



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

ANALÝZA SOUČASNÉ DATABÁZE A NÁVRH NA JEJÍ VYLEPŠENÍ PRO MALOU SPOLEČNOST

ANALYSIS OF THE CURRENT DATABASE AND ITS PROPOSAL FOR IMPROVEMENT FOR A SMALL
COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Sebastián Badura

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Kříž, Ph.D.

BRNO 2024

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Sebastián Badura**
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:

Analýza současné databáze a návrh na její vylepšení pro malou společnost

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 na základě analýzy stávající databáze navrhnout její úpravy pro efektivnější správu dat.

Základní literární prameny:

KOCH, Miloš. Datové a funkční modelování. Brno: Akademické nakladatelství CERM, 2004. ISBN 80-214-2724-8.

KROENKE, David; AUER, David J a GONER, Jakub. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

LACKO, Ľuboslav. Mistrovství v SQL Server 2012. Brno: Computer Press, 2013. ISBN 978-80-251-3773-4.

OPPEL, Andrew J. SQL bez předchozích znalostí: [průvodce pro samouky]. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.

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

Táto záverečná práca sa zaoberá analýzou databázového systému spoločnosti a na jej základe následnom návrhu databáze. Spoločnosť, ktorej je navrhovaná databáza sa zaoberá montážou a servisom garážových brán a pohonov. Prvá časť záverečnej práce sa zaoberá teoretickými východiskami a približuje problematiku databáze. V nasledujúcej časti sa analyzuje súčasný stav databáze vo vybranej spoločnosti. Tretia časť vychádza z druhej a obsahuje vlastný návrh databáze pre túto spoločnosť.

Kľúčové slová

databáza, databázový systém, dáta, dátový model, ER diagram, MS SQL Server, SQL

Abstract

This thesis deals with the analysis of the company's database system and the subsequent database design based on the analysis. The company whose database is being designed is engaged in the installation and servicing of garage doors and operators. The first part of the thesis deals with the theoretical background and introduces the issues of the database. The following chapter analyses the current status of the database in the selected company. The third part builds on the second and contains the actual database design for this company.

Key words

database, database system, data, data model, ER diagram, MS SQL Server, SQL

Bibliografická citácia

BADURA, Sebastián. *Analýza současných databází a návrh na její vylepšení pro malou společnost*. Brno, 2024. Dostupné také z: <https://www.vut.cz/studenti/zav-prace/detail/160744>.

Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky.
Vedoucí práce Jiří Kříž.

Čestné prehlásenie

Prehlasujem, že predložená bakalárska práca je pôvodná a spracoval som ju samostatne. Prehlasujem, že citácie použitých prameňov je úplná, že som vo svojej práci neporušil autorské práva (v zmysle Zákona č. 121/2000 Sb., o právu autorskom a o právach súvisiacich s právom autorským).

V Brne dňa 8.5.2024

.....
podpis autora

Pod'akovanie

Chcem sa v prvom rade pod'akovať vedúcemu bakalárskej práce Ing. Jiřímu Křížovi, Ph.D. za pomoc a konzultovanie pri vypracovávaní práce. Ďalšia vďaka patrí rodičom a bratovi za psychickú a aj finančnú podporu, ktorú mi poskytli pri mojom štúdiu. V neposlednom rade sa chcem pod'akovať spoločnosti Saver, ktorá mi umožnila s nimi spolupracovať pri vyhotovovaní záverečnej práce a jednoduchej komunikácií s nimi.

Obsah

Úvod	12
Cieľ a metodika práce	13
1 Teoretické východiská	14
1.1 Základné pojmy	14
1.1.1 Dáta	14
1.1.2 Informácie	14
1.1.3 Znalosti	15
1.2 Databázový systém	15
1.2.1 Databáza	15
1.2.2 Systém riadenia databáze	15
1.2.3 Databázová aplikácia	16
1.3 Dátové modely	16
1.3.1 Lineárny dátový model	17
1.3.2 Hierarchický dátový model	17
1.3.3 Sieťový dátový model	18
1.3.4 Relačný dátový model	19
1.3.5 Objektový dátový model	19
1.4 Relačný dátový model	19
1.4.1 Základné pojmy	20
1.4.2 Relácie (vzťahy)	21
1.4.3 Typy kľúčov	21
1.4.4 Integrita relačného modelu	22
1.4.5 Normalizácia	24
1.5 SQL	25
1.5.1 Kategórie príkazov SQL	26
1.5.2 Dátové typy	26

1.5.3	Syntax základných príkazov.....	27
1.5.4	Spojovanie tabuliek príkazom JOIN	29
1.5.5	Agregačné funkcie.....	30
1.5.6	Pohľady	30
1.5.7	Procedúry	31
1.5.8	Spúšťače	31
1.6	Metodológia návrhu databázy	31
1.6.1	Konceptuálny návrh databázy	32
1.6.2	Logický návrh databáze	32
1.6.3	Fyzický návrh databázy.....	32
2	Analýza súčasného stavu.....	34
2.1	Základné informácie o firme	34
2.2	Organizačná štruktúra.....	35
2.3	Hardware	36
2.4	Software	36
2.5	Informačný systém	37
2.6	Databáza	37
2.7	SWOT analýza databáze	42
2.7.1	Zhodnotenie SWOT analýzy	42
2.8	EPC diagram	44
2.8.1	Slovný popis EPC diagramu	46
2.9	Zhodnotenie spoločnosti	47
3	Vlastný návrh riešenia	48
3.1	Zadanie od spoločnosti.....	48
3.2	Konceptuálny návrh	48
3.2.1	Identifikácia entít.....	49
3.2.2	Identifikácia relácií.....	49

3.2.3	Určenie kardinality	50
3.3	Logický návrh	50
3.3.1	Tabuľka Zamestnanec	51
3.3.2	Tabuľka Vyslaný	51
3.3.3	Tabuľka Zákazka	52
3.3.4	Tabuľka Stav_zak	54
3.3.5	Tabuľka Zákazník	54
3.3.6	Tabuľka Zak_udaje	55
3.3.7	Tabuľka Tovar	56
3.3.8	Tabuľka Sklad	57
3.3.9	Tabuľka Dodávateelia	58
3.3.10	Tabuľka Dod_udaje	58
3.3.11	ER Diagram	59
3.4	Fyzický návrh	61
3.4.1	Výber prostredia databázy a tvorba databázy	61
3.4.2	Tvorba tabuliek a cudzích kľúčov	61
3.4.3	Vloženie dát	62
3.4.4	Triggery	62
3.4.5	Pohľady	63
3.4.6	Procedúry	66
3.5	Zhodnotenie navrhutej databázy	68
	Záver	69
	Zoznam použitých zdrojov	70
	Zoznam použitých obrázkov	72
	Zoznam použitých tabuliek	74
	Zoznam použitých skratiek	75
	Zoznam príloh	76

Príloha č. I: Tvorba tabuliek	77
Príloha č. II: Vloženie testovacích dát	82
Príloha č. III: Triggers.....	87
Príloha č. IV: Pohľady	89
Príloha č. V: Procedúry	91

Úvod

Už od nepamäti bolo pre ľudstvo veľmi dôležité uchovávanie informácií, či sa jednalo o hlinené tabuľky, pergameny alebo neskôr aj knihy. Už vtedy si chceli určité informácie zapísať a prenechať pre ďalšie generácie, ktoré ich mohli využiť. Tieto záznamy boli nielen zdrojom poznania, ale aj pripomenutie minulosti a dedičstvom pre budúcnosť.

V súčasnosti, v dobe digitálnej revolúcie, dosiahli uchovávanie a spracovanie dát úplne novú úroveň. Moderné technológie umožňujú ľudstvu zbierať, uchovávať a analyzovať obrovské množstvo dát v rýchlosti, akú si nikdy predtým ani len nevedeli predstaviť. V tejto súvislosti zohrávajú databázové systémy kľúčovú úlohu.

Databázové systémy dnes nie sú len statickými úložiskami dát, ale dynamickými nástrojmi, ktoré pomáhajú organizáciám efektívne riadiť svoje procesy a analyzovať svoje informácie. Za posledné roky prešli tieto systémy výraznou obmenou a modernizáciou, aby sa prispôbili potrebám súčasného podnikateľského prostredia. Ich úloha v rámci spoločnosti sa stala ešte významnejšou, pretože správne navrhnutý a implementovaný databázový systém môže byť kľúčom k efektívnemu riadeniu spoločnosti, zvyšovaniu produktivity a konkurencieschopnosti firmy.

Cieľom tejto záverečnej práce je priblížiť problematiku databáze a následne ju analyzovať vo vopred vybranej spoločnosti. Súčasťou analýzy terajšieho databázového systému spoločnosti je aj objavenie jeho chýb, nedostatkov a navrhnutie vylepšení pre jej efektívnejšie fungovanie. Tretia časť záverečnej práce sa zaoberá navrhnutím vlastného databázového systému, ktorý má splňať požiadavky spoločnosti. Na jeho vyhotovenie sa vychádza z analytickej časti, kde boli definované aj priestory na vylepšenie.

Cieľ a metodika práce

Cieľom práce je na základe analýzy stávajúcej databáze navrhnuť jej úpravu pre efektívnejšiu správu dát.

Výsledkom záverečnej práce je navrhnutá plne funkčnú databáza, ktorá bude spĺňať požiadavky spoločnosti a zefektívni súčasný stav databáze spoločnosti. Analýza súčasného stavu prebieha na základe konzultácií so zamestnancami spoločnosti a osobnej skúsenosti získanej na praxi v tejto spoločnosti. S využitím získaných poznatkov analytickej časti, ktorej výsledkom sú nedostatky a možné vylepšenia databáze nasleduje vlastný návrh databáze.

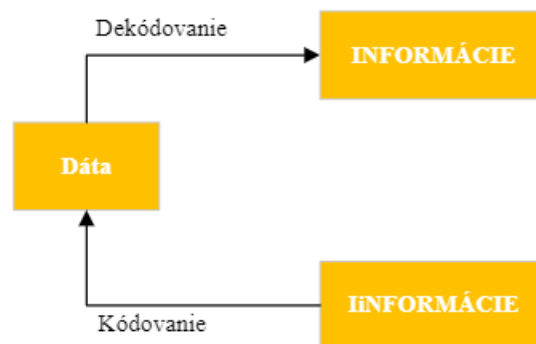
Praktická časť záverečnej práce obsahuje konkrétny návrh databáze, ktorý vychádza z predošlej časti. Tento návrh má spĺňať všetky požiadavky spoločnosti a má vypomôcť pri interných procesoch spoločnosti.

1 Teoretické východiská

1.1 Základné pojmy

1.1.1 Dáta

Dáta sa môžu nachádzať v akejkoľvek forme textu, obrázku, čísla, grafu alebo ostatných. Predstavujú surové nespracované fakty, ktoré majú určitý význam ale človek ich momentálne nepoužíva na analýzu alebo rozhodovanie. S dátami môže človek pracovať rôznymi spôsobmi – uložiť na neskoršie spracovanie, transformovať do iných foriem, či už na papier alebo aj do počítača. Pri týchto činnostiach sa dáta kódujú (informácie sa stávajú dátami) a dekódujú (dáta sa menia na informácie). [1]



Obrázok 1: Kódovanie a dekódovanie dát [1]

1.1.2 Informácie

Informácie je možné vyjadriť ako znalosti získané štúdiom, komunikáciou, výskumom alebo výukou. Sú to výstupy analýzy a interpretácie dát. Zabezpečujú subjektu zníženie rozhodovacej nerozhodnosti. Sú taktiež chápané ako správa, ktorá spĺňa tri požiadavky:

- syntetická relevancia – prijímací subjekt musí byť schopný ju detegovať a porozumieť,
- sémantická relevancia – subjekt musí vedieť, čo správa znamená, čo vypovedá o ňom a jeho okolí,
- pragmatická relevancia – správa musí mať pre prijímací subjekt význam. [1,2]

1.1.3 Znalosti

Ak dané informácie vie osoba prakticky hneď využiť, tak sa jedná o znalosti. Najčastejším zdrojom znalostí sú skúsenosti, no môže sa nadobúdať aj učením, praxou alebo štúdiom. [3]

Taktiež sú znalostí považované za informácie o tom, ako využiť iné informácie. To ovplyvňuje dennodenné procesy rozhodovania. Preto prax aj literatúra tvrdí, že vo viacerých prípadoch je dôležitejšie rýchle rozhodnutie ako to kvalitnejšie. Takže doba na rozhodnutie by nemala prekročiť čas, aký zaberie problém. [1]

1.2 Databázový systém

Databázový systém pomáha jeho užívateľom zhromažďovať, ukladať, modifikovať a prehliadať veľké množstvo dát, ktoré sú súčasťou databáze. Takéto dáta sú vo väčšine prípadov modelované v riadkoch a stĺpcoch v sérii tabuliek, ktoré sú navzájom prepojené. Takáto forma dát zabezpečuje zefektívnenie spracovania a vyhľadávania údajov.

Databázový systém sa skladá z viacerých komponentov a to užívateľov, databázovú aplikáciu, systém riadenia databáze (DBMS) a vlastnú databázu. [4,5,6]

1.2.1 Databáza

Databáza je usporiadaný súbor štruktúrovaných informácií alebo dát, zvyčajne uložených elektronicky v počítačovom systéme. [6]

Slúži na ukladanie, údržbu a prístupu k akémukoľvek druhu dát. Zbierajú informácie o osobách, miestach alebo veciach, ktoré sú uložené na jedno miesto, aby ich bolo možné sledovať a analyzovať. [6]

V počítačových databázach prevažne ukladajú dátové záznamy, ktoré obsahujú informácie o predaji, zákazníkoch, finančné údaje, informácie o produktoch, skladové informácie a iné. [7]

Databáza nemusí mať len formu v počítačovom systéme. Môže to byť aj napríklad kartotéka, kde sa uskladnia všetky potrebné dokumenty podľa určitých pravidiel. [8]

1.2.2 Systém riadenia databáze

Hlavným cieľom systému riadenia databáze je vytvárať, spracovávať, upravovať a spravovať databázy. Jedná sa o zložitý a komplexný softvér, ktorý si vo väčšine prípadov spoločnosť,

ktorá chce viesť databázu, zakupuje u dodávateľa formou licencie. Medzi najznámejšie takéto produkty patria:

- Microsoft SQL Server od spoločnosti Microsoft
- MySQL od Oracle Corporation
- Oracle Database od Oracle Corporation
- DB2 od spoločnosti IBM
- Microsoft Access od spoločnosti Microsoft

Na splnenie už spomínaných cieľov je potrebné aby systém riadenia databáze poskytoval určité funkcie. Presne sa jedná o tieto funkcie:

- vytvorenie databáze
- vytvorenie tabuliek
- vytvorenie podporných štruktúr
- čítanie dát z databáze
- úpravy (vkladanie, aktualizácia alebo odstránenie) databázových dát
- údržba databázových štruktúr
- vynucovanie pravidiel
- kontrola súbežnosti
- zaistenie bezpečnosti
- zálohovanie a obnovenie

Na plnenie väčšinu týchto funkcií systém prijíma príkazy v jazyku SQL (Structured Query Language) a v inom formáte a prevádza ich na akcie s databázovými súbormi. [4]

1.2.3 Databázová aplikácia

Databázová aplikácia je jeden alebo viac počítačových programov, ktoré slúžia na prepojenie užívateľov spolu so systémom riadenia databáze. Aplikáčne programy na čítanie a úpravu databázových dát využívajú odosielanie príkazov v SQL jazyku systému riadeniu databáze. Okrem toho aplikácie ešte znázorňujú užívateľom dáta v podobe formulárov alebo zostáv. [4]

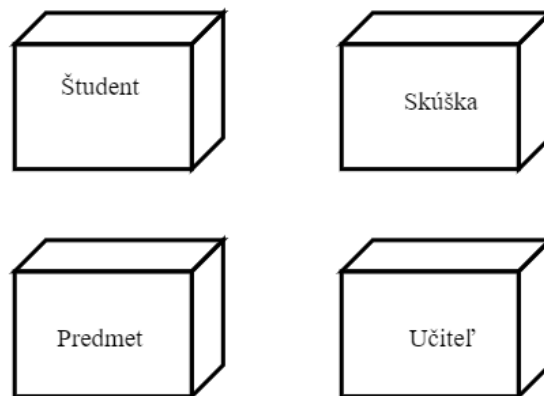
1.3 Dátové modely

Dátový model označuje dátovú štruktúru, ktorá sa využíva na organizáciu a správu údajov v databáze. Pomáhajú definovať a štruktúrovať údaje, ktoré sú potrebné pre spoločnosti.

Umožňujú spoločnostiam rozhodovať o tom, ako sa budú údaje ukladať, sprístupňovať, upravovať a využívať v rámci organizácie. [8,9]

1.3.1 Lineárny dátový model

Každý dátový objekt lineárneho dátového modelu znázorňuje jeden súbor s vetami o príslušných objektoch v pevnej štruktúre. V tomto dátovom modeli sa nenachádzajú žiadne väzby, takže ani nie je možné určiť ako jednotlivé dátové objekty spolu súvisia. [1]



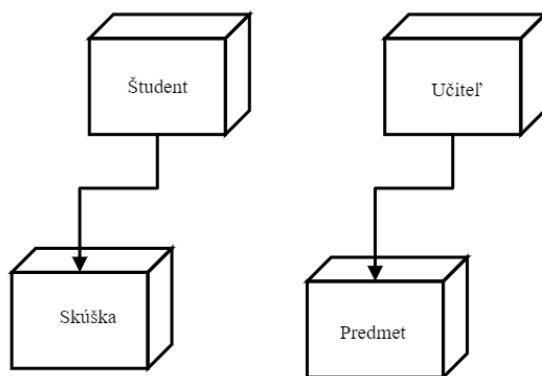
Obrázok 2 Lineárny dátový model [1]

Na zobrazenom príklade lineárneho dátového modelu je vidno, že medzi objektmi sa nenachádzajú žiadne väzby, takže nie je možné priamo zistiť, ktorý študent zložil skúšku a z akého predmetu. Jediný vzťah, ktorý sa v tomto type modelu vyskytuje je „predchádzajúci“ a „nasledujúci“. [1]

Považuje sa za najzakladanejší dátový model a je možné ho implementovať na hocijaké médium. Ideálnym príkladom takéhoto modelu je kartotéka pacientov. [1]

1.3.2 Hierarchický dátový model

Dátové objekty v hierarchickom dátovom modeli sú rozložené do určitých úrovní alebo hierarchií. Toto rozloženie objektov sa označuje, že je tvorené rodičovským segmentom alebo že sú údaje organizované do stromovej štruktúry. Znamená to, že určité objekty sú nadradené (rodičia) voči tým ostatným podriadeným (deti). Nadriadené objekty obsahujú aj dáta z podriadených objektov. [1,10]

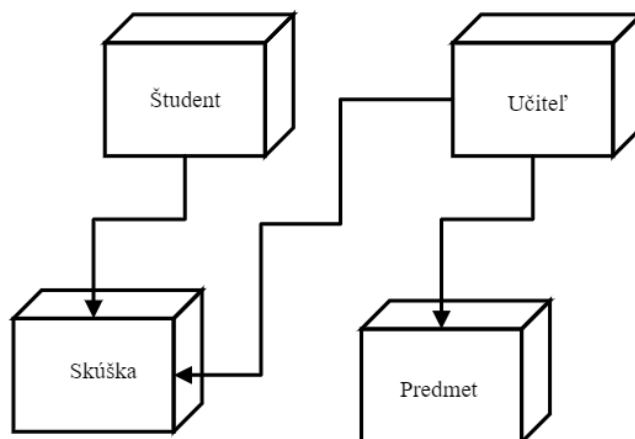


Obrázok 3 Hierarchický dátový model [1]

Väzby medzi jednotlivými objektami sú znázorňované ukazovateľmi – pointery, ktoré vytvárajú databázový systém, na ktorom je model implementovaný. Na podriadené objekty je možné sa dostať len cez tie ich nadriadené. [1]

1.3.3 Sieťový dátový model

Sieťový dátový model sa považuje za rozšírenie hierarchického dátového modelu. Na rozdiel od neho pointery nie sú uvedené len medzi nadriadenými a podriadenými objektami, ale môžu viesť aj naprieč modelom v rôznych smeroch. Tieto pointery naznačujú, kade sú medzi jednotlivými objektami vedené väzby. V sieťovom modeli sa už nenachádzajú objekty rodičov a detí ale sú označované už len ako segmenty. [1]

