



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

OPTIMALIZACE SQL DATABÁZE VE VÝROBNÍ SPOLEČNOSTI

SQL DATABASE OPTIMIZATION IN A MANUFACTURING COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Adam Hrušovský

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Kříž, Ph.D.

BRNO 2024

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Adam Hrušovský**
Vedoucí práce: **Ing. Jiří Kříž, Ph.D.**
Akademický rok: 2023/24
Studijní program: Manažerská informatika

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

Optimalizace SQL databáze ve výrobní společnosti

Charakteristika problematiky úkolu:

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

Cíle, kterých má být dosaženo:

Cílem práce je optimalizace stávající databáze a návrh jejího efektivnějšího provozu.

Základní literární prameny:

KOCH, Miloš a Bernard NEUWIRTH. Datové a funkční modelování. Vyd. 4., rozš. Brno: Akademické nakladatelství CERM, 2010. ISBN 978-80-214-4125-5.

KŘÍŽ, Jiří a Petr DOSTÁL. Databázové systémy. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3064-8.

POKORNÝ, Jaroslav a Michal VALENTA. Databázové systémy. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.

STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES. Naučte se SQL za 28 dní: [stačí hodina denně]. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2023/24

V Brně dne 4.2.2024

L. S.

Ing. Jiří Kříž, Ph.D.
garant

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Bakalárska práca sa zaoberá optimalizáciu existujúcej SQL databázy vo výrobnej spoločnosti. Práca je rozdelená na tri hlavné kapitoly. V prvej kapitole sa venuje teoretickým východiskám, ktoré slúžia ako podklad pre ďalšie časti. Druhá kapitola sa zaoberá analýzou súčasného stavu v spoločnosti. Následne, v tretej kapitole, sú navrhnuté vlastné riešenia problémov.

Kľúčové slová

Dáta, optimalizácia, SQL, MS SQL Server, relácie, databáza, databázová štruktúra

Abstract

Bachelor's thesis deals with the optimization of an existing SQL database in a manufacturing company. The thesis is divided into three main chapters. The first chapter deals with the theoretical background, which serves as a basis for the following parts. The second chapter deals with the analysis of the current state in the company. Subsequently, the third chapter proposes its own solutions to the problems.

Keywords

Data, optimization, SQL, MS SQL Server, relations, database, database structure

Bibliografická citácia

HRUŠOVSKÝ, Adam. *Optimalizace SQL databáze ve výrobní společnosti* [online]. Brno, 2024 [cit. 2024-04-15]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/160749>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Ing. Jiří Kříž, Ph.D.

Čestné prehlásenie

Prehlasujem, že predložená bakalárska práca je pôvodná a spracoval som ju samostatne. Prehlasujem, že citácia použitých prameňov je úplná a že som vo svojej práci neporušil autorská práva (v zmysle zákona č. 121/2000 Zb., o autorskom práve a o právach súvisiacich s právom autorským).

V Brne dňa 15. 4. 2024

Adam Hrušovský

autor

Pod'akovanie

V prvom rade by som sa chcel poďakovať vedúcemu mojej bakalárskej práce pánovi Ing. Jiřímu Křížovi, Ph.D. za jeho ochotu, čas a odborné rady, ktoré mi pomohli pri spracovaní tejto práce. Taktiež by som rád poďakoval svojej rodine za psychickú a finančnú podporu.

OBSAH

ÚVOD	10
CIELE PRÁCE, METÓDY A POSTUPY SPRACOVANIA.....	11
1 TEORETICKÉ VÝCHODISKÁ PRÁCE	12
1.1 Základné pojmy.....	12
1.1.1 Informácia.....	12
1.1.2 Dáta	12
1.1.3 Znalosti.....	13
1.2 Dátové modely	14
1.2.1 Lineárny dátový model.....	14
1.2.2 Hierarchický dátový model	14
1.2.3 Sieťový dátový model	15
1.2.4 Relačný dátový model.....	16
1.2.5 Objektový dátový model	16
1.3. Databázový systém.....	16
1.3.1 História spracovania dát	17
1.3.2 Databáza	18
1.3.2 Systém riadenia bázy dát.....	19
1.3.3 Architektúry systému riadenia bázy dát	20
1.3.4 Databázové pojmy.....	21
1.4 Relačný model	22
1.4.1 Teória množín.....	22
1.4.2 Terminológia relácie	23
1.4.3 Integritné obmedzenia pre entity	24
1.4.4 Integritné obmedzenia pre vzťahy	26
1.4.5 Relačné schéma modelu	27
1.4.6 Normalizácia.....	27
1.5 Jazyk SQL.....	29
1.5.1 Syntaxe SQL.....	29
1.5.2 Dátové typy.....	30
1.5.3 Pevné vykonávacie plány	30
1.5.4 Indexovanie databázy	31
1.5.5 MS SQL Server.....	31
1.5.6 MS SQL Server Management Studio.....	32

2	ANALÝZA SÚČASNÉHO STAVU	33
2.1	Základné informácie o spoločnosti	33
2.2	Organizačná štruktúra.....	33
2.3	Hardvérové vybavenie.....	34
2.4	Softvérové vybavenie	36
2.5	Analýza databázy	36
2.5.1	Informácie o databáze	36
2.5.2	Stav databázy	37
2.5.3	Nevyužívané tabuľky	38
2.5.4	Nekorektné nakonfigurovanie primárnych a cudzích kľúčov	39
2.5.5	Rozdielny jazyk v tabuľkách	40
2.5.6	Názvová konvencia	40
2.5.7	Absencia dokumentácie databázy	41
2.5.8	Jeden prihlasovací účet do aplikácií databázy	41
2.5.9	Inkrementálne zálohovanie databázy na hodinovej báze	42
2.5.10	Zavedenie indexov	42
2.6	Zhrnutie analýzy	42
3	VLASTNÝ NÁVRH RIEŠENÍ	44
3.1	Odstránenie nevyužívaných tabuliek	44
3.2	Úprava konfigurácie primárnych a cudzích kľúčov.....	44
3.3	Úprava jazyku databázy	45
3.4	Úprava názvovej konvencie.....	48
3.5	Návrh dokumentácie databázy	49
3.6	Právomoci prístupu do aplikácií databázy.....	50
3.7	Zálohovanie pomocou zrkadlenia diskov	51
3.8	Vytvorenie indexov	52
3.9	Zhodnotenie návrhu.....	54
	ZÁVER	55
	ZOZNAM POUŽITÝCH ZDROJOV	56
	ZOZNAM POUŽITÝCH OBRÁZKOV	58

ÚVOD

Začiatky databáz siahajú do čias tradičných kartoték, ktoré umožňovali štruktúrované uchovávanie údajov. S postupným rozvojom technológií sa však tradičné kartotéky transformovali na elektronické databázy a moderné databázové systémy, ako ich poznáme dnes. Databázy sú kľúčovým prvkom v informačných technológiách a ich význam sa prejavuje v mnohých oblastiach, ako sú výrobné spoločnosti, školstvo či zdravotníctvo. Databázové systémy zabezpečujú spoľahlivé ukladanie a rýchly prístup k veľkému množstvu dát, čím podporujú rozhodovací proces a zefektívňujú pracovné postupy v rôznych odvetviach.

Častým javom je, že spoločnosti prevzatím databázy od sesterskej firmy, alebo zakúpením hotovej databázovej štruktúry, nezriedka už nevenujú dostatočnú pozornosť jej údržbe. Databáza môže fungovať roky bez revízií, aktualizácií a optimalizácií, čo môže viesť k rôznym nedostatkom a zníženej efektívnosti. Bez primeranej starostlivosti môže dochádzať k znečisteniu dát, stratám údajov, či neefektívnemu využitiu kapacít databázového systému.

Cieľom tejto bakalárskej práce je analyzovať súčasný stav databázy vo výrobnej spoločnosti a navrhnúť riešenia pre identifikované nedostatky. Optimalizáciou databázy sa bude zabezpečovať zvýšenie jej efektívnosti a celková lepšia prevádzka.

CIELE PRÁCE, METÓDY A POSTUPY SPRACOVANIA

Cieľom práce je optimalizácia databázy a návrh jej efektívnejšej prevádzky. Práca je rozdelená do troch hlavných kapitol.

V prvej kapitole práce sú teoretické východiská, ktoré opisujú základné pojmy k téme práce a slúžia ako podklad pre zvyšné časti práce.

Druhá kapitola práce sa zaoberá analýzou súčasného stavu. V tejto časti práce je opísaný podnik, v ktorom sa databáza používa a jeho informačné technológie. Následne je vykonaná analýza databázy, ktorá zahŕňa identifikáciu jej nedostatkov.

V poslednej kapitole práce sú na základe analýzy zistených nedostatkov navrhnuté riešenia pre optimalizáciu databázy.

1 TEORETICKÉ VÝCHODISKÁ PRÁCE

1.1 Základné pojmy

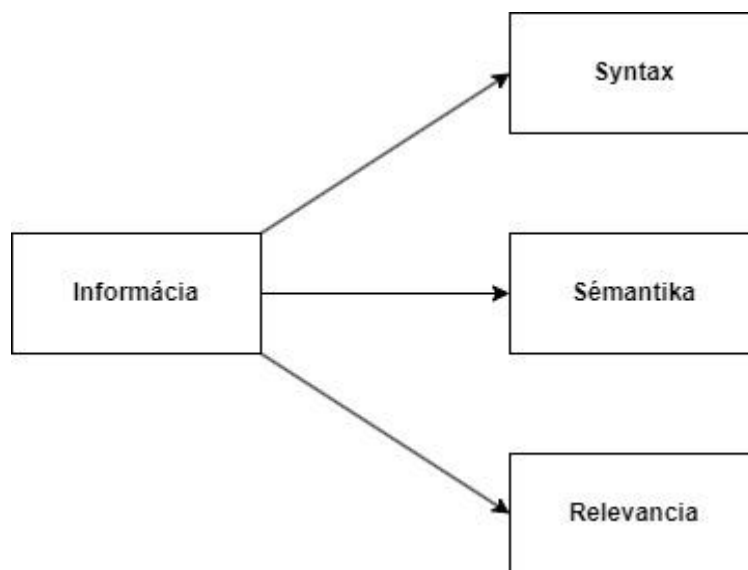
1.1.1 Informácia

Informácia je správa alebo vnem, ktorý spĺňa určité kritériá. Tieto kritériá určujú, či je informácia relevantná a užitočná. [6]

Prvým kritériom je syntaktická relevancia. Syntaktická relevancia znamená, že informácia musí byť správne formulovaná. To znamená, že musí byť gramaticky správna a musí mať jasný význam. [6]

Druhým kritériom je sémantická relevancia. Sémantická relevancia znamená, že informácia musí mať zmysel. To znamená, že musí byť zrozumiteľná a musí sa vzťahovať k danej téme. [6]

Tretím kritériom je pragmatická relevancia. Pragmatická relevancia znamená, že informácia musí byť užitočná. To znamená, že musí mať pre príjemcu nejaký význam alebo hodnotu. [6]



Obrázok č. 1: Informácia (Zdroj: [6] s. 4)

Informácie môžu byť ďalej klasifikované na základe rôznych pohľadov. Klasifikujú sa napríklad na operatívne, strategické a taktické informácie. [6]

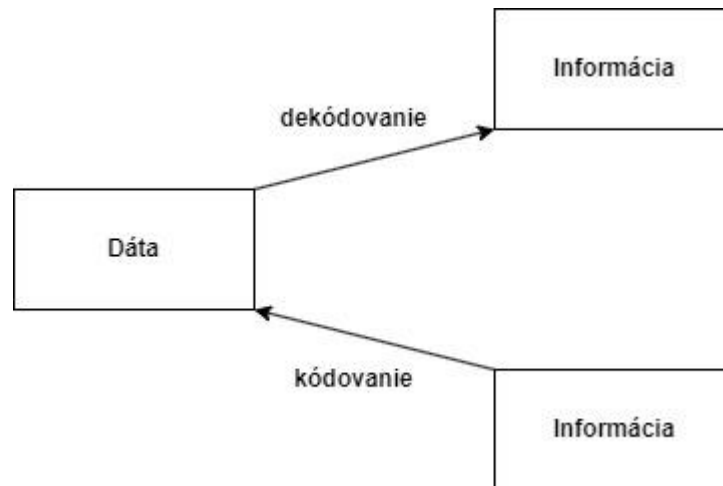
1.1.2 Dáta

Dáta sú správy, ktoré môžu byť využité pri rozhodovaní. Ak sú dáta takto využité, stávajú sa informáciami. Preto môžeme povedať, že dáta sú potenciálnymi

informáciami. [6]

Správa, ktorú človek zachytí a porozumie, sa nazýva dáta. Dáta môžu byť neskôr spracované do inej formy alebo uložené pre neskoršie spracovanie. Sú vyjadrené prostredníctvom fyzického média, napríklad atramentu a papiera a majú určitý význam. [6]

Informácia sa transformuje na dáta, keď sa zapíše na vhodné médium. Tento proces zahŕňa kódovanie informácie, čo znamená, že sa informácia vyjadrí v určitej forme, ktorá je vhodná pre médium. Kódovanie tiež zahŕňa presné definovanie postupu, ako informáciu zapísať. Dáta môžu byť neskôr dekódované, čo znamená, že sa informácia z nich môže opäť získať. Dekódovanie umožňuje vrátiť sa k pôvodnej informácii pre konkrétneho príjemcu. [6]



Obrázok č. 2: Dáta (Zdroj [6] s. 5)

1.1.3 Znalosti

Znalosti sú schopnosť využiť informácie a dáta v rôznych kontextoch. Keď sa stretneme s problémom, náš mozog automaticky aktivuje existujúce znalosti a vyberie relevantné informácie na jeho riešenie. Tento proces rozhodovania na základe znalostí je bežný a používa sa v mnohých situáciách každodenného života. [6]

V praxi a odbornej literatúre sa často uvádza, že čas potrebný na rozhodnutie by nemal byť prekročený časovým rámcom stanoveným závažnosťou problému. Čas potrebný na rozhodnutie nie je prekročený časovým rámcom stanoveným závažnosťou problému. [6]

1.2 Dátové modely

Pre každý druh dátového objektu, ktorý je súčasťou informačného systému, sa vytvorí jedinečná dátová štruktúra. Pri navrhovaní informačného systému napríklad pre univerzitu sa vytvoria aspoň tieto objekty: študent, pedagóg, skúška a predmet. V skutočnom svete však medzi týmito objektami často existujú vzťahy. Napríklad študent môže absolvovať komplexné skúšky alebo byť hodnotený konkrétnym pedagógom. Preto je dôležité, aby informačný systém presne zobrazoval tieto vzťahy medzi dátami. [6]

V súčasnosti existuje päť hlavných typov dátových modelov, ktoré sa používajú pri návrhu informačných systémov. Tieto modely ponúkajú rôzne metódy pre štruktúrovanosť a organizáciu dát v informačnom systéme. Je dôležité vybrať ten, ktorý najlepšie vyhovuje špecifickým požiadavkám projektu. [6]

1.2.1 Lineárny dátový model

V lineárnych dátových modeloch sú jednotlivé skupiny objektov usporiadané v tabuľkách bez akýchkoľvek väzieb medzi nimi. Tento model je jedinečný tým, že ho možno implementovať na akomkoľvek médiu. Príkladom lineárneho modelu môže byť kartotéka pacientov, kde sú jednotlivé karty s informáciami o pacientoch systematicky usporiadané v krabici. Každá karta predstavuje jednu vetu v databázovom súbore a medzi kartami neexistuje žiadny vzťah, okrem lineárneho poradia predchodcu a nasledovníka. [6]

Implementácia lineárneho modelu je charakterizovaná tým, že jednotlivé entity nemajú vzájomné spojenia. To z neho robí vhodný model pre špecifické prípady, ako je napríklad kartotéka pacientov. Tento prístup k modelovaniu dát môže byť užitočný v situáciách, v ktorých nie sú potrebné komplexné vzťahy medzi rôznymi časťami informačného systému. [6]

1.2.2 Hierarchický dátový model

Hierarchický dátový model je založený na stromovej štruktúre, ktorá umožňuje vyjadriť vzťahy medzi záznamami typu rodič/potomok. Každý záznam v tomto modeli je uzlom v stromovej štruktúre a môže mať jedného alebo viacerých potomkov. [11]

Existujú tri základné pravidlá, ktoré musia byť dodržané: [11]

1. Záznam nemôže patriť do viac ako jednej skupiny;

2. Vlastníkom je rodič a člen je označený ako potomok;
3. Každý potomok môže mať iba jedného rodiča, vždy je jeden prvok nadradený a tým sa prvky duplikujú.

