabulky, a provádět další změny v databázovém schématu.

DROP: Slouží k odstraňování tabulek, databází a dalších objektů z databáze.

SQL je standardizovaný jazyk, což znamená, že existuje několik verzí standardu SQL (například SQL-92, SQL:1999, SQL:2003), ale základní syntaxe a funkcionality jsou většinou podobné ve všech relačních databázových systémech. To umožňuje, aby byly dovednosti v SQL přenositelné mezi různými databázovými platformami.

SQL je široce používán v oblasti správy a analýzy dat, vývoje webových aplikací, business intelligence a mnoha dalších oblastech, kde je potřeba efektivně pracovat s daty uloženými v databázích.

3.5.2 SQL Funkce a procedury

Funkce a procedury jsou klíčovými konstrukcemi v jazyce SQL, které umožňují vývojářům a databázovým administrátorům automatizovat, znovupoužívat a centralizovat logiku databázových operací. Tyto konstrukce přinášejí mnoho výhod při práci s daty a správě databází. Zde se zaměříme na SQL funkce a procedury a jejich význam.

SQL Funkce

SQL funkce jsou bloky kódu, které přijímají vstupy, provádějí určité operace a vracejí výsledky. Funkce mohou být vestavěné nebo definované uživatelem.

Základní příklady SQL funkcí zahrnují:

SUM(): Tato funkce slouží k součtu hodnot ve sloupci.

AVG(): Spočítá průměr hodnot ve sloupci.

COUNT(): Určí počet řádků v tabulce nebo počet hodnot ve sloupci.

CONCAT(): Slouží k spojení textových řetězců.

DATE(): Konvertuje textový řetězec na datum.

UPPER() / LOWER(): Mění písmena textového řetězce na velká nebo malá.

IFNULL() / COALESCE(): Nahrazuje NULL hodnoty určitým výchozím hodnotami.

USER(): Vrací jméno aktuálního uživatele v databázi.

RAND(): Generuje náhodné číslo.

SQL funkce lze používat ve SELECT klauzuli, WHERE klauzuli a v rámci výrazů pro manipulaci s daty.

SQL Procedury

SQL procedury jsou soubory kódů, které mohou obsahovat více SQL příkazů a jsou uloženy v databázovém systému. Procedury mohou přijímat vstupy, provádět operace nad daty a mohou být volány na vyžádání.

Hlavní výhody SQL procedur zahrnují:

Centralizace kódu: Kód procedur je uložen a spravován v databázi, což zjednodušuje jeho správu a aktualizace.

Opakované použití: Procedury mohou být volány z různých částí aplikace, což umožňuje znovupoužití logiky operací.

Bezpečnost a řízení přístupu: Procedury mohou být přiřazeny uživatelským oprávněním, což umožňuje řídit přístup k datům.

Transakce: Procedury lze provádět v rámci transakcí, což zajišťuje konzistenci dat.

Optimalizace výkonu: Procedury mohou být optimalizovány pro rychlé provedení operací nad daty.

Toto jsou základní příklady SQL funkcí a procedur, se kterými je důležité se seznámit před návrhem a implementací samotného řešení.

3.5.3 Python

Python je vysokoúrovňový programovací jazyk, který je široce používán pro různé účely, včetně analýzy dat a správy dat. Má mnoho knihoven a nástrojů, které usnadňují práci s daty, což z něj činí populární volbu pro kontrolu kvality dat v databázích, včetně databáze MySQL.

S cílem poskytnout čtenářům základní povědomí o klíčových termínech a konceptech spojených s používáním programovacího jazyka Python v kontextu práce s databázemi, je třeba provést pohled na některé klíčové prvky. Python, jako mnohostranný a rozšířený programovací jazyk, má široké uplatnění v práci s různými typy databází a datových úložišť.

Základním pojmem, který je nutné zdůraznit, je SQL (Structured Query Language), což je standardní jazyk pro správu a manipulaci s relačními databázemi. Python nabízí několik knihoven, jako například "sqlite3," "psycopg2" pro PostgreSQL nebo "mysql-connector-python" pro MySQL, které umožňují propojení s databází a provádění SQL dotazů z Pythonového kódu.

Důležitým konceptem v práci s databázemi v Pythonu je ORM (Object-Relational Mapping), což je technika mapování dat mezi objekty v programu a tabulkami v databázi. Toto mapování usnadňuje práci s daty a zjednodušuje vytváření, čtení, aktualizaci a mazání záznamů v databázi.

V rámci práce s databázemi je nutné mít povědomí o přístupových údajích, jako jsou přihlašovací údaje (uživatelské jméno a heslo) k databázím, které jsou zabezpečené heslem. Důležitý je také koncept transakcí, které umožňují zaručit konzistenci dat v databázi při provádění více operací současně.

V rámci práce s Pythonem a databázemi je klíčové porozumět způsobu připojení k databázi, provádění dotazů, zpracování výsledků a správa chyb. K tomu lze využít vestavěné moduly pro práci s databázemi v Pythonu.

Celkově lze říci, že Python v kontextu práce s databázemi nabízí široké možnosti a nástroje, které umožňují efektivní manipulaci s daty a zajištění bezpečného a spolehlivého přístupu k datovým zdrojům.

Toto jsou základní termíny a koncepty představují základní pojmy, které čtenáři umožňují lépe porozumět této problematice.

4 Vlastní práce

V praktické části diplomové práce se zaměřuji na řešení komplexního problému týkajícího se datové kvality. Tento výzkum vychází z konkrétních obchodních požadavků a potřeb společnosti.

Začněme tím, že zdůrazníme, proč je tato problematika tak důležitá pro naši společnost. Jedním z hlavních faktorů je neustálý růst společnosti. Počet interních zaměstnanců překročil hranici 250 lidí, což nám ukázalo nutnost zvýšit datovou kvalitu pro nově příchozí zaměstnance i ty, kteří jsou již dlouhou dobu.

Je třeba zdůraznit, že naše snaha o zachování anonymity společnosti je důsledkem ochrany obchodních tajemství, strategií a citlivých informací před konkurencí. Ve světě, kde konkurence nikdy nespí, se snažíme minimalizovat riziko, že by naše oborové informace byly veřejně dostupné. To by mohlo ovlivnit náš konkurenční postavení a strategii.

Naše hlavní technika pro zlepšení kvality dat spočívá v napsání Python aplikace, která před nahráváním dat do datového skladu provede všechny potřebné typy kontrol nad daty. Pomocí této aplikace, která bude podrobně popsána v následujících kapitolách, jsme schopni identifikovat a odstranit potenciální chyby, které by mohly ohrozit spolehlivost a použitelnost našich dat.

4.1 Návrh řešení pro zlepšení datové kvality v podnikové praxi

V této kapitole je návrh samotného řešení pro zlepšení datové kvality v podnikové praxi. Zaměřili jsme se na problém kvality dat v datové sadě "Zaměstnanci", která uložena v databázi. Nejprve jsme provedli charakterizaci vybrané datové sady. Dále jsme představili obchodní požadavky na kvalitu dat pro tuto konkrétní datovou sadu. Následně jsme se věnovali popisu aktuálního stavu zpracování dat, abychom detailněji specifikovali, kde a jak byly prováděny kontroly datové kvality.

Samotná implementace řešení na základě představených požadavků spočívala v návrhu aplikace napsané v programovacím jazyce Python. Tato aplikace měla za úkol identifikovat všechny problémy na základě obchodních požadavků a následně předložit řešení pro opravu nalezených problémů během kontroly datové kvality ve vybraném datasetu.

Při přípravě aplikace pro kontrolu datové kvality ve vybraném datasetu využita knihovna Jupyter Notebook, která poskytuje prostředí pro vytváření a spouštění skriptů.

Na základě výstupu z této aplikace budeme schopni navrhnout opatření a řešení pro odstranění nalezených chyb a problémů. Tímto způsobem zajistíme, že data budou odpovídat obchodním požadavkům a budou v optimálním stavu pro následné analytické a reportovací účely.

4.1.1 Charakteristika datasetu

Tento dataset představuje detailní informace o zaměstnancích společnosti, které jsou klíčovými údaji pro správu lidských zdrojů a plánování personálního obsazení. Zde je podrobnější charakteristika jednotlivých atributů v tomto datasetu:

Zaměstnanecké číslo: Každý zaměstnanec má unikátní identifikátor v podobě zaměstnaneckého čísla. Toto číslo používa k rychlé identifikaci konkrétního zaměstnance v rámci společnosti.

Křestní jméno a Příjmení: Jméno a příjmení zaměstnance slouží k identifikaci zaměstnance a komunikaci s ním.

Pohlaví: Informace o pohlaví (Muž/Žena) pomáhá při sledování rovnosti pohlaví v pracovním prostředí a při plánování rozvoje zaměstnanců.

Datum narození: Datum narození zaměstnance využito pro výpočet věku zaměstnance a pro plánování narozeninových oslav.

Oddělení: Zaměstnanci jsou rozděleni do různých oddělení (Prodej, IT, Marketing, Personál, Finance). Tato informace je klíčová pro organizační strukturu společnosti.

Pozice: Pozice, kterou zaměstnanec zastává, poskytuje informace o jeho pracovním zařazení a odpovědnostech.

Plat (CZK): Měsíční plat v českých korunách (CZK) ukazuje finanční aspekt zaměstnání. Tato informace je důležitá pro správu mzdového systému a rozpočtování.

Město: Reprezentuje místo, odkud pracuje zaměstnanec. Tento sloupec obsahuje informaci o geografickém umístění, kde zaměstnanec má svou pracovní lokalitu.

Datum nástupu: Datum, kdy zaměstnanec nastoupil do společnosti, umožňuje sledovat délku zaměstnání a záznamy o nástupu nových zaměstnanců.

Email: Kontaktní emailová adresa zaměstnance slouží pro interní i externí komunikaci a distribuci informací.

Telefonní číslo: Telefonní číslo zaměstnance je dalším kontaktním bodem, který umožňuje rychlou komunikaci.

4.1.2 Popis datasetu

Následující tabulka podrobně ilustruje strukturu datové sady a její sloupce. Tato datová sada má za účelem poskytnout úplný přehled o všech záznamech, které obsahuje. Každý sloupec má svou specifickou roli a obsahuje určité informace, které jsou klíčové pro analýzu a porozumění datům.