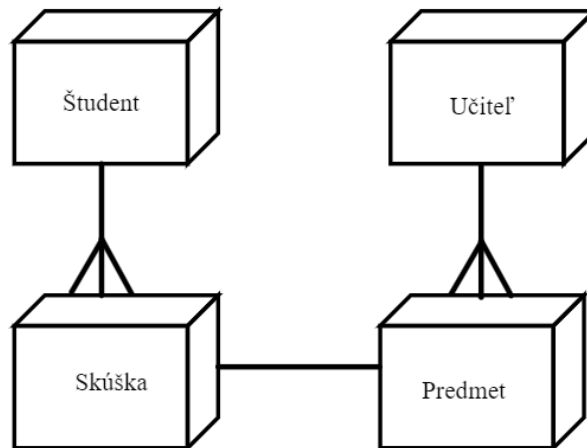


Obrázok 4 Sieťový dátový model [1]

Implementácia sieťového modelu je zložitá a zároveň veľmi náročná na udržanie a neskoršiu úpravu. Bol to najpoužívanejší databázový model pred zavedením relačného modelu. [10]

1.3.4 Relačný dátový model

Niektoré z najpopulárnejších databáz sú založené presne na relačnom dátovom modeli. Údaje v tomto modeli sú usporiadané v dvojrozmerných tabuľkách a vzťah sa udržiava uložením spoločného poľa. Základnou štruktúrou sú teda tabuľky a všetky potrebné údaje týkajúceho sa určitého typu sú uložené v riadkoch týchto tabuľkách. [10]



Obrázok 5 Relačný dátový model [1]

1.3.5 Objektový dátový model

Najnovší dátový model je objektový dátový model, no nie je ešte tak prepracovaný ako relačný dátový model. V objektovom modeli sú dáta skladované vo forme objektov. Tieto objekty okrem svojich atribútov majú definované aj metódy, ktoré určujú chovanie objektu. Ak sú objekty rovnakého typu tak ide o triedy objektov a konkrétny záznam predstavuje instanciu objektu. Každý objekt má pridelený aj unikátny identifikátor, ktorý slúži na priame väzby medzi objektami podobne ako v sieťovom modeli. Okrem týchto väzieb tak sa môžu v objektovom modeli nachádzať aj relačné väzby. [1,10]

Výhodou objektového dátového modelu je, že jediným spôsobom ako je možné s objektom pracovať je volanie niektorej z metód objektu. To zabezpečuje vysokú dátovú abstrakciu a nezávislosť dát. [1]

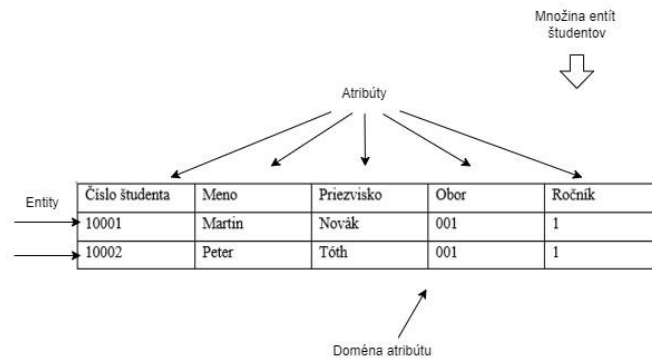
1.4 Relačný dátový model

Relačné dátové modely umožňujú zachytiť v modeli nie len dáta o skúmaných objektoch ale aj vzťahy medzi týmito objektami. Tento typ modelov je založený na teórii relácií. [1]

1.4.1 Základné pojmy

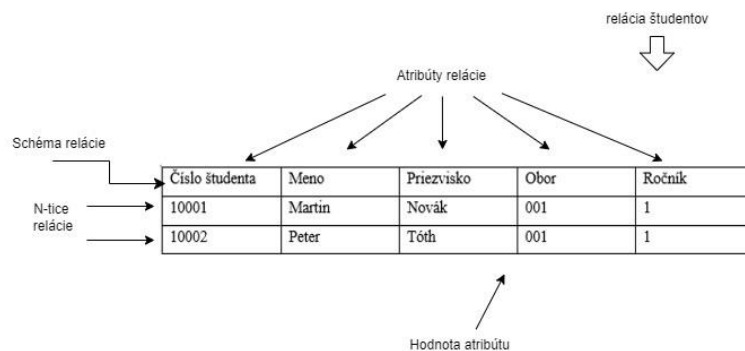
V kontexte dátových modelov sa využíva viac možností pomenovania základných pojmov, podľa toho, z akého pohľadu sa na objekt pozerá. Najbežnejšie sa na objekt pozerá z pohľadu teórie množín alebo z pohľadu teórie relácií. [1]

1. Terminológia z pohľadu teórie množín



Obrázok 6 Terminológia z pohľadu teórie množín [1]

2. Terminológia z pohľadu teórie relácií



Obrázok 7 Terminológia z pohľadu teórie relácií [1]

V oboch prípadoch terminológie sa jedná o dvojrozmernú tabuľku, ktorá sa skladá z riadkov a stĺpcov a vyznačuje sa nasledujúcimi vlastnosťami:

- riadky obsahujú dáta o entite (N-tice),
- stĺpce obsahujú dáta o atribútoch entity (N-tice),
- bunky uchovávajú jedinou hodnotu,
- všetky položky v stĺpci sú rovnakého druhu,
- všetky stĺpce majú jedinečný názov,
- žiadne dva riadky nesmú obsahovať rovnaké dátové hodnoty.[1]

1.4.2 Relácie (vzťahy)

Relácie patria medzi základné prvky relačného modelu a zabezpečujú prepojenie medzi jednotlivými tabuľkami modelu. Relácie určujú, ako sú dáta v rôznych tabuľkách spojené a ako medzi sebou interagujú. Sú založené na stĺpcoch jednotlivých tabuliek, ktoré obsahujú rovnaké dáta. Tabuľku Zákazník je možné prepojiť s tabuľkou Objednávka v prípade ak v oboch tabuľkách sa nachádza uložené ID zákazníka. Takéto údaje dostávajú označenie kľúčov. [11]

1.4.3 Typy kľúčov

Ako je už v predošlej podkapitole znázornené, tak kľúče predstavujú stĺpec so špeciálnymi vlastnosťami, ktoré slúžia na prepojenie medzi tabuľkami. Okrem prepojenia tabuliek slúži aj na presné identifikovanie riadkov v tabuľke.

Kľúče môžu byť jedinečné ak ich hodnota identifikuje iba jeden riadok alebo aj nejedinečné, ak ich jedna hodnota sa vyskytuje vo viacerých riadkoch. Typickým príkladom jedinečného čísla je identifikačné číslo, ktoré je na každom riadku iné. Taktiež dobrým príkladom pre nejedinečné kľúče je priezvisko. Napríklad také populárne priezvisko Novák môže byť v celej databáze iba raz ale aj niekoľko krát. [4]

Existuje ešte aj viac typov kľúčov ale tie najdôležitejšie sú:

- **Primárny kľúč** – predstavuje unikátny identifikátor, ktorý je minimálny a pomáha jednoznačne identifikovať a pristupovať k záznamom v tabuľke. Každá tabuľka v databáze má mať definovaný primárny kľúč.
- **Kandidátny kľúč** – má rovnaké vlastnosti ako primárny kľúč ale nie je vybraný ako primárny kľúč. Z kandidátnych sa vyberie jeden ako primárny kľúč a ostatné sú určené ako alternatívne.
- **Cudzí kľúč** – je to kľúč, ktorý odkazuje na primárny kľúč v inej tabuľke. Používa sa na vytvorenie vzťahu medzi dvoma tabuľkami v databáze. [11]

1.4.4 Integrita relačného modelu

Na zabezpečenie integrity dát v relačnom modeli sú určené integritné obmedzenia, ktoré je potrebné dodržiavať. Tieto opatrenia sa delia na:

1. integritné obmedzenia pre entity

- a. **doménová integrita** – zabezpečuje, že hodnoty v určitých atribútoch alebo stĺpcoch spĺňajú definovaný rozsah, dátový typ, povinné zadanie položky, jedinečnosť hodnoty v rámci stĺpca, implicitnú hodnotu a zoznam prípustných hodnôt (číselník), [1]
- b. **entitná integrita** – zabezpečuje, že každá relácia musí mať určený primárny kľúč a že tento údaj nemôže byť nulový a duplicitný, [1]
- c. **referenčná integrita** – predstavuje, že hodnota cudzieho kľúča v jednej tabuľke existuje v rovnakej hodnote aj v iných spojených tabuľkách. Vďaka tomu cudzie kľúče v spojení s primárnymi kľúčmi vytvárajú väzby medzi jednotlivými tabuľkami. [1]

Číslo študenta	Meno	Priezvisko	Obor	Ročník
10001	Martin	Novák	001	1
10002	Peter	Tóth	001	1

Diagram above the table shows arrows indicating key types: 'Primárny kľúč' points to 'Číslo študenta', 'Kandidátne kľúče' points to 'Meno' and 'Priezvisko', and 'Cudzí kľúč' points to 'Obor'.

Obrázok 8 Typy kľúčov (Zdroj: vlastné spracovanie)

2. Integritné obmedzenia pre vzťahy

- a. **vzťah 1:1** – predstavuje spojenie vždy jednej n-tice relácie odpovedá práve jedna n-tice inej relácie (osoba – vodičský preukaz), [1]



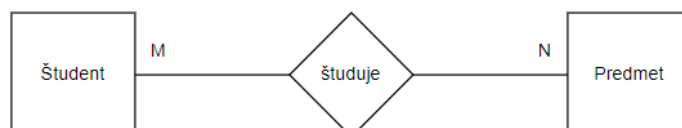
Obrázok 9 Vzťah 1:1 [1]

- b. **vzťah 1:N** – približuje spojenie vždy jednej n-tice relácie odpovedá jednej alebo viac n-ticiam inej relácie (študent – skúška), [1]



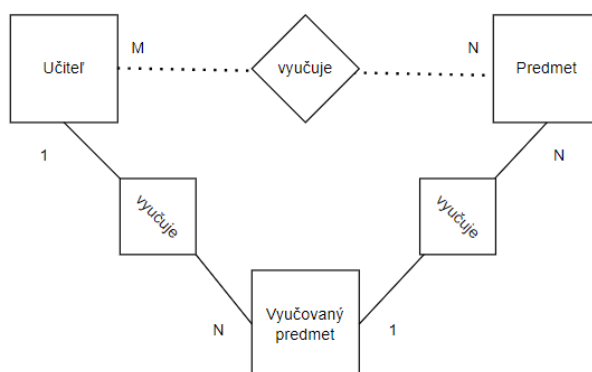
Obrázok 10 Vzťah 1:N [1]

c. **vzťah N:M** – predstavuje, že niekoľko n-tic relácie odpovedá jednej alebo viacerým n-ticiam relácie (študent – predmet). [1]



Obrázok 11 Vzťah N:M [1]

Vzťah N:M neumožňuje priame spojenie medzi tabuľkami pomocou cudzích kľúčov, preto je potrebné vykonať dekompozíciu – vytvorenie spojovaciú tabuľku, ktorá umožní prepojenie týchto entít. [1]



Obrázok 12 Dekompozícia vzťahu N:M [1]



Obrázok 13 Dekompozícia vzťahu N:M v tabuľkách [1]

1.4.5 Normalizácia

Normalizácia je proces usporiadania potrebných údajov v databáze. Zahŕňa vytváranie tabuliek a vzťahov medzi nimi podľa pravidiel určených na ochranu údajov a flexibility databázy odstránením nepotrebných údajov a nekonzistentnej závislosti. Práve tá môže sťažovať prístup k údajom, pretože cesta k vyhľadávaniu údajov môže chýbať alebo byť nesprávna. Pravidlá na splnenie normalizácie databázy sa nazývajú normálne formy. Ak sa dodrží prvé pravidlo, tak databáza sa označuje že je v prvej normálnej forme. [12]

1. Prvá normálna forma

Spočíva v eliminácii opakujúcich sa dát v jednotlivých tabuľkách a vytvára samostatnú tabuľku pre každú skupinu súvisiacich údajov. Taktiež odporúča nepoužívanie viacerých polí v jednej tabuľke na uloženie podobných údajov. [12]

Všetky atribúty entity musia byť jednoduché nie zložené alebo viachodnotové. Vhodným príkladom je adresa, ktorá má byť rozložená na jednotlivé časti – ulica, popisné číslo, obec, PSČ a nie všetky tieto údaje ako jeden atribút. [1]

2. Druhá normálna forma

Aby databáza mohla byť označovaná za databázu v druhej normálnej forme, tak je potrebné aby spĺňala prvú normálnu formu a aby všetky jej atribúty boli závislé na primárnom kľúči.

Pri takej kontrole je potrebné zaistiť aby atribúty jednotlivých tabuliek záviseli na zvolenom primárnom kľúči. Môže sa jednať o tom, že firma závisí na IČO spoločnosti, meno študenta závisí na rodnom čísle študenta alebo iné. [1]

3. Tretia normálna forma

Rovnako aj v tomto prípade ak databáza má byť v tretej normálnej forme je potrebné aby spĺňala prvú aj druhú normálnu formu a aby každý neklúčový atribút bol funkčne závislý iba na primárnom kľúči a nezávislý na všetkých ostatných neklúčových atribútoch.

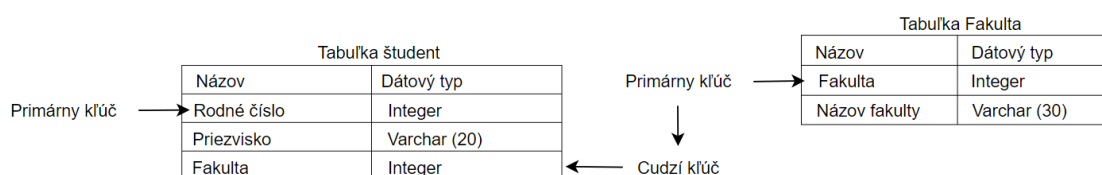
Na príklade tabuľky študent je atribút Fakulta, ktorý je závislý na Rodnom čísle študenta, ktorý študuje na danej fakulte, ale názov fakulty nie je závislý na Rodnom čísle študenta, ale na atribúte Fakulta a na rodnom čísle je závislý iba sprostredkovane, cez neklúčový atribút Fakulta. [1]

Tabuľka Študent

Názov	Dátový typ
Rodné číslo	Integer
Priezvisko	Varchar (20)
Fakulta	Integer

Obrázok 14 Tabuľka Študent pred úpravou na 3. normálnu formu (Zdroj: vlastné spracovanie)

Pre normalizáciu na tretiu normálnu formu je potrebné vykonanie dekompozície aby vznikla nová tabuľka Fakulta, ktorá bude prepojená s tabuľkou Študent cez cudzí kľúč.



Obrázok 15 Normalizácia na 3. normálnu formu (Zdroj: Vlastné spracovanie)

4. Boyce – Coddova normálna forma

Predstavuje variáciu tretej normálnej formy, takže aby databáza mohla byť v Boyce – Coddove normálnej forme, tak musí spĺňať tretiu normálnu formu, ale toto pravidlo neplatí v opačnom prípade. Táto forma je určená pre špeciálne prípady databáze, preto nie vždy je možné ju dosiahnuť. Na jej splnenie je potrebné aby medzi kandidátnymi kľúčmi nebola žiadna funkčná závislosť a to za nasledujúcich podmienok:

- relácia musí mať aspoň dva kandidátne kľúče,
- najmenej dva z kandidátnych kľúčov musia byť zložené,
- kandidátne kľúče sa v niektorých atribútoch musia prekrývať. [1]

1.5 SQL

SQL (Structured Query Language) je štandardný jazyk používaný na komunikáciu s relačnými databázami, ktorý umožňuje tvoriť databázové dotazy. Okrem toho, že jazyk SQL umožňuje tvorbu databáze taktiež zahrňuje nástroje na manipuláciu s dátami. Patrí do kategórie deklaratívnych programovacích jazykov, ktoré nepotrebujú byť napísané v žiadnom samostatnom programe ale sú vkladané do iného programovacieho jazyka. Jazyk SQL bol postupne prijímaný medzi rôznymi dodávateľmi databázových softvérov ako štandard a stal sa spojovacím článkom medzi týmito aplikáciami. V koncepcii klient/server sú dotazy

špecifikované na strane klienta, odkiaľ sa odosielajú na server, ktorý dotazy v SQL jazyku realizuje a následne zašle výsledok na stranu klienta. [13]

1.5.1 Kategórie príkazov SQL

Kategórie príkazov SQL sa na základe ich funkcií delia do určitých kategórií:

a) jazyk DDL (Data Definition Language) [13]

- zahŕňa príkazy, ktoré užívateľovi povoľujú vytváranie databázových objektov (tabuľky, pohľady a indexy) a upravovať ich štruktúru
- jedná sa o príkazy obsahujúce CREATE, ALTER a DROP

b) jazyk DQL (Data Query Language) [13]

- obsahuje príkazy, ktoré načítajú dáta z databázy
- sú založené na jednom kľúčovom slove SELECT

c) jazyk DML (Data Definiton Language) [13]

- súčasťou sú príkazy, ktoré umožňujú pridávanie dát do databázy v podobe aj riadkov aj tabuliek, odoberať dáta a meniť stávajúce
- obsahuje príkazy INSERT, UPDATE a DELETE

d) jazyk DCL (Data Control Language) [13]

- súčasťou sú príkazy, ktoré správcovi dovoľujú riadiť prístup k dátam v databáze a riadiť systémové oprávnenia DBMS, ako sú funkcie pre spustenie alebo vypnutie databázy
- obsahuje príkazy, ktorých súčasťou sú slová GRANT alebo ALTER

e) príkazy riadiace transakcie [13]

- databázová transakcia je balík príkazov, ktoré databázový užívateľ potrebuje spracovať ako nedeliteľnú jednotku
- neodpovedajú presne syntaxi príkazov jazyka SQL ale majú značný vplyv na chovanie týchto príkazov.

1.5.2 Dátové typy

Jazyk SQL dokáže uchovávať rôzne dátové typy pre rozličné typy údajov. Podľa vlastnosti je možné tieto typy deliť do nasledujúcich skupín:

a) presné numerické hodnoty [14]:

- TINYINT – malé celé čísla v rozsahu 0 – 255 (1 bajt)

- SMALLINT – reprezentuje krátke celé čísla v rozsahu -32 768 až 32 767 (2 bajty)
- INT – reprezentuje celé čísla (4 bajty)
- BIGINT – používaný na reprezentáciu celých čísel s veľmi veľkým rozsahom hodnôt (8 bajtov)
- BIT – reprezentuje hodnoty bitov
- DECIMAL – predstavuje pevné bodové čísla s presne definovanou presnosťou a škálou (DECIMAL (10,2) môže mať hodnotu až 10 číslic celkom a z toho 2 sú desatinné miesta)
- MONEY – reprezentuje peňažnú hodnotu s presnosťou na 4 desatinné miesta

b) približné čísla [14]:

- FLOAT – číslo s pohyblivou desatinnou čiarkou (8 bajtov)
- REAL – číslo s pohyblivou desatinnou čiarkou (4 bajty)
- DOUBLE PRECISION – číslo s pohyblivou desatinnou čiarkou s dvojitou presnosťou

c) hodnoty vyjadrujúce čas a dátum [14]:

- DATE – kalendárny dátum
- TIME – konkrétny denný čas
- TIMESTAMP – časový okamžik
- INTERVAL – časový interval
- DATETIME – kalendárny dátum a aj denný čas

d) znakové hodnoty [14]:

- CHARACTER – reťazec ľubovoľných znakov v rozsahu 1 až 255 znakov
- VARCHAR (n) – pole znakov s určenou n dĺžkou

e) ostatné dátové typy [14]:

- BOOLEAN – logické hodnoty (true/false/unknown)
- BLOB – rozsiahly binárny dátový objekt (obrázok, zvuk, súbor...)

1.5.3 Syntax základných príkazov

Nasledujúca kapitola obsahuje základné príkazy jazyka SQL, ktoré umožňujú vytvárať upravovať a odstraňovať dáta, ako aj získavať informácie z databázových tabuliek a ich syntax.

[14]

- CREATE – používa sa na vytváranie nových objektov v databáze (tabuľky, indexy, pohľady...)