Hierarchický dátový model je obzvlášť vhodný pre aplikácie, kde je potrebné vyjadriť hierarchické vzťahy medzi dátami, ktoré majú prirodzene hierarchickú štruktúru. Príkladom takýchto aplikácií môžu byť organizačné alebo skladové systémy. Nájdenie konkrétnych dát v tomto modeli vyžaduje postupnú navigáciu cez záznamy v stromovej štruktúre. V tejto štruktúre má každý záznam rodiča a môže mať jedného alebo viacerých potomkov. [11]

1.2.3 Sieťový dátový model

V sieťovom dátovom modeli sú dáta organizované do rovinatej štruktúry grafu, kde každý záznam môže byť spojený s ľubovoľným počtom ďalších záznamov. Tento model umožňuje, aby každý záznam obsahoval odkazy na iné entity v databáze, ktoré s ním logicky súvisia. To umožňuje využitie mnohonásobného rodičovstva, teda situácie, keď jeden záznam môže byť súčasne rodičom viacerých iných záznamov. Sieťový model je v podstate rozšírením hierarchického modelu o podporu mnohonásobného rodičovstva. Jednotlivé entity tak môžu vytvoriť sieť s veľmi rôznorodou štruktúrou. [11]

V tomto modeli sa zavádzajú určité pojmy: [11]

- Typ záznamu (Record) je názov pre typ entity
- Komponent je názov pre atribút
- Výskyt záznamu – je názov jednotlivé entity v samotnej databáze
- Vzťah medzi dvoma typmi záznamov kardinality 1:1 alebo 1:N sa nazýva set. Sety prepájajú záznamy rôzneho alebo rovnakého typu a spojenie môže byť realizované na jeden alebo viac záznamov. V sieťovom modeli sa definujú iba binárne vzťahy typov 1:1 a 1:N medzi dvoma typmi záznamov R a T. Ostatné typy vzťahov sa už dopredu rozložia na konceptuálnej úrovni. Set je definovaný pomocou svojho vlastníka a členov.

Pri aktualizácii sieťového modelu je potrebné zmeniť štruktúru databázy. Pôvodný návrh štruktúry je možné implementovať iba vytvorením novej databázy. Manipulácia s dátami v sieťovom modeli sa vykonáva postupným prechádzaním grafu cestou

definovanou ukazovateľmi. [11]

1.2.4 Relačný dátový model

Relačný dátový model je jedným z najčastejšie používaných dátových modelov v súčasnosti. Tento model vzniká spojením niekoľkých lineárnych modelov prostredníctvom prvkov nazývaných relačné kľúče. Toto spojenie nie je trvalé, ale vzniká iba vtedy, keď je potrebné získať dáta z rôznych tabuliek. Po ukončení práce s modelom spojenie zaniká. [6]

Relačný model umožňuje flexibilné spojenie dátových entít, čo umožňuje efektívne získavanie informácií z rôznych zdrojov. Tento prístup je vhodný v situáciách, kde je potrebné dočasne spojiť dáta z viacerých tabuliek na základe definovaných vzťahov, aby sa získala konkrétna informácia. [6]

1.2.5 Objektový dátový model

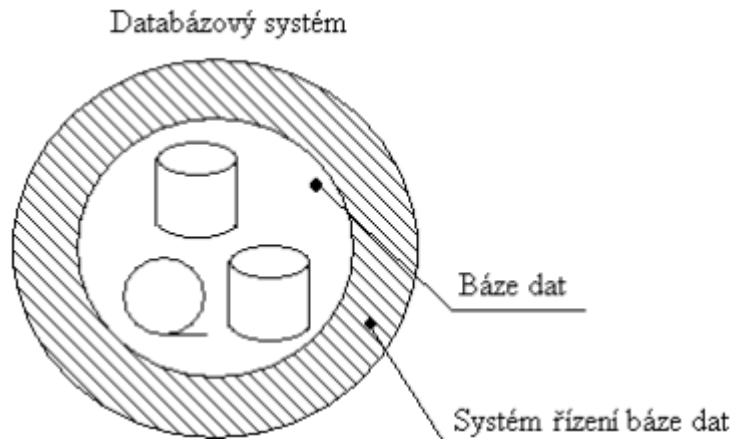
Objektový dátový model je založený na objektoch, ktoré obsahujú nielen atribúty, ale aj metódy. Každý objekt v databáze je identifikovaný jedinečným identifikátorom OID, ktorý umožňuje priame vzťahy medzi objektmi. Okrem toho môžu v objektovom modeli existovať aj relačné vzťahy. [6]

V objektových modeloch sú dáta a ich správanie zapuzdrené do objektov. To znamená, že jediným spôsobom, ako pristupovať k dátam objektu, je prostredníctvom jeho metód. Tento prístup umožňuje dosiahnuť vysokú úroveň dátovej abstrakcie a nezávislosti dát, čo prispieva k efektívnemu a bezproblémovému spravovaniu informácií. [6]

1.3. Databázový systém

Databázový systém je organizovaný spôsob spracovania údajov, ktorý pozostáva z databázy a systému riadenia databáz. Informácie sú uložené v databáze, ktorá je samostatná od aplikácií. Aplikácie vyberajú údaje z databázy podľa požiadaviek používateľa. [11]

Od nepamäti ľudia potrebovali zapisovať a uchovávať informácie. Táto potreba sa v súčasnej modernej spoločnosti stala ešte naliehavejšou, pretože databázové systémy hrajú kľúčovú úlohu v mnohých oblastiach života, napríklad v evidencii občanov, zdravotnej starostlivosti, ekonomike a ďalších. [7]



Obrázok č. 3: Databázový systém (Zdroj: [11] s. 8)

Databázový systém sa všeobecne skladá zo systému riadenia databáz a samotnej databázy. Tieto dva komponenty umožňujú zhromažďovať, ukladať a udržiavať konkrétne informácie, ktoré sú dôležité v práci alebo v živote. Centrálne ukladanie informácií v databáze je v dnešnej dobe jedným z najčastejších dôvodov na kúpu počítača. [7]

1.3.1 História spracovania dát

Tradičné kartotéky boli prvými úložiskami dát, ktoré umožňovali štruktúrovanie a usporiadanie údajov podľa rôznych kritérií a triedenie nových položiek. Všetky operácie s nimi boli vykonávané manuálne. Správa takýchto kartoték bola v mnohých ohľadoch podobná správe dnešných databáz. [11]

V roku 1890 sa začalo s strojovým, elektromechanickým spracovaním údajov, pričom sa ako pamäťové médium používali dierkové štítky. S príchodom počítačov sa v roku 1959 predstavil prvý jazyk pre spracovanie údajov COBOL. Od roku 1965 sa začali vyvíjať prvé systémy riadenia databáz, ktoré boli prevádzkované na sálových počítačoch. Poprednými postavami vo vývoji databáz boli Charles Bachman, ktorý vytvoril sieťovú koncepciu databáz, a Edgar F. Codd, ktorý v roku 1970 navrhol relačný dátový model. [11]

V sedemdesiatych rokoch 20. storočia sa v spoločnosti IBM uskutočnil výskum v oblasti relačných databáz, v rámci ktorého bol vytvorený jazyk SEQUEL ako sada príkazov pre ovládanie týchto databáz. Neskôr bol SEQUEL premenovaný na SQL. Prvou databázou, ktorá implementovala jazyk SQL, bola Oracle, ktorú v roku 1980 predstavila spoločnosť Relational Software, Inc. Druhou databázou, ktorá

implementovala SQL, bola DB2 od firmy IBM. Osemdesiate roky možno považovať za zlatý vek databáz, pretože v tomto období boli vyvinuté ďalšie systémy, ako napríklad Progres, Informix alebo SyBase. Všetky tieto systémy používali variant jazyka SQL. [11]

V histórii hromadného spracovania údajov sa dá sledovať vývoj dvoch typov prístupov k údajom. Prvým typom bol súborový prístup, ktorý bol predchodcom databázového prístupu. Pri súborovom prístupe sa údaje ukladali na magnetické pásky, čo výrazne obmedzovalo prístup k nim. Súčasťou súboru údajov bol jeho popis, ktorý bol potrebný pre aplikácie na prístup k údajom. Aplikácie v tomto prípade boli úzko viazané na štrukturalizáciu údajov. [11]

1.3.2 Databáza

Databázu možno chápať ako súbor logicky prepojených súborov a ich definícií. Databáza je komplexná a centralizovaná dátová štruktúra, ktorá je spracovávaná. Pre databázy bola navrhnutá špecifická interná dátová štruktúra, ktorá je uplatniteľná pre všetky možné scenáre a metódy použitia. [11]

Pri databázovom prístupe sa definícia údajov a ich údržba oddeľuje od užívateľských programov. Údaje sa už neukladajú do izolovaných súborov, ale do komplexnejšej štruktúry údajov nazývanej databáza, ktorá je centralizovane spracovávaná. [12]

Zásadne databáza zahŕňa: [12]

- údajové prvky a vzťahy medzi nimi,
- integritné obmedzenia,
- schému.

Údajové prvky sú nástroje, ktoré sa používajú na zaznamenávanie hodnôt. Napríklad NAZOV_SKOLY, RIADITEL atď. Vzťahy medzi údajmi sa reprezentujú pomocou sofistikovanejších dátových štruktúr. Napríklad informácia, že pán X je riaditeľom školy Y, môže byť zaznamenaná prostredníctvom záznamu obsahujúceho dvojicu (X, Y). Integritné obmedzenia sú podmienky, ktoré musia byť splnené údajmi v databáze. Napríklad hodnota školy (v miliónoch Kč) musí byť číslo z rozsahu <1, 100>. [12]

Schéma databázy je popis údajov, ktorý je zrozumiteľný pre užívateľa databázy a

zodpovedajúcej vrstve softvéru. Integritné obmedzenia sa zvyčajne zahrňujú do schémy databázy. [12]

V súčasnosti je do technológie databáz zahrňané aj modelovanie údajov, čo zahŕňa techniky návrhu databázy. [12]

1.3.2 Systém riadenia bázy dát

Systém poskytuje softvérové nástroje na vytváranie štruktúry, ukladanie, vyhľadávanie a ochranu údajov. Zároveň zabezpečuje databázu a umožňuje komunikáciu medzi užívateľom a systémom. V podstate je to programový nástroj, ktorý koordinuje zdieľaný prístup k databáze a poskytuje mechanizmy na zabezpečenie bezpečnosti a integrity údajov. [11]

Poskytuje nasledujúce služby: [7]

- Definícia dát - ponúka prostriedky na stanovenie a uchovávanie údajovej entity
- Údržba dát - udržiava entitu tým, že každému z jej členov vyhradzuje záznam, ktorý pozostáva z položiek popisujúcich čiastkové informácie o danom člene
- Manipulácia s dátami - poskytuje funkcie, ktoré umožňujú používateľovi vkladať, aktualizovať, odstraňovať a triediť údaje v databáze
- Zobrazovanie dát - môže ponúkať konkrétne metódy prezentácie údajov používateľovi
- Integrita dát - poskytuje metódu alebo metódy na zabezpečenie správnosti údajov

S databázovým systémom sú spojené dva typy jazykov:

1. Jazyk pre definíciu dát (JDD²) - tento jazyk slúži na vytvorenie všetkých potrebných definícií dát v databáze v rámci danej aplikačnej domény. Celkový popis dát v jednej databáze sa označuje ako (logická) schéma databázy.
2. Jazyk pre manipuláciu dát (JDM³) - tento jazyk má dva hlavné účely. Po prvé, slúži na aktualizáciu dát, teda na pridávanie, odstraňovanie a menenie dát v databáze. Po druhé, používa sa na výber dát podľa zadaných požiadaviek. Aktivita druhého typu, známa ako dopytovanie, je pre užívateľov databázových systémov kritická. Táto činnosť prebieha v reči schémy databázy alebo v prostriedkoch, ktoré sú jazykovo na ešte vyššej úrovni. Príslušná časť jazyka pre manipuláciu dát sa nazýva dopytovací jazyk.

Tieto dva typy jazykov sú neoddeliteľné od efektívneho fungovania a správy databázového systému, umožňujúc tvorbu, úpravu a dotazovanie nad dátami v databáze. [12]

Systém riadenia databázy je súbor postupov a dátových štruktúr, ktoré poskytujú nezávislosť aplikácií od podrobností o tom, ako sa údaje ukladajú a spravujú v počítači. [11]

Príklady systémov riadenia databáz zahŕňajú Microsoft SQL Server, MySQL, Oracle, Informix, SyBase, Microsoft Access a ďalšie. [11]

1.3.3 Architektúry systému riadenia bázy dát

Centralizované platformy

V centralizovanom systéme sa všetko spracovanie vykonáva na primárnom hostiteľskom počítači. Systém riadenia databázy, aplikácie, ktoré pracujú s databázou a komunikačný softvér sú uložené na tomto počítači. Užívatelia prístupujú k databáze prostredníctvom terminálov, ktoré sú pripojené k primárnemu hostiteľskému počítaču buď lokálne, alebo na diaľku. [7]

Systémy na osobných počítačoch

Keď systém riadenia databázy funguje na jednom počítači, tento počítač plní úlohu hostiteľského počítača aj terminálu. V porovnaní s väčšími systémami sú funkcie systému riadenia databázy a databázových aplikácií zjednotené v jednom programe. Databázové aplikácie na počítači spracúvajú užívateľský vstup, výstup na obrazovku a prístup k údajom na disku. [7]

Zlúčenie rôznych funkcií do jedného systému poskytuje systému riadenia databázy silu, flexibilitu a rýchlosť, ale môže viesť k zníženiu bezpečnosti a integrity údajov. Tieto systémy sa pôvodne vyvíjali ako samostatné, ale dnes sú väčšinou integrované do lokálnych sietí. V lokálnej sieti sú údaje uložené na serveri súborov, ktorý beží pod sieťovým operačným systémom. [7]

Databázové systémy klient/server

V databázovom systéme klient/server sa spracovanie rozdelí medzi dva počítače: klientsky počítač, na ktorom beží databázová aplikácia a databázový server, na ktorom beží systém riadenia databázy alebo jeho časť. Zdieľané zdroje, ako diskový priestor a tlačiarne, sú stále uložené na serveri súborov lokálnej siete. [7]

Databázový server môže byť umiestnený na rovnakom počítači ako server

súborov, alebo na samostatnom počítači. Databázová aplikácia je spustená na klientskom počítači, ktorý sa nazýva systém front-end a vykonáva všetky spracovania obrazoviek a užívateľských vstupov a výstupov. Analogicky, systém back-end, ktorý sa spustí na databázovom serveri, zabezpečuje spracovanie dát a prístup na disk. [7]

Systémy distribuovaného spracovania

V tejto architektúre sa dáta medzi rôznymi hosťiteľskými systémami zdieľajú tak, že sa aktualizácie prenášajú prostredníctvom priameho alebo diaľkového spojenia. Aplikácia, ktorá sa spustí na jednom alebo viacerých hosťiteľských počítačoch, vyberá časti dát, ktoré sa zmenili v priebehu časového intervalu a posieľa ich centrálnej jednotke alebo iným hosťiteľským počítačom v sieti. Ostatné databázy sa potom aktualizujú, aby boli všetky systémy synchronizované. Tento model je obzvlášť užitočný v situáciách, kde je potrebné zdieľať dáta medzi rôznymi hosťiteľskými systémami a udržiavať ich synchronizované. [7]

1.3.4 Databázové pojmy

Entita

Abstrakcia je konceptuálna reprezentácia existujúcich entít. Je to abstraktné znázornenie reálnych objektov, ktoré odráža plánovanú štruktúru databázy. Príklady takýchto entít zahŕňajú zákazníkov a objednávky, produkty a sklady, alebo študentov a ich predmety. [11]

