



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH DATABÁZE PRO PŘEPRAVNÍ FIRMU

DATABASE DESIGN FOR TRANSPORT COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Zdeněk Záboj

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Kříž, Ph.D.

BRNO 2018

Zadání bakalářské práce

Ústav:	Ústav informatiky
Student:	Zdeněk Záboj
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Manažerská informatika
Vedoucí práce:	Ing. Jiří Kříž, Ph.D.
Akademický rok:	2017/18

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

Návrh databáze pro přepravní firmu

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 identifikace klíčových vlastností a nejnovějších trendů v oblasti používání a funkcionality databázového systému a jeho následný návrh pro potřeby přepravní firmy, který bude reflektovat na tyto vlastnosti. Návrh bude obsahovat komplexní funkcionalitu jak pro administrátora databáze tak i pro koncové uživatele.

Základní literární prameny:

CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.

KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015. ISBN 978-80-251-4-52-0.

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

STEPHENS, Ryan K., Ronald R. PLEW a Arie JONES. Naučte se SQL za 28 dní: stačí hodina denně.
Brno: Computer Press, 2010. ISBN 9788025127001.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2017/18

V Brně dne 28.2.2018

L. S.

doc. RNDr. Bedřich Půža, CSc.
ředitel

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Cílem této práce je navrhnout databázi pro začínající přepravní společnost, pro kterou vzhledem k její velikosti nemají význam řešení dostupná v současné době na trhu. V rámci této práce se budu zabývat především návrhem struktury této databáze a její implementací na platformě Microsoft SQL Server za pomoci nástrojů Microsoft Management Studia.

Klíčová slova

databáze, Microsoft SQL Server, Microsoft Management Studio, SQL, přepravní společnost

Abstract

The aim of this thesis is to design a database for an incipient transport company, for which the solutions available on the market nowadays don't make sense due to its size. Within this thesis I am going to concentrate on designing the structure of the database and its implementation on Microsoft SQL Server platform using Microsoft Management Studio tools.

Key words

database, Microsoft SQL server, Microsoft Management Studio, SQL, transport company

Bibliografická citace

ZÁBOJ, Z. *Návrh databáze pro přepravní firmu*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2018. 67 s. Vedoucí bakalářské práce Ing. Jirí Kříž, Ph.D..

Čestné prohlášení

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

V Brně dne 15. května 2018

podpis autora

Poděkování

Tímto bych chtěl poděkovat vedoucímu této bakalářské práce Ing. Jiřímu Křížovi, Ph.D. za konzultace a cenné rady, které mi poskytl během zpracování této bakalářské práce. Dále bych chtěl poděkovat rodině a přátelům za podporu poskytnutou během tvorby této práce.

OBSAH

ÚVOD	12
CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ	13
1 TEORETICKÁ VÝCHODISKA PRÁCE.....	14
1.1 1.1 Data, informace a znalosti.....	14
1.1.1 Data.....	14
1.1.2 Informace	14
1.1.3 Znalosti	15
1.2 Databázový systém.....	15
1.2.1 Databáze.....	15
1.2.2 Systém řízení databáze.....	16
1.2.3 Databázové aplikace	16
1.3 Datové modely	16
1.3.1 Lineární datový model	17
1.3.2 Hierarchický datový model.....	17
1.3.3 Síťový datový model.....	17
1.3.4 Relační datový model	18
1.3.5 Objektový datový model.....	18
1.4 Funkční modelování.....	18
1.4.1 Vývojový diagram	18
1.5 Relační datový model.....	19
1.5.1 Relace.....	20
1.5.2 Integrita relačního modelu	20
1.6 Normalizace	24
1.6.1 První normální forma.....	24
1.6.2 Druhá normální forma	24

1.6.3	Třetí normální forma.....	24
1.7	Návrh databáze.....	24
1.7.1	Konceptuální návrh databáze.....	24
1.7.2	Logický návrh databáze	25
1.7.3	Fyzický návrh databáze	25
1.8	1.7 Jazyk SQL.....	25
1.8.1	Příkazy jazyka DDL.....	26
1.8.2	Příkazy jazyka DQL.....	26
1.8.3	1.7.3 Příkazy jazyka DML.....	26
1.8.4	1.7.4 Příkazy jazyka DCL.....	26
1.8.5	1.7.5 Příkazy řízení transakcí.....	27
1.9	Datové typy	27
1.9.1	Celá čísla.....	27
1.9.2	Čísla s pohyblivou desetinou čárkou	27
1.9.3	Čas	28
1.9.4	Text	28
2	ANALÝZA SOUČASNÉHO STAVU.....	30
2.1	Představení projektu.....	30
2.2	Analýza procesů probíhajících v organizaci	30
3	VLASTNÍ NÁVRHY ŘEŠENÍ.....	33
3.1	Konceptuální návrh	33
3.1.1	Identifikace entit	33
3.2	Identifikace základních vztahů.....	34
3.3	Logický návrh	35
3.3.1	Tabulka 1: Pozice.....	35
3.3.2	Tabulka 2: Řidičské oprávnění	36