CREATE Názov_objektu

Obrázok 16 Príkaz CREATE (Zdroj: vlastné spracovanie)

- INSERT – slúži na pridanie nových záznamov do tabuľky

INSERT INTO Návov_tabuľky (Stĺpec1, Stĺpec2) **VALUES** ('Peter', 25)

Obrázok 17 Príkaz INSERT (Zdroj: Vlastné spracovanie)

- ALTER – používa sa na zmenu už existujúcich objektov v databáze

ALTER Názov_objektu

Obrázok 18 Príkaz ALTER (Zdroj: vlastné spracovanie)

- DROP – slúži na odstránenie existujúcich objektov z databázy

DROP TABLE Názov_tabuľky

Obrázok 19 Príkaz DROP (Zdroj: Vlastné spracovanie)

- SELECT – základný dotaz používaný na výber dát z databázových tabuliek
- FROM – definuje z ktorých objektov sa majú dáta vybrať
- JOIN – klauzula používaná na spojenie dátových množín z dvoch alebo viacerých tabuliek na základe rovnakého stĺpca. Tejto klauzule sa bude podrobnejšie zaoberať ďalšia podkapitola
- WHERE – slúži na filtrovanie výsledkov na základne zadaných podmienok
- GROUP BY - používa sa na zoskupenie výsledkov podľa jedného alebo viacerých stĺpcov
- ORDER BY – slúži na zoradenie výsledkov podľa určitého stĺpca alebo viacerých stĺpcov

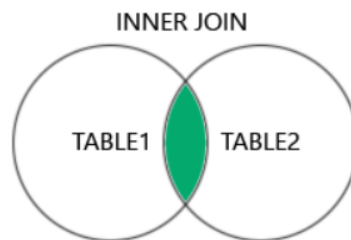
```
SELECT Názvy_stĺpcov FROM Názov_tabuľka
JOIN Názov_tabuľky ON Podmienka_spojenia
WHERE Kritérium_výberu
ORDER BY Zoradenie_stĺpcov
```

Obrázok 20 Základné príkazy výberu informácií (Zdroj: Vlastné spracovanie)

1.5.4 Spojovanie tabuliek príkazom JOIN

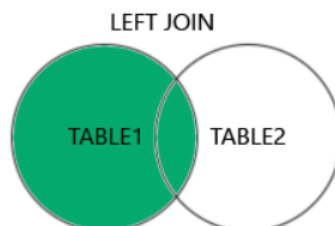
U spojovanie databázových tabuliek je potrebné vybrať správny typ spojenia a následne ho zrealizovať. Spojovanie pomocou príkazu JOIN je možné rozdeliť do štyroch skupín [14]:

- **vnútorné spojenie (INNER JOIN)** – výsledná tabuľka obsahuje záznamy, ktoré si odpovedajú v oboch tabuľkách



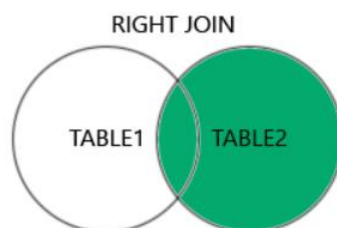
Obrázok 21 INNER JOIN [15]

- **spojenie z ľavej strany (LEFT JOIN)** – všetky záznamy z ľavej tabuľky a zhodné záznamy z pravej tabuľky, v prípade nezahody s pravou tabuľkou zobrazí sa hodnota NULL



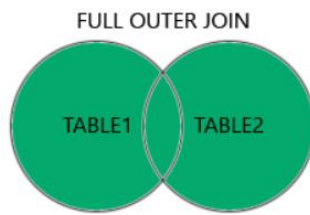
Obrázok 22 LEFT JOIN [15]

- **spojenie z pravej strany (RIGHT JOIN)** - všetky záznamy z pravej tabuľky a zhodné záznamy z ľavej tabuľky, v prípade nezahody s ľavou tabuľkou zobrazí sa hodnota NULL



Obrázok 23 RIGHT JOIN [15]

- **úplné spojenie (FULL OUTER JOIN)** – zobrazí všetky údaje aj v prípade nezahody



Obrázok 24 FULL OUTER JOIN [15]

1.5.5 Agregáčn  funkcie

Služia na kombináciu viacerých riadkov záznamov do jedného riadku. Tieto funkcie sú kľúčové pre agregáciu dát a výpočet štatistík v databázach. [13]

Tabuľka 1 Agregáčn  funkcie (Zdroj: Vlastn  spracovanie)

N�zov funkcie	Popis
AVG	vypo�íta priemern� hodnotu st�lpcu
COUNT	zist� po�et n�jden�ch hodn�t v st�lpci
MAX	vyhľadá maxim�lnu hodnotu st�lpcu
MIN	vyhľadá minim�lnu hodnotu st�lpcu
SUM	spo��ta všetky hodnoty dan�ho st�lpcu

1.5.6 Pohľady

Pohľady v jazyku SQL predstavuj  druh ur itej virtu lnej tabuľky, ktor  ma taktiež formu riadkov a st lpcov ako klasick  tabuľky ale pohľady nie s  ukladan  na disk ako tabuľka. Prezentyje údaje z jednej alebo viacer ch tabuliek na z klade prisp soben ho dotazu a formuje ich ako keby poch dzali z jedn ho zdroja. [13]

Je mon  ho vytvoriť v berom z znamov z jednej alebo viacer ch tabuliek pr tomn ch v datab ze. M e obsahovať buď všetky riadky tabuľky alebo konkr tne riadky na z klade ur it ch podmienok. [16]

```

CREATE VIEW Pohľad1 AS
SELECT Stĺpec1, Stĺpec2...
FROM Tabuľka1
JOIN Tabuľka2 ON....

SELECT * FROM Pohľad1

```

Obrázok 25 Tvorba pohľadov (Zdroj: Vlastné spracovanie)

1.5.7 Procedúry

Procedúry predstavujú pripravené dotazy, ktoré sú uložené k opakovanému použitiu jej zvolaním. Uloženým procedúra môže taktiež konať na základe zadaných parametrov. Tieto parametre prijíma ako vstup a vykonáva určitú úlohu a môže ale nemusí vrátiť hodnotu. [17]

```

CREATE PROCEDURE Procedúra1
parameter1 data_typ...
AS
BEGIN
SQL_dotaz
END

```

Obrázok 26 Tvorba procedúr (Zdroj: Vlastné spracovanie)

```
EXEC Procedúra1 parameter1
```

Obrázok 27 Volanie procedúr (Zdroj: Vlastné spracovanie)

1.5.8 Spúšťače

Spúšťač často tiež nazývaný Trigger je uložená procedúra v databáze, ktorá sa automaticky vyvolá vždy, keď v databáze nastane špeciálna určitá udalosť. Môže byť napríklad vyvolaný pri vložení riadku do určenej tabuľky alebo pri aktualizácii stĺpcov. Predstavuje určitú reakciu na základe odohranej udalosti v databázovom systéme. [18]

1.6 Metodológia návrhu databázy

Správny návrh databáze je kľúčovým krokom pri vytváraní efektívneho a spoľahlivého databázového systému, ktorý má spĺňať potreby spoločnosti. Metodológia návrhu databázy je štruktúrovaný postup využívajúci určité procedúry, techniky, nástroje a dokumentáciu, ktoré majú efektívne pomôcť pri samotnom návrhu. Skladá sa primárne z troch častí: konceptuálny, logický a fyzický návrh.

1.6.1 Konceptuálny návrh databázy

Predstavuje prvý krok pri navrhovaní databázového systému. Tento krok je kritický pre úspech celého projektu, pretože definuje základné objekty a vzťahy. Dôležitou súčasťou konceptuálneho návrhu je komunikácie so zadávateľom projektu. Z toho dôvodu sa veľa času trávi na stretnutiach so spoločnosťou a navrhovateľ databázy sa snaží pochopiť štruktúru a vnútorné procesy spoločnosti zadávateľa. [19]

Cieľom konceptuálneho návrhu je poskytnúť dátovo orientovaný pohľad na organizáciu a vzťahy medzi týmito dátami. Výsledok pohľadu je prezentovaný v podobe ER diagramu. Na vytvorenie vhodného ER diagramu je potrebné splniť tieto kroky [20,21]:

- Identifikácia entít
- Identifikácia relácií
- Identifikácia atribútov
- Určenie domény atribútu
- Určenie primárnych kľúčov
- Kontrola redundancií modelu
- Kontrola užívateľských transakcií

1.6.2 Logický návrh databázy

Logický návrh stanovuje štruktúru dátových prvkov a vzťahy medzi nimi. Je nezávislý od fyzickej databázy, ktorá podrobne opisuje ako budú údaje implementované. Logický návrh posúva prvky konceptuálneho o krok ďalej tým, že k nim pridáva ďalšie potrebné informácie a premieňa konceptuálny návrh do konkrétnych entít a vzťahov. Taktiež pridáva podrobnosti o štruktúre dát, normalizuje a optimalizuje databázu. [22]

Výsledkom logického návrhu by mal byť upravený ER diagram, ktorý identifikuje entity a vzťahy medzi nimi, ktoré sa nachádzajú v navrhovanej databáze. Špecifikuje podnikovú schému, ktorá graficky reprezentuje celkovú logickú štruktúru databázy. [23]

1.6.3 Fyzický návrh databázy

Fyzický návrh databázy predstavuje štruktúru databázy a definuje, ako sú údaje uložené, organizované a prístupné na fyzickej úrovni. Poskytuje podrobný pohľad na databázovú schému, špecifikuje tabuľky, stĺpce, dátové typy, vzťahy, indexy a obmedzenia. [24]

Opisuje ako sú údaje v databáze uložené a pomocou neho je navrhnutá skutočná schéma databázy. Definujú sa pohľady, spúšťače, procedúry, transakcie a ďalšie koncepty vrátane primárny a cudzích kľúčov. Výsledkom tejto časti je komplexný súbor dokumentov, skriptu a technických detailov, ktoré popisujú štruktúru a funkciu databázového systému. [25]

Fyzickým návrhom nekončí životný cyklus databáze, po jeho implementovaní sa načítajú všetky potrebné dáta a sú testované. Ak sa overí či databázový systém funguje podľa očakávaní, tak sa nasadí do produkčného prostredia a monitoruje sa jeho prevádzka a výkonnosť. Naďalej sa spravujú databázové objekty, opravujú chyby a pravidelne sa vykonáva údržba databázy. [26]

2 Analýza súčasného stavu

2.1 Základné informácie o firme

Firma, ktorou sa zaoberá táto záverečná práca si neželala zverejňovať úplne všetky interné informácie. Z tohto dôvodu sú informácie, ktoré sa vyskytujú v záverečnej práci trochu obmedzené.

Firma pôsobí na trhu už od roku 1995 a vystupuje pod menom Saver. Sústreďí sa na autorizovanú montáž a servis garážových brán, domových dverí značky Hörmann. Ďalej v svojej ponuke má pohony, závory a iné príslušenstvo. Tieto produkty poskytujú tak ako pre rodinné domy aj do priemyselných areálov alebo iným podnikateľským subjektom. Produkty typu garážové brány, dvere alebo závory sú vo väčšine prípadov vyrábané na mieru podľa predstáv zákazníka. Ostatné výrobky ako pohony a iné príslušenstvo pochádzajú zo sériovej výroby.

Jedná sa o menšiu rodinnú firmu, ktorá začínala montážou a servisom iba v rámci Košíc. Postupom času a upevňovaním svojho miesta na trhu sa spoločnosť rozrástla a momentálne poskytuje svoje služby v Košickom a Prešovskom kraji Slovenskej republiky. Sídlo naďalej zostalo v Košiciach, v ktorých sa teraz nachádzajú aj tri menšie sklady. Z dôvodu výroby na mieru firma ani nepotrebuje disponovať väčším priestorom na uskladňovanie tovaru. V Košiciach sa okrem sídla a skladov nachádza aj na predajni showroom, kde je možné vidieť niektoré z ponúkaných brán, dverí, pohonov alebo ostatných produktov.



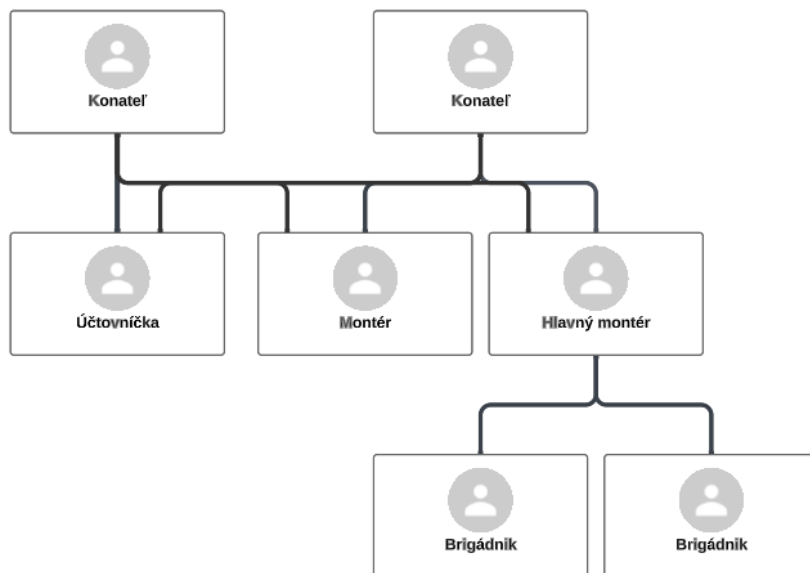
Obrázok 28: Logo spoločnosti [27]

2.2 Organizačná štruktúra

Spoločnosť má funkcionálnu organizačnú štruktúru, ktorá sa skladá z dvoch konateľov spoločnosti, účtovníčky, dvoch montérov na hlavný pracovný pomer a ďalších dvoch brigádnikov.

Konatelia majú primárne na starosti komunikáciu so zákazníkom a objednávanie tovaru. V ojedinelých prípadoch idú na montáž osobne aj oni. Účtovníčka sa stará o účtovníctvo, čiastočnú komunikáciu so zákazníkom, kancelárske práce a prípravu podkladov pre montáž (montážny list, údaje o zákazníkovi...) alebo servis. Montéri, ktorí sa nachádzajú vo firme, tak jedného z nich je možné považovať za hlavného aj keď sú na jednej úrovni a to iba preto, lebo má pod sebou dvoch brigádnikov, o ktorých sa stará. Úloha brigádnikov v spoločnosti spočíva v pomocných prácach na montáži alebo servise.

Nakoľko sa jedná o rodinnú firmu, tak táto štruktúra je len pre teoretické priblíženie vzťahov od vzniku spoločnosti. V praxi funguje spoločnosť jednotne a každý robí, to čo najlepšie vie a všetci si navzájom pomáhajú a dopĺňajú sa aby zabezpečili najlepší chod firmy.



Obrázok 3: Organizačná štruktúra spoločnosti (Zdroj: vlastné spracovanie)

2.3 Hardware

Nakoľko sa firma orientuje v stavebníctve, tak nedisponuje veľkým počtom ani nijakým zložitým hardwarom. Firma priamo v kancelárii využíva tri stolové počítače značky Apple, ktoré využíva na elektronickú komunikáciu so zákazníkmi, účtovanie, marketingové účely, spravovanie webovej stránky a stránkach na sociálnych sieťach a ostatné. Konatelia spoločnosti majú tiež vlastné notebooky na prípadnú prácu mimo kancelárie, čo v najčastejšom prípade je komunikácia so zákazníkom alebo úprava webových stránok a sociálnych sietí.

Nakoľko montéri si často aj počas výkonu práce potrebuje niečo overiť v systéme a so sebou nenosia notebooky, tak využívajú služobné telefóny tiež značky Apple. Tie sú prepojené s počítačmi v kancelárii, takže si vedú kdekoľvek a kedykoľvek overiť zadanie montáži, komunikáciu so zákazníkom alebo prípadne nahráť fotodokumentáciu vykonanej práce.

Okrem týchto zariadení spoločnosť využíva na dennodenný chod bežné kancelárske zariadenia ako sú tlačiareň, skener, pevná linka alebo iný kancelársky hardware.

2.4 Software

Ako sa už spomína v časti hardware, že spoločnosť využíva zariadenia od spoločnosti Apple, tak je jasné, že majú nainštalovaný operačný systém iOS, s ktorým súvisí aj antivírus od spoločnosti Apple. Na bežnú kancelársku prácu využívajú balík Microsoft 365 (Word, Excel, PowerPoint...), ktorého súčasťou je aj Outlook, ktorý spoločnosť využíva na e-mailovú komunikáciu s odberateľom alebo dodávateľom.

Na internú komunikáciu spoločnosť tiež využíva Outlook ak je potrebné priložiť súbor väčšieho objemu. Ak sa jedná o urgentnú informáciu alebo informáciu týkajúcej sa zákazky, na ktorej sa už pracuje, tak firma využíva klasický telefónny rozhovor alebo SMS či aplikáciu WhatsApp. Túto aplikáciu využívajú primárne montážnici, ktorí si preposielajú fotografie. Môže sa jednať o fotodokumentáciu vykonanej práce, nedostatky, ktoré musí zákazník opraviť aby mohla prebehnúť montáž alebo z iných dôvodov.

Na čiastočné účtovníctvo spoločnosť využíva Omegu. Tá sa jej stará o evidenciu faktúr, pokladne, miezd zamestnancov a ostatné funkcie, ktoré tento systém ponúka. Software pracuje na vzdialenom počítači, na ktorý je možné sa prihlásiť, z ktoréhokoľvek zariadenia na základe prihlasovacích údajov. O zvyšnú časť účtovníctva sa im stará externá spoločnosť, ktorá má rovnaký prístup na tento vzdialený počítač s vlastnými prihlasovacími údajmi.

2.5 Informačný systém

Firma nevyužíva žiadny špeciálny informačný systém. Ako už bolo v práci spomenuté na evidenciu faktúr, bločkov, miezd alebo ostatných účtovných dokladov využíva Omegu. Rozvrh práce, kedy ktorú zákazku treba vykonať alebo iné údaje spojené s dátumom firma nahráva do Apple Reminders, ktorý majú všetci zamestnanci zosynchronizovaný vďaka čomu vedia všetky dôležité informácie o jednotlivých zákazkách v daný deň hneď.

2.6 Databáza

Firma zvolila kombinovaný prístup k uchovávaniu informácií. Určitá časť dát, ktorá sa skladá z informácií týkajúcich sa faktúr, pohybu tovaru na sklade a ostatných účtovných dokladov (príjmy/výdaje z pokladne, mzdy, interné doklady...) sa nahráva do už spomínaného účtovného softwaru Omega. Originály týchto dokladov spoločnosť uchováva ešte aj v papierovej podobe organizovaným zakladaním do zakladačov alebo kartotéky. Všetky údaje, ktoré vstupujú do Omegy sú zálohované na spomínanom vzdialenom počítači a taktiež kópie sú vytlačené alebo kopírované a uložené u externej účtovníčky. Informácie v určenej forme, ktoré sa vkladajú do softwaru Omega sa nachádzajú v nasledujúcich tabuľkách.

V sekcii prijatých faktúr v Omegi firma udržiava základné informácie, ktoré by mala každá faktúra obsahovať. Jedná sa ako o daňové a zákonné náležitosti ale aj informácie, ktoré slúžia na prehľadnosť v kontaktovaní či firmy alebo zákazníka.

Tabuľka 2: Prijaté faktúry v software Omega (Zdroj: Vlastné spracovanie)

Prijaté faktúry	
Atribút	Popis
Číslo faktúry	Číslo faktúry
Bankové spojenie	IBAN, číslo účtu
Variabilný symbol	Variabilný symbol
Špecifický symbol	Špecifický symbol
Kontaktné údaje dodávateľa	Názov firmy, adresa sídla, IČO, DIČ, kontaktná osoba, telefón, e-mail
Kontaktné údaje odberateľa	Názov firmy, adresa sídla, IČO, DIČ, kontaktná osoba, telefón, e-mail
Dátum vystavenia	Dátum kedy bola faktúra vyhotovená
Dátum splatnosti	Dátum do kedy má byť faktúra zaplatená
Dátum uskutočnenia zdaniteľného plnenia	Dátum kedy vstupuje faktúra do priznania DPH
Text	Popis za čo bola vystavená faktúra
Položky faktúry	Jednotlivé položky faktúry
Čiastky	Cena bez DPH, sadzba dane, daň, Cena celkom

Nakoľko vydané faktúry sa nelíšia v mnohých veciach od tých prijatých, tak informácie, ktoré uchovávajú sa tiež veľmi nelíšia. Jediný väčší rozdiel nastáva na strane odberateľa a dodávateľa.