Atribút

Vlastnosti entít, ktoré sú dôležité z hľadiska skúmanej oblasti a jej evidencie, sa nazývajú atribúty. Každéj entite z množiny entít je priradená hodnota atribútu z nejakej neprázdnej množiny. Napríklad, pre entitu "zamestnanec", sú atribútmi osobné číslo, meno, priezvisko atď. Tieto atribúty sú podstatné pre identifikáciu a charakterizáciu zamestnanca v rámci organizácie. [11]

Výskyt entity

Prvok množiny entity je nazývaný entitným výskytom. Entitný výskyt je identifikovaný vlastnosťami, ktoré sú definované v rámci množiny entity. Napríklad, v prípade entity "zamestnanec" môže entitný výskyt zahŕňať jedinečný identifikátor 242555, meno Juraj Novák, adresu Kovačická 12, lokalitu Brezno a povolanie montážneho pracovníka. Tieto vlastnosti detailne popisujú a charakterizujú konkrétny prípad zamestnanca. [11]

Relácia

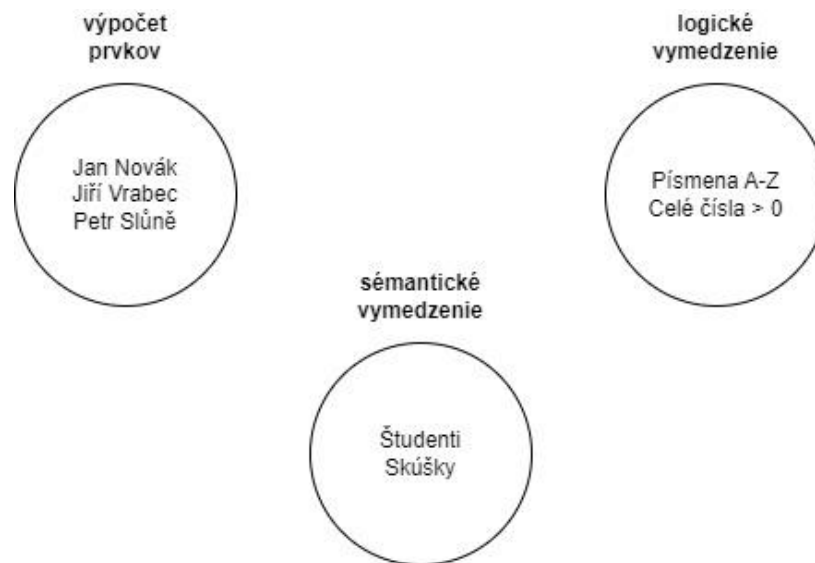
Medzi dvoma alebo viacerými entitami je možné vytvoriť spojenie, ktoré sa nazýva relácia. Relácia je zvyčajne vyjadrená slovesom, ako napríklad "má", "obsahuje", "požičal si" a podobne. Napríklad, veta "učiteľ vyučuje študenta" ilustruje vzťah "vyučuje", ktorý existuje medzi entitami učiteľ a študent. Tento vzťah definuje interakciu alebo vzájomný vzťah medzi učiteľom a študentom v kontexte vyučovania. [11]

1.4 Relačný model

1.4.1 Teória množín

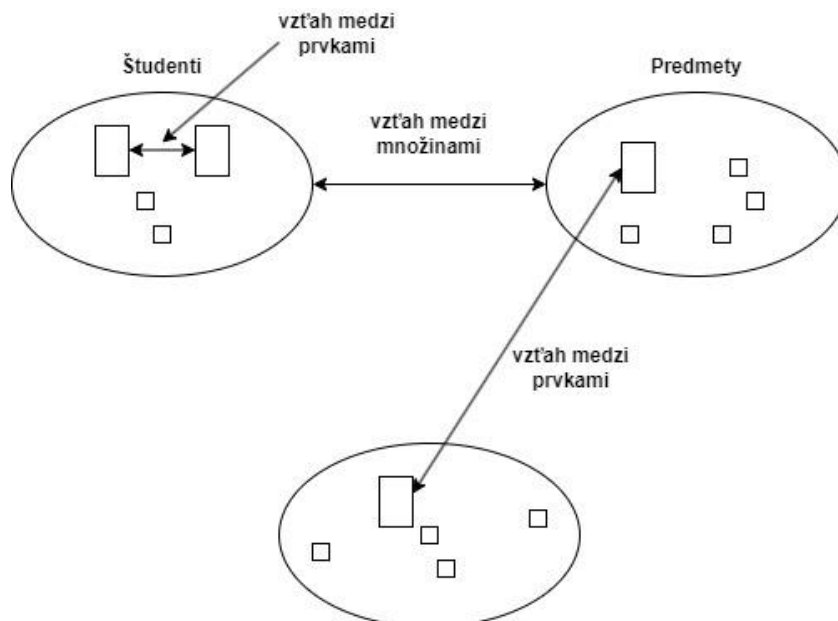
V oblasti modelovania dát sa na definíciu a popis prvkov dátového modelu využíva teória množín. Teória množín poskytuje základné pojmy a nástroje na popis štruktúry dát. [6]

Množiny sa môžu popisovať pomocou prvkov, ktoré patria do danej množiny. Prvky množiny sa môžu vymenovať, opísať pomocou logických výrazov, alebo definovať prostredníctvom ich významu. [6]



Obrázok č. 4: Typy definícií množín pomocou ich prvkov (Zdroj: [6] s. 24)

V rámci konceptu množín sa rozlišujú dva základné typy vzťahov: vzťahy medzi prvkami v rámci jednej množiny a vzťahy medzi rôznymi množinami. Vzťahy medzi množinami môžu existovať medzi množinami ako celkami, alebo medzi jednotlivými prvkami týchto množín. [6]

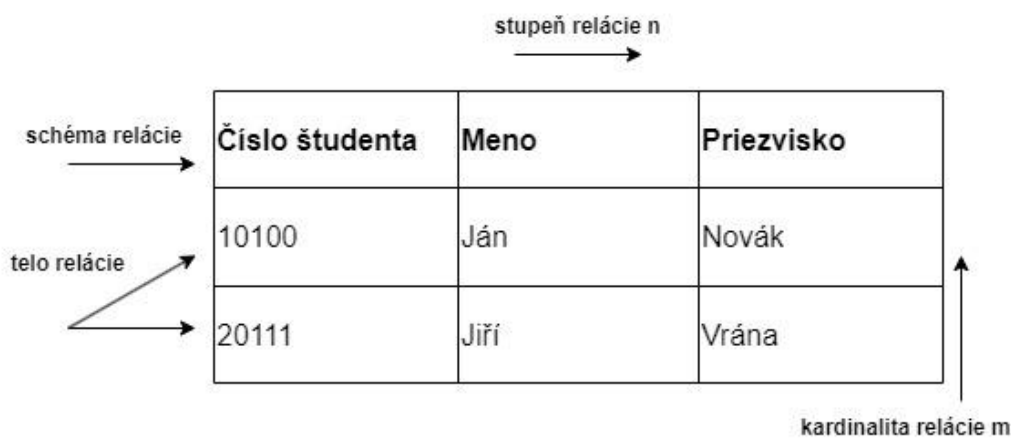


Obrázok č. 5: Typy vzťahov v množinách (Zdroj: [6] s. 25)

Na manipuláciu s množinami a ich vzťahmi sa využívajú rôzne nástroje. Tieto nástroje zahŕňajú operácie, ako sú zjednotenie, prienik, testovanie rovnosti, inklúzia a doplnok. Nástroje poskytujú prostriedky na analyzovanie, porovnávanie a manipuláciu s množinami v rámci dátového modelovania. [6]

1.4.2 Terminológia relácie

„Ak máme množiny, v terminológii teórie množín „domény“ napríklad čísel študentov – D_1 , mien študentov – D_2 a priezvisk študentov D_3 , potom relácie na doménach D_1, D_2, \dots, D_n je dvojica $R = (R, R^*)$, kde $R = R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$ je schéma relácie a $R^* \subseteq D_1 \times D_2 \times \dots \times D_n$ je telo relácie.“ [6, s. 25]



Obrázok č. 6: Popis relačnej schémy (Zdroj: [6] s. 26)

Telo relačného modelu je podmnožinou kartézskeho súčinu, ktorá obsahuje iba skutočné záznamy. Schéma relácie je definovaná ako množina atribútov, ktoré sú prítomné v každom zázname. Stupeň relácie je počet atribútov v schéme relácie a kardinalita relácie je počet záznamov v relácii. Stupeň relácie je konštantný, zatiaľ čo kardinalita môže byť premenná. [6]

Relačný model poskytuje jednotnú dátovú štruktúru vo forme n-árnych relácií, ktoré sú definované svojimi schémami. Táto štruktúra sa využíva na reprezentáciu a správu dát v databázových systémoch. [6]

Pravidlá pre tabuľkovú prezentáciu relácie: [6]

- každá n-tica predstavuje jeden riadok v tabuľke
- poradie riadkov v tabuľke nemá význam
- poradie riadkov v tabuľke nemá význam a každý riadok je unikátny (neexistujú duplicity)
- poradie stĺpcov v tabuľke nie je dôležité
- význam každého stĺpca je identifikovaný názvom atribútu
- žiadne dva atribúty v tabuľke nemajú rovnaký názov
- hodnoty v každom stĺpci (nie sú ďalej rozdelené na menšie časti)

1.4.3 Integritné obmedzenia pre entity

Integrita hodnôt

Hodnota každého atribútu relácie musí byť prípustná v rámci definovanej množiny hodnôt pre daný atribút. Toto zabezpečuje: [6]

1. Definícia domény ako sady možných hodnôt.
2. Špecifikácia povolených hodnôt pre každý atribút, zahŕňajúca:
 - dátový typ atribútu
 - povinnosť zadávať hodnotu (neprázdna hodnota)
 - jedinečnosť hodnôt v rámci stĺpca
 - určenie rozsahu hodnôt (minimálna a maximálna hodnota)
 - stanovenie štandardných hodnôt
 - definovanie masky pre vkladanie hodnôt
 - zoznam prípustných hodnôt pre atribút

Entitná integrita

Každá relácia má jeden alebo viac atribútov, ktorých hodnoty jednoznačne identifikujú každý riadok v tejto relácii. Tieto atribúty sa nazývajú primárny kľúč. [6]

Tvorí množinu atribútov, ktorá spĺňa nasledujúce charakteristiky: [6]

- Je jednoznačný, čo znamená, že neexistuje iná relácia s rovnakými hodnotami v tomto kľúči.
- Je minimálny, teda neumožňuje vypustiť žiadny atribút bez straty jeho jednoznačnosti.

Pre primárny kľúč platia nasledovné pravidlá: musí byť kompletný, čo znamená, že žiadna hodnota nesmie chýbať a každá n-tica v relácii musí byť identifikovateľná hodnotou primárneho kľúča. [6]

Kandidátny kľúč je skupina atribútov, ktoré jednoznačne identifikujú každý záznam v relácii. Na rozdiel od primárneho kľúča však nie je špecificky vybraný. V relácii môže existovať viacero kandidátnych kľúčov, pričom jeden z nich je následne označený ako primárny kľúč, zatiaľ čo ostatné sú označené ako alternatívne kľúče. [6]

Referenčná integrita

Cudzí kľúč predstavuje atribút, ktorý vykazuje tieto nezávislé vlastnosti: [6]

- každá hodnota v ňom je buď úplne definovaná alebo nezadaná
- existuje iná relácia s primárnym kľúčom, pričom každá definovaná hodnota v primárnom kľúči zodpovedá hodnote v cudzom kľúči inej relácie

Vďaka spojeniu cudzieho kľúča a primárneho kľúča z iného tabuľkového záznamu máme možnosť vytvárať vzťahy medzi reláciami. Integritné pravidlá v tomto prípade zahŕňajú: [6]

- definovanie cudzieho a primárneho kľúča na rovnakej doméne
- udržiavanie konzistentných hodnôt cudzieho kľúča v celej databáze

	Názov položky	Typ	Dĺžka
kandidátne kľúče	Číslo študenta	C	5
primárny kľúč	Rodné číslo	C	10
	Dátum narodenia	D	
	Meno	C	12
	Priezvisko	C	15
	Odbor	C	4
cudzí kľúč	Ročník	N	1

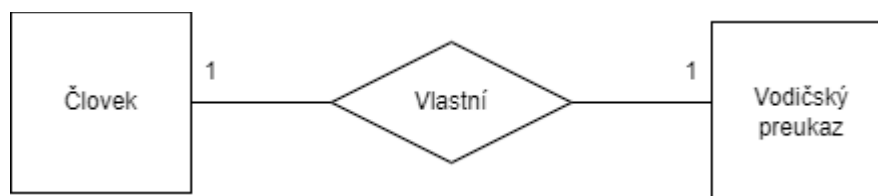
Obrázok č. 7: Príklady jednotlivých kľúčov (Zdroj: [6] s. 29)

1.4.4 Integritné obmedzenia pre vzťahy

Vzťah 1:1

Vzťah 1:1 medzi dvoma reláciami znamená, že každej n-tici v jednej relácii zodpovedá presne jedna n-tica v druhej relácii. V praxi to znamená, že každá hodnota v jednej relácii môže byť priradená iba k jednej hodnote v druhej relácii. [6]

Príkladom vzťahu 1:1 môže byť vzťah medzi reláciami "človek" a "vodičský preukaz". Každému človeku môže zodpovedať iba jeden vodičský preukaz, alebo žiadny. To znamená, že existuje jeden-jeden vzťah medzi týmito dvoma entitami. [6]



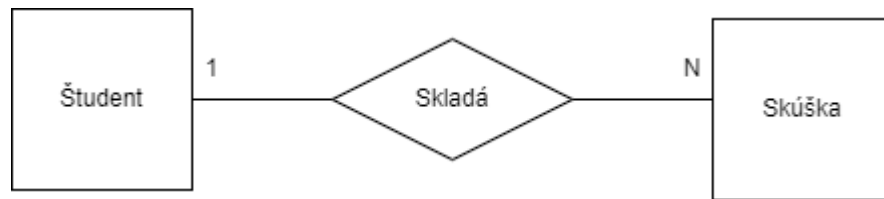
Obrázok č. 8: Vzťah 1:1 (Zdroj: [6] s. 31)

Vzťah 1:N

Vzťah 1:N medzi dvoma reláciami znamená, že každej n-tici v jednej relácii môže zodpovedať jedna alebo viac n-tíc v druhej relácii. V praxi to znamená, že každá hodnota v jednej relácii môže byť priradená k jednej alebo viacerým hodnotám v druhej relácii. [6]

Príkladom vzťahu 1:N môže byť vzťah medzi reláciami "študent" a "skúška". Každému študentovi môže zodpovedať viacero skúšok, ale každá skúška je priradená

len jednému študentovi. To znamená, že existuje jeden-mnoho vzťah medzi týmito dvoma entitami. [6]

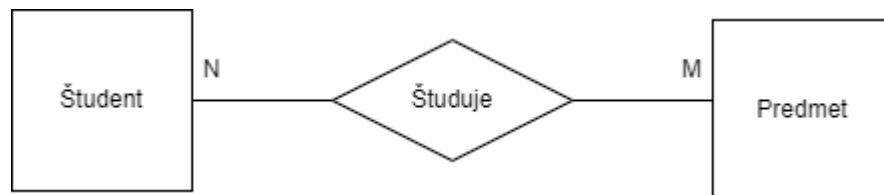


Obrázok č. 9: Vzťah 1:N (Zdroj [6] s. 31)

Vzťah N:M

Vzťah N:M medzi dvoma reláciami znamená, že jednej alebo viacerým n-ticiam v jednej relácii môže zodpovedať niekoľko n-tíc v druhej relácii. V praxi to znamená, že jedna alebo viac hodnôt v jednej relácii môže byť priradená k viacerým hodnotám v druhej relácii. [6]

Príkladom vzťahu N:M môže byť vzťah medzi reláciami "študent" a "predmet". Viacerí študenti môžu byť priradení k jednému alebo viacerým predmetom, a zároveň jeden predmet môže byť priradený k viacerým študentom. To znamená, že existuje mnoho-mnoho vzťah medzi týmito dvoma entitami. [6]



Obrázok č. 10: Vzťah N:M (Zdroj: [6] s. 32)

1.4.5 Relačné schéma modelu

Prípustný model relačnej databázy je kolekcia relácií, ktoré spĺňajú všetky integritné obmedzenia definované pre túto databázu. Tým sa zabezpečuje, že dáta v týchto reláciách sú konzistentné a v súlade s definovanými normami. [6]