3.3.3	Tabulka 3: Typ dokladu	36
3.3.4	Tabulka 4: Typ servisní operace	36
3.3.5	Tabulka 5: Měrná jednotka	37
3.3.6	Tabulka 6: Poskytnutá služba	37
3.3.7	Tabulka 7: Oblast	38
3.3.8	Tabulka 8: Typ zákazníka	38
3.3.9	Tabulka 9: Typ vozidla	39
3.3.10	Tabulka 10: PSČ	39
3.3.11	Tabulka 11: Značka	39
3.3.12	Tabulka 12: Dodavatel servisních služeb	40
3.3.13	Tabulka 13: Pracovník	40
3.3.14	Tabulka 14: Řidič a řidičské oprávnění	41
3.3.15	Tabulka 15: Parametry	42
3.3.16	Tabulka 16: Vozidlo	42
3.3.17	Tabulka 17: Zákazník	43
3.3.18	Tabulka 18: Zakázka	44
3.3.19	Tabulka 19. Řádky zakázky	45
3.3.20	Tabulka 20: Plánovaný servis	45
3.3.21	Tabulka 21: Parametry vozidla	46
3.3.22	Tabulka 22: Servis informace	46
3.3.23	Tabulka 23: Doklady o vozidle	47
3.3.24	Tabulka 24: kniha jízd	48
3.3.25	ER diagram	49
3.4	Fyzický návrh	51
3.4.1	Vytvoření tabulek a přidání primárních a cizích klíčů	51
3.4.2	Naplnění tabulek testovacími daty	52

3.4.3	Pohledy	53
3.4.4	Triggery	57
	ZÁVĚR	63
	SEZNAM POUŽITÝCH ZDROJŮ	64
	SEZNAM POUŽITÝCH OBRÁZKŮ	65
	SEZNAM POUŽITÝCH TABULEK.....	66
	SEZNAM PŘÍLOH.....	67

ÚVOD

V současné době mají informace nesmírný význam pro jednotlivce i organizace, proto je důležité tyto informace zaznamenávat ve formě dat tak, aby dostupnost těchto informací byla, pokud možno co největší. K tomuto účelu je výhodné informace evidovat ve formě dat v elektronické podobě za využití databází.

Tato bakalářská práce se zabývá návrhem databáze pro začínající přepravní firmu. Účelem této databáze je poskytnout přepravní firmě nástroj, který umožní zaznamenávat data spojená s její činností.

CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

Cílem této bakalářské práce je identifikace klíčových vlastností a nejnovějších trendů v oblasti používání a funkcionality databázového systému a jeho následný návrh pro potřeby přepravní firmy, který bude respektovat výše popsané cíle. Návrh bude obsahovat komplexní funkcionalitu jak pro administrátora databáze, tak i pro koncové uživatele.

Díličními cíli práce jsou:

- zpracování teoretických východisek této práce,
- zpracování analýzy současného stavu,
- vlastní návrh řešení.

Pro určení požadavků na databázi jsem využil osobního rozhovoru s podnikatelem, během kterého jsme postupně určili jednotlivé požadavky na databázi. Z těchto požadavků jsem následně vycházel během vlastního návrhu databáze. Dále jsem vytvořil několik uživatelských pohledů a triggerů, které mají za úkol zajistit správné fungování databáze.

1 TEORETICKÁ VÝCHODISKA PRÁCE

Následující kapitola této práce je zaměřena na teoretická východiska a definici pojmů vztahujících se k pojmu databáze. Jedná se zejména o pojmy: databáze, systém řízení databáze, databázový systém, relační databáze a jazyk SQL.

1.1 1.1 Data, informace a znalosti

Pro pochopení problematiky spojené s datovým modelováním je třeba chápat následující pojmy (2, s. 4).

1.1.1 Data

„Surová (nezpracovaná) fakta, která mají určitou důležitost pro jednotlivce nebo organizaci.“ (1, s. 36)

1.1.2 Informace

„Data, která prošla zpracováním nebo dostala strukturu, která jim dává pro jednotlivce nebo organizaci význam.“ (1, s. 36)

Aby mohla být zpráva chápána jako informace, musí splňovat tři následující podmínky:

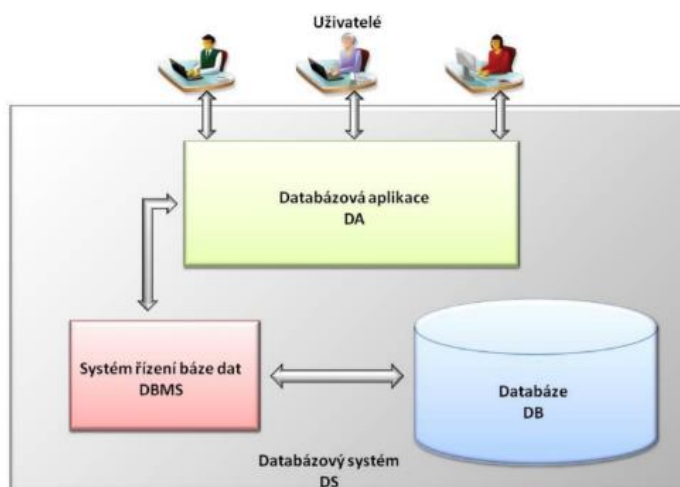
- sémantická relevance,
- syntaktická relevance,
- pragmatická relevance (2, s. 4).