Název Sloupce	Popis
Zamestnanecke_cislo	INT PRIMARY KEY
Krestni_jmeno	VARCHAR(50)
Prijmeni	VARCHAR(50)
Pohlavi	VARCHAR(10)
Datum_narozeni	DATE
Oddeleni_ID	INT
Pozice	VARCHAR(50)
Plat_CZK	DECIMAL(10, 2)
Datum_Nastupu	DATE
Email	VARCHAR(100)

Telefonni_cislo	VARCHAR(50)
Mesto	VARCHAR(50)
Status_ID	INT
Plat_ID	INT

Tab. 1 – Popis jednotlivých sloupců datové sady

Abychom jsme plně pochopili vztahy mezi jednotlivými sloupci v tabulce "Zamestnanci", je nezbytné prozkoumat, jak jsou vzájemně propojeny:

Zamestnanecke_cislo: Primární klíč tabulky, který jednoznačně identifikuje každého zaměstnance.

Oddeleni_ID: Cizí klíč, který odkazuje na tabulku s informacemi o odděleních. Pomocí tohoto sloupce můžeme určit, v jakém oddělení daný zaměstnanec pracuje.

Status_ID: Tento sloupec odkazuje na tabulku s definicemi stavů zaměstnanců. Pomocí něj můžeme určit, zda je zaměstnanec aktivní nebo dočasně neaktivní.

Plat_ID: Tento sloupec odkazuje na tabulku s definicemi platů zaměstnanců. Pomocí něj lze zjistit, jaký má zaměstnanec platový stupeň.

Vztahy v tabulce jsou tedy založeny na cizích klíčích, které odkazují na jiné tabulky a umožňují spojovat informace z různých tabulek dohromady pro komplexní správu zaměstnanců ve společnosti.

4.1.3 Obchodní požadavky

V této kapitole představujeme definované obchodní požadavky na správu kvality dat v tabulce 'Zaměstnanci', které byly poskytnuty společností.

Chybějící Data

Obchodní Požadavek: Všechny záznamy zaměstnanců musí obsahovat úplná data včetně "Krestni_jmeno", "Prijmeni", "Pohlavi", "Datum_narozeni", "Pozice", "Datum_nastupu", "Email", "Telefonni_cislo", "Mesto", "Plat_CZK".

Akce: Při vkládání nových záznamů do tabulky "Zamestnanci" musí systém provádět kontrolu, zda všechna povinná pole jsou vyplněna. Pokud nějaké pole chybí, systém musí zobrazit chybové hlášení a zamezit vložení chybějícíh dat.

Duplikáty

Obchodní Požadavek: Každý zaměstnanec musí byt unikátní pro jednoznačnou identifikaci.

Akce: Při vkládání nových záznamů nebo aktualizaci stávajících záznamů musí systém provádět kontrolu, zda nové zaměstnanecké číslo již v databázi neexistuje.

Pokud ano, systém musí systém musí zobrazit chybové hlášení a zamezit vložení duplikátu.

Neplatné Hodnoty Platu

Obchodní Požadavek: Platy zaměstnanců nesmí být záporné nebo nereálně vysoké.

Akce: Při vkládání nebo aktualizaci platových údajů musí systém provádět kontrolu, zda hodnoty jsou v rozumném rozmezí definovaném v rámci organizace. Pokud plat neexistuje (je None), nebo je menší než nebo rovno 0, nebo větší než 250 000 CZK systém musí upozornit na neplatnou hodnotu a nepovolit její uložení.

Neplatné Formáty Dat

Obchodní Požadavek: Všechna data, jako jsou emailové adresy, telefonní čísla, jména, příjmení, pohlaví, pozice a město, musí být ve správném formátu. Data musí odpovídat stanoveným formátovým požadavkům.

Akce: Při vkládání nebo aktualizaci dat musí systém provádět kontrolu, zda data odpovídají očekávaným formátům.

Systém musí ověřit, že emailové adresy mají platný formát, což znamená, že obsahují text následovaný symbolem "@" a následně doménovou část, která obsahuje alespoň jeden znak následovaný tečkou. Pokud data neodpovídají požadovanému formátu, systém musí upozornit na chybu a nepovolit uložení záznamu.

Odstranění diakritiky z textu: Systém musí provádět kontrolu, zda data obsahují diakritická znaménka, a pokud ano, měl by tato znaménka odstranit. Diakritika se může objevit v textu, a odstraněním těchto znaků zajišťujeme, že text je v čistém základním formátu.

Telefonní čísla musí být ve správném formátu, což znamená, že by měla obsahovat pouze číslice. Přítomnost znaků jako "+", "-", "(", ")", nebo mezery považována za neplatnou a systém musí provést jejich odstranění.

4.1.4 Popis Integrace s datovým tokem

Než začneme implementovat řešení na základě našich obchodních požadavků, je nezbytné popsat kde a jak prováděny kontroly datové kvality. Po pečlivém zvážení bylo rozhodnuto, že kontrola datové kvality bude prováděna přímo v MySQL databázi jako samostaný krok před spuštěním ETL (Extract, Transform, Load) procesu. Toto strategické rozhodnutí nám umožní zajistit, že data, která budou předávána do našeho Data Warehouse (DWH), budou již v okamžiku nahrávání disponovat vysokou kvalitou.

Tímto způsobem můžeme minimalizovat riziko přenosu nekvalitních dat do našeho DWH, což by mohlo vést k nepřesnostem a komplikacím v procesu business intelligence a analýzy. Datová kvalita je zajištěna již na počátku datového toku, což přispěje k dosažení našich obchodních cílů a zlepšení rozhodovacího procesu.

Čištění, unifikace a deduplikace dat bylo prováděno na zdrojovém systému před samotným ETL (Extract, Transform, Load) procesem, a to z několika důvodů:

Snížení zátěže ETL procesu

Čím více čistých a kvalitních dat na vstupu ETL procesu, tím efektivněji bude proces probíhat. Úpravy dat na zdrojovém systému sníží nároky na transformaci a validaci během ETL.

Zlepšení rychlosti ETL

ETL proces rychlejší, pokud nemusí provádět rozsáhlé úpravy a čištění dat. To sníží dobu, kterou trvá načítání dat do cílového úložiště.

Zajištění konzistence dat

Čištění a unifikace dat na zdrojovém systému zajisti konzistenci dat napříč různými zdroji, což usnadňuje analýzy a reportování.

Prevence problémů při integraci

Očištěná a unifikovaná data snižují riziko problémů během integrace do cílového úložiště, což může být náročné, pokud se snažíte spojit data s různou kvalitou a formátem.

Nicméně je důležité brát v úvahu několik věcí:

Ztráta původních dat

Provádění čištění dat na zdrojovém systému může vést ke ztrátě původních dat, což může být nežádoucí, zejména v situacích, kde jsou původní data potřebná pro účely jako jsou audit nebo rekonstrukce.

Zpracování chyb

Je důležité mít mechanismus pro zachycení a zpracování chyb při čištění dat. Chyby by měly být zaznamenány a spravovány tak, aby bylo možné provádět opravy a sledovat data zpětně.

Pro řešení výše uvedených nedostatků bylo rozhodnuto, že před spuštěním procesu čištění dat provedeny následující kroky:

Vytváření Zálohy Původních Dat

Před spuštěním procesu čištění dat byla vytvořena záloha původních dat.

Opravy na Kopii

Čištění dat probehlo na kopii dat, ne na původních datech. Tímto způsobem původní data zůstali nedotčena a mohou být obnoveny v případě potřeby.

Logování Chyb

Při zpracování skriptu pro odhalení chyb jsou zaznamenány všechny nalezené chyby do logovací tabulky v databázi. Chyby obsahují informace o povaze chyby, místě, kde byla nalezena, a další relevantní informace.

Opravy Chyb

Po identifikaci chyb a jejich zaznamenání do logu proces oprav byl prováděn na kopii dat, což minimalizovalo riziko nežádoucího vlivu na původní data. Tím byla zajištěna bezpečnost původních datových záznamů.

Znovu Spuštěný Proces Kontroly

Po provedení oprav byl spuštěn proces kontroly dat, který má klíčový význam pro zajištění integrity datového prostředí. Tento proces má za cíl pečlivě prověřit, zda všechny identifikované chyby byly úspěšně odstraněny a zda data nyní splňují stanovené standardy kvality a konzistence.

4.1.5 Chybějící Data

Prvním krokem, jak jsme zmínili, bylo provedení opatření na zajištění bezpečnosti a integrity dat. K tomuto účelu jsme nejprve vytvořil zálohu a kopii tabulky 'Zaměstnanci', na které budeme provádět kontrolu datové kvality. Tímto způsobem jsme zajistili, že neporušíme žádná produkční data a že máme k dispozici bezpečnou a izolovanou kopii dat pro analytické účely.

Dalším krokem v procesu zajištění kontroly datové kvality bylo založení logovací tabulky s názvem "ApplicationLog ". Tato logovací tabulka hraje klíčovou roli při sledování a veškerých chyb a problémů, které mohou být identifikovány během zpracování skriptu pro odhalení chyb. Tuto tabulku založili následujícím příkazem:

```
create table if not exists ApplicationLog
(
    LogID            int auto_increment
        primary key,
    Timestamp        timestamp default CURRENT_TIMESTAMP not null,
    EventType        enum ('Success', 'Failure')         not null,
    EventDescription text                                null
);
```

Princip fungování této logovací tabulky je takový, že veškeré nalezené chyby jsou systematicky zaznamenávány do této tabulky v databázi. Každý záznam v logovací tabulce obsahuje důležité informace týkající se chyby, včetně povahy chyby, místa, kde byla nalezena, data a času, kdy byla identifikována. Tímto způsobem máme k dispozici důležité údaje, které nám umožňují sledovat historii a vývoj chyb v našem datovém prostředí.

Zaznamenávání chyb v logovací tabulce nám umožňuje provádět detailní analýzu a reportování o kvalitě dat v čase. To je důležité pro sledování trendů a opatření ke zlepšení kvality dat. Dále nám tento přístup umožňuje rychlou reakci na identifikované problémy a jejich okamžité řešení, což přispívá k udržení kvality datového prostředí.

Nyní přicházím k samotnému návrhu aplikace pro kontrolu datové kvality na základě výše uvedených požadavků.

Pro splnění obchodního požadavku je nutné prověřit, zda všechny záznamy zaměstnanců obsahují úplná data. Pokud nějaké pole chybí, systém musí zobrazit chybové hlášení a zamezit vložení dat chybějícíh dat. Konkrétně se kontrolují následující atributy: "Krestni_jmeno", "Prijmeni", "Pohlavi", "Datum_narozeni", "Pozice", "Datum_nastupu", "Email", "Telefonni_cislo", "Mesto", "Plat_CZK".