V niektorých prípadoch môžu byť integritné obmedzenia definované pre jednotlivé relácie v databáze. V takom prípade sa celkový model databázy zapisuje ako kolekcia párov relácií a ich integritných obmedzení. [6]

1.4.6 Normalizácia

Normalizácia je krokový proces, pri ktorom sa relácie postupne rozkladajú na vhodnejšie tvary. Cieľom je zachovať bezstratovosť dát, udržať existujúce závislosti a

eliminovať duplicitné informácie. Existuje 5 normálnych foriem, ktoré definujú, ako by mali byť relácie navrhnuté, aby splnili tieto požiadavky. [6]

Pri normalizácii sa návrh dátových štruktúr upravuje tak, aby spĺňal určité požiadavky. Tieto požiadavky sú zamerané na efektívne ukladanie dát, minimalizáciu duplicitných údajov a zachovanie integrity a konzistencie dát. [6]

1. normálna forma (multizávislosť)

Relácia spĺňa prvú normálnu formu, ak všetky jej atribúty sú atomické, to znamená, že neobsahujú viaceré hodnoty. [11]

Databáza je najjednoduchšia a najprehľadnejšia, ak sú jej relácie definované tak, že ich atribúty obsahujú jednoduché hodnoty. [6]

2. normálna forma (funkčná závislosť)

Relácia je v druhej normálnej forme, ak spĺňa podmienky prvej normálnej formy a ak každý neklúčový atribút závisí od celého primárneho kľúča, nie len od jeho časti. Potreba riešiť druhú normálnu formu vzniká iba v prípade, ak primárny kľúč obsahuje viac ako jednu hodnotu. [11]

3. normálna forma (tranzitívna závislosť)

Relácia je v tretej normálnej forme, ak je v druhej normálnej forme a ak hodnota žiadneho neklúčového atribútu nie je určená hodnotami iného neklúčového atribútu, ktorý je zase určený hodnotami primárneho kľúča. [11]

Boyce – Coddova normálna forma

Boyce-Coddova normálna forma je prísnejším variantom tretej normálnej formy. Relácia, ktorá je v Boyce-Coddovej normálnej forme, je automaticky aj v tretej normálnej forme, ale opačné nie je pravda. [6]

Relácia je v Boyce-Coddovej normálnej forme, ak sú jej kandidátne kľúče navzájom nezávislé. To znamená, že hodnota žiadneho kandidátneho kľúča nie je určená hodnotami iných kandidátnych kľúčov. [6]

4. normálna forma

Relácia je v štvrtej normálnej forme, ak sú všetky jej viachodnotové závislosti funkčne závislé od kandidátnych kľúčov. To znamená, že každá skupina opakujúcich sa hodnôt v relácii je jednoznačne určená hodnotami kandidátnych kľúčov. [6]

5. normálna forma

Piata normálna forma je prísny koncept, ktorý sa uplatňuje len v špecifických

prípadoch. Týka sa to situácií, kde medzi viacerými reláciami existuje cyklická závislosť. V takom prípade musia všetky relácie v cykle obsahovať rovnaké hodnoty pre všetky spoločné atribúty. [6]

1.5 Jazyk SQL

Jazyk SQL sa používa na manipuláciu a získavanie dát z relačných databáz. Pomocou jazyka SQL je možné vykonávať rôzne operácie, ako napríklad upravovať štruktúru databázy, meniť nastavenia zabezpečenia systému, pridávať užívateľské opatrenia k databáze alebo tabuľkám, vykonávať dotazy na informácie z databázy a aktualizovať obsah databázy. [15]

1.5.1 Syntaxe SQL

Vytvorenie tabuliek

Na vytvorenie tabuliek sa používa príkaz CREATE TABLE. [7]

```
CREATE TABLE meno_tabulky (meno_stlpcu typ integritné obmedzenie)
```

Manipulácia s tabuľkami

Príkaz pre úpravu vytvorenej tabuľky je ALTER TABLE. Pre pridanie stĺpcov do tabuľky používame príkaz ADD, pre úpravu vlastností stĺpcov príkaz MODIFY, pre zmazanie stĺpcov sa používa DROP, pre zmazanie tabuľky sa používa DROP TABLE. [7]

```
ALTER TABLE meno_tabulky  
ADD (meno_stlpcu typ_stlpcu [integritné obmedzenie], ...)  
MODIFY (meno_stlpcu typ_stlpcu [integritné obmedzenie], ...)  
DROP meno_stlpcu  
DROP TABLE meno_tabulky
```

Vkladanie a aktualizácia dát

Na vkladanie dát do tabuľky slúži príkaz INSERT INTO. Pre aktualizáciu už vložených dát slúži príkaz UPDATE. Príkaz, ktorý slúži na vymazanie riadku, je DELETE. Príkaz WHERE sa používa pre určenie kritérií výberu a príkaz FROM pre určenie destinácie odkiaľ sa majú čerpať údaje. [7]

```
INSERT INTO meno_tabulky [(mená stĺpcov)] VALUES (zoznam hodnôt)
```

UPDATE meno_tabulky
SET stlpec = hodnota, ... [WHERE podmienky]
DELETE FROM meno_tabulky [WHERE podmienky]

Zobrazenie, zoskupenie a zoradenie dát

Príkaz SELECT slúži pre vybranie a zobrazenie dát z tabuliek podľa nami nastavených kritérií. [7]

Príkaz GROUP BY používame pre zoskupenie stĺpcov podľa rovnakých hodnôt a príkaz ORDER BY pre ich zoradenie. GROUP BY sa uvádza pred ORDER BY a kritérium sa udáva vždy až na koniec príkazu [7]

SELECT meno_stlpcu FROM meno_tabulky [GROUP BY výrazy pre zoskupenie] [ORDER BY podľa čoho zoradiť]

1.5.2 Dátové typy

Hodnoty položiek v databáze budú pozostávať z rôznych typov údajov. Väčšina záznamov bude obsahovať jednoduchý text, pričom poštové smerovacie čísla a čísla domov budú predstavovať úplné kladné čísllice. Položka „vlastníctvo vodičského preukazu“ bude mať logickú hodnotu buď „áno“ alebo „nie“. Základné dátové typy v jazyku SQL sú nasledujúce: [7]

- INTEGER celé číslo ($-2^{31} - 1$ až 2^{31})
- SMALLINT celé číslo (-32768 až 32767)
- NUMERIC (m, n) des. číslo (m – počet plat. číslic, n – počet des. miest)
- FLOAT(n) reálne číslo (max. 38 des. miest, n – počet plat. číslic)
- CHAR(n) reťazec znakov do max. dĺžky 255 (n – počet znakov)
- VARCHAR(n) reťazec znakov premennej dĺžky (n – počet znakov)
- DATE dátum
- TIME čas

1.5.3 Pevné vykonávacie plány

SQL Server disponuje optimalizátorom, ktorý dokáže optimalizovať vykonávacie

plány pre väčšinu dotazov. Vďaka tomu sa zvýši ich efektivita. V prípade neštandardných dotazov je možné vytvoriť lepší vykonávací plán, ktorý sa zameriava na konkrétny problém, čím sa dosiahne ešte lepšia efektivita. [8]

Na základe vykonávacích plánov vieme vytvoriť odhadované vykonávacie plány, ktoré nám ďalej umožňujú optimalizovať ich štruktúru, alebo vytvoriť indexy pre zrýchlenie dotazov. [4]

1.5.4 Indexovanie databázy

Indexy v databáze predstavujú iný spôsob usporiadania dát, ako je ich fyzické uloženie na disku. Môžu byť vytvorené na jednom stĺpci tabuľky alebo na kombinácii viacerých stĺpcov. Používajú sa z troch hlavných dôvodov: na zvýšenie výkonu dotazov, na filtrovanie dát podľa obsahu polí a na presadzovanie referenčnej integrity. [12]

Keď index slúži ako primárny kľúč, hovoríme o primárnom indexovaní. V ostatných prípadoch sa používa sekundárne indexovanie. Pri použití sekundárneho indexovania je potrebné brať do úvahy, že v údajoch môže mať rovnakú hodnotu indexu viacero záznamov. [2]

Poznáme taktiež aj zhlukované indexy. Pri týchto indexoch sú dáta uložené v rovnakom bloku ako index. [12]

1.5.5 MS SQL Server

Microsoft SQL Server je systém správy relačných databáz, ktorý umožňuje ukladanie a správu dát. Nástroje a aplikácie sa pripájajú k databáze SQL Server pomocou jazyka Transact-SQL (T-SQL). [18]

Databázový systém MS SQL Server je výkonný nástroj, ktorý podporuje rôzne typy analýzy údajov. Používa sa na implementáciu dátových skladov, vytváranie zostáv a prezentáciu údajov. Má množstvo nástrojov, ktoré sú súčasťou dátovej platformy MS SQL Server a sú to napríklad: [10]

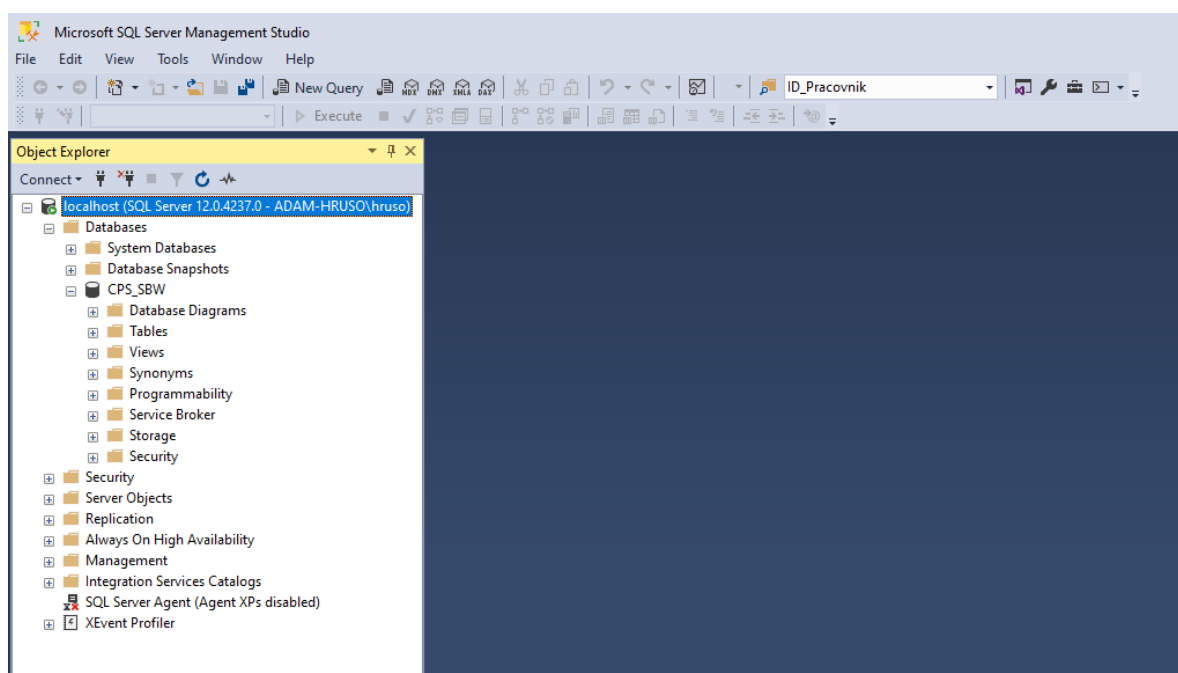
- **Replikácia dát:** Pre aplikácie, ktoré spracúvajú rozptýlené alebo mobilné dáta, zabezpečujú replikačné služby.
- **Notifikačné služby:** Funkcie na doručovanie upozornení, ktoré podporujú vytváranie a implementáciu škálovateľných aplikácií. Umožňujú prenos

informácií v reálnom čase, prispôsobených potrebám každého používateľa, cez celý rad mobilných a pripojených zariadení.

- **Integračné služby:** Funkcie pre skladovanie údajov a celopodnikovú integráciu údajov, ktoré zahŕňajú extrakciu, transformáciu a načítanie (ETL).
- **Analytické služby:** Viacdimenzionálne úložisko s nástrojmi OLAP na analýzu veľkých a komplikovaných súborov údajov.
- **Oznamovacie služby:** Komplexné riešenie pre vytváranie, správu a doručovanie statických a interaktívnych správ na webe.

1.5.6 MS SQL Server Management Studio

Integrované prostredie MS SQL Server Management Studio (SSMS) poskytuje nástroje na správu všetkých komponentov SQL Serveru. SSMS umožňuje pristupovať, konfigurovať, spravovať a vyvíjať SQL Server. Je komplexným riešením, ktoré kombinuje grafické nástroje s editormi skriptov. SSMS poskytuje administrátorom databáz a vývojárom všetkých úrovní prístup k SQL Serveru. [17]



Obrázok č. 11: MS SQL Server Management Studio (Zdroj: Vlastné spracovanie)

2 ANALÝZA SÚČASNÉHO STAVU

V nasledujúcej kapitole bola vykonaná analýza súčasného stavu spoločnosti, pre ktorú bol vypracovaný aj návrh riešení. V analýze boli charakterizované základné informácie o spoločnosti a jej organizačná štruktúra. Ďalej bolo preskúmané hardwarové a softwarové vybavenie spoločnosti. Nakoniec bola zanalyzovaná databáza spoločnosti, v ktorej sme zistili nedostatky, ktoré sme pomenovali a navrhli vylepšenia.

2.1 Základné informácie o spoločnosti

Dáta, ktoré boli použité v rámci práce, boli získané z reálnej spoločnosti XY s.r.o. Konateľ spoločnosti vyjadril želanie, aby niektoré informácie o firme boli zmenené alebo cenzurované, z dôvodu aby bola zachovaná anonymita spoločnosti.

Sídlo spoločnosti je v obci blízko Banskej Bystrice na Slovensku a bola založená v priebehu 20. storočia, pričom najväčší rozmach nastal v roku 1992 po vstupe nemeckého partnera. Aktuálne zamestnáva približne 500 kmeňových zamestnancov a z dôvodu flexibility, respektíve možnosti reagovať na výkyvy trhu, cca. 15-20 agentúrnych zamestnancov. Spoločnosť sa špecializuje hlavne na oblasť automobilového priemyslu, sekundárne na stavebný priemysel a výrobu plastových dielov, pričom aktivity sú prioritne zamerané na výrobu a dodávku produktov pre automobilový sektor. Medzi hlavné oblasti špecializácie patria aktuátorika, hnacie káble, lanová technika, brzdové a dverové systémy, konštrukcia a výroba plastových dielov. Väčšina výrobkov je exportovaná za hranice Slovenska firmám pôsobiacim v automobilovom segmente.

Právna forma firmy je štruktúrovaná ako spoločnosť s ručením obmedzeným. Táto právna forma poskytuje určitú ochranu majiteľom spoločnosti a zároveň umožňuje efektívne riadenie a prevádzkovanie obchodných aktivít.

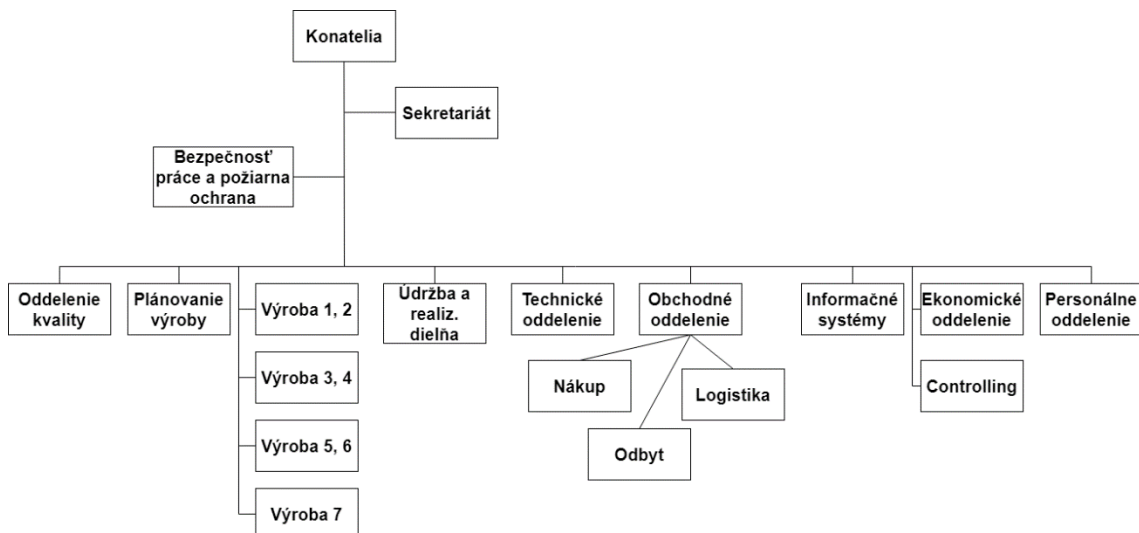
2.2 Organizačná štruktúra

Na uvedenej organizačnej schéme je znázornená zjednodušená organizačná štruktúra spoločnosti. Spoločnosť sa skladá z viacerých oddelení, z ktorých každé má svoje vlastné špecifické úlohy a zodpovednosti. Medzi oddelenia spoločnosti patria oddelenie kvality, oddelenie plánovania výroby, oddelenie výroby, oddelenie údržby, technické oddelenie, obchodné oddelenie, oddelenie informačných systémov,

ekonomické oddelenie, personálne oddelenie, oddelenie bezpečnosti práce a požiarnej ochrany a o oddelenie kontroly (controlling).