1.1.3 Znalosti

Pojem znalost lze chápat jako porozumění informacím a jejich následné využití zkombinované s dříve přijatými informacemi (2, s. 4).

1.2 Databázový systém

Databázový systém je složen ze čtyř hlavních komponent, kterými jsou: databáze, databázová aplikace, systém řízení báze dat a uživatelé (3, s. 29).



Obrázek č. 1 Schéma databázového systému

(Zdroj: 7)

1.2.1 Databáze

Pojem databáze lze definovat jako soubor vzájemně souvisejících záznamů obsahující vlastní popis. U relačních databází lze předchozí definici upravit jako soubor souvisejících tabulek, které obsahují vlastní popis. To, že relační databáze obsahují vlastní popis, znamená, že popis struktury databáze je součástí databáze samotné, a proto vždy při průzkumu databáze postačí informace, které z ní získáme, a není třeba hledat informace jinde. Související tabulky jsou takové tabulky, které jsou společným sloupcem (3, s. 30).

1.2.2 Systém řízení databáze

Systém řízení databáze je softwarové řešení interakcí mezi uživateli, databází a databázovými aplikacemi. Jedná se například o Microsoft Access, Microsoft SQL Server, Oracle Database, DB2 (3, s. 31).

Systém řízení báze dat poskytuje služby potřebné k údržbě či uspořádání databáze, může jít například o následující služby:

- uložení, vyvolání a aktualizaci dat,
- přesun dat,
- zálohování a obnova databáze,
- řízení souběžného přístupu více uživatelů,
- řízení transakci,
- podpora dotazovacího jazyka,
- bezpečnostní opatření, která slouží k zabezpečení dat (4, s. 10).

1.2.3 Databázové aplikace

Lze říci, že se jedná o program, který vyvolává požadavky prostřednictvím jazyka SQL a pracuje s databází prostřednictvím systému řízení databáze (1, s. 39).

1.3 Datové modely

Každý datový objekt, kterým například může být člověk, představuje v datovém modelu určitou entitu. Pro takovou entitu musí být definovány atributy entity. Tím nám vzniká struktura objektu, která se nazývá věta (2, s. 11).

1.3.1 Lineární datový model

V lineárních datových modelech neexistují vazby mezi jednotlivými tabulkami. Lineární datový model je jediný datový model, který může být implementován na libovolném médiu (5, s. 21).

Vhodným příkladem lineárního datového modelu může být například lékařská evidence pacientů realizovaná pomocí kartoték. V tomto případě karta pacienta reprezentuje jednu větu databázového souboru. Mezi jednotlivými kartami existuje pouze vztah předchůdce a následovníka (5, s. 21).

1.3.2 Hierarchický datový model

Tento datový model je tvořen „rodičovským“ segmentem, jehož vazby vedou směrem na jemu podřízené segmenty (potomci). Tyto segmenty mají jinou strukturu a obsah než „rodičovský“ segment. Vazby na jednotlivé segmenty jsou realizovány pomocí pointerů, které realizují databázový systém. Na podřízené segmenty se nelze dostat jinak než přes rodičovský segment (5, s. 22).

Výhodou tohoto datového modelu je rychlost a přehlednost vyhledávání pomocí pointerů. Nevýhodou je časová náročnost na přerozdělení datových souborů během vkládání a složitá rekonstrukce dat v případě poškození databáze (5, s. 22).

1.3.3 Síťový datový model

Lze říci, že síťový datový model je obdobou hierarchického datového modelu. Nicméně u tohoto datového modelu nejsou segmenty rozdělovány na „rodiče“ a „potomky“ a jednotlivé pointery nevedou jenom z „rodičovských“ segmentů na „potomky“, ale obecně mezi segmenty v různých směrech. Další výhodou oproti hierarchickému modelu je možnost propojení libovolných segmentů, které umožňuje rychlý přístup k datům (5, s. 23).

1.3.4 Relační datový model

Tento datový model patří aktuálně k nejpoužívanějším datovým modelům. Tento datový model vzniká z několika lineárních modelů, které jsou navzájem propojeny pomocí relačních klíčů. Toto propojení nicméně není trvalé, vzniká v případě potřeby práce s daty, které jsou uloženy v jednotlivých tabulkách a zaniká v momentě ukončení práce s modelem (5, s. 24).

1.3.5 Objektový datový model

Objektový datový model je nejnovější z datových modelů. Tento datový model je založen na základním prvku, kterým je objekt. Lze říci, že odpovídá přibližně pojmu věta, nicméně s tím rozdílem že má kromě atributů definovány i metody, které určují jeho chování (5, s. 24).