Tabuľka 3: Vydané faktúry v software Omega (Zdroj: Vlastné spracovanie)

Vydané faktúry	
Atribút	Popis
Číslo faktúry	Číslo faktúry
Variabilný symbol	Variabilný symbol
Špecifický symbol	Špecifický symbol
Kontaktné údaje zákazníka – fyzická osoba	Meno a priezvisko, adresa, e-mail, telefónne číslo
Kontaktné údaje zákazníka – právnická osoba	Názov spoločnosti, IČO, DIČ, sídlo firmy, kontaktná osoba, e-mail, telefónne číslo
Dátum vystavenia	Dátum kedy bola faktúra vyhotovená
Dátum splatnosti	Dátum do kedy má byť faktúra zaplatená
Dátum uskutočnenia zdaniteľného plnenia	Dátum kedy vstupuje faktúra do priznania DPH
Text	Popis za čo bola vystavená faktúra
Položky faktúry	Jednotlivé položky faktúry
Čiastky	Cena bez DPH, sadzba dane, daň, Cena celkom

Spoločnosť využíva aj možnosť softwaru evidovať sklady a pohyby na nich. O jednotlivých skladoch ju zaujímajú iba základné informácie.

Tabuľka 4: Sklady v software Omega (Zdroj: Vlastné spracovanie)

Sklad	
Atribút	Popis
Názov	Názov skladu
Adresa	Adresa skladu
Typ	Na čo konkrétne je sklad určený (pohony, brány, skrutky...)

Na základe skladov firma eviduje aj pohyb zásob, vo forme účtovných dokladov príjemka a výdajka. Okrem týchto dokladov firma zadáva do Omegy aj určité informácie, ktoré sú pre nich podstatné ako pri skladovaní aj pri získavaní detailov o produkte.

Tabuľka 5: Zásoby v software Omega (Zdroj: Vlastné spracovanie)

Zásoby	
Atribút	Popis
Kód produktu	Jedinečný identifikačný kód produktu
Názov	Názov produktu
Typ	O aký produkt sa jedná (pohon, brána, závora...)
Jednotky skladovania	Kusy, metre...
Nákupná cena	Za akú cenu bol produkt zakúpený
Sklad	V ktorom sklade sa produkt nachádza

Spoločnosť v Omega spravuje aj evidenciu vlastných zamestnancov, ktorá súvisí aj so zaúčtovaním miezd. Zaujímajú ju klasické informácie, ktoré sú doplnené o náplň práce jednotlivých zamestnancov.

Tabuľka 6: Zamestnanci v software Omega (Zdroj: Vlastné spracovanie)

Zamestnanci	
Atribút	Popis
Meno a priezvisko	Meno a priezvisko
Rodné číslo	Rodné číslo
Dátum narodenia	Dátum narodenia
Dátum nástupu	Dátum od kedy je vo firme zamestnaný
Pozícia	Pozícia, na ktorej pracuje
Náplň práce	Náplň práce zamestnanca

Okrem informácií, ktoré spoločnosť uchováva v software Omega, tak firma uchováva aj určité informácie mimo neho. Jedná sa primárne o interné doklady, ktoré slúžia na zlepšenie procesov a zefektívnenie práce. Tieto informácie využívajú už uložené dáta v Omega a zhromažďuje ich na jednom mieste, kde ako celok poskytujú väčší prehľad o niektorých procesoch. Jedná sa o informácie, ktoré opisujú zákazku a o montážny list, ktorý dostávajú montážnici pred výkonom svojej práce.

Firma na zjednodušenie procesov vo firme, zrýchlenému vyhľadávaniu informácií zbiera údaje z každej zákazky zvlášť a uskladňuje ich dokopy. Tieto informácie zbiera z entít zákazníka, vydannej faktúry a zásob.

Tabuľka 7: Informácie o zákazke (Zdroj: Vlastné spracovanie)

Zákazka	
Atribút	Popis
Číslo zákazky	Identifikačné číslo
Zákazník	Informácie o zákazníkovi
Číslo faktúry	Číslo faktúry
Práca	Či sa jedná o servis alebo montáž
Stav zákazky	V akom stave sa momentálne zákazka nachádza

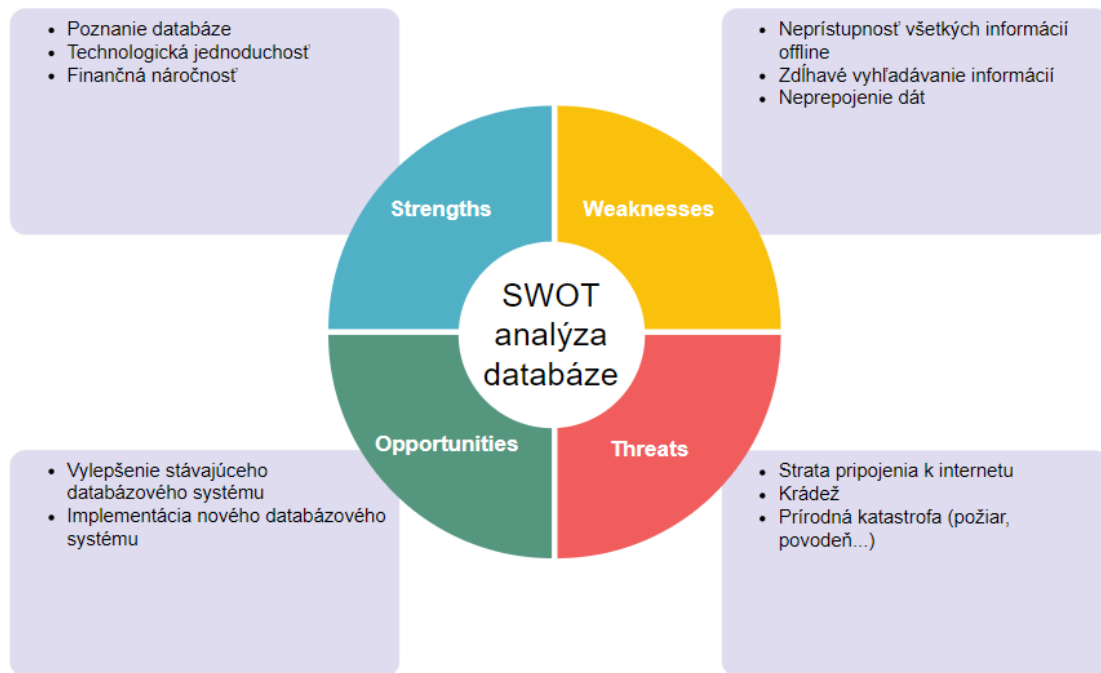
Pred každou vykonanou zákazkou montéri, ktorí sú vyslaný na vykonanie danej zákazky dostanú dokument montážny list. Nakoľko montéri sú vyslaný každý deň na inú adresu ohľadom inej zákazky, tak nie je v ich možnostiach pamätať, ktorý produkt majú kam namontovať a nastaviť. Informácie zobrazené v tomto dokumente spoločnosť zbiera z agend vydané faktúry, zákazníci, zásoby a zákazka.

Tabuľka 8: Informácie v montážnom liste (Zdroj: Vlastné spracovanie)

Montážny list	
Atribút	Popis
Číslo montáže	Identifikačné číslo
Zákazník	Informácie o zákazníkovi
Miesto montáže/servisu	Miesto vykonania montáže/servisu
Produkt	Produkt, ktorý bol namontovaný zákazníkovi
Stav montáže	V akom stave sa nachádza montáž
Pracovník	Kto bol vyslaný na montáž
Podpis pracovníka	Podpis po dokončení montáže
Podpis zákazníka	Podpis zákazníka po dokončení montáže

2.7 SWOT analýza databáze

Na základe predstavenie databáze bola vyhotovená jej analýza a to pomocou SWOT analýzy. Z dôvodu zamerania záverečnej práce je analýza sústredená iba na samotnú databázu a nie ako to býva zvyčajne na celú spoločnosť. SWOT analýza sa zamerala na štyri stránky tejto databáze - silné, slabé, príležitosti a hrozby.



Obrázok 29: SWOT analýza databáze (Zdroj: Vlastné spracovanie)

2.7.1 Zhodnotenie SWOT analýzy

Medzi silné stránky databáze je jej samotné poznanie zamestnancov. Firma sa už na trhu nachádza dlhšiu dobu a už od jej začiatkov uchovávala dáta podobným až rovnakým spôsobom. Takže je to zaužívaný a dobre poznaný proces jednotlivými zamestnancami. Ďalšou výhodou je jej technologická jednoduchosť, kde netreba žiadny špeciálny software na ukladanie dát. V prípade nástupu nového zamestnanca, by nebolo náročné ho naučiť týchto procesov vo firme. Ako je spomenuté firma nevyužíva žiadny špeciálny software a to sa preukazuje aj na ušetrených finančných nákladoch.

Medzi najväčšie nevýhody databáze určite patrí neprístupnosť všetkých dokumentov a potrebných informácií bez použitia internetového pripojenia. Aj keď firma uchováva určité dokumenty vo forme vytlačených originálov, tak niektoré sa nachádzajú iba na spomínanom vzdialenom serveri. S týmto súvisí aj slabé alebo nedostatočné prepojenie medzi jednotlivými

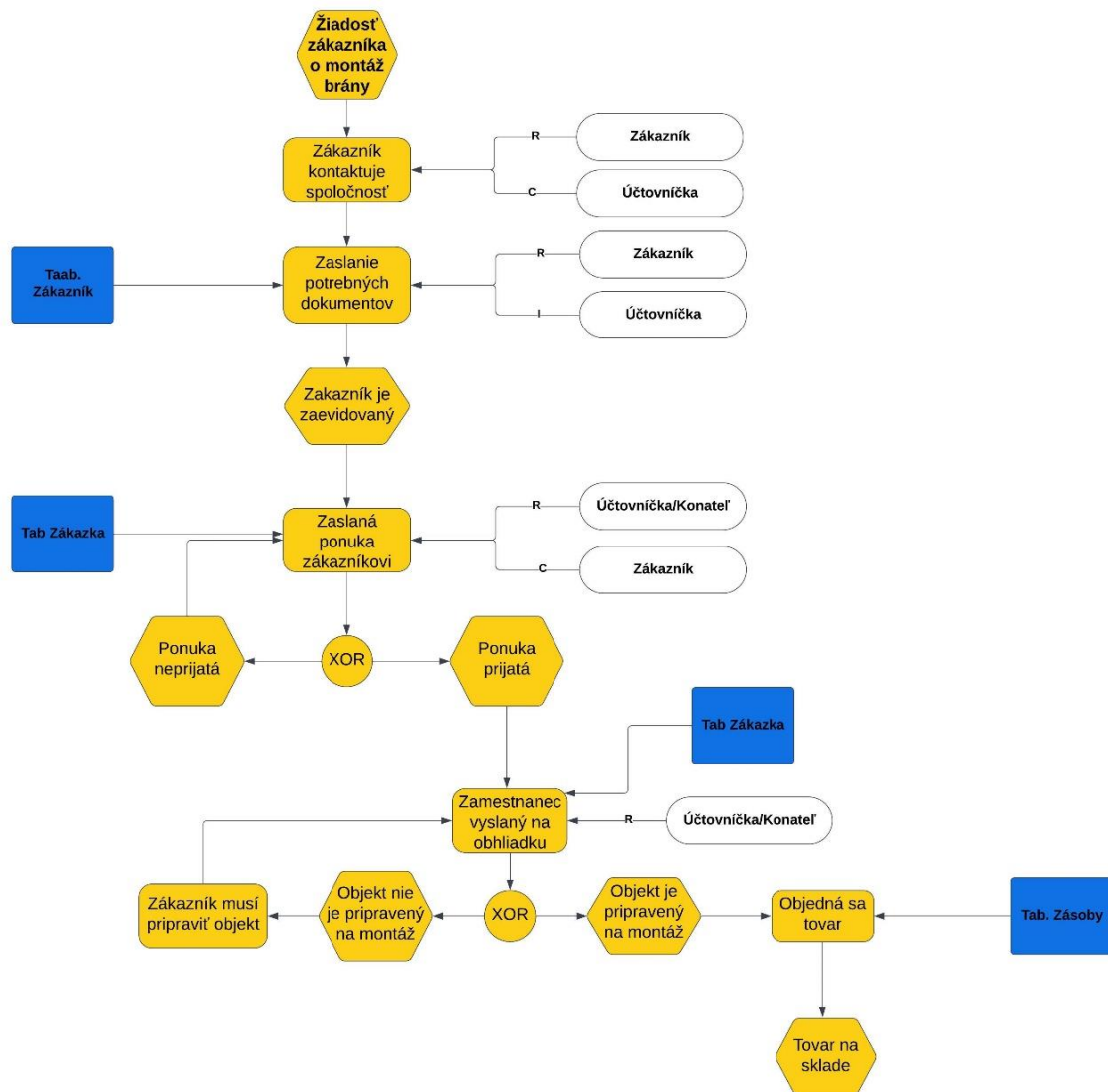
informáciami. A na tieto dva nedostatky naráža aj ďalší a to je zdĺhavé vyhľadávanie informácií. Či už sa jedná iba pri jednoduchých hľadaní medzi papierovými dokumentami alebo potom už pri náročnejších analýzach, kde je potrebné zozbierať informácie z viacerých zdrojov a odniesť si z toho určitý poznatok.

Ako hrozby firma môže očakávať dva druhy poškodenia dát a to fyzických alebo tých na serveri. Pri poškodení fyzických dokumentoch hrozí, že budú zničené požiarom, prírodnou katastrofou alebo iným nepriamym zničením. Ale môže dôjsť aj k úmyselnej krádeži dokumentov treťou osobou.

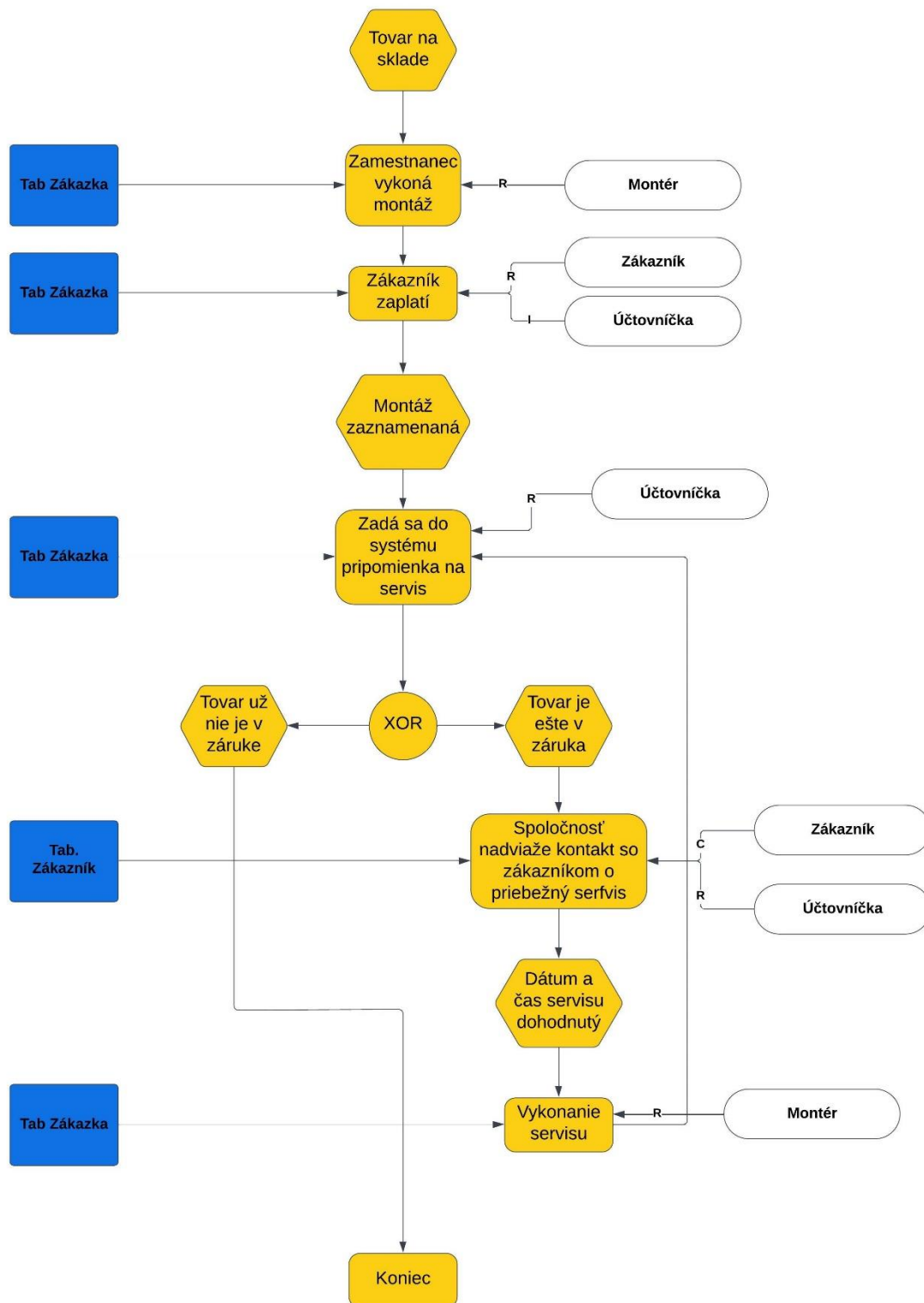
Ak by spoločnosť mala v záujme vylepšenie svojej aktuálnej databáze môže zvoliť medzi dvoma spomenutými - implementácie nového databázového systému alebo vylepšenie stávajúcej databáze.

2.8 EPC diagram

Na priblíženie primárneho procesu vo firme ako je predaj produktu zákazníkovi je v záverečnej práci využitý Event-Driven Process Chain diagram. Tento proces je zaznamenaný od prvotného kontaktu so zákazníkom až po jeho konečný, kedy mu bolo doručený finálny produkt a aj splnený servis, na ktorý má zákazník nárok.



Obrázok 30: EPC Diagram predaja tovaru časť 1. (Zdroj: Vlastné spracovanie)



Obrázok 31: EPC Diagram predaja tovaru časť 2. (Zdroj: Vlastné spracovanie)

2.8.1 Slovný popis EPC diagramu

Kontakt medzi spoločnosťou a zákazníkom nastáva v momente, kedy zákazník osloví firmu o vykonanie služieb, ktoré poskytujú. Zákazník má na výber medzi fyzickým oslovením na kamennej predajni v Košiciach alebo určitou formou neosobnej komunikácie (e-mail, telefónny rozhovor). Po kontaktovaní firma vyžiada od zákazníka potrebné informácie, na základe ktorých vyhotovia ponuku. Tieto informácie sú ako kontaktné údaje, služba (servis alebo montáž), produkt, o ktorý má zákazník záujem. V prípade montáže garážovej brány, tak musí zákazník poskytnúť aj rozmery otvoru, do ktorého bude garážová brána montovaná.

V momente, keď zákazník pošle všetky potrebné informácie a dokumenty, tak firma si založí tieto údaje medzi ostatných zákazníkov a využije ju na už spomínaný interný dokument zákazka. Po analýze požiadaviek klienta, firma vyhotoví ponuku, ktorú následne obratom zašle. Ak zákazník ponuku neprijme, tak môže pomeniť svoje požiadavky alebo sa dohodnúť so spoločnosťou na iných podmienkach.

Ak už zákazníkovi ponuka vyhovuje, tak spoločnosť vyšle zamestnanca na obhliadku priestorov, kde má byť práca vykonaná. Na mieste zamestnanec skontroluje, či zaslané informácie ohľadom otvoru a pripravenosti priestorov odpovedajú realite. Ak je potrebné ešte vykonať nejaké úpravy, tak je o tom zákazník informovaný a musí tieto nedostatky opraviť. Akonáhle je priestor pripravený k montáži, spoločnosť objedná produkt, ktorý si zákazník želal. V prípade garážovej brány na mieru, sa produkt už začne vyrábať o niečo skôr ako sa pripraví priestor na montáž z dôvodu dlhšej výroby.

V momente prijatia tovaru od dodávateľa na sklad, tak okrem vytvorenia interného dokladu príjemky sa ešte informuje o tom zákazník, aby sa mohli dohodnúť na montáži. Nakoľko sú produkty objednávané priamo na dopyt zákazníka a snažia sa ho namontovať čo najskôr. Z tohto dôvodu firma nevyžaduje žiadny špeciálny skladový systém, ktorý by riešil uskladnenie, zoznam uskladneného tovaru a ostatné služby takýchto systémov. Po vykonanej montáži zákazník zaplatí za poskytnutú službu a montáž sa zaznamená ako vykonaná. Spoločnosť poskytuje pri vlastnej montáži autorizovaný servis na pohony po dĺžku 5 rokov a na krídla garážovej brány 10 rokov. Po montáži sa tieto dátumy zaznamenajú do Apple Reminderu, aby každý rok v približnom dátume pripomenuli spoločnosti, že na ktoré zákazky je potrebné vykonať bezplatný servis. Ak tovar už nie je v tejto záručnej dobe, tak spoločnosť tiež môže chodiť na pravidelný servis ale za túto službu si už zákazník normálne zaplatí.