Každé oddelenie je riadené a jeho fungovanie je zabezpečované jeho nadriadeným. Niektoré oddelenia sú rozdelené na ďalšie časti, pričom oddelenie výroby je členené na výrobu 1, 2, 3, 4, 5, 6 a 7, kde sa jednotlivé výroby špecializujú na iný segment a sú umiestnené v samostatných výrobných priestoroch. Obchodné oddelenie je rozdelené na nákup, logistiku a odbyť, pričom každá z týchto častí je riadená vlastným vedúcim, ktorý je zodpovedný za daný obor a podriadený vedúcemu obchodného oddelenia.

Vedenie spoločnosti je zabezpečované dvoma konateľmi s vlastným sekretariátom. Spoločnosť je riadená a komunikácia v nej je zabezpečovaná prostredníctvom hierarchickej organizačnej štruktúry. Táto štruktúra umožňuje efektívne vykonávanie pracovných úloh v rámci jednotlivých oddelení a ich častí.

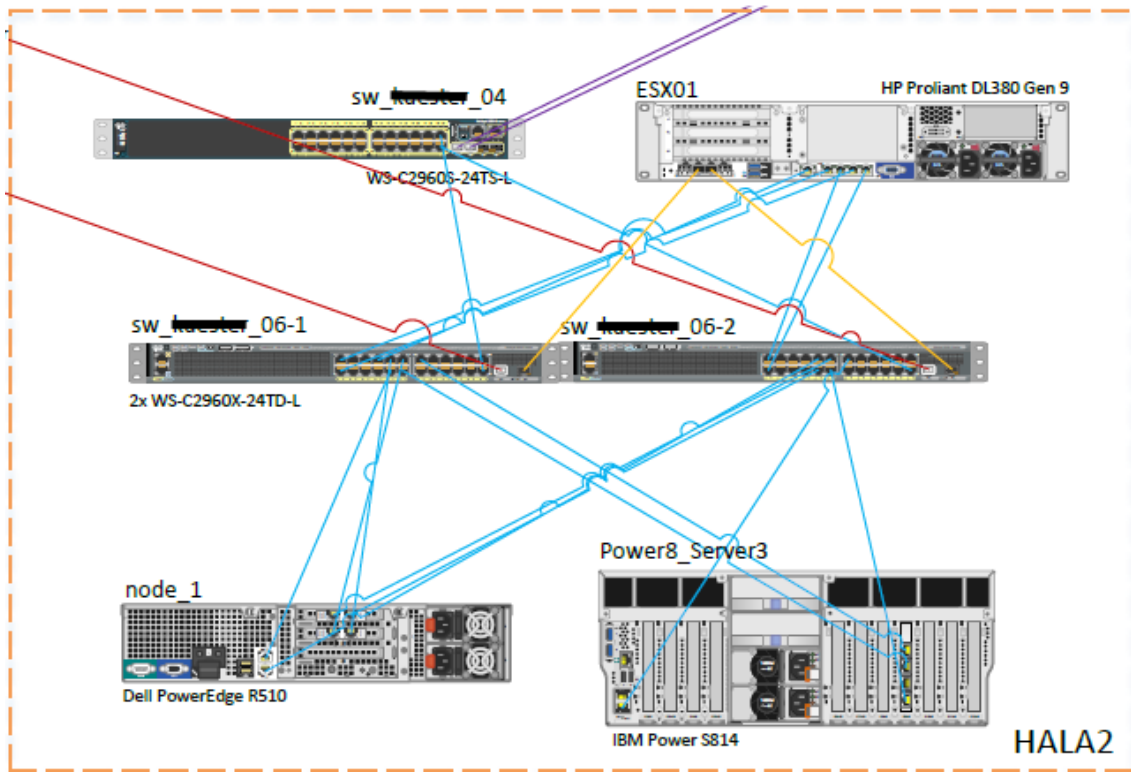


Obrázok č. 12: Organizačná schéma spoločnosti (Zdroj: Vlastné spracovanie)

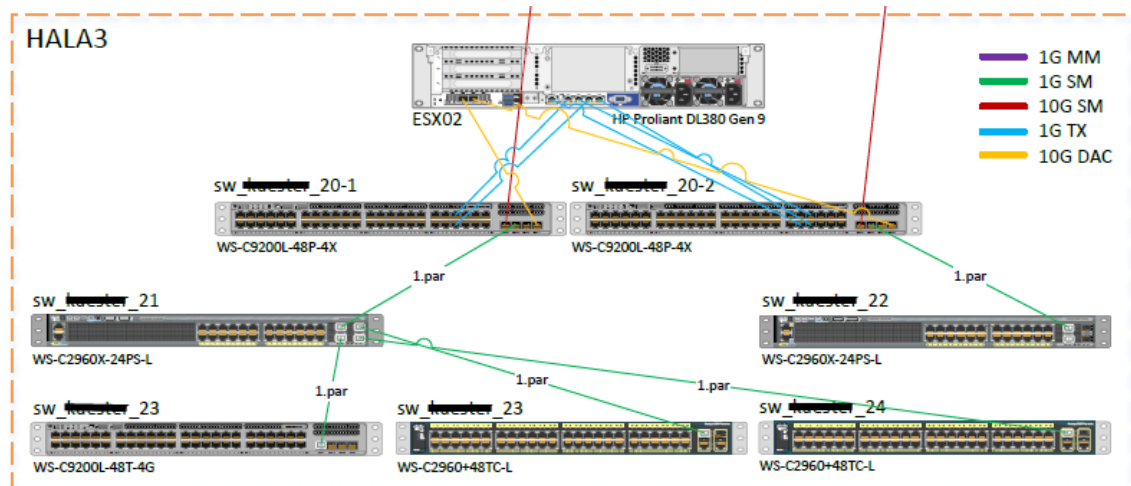
2.3 Hardvérové vybavenie

Hardvérové vybavenie je znázornené na nasledujúcich schémach, ktoré boli upravené tak, aby sa zameriavali na hardvérové komponenty spojené výlučne so SQL serverom a jeho databázou. SQL server a jeho databáza sú prevádzkované v serverovej miestnosti umiestnenej v hale tri. Schémy obsahujú informácie o kabeláži aj do iných častí spoločnosti. Uvedené informácie majú informačný charakter a nie sú relevantné v rámci tejto analýzy. Konfigurácia hardvéru umožňuje efektívnu prevádzku SQL servera a správu jeho databázy. Konfigurácia je optimalizovaná pre špecifické

požiadavky a potreby. Správca siete je zodpovedný a má na starosť hardvérové vybavenie spoločnosti.



Obrázok č. 13: Schéma hardvérového vybavenia haly 2 (Zdroj: [14])



Obrázok č. 14: Schéma hardvérového vybavenia haly 3 (Zdroj: [14])

Spoločnosť má dve serverovne, ktoré sú strategicky umiestnené v hale 2 a hale 3. V prvej serverovni sú umiestnené tri prepnáče, stolové servery Dell PowerEdge R510 a IBM Power S814 a rackový server HP Proliant DL380 Gen 9. Druhá serverovňa obsahuje sedem prepnáčov a rackový server HP Proliant DL380 Gen 9.

Spoločnosť využíva serverovne ako hlavný hub pre správu a údržbu databázového systému, vrátane prevádzky SQL servera. Zamestnanecké počítače z IT oddelenia sú pripojené na serverovne, kde pracujú programátori. Pripojenie počítačov na serverovne sa realizuje prostredníctvom ethernetových káblov alebo Wi-Fi siete, čo umožňuje rýchly a spoľahlivý prístup k potrebným zdrojom a aplikáciám. Celková konfigurácia serverovní umožňuje účinnú správu a prevádzku SQL servera a jeho pridruženej databázy.

2.4 Softvérové vybavenie

Softvérové vybavenie spoločnosti je tvorené virtuálnymi servermi s operačným systémom Windows, ktoré sú implementované pomocou platformy vmWare. Jeden virtuálny server je využívaný pre aplikácie a ďalší virtuálny server pre databázu. Spoločnosť používa licencovaný Microsoft SQL Server 2014 Enterprise na správu databázových operácií. Zamestnanecké počítače sú vybavené operačným systémom Microsoft Windows 11 Pro, ktorý zabezpečuje plynulý chod počítača a pravidelné aktualizácie. Každý zamestnanecký počítač disponuje zakúpenou licenciou na kancelársky balíček Microsoft Office Professional 2019, ktorý obsahuje aplikácie ako Word, Excel, Outlook, OneNote, PowerPoint, Access a Skype.

Zálohovanie oboch virtuálnych serverov pomocou Veeam software sa vykonáva každý deň. Zálohované údaje oboch virtuálnych serverov sa ukladajú buď lokálne, na diskové pole pre zálohovanie, alebo off-line na NAS (Network Attached Storage), čo je dátové úložisko pripojené k sieti. Za softvérové vybavenie je taktiež zodpovedný správca siete.

2.5 Analýza databázy

2.5.1 Informácie o databáze

Databáza, ktorou som sa zaoberal, je využívaná ako úložisko dôležitých informácií nevyhnutných pre plynulý chod výrobných liniek, efektívne riadenie procesov a dobre naplánované výrobné postupy. Okrem toho je používaná ako nástroj na archiváciu údajov počas celého obdobia zmluvného vzťahu so zákazníkmi. Má sa na mysli uchovávanie informácií v oblasti automobilového priemyslu s časovým horizontom približne 20 rokov.

Táto databáza umožňuje nielen efektívne riadiť súčasné výrobné operácie, ale aj zhromažďovať a udržiavať dôležité údaje pre budúce referencie a analýzy. Vďaka nej je zabezpečené dodržiavanie zmluvných povinností a všetky potrebné informácie sú k dispozícii nielen počas súčasného obdobia, ale aj v dlhodobom horizonte.

Databáza je periodicky aktualizovaná o nové výrobné procesy, napríklad pri implementácii novej výrobnjej linky, alebo pridávaní nových tabuliek v prípade potreby vytvorenia nových úkonov. Tento dynamický prístup k úpravám a rozširovaniu databázy zabezpečuje, že je schopná efektívne podporovať rast a zmeny v rámci výrobných operácií spoločnosti.

Správa a programovanie databázy je zabezpečované programátormi, ktorí disponujú plným prístupom k databáze.

2.5.2 Stav databázy

Aktuálna databáza nebola dlhodobo optimalizovaná a preto možno od jej optimalizácie a odstránenia množstva nedostatkov očakávať celkové zlepšenie jej chodu. Pôvodná databáza bola prebratá z inej výrobnjej prevádzky rovnakej spoločnosti. Napriek tomu, že bola na začiatku čiastočne optimalizovaná, neskôr sa vykonávali len čiastočné zmeny, ako pridávanie nových tabuliek, indexov, atď.

V databáze sa nachádza množstvo tabuliek, ktoré sa počas viacerých rokov nepoužívajú a sú zastarané. Ich prítomnosť v databáze má negatívny vplyv na prehľadnosť a efektivitu celého systému.

Databáza môže mať problémy v dôsledku nesprávneho nakonfigurovania cudzích a primárnych kľúčov, ktoré môžu viesť k absencii väzieb medzi jednotlivými tabuľkami. Absencia väzieb môže spôsobiť redundanciu dát (duplicitné údaje), zlú integritu dát (hodnoty v tabuľke sú nesprávne odkázané na hodnoty v inej tabuľke), ako aj nesprávnosť aktualizácií a zmien.

Ďalším významným problémom databázy z pohľadu prehľadnosti a efektivity je kombinácia slovenských, anglických a nemeckých názvov atribútov a tabuliek.

Problematická je tiež názvová konvencia v databáze. Názvová konvencia by mala byť stanovená na začiatku životného cyklu aplikácie alebo databázy a mala by sa dôsledne dodržiavať. Prínosy používania takejto konvencie sú viaceré. Jednotné pravidlá pre pomenovanie premenných, tabuliek, stĺpcov a ďalších entít v rámci projektu prispievajú k zvýšenej čitateľnosti kódu a zlepšujú porozumenie štruktúre

databázy. Celý vývojový tím má jednoduchší prehľad a lepšie porozumie vzťahom medzi entitami.

Momentálne absentuje dokumentácia a popis databázy, čo predstavuje potenciálny problém v prípade výpadku informácií medzi programátormi v rámci spoločnosti. V situácii, kedy by došlo k incidentu alebo by bolo potrebné riešiť problém v databáze, neexistuje adekvátne dokumentácia na usmernenie personálu. To môže viesť k nepochopeniu štruktúry databázy a komplikovať riešenie problémov.

V situácii, keď do tímu vstupujú noví programátori alebo zamestnanci, zapríčiňuje absencia adekvátnej dokumentácie zníženie efektivity ich zaškolenia. Bez jasného popisu databázy sa nový personál môže stretávať s problémami pri porozumení fungovania a vzájomných vzťahov medzi rôznymi entitami v databáze.

V súčasnej konfigurácii je využívaný jediný účet na prístup do všetkých aplikácií databázy. V tomto prípade môže vzniknúť problém, kde aplikácia linky X, z dôvodu náhody alebo chyby v kóde, môže ohroziť dáta linky Y, pretože disponuje rovnakými právomocami. V prípade, že dochádza k zdieľaniu prístupových údajov, môže dôjsť k nesprávnym manipuláciám s údajmi a nepredvídateľným dôsledkom pre rôzne časti systému.

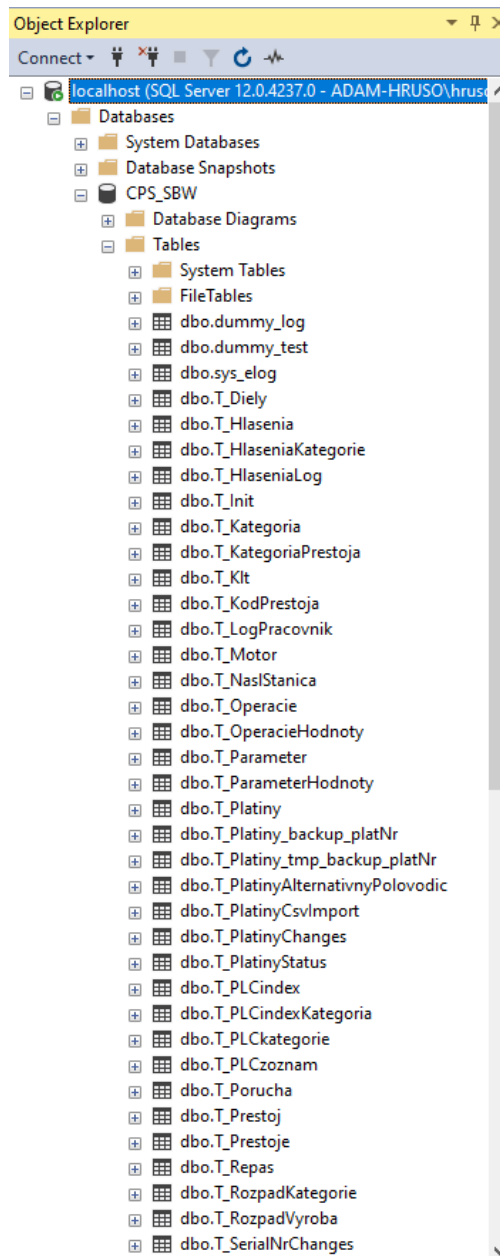
Zálohovanie dát na hodinovej báze môže predstavovať významný faktor, ktorý by mohol spôsobiť značné škody spoločnosti vo výrobe. Napríklad v prípade, že by došlo k zničeniu databázového disku, všetky dáta, ktoré neboli zálohované počas tejto hodiny, by mohli byť stratené. Týmto spôsobom môže nastať situácia, kde nezálohované informácie môžu mať negatívny vplyv na plynulý chod výrobných procesov a spôsobiť prerušenie v rámci aktuálnych operácií.