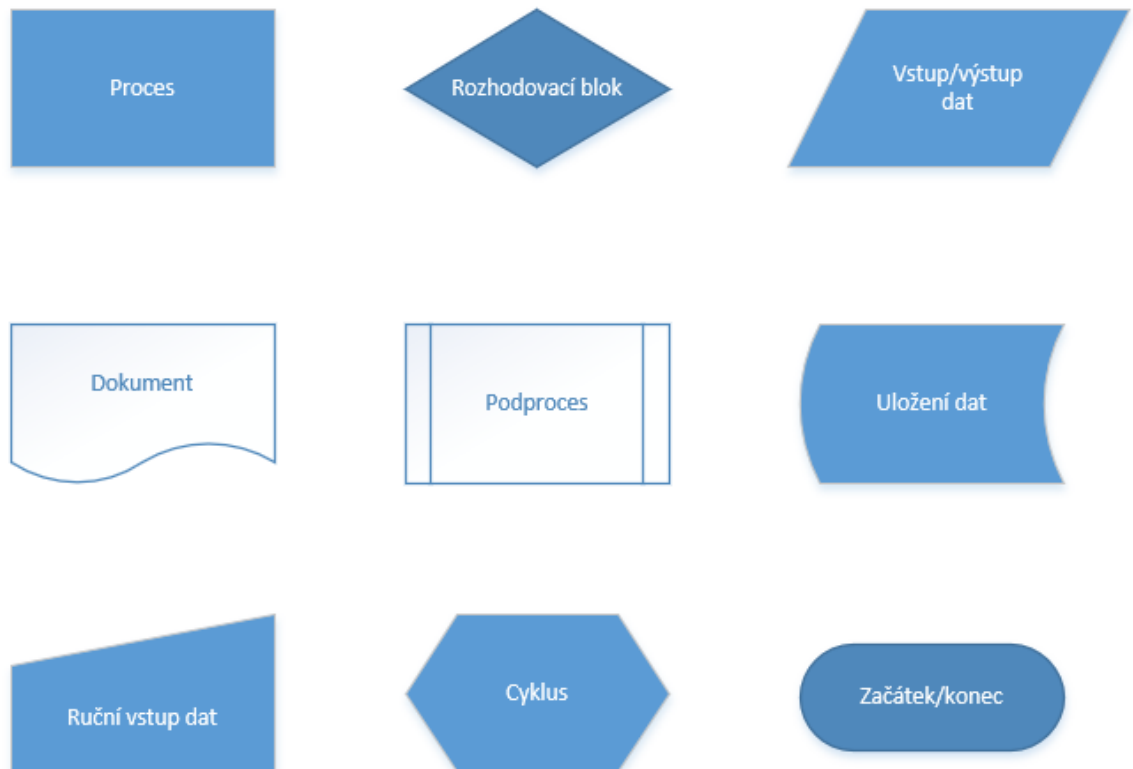
Pokud takovým objektem bude „zkouška“ studenta, tak jeho atributem budou například údaje, jako jsou číslo studenta, datum, předmět, známka, aj. Tento objekt může mít definované své metody, například metodu „vytvoř záznam o zkoušce“, která překontroluje, zda student splnil podmínky k absolvování zkoušky, případně zda již nemá tuto zkoušku hotovou (5, s. 25).

1.4 Funkční modelování

„Zabývá-li se datová modelování problematikou dat, která potřebujeme pro informační systém, funkční modelování se zabývá zkoumáním a algoritmizací činností, procesů, které v informačním systému probíhají.“ (2, s. 76)

1.4.1 Vývojový diagram

Vývojový diagram slouží k zachycení větvení procesů při rozhodování a plnění respektive neplnění podmínek. Následující obrázek obsahuje základní značky používané u vývojových diagramů (2, s. 90).



Obrázek č. 2 Základní značky vývojového diagramu

(Vlastní zpracování dle: 2, s. 90)

1.5 Relační datový model

Jak jsem již zmínil v subkapitole 1.3.4, relační datový model patří v současné době k nejpoužívanějším datovým modelům. Tento datový model vychází z teorie relací, lze říci, že tento datový model nám umožňuje nejenom zachycení samotných dat ale také jejich vzájemné vztahy (5, s. 23).

Milníkem, který lze chápat jako zrod relačního datového modelu je práce „A relations model of data for large shared data banks“ jejímž autorem je britský matematik a informatik Edgar Frank Codd (1, s. 62).

Cíle relačního modelu byly stanoveny následovně: vysoký stupeň nezávislosti dat, umožnění expanze množinově orientovaných dat pro práci s daty, ustavení základů pro zvládnutí problematiky týkající se sémantiky dat, konzistence dat a redundance dat (1, s. 62).

Tento datový model obsahuje pět základních složek: relace, atribut, doména, doménová n-tice a relační databáze (1, s. 63).

1.5.1 Relace

Relaci můžeme definovat jako tabulku, které obsahuje řádky a sloupce (3, s. 78). Relace se skládá ze schématu relace a samotného těla relace (2, s. 26).

Mezi hlavní vlastnosti relace patří:

- každá n-tice obsahuje data která se týkají nějaké entity,
- každý sloupec obsahuje data spojená s atributem entity,
- každý sloupec musí mít jedinečný název,
- na pořadí sloupců ani řádků nezáleží,
- každý řádek musí být unikátní, řádky se tudíž nesmí opakovat (3, s. 78).