Návrh řešení pro kontrolu chybějících dat

Níže je představen skript, jenž byl vyvinut s primárním záměrem zabezpečení kvality dat v tabulce "Zaměstnanci" a vykonává validaci prázdných hodnot.

```
import mysql.connector
from mysql.connector import Error

def connect_to_database():
    db_config = {
        'host': 'gsqdev.kul-dc.com',
        'user': 'dfdbadmin',
        'password': '******',
        'database': 'datafactory_gsd_db'
    }

    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Chyba při připojování k databázi: {e}")
    return None

def log_application_event(event_type, event_description):
    connection = connect_to_database()
    if connection is not None:
        insert_query = "INSERT INTO ApplicationLog (EventType,
EventDescription) VALUES (%s, %s)"

        try:
            cursor = connection.cursor()
            cursor.execute(insert_query, (event_type, event_description))
            connection.commit()
        except Error as e:
            print(f"Chyba při logování události: {e}")
        finally:
            if connection.is_connected():
                cursor.close()
                connection.close()

def check_and_print_empty_values():
    connection = connect_to_database()
    if connection is not None:


        select_query = "SELECT Krestni_jmeno, Prijmeni, Pohlavi,
Datum_narozeni, Pozice, Datum_nastupu, Email, Telefonni_cislo, Mesto,
Plat_CZK FROM Zamestnanci"

        try:
            cursor = connection.cursor(dictionary=True)
            cursor.execute(select_query)
            error_count = 0  # Inicializace počtu nalezených chyb
```

```
for row in cursor.fetchall():
                has_error = False  # Proměnná pro indikaci nalezené chyby
v jednom záznamu

                for field, value in row.items():
                    if value is None or value == '':
                        if not has_error:
                            log_application_event('Failure', f'Záznam s
chybějící hodnotou v poli "{field}": {row}')
                            has_error = True
                            error_count += 1  # Inkrementace počtu chyb
            # Pokud byly nalezeny chyby, spadneme s chybovou hláškou
            if error_count > 0:
                log_application_event('Failure', f"Nalezeno celkem
{error_count} chyb.")
                raise SystemExit(1)
            else:
                log_application_event('Success', "Všechny záznamy
obsahují kompletní data")

        except Error as e:
            log_application_event('Failure', f"Chyba při provádění
kontroly prázdných hodnot: {e}")
        finally:
            if connection.is_connected():
                cursor.close()
                connection.close()

if __name__ == '__main__':
    check_and_print_empty_values()
```

Pro získání kompletního pochopení fungování skriptu jsou prezentovány podrobné popisy všech klíčových funkcí, které jsou použity v tomto navrženém skriptu, spolu s důležitými aspekty jeho vykonávání.

Připojení k databázi: Na začátku skriptu je definována funkce "connect_to_database", která slouží k připojení k databázi. Tato funkce využívá konfigurační údaje a pokusí se navázat spojení. V případě úspěchu vrací objekt připojení, v opačném případě vypíše chybovou zprávu.

Zkontrolování chybějících hodnot: Hlavní funkcí je "check_and_print_empty_values", která využívá připojení k databázi k provádění SQL dotazů.

Tato funkce vykonává následující kroky:

- Sestaví SQL dotaz pro získání všech dat z tabulky "Zamestnanci".
- Prochází získané řádky a pole v těchto řádcích.

- Pokud nalezne prázdné nebo "None" hodnoty v datech (což indikuje chybějící data), použije funkci "log_application_event" k zaznamenání této chyby do tabulky "ApplicationLog". Tato tabulka obsahuje informace o chybách.
- Pokud byly nalezeny chyby, skript vypíše počet nalezených chyb a ukončí se s chybovým kódem (exit code 1).

Záznam úspěšnosti nebo neúspěšnosti: V případě, že byly prováděny kontroly a nebyly nalezeny žádné chyby, skript vypíše zprávu "Všechny záznamy obsahují kompletní data." Tímto způsobem je možné sledovat úspěšnost běhu skriptu.

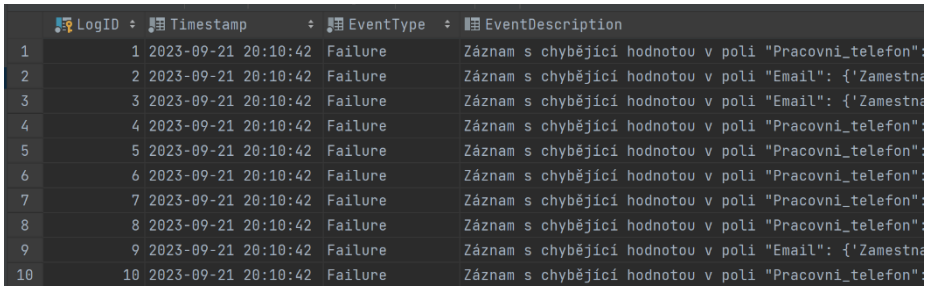
Zaznamenání úspěšnosti nebo neúspěšnosti: Funkce `log_application_event` se používá k zaznamenání různých událostí, jako jsou chyby nebo úspěchy, do tabulky "ApplicationLog". Tato tabulka slouží k monitorování a auditování chování skriptu a sledování, jak často dochází k neúspěšným událostem.

Charakteristika

Po vykonání skriptu bylo detekováne celkove 426 chyb v datech. Zajímavou pozorování je, že většina těchto chyb byla nalezena v poli "Email" a také v poli "Telefonni_cislo". To naznačuje, že chyby souvisejí s chybějícími nebo neplatnými hodnotami v těchto dvou konkrétních polích. Je však třeba poznamenat, že i v ostatních polích byly nalezeny chybějící hodnoty, avšak výrazně v menším množství.

To ukazuje, že v tabulce 'Zamestnanci' je třeba provést analýzu a zavést pro každý datový atribut pravidla validace, návrh řešení uveden v dalším kroku. Tímto krokem jsme schopni v snížit počet neplatných a chybějících hodnot v tabulce a zlepšit celkovou kvalitu dat.

Všechny zjištěné chybějící údaje v tabulce "Zaměstnanci" byly pečlivě zdokumentovány v logovací tabulce (Obr. 3), což poskytuje možnost sledovat a analyzovat všechny identifikované nedostatky. Za účelem monitorování a řízení těchto chyb jsme vytvořili přílohu ve formátu souboru CSV, která obsahuje detailní informace o všech identifikovaných chybách.



	LogID	Timestamp	EventType	EventDescription
1	1	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Pracovni_telefon"
2	2	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Email": {'Zamestna
3	3	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Email": {'Zamestna
4	4	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Pracovni_telefon"
5	5	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Pracovni_telefon"
6	6	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Pracovni_telefon"
7	7	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Pracovni_telefon"
8	8	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Pracovni_telefon"
9	9	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Email": {'Zamestna
10	10	2023-09-21 20:10:42	Failure	Záznam s chybějící hodnotou v poli "Pracovni_telefon"

Obr. 3 – Výsledky kontroly chybějících dat

Navržené řešení pro odhalení chybějících hodnot lze aplikovat jako běžný postup pro kontrolu a zajištění kvality dat ve firmě. Toto řešení poskytuje efektivní nástroje pro identifikaci a správu chybějících dat v tabulce "Zamestnanci".

Jednou z výhod tohoto navrženého řešení je, že lze snadno integrovat do stávajícího datového toku a procesů správy dat. Společnost může pravidelně používat tento nástroj k monitorování dat a provádění kontroly datové kvality. To umožní identifikaci a řešení chybějících údajů v reálném čase.

Návrh řešení pro snížení počtu chybějících hodnot

Po detekci chybějících hodnot v tabulce "Zaměstnanci" skriptem bylo rozhodnuto o předání těchto nalezených nedostatků dodavateli pro další analýzu a korekci. Tato opatření byla zavedena za účelem zajištění, že chybějící data budou správně doplněna a kvalita dat bude obnovena v souladu s požadavky a standardy stanovenými pro tuto databázi. Spolupráce s dodavatelem byla považována za efektivní způsob, jak rychle a důkladně řešit tuto konkrétní problematiku a obnovit integritu dat.

Následným krokem byla prováděna pečlivá analýza datových atributů v tabulce "Zamestnanci" bylo zjištěno, že tabulka nemá nastavena povinná pole pro následující atributy: "Krestni_jmeno", "Prijmeni", "Pohlavi", "Datum_narozeni", "Pozice", "Datum_nastupu", "Email", "Telefonni_cislo", "Mesto", "Plat_CZK". Tato absence povinných polí vede k chybám a nedostatečné kvalitě dat.

V této souvislosti bylo navrženo, aby tato pole byla upravena a nastavena jako povinná (NOT NULL) s aplikací příslušných validačních pravidel. Tímto způsobem bylo dosaženo zajištění, že veškeré záznamy o zaměstnancích obsahují nezbytná a validní data, což přispělo ke zvýšení celkové kvality a spolehlivosti dat v tabulce "Zaměstnanci."

Navržená pravidla validace pro jednotlivé datové atributy

Krestni_jmeno: Sloupec pro křestní jméno zaměstnance. Je definován jako VARCHAR s maximální délkou 50 znaků a je povinný (NOT NULL).

Prijmeni: Sloupec pro příjmení zaměstnance. Stejně jako křestní jméno, je definován jako VARCHAR s maximální délkou 50 znaků a je povinný (NOT NULL).

Pohlavi: Sloupec, definován jako VARCHAR s maximální délkou 10 znaků a je povinný (NOT NULL).

Datum_narozeni: Sloupec pro datum narození zaměstnance ve formátu DATE. Je povinný (NOT NULL).

Pozice: Sloupec, definován jako VARCHAR s maximální délkou 50 znaků a je povinný (NOT NULL).

Datum_nastupu: Sloupec pro datum nástupu zaměstnance do práce, definován ve formátu DATE. Je povinný (NOT NULL).

Email: Sloupec pro e-mailovou adresu zaměstnance. Je povinný (NOT NULL) a validován na formát e-mailu.

Telefonni_cislo: Sloupec pro telefonní číslo zaměstnance. Je povinný (NOT NULL) a validován na platný formát telefonního čísla.

Mesto: Sloupec pro bydliště zaměstnance, povoleny jsou písmena, mezery a interpunkce. Je povinný (NOT NULL).

Plat_CZK: Sloupec pro plat zaměstnance v českých korunách (CZK), je definován jako DECIMAL s přesností na dvě desetinná místa. Je povinný (NOT NULL).

Tato pravidla validace zajisti, že data v tabulce "Zamestnanci" odpovídají specifikacím a minimalizují riziko neplatných a nekonzistentních hodnot v nových přicházejících datech.