2.9 Zhodnotenie spoločnosti

Spoločnosť so svojimi dlhoročnými skúsenosťami je možné už považovať za silnú a stabilnú spoločnosť na trhu s garážovými bránami. Veľa z interných procesov už má naučené a vie čo je pre ňu najlepšie. Na základe SWOT analýzy boli zistené silné stránky súčasnej databázy, kde je najdôležitejšia asi jej jednoduchosť a jej poznanie. No analýza odhalila aj jej slabé stránky, ktoré prevažne súviseli s prepojením dát a ich komplexnejšia analýza. Ak by firma chcela poistiť svoju pozíciu alebo expandovať, tak by mala na nich popracovať. Aj napriek nedostatkom je databáza funkčná a účinná ale isto by privítala určité zmeny a vylepšenia.

3 Vlastný návrh riešenia

Nasledujúca kapitola sa zaoberá návrhom vlastnej databáze pre spomínanú spoločnosť. Databáza má spĺňať požiadavky spoločnosti, ktoré zadala po spoločnej konzultácii. Cieľom navrhutej databáze je zabezpečenie zvýšenie prehľadnosti údajov, zefektívneniu interných procesov a k zlepšeniu podnikových analýz. Nakoľko návrh databáze sa môže považovať za zložitejší proces, tak je táto časť rozdelená na tri menšie kroky. Zo začiatku sa práca zaoberá konceptuálnym návrhom, neskôr sa zameriava na logický návrh a v tretej časti sa vyhotovuje fyzický návrh databázy.

3.1 Zadanie od spoločnosti

Spoločnosť vyžaduje databázu, ktorá bude schopná evidovať základné informácie o jednotlivých zákazkách ako sú typ zákazky, kedy bola vykonaná, ktorý zamestnanec bol na ňu vyslaný a ďalšie potrebné informácie. Taktiež ma uchovávať základné informácie o zákazníkoch spoločnosti pre ďalšiu potrebu. Na jednotlivé zákazky je potrebné evidovať, ktorý tovar bol montovaný. Tým pádom sa majú uchovávať aj informácie o tovare a dodávateľov. Databáza má mať aj skladový charakter, kde spoločnosť zaujíma tovar a jeho množstvo na príslušnom sklade.

3.2 Konceptuálny návrh

Táto časť kapitoly sa zameriava na postupnú tvorbu konceptuálneho návrhu databáze na základe zadaných požiadaviek spoločnosti. Konceptuálny návrh predstavuje kľúčový krok v tvorbe databáze, nakoľko pokladá základnú štruktúru. Podstatou tejto podkapitoly je určenie všetkých entít, ktoré je potrebné evidovať v databáze. Podľa požiadaviek spoločnosti je potrebné určiť nielen entity ale aj následné vzťahy medzi nimi. Správne určenie entít a ich vzťahov medzi nimi predstavujú dôležitý základ pre efektívnu databázu a správu dátových tokov v systéme.

3.2.1 Identifikácia entít

V tomto kroku je potrebné identifikovať všetky entity, ktoré sa nachádzajú v databáze spoločnosti. Každá takáto entita má svoj jednoduchý popis na priblíženie uchovávaných informácií a taktiež frekvencia alebo počet výskytov danej entity.

Tabuľka 9 Entity databáze (Zdroj: Vlastné spracovanie)

Entita	Popis	Počet
Zamestnanec	osobné údaje o zamestnancoch	jednotky
Zákazka	informácie o jednotlivých zákazkách	stovky
Zákazník	základné informácie o zákazníkovi	stovky
Zákazník údaje	kontaktné údaje zákazníka	stovky
Tovar	popis tovaru a identifikačné informácie	stovky
Sklad	základné údaje o skladoch	jednotky
Dodávatelia	kontaktné údaje dodávateľov	jednotky
Dodávatelia údaje	kontaktné údaje dodávateľa	stovky

3.2.2 Identifikácia relácií

Táto kapitola priamo vychádza z predchádzajúcej, kde boli určené entity, ktorými sa zaoberá databáza. Tieto entity neexistujú len samostatne, ale sú navzájom prepojené a vytvárajú určité vzťahy. V tejto kapitole sa tieto vzťahy ďalej preskúmavajú a sú priblížené v nasledujúcej tabuľke.

Tabuľka 10 Vzájomné vzťahy entít (Zdroj: Vlastné spracovanie)

Názov entity	Názov väzby	Názov entity
Zamestnanec	je vyslaný na	Zákazka
Zákazník	objednal	Zákazka
Zákazník údaje	majú	Zákazník
Tovar	je ku	Zákazka
Sklad	má pridelený	Tovar
Dodávateľ	dodáva	Tovar
Dodávateľ údaje	majú	Dodávateľ

3.2.3 Určenie kardinality

Určenie kardinality poskytuje hlbší pohľad na vzťahy medzi entitami a to koľkokrát môže jedna entita vstupovať vo vzťahu k druhej entite. Tento proces umožňuje lepšie pochopenie štruktúry databáze a vzájomných vzťahov medzi entitami. Vďaka tomuto je jednoduchšie rozhodovanie o prípadnej potrebnej dekompozícii tabuliek.

Tabuľka 11 Kardinalita väzieb (Zdroj: Vlastné spracovanie)

Názov entity	Kardinalita	Názov entity
Zamestnanec	M:N	Zákazka
Zákazník	1:N	Zákazka
Zákazník údaje	N:1	Zákazník
Tovar	N:1	Zákazka
Sklad	1:N	Tovar
Dodávateľ	1:N	Tovar
Dodávateľ údaje	N:1	Dodávateľ

3.3 Logický návrh

V nasledujúcej časti návrhu databáze sa práca zaoberá logickým návrhom, ktorý prevádza konceptuálny návrh do konkrétnej podoby databázovej schémy. Počas tohto návrhu sa detailne identifikujú jednotlivé entity (tabuľky), ich atribúty a vzťahy medzi nimi. Cieľom logického návrhu je vytvoriť štruktúru databázy, ktorá zachováva integritu dát, optimalizuje výkon a efektívne spĺňa požiadavky spoločnosti.

Súčasťou jednotlivých tabuliek je názov atribútov, v akom dátovom type budú atribúty uložené, či sa jedná o primárny alebo cudzí kľúč a krátky popis. V návrhu sa ráta aj s výskytom nulových hodnôt, ktorý je zabezpečený podmienkou NOT NULL.

3.3.1 Tabuľka Zamestnanec

V tabuľke Zamestnanec sa nachádzajú všetky základné informácie o zamestnancoch spoločnosti. Každému zamestnancovi je priradené jedinečné identifikačné číslo, pod ktorým je uložený v systéme. Okrem neho sa v tabuľke nachádza meno a priezvisko, dátum narodenia, kontaktné údaje a rodné číslo. Ako ID mohlo byť aj použité rodné číslo ale z dôvodu nízkeho počtu zamestnancov (<10) je voľba jedinečného identifikačného čísla jednoduchšia.

Tabuľka 12 Tabuľka Zamestnanec (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
zamestnanec_ID	integer	PK	áno	identifikačné číslo zamestnanca
meno	varchar 20	-	áno	krstné meno zamestnanca
priezvisko	varchar 20	-	áno	priezvisko zamestnanca
dat_narodenia	Datetime	-	áno	dátum narodenia zamestnanca
tel_cislo	varchar 13	-	áno	telefónne číslo na zamestnanca
e_mail	varchar 50	-	áno	pracovný email zamestnanca
rod_cislo	varchar 10	-	áno	rodné číslo zamestnanca

3.3.2 Tabuľka Vyslaný

Tabuľka vyslaný slúži na dekompozičné účely vzťahu medzi tabuľkami Zamestnanec a Zákazka. Tento vzťah vznikol z dôvodu, že na jednu zákazku je možné vyslať aj viacerých zamestnancov a jeden zamestnanec je vyslaný na niekoľko zákaziek. Tým pádom sa v tabuľke nachádzajú iba identifikačné čísla spomínaných dvoch tabuliek.

Tabuľka 13 Tabuľka Vyslaný (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
zamestnanec_ID	integer	FK	áno	identifikačné číslo zamestnanca, ktorý bol vyslaný na zákazku.
zakazka_ID	integer	FK	áno	identifikačné číslo zákazky.

3.3.3 Tabuľka Zákazka

Tabuľku Zákazka je možné považovať za tabuľku faktov, nakoľko je na ňu naviazaných niekoľko väzieb z ostatných tabuliek v databáze. Je možné označiť túto tabuľku ako najdôležitejšiu tabuľku databázy, keďže jej účelom je zozbierať všetky potrebné informácie o jednotlivých zákazkách. Tabuľka má formu momentálneho montážneho listu, ktorá spoločnosť využíva pri informovaní montérov čo majú urobiť.

Jednotlivé väzby sú prepojené na základe identifikačného čísla s tabuľkami zamestnanec, zákazník a tovar. Identifikačné číslo tovaru nemá podmienku Not Null z dôvodu, že zákazka môže mať typ servis, kedy sa nemontuje žiadny tovar. Tým pádom sa v tabuľka nachádza typ zákazky, dátum vykonania montáže a servisu, adresa montáže, ktorá môže byť iná ako tá pri kontaktných údajov zákazníka. Okrem týchto informácia tabuľka uchováva aj sumu, za ktorú bola práca vykonaná, sumu, za ktorú sa predal montovaný tovar a čiastka spolu. Všetky sumy v databáza sú uvádzané v eurách.

Tabuľka 14 Tabuľka Zákazka (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
zakazka_ID	integer	PK	áno	identifikačné číslo zákazky
zakaznik_ID	integer	FK	áno	identifikačné číslo zákazníka, ktorému sa vyhotovuje zákazka

tovar_ID	varchar 15	FK	nie	identifikačné číslo tovaru, ktorý je montovaný
stav_zak_ID	integer	FK	áno	identifikačné číslo stavu zákazky, v ktorom sa momentálne nachádza
typ_zak	varchar 6	-	áno	zákazka môže byť servis alebo montáž
dat_mon	datetime	-	nie	dátum vykonania montáže
dat_ser	datetime	-	nie	dátum vykonania servisu
mesto	varchar 30	-	áno	mesto vykonania zákazky
ulica	varchar 30	-	áno	ulica a číslo vykonania zákazky
psc	varchar 6	-	áno	poštové smerovacie číslo adresy vykonania zákazky
suma_praca	decimal(10,2)	-	nie	suma za, ktorá spoločnosť poskytla vykonanú prácu
pred_cena	decimal(10,2)	-	nie	predajná cena tovaru
ciastka_spolu	súčet záznamov suma_praca a pred_cena	-	nie	celková suma za vyhotovenú zákazku

3.3.4 Tabuľka Stav_zak

Tabuľka Stav_zak slúži ako číselník pre tabuľku Zákazka, ktorý bude obsahovať informácie o tom v ako stave sa nachádza konkrétna Zákazka. Môže nadobudnúť hodnotu čaká sa na zákazníka, ak je potrebné spraviť ešte nejaké úpravy. Taktiež sa môže ešte čakať na objednaný tovar. Ak už je všetko pripravené na montáž alebo už prebieha tak stav takejto zákazky je označený ako v procese vyhotovenia a samozrejme môže byť ešte zákazka je hotová.

Tabuľka 15 Stav zákazky (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
stav_zak	integer	PK	áno	identifikačné číslo stavu zákazky
stav	varchar 25	-	áno	popis v akom stave sa nachádza zákazka

3.3.5 Tabuľka Zákazník

Tabuľka Zákazníka obsahuje informácie o jednotlivých zákazníkoch spoločnosti. Medzi tie základné uchovávané údaje patrí identifikačné číslo zákazníka, meno a priezvisko. Ďalej sa tam nachádza aj atribút typ zákazníka, pri ktorom sa v databáze rozlišuje právnická a fyzická osoba. V tabuľke sa taktiež nachádza väzba na tabuľku Zak_udaje, ktorá obsahuje podrobnejšie informácie o jednotlivých zákazníkoch.

Tabuľka 16 Tabuľka Zákazník (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
zakaznik_ID	integer	PK	áno	identifikačné číslo zákazníka
meno	varchar 20	-	áno	krstné meno zákazníka
priezvisko	varchar 20	-	áno	priezvisko zákazníka
typ_zak	varchar 2	-	áno	či sa jedná o fyzickú (FO) alebo právnickú osobu
zak_udaje	integer	FK	áno	identifikačné číslo údajov a zákazníkovi

3.3.6 Tabuľka Zak_udaje

V tabuľke Zak_udaje sa nachádzajú podrobnejšie informácie o kontaktných údajov zákazníka. Táto tabuľka vznikla z dôvodu, že niektorí zákazníci môžu mať viac kontaktných údajov ako len jeden a pre väčšiu prehľadnosť je potrebné uchovávať všetky kontaktné údaje. Patrí sem email, telefónne číslo, adresa zákazníka a v prípade potreby DIČ a IČO ak sa jedná o právnickú osobu.

Tabuľka 17 Tabuľka Zak_udaje (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
zak_udaje_ID	integer	PK	áno	identifikačné číslo údajov k zákazníkovi
e_mail	varchar 50	-	áno	e-mail na zákazníka
tel_cislo	varchar 13	-	áno	telefónne číslo na zákazníka
mesto	varchar 30	-	áno	mesto bydliska zákazníka
ulica	varchar 30	-	áno	ulica bydliska zákazníka
psc	varchar 6	-	áno	psč bydliska zákazníka
nazov	varchar 30	-	nie	názov spoločnosti v prípade PO
dic	varchar 10	-	nie	DIČ v prípade platca dane
ico	varchar 8	-	nie	IČO v prípade PO zákazníka

3.3.7 Tabuľka Tovar

Tabuľka Tovar uchováva všetky potrebné informácie o tovare, s ktorým spoločnosť obchoduje. Okrem bežných informácií ako sú názov, množstvo, merná jednotka a nákupná cena sa v tabuľke nachádzajú aj väzby na ostatné tabuľky. Tieto väzby sa odkazujú na tabuľku Dodávateľ a tabuľku Skladu. Na základe týchto väzieb je možné určiť, ktorý dodávateľ je zodpovedný za tento tovar a na ktorom sklade sa tovar nachádza.

Tabuľka 18 Tabuľka Tovar (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
tovar_ID	varchar 8	PK	áno	identifikačné číslo tovaru – kód tovaru
dod_ico	varchar 8	FK	áno	identifikačné číslo (IČO) dodávateľa tovaru
sklad_ID	integer	FK	áno	identifikačné číslo skladu, v ktorom sa tovar nachádza
nazov	varchar 50	-	áno	názov tovaru
mnozstvo	integer	-	áno	množstvo tovaru, ktoré sa nachádza na sklade
merna_jednotka	varchar 3	-	áno	merná jednotka, v ktorej sa uchováva množstvo tovaru (metre, kusy)
nakup_cena	integer	-	áno	cena, za ktorú spoločnosť tovar objednala

3.3.8 Tabuľka Sklad

Tabuľka sklad pozostáva iba zo základných informácií ako je identifikačné číslo skladu, adresa skladu a slovný popis na čo je sklad určený a čo sa v ňom uskladňuje.

Tabuľka 17 Tabuľka Sklad (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
sklad_ID	integer	PK	áno	identifikačné číslo skladu
ulica	varchar 30	-	áno	ulica, na ktorej sa sklad nachádza
psc	varchar 6	-	áno	PSČ adresy skladu
popis	varchar 150	-	áno	slovný opis na čo je sklad zameraný a aký typ tovaru uskladňuje

3.3.9 Tabuľka Dodávateľa

V tabuľke Dodávateľa sa nachádzajú základné informácie o dodávateľoch spoločnosti. Primárnym kľúčom tabuľky je IČO dodávateľskej spoločnosti. Ďalej sa v tabuľke nachádza názov spoločnosti, DIČ v prípade platcu dane a väzba na tabuľku Dod_udaje, kde sa nachádzajú podrobnejšie údaje o dodávateľovi.

Tabuľka 19 Tabuľka Dodávateľa (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
dod_ico	varchar 8	PK	áno	IČO dodávateľa
nazov	varchar 20	-	áno	názov dodávateľskej spoločnosti
dic	varchar 10	-	nie	v prípade platca dani, tak DIČ spoločnosti
dod_udaje_ID	integer	FK	áno	identifikačné číslo kontaktných údajov dodávateľa

3.3.10 Tabuľka Dod_udaje

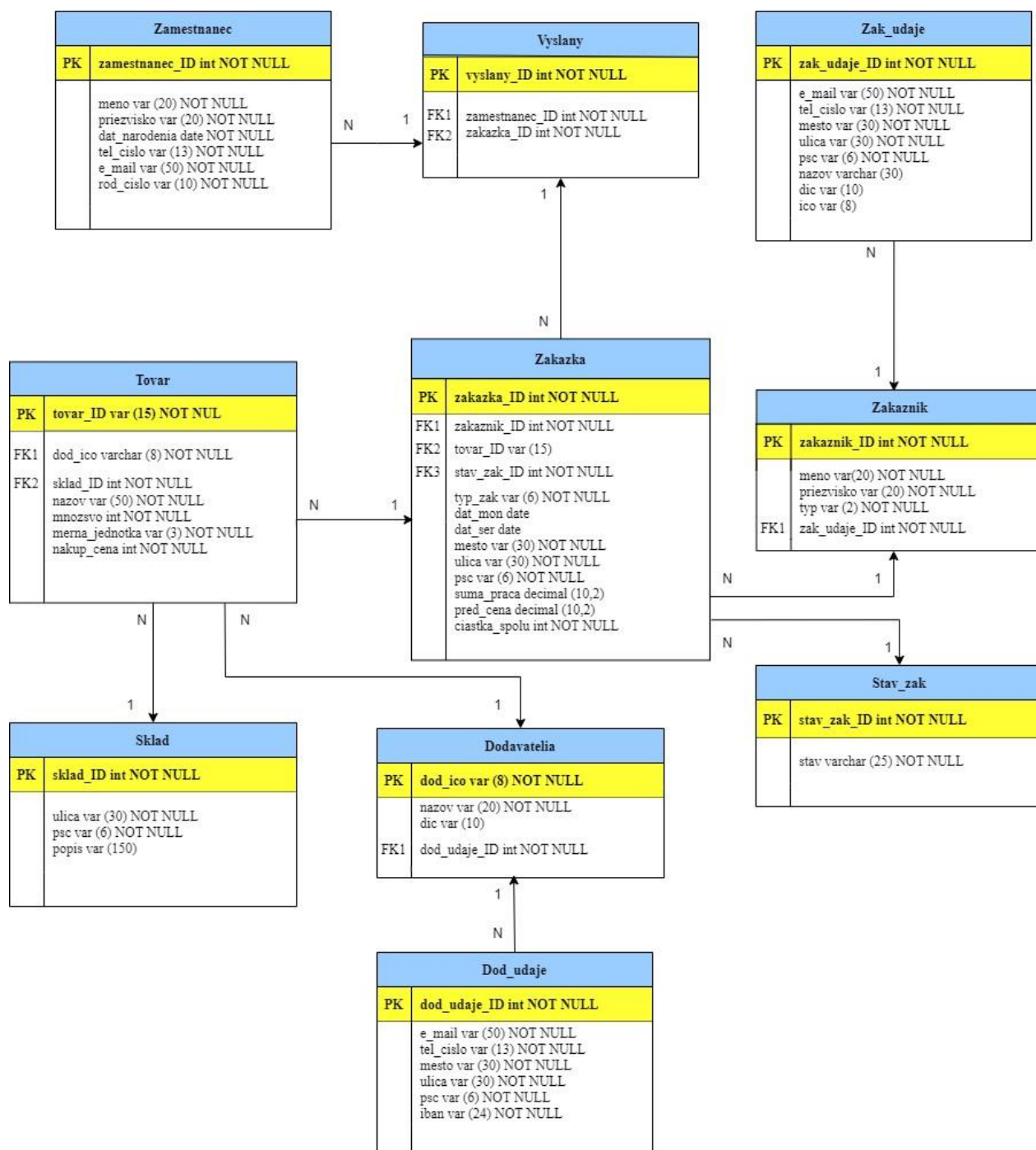
Tabuľka Dod_udaje uchováva podrobné kontaktné údaje na jednotlivých dodávateľov. Medzi takéto údaje patri email, telefónne číslo, sídlo spoločnosti a bankové spojenie na úhradu faktúr. Účelom tabuľky je uchovávať všetky potrebné kontaktné údaje dodávateľa a to aj v prípade ak by ich mal viac, z toho dôvodu sú prepojené väzbou na tabuľku Dodávateľa.