1.5.2 Integrita relačního modelu

Integritu relačního modelu lze rozdělit na integritní omezení samotných entit a integritní omezení jejich vzájemných vztahů (1, s. 71).

Integritní omezení pro entity se dělí na: doménovou integritu, entitní integritu a referenční integritu (2, s. 27).

1.5.2.1 Doménová integrita

Obsahuje definování množiny hodnot pro atribut a jejich následnou specifikaci, jedná se zejména o stanovení datového typu a případně jeho velikost, zda je nutné, aby byla položka zadána, zda bude nějaká hodnota automaticky nastavena (5, s. 29).

1.5.2.2 Entitní integrita

Je zajišťována pomocí primárních klíčů. Primární klíč může být reprezentován jedním atributem, případně může být složen z více atributů a musí být jednoznačný a minimální. Jednoznačnost značí, že pro každou větu tabulky existuje unikátní primární klíč, kterým lze tuto větu identifikovat. Minimální primární klíč je takový, u kterého nelze vypustit jakýkoliv atribut, aniž by nebylo porušeno pravidlo jednoznačnosti (5, s. 29).

Kandidátní klíč splňuje vlastnosti primárního klíče. V relaci může být takovýchto kandidátních klíčů více a jeden z nich je vybrán jako primární (3, s. 82).

1.5.2.3 Referenční integrita

Je zajišťována cizími klíči. Za pomoci cizích a primárních klíčů vzniká spojení jednotlivých relací. Platí zde podmínka, že cizí klíč a jemu odpovídající primární klíč musí být definovány na stejné doméně (2, s. 28).

Integritní omezení pro vztahy vztahů relací definuje kardinalitu jejich vztahu na: 1:1, N:M, 1:N respektive N:1 (5, s. 31).

Vztahem 1:1 rozumíme fakt, že každé větě jedné relace odpovídá jedna věta jiné relace. Tento vztah si lze například představit na příkladu člověk a datum narození (5, s. 31).



Obrázek č. 3 Vztah 1:1

(Vlastní zpracování dle: 5, s. 32)

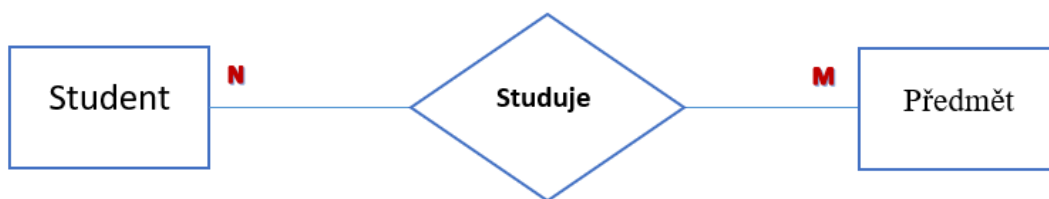
Vztahem 1:N respektive N:1 rozumíme fakt, že jedné větě relace odpovídá jedna nebo více vět jiné relace. Tento vztah si lze představit na příkladu studenta, který skládá jednotlivé zkoušky. V případě vztahu N:1 se například může jednat o studenty skládající konkrétní zkoušku. Lze tedy říct, že vztah N:1 je to stejné jako vztah 1:N z druhé strany (5, s. 33).



Obrázek č. 4 Vztah 1:N

(Vlastní zpracování dle: 5, s. 33)

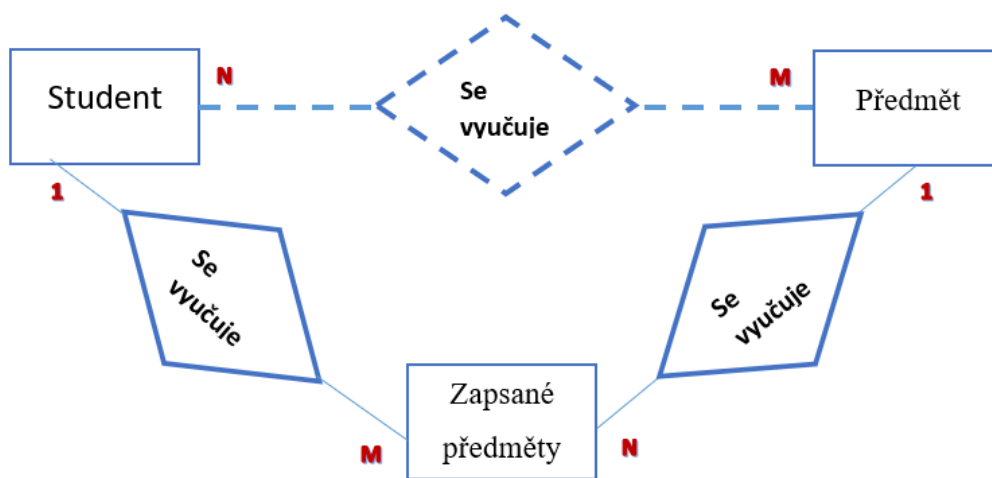
Vztahem N:M rozumíme fakt, že několika větám jedné relace odpovídá několik vět jiné relace. Tento vztah si lze například představit na příkladu studentů studujících předměty, kdy jeden student studuje více předmětů a zároveň je jeden předmět studován více studenty (5, s. 34).