Tieto nedostatky predstavujú iba niektoré z problémov, ktoré je nevyhnutné riešiť a optimalizovať pre dosiahnutie efektívnejšieho a spoľahlivejšieho fungovania celej databázy. Celková situácia si vyžaduje dôkladnú analýzu a následné úpravy v štruktúre a obsahu databázy, aby sa dosiahol optimálny stav, ktorý bude zodpovedať aktuálnym potrebám spoločnosti.

2.5.3 Nevyužívané tabuľky

V databáze sa nachádza rozsiahle množstvo tabuliek, ktoré pochádzajú z obdobia niekoľkých rokov a momentálne nie sú aktívne využívané. Táto situácia má výrazný následok na značné spomalenie jej celkového fungovania. Sú to napríklad T_Prestoje,

T_KodPrestoje, T_Motor, a ďalšie.



Obrázok č. 15: Skrátený zoznam tabuliek (Zdroj: Vlastné spracovanie)

2.5.4 Nekorektné nakonfigurovanie primárnych a cudzích kľúčov

Primárne a cudzie kľúče boli vytvorené, avšak niektoré tabuľky sú nesprávne nakonfigurované, čo vedie k chybám vo vzťahoch medzi nimi. Napríklad v tabuľke T_Parameter sa nenachádza primárny kľúč, aj keď by ním mal byť ID_Parameter. Je nevyhnutné upraviť tieto tabuľky tak, aby bolo možné vytvárať správne vzťahy medzi nimi a zabezpečiť tak korektnú funkcionálnu celkovú databázu.

dbo.T_Parameter	
Columns	
ID_Parameter	(int, not null)
Hodnota	(sql_variant, not null)
ID_PLIndex	(int, not null)
ID_DBIndex	(int, null)
ID_DBIndex2	(int, null)
Od	(datetime, null)
Do	(datetime, null)
ID_Pracovnik	(int, null)

Obrázok č. 16: Tabuľka T_Parameter a jej atribúty (Zdroj: Vlastné spracovanie)

2.5.5 Rozdielny jazyk v tabuľkách

V databáze prevažujú tabuľky a ich atribúty s názvami v slovenskom jazyku. Paralelne s tým sa však v databáze objavujú tabuľky a atribúty s názvami v anglickom a nemeckom jazyku. Rovnako sa vyskytuje kombinácia anglických a slovenských názvov v jednej tabuľke. Pre dosiahnutie prehľadnosti a zlepšenia prepojenia vzťahov je vhodné mať všetky názvy v jednom jazyku. Jedným z príkladov je tabuľka T_PlatingCsvImport, kde atribúty sú v anglickom jazyku ako FileNameOriginal, DataProcessed a ďalšie, čo je len jeden z mnohých príkladov.

T_PlatingCsvImport	
ID_PlatingCsvImport	
FileNameInHistoryDir	
FileNameOriginal	
DateProcessed	
SuccessfullyImported	
NeedToWaitForPartNumbers	

Obrázok č. 17: Tabuľka s anglickými názvami (Zdroj: Vlastné spracovanie)

2.5.6 Názvová konvencia

Pri vytváraní a úprave tabuliek a atribútov databázy je dôležité riadiť sa existujúcimi štandardmi. V súčasnej databáze identifikujeme nedostatky v názvovej konvencii, napríklad použitie podčiarky miesto medzier v jednej tabuľke a absencia medzier v atribútoch v druhej tabuľke. Taktiež pozorujeme nekonzistentnosť v používaní veľkých písmen na začiatku atribútov v rôznych častiach databázy. Príklady zahŕňajú jasne definovaný formát primárneho kľúča v niektorých tabuľkách (napr.

ID_Vozik), zatiaľ čo iné tabuľky používajú jednoducho "id". Rovnaká konzistentnosť by mala byť zachovaná vo všetkých častiach databázy, aby sa dosiahla jednotnosť a lepšia prehľadnosť. Tieto aspekty je potrebné zohľadniť a zlepšiť s cieľom dosiahnuť optimálnu štruktúru a konzistentnosť celej databázy, čo by mohlo prispieť k efektívnejšiemu a spoľahlivejšiemu fungovaniu.

T Hlasenia	
ID_Hlasenia	
Text	
KodHlasenia	
ID_Stanica	
ID_Translatings	
ID_PLCzoznam	
Poradie	
zvyšovať_status_dielu	
nio_po_prvom_vyskyte	
Ukoncene	
srot_po_prvom_vyskyte	
ID_HlaseniaKategorie	

T Vozik	
ID_Vozik	
ID_Stanica	
ID_Diely	
Stav	
ID_ZmenaTypu	
ID_Klt1	
ID_Klt2	
ID_Klt3	
ID_Klt4	
ID_Klt5	
ID_Klt6	
ID_Klt7	
ID_Klt8	
ID_VyrobaOperacie1	
ID_VyrobaOperacie2	
ID_VyrobaOperacie3	
ID_VyrobaOperacie4	
ID_VyrobaOperacie5	
ID_VyrobaOperacie6	
ID_VyrobaOperacie7	
ID_VyrobaOperacie8	

Obrázok č. 18: Názvova konvencia pri tabuľkách T_Hlasenia a T_Vozik (Zdroj: Vlastné spracovanie)

2.5.7 Absencia dokumentácie databázy

V aktuálnej databáze absentuje akákoľvek forma dokumentácie, napríklad opis tabuliek, atribútov a zásad vzťahov. Toto predstavuje významný problém, pretože poskytnúť vysvetlenie týkajúce sa týchto častí sú schopní výlučne programátori z IT oddelenia a nikto iný. Vytvorenie dokumentácie a existencia popisného manuálu, by prispelo pri školení nových zamestnancov a zvýšilo efektívnosť práce s databázou.

2.5.8 Jeden prihlasovací účet do aplikácií databázy

Pri používaní jedného prihlasovacieho účtu pre prístup k aplikáciám databázy sa

môžu vyskytnúť značné bezpečnostné problémy. Problémová situácia môže nastať v prípade, keď aplikácia jednej linky chybou v kóde môže ohroziť dáta druhej linky, nakoľko disponuje rovnakými právomocami. Taktiež pri zdieľaní prístupových údajov môže dôjsť k ich zneužitiu a to by mohlo mať vážne následky pre integritu, dôvernosť a dostupnosť informácií.

2.5.9 Inkrementálne zálohovanie databázy na hodinovej báze

Aktuálne je implementované inkrementálne zálohovanie databázy na hodinovej báze. Hoci sa to môže zdať ako nepodstatný problém, v skutočnosti má výrazný dopad. V prípade poškodenia databázového disku (napríklad jeho zhorenie) by mohli byť všetky dáta z výroby za danú hodinu stratené. Tento incident by mohol spoločnosť stáť tisíce eur.

Riešenie uvedeného problému by bolo dosiahnuté implementáciou zrkadlenia diskov v diskovom poli RAID, čím by sa zabezpečila vyššia úroveň bezpečnosti dát a minimalizovalo by sa riziko straty informácií v prípade havárie diskového systému.

2.5.10 Zavedenie indexov

Návrh na implementáciu nových indexov v databáze pomocou funkcie "Odhadovaný plán vykonávania" v nástroji Microsoft SQL Management Studio a jeho zavedenie, by mohlo byť veľkým prínosom pre zvýšenie rýchlosti vyhľadávania a filtrácie dát. Vytvorenie týchto nových indexov bude tiež prínosom pre optimalizáciu a efektivitu dotazovania v rámci databázového systému.

2.6 Zhrnutie analýzy

Na základe analýzy a popisu všetkých problémov je zrejmé, že všetky opravy týchto nedostatkov by výrazne prispeli k optimalizácii chodu databázy. Odstránenie problémov, ako sú nevyužívané tabuľky a zle nakonfigurované primárne a cudzie kľúče, by boli značným prínosom k zefektívneniu databázových operácií.

Zosúladením odlišného jazyka a korekciou nesprávnej názvovej konvencie v tabuľkách by sa zvýšila prehľadnosť databázy.

Vytvorením návrhu dokumentácie databázy by sa po vytvorení samotnej dokumentácie zabezpečilo, že budú dostupné presné informácie o databáze a jej častiach.

Zvýšenie bezpečnosti databázy môže byť dosiahnuté odstránením problému jediného prihlasovacieho účtu do databázových aplikácií výroby, čo by viedlo k vytvoreniu viacerých účtov so špecifickými právomocami.

Na zrýchlenie vyhľadávania dát a filtrovania by bolo vhodné vytvoriť indexy, ktoré by boli prospešné pre výrobné procesy. Tieto kroky by prispeli k celkovej optimalizácii databázového systému.

3 VLASTNÝ NÁVRH RIEŠENÍ

V nasledujúcej kapitole budú predstavené návrhy na zlepšenie nedostatkov, ktoré boli zistené v predošlej kapitole analýzy súčasného stavu.

3.1 Odstránenie nevyužívaných tabuliek

Po vykonaní analýzy existujúcej databázy som identifikoval, že niektoré tabuľky, a to konkrétne T_Prestoje, T_KodPrestoje, T_Motor, T_Init, T_HlaseniaKategorie, T_Platiny_backup_platNr a T_Platiny_tmp_backup_platNr, nevykazujú aktívne využívanie v priebehu rozsiahlejšieho obdobia. Zistená skutočnosť má potenciálne negatívny vplyv na efektivitu a výkon databázového systému.

Po konzultáciách s programátormi z IT oddelenia mi bolo potvrdené, že uvedené tabuľky nie sú v súčasnej podobe databázy momentálne nevyhnutné a dlhší čas nie sú integrované do výrobného procesu spoločnosti. Je dôležité zdôrazniť, že jedna časť týchto tabuliek bola vytvorená na experimentovanie s databázovým prostredím, zatiaľ čo druhá časť slúžila pre špecifické potreby výrobných procesov.

Zálohovanie dát nie je potrebné riešiť, pretože dáta obsiahnuté v týchto tabuľkách boli predtým zálohované v rámci pravidelných procesov zálohovania databázy spoločnosti. S odstránením tabuliek prichádza riziko negatívneho vplyvu na existujúce aspekty, ako napríklad narušenie integrity, závislosti a podobne. Aktuálne procesy však nevykazujú závislosť na uvedených tabuľkách, čo znamená, že implementácia tohto zlepšenia by mala byť bezproblémová.

S cieľom minimalizovania rizík spojených s okamžitým vymazaním tabuliek boli dané tabuľky presunuté na iný databázový server. Tento krok bol podniknutý z dôvodu potenciálneho nebezpečenstva, že niektoré z týchto tabuliek môžu byť stále aktívne používané.

3.2 Úprava konfigurácie primárnych a cudzích kľúčov

V tabuľke T_Parameter sa síce nachádza atribút ID_Parameter, ktorý by sme automaticky považovali za primárny kľúč, ale nebol ako primárny kľúč definovaný, čo znamená, že neexistovala možnosť vytvárať nadväzujúce relačné vzťahy s údajmi uloženými v tejto tabuľke. Vzhľadom na túto skutočnosť bol vytvorený primárny kľúč s označením ID_Parameter s cieľom umožniť korektné identifikovanie záznamov v tejto

tabuľke a vytváranie relačných väzieb s inými entitami v rámci databázového systému.

	Column Name	Data Type	Allow Nulls
▶	ID_Parameter	int	<input type="checkbox"/>
	Hodnota	sql_variant	<input type="checkbox"/>
	ID_PLIndex	int	<input type="checkbox"/>
	ID_DIndex	int	<input checked="" type="checkbox"/>
	ID_DIndex2	int	<input checked="" type="checkbox"/>
	Od	datetime	<input checked="" type="checkbox"/>
	Do	datetime	<input checked="" type="checkbox"/>
	ID_Pracovnik	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Obrázok č. 19: Tabuľka T_Parameter (Zdroj: Vlastné spracovanie)

Rovnaký postup bol použitý aj pri zvyšných tabuľkách T_PlatingCsvImport a T_PlatingStatus. Táto oprava umožňuje zabezpečiť integritu údajov a poskytuje stabilnú identifikáciu jednotlivých záznamov v daných tabuľkách. Vytvára sa základ pre kvalitnú správu databázových vzťahov a zabezpečuje, že entitné vzťahy sú správne definované a udržateľné.

	Column Name	Data Type	Allow Nulls
▶	ID_PlatingCsvImport	int	<input type="checkbox"/>
	FileNameInHistoryDir	varchar(255)	<input type="checkbox"/>
	FileNameOriginal	varchar(255)	<input type="checkbox"/>
	DateProcessed	datetime	<input type="checkbox"/>
	SuccessfullyImported	bit	<input type="checkbox"/>
	NeedToWaitForPartNum...	bit	<input type="checkbox"/>
			<input type="checkbox"/>

Obrázok č. 20: Tabuľka T_PlatingCsvImport (Zdroj: Vlastné spracovanie)

	Column Name	Data Type	Allow Nulls
▶	ID_PlatingStatus	int	<input type="checkbox"/>
	Name	varchar(50)	<input checked="" type="checkbox"/>
	Description	varchar(255)	<input type="checkbox"/>
			<input type="checkbox"/>

Obrázok č. 21: T_PlatingStatus (Zdroj: Vlastné spracovanie)

3.3 Úprava jazyku databázy

V nasledujúcej časti bol iniciovaná oprava, ktorá bola zameraná na korekciu

názvov tabuliek, v ktorých bol použitý jazyk odlišný od slovenčiny. Účelom tejto opravy bolo dosiahnuť jednotu v jazykovom označovaní tabuliek v rámci existujúceho databázového systému.

Prebehla analýza všetkých existujúcich tabuliek s cieľom identifikovať názvy, ktoré nezodpovedali štandardom slovenského jazyka. Následne boli identifikované tabuľky, ktoré vykazovali takéto odchýlky a na základe toho boli podniknuté kroky na ich aktualizáciu. Zároveň boli tabuľky upravené aj podľa názvovej konvencie, ktorej zvyšné úpravy tabuliek sú v ďalšom podnadvise kapitoly.

Takýto jednotný prístup k pomenovávaniu tabuliek prispieva k ďalšej štruktúrovanej a systematickej organizácii v rámci databázového prostredia. Týmto spôsobom je posilnená celková kvalita a efektívnosť správy databázového systému.

Tabuľka T_TaktZeit

Tabuľka bola preložená do slovenského jazyka a premenovaná na T_CasCyklu spolu s jej atribútmi. Aplikovaný bol aj štandard názvovej konvencie.

Column Name	Data Type	Allow Nulls
ID_TaktZeit	int	<input type="checkbox"/>
ID_Stаницa	int	<input type="checkbox"/>
Od	datetime	<input checked="" type="checkbox"/>
do	datetime	<input checked="" type="checkbox"/>
ID_Pracovnik	int	<input checked="" type="checkbox"/>
ID_Vyroba	int	<input checked="" type="checkbox"/>
Preklopene	bit	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
ID_CasCyklu	int	<input type="checkbox"/>
ID_Stаницa	int	<input type="checkbox"/>
Od	datetime	<input checked="" type="checkbox"/>
Do	datetime	<input checked="" type="checkbox"/>
ID_Pracovnik	int	<input checked="" type="checkbox"/>
ID_Vyroba	int	<input checked="" type="checkbox"/>
Preklopene	bit	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Obrázok č. 22: Úprava tabuľky T_TaktZeit (Zdroj: Vlastné spracovanie)

Tabuľka T_PlatinyAlternativnyPolovodic

Atribúty tabuľky boli preložené do slovenského jazyka a tabuľka preložená na T_DoskaAlternativnyPolovodic.