Tabuľka 20 Tabuľka Dod_udaje (Zdroj: Vlastné spracovanie)

Atribút	Dátový typ	PK/FK/-	Podmienka Not Null	Popis
dod_udaje_ID	integer	PK	áno	identifikačné číslo údajov k dodávateľovi
e_mail	varchar 50	-	áno	e-mail na dodávateľa
tel_cislo	varchar 13	-	áno	telefónne číslo na dodávateľa
mesto	varchar 30	-	áno	mesto bydliska dodávateľa
ulica	varchar 30	-	áno	ulica bydliska dodávateľa
psc	varchar 6	-	áno	PSČ adresa dodávateľa
iban	varchar 24	-	áno	bankové spojenie dodávateľa

3.3.11 ER Diagram

ER diagram databáze pozostáva z jednotlivých tabuliek a väzbami medzi nimi. Kardinalita týchto väzieb je označená Chenovou notáciou, kde 1 predstavuje, že entita v jednej strane môže byť spojená práve iba s jednou entitou na strane druhej a označenie N znamená, že entita v jednej strane vzťahu môže mať ľubovoľný počet spojených entít na strane druhej. [1]



Obrázok 32 ER diagram (Zdroj: Vlastné spracovanie)

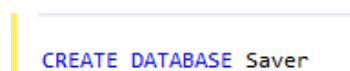
3.4 Fyzický návrh

Táto časť záverečnej práce sa zaoberá samotným fyzickým návrhom databázy pomocou SQL skriptu. Cieľom časti ja navrhnuť celú plne funkčnú databázu na základe už vytvoreného ER diagramu. Taktiež sa bude zaoberať návrhom spúšťačov, pohľadov a procedúr, ktoré by mali pomôcť k zefektívneniu databáze. Celý fyzický návrh vychádza z logického návrhu.

3.4.1 Výber prostredia databázy a tvorba databázy

Nakoľko fyzický návrh už priamo súvisí s navrhnutím aj SQL skriptu databázy je potrebné zvoliť prostredie pre ňu. Databázový systém bude implementovaný do systému od spoločnosti Microsoft a to presne MS SQL Server Management Studio 18. Zvolený databázový systém spĺňa všetky požiadavky, ktoré sú kladené na databázu a je dostatočne kompatibilný s rôznymi ostatnými nástrojmi pre prípadné ďalšie spracovanie dát.

Ako prvý krok je potrebné vytvoriť databázu, ktorá bude niesť názov spoločnosti.



Obrázok 33 Vytvorenie databázy (Zdroj: Vlastné spracovanie)

3.4.2 Tvorba tabuliek a cudzích kľúčov

Aby bola tabuľka prehľadná a funkčná je potrebné vytvoriť tabuľky pre všetky dôležité entity. Vytváranie tabuliek vzniká na základe použitia príkazu CREATE TABLE, ktorý nasleduje všetkými potrebnými atribútmi a ich dátovými typmi. Pri vzniku tabuliek sú hneď určené aj cudzie kľúče, ktoré predstavujú väzby medzi jednotlivými tabuľkami.

```
-- Vytvorenie tabuľky Tovar
CREATE TABLE Tovar (
    tovar_ID varchar(15) PRIMARY KEY,
    dod_ico varchar(8) NOT NULL,
    sklad_ID int NOT NULL,
    nazov varchar(50) NOT NULL,
    mnozstvo int NOT NULL,
    merna_jednotka varchar(3) NOT NULL,
    nakup_cena int NOT NULL,
    pred_cena int NOT NULL,
    FOREIGN KEY (dod_ico) REFERENCES Dodavatelja(dod_ico),
    FOREIGN KEY (sklad_ID) REFERENCES Sklad(sklad_ID)
);
GO
```

Obrázok 34 Tvorba tabuliek (Zdroj: Vlastné spracovanie)

3.4.3 Vloženie dát

Kvôli účinnejšiemu zisteniu fungovania databázy sú do nej nahraté vstupné dáta. Tieto dáta predstavujú testovacie dáta, ktoré sú takmer identické s tými, ktoré spoločnosť využíva a spracúva.

```
INSERT INTO Zakazka (zakaznik_ID, tovar_ID, stav_zak_ID, typ_zak, dat_mon, dat_ser, ulica, mesto, psc, suma_prace, pred_cena)
VALUES
(1, 'INTI1', 4, 'Montáž', '2023-03-03', NULL, 'Hlavná 12', 'Košice', '040 01', 10, 27.40),
(2, '1106030RE', 4, 'Montáž', '2023-03-08', NULL, 'Hviezdoslavova 22', 'Kysak', '080 01', 160, 449),
(2, 'PMC307', 4, 'Montáž', '2023-03-08', NULL, 'Hviezdoslavova 22', 'Kysak', '080 01', 60, 429),
(3, 'PMC307', 4, 'Montáž', '2023-03-16', NULL, 'Kukucínova 6', 'Prešov', '080 05', 40, 429),
(4, 'INTI1', 4, 'Montáž', '2023-03-18', NULL, 'Gagarinova 74', 'Košice', '040 12', 10, 27.40),
(5, NULL, 4, 'Servis', '2022-03-15', '2023-03-22', 'Budmír 19', 'Budimír', '044 43', NULL, NULL),
(6, 'SMC730', 4, 'Montáž', '2023-03-25', NULL, 'Svätopolukova 26', 'Košice', '040 01', 60, 522),
(7, NULL, 4, 'Servis', '2022-04-02', '2023-04-04', 'Štúrova 45', 'Košice', '040 01', NULL, NULL),
(8, 'F1000', 4, 'Montáž', '2023-04-05', NULL, 'Bardejovská 23', 'Košice', '040 11', 80, 384),
(9, 'NICEPO', 4, 'Montáž', '2023-04-08', NULL, 'Hlavná 6', 'Košice', '040 01', 10, 5.50),
(10, 'F1000', 4, 'Montáž', '2023-04-09', NULL, 'Námestie osloboditeľov 12', 'Košice', '040 01', 80, 384);
```

Obrázok 35 Vloženie dát do tabuliek (Zdroj: Vlastné spracovanie)

3.4.4 Triggery

Nasledujúca kapitola sa zaoberá tvorbou triggerov (spúšťačov). Triggery boli vytvorené k zefektívneniu fungovania a používania databázy.

Trigger Aktuálne stavy

Trigger slúži na zrýchlenie zapisovania údajov, ktoré súvisia so stavom zákazky. Jeho úlohou je na základe pridaného záznamu v tabuľke Vyslany pomocou ID zákazky zmeniť stav tejto zákazky na V procese vyhotovenia. To je možné za predpokladu, že ak je zaevidované, ktorý zamestnanec je vyslaný na zákazku, tak už sa na nej pracuje.

```
CREATE TRIGGER Aktualne_stavy
ON Vyslany
AFTER INSERT
AS
BEGIN
] UPDATE Zakazka
SET stav_zak_ID = 3
FROM Zakazka z
INNER JOIN inserted i ON z.zakazka_ID = i.zakazka_ID;
END;
```

Obrázok 36 Trigger aktuálne stavy (Zdroj: Vlastné spracovanie)

Triggery na automatizovanie skladov

V tejto podkapitole sa nachádzajú dva triggery, ktorých úlohou je zabezpečiť väčší prehľad v množstve tovarov na sklade. Prvý spomínaný spúšťač zníži množstvo určitého tovaru na základe jeho identifikačného čísla pri vytvorení záznamu v tabuľke zakazka.

```

CREATE TRIGGER Trg_ZnizenieMnozstva
ON Zakazka
AFTER INSERT
AS
BEGIN
    DECLARE @tovarID varchar(15);
    SELECT @tovarID = inserted.tovar_ID FROM inserted;

    UPDATE Tovar
    SET mnozstvo = mnozstvo - 1
    WHERE tovar_ID = @tovarID;
END;

```

Obrázok 37 Trigger na zníženie množstva (Zdroj: Vlastné spracovanie)

Okrem znižovania množstva môže nastať aj situácia kedy sa množstvo zvýši a to pri upravení zle zadanej zákazky. Takže v prípade odstránenie jedného záznamu zákazky sa automaticky zvýši aj množstvo tovaru s rovnakým identifikačným číslom.

```

CREATE TRIGGER Trg_ZvysenieMnozstva
ON Zakazka
AFTER DELETE
AS
BEGIN
    DECLARE @tovarID varchar(15);
    SELECT @tovarID = deleted.tovar_ID FROM deleted;

    UPDATE Tovar
    SET mnozstvo = mnozstvo + 1
    WHERE tovar_ID = @tovarID;
END;

```

Obrázok 38 Trigger na zvýšenie množstva (Zdroj: Vlastné spracovanie)

3.4.5 Pohľady

Táto kapitola obsahuje tvorbu pohľadov a ich využitie. Práve tieto pohľady boli vytvorené na základe toho, čo spoločnosť najčastejšie potrebuje zobrazit' alebo zistiť. Väčšina pohľadov je využívaná na dennej alebo týždennej báze.

Inventúra

Prvý pohľad je nazvaný Inventúra a to z dôvodu, že má používateľovi ukázať všetok tovar, ktorý sa v skladoch nachádza. Účelom pohľadu nie je len zobrazit' stav na skladoch ale aj informácie o tovare a dodávateľovi v prípade následnej potreby objednať tovar.

```

CREATE VIEW Inventura
AS
SELECT t.tovar_ID, t.nazov AS Tovar,
       d.nazov AS Dodávateľ, du.e_mail,
       du.tel_cislo, t.mnozstvo, t.merna_jednotka,
       t.nakup_cena, t.sklad_ID
FROM   dbo.Tovar t
       LEFT JOIN dbo.Dodavatelja d ON t.dod_ico = d.dod_ico
       LEFT JOIN dbo.Dod_udaje du ON d.dod_udaje_ID = du.dod_udaje_ID

SELECT * FROM Inventura

```

Obrázok 39 Pohľad Inventúra (Zdroj: Vlastné spracovanie)

	tovar_ID	Tovar	mnozstvo	merna_jednotka	nakup_cena	Dodávateľ	e_mail	tel_cislo
1	1106030RE	Výklopná brána Berry Pearlgrain	2	ks	350	Hormann	hormann@email.com	+421912345678
2	F1000	CAME FERNI nevrátny motor 230V	1	ks	322	Came	came@email.com	+421955555555
3	CH703	Garážová brána Renomatic	3	ks	749	Hormann	hormann@email.com	+421912345678
4	INTI1	NICE Era INTI1 diaľkový ovládač	13	ks	22	Nice	nice@email.com	+421912345678
5	NICEPO	NICE Portos konzola pre fotobunky BF	8	ks	4	Nice	nice@email.com	+421912345678
6	PMC307	Pohon Promatic 4	4	ks	352	Hormann	hormann@email.com	+421912345678
7	SMC730	Pohon SupraMatic P 4	6	ks	489	Hormann	hormann@email.com	+421912345678

Obrázok 40 Výsledná tabuľka pohľadu Inventúra (Zdroj: Vlastné spracovanie)

Plán práce

Aby spoločnosť mohla naplánovať svoju prácu čo najefektívnejšie a prípadne naplánovať lepšie ako už je rozvrhnuté, tak si vyberie všetky zákazky, ktoré sú už v procese vyhotovenia. To znamená tie, na ktorých sa už pracuje alebo čaká na začatie ale všetko je pripravené na to. Taktiež sa zobrazia všetky potrebné informácie ohľadom týchto zákaziek ako dátum, adresa montáže/servisu a údaje o zákazníkovi.

```

CREATE VIEW Plan_prace
AS
SELECT CASE WHEN za.dat_ser IS NULL THEN za.dat_mon ELSE za.dat_ser END AS [Dátum montáže],
       za.zakazka_ID, zk.meno + ' ' + zk.priezvisko AS Zákazník,
       za.ulica + ', ' + za.mesto AS Adresa,
       zu.tel_cislo AS [Telefónné číslo],
       t.nazov AS Tovar
FROM   dbo.Zakazka za
       LEFT JOIN dbo.Tovar t ON t.tovar_ID = za.tovar_ID
       LEFT JOIN dbo.Zakaznik zk ON zk.zakaznik_ID = za.zakaznik_ID
       LEFT JOIN dbo.Zak_udaje zu ON zu.zak_udaje_ID = zk.zak_udaje_ID
WHERE  (za.stav_zak_ID = 3)

SELECT * FROM Plan_prace

```

Obrázok 41 Pohľad Plán práce (Zdroj: Vlastné spracovanie)

	Dátum montáže	zakazka_ID	Zákazník	Adresa	Telefónné číslo	Tovar
1	2023-04-05	9	Lucia Sedláková	Bardejovská 23, Košice	+421900789012	CAME FERNI nevrátny motor 230V
2	2023-04-08	10	Jana Hrušková	Hlavná 6, Košice	+421942345678	NICE Portos konzola pre fotobunky BF

Obrázok 42 Výsledná tabuľka pohľadu Plán práce (Zdroj: Vlastné spracovanie)

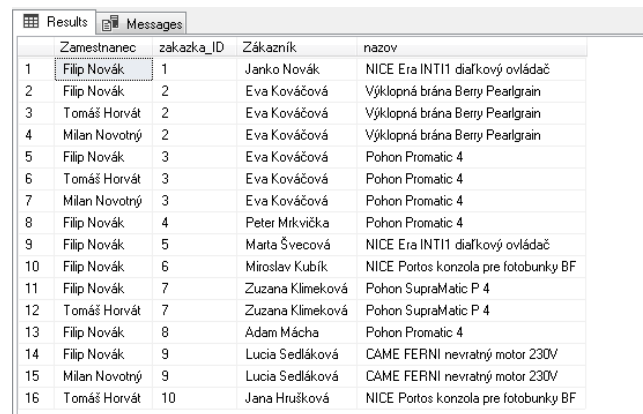
Vyslaný zamestnanci

Pohľad vyslaný zamestnanci slúži na zobrazenie, ktorý zamestnanec bol vyslaný, na ktorú zákazku. Takýto pohľad je v prípade zhrnutia práce daného zamestnanca alebo v prípade zistenia, na ktorú zákazku je naplánovaný alebo už vyslaný zamestnanec.

```
CREATE VIEW Vyslany_zamestnanci
AS
SELECT zc.meno + ' ' + zc.priezvisko AS Zamestnanec,
       za.zakazka_ID,
       zk.meno + ' ' + zk.priezvisko AS Zákaznik,
       t.nazov
FROM dbo.Zamestnanec zc
INNER JOIN dbo.Vyslany vy ON zc.zamestnanec_ID = vy.zamestnanec_ID
INNER JOIN dbo.Zakazka za ON vy.zakazka_ID = za.zakazka_ID
INNER JOIN dbo.Zakaznik zk ON za.zakaznik_ID = zk.zakaznik_ID
INNER JOIN dbo.Tovar t ON t.tovar_ID = za.tovar_ID

SELECT * FROM Vyslany_zamestnanci
```

Obrázok 43 Pohľad Vyslaný zamestnanci (Zdroj: Vlastné spracovanie)



	Zamestnanec	zakazka_ID	Zákaznik	nazov
1	Filip Novák	1	Janko Novák	NICE Era INT11 diaľkový ovládač
2	Filip Novák	2	Eva Kováčová	Výklopná brána Berry Pearlgrain
3	Tomáš Horvát	2	Eva Kováčová	Výklopná brána Berry Pearlgrain
4	Milan Novotný	2	Eva Kováčová	Výklopná brána Berry Pearlgrain
5	Filip Novák	3	Eva Kováčová	Pohon Promatic 4
6	Tomáš Horvát	3	Eva Kováčová	Pohon Promatic 4
7	Milan Novotný	3	Eva Kováčová	Pohon Promatic 4
8	Filip Novák	4	Peter Mirkvička	Pohon Promatic 4
9	Filip Novák	5	Marta Švecová	NICE Era INT11 diaľkový ovládač
10	Filip Novák	6	Miroslav Kubík	NICE Portos konzola pre fotobunky BF
11	Filip Novák	7	Zuzana Klimeková	Pohon SupraMatic P 4
12	Tomáš Horvát	7	Zuzana Klimeková	Pohon SupraMatic P 4
13	Filip Novák	8	Adam Mácha	Pohon Promatic 4
14	Filip Novák	9	Lucia Sedláková	CAME FERNI nevrátny motor 230V
15	Milan Novotný	9	Lucia Sedláková	CAME FERNI nevrátny motor 230V
16	Tomáš Horvát	10	Jana Hrušková	NICE Portos konzola pre fotobunky BF

Obrázok 44 Výsledná tabuľka pohľadu Vyslaný zamestnanci (Zdroj: Vlastné spracovanie)

Analýza predaja

Už z názvu je jasné, že nasledujúci pohľad slúži na jednoduchú analýzu predaja. Táto analýza skúma oblasti, kde sa vyskytuje najviac zákaziek. Nakoľko firma pôsobí na celom východnom Slovensku ale primárne Košice, Košice – okolie a pre určité priemyselné zákazky firmám v Prešove, tak sa zistí koľko zákaziek a za akú priemernú cenu bolo na týchto adresách.

```
CREATE VIEW Analyza_predaja
AS
SELECT CASE WHEN mesto = 'Košice' THEN 'Košice' WHEN mesto = 'Prešov' THEN 'Prešov' ELSE 'Okolie' END AS Oblast,
       COUNT(zakazka_ID) AS [Počet zákaziek],
       AVG(ciastka_spolu) AS [Priemerná cena]
FROM dbo.Zakazka
GROUP BY CASE WHEN mesto = 'Košice' THEN 'Košice' WHEN mesto = 'Prešov' THEN 'Prešov' ELSE 'Okolie' END

SELECT * FROM Analyza_predaja
```

Obrázok 45 Pohľad Analýza predaja (Zdroj: Vlastné spracovanie)

	Oblasť	Počet zákaziek	Priemerná cena
1	Košice	7	231.471428
2	Okolie	3	366.000000
3	Prešov	1	469.000000

Obrázok 46 Výsledná tabuľka pohľadu Analýza predaja (Zdroj: Vlastné spracovanie)

3.4.6 Procedúry

Kapitola procedúry približuje vznik a návrh vytvorených procedúr v databáze. Navrhnuté operácie predstavujú zložitejšie operácie, ktoré sa vyskytujú v spoločnosti a jej databáze pravidelnejšie.