Obrázek č. 5 Vztah N:M

(Vlastní zpracování dle: 5, s. 34)

Vztahy N:M se řeší pomocí dekompozice původního vztahu N:M na vztah 1:N a 1:M za pomoci průnikové tabulky. Dekompozici je třeba provést z toho důvodu že samotný vztah N:M neumožňuje vytvořit vazbu mezi tabulkami (5, s. 34).



Obrázek č. 6 Dekompozice vztahu N:M

(Vlastní zpracování dle: 5, s. 35)

1.6 Normalizace

Procesem normalizace rozumíme úpravu datových struktur dle jednotlivých normalizačních stupňů. Normalizaci lze odůvodnit potřebou minimalizování redundance a efektivního ukládání dat. Rozlišujeme několik normalizačních stupňů. Nejvíce jsou využívány první tři stupně, tyto stupně jsou popsány níže (2, s. 54).

1.6.1 První normální forma

Aby byla splněna první normální forma, je třeba, aby všechny atributy byly atomické (2, s. 55).

1.6.2 Druhá normální forma

Pro splnění podmínek druhé normální formy je třeba, aby všechny atributy byly závislé na primárním klíči a zároveň byly splněny podmínky pro první normální formu (2, s. 56).

1.6.3 Třetí normální forma

Pro splnění podmínek třetí normální formy je třeba, aby všechny atributy, které nejsou klíči, byly vzájemně nezávislé. Zároveň musí být splněny podmínky spojené s druhou normální formou (2, s. 58).

1.7 Návrh databáze

Návrh databáze se skládá z konceptuálního, logického a fyzického návrhu databáze tyto jednotlivé fáze jsou stručně popsány v následujících podkapitolách (1, s. 206).

1.7.1 Konceptuální návrh databáze

V rámci této fáze návrhu databáze je vytvořen model dat, které využívá organizace bez úvah o jeho následné implementaci (1, s. 206).

1.7.2 Logický návrh databáze

Tato část návrhu vychází z konceptuálního návrhu databáze ovšem v této části je již uvažován konkrétní návrh databáze, který je ovšem nezávislý na volbě systému řízení dat (1, s. 206).

1.7.3 Fyzický návrh databáze

V rámci této fáze návrhu databáze je zvolen systém řízení báze dat, jsou vytvořeny jednotlivé tabulky, návrh uživatelských pohledů, návrh bezpečnostních mechanismů a podobně (1, s. 208).

1.8 1.7 Jazyk SQL

Jazyk SQL neboli Structured Query Language je neprocedurální jazyk, který se využívá ke komunikaci s relačními databázemi. Jedná se o nejrozšířenější jazyk využívaný pro tvorbu databázových dotazů. Počátky tohoto jazyka sahají do 70. let minulého století, kdy výzkumníci ze společnosti IBM vyvinuli experimentální databázi Systém/R. Součástí této databáze byl jazyk SEQEL později se zjistilo, že SEQEL má jako ochrannou známku zaregistrovanou jiná společnost, a tak byl jazyk SEQEL přejmenován na SQL (4, s. 33-37; 6, s. 31).

Základní syntaktické zvyklosti spojené s jazykem SQL:

- příkazy začínají klíčovým slovem,
- příkazy jsou navzájem oddělené oddělovačem,
- názvy jednotlivých objektů mohou obsahovat písmena, číslice, případně znak podtržítka na oddělení jednotlivých slov v názvu,
- znakové řetězce musí být ohraničeny apostrofy,

- implementace jazyka SQL mají vyhrazenou sadu slov, která mají určitý význam, a proto je nejde použít pro názvy jednotlivých objektů a podobně,
- k označení jednořádkového komentáře se používají dvě bezprostředně za sebou napsané pomlčky. Pro začátek víceřádkového komentáře se používá kombinace lomítka a hvězdičky a pro jeho konec kombinace hvězdičky a lomítka (4, s. 39-40).

Příkazy jazyka SQL můžeme vzhledem k jejich funkcím rozdělit do pěti kategorií. Jedná se o příkazy jazyka DDL, příkazy jazyka DQL, příkazy jazyka DML, příkazy jazyka DCL a příkazy k řízení transakcí (4, s. 40).

1.8.1 Příkazy jazyka DDL

Příkazy tohoto jazyka umožňují uživatelům vytvářet databázové objekty, dále pak tyto objekty upravovat případně tyto objekty odstraňovat. Tyto příkazy mají klíčová slova CREATE, ALTER, DROP (4, s. 40).

1.8.2 Příkazy jazyka DQL

Příkazy tohoto jazyka umožňují načítání dat z databáze. Klíčovým slovem těchto příkazů je SELECT (4, s. 40).

1.8.3 1.7.3 Příkazy jazyka DML

Tento jazyk obsahuje příkazy pro vkládání dat do databáze, úpravou takto vložených dat případně jejich následné mazání. Klíčovými slovy těchto příkazů jsou INSERT, UPDATE a DELETE (4, s. 41).