Implementace řešení

Po obdržení chybějících hodnot ze strany dodavatele byl znovu zahájen proces kontroly nad tabulkou zaměstnanců, který tentokrát vygeneroval zprávu, že všechny záznamy obsahují úplná data. V důsledku toho byla provedena další kroky, které zahrnovaly nastavení nových pravidel pro správu tabulky zaměstnanců (Obr. 4).

```
CREATE TABLE Zamestnanci
(
    Zamestnanecke_cislo int  AUTO_INCREMENT primary key,
    Krestni_jmeno       varchar(50)     NOT NULL,
    Prijmeni            varchar(50)     NOT NULL,
    Pohlavi             varchar(10)     NOT NULL,
    Datum_narozeni      date            NOT NULL,
    Oddeleni_ID         int             NULL,
    Pozice              varchar(50)     NOT NULL,
    Datum_nastupu       date            NOT NULL,
    Email               varchar(100)    NOT NULL,
    Telefonni_cislo     varchar(50)     NOT NULL,
    Mesto               varchar(50)     NOT NULL,
    Plat_CZK            decimal(10, 2) NOT NULL,
    Status_ID           int             NULL,
    Plat_ID             int             NULL
);
```

Tímto způsobem byl vyřešen obchodní požadavek prostřednictvím navrhu skriptu na kontrolu dat a současně byla navržena nová struktura zdrojové tabulky. Tyto kroky povedou ke zlepšení celkové datové kvality a zajistí, že budoucí data budou více validní a přesná, což napomůže plnit obchodní požadavky na kvalitu dat a zajisti, že záznamy budou mít méně chyb před jejich vkladem do datového skladu.

4.1.6 Duplikáty

Pro splnění obchodního požadavku bylo nutné ověřit, zda každý zaměstnanec je unikátní. Na začátku jsme nejprve definovali kritéria pro identifikaci duplicitních záznamů. Tato kritéria nám pomáhají určit, kdy můžeme považovat dva záznamy za duplikáty. Následně jsme provedli kontrolu a odstranění těchto duplicitních záznamů, což zajišťuje jedinečnost v tabulce "Zaměstnanci".

Definována kritéria pro duplicitní záznamy

Nejprve byla provedena kontrola na základě následujících polí: Křestní_Jméno, Příjmení, Email a Telefonní_cislo, kde byly dva záznamy považovány za duplikáty, pokud zaměstnanci měli identická Křestní_Jméno, Příjmení, Email a Telefonní_cislo.

Dále byla provedena kontrola založená na e-mailových adresách, kde byli dva zaměstnanci považováni za duplikáty, pokud sdíleli stejnou e-mailovou adresu.

Následně byla prováděna kontrola na základě telefonního čísla, kde byli dva zaměstnanci považováni za duplikáty v případě, že sdíleli totožné telefonní číslo.

Tímto způsobem byly zavedeny kontrolní mechanismy, které zajistily integritu dat a minimalizovaly riziko vzniku duplicit v tabulce "Zaměstnanci".

Jako následující fázi implementace jsme navrhli a vytvořili novou logovací tabulku, která byla pojmenována " ApplicationLog_02". Tento krok byl strategicky motivován plánem odesílání výsledků z původní tabulky "ApplicationLog" našemu dodavateli. Cílem vytvoření této nové tabulky bylo zajištění oddělení dat a zabránění možnému zkřížení informací.

```
create table if not exists ApplicationLog_02
(
    LogID            int auto_increment
        primary key,
    Timestamp        timestamp default CURRENT_TIMESTAMP not null,
    EventType        enum ('Success', 'Failure')         not null,
    EventDescription text                                null
);
```

Návrh řešení pro kontrolu duplicitních záznamů

Pro identifikaci a zpracování duplikátů byl vyvinut další Python skript který má za úkol zajistit duplicitní záznamy:

```
import mysql.connector
from mysql.connector import Error

def connect_to_database():
    db_config = {
        'host': 'gsqdev.kul-dc.dhl.com',
        'user': 'dfdbadmin',
        'password': '******',
        'database': 'datafactory_gsd_db'
    }

    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Chyba při připojování k databázi: {e}")
    return None

def log_event(connection, event_type, event_description):
    insert_query = "INSERT INTO ApplicationLog_02 (EventType,
EventDescription) VALUES (%s, %s)"

    try:
        cursor = connection.cursor()
        cursor.execute(insert_query, (event_type, event_description))
        connection.commit()
    except Error as e:
        print(f"Chyba při logování události: {e}")

def check_duplicate_records(connection):
    select_query = """
    SELECT Telefonni_cislo, COUNT(*)
    FROM Zamestnanci
    GROUP BY Krestni_jmeno, Prijmeni, Email, Telefonni_cislo
    HAVING COUNT(*) > 1
    """

    try:
        cursor = connection.cursor(dictionary=True)
        cursor.execute(select_query)
        duplicate_records = cursor.fetchall()

        if duplicate_records:
            for row in duplicate_records:
                log_event(connection, 'Failure', f"Nalezen duplicitní
záznam s jménem: {row['Krestni_jmeno']} {row['Prijmeni']}, e-mailem:
{row['Email']} a telefonním číslem: {row['Telefonni_cislo']}")
            print(f"Nalezeny duplicitní záznamy:
{len(duplicate_records)}")
            raise SystemExit(1)
```

```
        else:
            log_event(connection, 'Success', "Nebyly nalezeny duplicitní
záznamy")

        return duplicate_records
    except Error as e:
        log_event(connection, 'Failure', f"Chyba při kontrole
duplicitních záznamů: {e}")
        return []

if __name__ == '__main__':
    connection = connect_to_database()
    if connection is not None:
        duplicate_records = check_duplicate_records(connection)
        connection.close()
```

Pro dosažení plného pochopení fungování skriptu jsou zde poskytnuty podrobné popisy všech klíčových funkcí, které byly použity v rámci navrženého skriptu, a to spolu s významnými aspekty jeho provádění.

Připojení k databázi: Skript začíná funkcí connect_to_database(), která zajišťuje připojení k MySQL databázi. Zde jsou specifikovány přístupové údaje, jako je hostname, uživatelské jméno, heslo a název databáze.

Logování událostí: Pro zaznamenání výsledků kontroly a dalších událostí skript používá tabulku "ApplicationLog_02". Funkce log_event() je zodpovědná za zápis těchto událostí do logu. Zde jsou ukládány typ události (úspěch nebo chyba) a popis události.

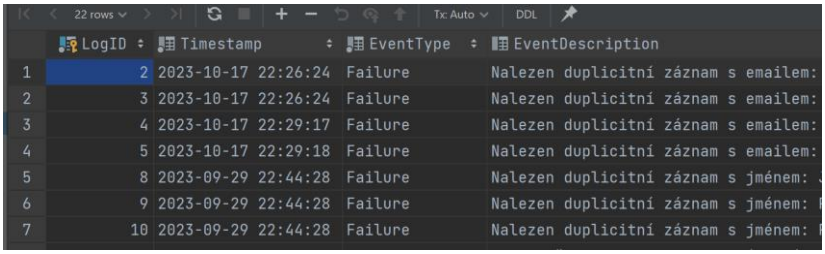
Kontrola duplicitních záznamů: Hlavní funkce check_duplicate_records(connection) provádí kontrolu na základě kritérií: "Krestni_jmeno", "Prijmeni", "Email" a "Telefonni_cislo". Skript hledá záznamy, které mají stejnou kombinaci těchto polí a počítá jejich počet. Pokud je nalezen alespoň jeden záznam, který je duplicitní, je tato událost zalogována a skript spadne s chybovým hlášením.

Hlavní část skriptu: Ve funkci if __name__ == '__main__': je spuštěna kontrola duplicitních záznamů. Skript nejprve naváže spojení s databází, provede kontrolu, a pokud najde duplicity, zaloguje je, a skript skončí s chybovým kódem. Pokud nejsou nalezeny duplicity, zaloguje, že nebyly nalezeny, a ukončí se bez chyby.

Charakteristika

V našem konkrétním případě, kdy jsme aplikovali náš skript na tabulku "Zamestnanci", bylo identifikováno celkem 242 duplicitních záznamů. Všechny tyto nalezené duplikáty byly systematicky zaznamenány v logovací tabulce "ApplicationLog_02" (Obr. 4). Komplexní souhrn všech identifikovaných chyb lze podrobně prostudovat v příloze.

Při samotném kontrolování každého pole jsme zjistili, že data obsahují ještě více duplicit. Tato situace vznikla kvůli nepřesným datům od poskytovatelů. Jeden z prvních problémů, na který jsme narazili, je, že někteří zaměstnanci mohou mít zcela různá jména a příjmení, ale sdílí stejnou e-mailovou adresu nebo telefonní číslo. Všechny tyto problémy byly vyřešeny v následujícím kroku, kdy provedeme další kontrolu a úpravy dat.



	LogID	Timestamp	EventType	EventDescription
1	2	2023-10-17 22:26:24	Failure	Nalezen duplicitní záznam s emailem:
2	3	2023-10-17 22:26:24	Failure	Nalezen duplicitní záznam s emailem:
3	4	2023-10-17 22:29:17	Failure	Nalezen duplicitní záznam s emailem:
4	5	2023-10-17 22:29:18	Failure	Nalezen duplicitní záznam s emailem:
5	8	2023-09-29 22:44:28	Failure	Nalezen duplicitní záznam s jménem:
6	9	2023-09-29 22:44:28	Failure	Nalezen duplicitní záznam s jménem:
7	10	2023-09-29 22:44:28	Failure	Nalezen duplicitní záznam s jménem:

Obr. 4 – Výsledky kontroly duplicitních záznamů

Návrh řešení pro odstranění duplicit

Navrhované řešení je takové, že pokud se duplicity objeví během kontroly všech polí zároveň, konkrétně polí "Krestni_jmeno", "Prijmeni", "Email" a "Telefonni_cislo", je deduplikovany následujícím skriptem. V případě záznamu, které sdílí stejnou e-mailovou adresu nebo telefonní číslo, tyto záznamy zaznamenány v logovací tabulce a následně odeslány zpět dodavatelům k další kontrole, analýze a případné opravě těchto záznamů. Tímto způsobem postupně zvyšujeme kvalitu dat v naší databázi a zajistíme, že obsahuje pouze unikátní a správné záznamy.

```
import mysql.connector
from mysql.connector import Error

def connect_to_database():
    db_config = {
        'host': 'gsqdev.kul-dc.dhl.com',
        'user': 'dfdbadmin',
        'password': '*******',
        'database': 'datafactory_gsd_db'
    }

    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            return connection
    except Error as e:
        log_event(connection, 'Failure', f"Chyba při připojování k
databázi: {e}")
    return None

def log_event(connection, event_type, event_description):
    insert_query = "INSERT INTO ApplicationLog_02 (EventType,
EventDescription) VALUES (%s, %s)"

    try:
        cursor = connection.cursor()
        cursor.execute(insert_query, (event_type, event_description))
        connection.commit()
    except Error as e:
        print(f"Chyba při logování události: {e}")

def remove_duplicate_records(connection):
    # Vytvoření dočasné tabulky pro unikátní záznamy
    create_temp_table = """
    CREATE TEMPORARY TABLE Temp_Zamestnanci AS
    SELECT DISTINCT Krestni_jmeno, Prijmeni, Email, Telefonni_cislo
    FROM Zamestnanci;
    """

    # Smazání původní tabulky a její nahrazení dočasnou tabulkou
    replace_original_table = """
    RENAME TABLE Temp_Zamestnanci TO Zamestnanci;
    """

    try:
        cursor = connection.cursor()
        cursor.execute(create_temp_table)
        cursor.execute(replace_original_table)
        connection.commit()
        log_event(connection, 'Success', "Odstraněny duplicity z tabulky
Zamestnanci.")
    except Error as e:
        log_event(connection, 'Failure', f"Chyba při odstraňování
duplicity: {e}")

if __name__ == '__main__':
    connection = connect_to_database()
    if connection is not None:
        remove_duplicate_records(connection)
        connection.close()
```

Abychom dosáhli úplného porozumění funkčnosti skriptu, zde jsou prezentovány detailní popisy všech klíčových funkcí, které byly integrovány do navrhovaného skriptu.

Připojení k databázi: Nejprve se skript pokusí připojit k databázi s použitím konfiguračních informací, jako jsou hostitel, uživatel a heslo. Pokud je připojení úspěšné, vrátí se platné spojení s databází. Pokud selže, zobrazí se chybová zpráva.

Vytvoření dočasné tabulky: Skript začíná tím, že vytvoří dočasnou tabulku s názvem "Temp_Zamestnanci".

Do této tabulky se zkopírují pouze unikátní záznamy z původní tabulky "Zamestnanci". Unikátnost je určena na základě kombinace kritérií, jako jsou "Krestni_jmeno", "Prijmeni", "Email` a "Telefonni_cislo". To znamená, že do dočasné tabulky budou zkopírovány pouze jedinečné kombinace těchto polí.

Smazání původní tabulky: Po vytvoření dočasné tabulky se smaže původní tabulka "Zamestnanci". To znamená, že všechny její záznamy budou odstraněny.

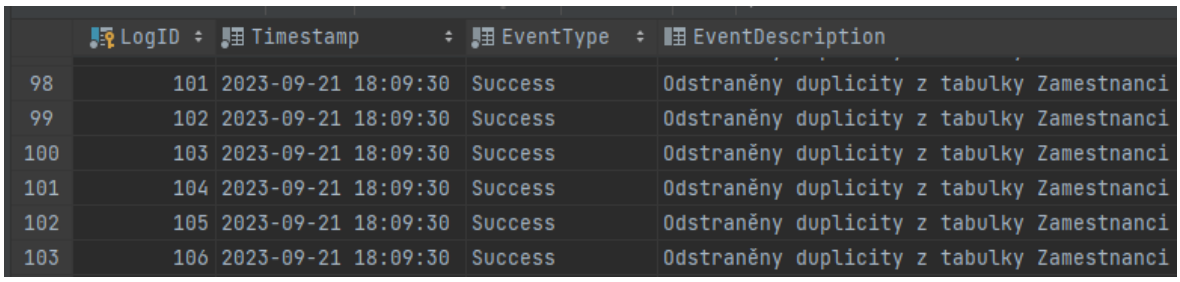
Nahrazení původní tabulky: Dočasná tabulka "Temp_Zamestnanci" se nyní přejmenuje na původní název "Zamestnanci". Tím je nahrazena původní tabulka, ale obsahuje pouze unikátní záznamy.

Logování výsledku: Po dokončení odstranění duplicity skript zaznamená výsledek (úspěch nebo selhání) do logovací tabulky "ApplicationLog_02". Pokud odstranění proběhlo úspěšně, loguje "Success", pokud došlo k chybě, loguje "Failure".

Potvrzení změn a ukončení: Po dokončení operace smazání a nahrazení původní tabulky se změny potvrdí, a spojení s databází se uzavře.

Charakteristika

Po spuštění skriptu byly duplicity odstraněny v souladu s předchozími specifikacemi, což potvrzuje úspěšné dosažení stanoveného cíle (Obr. 5).



	LogID	Timestamp	EventType	EventDescription
98	101	2023-09-21 18:09:30	Success	Odstraněny duplicity z tabulky Zamestnanci
99	102	2023-09-21 18:09:30	Success	Odstraněny duplicity z tabulky Zamestnanci
100	103	2023-09-21 18:09:30	Success	Odstraněny duplicity z tabulky Zamestnanci
101	104	2023-09-21 18:09:30	Success	Odstraněny duplicity z tabulky Zamestnanci
102	105	2023-09-21 18:09:30	Success	Odstraněny duplicity z tabulky Zamestnanci
103	106	2023-09-21 18:09:30	Success	Odstraněny duplicity z tabulky Zamestnanci

Obr. 5 – Výsledky odstranění duplicitních záznamů

4.1.6 Neplatné Hodnoty Platu

Pro splnění obchodního požadavku je nutné provádět kontrolu hodnoty platu a ověřovat, zda tato hodnota spadá do rozumného rozmezí definovaného v rámci organizace. Konkrétně jsou prováděny následující kontroly:

Kontrola existence hodnoty: Pokud plat zaměstnance neexistuje (je None), systém ho považuje za neplatný a nepovolí jeho uložení.

Kontrola minimálního platu: Pokud plat zaměstnance je menší nebo rovno 0 CZK, což je nepřijatelné, systém zaznamená tuto hodnotu jako neplatnou a nedovolí její uložení.

Kontrola maximálního platu: Pokud plat zaměstnance přesahuje 250 000 CZK, což je také mimo stanovené rozmezí, systém ho označí jako neplatný a zabrání jeho uložení.

Návrh řešení pro kontrolu neplatných hodnot platu

Návrh řešení pro kontrolu neplatných hodnot platu zahrnuje zavedení těchto kontrolních kroků při zadávání nebo aktualizaci platů zaměstnanců do systému.

Pokud během těchto procesů systém zjistí neplatnou hodnotu platu, nepovolí uložení takové hodnoty. Pro účely provádění důkladné kontroly byl zaveden další skript:

```
import mysql.connector
from mysql.connector import Error

def connect_to_database():
    db_config = {
        'host': 'gsqdev.kul-dc.dhl.com',
        'user': 'dfdbadmin',
        'password': '******',
        'database': 'datafactory_gsd_db'
    }

    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Chyba při připojování k databázi: {e}")
    return None

def log_event(connection, event_type, event_description):
    insert_query = "INSERT INTO ApplicationLog_02 (EventType,
EventDescription) VALUES (%s, %s)"

    try:
        cursor = connection.cursor()
        cursor.execute(insert_query, (event_type, event_description))
        connection.commit()
    except Error as e:
        print(f"Chyba při logování události: {e}")

def check_salary_values():
    connection = connect_to_database()
    if connection is not None:
        select_query = "SELECT Zamestnanecke_cislo, Plat_CZK FROM
Zamestnanci"

        try:
            cursor = connection.cursor(dictionary=True)
            cursor.execute(select_query)
            for row in cursor.fetchall():
                zamestnanecke_cislo = row['Zamestnanecke_cislo']
                plat = row['Plat_CZK']
                if plat is None:
                    typ_chyby = 'Neplatný plat'
                    popis_chyby = f'Plat pro zaměstnance s číslem
{zamestnanecke_cislo} neexistuje.'
                    log_event(connection, 'Failure', popis_chyby)
                    print(popis_chyby)
                elif plat <= 0:
                    typ_chyby = 'Neplatný plat'


popis_chyby = f'Plat pro zaměstnance s číslem {zamestnanecke_cislo} je
neplatný, protože je menší nebo roven 0 CZK ({plat} CZK).'
                    log_event(connection, 'Failure', popis_chyby)
```

```
                    print(popis_chyby)
                elif plat > 250000:
                    typ_chyby = 'Neplatný plat'
                    popis_chyby = f'Plat pro zaměstnance s číslem
{zamestnanecke_cislo} je neplatný, protože přesahuje 250 000 CZK ({plat}
CZK).'
                    log_event(connection, 'Failure', popis_chyby)
                    print(popis_chyby)
        except Error as e:
            print(f"Chyba při provádění kontroly platových údajů: {e}")
        finally:
            if connection.is_connected():
                cursor.close()
                connection.close()

if __name__ == '__main__':
    check_salary_values()
```

Tento skript se skládá z následujících kroků a provádí operace v souladu s výše uvedenými požadavky:

connect_to_database(): Tato funkce zajišťuje připojení k databázi. Je zde definována konfigurace pro připojení (server, uživatel, heslo, databáze). Pokud je připojení úspěšné, funkce vrátí otevřené spojení s databází, jinak vypíše chybu a vrátí None.

log_event(connection, event_type, event_description): Tato funkce zapisuje logovací události do tabulky "ApplicationLog_02". Přijímá tři argumenty: connection (otevřené spojení k databázi), event_type (typ události, jako "Success" nebo "Failure") a event_description (popis události). Funkce vytváří SQL dotaz pro vložení nového záznamu do logovací tabulky.

check_salary_values(): Tato hlavní funkce provádí kontrolu platů zaměstnanců. Nejprve se připojí k databázi pomocí connect_to_database(). Poté provede SQL dotaz, který vybere všechny zaměstnance a jejich platy z tabulky Zamestnanci.

Pro každého zaměstnance zkontroluje plat: Pokud plat neexistuje (je None), funkce vytvoří logovací záznam označený jako "Neplatný plat" a s popisem, který upozorňuje na neexistující plat.

Pokud je plat menší nebo roven 0 CZK, vytvoří se logovací záznam s označením "Neplatný plat" a popisem, který upozorňuje na neplatnou nízkou hodnotu platů.

Pokud plat přesahuje 250 000 CZK, vytvoří se logovací záznam s označením "Neplatný plat" a popisem, který upozorňuje na neplatně vysokou hodnotu platů.

Pokud dojde k chybě při provádění kontroly, skript vypíše chybovou zprávu a nepovolí další kroky.