Column Name	Data Type	Allow Nulls
ID_PlatinyAlternativnyPo...	int	<input type="checkbox"/>
KodPlatiny	nvarchar(30)	<input type="checkbox"/>
Poznamka	nvarchar(255)	<input type="checkbox"/>
InsertDate	datetime	<input checked="" type="checkbox"/>
FileName	nvarchar(256)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
ID_DoskaAlternativnyPol...	int	<input type="checkbox"/>
KodPlatiny	nvarchar(30)	<input type="checkbox"/>
Poznamka	nvarchar(255)	<input type="checkbox"/>
VlozDatum	datetime	<input checked="" type="checkbox"/>
NazovSuboru	nvarchar(256)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Obrázok č. 23: Úprava tabuľky T_PlatinyAlternativnyPolovodic (Zdroj: Vlastné spracovanie)

Tabuľka T_PlatinyCsvImport

Všetky atribúty v anglickom jazyku boli preložené do slovenského. Názov tabuľky premenovaný na T_DoskaCsvImport.

Column Name	Data Type	Allow Nulls		Column Name	Data Type	Allow Nulls
ID_PlatingCsvImport	int	<input type="checkbox"/>	➔	ID_DoskaCsvImport	int	<input type="checkbox"/>
FileNameInHistoryDir	varchar(255)	<input type="checkbox"/>		NazovSuboruHistorieDir	varchar(255)	<input type="checkbox"/>
FileNameOriginal	varchar(255)	<input type="checkbox"/>		NazovSuboruOriginal	varchar(255)	<input type="checkbox"/>
DateProcessed	datetime	<input type="checkbox"/>		SpracovaneData	datetime	<input type="checkbox"/>
SuccessfullyImported	bit	<input type="checkbox"/>		UspesneImportovane	bit	<input type="checkbox"/>
NeedToWaitForPartNum...	bit	<input type="checkbox"/>		CakatNaCisloSuciastky	bit	<input type="checkbox"/>
		<input type="checkbox"/>				<input type="checkbox"/>

Obrázok č. 24: Úprava tabuľky T_PlatingCsvImport (Zdroj: Vlastné spracovanie)

Tabuľka T_PlatingChanges

Tabuľka premenovaná na T_DoskaZmeny. Atribúty boli preložené z anglického do slovenského jazyka a taktiež bola aplikovaná názvová konvencia.

Column Name	Data Type	Allow Nulls		Column Name	Data Type	Allow Nulls
ID_PlatingChanges	int	<input type="checkbox"/>	➔	ID_DoskaZmeny	int	<input type="checkbox"/>
ID_Vyroba	int	<input type="checkbox"/>		ID_Vyroba	int	<input type="checkbox"/>
ID_Plating_old	int	<input checked="" type="checkbox"/>		ID_DoskaNove	int	<input checked="" type="checkbox"/>
ID_Plating_new	int	<input type="checkbox"/>		ID_DoskaStare	int	<input type="checkbox"/>
change_time	datetime	<input type="checkbox"/>		ZmenaCasu	datetime	<input type="checkbox"/>
		<input type="checkbox"/>				<input type="checkbox"/>

Obrázok č. 25: Úprava tabuľky T_PlatingChanges (Zdroj: Vlastné spracovanie)

Tabuľka T_PlatingStatus

Zmena názvu tabuľky na T_DoskaStav a taktiež aj preklad atribútov.

Column Name	Data Type	Allow Nulls		Column Name	Data Type	Allow Nulls
ID_PlatingStatus	int	<input type="checkbox"/>	➔	ID_DoskaStav	int	<input type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>		Nazov	varchar(50)	<input checked="" type="checkbox"/>
Description	varchar(255)	<input type="checkbox"/>		Popis	varchar(255)	<input type="checkbox"/>
		<input type="checkbox"/>			<input type="checkbox"/>	

Obrázok č. 26: Úprava tabuľky T_PlatingStatus (Zdroj: Vlastné spracovanie)

Tabuľka T_SerialNrChanges

Aplikovanie názvovej konvencie spolu s prekladom atribútov tabuľky. Premenenie tabuľky na T_ZmenaSerievehoC.

Column Name	Data Type	Allow Nulls		Column Name	Data Type	Allow Nulls
ID_SerialNrChanges	int	<input type="checkbox"/>	➔	ID_ZmenaSerievehoC	int	<input type="checkbox"/>
new_serial_nr	varchar(50)	<input type="checkbox"/>		NoveSerieveC	varchar(50)	<input type="checkbox"/>
original_serial_nr	varchar(50)	<input checked="" type="checkbox"/>		OriginalSerieveC	varchar(50)	<input checked="" type="checkbox"/>
ID_Vyroba	int	<input checked="" type="checkbox"/>		ID_Vyroba	int	<input checked="" type="checkbox"/>
ID_Pracovnik	int	<input checked="" type="checkbox"/>		ID_Pracovnik	int	<input checked="" type="checkbox"/>
ChangeTime	datetime	<input checked="" type="checkbox"/>		ZmenaCasu	datetime	<input checked="" type="checkbox"/>

Obrázok č. 27: Úprava tabuľky T_SerialNrChanges (Zdroj: Vlastné spracovanie)

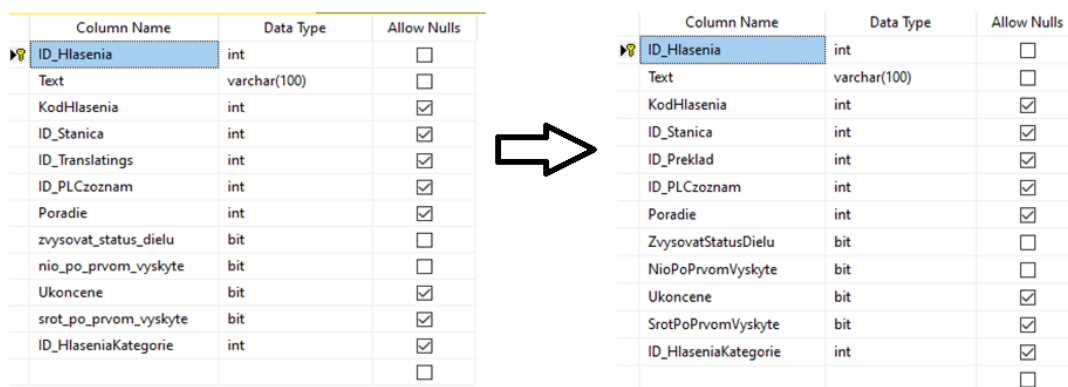
3.4 Úprava názvovej konvencie

Princíp úpravy názvovej konvencie spočíva v dosiahnutí jednotnej štylizácie a formátovania pre všetky tabuľky a ich atribúty v rámci databázového modelu. Tabuľky, u ktorých dochádzalo k zmene jazyka v predchádzajúcej časti, boli podrobené úpravám s cieľom zosúladiť ich názvy s ustanoveným štandardom názvovej konvencie. Avšak nie všetky tabuľky boli dotknuté týmto procesom, keďže boli spracované len tie, ktoré mali chyby v konfigurácii kľúčových väzieb. Zvyšné tabuľky budú spracované v tejto časti.

Po konzultáciách so správcom databázy bola zadefinovaná názvová konvencia, ktorá zabezpečí jednotné, zrozumiteľné a prehľadné označovanie databázových entít, pričom zohľadňuje existujúce zaužívané procesy v spoločnosti. Názvová konvencia vhodná pre databázovú štruktúru by mala byť riadená nasledujúcim štandardom. Každý názov tabuľky bude mať nasledujúci tvar – T_(Názov tabuľky) (napr. T_Hlasenia). Každý atribút s primárnym a cudzím kľúčom bude mať tvar - ID_(Názov kľúču) (napr. ID_Hlasenia). Začiatočné písmená sú vždy písané s veľkými písmenami. Medzery nie sú zahrnuté; namiesto nich sa každé nové slovo začína veľkým písmenom (napr. ZvysovatStatusDielu). Výnimkou sú názvy tabuliek, kde je medzera nahradená podčiarkovníkom (napr. T_Hlasenia) a pri atribútoch obsahujúcich primárny alebo cudzí kľúč (napr. ID_Hlasenia).

Zámerom týchto úprav je dosiahnuť konzistentný vizuálny dojem a jednotný štýl v rámci celého databázového modelu. Týmto spôsobom sa zabezpečuje, že názvy tabuliek a ich atribútov sú vyjadrené rovnakým spôsobom, čím sa zvyšuje prehľadnosť a súvislosť pri správe a údržbe databázy.

Tabuľka T_Hlasenia




Column Name	Data Type	Allow Nulls
ID_Hlasenia	int	<input type="checkbox"/>
Text	varchar(100)	<input type="checkbox"/>
KodHlasenia	int	<input type="checkbox"/>
ID_Stanica	int	<input checked="" type="checkbox"/>
ID_Translatings	int	<input checked="" type="checkbox"/>
ID_PLCzoznam	int	<input checked="" type="checkbox"/>
Poradie	int	<input checked="" type="checkbox"/>
zvysovat_status_dielu	bit	<input type="checkbox"/>
nio_po_prvom_vyskyte	bit	<input type="checkbox"/>
Ukoncene	bit	<input checked="" type="checkbox"/>
srot_po_prvom_vyskyte	bit	<input checked="" type="checkbox"/>
ID_HlaseniaKategorie	int	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
ID_Hlasenia	int	<input type="checkbox"/>
Text	varchar(100)	<input type="checkbox"/>
KodHlasenia	int	<input checked="" type="checkbox"/>
ID_Stanica	int	<input checked="" type="checkbox"/>
ID_Preklad	int	<input checked="" type="checkbox"/>
ID_PLCzoznam	int	<input checked="" type="checkbox"/>
Poradie	int	<input checked="" type="checkbox"/>
ZvysovatStatusDielu	bit	<input type="checkbox"/>
NioPoPrvomVyskyte	bit	<input type="checkbox"/>
Ukoncene	bit	<input checked="" type="checkbox"/>
SrotPoPrvomVyskyte	bit	<input checked="" type="checkbox"/>
ID_HlaseniaKategorie	int	<input checked="" type="checkbox"/>

Obrázok č. 28: Úprava tabuľky T_Hlasenia (Zdroj: Vlastné spracovanie)

Tabuľka T_Zasuvka

Column Name	Data Type	Allow Nulls
ID_Socket	int	<input type="checkbox"/>
Prikaz	varchar(20)	<input type="checkbox"/>
Parameter1	varchar(100)	<input checked="" type="checkbox"/>
Parameter2	varchar(100)	<input checked="" type="checkbox"/>
Parameter3	varchar(100)	<input checked="" type="checkbox"/>
Parameter4	varchar(100)	<input checked="" type="checkbox"/>
ID_Applications	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

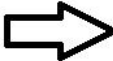


Column Name	Data Type	Allow Nulls
ID_Zasuvka	int	<input type="checkbox"/>
Prikaz	varchar(20)	<input type="checkbox"/>
Parameter1	varchar(100)	<input checked="" type="checkbox"/>
Parameter2	varchar(100)	<input checked="" type="checkbox"/>
Parameter3	varchar(100)	<input checked="" type="checkbox"/>
Parameter4	varchar(100)	<input checked="" type="checkbox"/>
ID_Aplikacie	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Obrázok č. 29: Úprava tabuľky T_Zasuvka (Zdroj: Vlastné spracovanie)

Tabuľka T_StanicaHotoveDiely

Column Name	Data Type	Allow Nulls
ID_StanicaHotoveDiely	int	<input type="checkbox"/>
ID_Stanica	int	<input type="checkbox"/>
ID_Kategoria	int	<input type="checkbox"/>
TypScript	int	<input checked="" type="checkbox"/>
ID_LinkaStanica	int	<input checked="" type="checkbox"/>
ID_LinkaKategoria	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>



Column Name	Data Type	Allow Nulls
ID_StanicaHotoveDiely	int	<input type="checkbox"/>
ID_Stanica	int	<input type="checkbox"/>
ID_Kategoria	int	<input type="checkbox"/>
TypSkriptu	int	<input checked="" type="checkbox"/>
ID_LinkaStanica	int	<input checked="" type="checkbox"/>
ID_LinkaKategoria	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Obrázok č. 30: Úprava tabuľky T_StanicaHotoveDiely (Zdroj: Vlastné spracovanie)

3.5 Návrh dokumentácie databázy

Vzhľadom na to, že s databázou pracuje tím programátorov a len oni vedia popísať jej časti a všetky funkcie, tak bolo navrhnuté riešenie, v ktorom sú popísané časti, ktoré by bolo vhodné aplikovať pri tvorbe dokumentácie.

Identifikácia databázových objektov

- **Tabuľky** – bude vytvorený zoznam tabuliek s príslušnými popismi a popismi ich stĺpcov, pričom primárne a cudzie kľúče budú špecifikované.
- **Pohľady** – existujúce pohľady budú zaznamenané a popísané.
- **Procedúry a funkcie** – všetky uložené procedúry a funkcie budú zdokumentované vrátane popisu a poskytnutia informácií o vstupných a výstupných parametroch.
- **Indexy** – všetky budú zaznamenané, spolu s ich významom a informáciami o stĺpcoch.

Dokumentácia verzii

- Vytvorenie textového dokumentu na firemnom intranete so stručným popisom vykonávaných zmien, dátumom zmeny a číslom verzie.

Školenie tímu

- Pre zamestnancov, ktorí pracujú s databázou, budú organizované školenia o význame vykonaných zmien (napr. názvoslovie, indexácia, relačné väzby).

Pravidelná aktualizácia a revízia

- Budú stanovené pravidlá na pravidelnú revíziu dokumentácie.
- V prípade každej softvérovej aktualizácie databázového prostredia, alebo operačného systému, bude databáza aktualizovaná.

Podpora pre nástroje na správu databáz

- Ak sa používajú nástroje na správu databáz (napr. Microsoft SQL Server Management Studio), bude zabezpečená integrácia dokumentácie priamo do týchto nástrojov.
- Pomocou integrovanej dokumentácie by bolo možné vo vybranom nástroji priamo vyhľadávať informácie o konkrétnych funkciách a postupoch.

3.6 Právomoci prístupu do aplikácií databázy

Vzhľadom na analýzou identifikovaný závažný problém, ktorým je existencia jediného prihlasovacieho účtu do aplikácií databázy, je navrhované vytvorenie viacerých účtov s rôznymi právomocami. Tie budú priradené zamestnancom, ktorí pracujú s aplikáciami databázy na základe ich potrieb pri vykonávaní práce.

Administrátor aplikácií

- Bude vytvorený účet s plným prístupom ku všetkým funkcionalitám aplikácií databázy vrátane konfigurácie a správy.
- Administrátorský účet bude využívať výhradne správca databázy a jej aplikácií.

Zamestnanec s plným prístupom

- Plný prístup k aplikáciám databázy bude umožnený, pričom nebude možné meniť ich konfigurácie.
- Účet bude určený primárne pre vedúcich výroby.

Zamestnanec s obmedzeným prístupom

- Užívateľský účet bude vytvorený s obmedzeným prístupom k určitým funkciám aplikácií databázy v závislosti od potrieb zamestnanca.

- Bude určený pre pracovníkov, ktorí pracujú na výrobných linkách, ktoré využívajú aplikácie databázy.

Len čitateľský účet

- Bude vytvorený účet so schopnosťou iba čítať údaje z aplikácií, bez možnosti meniť alebo mazať informácie.
- Určené pre kolegov z Nemecka (napr. pre vytvorenie analýzy výrobných procesov).

Zavedením týchto krokov bude umožnené vytvorenie rôznych účtov s prístupovými právami podľa potrieb. Týmto spôsobom bude dosiahnuté zvýšenie bezpečnosti a efektivity práce, keďže každý zamestnanec bude mať priradený účet s určitými právomocami.

3.7 Zálohovanie pomocou zrkadlenia diskov

V súčasnosti sa vykonáva inkrementálne zálohovanie databázy na hodinovej báze. Tento prístup môže predstavovať významný problém pre spoločnosť, pretože v prípade poškodenia disku môžu byť dáta z výroby stratené, ak neboli zálohované počas poslednej hodiny. V prípade vzniku takejto situácie by mohli byť pre firmu spôsobené obrovské škody v hodnote tisícoch eur.