1.8.4 1.7.4 Příkazy jazyka DCL

Jazyk DCL zahrnuje příkazy spojené s řízením přístupu k datům uloženým v databázi, klíčovými slovy jsou ALTER a GRANT (4, s. 41).

1.8.5 1.7.5 Příkazy řízení transakcí

Transakce je sada příkazů, která se buď provede jako celek, případně se neprovede vůbec. Transakce musí být atomická, což znamená, že je transakce buď úspěšná, nebo neúspěšná, konzistentní což znamená, že transakce převede všechny data, kterých se týká z jednoho stavu do dalšího. Dále musí transakce plnit svou funkci bez ohledu na ostatní existující transakce, změny provedené transakcí musí být zachovány i v případě chyby databáze či při jejím vypnutí (4, s. 161).

1.9 Datové typy

Následující podkapitoly obsahují tabulky se základními informacemi o základních datových typech (2, s. 14).

1.9.1 Celá čísla

Následující tabulka obsahuje informace o datových typech pro celá čísla (2, s. 14).

Tabulka č. 1 Datové typy pro celá čísla

(Vlastní zpracování dle: 2, s. 14)

Název	Rozsah od	Rozsah do
TINYINT	-128	127
SMALLINT	-32 768	32 767
MEDIUMINT	-8 388 608	8 388 607
INT	-2 147 483 648	2 147 483 647
BIGINT	-9 233 372 036 854 775 808	9 233 372 036 854 775 807

1.9.2 Čísla s pohyblivou desetinou čárkou

Následující kapitola obsahuje základní informace o datových typech pro čísla s pohyblivou desetinou čárkou (2, s. 14).

Tabulka č. 2 Datové typy pro čísla s pohyblivou desetinou čárkou

(Vlastní zpracování dle: 2, s. 14)

Název	Rozsah od	Rozsah do
-------	-----------	-----------

FLOAT	-3402823466E+38	3402823466E+38
DOUBLE	-17975931348623157E+308	17976931348623157E+308
DECIMAL	-17975931348623157E+308	17976931348623157E+308

U datového typu DECIMAL, který se zapisuje jako DECIMAL(m,d), pomocí m a d nastavujeme počet číslic před a za desetinou čárkou (2, s. 14).

1.9.3 Čas

Následující tabulka obsahuje základní informace o datových typech pro ukládání časových údajů (2, s. 15).

Tabulka č. 3 Datové typy spojené s časem

(Vlastní zpracování dle: 5, s. 15)

Název	Formát zápisu	Rozsah od	Rozsah do
DATE	RRRR-MM-DD	1000-01-01	9999-12-31
DATETIME	RRRR-MM-DD HH:MM:SS	1000-01-01 00:00:00	9999-12-31 23:59:59
TIME	HHH:MM:SS	-838:59:59	838:59:59
TIMESTAMP	RRRR-MM-DD HH:MM:SS	1970-01-01 00:00:00	2037-01-01 00:00:00

1.9.4 Text

Následující tabulka obsahuje základní informace o datových typech sloužících k ukládání textu (2, s. 15).

Tabulka č. 4 Datové typy pro text

(Vlastní zpracování dle: 2, s. 15)

Název	Rozsah od	Rozsah do
CHAR	0	255
VARCHAR	0	255
TINYTEXT	0	255
TEXT	0	65 535
MEDIUMTEXT	0	16 777 215
LONGTEXT	0	4 294 967 295

Zápis datového typu CHAR ve tvaru CHAR(m) slouží k nastavení délky řetězce, v případě že je uložen řetězec kratší jsou chybějící znaky doplněny mezerami (2, s. 15).

Zápis datového typu VARCHAR ve tvaru VARCHAR(m) slouží k nastavení délky řetězce, v případě že je uložen řetězec kratší nejsou chybějící znaky doplněny (2, s. 15).

2 ANALÝZA SOUČASNÉHO STAVU

V následující kapitole představím projekt přepravní společnosti, pro který navrhují databázi a následně provedu analýzu základních požadavků a činností ve firmě.

2.1 Představení projektu

Základní idea tohoto projektu byla formována na základě nevyužití dodávky značky Iveco, která byla původně zakoupena pro osobní potřebu. Náplní tohoto projektu má být poskytnutí služeb spojených s přepravou a tím zajištění vedlejšího příjmu k hlavní činnosti podnikatele. Vzhledem k různorodosti požadavků zákazníků je v současné době uvažováno o nákupu nákladního vozidla značky Avia.

2.2 Analýza procesů probíhajících v organizaci

V rámci běžné činnosti by docházelo k rutinním činnostem. V rámci těchto rutinních činností bude docházet k pravidelnému vytváření objednávek, ve kterých bude evidován požadavek zákazníka na provedení dopravní transakce.

Společnost bude poskytovat čtyři základní služby: přepravu samotnou, čekání v rámci přepravy, nakládání nákladu a vykládání nákladu. Tyto služby bude možné v objednávkách možné dle potřeby kombinovat. Je však nutné počítat s tím, že v momentě vzniku objednávky ještě není známo množství služeb, které bude zákazníkovi účtováno. Může být provedena kalkulace na základě předběžných orientačních množství, ale například pro dobu čekání není možné předem se 100% přesností určit dobu, kterou bude nutné zaevidovat, to samé může platit i například pro nakládku a vykládku, které mohou být doobjednány na základě dohody přímo při realizaci zakázky.