Zákazky z minulého roka

Táto procedúra má na základe zadaného mesiaca zobrazit' všetky zákazky, ktoré sa vyskytli v spoločnosti v tomto mesiaci ale minulý rok. Takéto zobrazenie má slúžiť na zistenie, ktoré zákazky majú ešte nárok na pravidelný servis a ktoré už nie. Okrem zobrazenia informácií o zákazke sa zobrazia aj údaje o zákazníkovi v prípade naplánovania servisu.

```

CREATE PROCEDURE ZakMinRok
    @Mesiac INT
AS
BEGIN
    DECLARE @PreviousYearDate DATE
    DECLARE @SelectedMonthFirstDay DATE
    DECLARE @SelectedMonthLastDay DATE

    SET @PreviousYearDate = DATEADD(YEAR, -1, GETDATE())
    SET @SelectedMonthFirstDay = DATEFROMPARTS(YEAR(@PreviousYearDate), @Mesiac, 1)
    SET @SelectedMonthLastDay = EOMONTH(@SelectedMonthFirstDay)

    SELECT za.zakazka_ID, za.dat_mon, za.dat_ser,
           za.ulica + ', ' + za.mesto AS Adresa,
           t.nazov,
           zk.meno + ' ' + zk.priezvisko AS Zákazník,
           zu.tel_cislo
    FROM Zakazka za
    LEFT JOIN Tovar t ON t.tovar_ID = za.tovar_ID
    LEFT JOIN Zakaznik zk ON zk.zakaznik_ID = za.zakaznik_ID
    LEFT JOIN Zak_udaje zu ON zu.zak_udaje_ID = zk.zak_udaje_ID
    WHERE MONTH(CASE WHEN za.dat_ser IS NOT NULL THEN za.dat_ser ELSE za.dat_mon END) = @Mesiac
END

EXEC ZakMinRok 3

```

Obrázok 47 Procedúra Zákazky z minulého roka (Zdroj: Vlastné spracovanie)

zakazka_ID	dat_mon	dat_ser	Adresa	nazov	Zákazník	tel_cislo
1	2023-03-03	NULL	Hlavná 12, Košice	NICE Era INT11 diaľkový ovládač	Janko Novák	+421901112334
2	2023-03-08	NULL	Hviezdoslavova 22, Kysak	Výklopná brána Berry Pearlgrain	Eva Kováčová	+421901230562
3	2023-03-08	NULL	Hviezdoslavova 22, Kysak	Pohon Promatic 4	Eva Kováčová	+421901230562
4	2023-03-16	NULL	Kukucinova 6, Prešov	Pohon Promatic 4	Peter Mirkvička	+421944556677
5	2023-03-18	NULL	Gagarinova 74, Košice	NICE Era INT11 diaľkový ovládač	Martha Švecová	+421909123456
6	2022-03-15	2023-03-22	Budmír 19, Budmír	NICE Portos konzola pre fotobunky BF	Miroslav Kubík	+421918765432
7	2023-03-25	NULL	Svatopolukova 26, Košice	Pohon SupraMatic P 4	Zuzana Klimeková	+421904567890

Obrázok 48 Výsledná tabuľka procedúry Zákazky z minulého roka (Zdroj: Vlastné spracovanie)

Pridanie množstva tovaru

Pridanie množstva na sklade prebieha vždy v prípade prijatia tovaru na sklad. Nakoľko spoločnosť často nakupuje a predáva rovnaký typ tovaru, tak aby nebolo potrebné upravovať atribút množstvo v tabuľke tovaru vždy, tak vznikla táto procedúra. Funguje na základe zadania ID tovaru a pri tomto identifikačnom čísle zvýši jeho množstvo o jedno a vyhodnotí úspešnosť operácie.

```
CREATE PROCEDURE Pridaj_mnozstvo
    @TovarID varchar(15)
AS
BEGIN
    IF EXISTS (SELECT 1 FROM Tovar WHERE tovar_ID = @TovarID)
    BEGIN
        UPDATE Tovar
        SET mnozstvo = mnozstvo + 1
        WHERE tovar_ID = @TovarID;
        -- Vrätene správy o úspešnom vykonaní operácie
        PRINT 'Množstvo pre tovar s ID ' + @TovarID + ' bolo úspešne zvýšené o 1.';
    END
    ELSE
    BEGIN
        -- Vrätene chybovej správy v prípade, že záznam s daným tovar_ID neexistuje
        PRINT 'Tovar s ID ' + @TovarID + ' neexistuje.';
    END
END;

EXECUTE Pridaj_mnozstvo F1000
```

Obrázok 49 Procedúra Pridanie množstva tovaru (Zdroj: Vlastné spracovanie)

```
(1 row affected)
Množstvo pre tovar s ID F1000 bolo úspešne zvýšené o 1.

Completion time: 2024-03-27T11:45:07.4308352+01:00
```

Obrázok 50 Výsledná správa po pridaní tovaru (Zdroj: Vlastné spracovanie)

Naplánovanie servisov

Plánovanie servisov je operácia, ktorá robí pravdepodobne najväčšie ťažkosti spoločnosti. Z tohto dôvodu vznikla procedúra, ktorá automaticky naplánuje servisy všetkým zákazníkom pre určitý mesiac na základe informácií z minulého roka. Tieto servisy naplánuje iba zákazníkom, ktoré majú ešte nárok na pravidelný servis a to je 5 rokov podľa predajných podmienok. Okrem naplánovania servisov tak aj automaticky priradí údaje k tejto novovzniknutej zákazke.

```

CREATE PROCEDURE Navrhni_servisy
@VybranyMesiac DATE
AS
BEGIN
    DECLARE @AktualnyRok INT;
    DECLARE @MinulyRok INT;
    SET @AktualnyRok = YEAR(GETDATE());
    SET @MinulyRok = @AktualnyRok - 1;

    SELECT zakazka_ID, dat_mon
    INTO #MinuleZakazky
    FROM Zakazka
    WHERE YEAR(dat_mon) = @MinulyRok
    AND MONTH(dat_mon) = MONTH(@VybranyMesiac)
    OR (YEAR(dat_ser) = @MinulyRok
    AND MONTH(dat_ser) = MONTH(@VybranyMesiac));

    DECLARE @ZakazkaID INT, @DatMon DATE;
    DECLARE cur CURSOR FOR
    SELECT zakazka_ID, dat_mon
    FROM #MinuleZakazky;
    OPEN cur;
    FETCH NEXT FROM cur INTO @ZakazkaID, @DatMon;

    WHILE @@FETCH_STATUS = 0
    BEGIN
        IF DATEDIFF(YEAR, @DatMon, GETDATE()) <= 5
        BEGIN
            INSERT INTO Zakazka (zakaznik_ID, tovar_ID, dat_mon, dat_ser, stav_zak_ID, typ_zak, ulica, mesto, psc, suma_prace, pred_cena)
            SELECT zakaznik_ID, tovar_ID, @DatMon, DATEADD(YEAR, DATEDIFF(YEAR, @DatMon, GETDATE()), @DatMon), 2, 'Servis', ulica, mesto, psc, NULL, NULL
            FROM Zakazka
            WHERE zakazka_ID = @ZakazkaID;
        END;

        FETCH NEXT FROM cur INTO @ZakazkaID, @DatMon;
    END;

    CLOSE cur;
    DEALLOCATE cur;

    DROP TABLE #MinuleZakazky;
END;

```

Obrázok 51 Procedúra Naplánovanie servisov (Zdroj: Vlastné spracovanie)

3.5 Zhodnotenie navrhnutej databázy

Navrhnutá databáza pre spomínanú spoločnosť ponúka niekoľko významných výhod, ktoré prispievajú k efektívnej evidencii dát a zlepšeniu riadeniu podniku. Nakoľko je databáza navrhnutá na základe podrobných konzultáciách so spoločnosťou, tak je šitá presne na mieru jej požiadavkám. Navrhnuté riešenie poskytuje jednotné a centralizované úložisko údajov, ktoré sú potrebné pre každodenný chod spoločnosti. Tieto údaje je možné si zobrazit' aj rôznych pohľadov pre ich lepšie pochopenie a uľahčenie práce s nimi.

Záver

Cieľom tejto práce bolo na základe analýzy súčasného stavu databáze vybranej spoločnosti navrhnúť jej úpravu a vylepšenie k dosiahnutiu efektívnej správe dát. Návrh databáze prebiehal s neustálou komunikáciou so spoločnosťou aby tento databázový systém odpovedal ich presným požiadavkám. Na pochopenie interných procesov, fungovania spoločnosti a komunikácie so zákazníkom pomohla odpracovaná prax. Vďaka tomuto navrhnutý databázový systém presne odpovedá požiadavkám spoločnosti a obsahuje všetko čo je pre spoločnosť potrebné.

Najväčším problémom spoločnosti bolo plánovanie servisov a následné plánovanie zákaziek. Navrhnutá databáza dokáže zobrazit' všetky zákazky na základe zadaných parametrov, ktoré si vyžadujú servis a zároveň aj navrhne dátumy, kedy by bolo vhodné vykonať servis. Okrem plánovania servisov je databázový systém schopný aj zobrazit' plán všetkých zákaziek, ktoré treba vykonať.

Navrhnutá databáza umožní spoločnosti zefektívnenie denno-denných procesov ako aj tých, ktoré sa nevyskytujú až tak často. Na základe nej dokáže používateľ zistiť všetky potrebné informácie o zákazníkoch, ich zákazkách alebo kľudne aj o tovare na skladoch.

Zoznam použitých zdrojov

- [1] KOCH, Miloš. *Datové a funkční modelování*. Brno: Akademické nakladatelství CERM, 2004. ISBN 80-214-2724-8.
- [2] JAIN, Sanjay, 2023. *Data vs. Information: What's the Difference?* [online]. [cit. 2023-11-08]. Dostupné z: <https://bloomfire.com/blog/data-vs-information/>
- [3] ČERNÝ, Jan, 2017. *Data, informace a cesta ke znalostem* [online]. [cit. 2023-11-08]. Dostupné z: <https://www.informacnigramotnost.cz/data-informace-znalosti/>
- [4] KROENKE, David; AUER, David J a GONER, Jakub. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.
- [5] ŠKODA, Jan, 2019. *Databázové systémy* [online]. [cit. 2023-11-12]. Dostupné z: <https://www.vovcr.cz/odz/tech/393/page03.html>
- [6] ORACLE, 2023. *What is a Database* [online]. [cit. 2023-11-15]. Dostupné z: <https://www.oracle.com/in/database/what-is-database/#link1>
- [7] LUTKEVICH, Ben a Adam HUGHES, 2023. *Database (DB)* [online]. [cit. 2023-11-15]. Dostupné z: <https://www.techtarget.com/searchdatamanagement/definition/database>
- [8] Autor neznámy, *Database* [online]. [cit. 2023-11-15]. Dostupné z: <https://www.javatpoint.com/what-is-database>
- [9] SUSZTEROVA, Sandra, 2023. *What Is a Data Model?* [online]. [cit. 2024-04-07]. Dostupné z: <https://www.gooddata.com/blog/what-a-data-model/>
- [10] Autor neznámy, 2024. *What is a data model?* [online]. [cit. 2024-04-11]. Dostupné z: <https://www.erwin.com/learn/data-model.aspx>
- [11] Autor neznámy, *Types of Database Model* [online]. [cit. 2024-04-12]. Dostupné z: <https://www.studytonight.com/dbms/database-model.php>
- [12] Autor neznámy. *Relace mezi tabulkami v datovém modelu* [online]. [cit. 2024-04-12]. Dostupné z: <https://support.microsoft.com/cs-cz/office/relace-mezi-tabulkami-v-datov%C3%A9m-modelu-533dc2b6-9288-4363-9538-8ea6e469112b>
- [13] Autor neznámy, *Description of the database normalization basics* [online], 2023. [cit. 2024-04-12]. Dostupné z: <https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>

- [14] OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.
- [15] LACKO, Luboslav. *Mistrovství v SQL Server 2012*. Brno: Computer Press, 2013. ISBN 978-80-251-3773-4.
- [16] Autor neznámý, *SQL Joins* [online]. [cit. 2024-04-14]. Dostupné z: https://www.w3schools.com/sql/sql_join.asp
- [17] Autor neznámý, *SQL Views* [online], 2024. [cit. 2024-04-14]. Dostupné z: <https://www.geeksforgeeks.org/sql-views/>
- [18] Autor neznámý, *SQL | Procedures in PL/SQL* [online], 2018. [cit. 2024-04-22]. Dostupné z: <https://www.geeksforgeeks.org/sql-procedures/>
- [19] Autor neznámý, *SQL | Triggers* [online], 2022. [cit. 2024-04-22]. Dostupné z: <https://www.geeksforgeeks.org/sql-triggers/>
- [20] Autor neznámý, *Conceptual Data Modeling* [online]. [cit. 2024-04-22]. Dostupné z: <https://www.erwin.com/learn/conceptual.aspx>
- [21] CONOLLY, Thomas, Carolyn E BEGG a Richard HOLOWCZAK. *Mistrovství – databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009, 584 s. : il. ISBN 978-80-251-2328-7
- [22] Kříž, Jiří. *Relační databáze – RD* [online]. Vysoké učení technické v Brně, [cit. 2024-04-24]
- [23] Autor neznámý, *What is a logical data model?* [online]. [cit. 2024-04-24]. Dostupné z: <https://www.tibco.com/glossary/what-is-a-logical-data-model>
- [24] Autor neznámý, *Introduction of ER Model* [online], 2024. [cit. 2024-04-24]. Dostupné z: <https://www.geeksforgeeks.org/introduction-of-er-model/>
- [25] TICONG, Liz, 2024. *What is a Physical Data Model? Definition and Examples* [online]. [cit. 2024-04-28]. Dostupné z: <https://www.datamation.com/big-data/physical-data-models/>
- [26] JENIFA, Ashlin, 2023. *Physical Data Model in DBMS: What You Need to Know* [online]. [cit. 2024-04-28]. Dostupné z: <https://geekflare.com/physical-data-model/>
- [27] NORTH, Sarah a Xiaohua XU. *Database Systems Development Life Cycle* [online]. [cit. 2024-04-28]. Dostupné z: <https://pressbooks.pub/decisions/chapter/565/>

Zoznam použitých obrázkov

Obrázok 1: Kódovanie a dekódovanie dát [1]	14
Obrázok 2 Lineárny dátový model [1]	17
Obrázok 3 Hierarchický dátový model [1]	18
Obrázok 4 Sieťový dátový model [1]	18
Obrázok 5 Relačný dátový model [1]	19
Obrázok 6 Terminológia z pohľadu teórie množín [1]	20
Obrázok 7 Terminológia z pohľadu teórie relácií [1]	20
Obrázok 8 Typy kľúčov (Zdroj: vlastné spracovanie)	22
Obrázok 9 Vzťah 1:1 [1]	22
Obrázok 10 Vzťah 1:N [1]	23
Obrázok 11 Vzťah N:M [1]	23
Obrázok 12 Dekompozícia vzťahu N:M [1]	23
Obrázok 13 Dekompozícia vzťahu N:M v tabuľkách [1]	23
Obrázok 14 Tabuľka Študent pred úpravou na 3. normálnu formu (Zdroj: vlastné spracovanie)	25
Obrázok 15 Normalizácia na 3. normálnu formu (Zdroj: Vlastné spracovanie)	25
Obrázok 16 Príkaz CREATE (Zdroj: vlastné spracovanie)	28
Obrázok 17 Príkaz INSERT (Zdroj: Vlastné spracovanie)	28
Obrázok 18 Príkaz ALTER (Zdroj: vlastné spracovanie)	28
Obrázok 19 Príkaz DROP (Zdroj: Vlastné spracovanie)	28
Obrázok 20 Základné príkazy výberu informácií (Zdroj: Vlastné spracovanie)	28
Obrázok 21 INNER JOIN [15]	29
Obrázok 22 LEFT JOIN [15]	29
Obrázok 23 RIGHT JOIN [15]	29
Obrázok 24 FULL OUTER JOIN [15]	30
Obrázok 25 Tvorba pohľadov (Zdroj: Vlastné spracovanie)	31
Obrázok 26 Tvorba procedúr (Zdroj: Vlastné spracovanie)	31
Obrázok 27 Volanie procedúr (Zdroj: Vlastné spracovanie)	31
Obrázok 28: Logo spoločnosti [27]	34
Obrázok 29: SWOT analýza databáze (Zdroj: Vlastné spracovanie)	42
Obrázok 30: EPC Diagram predaja tovaru časť 1. (Zdroj: Vlastné spracovanie)	44
Obrázok 31: EPC Diagram predaja tovaru časť 2. (Zdroj: Vlastné spracovanie)	45

Obrázok 32 ER diagram (Zdroj: Vlastné spracovanie)	60
Obrázok 33 Vytvorenie databáze (Zdroj: Vlastné spracovanie)	61
Obrázok 34 Tvorba tabuliek (Zdroj: Vlastné spracovanie)	61
Obrázok 35 Vloženie dát do tabuliek (Zdroj: Vlastné spracovanie)	62
Obrázok 36 Trigger aktuálne stavy (Zdroj: Vlastné spracovanie)	62
Obrázok 37 Trigger na zníženie množstva (Zdroj: Vlastné spracovanie)	63
Obrázok 38 Trigger na zvýšenie množstva (Zdroj: Vlastné spracovanie)	63
Obrázok 39 Pohľad Inventúra (Zdroj: Vlastné spracovanie)	64
Obrázok 40 Výsledná tabuľka pohľadu Inventúra (Zdroj: Vlastné spracovanie)	64
Obrázok 41 Pohľad Plán práce (Zdroj: Vlastné spracovanie)	64
Obrázok 42 Výsledná tabuľka pohľadu Plán práce (Zdroj: Vlastné spracovanie)	64
Obrázok 43 Pohľad Vyslaný zamestnanci (Zdroj: Vlastné spracovanie)	65
Obrázok 44 Výsledná tabuľka pohľadu Vyslaný zamestnanci (Zdroj: Vlastné spracovanie)	65
Obrázok 45 Pohľad Analýza predaja (Zdroj: Vlastné spracovanie)	65
Obrázok 46 Výsledná tabuľka pohľadu Analýza predaja (Zdroj: Vlastné spracovanie) ... 66	66
Obrázok 47 Procedúra Zákazky z minulého roka (Zdroj: Vlastné spracovanie)	66
Obrázok 48 Výsledná tabuľka procedúry Zákazky z minulého roka (Zdroj: Vlastné spracovanie)	66
Obrázok 49 Procedúra Pridanie množstva tovaru (Zdroj: Vlastné spracovanie)	67
Obrázok 50 Výsledná správa po pridaní tovaru (Zdroj: Vlastné spracovanie)	67
Obrázok 51 Procedúra Naplánovanie servisov (Zdroj: Vlastné spracovanie)	68

Zoznam použitých tabuliek

Tabuľka 1 Agregáčn� funkcie (Zdroj: Vlastn� spracovanie).....	30
Tabuľka 2: Prijat� fakt�ry v software Omega (Zdroj: Vlastn� spracovanie)	38
Tabuľka 3: Vydan� fakt�ry v software Omega (Zdroj: Vlastn� spracovanie)	39
Tabuľka 4: Sklady v software Omega (Zdroj: Vlastn� spracovanie)	39
Tabuľka 5: Z�soby v software Omega (Zdroj: Vlastn� spracovanie).....	40
Tabuľka 6: Zamestnanci v software Omega (Zdroj: Vlastn� spracovanie).....	40
Tabuľka 7: Inform�cie o z�kazke (Zdroj: Vlastn� spracovanie)	41
Tabuľka 8: Inform�cie v mont�žnom liste (Zdroj: Vlastn� spracovanie).....	41
Tabuľka 9 Entity datab�ze (Zdroj: Vlastn� spracovanie)	49
Tabuľka 10 Vz�jomn� vzťahy entit (Zdroj: Vlastn� spracovanie).....	49
Tabuľka 11 Kardinalita v�zieb (Zdroj: Vlastn� spracovanie)	50
Tabuľka 12 Tabuľka Zamestnanec (Zdroj: Vlastn� spracovanie).....	51
Tabuľka 13 Tabuľka Vyslan�y (Zdroj: Vlastn� spracovanie).....	52
Tabuľka 14 Tabuľka Z�kazka (Zdroj: Vlastn� spracovanie)	52
Tabuľka 15 Stav z�kazky (Zdroj: Vlastn� spracovanie)	54
Tabuľka 16 Tabuľka Z�kazn�k (Zdroj: Vlastn� spracovanie)	55
Tabuľka 17 Tabuľka Zak_udaje (Zdroj: Vlastn� spracovanie).....	56
Tabuľka 18 Tabuľka Tovar (Zdroj: Vlastn� spracovanie)	57
Tabuľka 19 Tabuľka Dod�vatelia (Zdroj: Vlastn� spracovanie).....	58
Tabuľka 20 Tabuľka Dod_udaje (Zdroj: Vlastn� spracovanie).....	59

Zoznam použitých skratiek

DML – Data Manipulation Language

DDL – Data Definiton Language

DCL – Data Control Language

ER diagram – Entity Relationship diagram

FK – Foreign Key

PK – Primary Key

SQL – Structured Query Language

Zoznam príloh

Príloha č. I: Tvorba tabuliek

Príloha č. II: Vloženie testovacích dát

Príloha č. III: Triggers

Príloha č. IV: Pohľady

Príloha č. V: Procedúry

Príloha č. I: Tvorba tabuliek

-- Vytvorenie tabuľky Zamestnanec

```
CREATE TABLE Zamestnanec (  
    zamestnanec_ID int PRIMARY KEY IDENTITY,  
    meno varchar(20) NOT NULL,  
    priezvisko varchar(20) NOT NULL,  
    dat_narodenia date NOT NULL,  
    tel_cislo varchar(13) NOT NULL,  
    e_mail varchar(50) NOT NULL,  
    rod_cislo varchar(10) NOT NULL  
);  
GO
```

-- Vytvorenie tabuľky Dod_udaje

```
CREATE TABLE Dod_udaje (  
    dod_udaje_ID int PRIMARY KEY IDENTITY,  
    e_mail varchar(50) NOT NULL,  
    tel_cislo varchar(13) NOT NULL,  
    mesto varchar(30) NOT NULL,  
    ulica varchar(30) NOT NULL,  
    psc varchar(6) NOT NULL,  
    iban varchar(24) NOT NULL  
);  
GO
```