Charakteristika

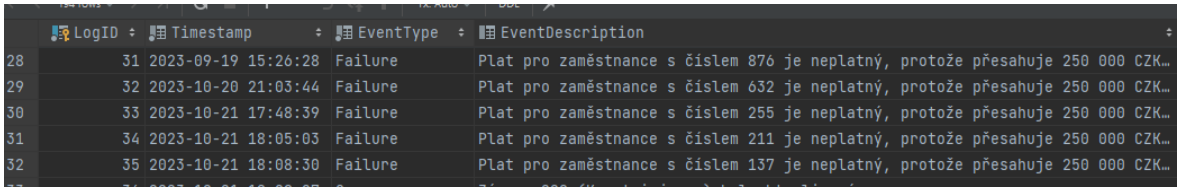
Po spuštění skriptu nad tabulkou "Zaměstnanci" a provádění kontroly hodnoty platu bylo odhaleno následující:

Chybějící platby: V našem konkrétním případě nebyly zjištěny žádné chybějící platby v důsledku kroků, které jsme provedli v předchozím procesu, konkrétně při kontrole chybějících dat.

Neplatný plat: V 78 případech byl nalezen plat zaměstnance, který byl nižší než 0 CZK. Toto je důsledkem špatných výpočtů platů nebo chyb při nahrávání dat. Zaměstnanci mají platové záznamy, ale tyto záznamy obsahují záporné částky, což je nepřijatelné.

Příliš vysoký plat: V 54 případech byly nalezeny platy zaměstnanců, které výrazně přesahovaly obvyklé hodnoty a byly nereálně vysoké. Toto může být způsobeno chybami při nahrávání dat nebo problémy s výpočtem. Zaměstnanci mají platy, které jsou mimořádně vysoké a nereálné.

Všechny tyto nalezené neplatné hodnoty platu byly systematicky zaznamenány v logovací tabulce "ApplicationLog_02" (Obr. 6). Komplexní souhrn všech identifikovaných chyb lze detailně prostudovat v příloze.



	LogID	Timestamp	EventType	EventDescription
28	31	2023-09-19 15:26:28	Failure	Plat pro zaměstnance s číslem 876 je neplatný, protože přesahuje 250 000 CZK…
29	32	2023-10-20 21:03:44	Failure	Plat pro zaměstnance s číslem 632 je neplatný, protože přesahuje 250 000 CZK…
30	33	2023-10-21 17:48:39	Failure	Plat pro zaměstnance s číslem 255 je neplatný, protože přesahuje 250 000 CZK…
31	34	2023-10-21 18:05:03	Failure	Plat pro zaměstnance s číslem 211 je neplatný, protože přesahuje 250 000 CZK…
32	35	2023-10-21 18:08:30	Failure	Plat pro zaměstnance s číslem 137 je neplatný, protože přesahuje 250 000 CZK…

Obr. 6 – Výsledky neplatných hodnot platů

Tyto problémy ukazují na různé nedostatky v evidenci platů zaměstnanců a mohou mít negativní dopad na řádné fungování organizace.

Návrh řešení pro odstranění neplatných hodnot platů

Při návrhu řešení bylo dohodnuto, že nalezené informace o identifikovaných chybách v datech, zejména týkajících se neplatných platových hodnot, uloženy do logovací tabulky "ApplicationLog_02" a zaslány zpět dodavateli dat. Dodavatel požádán o další analýzu a korekci problémových dat.

4.1.7 Neplatné Formáty Dat

Pro splnění obchodního požadavku je nutné prověřit a opravit, zaznamy v tabulce zamestnanci na predmet neplatnych formatu dat. Všechna data, jako jsou emailové adresy, telefonní čísla, jména, příjmení, pohlaví, pozice a město, musí být ve správném formátu podle uvedenych obchodnich pozadavku.

Na začátku jsme nejprve navrhli řešení pro kontrolu neplatných formátů dat, v souladu s obchodními požadavky, a následně jsme implementovali opravy a konverzi dat do požadovaného formátu. Toto opatření bylo zavedeno s cílem zajistit, aby všechna data v tabulce "Zaměstnanci" odpovídala požadovaným formátům a standardům.

Prvním krokem, který jsme podnikli, bylo navrhnout řešení pro odstranění diakritiky z textových dat. Diakritika zahrnuje znaky, které doplňují písmena o speciální znaky, jako jsou háčky, čárky nebo kroužky. Odstranění diakritiky má za cíl zjednodušit a normalizovat textová data.

Návrh řešení pro odstranění diakritiky

Navrhované řešení, které jsme vyvinuli, je prezentováno na (Obr. 14). Tento obrázek obsahuje detaily procesu odstranění diakritiky, včetně použitých algoritmů a postupů. Toto řešení nám umožňuje konvertovat textová data obsahující diakritiku na čistý a standardní a standardní text, což může přispět k zlepšení kvality a jednotnosti dat v našem systému.

```
import mysql.connector
from mysql.connector import Error
from unidecode import unidecode  # knihovna pro odstranění diakritiky

def connect_to_database():
    # Vaše konfigurace pro připojení k databázi
    db_config = {
        'host': 'gsqdev.kul-dc.dhl.com',
        'user': 'dfdbadmin',
        'password': '*******',
        'database': 'datafactory_gsd_db'
    }

    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Chyba při připojování k databázi: {e}")
    return None

def check_and_remove_diacritics():
    connection = connect_to_database()
    if connection is not None:
        try:
            cursor = connection.cursor()
            # Seznam polí, na kterých chcete provést odebírání diakritiky
            fields_to_normalize = ['Krestni_jmeno', 'Prijmeni',
'Pohlavi', 'Pozice', 'Mesto']

            for field in fields_to_normalize:
                select_query = f"SELECT Zamestnanecke_cislo, {field} FROM
Zamestnanci"
                cursor.execute(select_query)
                for row in cursor.fetchall():
                    zamestnanecke_cislo = row[0]
```

```
                    original_value = row[1]
                    normalized_value = unidecode(original_value)  #
Odebere diakritiku
                    if normalized_value != original_value:
                        update_query = f"UPDATE Zamestnanci SET {field} =
%s WHERE Zamestnanecke_cislo = %s"
                        cursor.execute(update_query, (normalized_value,
zamestnanecke_cislo))
                        connection.commit()

            print("Odebírání diakritiky dokončeno.")
        except Error as e:
            print(f"Chyba při odebírání diakritiky: {e}")
        finally:
            if connection.is_connected():
                cursor.close()
                connection.close()

if __name__ == '__main__':
    check_and_remove_diacritics()
```

Pro dosažení plného pochopení fungování skriptu jsou zde poskytnuty podrobné popisy všech klíčových funkcí, které byly použity v rámci navrženého skriptu:

connect_to_database(): Tato funkce slouží k připojení k databázi. Obsahuje konfigurační informace o databázi (host, uživatel, heslo, název databáze). Pokud je připojení úspěšné, vrátí otevřené připojení, jinak vrátí None.

log_event(connection, event_type, event_description): Tato funkce slouží k zaznamenání události v logovací tabulce. Přijímá tři argumenty: connection (připojení k databázi), event_type (typ události 'Success' nebo 'Failure') a event_description (popis události). Funkce vloží nový záznam do tabulky ApplicationLog_03 s časovým razítkem, typem události a popisem.

check_and_remove_diacritics(): Hlavní funkce skriptu. Připojí se k databázi a provede následující kroky:

Vytvoří seznam polí, na kterých chcete provést odebírání diakritiky (jméno, příjmení, pohlaví, pozice, město).

Pro každé pole provede následující kroky:

- Načte záznamy z tabulky Zamestnanci pro toto pole a zkontroluje, zda obsahují diakritiku.
- Pokud obsahují diakritiku, použije knihovnu unidecode, aby odstranil diakritiku a vytvořil normalizovanou hodnotu.
- Porovná původní hodnotu s normalizovanou hodnotou. Pokud jsou různé, provede aktualizaci záznamu v databázi na normalizovanou hodnotu.
- Po každé úspěšné aktualizaci záznamu zavolá funkci log_event() a zaznamená úspěšnou událost.

Nakonec funkce vypíše zprávu o dokončení odebírání diakritiky.

Skript se spustí v bloku if __name__ == '__main__':. Nejprve připojí k databázi a pak zavolá funkci check_and_remove_diacritics(). Ta provádí odebírání diakritiky na zadaných polích pro všechny záznamy v tabulce Zamestnanci. Po dokončení procesu vypíše zprávu o dokončení.

Charakteristika

V rámci implementace skriptu došlo k procesu odstranění diakritiky z určených polí v tabulce "Zamestnanci," přičemž výsledek těchto úspěšných operací byl systematicky zaznamenán v logovací tabulce "ApplicationLog_03," (Obr. 7). Kompletní výsledky této činnosti jsou dostupné v příloze této diplomové práce.

	LogID	Timestamp	EventType	EventDescription
261	261	2023-09-13 18:11:04	Success	Záznam 237 (Krestni_jmeno) byl aktualizován.
262	262	2023-09-13 18:11:04	Success	Záznam 238 (Krestni_jmeno) byl aktualizován.
263	263	2023-09-13 18:11:04	Success	Záznam 239 (Krestni_jmeno) byl aktualizován.
264	264	2023-09-13 18:11:04	Success	Záznam 240 (Krestni_jmeno) byl aktualizován.
265	265	2023-09-13 18:11:04	Success	Záznam 241 (Krestni_jmeno) byl aktualizován.
266	266	2023-09-13 18:11:04	Success	Záznam 242 (Krestni_jmeno) byl aktualizován.
267	267	2023-09-13 18:11:04	Success	Záznam 243 (Krestni_jmeno) byl aktualizován.
268	268	2023-09-13 18:11:04	Success	Záznam 244 (Krestni_jmeno) byl aktualizován.
269	269	2023-09-13 18:11:04	Success	Záznam 245 (Krestni_jmeno) byl aktualizován.
270	270	2023-09-13 18:11:04	Success	Záznam 246 (Krestni_jmeno) byl aktualizován.
271	271	2023-09-13 18:11:04	Success	Záznam 247 (Krestni_jmeno) byl aktualizován.

Obr. 7 – Výsledky odstranění diakritiky

Dalším krokem podle obchodního požadavku byla úprava telefonních čísel. To znamená, že telefonní čísla by měla obsahovat pouze číslice. Přítomnost znaků jako "+", "-", "(", ")", nebo mezery je považována za neplatnou a naše navrhované řešení provádí jejich odstranění.