Na riešenie tohto problému sa navrhuje implementovať redundantné pole nezávislých diskov, ďalej už len RAID, so zrkadlením (RAID 1). V tejto konfigurácii sa používajú aspoň dva disky, pričom sa dáta automaticky zrkadlia (kopírujú) na oba disky. V prípade poškodenia jedného disku sú dáta stále k dispozícii na druhom disku. Firma by nemusela čeliť strate dát z výroby za poslednú hodinu, pretože by mala k dispozícii duplikovanú kópiu dát na inom disku.

Operácie firmy môžu pokračovať bez prerušenia aj v prípade poruchy jedného z diskov. Oprava spočíva v tom, že sa poškodený disk nahradí novým a dáta sa automaticky obnovia zo zdravého disku. Takéto riešenie výrazne zvyšuje bezpečnosť dát a minimalizuje riziko straty informácií v prípade technických zlyhaní alebo havárií diskových jednotiek.

3.8 Vytvorenie indexov

INDEX IX_VyrobaHlasenia

Bol vytvorený náhodný dotaz na analýzu rýchlosti väzieb medzi entitami v databáze. Príkaz na vytvorenie dotazu je nasledovný.

```
use CPS_SBW;
SELECT s.ID_Stanica, v.SerioveC, hl.DatumCas, h.KodHlasenia, h.Text,v.ID_Diely,
p.Priezvisko + ' ' + p.Meno as MenoUzivatela,
      (SELECT COUNT (*) FROM T_HlaseniaZaznam hl JOIN T_Hlasenia h ON
(hl.ID_Hlasenia = h.ID_Hlasenia) WHERE ID_Vyroba = v.ID_Vyroba AND
h.KodHlasenia = h.KodHlasenia) as PocetSkusok
FROM
      T_HlaseniaZaznam hl
      JOIN T_Hlasenia h ON (hl.ID_Hlasenia = h.ID_Hlasenia)
      JOIN T_Vyroba v ON (hl.ID_Vyroba = v.ID_Vyroba)
      JOIN T_Stanica s ON (h.ID_Stanica = s.ID_Stanica)
      JOIN CPS_Firma.dbo.T_Pracovnik p ON (hl.ID_Pracovnik = p.ID_Pracovnik)
      LEFT OUTER JOIN CPS_Firma.dbo.T_Preklady t ON (h.ID_Preklady =
t.ID_Preklady)
      WHERE h.ID_Stanica in (11,12,16) order by DatumCas desc
```

Pomocou funkcie "Odhadovaný plán vykonávania" v nástroji Microsoft SQL Management Studio bolo identifikované, že vykonávanie dotazu je značne spomaľované a možnosť vytvorenia indexu by výrazne zvýšila jeho efektivitu o 25%. Príkaz na vytvorenie indexu je nasledovný.

```
USE [CPS_SBW]
GO
CREATE NONCLUSTERED INDEX [<IX_VyrobaHlasenia>]
ON [dbo].[T_HlaseniaZaznam] ([ID_Hlasenia])
INCLUDE ([ID_Vyroba])
GO
```

Implementáciou indexu bola rýchlosť vykonávania dotazu zlepšená z 4,3 sekundy na 3,3 sekundy.

INDEX IX_VyrobaOperacie

Pre lepšiu analýzu rýchlosti väzieb medzi entitami v databáze bol vytvorený ďalší dotaz, ktorého príkaz znie nasledovne.

```
use CPS_SBW;
SELECT DISTINCT A.SerioveC as SerioveCislo , A.GeneracnyStav as MakkaVerzia,
H.Nazov as Material, D.Nazov as NazovOperacie , D.ID_Operacie as IDOperacie ,
B.Hodnota as Hodnota ,F.KodHlasenia as KodChyby, G.Pocet as PocetSkusok ,
A.Koniec as Ukoncenie , B.Datum as DatumSkusky
FROM T_Vyroba A LEFT JOIN T_VyrobaOperacie B ON
(A.ID_Vyroba = B.ID_Vyroba)
LEFT JOIN T_OperacieHodnoty C ON(C.ID_OperacieHodnoty =
B.ID_OperacieHodnoty)
LEFT JOIN T_Operacie D ON(D.ID_Operacie = C.ID_Operacie)
LEFT JOIN T_HlaseniaZaznam E ON (E.ID_Vyroba =
A.ID_Vyroba)
LEFT JOIN T_Hlasenia F ON (F.ID_Hlasenia = E.ID_Hlasenia)
LEFT JOIN (select CC.SerioveC ,CC.ID_Vyroba, DD.Pocet from
T_Vyroba CC
LEFT JOIN
(select AA.ID_Vyroba , COUNT(BB.ID_Vyroba) as Pocet from
T_Vyroba AA
LEFT JOIN CPS_Firma.dbo.T_SkuskyNaEol BB ON
(AA.ID_Vyroba = BB.ID_Vyroba)
GROUP BY AA.ID_Vyroba) DD ON (CC.ID_Vyroba =
DD.ID_Vyroba)) G ON (G.ID_Vyroba = A.ID_Vyroba)
LEFT JOIN T_Diely H ON (A.ID_Diely = H.ID_Diely)
WHERE B.ID_Stanica in(1,2)
order by A.SerioveC , B.Datum desc , D.Nazov
```

Funkcia "Odhadovaný plán vykonávania" v Microsoft SQL Management Studiu bude opäť využitá. Na základe tejto funkcie bolo zistené, že dotaz by mohol byť zrýchlený o 17% vytvorením indexu. Príkaz na vytvorenie indexu je nasledovný.

```
USE [CPS_SBW]
GO
CREATE NONCLUSTERED INDEX [<IX_VyrobaOperacie>]
ON [dbo].[T_VyrobaOperacie] ([ID_Stanica])
INCLUDE ([ID_Vyroba],[ID_OperacieHodnoty],[Hodnota],[Datum])
GO
```

Implementáciou indexu sme zistili, že rýchlosť vykonávania dotazu sa zmenila z 5,1 sekúnd na 4,2 sekundy.

Vložením týchto indexov do databázy sa jej chod a výkon výrazne zlepšia. Vyhľadávacie operácie a dotazy sa zefektívnia, čo povedie k rýchlejšiemu získavaniu potrebných dát. Spracovanie dotazov sa taktiež optimalizuje a celková odozva systému sa zlepší. Riziko nastáva v prípade veľkých tabuliek, tabuliek s rýchlo rastúcim objemom a komplikovanými vzťahmi, kedy sa môže index zväčšiť a spomaliť.

3.9 Zhodnotenie návrhu

Návrh zlepšení predstavuje prístup k optimalizácii a zefektívneniu existujúcej databázy. Odstránením nevyužívaných tabuliek sa zjednoduší štruktúra databázy a eliminuje sa zbytočný dátový objem. Úpravou konfigurácie primárnych a cudzích kľúčov sa zabezpečia správne vzťahy medzi tabuľkami, čo má pozitívny vplyv na integritu dát a efektivitu operácií.

Zavedením jednotného jazyka v databáze a úpravou názvovej konvencie sa zlepši čitateľnosť a udržateľnosť kódu. Definovaním krokov pre spracovanie dokumentácie sa zabezpečí, že po vytvorení dokumentácie budú dostupné presné informácie o databáze, čo je kľúčové pre ďalšiu údržbu a rozvoj.

Rozdelením právomocí prístupu do aplikácií databázy na viaceré úrovne (administrátorky účet, zamestnanecký účet s plnými právomocami, atď.) sa zlepši bezpečnosť a riadenie prístupu, čo je dôležité pre ochranu citlivých informácií. Zavedením nového zálohovania s využitím zrkadlenia RAID sa minimalizuje strata dát v prípade havárie disku.

Vytvorením indexov sa zvyšuje efektivita vyhľadávacích operácií a prispieva sa k rýchlejšiemu vykonávaniu dotazov. Celkovo vzaté, táto optimalizácia má veľmi pozitívny vplyv na výkon, bezpečnosť a udržateľnosť databázy.

ZÁVER

V analýze súčasného stavu sme zistili určité nedostatky, ktoré spôsobujú menej efektívnu prevádzku databázy. Cieľom tejto bakalárskej práce bolo optimalizovať databázu výrobnjej spoločnosti a tým zefektívniť jej prevádzku.

Prvá časť práce sa zaoberá teoretickými poznatkami, ktoré slúžia ako podklad pre zvyšné dve časti.

V druhej časti sa práca zaoberá analýzou súčasného stavu. Analýza sa prevažne zameriava na stav databázy a jej nedostatkov, ale taktiež sa venuje základným informáciám o spoločnosti, hardvéru, softvéru a organizačnej štruktúre.

Tretia časť sa zameriava na samotný návrh riešení nájdených nedostatkov v analýze súčasného stavu. V databáze boli odstránené nevyužívané tabuľky, taktiež bola upravená konfigurácia primárnych kľúčov. Upravená bola i názvová konvencia databázy na jednotný štýl a jazyk databázy zmenený na slovenský. Venoval som sa taktiež návrhu dokumentácie databázy, vytvoreniu právomocí prístupu do aplikácií databázy a zálohovaniu pomocou zrkadlenia diskov. Posledným vylepšením bolo vytvorenie indexov.

Navrhnuté zlepšenia by mali zefektívniť prevádzku databázy, taktiež zlepšiť jej prehľadnosť, jednotnosť a bezpečnosť.

ZOZNAM POUŽITÝCH ZDROJOV

- [1] Advantages and Disadvantages of Normalization. Piping Mart [online]. [cit. 2023-11-30]. Dostupné z: <https://blog.thepipingmart.com/other/advantages-and-disadvantages-of-normalization/>.
- [2] Database Documentation – Lands of Trolls: Why and How. Redgate [online]. [cit. 2024-03-06]. Dostupné z: <https://www.red-gate.com/simple-talk/databases/sql-server/database-administration-sql-server/database-documentation-lands-of-trolls-why-and-how/>.
- [3] Disk mirroring (RAID 1). TechTarget [online]. [cit. 2024-03-06]. Dostupné z: <https://www.techtarget.com/searchstorage/definition/disk-mirroring>.
- [4] Display an Actual Execution Plan. Microsoft [online]. [cit. 2024-04-16]. Dostupné z: <https://learn.microsoft.com/en-us/sql/relational-databases/performance/display-an-actual-execution-plan?view=sql-server-ver16>.
- [5] Guide to SQL Server Permissions. Netwrix [online]. [cit. 2024-03-05]. Dostupné z: <https://blog.netwrix.com/2020/10/16/guide-to-sql-server-permissions/>.
- [6] KOCH, Miloš a Bernard NEUWIRTH. *Datové a funkční modelování*. Vyd. 4., rozš. Brno: Akademické nakladatelství CERM, 2010. ISBN 978-80-214-4125-5.
- [7] KŘÍŽ, Jiří a Petr DOSTÁL. *Databázové systémy*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3064-8.
- [8] LACKO, Ľuboslav. *Jak vyzrát na Microsoft SQL Server 2008: správa, konfigurace, programování*. Brno: Computer Press, 2009. ISBN 978-80-251-2101-6.
- [9] Learn SQL: Naming Conventions. SQLShack [online]. [cit. 2023-12-02]. Dostupné z: <https://www.sqlshack.com/learn-sql-naming-conventions/>.
- [10] MADRON, Lukáš. *Datové sklady a OLAP v prostředí MS SQL Serveru* [online]. Brno, 2008 [cit.2023-11-29]. Diplomová práce. VUT v Brně, Fakulta informačních technologií. Ing. VLADIMÍR BARTÍK, Ph.D. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=116351.

- [11] OTTE, Lukáš. *DATABÁZOVÉ SYSTÉMY*. Vysoká škola báňská – Technická univerzita Ostrava, 2013. [online]. [cit. 2023-29-11]. Dostupné z: https://projekty.fs.vsb.cz/463/edubase/VY_01_044/Databázové%20systémy.pdf.
- [12] POKORNÝ, Jaroslav a Michal VALENTA. *Databázové systémy*. 2. přepracované vydání. Praha: Česká technika - nakladatelství ČVUT, 2020. ISBN 978-80-01-06696-6.
- [13] Schéma organizačnej štruktúry spoločnosti XY s.r.o, 2024. Vlastná tvorba na základe interného dokumentu.
- [14] Schéma zapojenia hardvérového vybavenia hál spoločnosti XY s.r.o, 2024.
- [15] STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES. *Naučte se SQL za 28 dní: [stačí hodina denně]*. Brno: Computer Press, 2010. ISBN 978-80-251-2700-1.
- [16] What is an Execution Plan in SQL Server and How to Use It? TatvaSoft [online]. [cit. 2024-03-13]. Dostupné z: <https://www.tatvasoft.com/blog/optimize-sql-query/>.
- [17] What is SQL Server Management Studio (SSMS). Microsoft [online]. [cit. 2023-12-01]. Dostupné z: <https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>.
- [18] What is SQL Server. Microsoft [online]. [cit. 2023-11-30]. Dostupné z: <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16>.

ZOZNAM POUŽITÝCH OBRÁZKOV

Obrázok č. 1: Informácia (Zdroj: [6] s. 4)	12
Obrázok č. 2: Dáta (Zdroj [6] s. 5).....	13
Obrázok č. 3: Databázový systém (Zdroj: [11] s. 8).....	17
Obrázok č. 4: Typy definícií množín pomocou ich prvkov (Zdroj: [6] s. 24).....	22
Obrázok č. 5: Typy vzťahov v množinách (Zdroj: [6] s. 25).....	23
Obrázok č. 6: Popis relačnej schémy (Zdroj: [6] s. 26).....	23
Obrázok č. 7: Príklady jednotlivých kľúčov (Zdroj: [6] s. 29).....	26
Obrázok č. 8: Vzťah 1:1 (Zdroj: [6] s. 31)	26
Obrázok č. 9: Vzťah 1:N (Zdroj [6] s. 31).....	27
Obrázok č. 10: Vzťah N:M (Zdroj: [6] s. 32)	27
Obrázok č. 11: MS SQL Server Management Studio (Zdroj: Vlastné spracovanie)	32
Obrázok č. 12: Organizačná schéma spoločnosti (Zdroj: Vlastné spracovanie)	34
Obrázok č. 13: Schéma hardwarového vybavenia haly 2 (Zdroj: [14])	35
Obrázok č. 14: Schéma hardwarového vybavenia haly 3 (Zdroj: [14])	35
Obrázok č. 15: Skrátенý zoznam tabuliek (Zdroj: Vlastné spracovanie).....	39
Obrázok č. 16: Tabuľka T_Parameter a jej atribúty (Zdroj: Vlastné spracovanie).....	40
Obrázok č. 17: Tabuľka s anglickými názvami (Zdroj: Vlastné spracovanie).....	40
Obrázok č. 18: Názvov konvencia pri tabuľkách T_Hlasenia a T_Vozik (Zdroj: Vlastné spracovanie)	41
Obrázok č. 19: Tabuľka T_Parameter (Zdroj: Vlastné spracovanie).....	45
Obrázok č. 20: Tabuľka T_PlatinCsvImport (Zdroj: Vlastné spracovanie)	45
Obrázok č. 21: T_PlatinStatus (Zdroj: Vlastné spracovanie)	45
Obrázok č. 22: Úprava tabuľky T_TaktZeit (Zdroj: Vlastné spracovanie)	46
Obrázok č. 23: Úprava tabuľky T_PlatinAlternativyPolovodic (Zdroj: Vlastné spracovanie)	46
Obrázok č. 24: Úprava tabuľky T_PlatinCsvImport (Zdroj: Vlastné spracovanie)	47
Obrázok č. 25: Úprava tabuľky T_PlatinChanges (Zdroj: Vlastné spracovanie)	47
Obrázok č. 26: Úprava tabuľky T_PlatinStatus (Zdroj: Vlastné spracovanie)	47
Obrázok č. 27: Úprava tabuľky T_SerialNrChanges (Zdroj: Vlastné spracovanie)	47
Obrázok č. 28: Úprava tabuľky T_Hlasenia (Zdroj: Vlastné spracovanie).....	48

Obrázok č. 29: Úprava tabuľky T_Zasuvka (Zdroj: Vlastné spracovanie)	49
Obrázok č. 30: Úprava tabuľky T_StаницаHotoveDiely (Zdroj: Vlastné spracovanie) ..	49