-- Vytvorenie tabuľky Sklad

```
CREATE TABLE Sklad (  
    sklad_ID int PRIMARY KEY IDENTITY,  
    ulica varchar(30) NOT NULL,  
    psc varchar(6) NOT NULL,  
    popis varchar(150)  
);
```

GO

-- Vytvorenie tabuľky Dodavatelia

```
CREATE TABLE Dodavatelia (  
    dod_ico varchar(8) PRIMARY KEY,  
    nazov varchar(20) NOT NULL,  
    dic varchar(10),  
    dod_udaje_ID int NOT NULL,  
    FOREIGN KEY (dod_udaje_ID) REFERENCES Dod_udaje(dod_udaje_ID)  
);
```

GO

-- Vytvorenie tabuľky Tovar

```
CREATE TABLE Tovar (  
    tovar_ID varchar(15) PRIMARY KEY,  
    dod_ico varchar(8) NOT NULL,  
    sklad_ID int NOT NULL,  
    nazov varchar(50) NOT NULL,  
    mnozstvo int NOT NULL,
```

```
merna_jednotka varchar(3) NOT NULL,  
nakup_cena int NOT NULL,  
pred_cena int NOT NULL,  
FOREIGN KEY (dod_ico) REFERENCES Dodavatelia(dod_ico),  
FOREIGN KEY (sklad_ID) REFERENCES Sklad(sklad_ID)
```

```
);
```

```
GO
```

```
-- Vytvorenie tabuľky Zak_udaje
```

```
CREATE TABLE Zak_udaje (
```

```
zak_udaje_ID int PRIMARY KEY IDENTITY,
```

```
e_mail varchar(50) NOT NULL,
```

```
tel_cislo varchar(13) NOT NULL,
```

```
mesto varchar(30) NOT NULL,
```

```
ulica varchar(30) NOT NULL,
```

```
psc varchar(6) NOT NULL,
```

```
nazov varchar (30),
```

```
dic varchar(10),
```

```
ico varchar(8)
```

```
);
```

```
GO
```

```
-- Vytvorenie tabuľky Zakaznik
```

```
CREATE TABLE Zakaznik (
```

```
zakaznik_ID int PRIMARY KEY IDENTITY,
```

```
meno varchar(20) NOT NULL,
```

```

    priezvisko varchar(20) NOT NULL,
    typ varchar(2) NOT NULL,
    zak_udaje_ID int NOT NULL,
    FOREIGN KEY (zak_udaje_ID) REFERENCES Zak_udaje(zak_udaje_ID)
);
GO
-- Vytvorenie tabuľky Stav_zak
CREATE TABLE Stav_zak (
    stav_zak_ID int PRIMARY KEY IDENTITY,
    stav varchar(25) NOT NULL
);
GO
-- Vytvorenie tabuľky Zakazka
CREATE TABLE Zakazka (
    zakazka_ID int PRIMARY KEY IDENTITY,
    zakaznik_ID int NOT NULL,
    tovar_ID varchar(15),
    stav_zak_ID int NOT NULL,
    typ_zak varchar(6) NOT NULL,
    dat_mon date,
    dat_ser date,
    mesto varchar(30) NOT NULL,
    psc varchar(6) NOT NULL,
    suma_prace decimal(10, 2),

```



```

pred_cena decimal(10, 2),
ciastka_spolu decimal(10, 2)
        FOREIGN KEY (zakaznik_ID) REFERENCES Zakaznik(zakaznik_ID),
FOREIGN KEY (tovar_ID) REFERENCES Tovar(tovar_ID),
FOREIGN KEY (stav_zak_ID) REFERENCES Stav_zak(stav_zak_ID)
);
GO
-- Vymazanie pôvodného stĺpca
ALTER TABLE Zakazka
DROP COLUMN ciastka_spolu;
-- Pridanie nového stĺpca s definovaním výrazu pre výpočet
ALTER TABLE Zakazka
ADD ciastka_spolu AS (ISNULL(suma_prace, 0) + ISNULL(pred_cena, 0));
-- Vytvorenie tabuľky Vyslany
CREATE TABLE Vyslany (
        vyslany_ID int PRIMARY KEY IDENTITY,
        zamestnanec_ID int NOT NULL,
        zakazka_ID int NOT NULL,
        FOREIGN KEY (zamestnanec_ID) REFERENCES Zamestnanec(zamestnanec_ID),
        FOREIGN KEY (zakazka_ID) REFERENCES Zakazka(zakazka_ID)
);
GO

```

Príloha č. II: Vloženie testovacích dát

-- Nahrание dát do tabuľky Zamestnanec

```
INSERT INTO Zamestnanec (meno, priezvisko, dat_narodenia, tel_cislo, e_mail, rod_cislo)
```

```
VALUES
```

```
('Lucia', 'Schmiedlová', '1988-03-15', '+421912345678', 'lucia.schmiedlova@email.com',  
'8803151234'),
```

```
('Peter', 'Novák', '1973-08-20', '+421955555555', 'peter.novak@email.com', '7308201234'),
```

```
('Martin', 'Novák', '1975-09-10', '+421912345678', 'martin.novak@email.com', '7509101234'),
```

```
('Filip', 'Novák', '1996-12-05', '+421955555555', 'filip.novak@email.com', '9612051234'),
```

```
('Tomáš', 'Horvát', '2000-09-25', '+421912345678', 'tomas.horvat@email.com', '0009251234'),
```

```
('Milan', 'Novotný', '1979-04-30', '+421955555555', 'milan.novotny@email.com',  
'7904301234');
```

-- Nahrание údajov do tabuľky Dod_udaje

```
INSERT INTO Dod_udaje (e_mail, tel_cislo, mesto, ulica, psc, iban)
```

```
VALUES
```

```
('hormann@email.com', '+421912345678', 'Senec', 'Diaľničná cesta 51', '90301',  
'SK1234567890'),
```

```
('came@email.com', '+421955555555', 'Košice', 'Vodná 23', '04001', 'SK0987654321'),
```

```
('nice@email.com', '+421912345678', 'Košice', 'Štúrová 14', '04001', 'SK5678901234');
```

-- Nahrание údajov do tabuľky Dodavatelja

```
INSERT INTO Dodavatelja (dod_ico, nazov, dic, dod_udaje_ID)
```

```
VALUES
```

```
('35880350', 'Hormann', '2021804598', 1),
```

```
('87654321', 'Came', '87654321', 2),
```

```
('56789012', 'Nice', NULL, 3);
```

-- Vloženie údajov do tabuľky Sklad

```
INSERT INTO Sklad (ulica, psc, popis)
```

```
VALUES
```

```
('Zborovská 4', '04001', 'Sklad určený na drobný materiál, pohony a príslušenstvo potrebné priamo v predajni/showroome'),
```

```
('Zborovská 50', '04001', 'Sklad určený na menšie garážové brány, lamely a pohony'),
```

```
('Bardejovská 46', '04011', 'Sklad určený na lamely a garážové brány');
```

-- Vloženie údajov do tabuľky Tovar

```
INSERT INTO Tovar (tovar_ID, dod_ico, sklad_ID, nazov, mnozstvo, merna_jednotka, nakup_cena, pred_cena)
```

```
VALUES
```

```
('1106030RE',35880350, 3, 'Výklopná brána Berry Pearlgrain', 2, 'ks', 350, 449),
```

```
('CH703', 35880350, 3, 'Garážová brána Renomatic', 3, 'ks', 749, 879),
```

```
('PMC307', 35880350, 2, 'Pohon Promatic 4', 4, 'ks', 352, 429),
```

```
('SMC730', 35880350, 2, 'Pohon SupraMatic P 4', 6, 'ks', 489, 522),
```

```
('NICEPO', 56789012, 1, 'NICE Portos konzola pre fotobunky BF', 8, 'ks', 4.80, 5.50 ),
```

```
('INTI1', 56789012, 1, 'NICE Era INTI1 diaľkový ovládač', 12, 'ks', 22.50, 27.40),
```

```
('F1000', 87654321, 2, 'CAME FERNI nevratný motor 230V', 1, 'ks', 322, 384);
```

-- Vloženie dát do tabuľky Zak_udaje

```
INSERT INTO Zak_udaje (e_mail, tel_cislo, mesto, ulica, psc, nazov, dic, ico)
```

```
VALUES
```

```
('janko@gmail.com', '+42190112233', 'Košice', 'Hlavná 12', '040 01', NULL, NULL, NULL),
```

```
('eva@gmail.com', '+42191223344', 'Kysak', 'Hviezdoslavova 22', '044 81', NULL, NULL, NULL),
```

('peter@seznam.cz', '+421944556677', 'Prešov', 'Kukucínova 6', '080 05', 'Želežiarstvo Alfa', 'SK12345678', '12345678'),

('marta@centrum.sk', '+421909123456', 'Košice', 'Gagarinova 74', '040 12', NULL, NULL, NULL),

('miro@atlas.sk', '+421918765432', 'Budimír', 'Budimír 19', '044 43', NULL, NULL, NULL),

('zuzana@zoznam.sk', '+421904567890', 'Košice', 'Svätoplukova 26', '040 01', NULL, NULL, NULL),

('adam@gmail.com', '+421911234567', 'Košice', 'Štúrova 45', '040 01', NULL, NULL, NULL),

('lucia@gmail.com', '+421900789012', 'Košice', 'Bardejovská 23', '040 11', 'DPMK', 'SK98765431', '98765431'),

('jana@azet.sk', '+421942345678', 'Košice', 'Hlavná 6', '040 01', NULL, NULL, NULL),

('filip@post.sk', '+421909876543', 'Košice', 'Wupertálska 18', '040 23', 'Zlaty Dvor', 'SK87695213', '87695213');

-- Vloženie dát do tabuľky Zakaznik

INSERT INTO Zakaznik (meno, priezvisko, typ, zak_udaje_ID)

VALUES

('Janko', 'Novák', 'FO', 1),

('Eva', 'Kováčová', 'FO', 2),

('Peter', 'Mrkvička', 'PO', 3),

('Marta', 'Švecová', 'FO', 4),

('Miroslav', 'Kubík', 'FO', 5),

('Zuzana', 'Klimeková', 'FO', 6),

('Adam', 'Mácha', 'FO', 7),

('Lucia', 'Sedláková', 'PO', 8),

```

('Jana', 'Hrušková', 'FO', 9),
('Filip', 'Baláž', 'PO', 10);
-- Vloženie dát do tabuľky Stav_zak
INSERT INTO Stav_zak (stav)
VALUES
('Čaká sa na klienta'),
('Čaká sa na tovar'),
('V procese vyhotovenia'),
('Hotová');
INSERT INTO Zakazka (zakaznik_ID, tovar_ID, stav_zak_ID, typ_zak, dat_mon, dat_ser,
ulica, mesto, psc, suma_prace, pred_cena)
VALUES
(1, 'INTI1', 4, 'Montáž', '2023-03-03', NULL, 'Hlavná 12', 'Košice', '040 01', 10, 27.40),
(2, '1106030RE', 4, 'Montáž', '2023-03-08', NULL, 'Hviezdoslavova 22', 'Kysak', '080 01',
160, 449),
(2, 'PMC307', 4, 'Montáž', '2023-03-08', NULL, 'Hviezdoslavova 22', 'Kysak', '080 01',
60, 429),
(3, 'PMC307', 4, 'Montáž', '2023-03-16', NULL, 'Kukucínova 6', 'Prešov', '080 05', 40, 429),
(4, 'INTI1', 4, 'Montáž', '2023-03-18', NULL, 'Gagarinova 74', 'Košice', '040 12', 10, 27.40),
(5, NULL, 4, 'Servis', '2022-03-15', '2023-03-22', 'Budmír 19', 'Budimír', '044 43', NULL,
NULL),
(6, 'SMC730', 4, 'Montáž', '2023-03-25', NULL, 'Svätopolukova 26', 'Košice', '040 01', 60,
522),
(7, NULL, 4, 'Servis', '2022-04-02', '2023-04-04', 'Štúrova 45', 'Košice', '040 01', NULL,
NULL),
(8, 'F1000', 4, 'Montáž', '2023-04-05', NULL, 'Bardejovská 23', 'Košice', '040 11', 80, 384),

```

(9, 'NICEPO', 4, 'Montáž', '2023-04-08', NULL, 'Hlavná 6', 'Košice', '040 01', 10, 5.50),

(10, 'F1000', 4, 'Montáž', '2023-04-09', NULL, 'Námestie osloboditeľov 12', 'Košice', '040 01', 80, 384);

-- Vloženie dát do tabuľky Vyslany

INSERT INTO Vyslany (zamestnanec_ID, zakazka_ID)

VALUES

(4,1),

(4,2),

(5,2),

(6,2),

(4,3),

(5,3),

(6,3),

(4,4),

(4,5),

(4,6),

(4,7),

(5,7),

(4,8),

(4,9),

(6,9),

(5,10),

(4,11),

(5,11),

(6,11);

Príloha č. III: Triggers

```
CREATE TRIGGER Aktualne_stavy
ON Vyslany
AFTER INSERT
AS
BEGIN
    UPDATE Zakazka
    SET stav_zak_ID = 3
    FROM Zakazka z
    INNER JOIN inserted i ON z.zakazka_ID = i.zakazka_ID;
END;
```

```
CREATE TRIGGER Trg_ZnizenieMnozstva
ON Zakazka
AFTER INSERT
AS
BEGIN
    DECLARE @tovarID varchar(15);
    SELECT @tovarID = inserted.tovar_ID FROM inserted;
    UPDATE Tovar
    SET mnozstvo = mnozstvo - 1
    WHERE tovar_ID = @tovarID;
END;
```

```
CREATE TRIGGER Trg_ZvysenieMnozstva
ON Zakazka
AFTER DELETE
AS
BEGIN
    DECLARE @tovarID varchar(15);
    SELECT @tovarID = deleted.tovar_ID FROM deleted;

    UPDATE Tovar
    SET mnozstvo = mnozstvo + 1
    WHERE tovar_ID = @tovarID;
END;
```


Príloha č. IV: Pohľady

```
CREATE VIEW Analyza_predaja
```

```
AS
```

```
SELECT CASE WHEN mesto = 'Košice' THEN 'Košice' WHEN mesto = 'Prešov' THEN  
'Prešov' ELSE 'Okolie' END AS Oblasť, COUNT(zakazka_ID) AS [Počet zákaziek],
```

```
AVG(ciastka_spolu) AS [Priemerná cena]
```

```
FROM dbo.Zakazka
```

```
GROUP BY CASE WHEN mesto = 'Košice' THEN 'Košice' WHEN mesto = 'Prešov' THEN  
'Prešov' ELSE 'Okolie' END
```

```
CREATE VIEW Inventura
```

```
AS
```

```
SELECT t.tovar_ID, t.nazov AS Tovar, d.nazov AS Dodávateľ, du.e_mail, du.tel_cislo,  
t.mnozstvo, t.merna_jednotka, t.nakup_cena, t.sklad_ID
```

```
FROM dbo.Tovar AS t LEFT OUTER JOIN
```

```
dbo.Dodavatelia AS d ON t.dod_ico = d.dod_ico LEFT OUTER JOIN
```

```
dbo.Dod_udaje AS du ON d.dod_udaje_ID = du.dod_udaje_ID
```

```
CREATE VIEW Plan_prace
```

```
AS
```

```
SELECT CASE WHEN za.dat_ser IS NULL THEN za.dat_mon ELSE za.dat_ser END  
AS [Dátum montáže], za.zakazka_ID, zk.meno + ' ' + zk.priezvisko AS Zákazník,
```

```
za.ulica + ', ' + za.mesto AS Adresa, zu.tel_cislo AS [Telefónné číslo], t.nazov
```

```
AS Tovar
```

```
FROM dbo.Zakazka AS za LEFT JOIN
```

```

dbo.Tovar AS t ON t.tovar_ID = za.tovar_ID LEFT JOIN
dbo.Zakaznik AS zk ON zk.zakaznik_ID = za.zakaznik_ID LEFT JOIN
dbo.Zak_udaje AS zu ON zu.zak_udaje_ID = zk.zak_udaje_ID
WHERE (za.stav_zak_ID = 3)

CREATE VIEW Vyslany_zamestnanci
AS
SELECT    zc.meno + ' ' + zc.priezvisko AS Zamestnanec, za.zakazka_ID, zk.meno + ' ' +
zk.priezvisko AS Zákazník, t.nazov
FROM      dbo.Zamestnanec AS zc INNER JOIN
          dbo.Vyslany AS vy ON zc.zamestnanec_ID = vy.zamestnanec_ID INNER
JOIN
          dbo.Zakazka AS za ON vy.zakazka_ID = za.zakazka_ID INNER JOIN
          dbo.Zakaznik AS zk ON za.zakaznik_ID = zk.zakaznik_ID INNER JOIN
          dbo.Tovar AS t ON t.tovar_ID = za.tovar_ID

```

Príloha č. V: Procedúry

```
CREATE PROCEDURE Navrhni_servisy
    @VybranyMesiac DATE
AS
BEGIN
    DECLARE @AktualnyRok INT;
    DECLARE @MinulyRok INT;
    SET @AktualnyRok = YEAR(GETDATE());
    SET @MinulyRok = @AktualnyRok - 1;
    SELECT zakazka_ID, dat_mon
    INTO #MinuleZakazky
    FROM Zakazka
    WHERE YEAR(dat_mon) = @MinulyRok
    AND MONTH(dat_mon) = MONTH(@VybranyMesiac)
    OR (YEAR(dat_ser) = @MinulyRok
    AND MONTH(dat_ser) = MONTH(@VybranyMesiac));
    DECLARE @ZakazkaID INT, @DatMon DATE;
    DECLARE cur CURSOR FOR
    SELECT zakazka_ID, dat_mon
    FROM #MinuleZakazky;
    OPEN cur;
    FETCH NEXT FROM cur INTO @ZakazkaID, @DatMon;
    WHILE @@FETCH_STATUS = 0
    BEGIN
```

```

IF DATEDIFF(YEAR, @DatMon, GETDATE()) <= 5

BEGIN

    -- Navrhni ďalší servis s identickým dátumom pre súčasný rok

    INSERT INTO Zakazka (zakaznik_ID, tovar_ID, dat_mon, dat_ser, stav_zak_ID,
typ_zak, ulica, mesto, psc, suma_prace, pred_cena)

        SELECT zakaznik_ID, tovar_ID, @DatMon, DATEADD(YEAR, DATEDIFF(YEAR,
@DatMon, GETDATE()), @DatMon), 2, 'Servis', ulica, mesto, psc, NULL, NULL

    FROM Zakazka

    WHERE zakazka_ID = @ZakazkaID;

END;

    FETCH NEXT FROM cur INTO @ZakazkaID, @DatMon;

END;

CLOSE cur;

DEALLOCATE cur;

DROP TABLE #MinuleZakazky;

END;

```

```

CREATE PROCEDURE Pridaj_mnozstvo

    @TovarID varchar(15)

AS

BEGIN

    IF EXISTS (SELECT 1 FROM Tovar WHERE tovar_ID = @TovarID)

    BEGIN

        UPDATE Tovar

        SET mnozstvo = mnozstvo + 1

```

```

WHERE tovar_ID = @TovarID;

PRINT 'Množstvo pre tovar s ID ' + @TovarID + ' bolo úspešne zvýšené o 1.';

END

ELSE

BEGIN

    PRINT 'Tovar s ID ' + @TovarID + ' neexistuje.';

END

END;

CREATE PROCEDURE ZakMinRok

    @Mesiac INT

AS

BEGIN

    DECLARE @PreviousYearDate DATE

    DECLARE @SelectedMonthFirstDay DATE

    DECLARE @SelectedMonthLastDay DATE

    SET @PreviousYearDate = DATEADD(YEAR, -1, GETDATE())

    SET @SelectedMonthFirstDay = DATEFROMPARTS(YEAR(@PreviousYearDate),
    @Mesiac, 1)

    SET @SelectedMonthLastDay = EOMONTH(@SelectedMonthFirstDay)

    SELECT za.zakazka_ID, za.dat_mon, za.dat_ser, za.ulica + ', ' + za.mesto AS Adresa,
    t.nazov, zk.meno + ' ' + zk.priezvisko AS Zákazník, zu.tel_cislo

    FROM Zakazka za

    LEFT JOIN Tovar t ON t.tovar_ID = za.tovar_ID

    LEFT JOIN Zakaznik zk ON zk.zakaznik_ID = za.zakaznik_ID

```

```
LEFT JOIN Zak_udaje zu ON zu.zak_udaje_ID = zk.zak_udaje_ID

WHERE MONTH(CASE WHEN za.dat_ser IS NOT NULL THEN za.dat_ser ELSE
za.dat_mon END) = @Mesiac

END
```