Návrh řešení pro úpravu telefonních čísel

Náš navrhovaný přístup zahrnuje komplexní proces kontroly a normalizace telefonních čísel, který byl navržen tak, aby plně vyhovoval specifikacím obchodního prostředí. Cílem tohoto přístupu je zaručit konzistenci a validitu telefonních čísel, což má významný vliv na spolehlivost a kvalitu těchto dat.

```
import mysql.connector
from mysql.connector import Error
import re
from unidecode import unidecode
from datetime import datetime

def connect_to_database():
    db_config = {
        'host': 'gsqdev.kul-dc.dhl.com',
        'user': 'dfdbadmin',
        'password': '*******',
        'database': 'datafactory_gsd_db'
    }

    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Chyba při připojování k databázi: {e}")
    return None

def log_event(connection, event_type, event_description):
    try:
        cursor = connection.cursor()
        current_time = datetime.now()
        insert_query = "INSERT INTO ApplicationLog_03 (Timestamp,
EventType, EventDescription) VALUES (%s, %s, %s)"
        cursor.execute(insert_query, (current_time, event_type,
event_description))
        connection.commit()
    except Error as e:
        print(f"Chyba při logování události: {e}")

def check_and_normalize_phone_numbers():
    connection = connect_to_database()
    if connection is not None:
```

```
        try:
            cursor = connection.cursor()
            select_query = "SELECT Zamestnanecke_cislo, telefonni_cislo
FROM Zamestnanci"
            cursor.execute(select_query)
            rows = cursor.fetchall()

            for row in rows:
                zamestnanecke_cislo = row[0]
                telefonni_cislo = row[1]

                # Ověření formátu telefonního čísla a odstranění
nepovolených znaků
                cleaned_phone_number = re.sub(r'[^0-9]', '',
telefonni_cislo)

                # Aktualizace záznamu v databázi
                update_query = "UPDATE Zamestnanci SET telefonni_cislo =
%s WHERE Zamestnanecke_cislo = %s"
                cursor.execute(update_query, (cleaned_phone_number,
zamestnanecke_cislo))
                connection.commit()

                print(f"Telefonní číslo u zaměstnance
{zamestnanecke_cislo} bylo aktualizováno.")
                log_event(connection, 'Success', f"Telefonní číslo u
zaměstnance {zamestnanecke_cislo} bylo aktualizováno.")

        except Error as e:
            print(f"Chyba při kontrole a normalizaci telefonních čísel:
{e}")
            log_event(connection, 'Failure', f"Chyba při aktualizaci
telefonního čísla: {e}")
        finally:
            if connection.is_connected():
                cursor.close()
                connection.close()

if __name__ == '__main__':
    check_and_normalize_phone_numbers()
```

Abychom dosáhli úplného porozumění funkčnosti skriptu, zde jsou prezentovány detailní popisy všech klíčových funkcí, které byly integrovány do navrhovaného skriptu.

connect_to_database(): Tato funkce slouží k připojení k databázi na základě konfiguračních údajů. Měli byste upravit db_config podle svých přihlašovacích údajů pro připojení k databázi. V případě neúspěšného připojení vytiskne chybové hlášení a vrátí None.

log_event(connection, event_type, event_description): Tato funkce provádí zápis událostí do logovací tabulky.

Přijímá tři argumenty: aktivní spojení s databází (connection), typ události (event_type) a popis události (event_description). Skript používá aktuální čas pro vytvoření záznamu v logu.

check_and_normalize_phone_numbers(): Hlavní funkce skriptu, která kontroluje a normalizuje telefonní čísla zaměstnanců. Pro každé telefonní číslo provede následující kroky:

- Načte záznamy ze sloupce telefonni_cislo pro všechny zaměstnance z databáze.
- Pro každý záznam provede kontrolu formátu telefonního čísla a odstranění nepovolených znaků (znaky mimo číslice).
- Aktualizuje záznam v databázi s upraveným telefonním číslem.
- Zaznamenává, které telefonní číslo bylo aktualizováno.
- Zároveň zapisuje událost do logovací tabulky (úspěch nebo chyba) a ukládá popis aktualizace.

Charakteristika

Během implementace skriptu byl proveden proces normalizace telefonních čísel zaměstnanců, a to na základě specifikovaných polí v tabulce "Zamestnanci." Výsledky tohoto procesu byly systematicky dokumentovány v logovací tabulce "ApplicationLog_03" (Obr. 8). Úplné výsledky této aktivity jsou uvedeny v příloze této diplomové práce.

	LogID	Timestamp	EventType	EventDescription
1	814	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 220 bylo aktualizováno.
2	815	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 221 bylo aktualizováno.
3	816	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 222 bylo aktualizováno.
4	817	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 223 bylo aktualizováno.
5	818	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 224 bylo aktualizováno.
6	819	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 225 bylo aktualizováno.
7	820	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 226 bylo aktualizováno.
8	821	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 227 bylo aktualizováno.
9	822	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 228 bylo aktualizováno.
10	823	2023-09-21 19:33:44	Success	Telefonní číslo u zaměstnance 229 bylo aktualizováno.

Obr. 8 – Výsledky úpravy telefonních čísel

Posledním krokem v souladu s obchodními požadavky bylo vypracovat řešení pro kontrolu platnosti e-mailových adres. Platný formát e-mailové adresy vyžaduje, aby obsahovala text, následovaný symbolem "@", a poté doménovou část, která obsahuje alespoň jeden znak, následovaný tečkou. V případě, že data nevyhovují požadovanému formátu, systém musí identifikovat chybu a nedovolit uložení záznamu.

Navrh řešení pro kontrolu platnosti e-mailových adres

Navrhované řešení, které je níže prezentováno, zahrnuje detailní návrh procesu ověření platnosti emailových adres, který je aplikován na všechna testovací data v tabulce. Tento proces pomůže zajistit, že veškeré emailové adresy uložené v tabulce "Zaměstnanci" budou splňovat požadované standardy, a tím zvýší úroveň datové kvality v našem systému. Tím pádem budeme schopni eliminovat chyby spojené s neplatnými emailovými adresami.

```
import mysql.connector
from mysql.connector import Error
import re
import logging
import sys

# Konfigurace logování
logging.basicConfig(filename='script.log', level=logging.INFO,
format='%(asctime)s - %(levelname)s - %(message)s')

def connect_to_database():
    db_config = {
        'host': 'gsqdev.kul-dc.dhl.com',
        'user': 'dfdbadmin',
        'password': '4xhWrP~0Dp62',
        'database': 'datafactory_gsd_db'
    }

    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            return connection
    except Error as e:
        logging.error(f"Chyba při připojování k databázi: {e}")
    return None

def log_to_database(connection, event_type, event_description):
    insert_query = "INSERT INTO ApplicationLog_02 (EventType,
EventDescription) VALUES (%s, %s)"
```

```
    try:
        cursor = connection.cursor()
        cursor.execute(insert_query, (event_type, event_description))
        connection.commit()
        logging.info(f"Logováno: {event_type} - {event_description}")
    except Error as e:
        logging.error(f"Chyba při logování události: {e}")

def check_email_addresses():
    connection = connect_to_database()
    if connection is not None:
        select_query = "SELECT Zamestnanecke_cislo, Email FROM
Zamestnanci"

        try:
            cursor = connection.cursor(dictionary=True)
            cursor.execute(select_query)
            for row in cursor.fetchall():
                zamestnanecke_cislo = row['Zamestnanecke_cislo']
                email = row['Email']

                if not re.match(r"[^@]+@[^@]+\.[^@]+", email):
                    event_type = 'Failure'
                    event_description = f'Zaměstnanec s číslem
{zamestnanecke_cislo} má neplatnou emailovou adresu: {email}'
                    log_to_database(connection, event_type,
event_description)
                    logging.error(event_description)
                    sys.exit(1)  # Ukončení skriptu s chybovým kódem
        except Error as e:
            logging.error(f"Chyba při kontrole emailových adres: {e}")
        finally:
            if connection.is_connected():
                cursor.close()
                connection.close()

if __name__ == '__main__':
    check_email_addresses()
```

Pro dosažení plného pochopení fungování tohoto skriptu jsou poskytnuty podrobné popisy všech klíčových funkcí, které byly implementovány v rámci navrženého skriptu, spolu s význačnými aspekty jeho provádění.

connect_to_database(): Tato funkce se připojí k databázi s použitím konfiguračních údajů jako host, uživatel, heslo a název databáze. Pokud se připojení nezdaří, funkce vytiskne chybovou zprávu a vrátí None. V opačném případě vrátí objekt připojení.

log_to_database(connection, event_type, event_description): Tato funkce slouží k logování událostí do databáze ApplicationLog_02. Přijímá tři argumenty: objekt připojení k databázi, typ události a popis události.

Funkce provede vložení nového záznamu do tabulky ApplicationLog_02 s typem a popisem události. Pokud dojde k chybě při logování, funkce tuto chybu zaznamená v logu.

check_email_addresses(): Tato funkce provádí kontrolu emailových adres zaměstnanců v databázi. Nejprve se připojí k databázi. Poté provede dotaz na databázi, který získá seznam zaměstnanců a jejich emailové adresy. Pro každého zaměstnance ověří, zda jeho emailová adresa má platný formát (obsahuje text následovaný symbolem "@" a následně doménovou část s alespoň jedním znakem následovaným tečkou).

Pokud emailová adresa nemá platný formát, skript provede následující kroky:

- Zaloguje chybu do databáze ApplicationLog_02 s typem "Failure" a popisem, který zahrnuje zaměstnanecké číslo a neplatnou emailovou adresu.
- Vytvoří zprávu o chybě, kterou zaloguje do souboru logu.
- Ukončí běh skriptu s chybovým kódem 1.
- Po dokončení kontroly všech emailových adres zavře připojení k databázi a ukončí skript.

Skript také nastaví logování do souboru 'script.log' a zaznamenává informace o běhu skriptu, jako jsou úspěchy a chyby. Pokud najde neplatnou emailovou adresu, skončí a nepovolí uložení záznamu. Chyby jsou také logovány do databáze pro další analýzu a opravy.

Charakteristika

V našem konkrétním případě, kdy byl skript aplikován na tabulku "Zamestnanci," bylo identifikováno 68 záznamů s neplatnými formáty e-mailových adres. Všechny tyto identifikované záznamy byly systematicky zaznamenány v logovací tabulce "ApplicationLog_02" (Obr. 10). Úplné výsledky této aktivity jsou uvedeny v příloze této diplomové práce.

	LogID	Timestamp	EventType	EventDescription
1	2	2023-09-17 22:26:24	Failure	Zaměstnanec s číslem 121 má neplatnou emailovou adresu:
2	3	2023-09-17 22:26:24	Failure	Zaměstnanec s číslem 221 má neplatnou emailovou adresu:
3	4	2023-09-17 22:26:24	Failure	Zaměstnanec s číslem 456 má neplatnou emailovou adresu:
4	5	2023-09-17 22:26:24	Failure	Zaměstnanec s číslem 483 má neplatnou emailovou adresu:
5	8	2023-09-17 22:26:24	Failure	Zaměstnanec s číslem 372 má neplatnou emailovou adresu:
6	9	2023-09-17 22:26:24	Failure	Zaměstnanec s číslem 226 má neplatnou emailovou adresu:
7	10	2023-09-17 22:26:24	Failure	Zaměstnanec s číslem 239 má neplatnou emailovou adresu:

Obr. 9 – Výsledky platnosti e-mailových adres

Návrh řešení pro upravu platnosti e-mailových adres

Při návrhu našeho řešení bylo rozhodnuto, že veškeré nalezené informace týkající se identifikovaných chyb v datech, zejména neplatných e-mailových adres, systematicky zaznamenány v logovací tabulce. Současně tyto informace o chybách odeslány dodavateli dat, který byl požádán o další důkladnou analýzu a korekci problematických dat.

Tento postup nám umožní zajistit, že všechna data budou podrobena potřebným opravám a vylepšením. Tím dojde k výraznému zvýšení celkové kvality dat uložených v naší databázi. Toto je nezbytné pro udržení spolehlivosti a přesnosti našich dat v rámci organizace.

5 Výsledky a diskuse

5.1 Vyhodnocení analýzy a doporučení

V rámci této diplomní páce jsme provedli detailní analýzu obchodních požadavků týkajících se kvality dat v datové sadě "Zaměstnanci" v naší organizaci. Následně jsme provedli odpovídající zlepšení kvality dat v souladu s těmito požadavky. Konkrétně jsme implementovali několik důležitých kontrol datové kvality, které zahrnovaly:

Chybějící Data: Vytvořili jsme mechanismus kontroly, který zajistí vyplnění všech povinných polí pro záznamy zaměstnanců. V případě chybějících dat systém zobrazí chybové hlášení a nepovolí uložení neúplných záznamů.

Duplikáty: Implementovali jsme kontrolu na unikátnost Křestního_Jména, Příjmení, Emailu a Telefonního_cisla v databázi. Systém neumožní vkládání duplikátních záznamů a upozorní na existenci stejného Křestního_Jména, Příjmení, Emailu a Telefonního_cisla. Navíc jsme přidali mechanismus pro odstranění nalezených duplikátů.

Neplatné Hodnoty Platu: Vytvořili jsme kontrolu platových údajů zaměstnanců, abychom zajistili, že platy nejsou záporné nebo nereálně vysoké. V případě chybějících dat systém zobrazí chybové hlášení a nepovolí uložení neplatných platových hodnot.

Neplatné Formáty Dat: Vytvořili jsme kontrolu formátů dat, jako jsou emailové adresy, telefonní čísla, jména, příjmení, pohlaví, pozice a město. Systém ověřuje, zda data odpovídají očekávaným formátům a upozorňuje na chyby v případě neplatných formátů.

Tímto způsobem jsme plně splnili všechny obchodní požadavky na kvalitu dat.

Kromě toho bych chtěl zmínit přínosy diplomové práce. Tyto přínosy jsou následující:

Zlepšení datové kvality: Diplomová práce přinesla řešení pro zlepšení datové kvality v organizaci. Díky čištění, standardizaci a deduplikaci dat, provedeným vlastní aplikací v jazyce Python, byla dosažena zvýšená kvalita datové sady "Zaměstnanci".

Zvýšení spolehlivosti datových analýz: S vylepšenými daty bude organizace schopna provádět spolehlivé datové analýzy a reporty. To má pozitivní dopad na procesy rozhodování a strategické plánování.

Přizpůsobení konkrétním potřebám: Vytvoření vlastní aplikace v jazyce Python umožnilo přizpůsobit proces zlepšení datové kvality konkrétním potřebám a složitostem datového souboru organizace.

Připraveno na praktické nasazení: Výsledné řešení je připraveno na praktické nasazení v organizaci a bude pokračovat v poskytování výhod v oblasti datové kvality.

Na základě této práce a využití vlastní aplikace bylo možné dosáhnout následujících závěrů a doporučení:

Vytvoření aplikace: Navržená aplikace v programovacím jazyce Python poskytuje efektivní nástroj pro kontrolu a korekci datových chyb. Doporučuje se nadále rozvíjet a aktualizovat tuto aplikaci s ohledem na budoucí potřeby a rozšiřování datových atributů.

Důkladná validace: Aplikace by měla být nadále rozvíjena tak, aby umožňovala komplexní validaci dat, včetně ověřování konzistence, platnosti a integritu datových položek.

Sledování a dokumentace: Doporučuje se zavést systém pro sledování a dokumentaci prováděných operací nad daty. Tím bude zajištěna transparentnost a auditovatelnost datových operací.

Kontinuální vzdělávání: Zavedení pravidelných školení a vzdělávacích programů pro pracovníky, kteří budou aplikaci používat, zajistí efektivní využití nástroje.

Zajištění bezpečnosti: Zajišťujte bezpečný přístup a správu dat v aplikaci, aby byla informace chráněna před neoprávněným přístupem.

6 Závěr

Kvalita dat je klíčovým faktorem v rámci všech organizací, bez ohledu na jejich velikost. Zlepšení kvality dat v malých a středních podnicích přináší mnoho významných a dosažitelných výhod. Vysoká kvalita dat umožňuje podnikům lépe informovaná a kvalifikovaná rozhodnutí, identifikaci trendů, příležitostí a výzev, a vytvoření strategií a plánů, které jsou založeny na konkrétních datech spíše než na domněnkách a odhadech. Kvalitní data také přispívají ke zvýšení efektivity a produktivity, což vede ke snížení časových ztrát, omezení chyb a minimalizaci ztrát. Díky kvalitním datům je snazší jejich zpracování, správa a analýza, což umožňuje automatizaci a zefektivnění klíčových obchodních procesů. V konečném důsledku poskytují kvalitní data konkurenční výhodu na trhu. Pro zlepšení kvality dat v malých a středních firmách je klíčové implementovat standardní procesy a systémy pro sběr, ukládání a analýzu dat, a to i v menším měřítku. Důležitou součástí je také implementace strategie správy dat, která zahrnuje definování rolí a odpovědností za řízení dat, stanovení procesů pro ukládání dat a zavedení kontrolních opatření ke zvýšení kvality dat.

Myslím si, že tato práce bude užitečná všem, kteří mají zájem o oblast datové kvality, zejména těm, kteří se zabývají jejím řízením. Tato práce poskytuje návod, jak postupovat při řízení kvality dat, a přináší metodiku, kterou lze prakticky využít při řešení otázek týkajících se datové kvality v podnicích. Doufám, že se mi podařilo dosáhnout všech cílů, které jsem si pro tuto práci stanovil. Přesto věřím, že existuje široký prostor pro další rozvoj a prohlubování této problematiky.

Zdroje

AGRAWAL, Parag et al. Foundations of Uncertain-Data Integration. *Proc. VLDB Endow*. 2010, 3(1), s. 1080–1090.

CICHY, Corinna, RASS, Stefan. An Overview of Data Quality Frameworks. *IEEE Access 7.* 2019, pp. 24634–24648.

COUTURE, Nancy. Why data governance? [online]. [cit. 22.10.2022]. Dostupné z: https://www.cio.com/article/3245588/why-data-governance.html

DAMA International, DAMA – DMBOK, Data Management Body Of Knowledge Second Edition. Technics publications, 2017. ISBN 978-16-346-2236-3

DENNIS, Amber. Data Architecture with Data Governance [online]. [cit. 11.10.2022]. Dostupné z: https://www.dataversity.net/data-a rchitecture-with-data-governance-a-proactive-approach/

GÁLA, L., POUR, J., ŠEDIVÁ, Z. Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi. 3., aktualizované vydání. Praha: Grada Publishing, 2015. ISBN 978-80-247-5457-4

HÁJEK, Petr. Jak „mít pořádek“ v datech. [online]. [cit. 07.10.2022]. Dostupné z: https://profinit.eu/blog/jak-mit-poradek-v-datech/

LABERGE, R. Datové sklady - Agilní metody a business intelligence. Praha: Computer Press, Albatros Media a.s. 2016. ISBN 978-80-251-3729-1

LADLEY, John. Data Governance: How to Design, Deploy and Sustain an Effective Data Governance Program. 2., vydání. Cambridge, Massachusetts: Academic Press, 2019. ISBN: 9780128158326

MACURA, Marek. Integration of Data from Heterogeneous Sources using ETL Technology. *Computer Science*. 2014, 15(2), s. 109–132.

MAHANTI, R. Data Quality: Dimensions, Measurement, Strategy, Management, and Governance. Quality Press, 2019. ISBN 978-0-87389-977-2.

MARR, Bernard. How To Define A Data Use Case – With Handy Template. [online]. [cit. 09.10.2022]. Dostupné z: https://bernardmarr.com/default.asp?contentI D=1837

MBI. Řízení datových zdrojů a jejich kvality. [online]. [cit. 15.10.2022] Dostupné z: https://mbi.vse.cz/

OSIRI, John Kalu. Entrepreneurial Marketing: Creating a Customer Base. London: Unleash Publishing, 2014. ASIN B00HPL25ZA

POUR, J. a kol. Self service business intelligence: jak si vytvořit vlastní analytické, plánovací a reportingové aplikace. Praha: Grada Publishing, 2018. ISBN 978-80-271-0616-5

SLÁNSKÝ, D. Data a analytika pro 21. století. Praha: Professional Publishing, 2018. ISBN 978-80-88260-25-7.

SMITH, Rachel. Seven Important Characteristics Of Data Quality & Metrics To Track. [online]. [cit. 01.10.2022]. Dostupné z: https://www.clearpointstrategy.com/data-quality-metrics/

TOUS, Rubén. Data integration with XML and semantic web technologies: novel approaches in the design of modern data integration systems. PhD thesis. Pompeu Fabra University, 2008. ISBN 978-3-8364-7138-1.

TYRYCHTR, J. Business Intelligence. Praha: ČZU v Praze, 2014. ISBN 978-80-213-2516-6.

MELTON, JIM. SQL: The Standard and the Language. *Archive Opengroup.* [Online] [cit. 07.10.2022.] Dostupné z: http://archive.opengroup.org/public/tech/datam/sql.htm.

Seznam obrázků