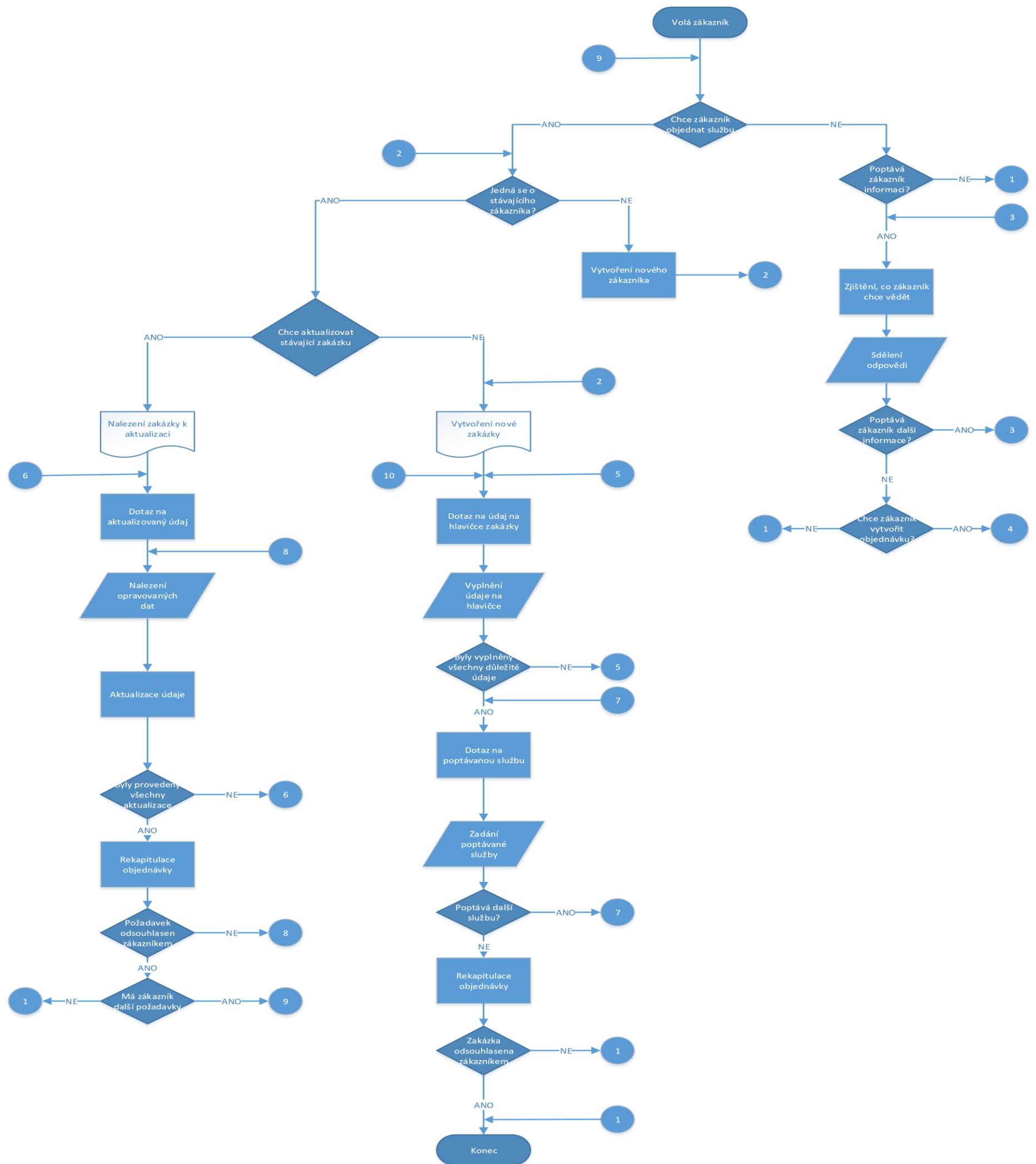
Standardně tedy zákazník společnost telefonicky či e-mailem kontaktuje s požadavkem, že chce dopravit určité zboží z jednoho místa na druhé. Bude tedy nutné předem specifikovat, odkud kam bude doprava probíhat (což je nutné pro alespoň přibližné

stanovení jak časové tak finanční náročnosti, která je stěžejní pro determinování počtu zakázek, která je společnost schopna obsloužit) a datum, kdy má dojít k odbavení zakázky. K zakázce bude rovněž dle volných kapacit vozidel a požadavků zákazníka přiřazeno konkrétní vozidlo, v jehož rozvrhu bude pro danou zakázku vyhrazeno dostatečné časové okno (toto přiřazení vzhledem k počtu vozidel bude prozatím probíhat ručně). Dále bude k zakázce přiřazen pracovník, který bude mít vyřízení této zakázky na starost. Je též třeba evidovat předběžný rozsah služeb, které mají být zákazníkovi poskytnuty (u těchto služeb bude poté ještě dle potřeby upraveno množství). Dále je třeba evidovat, zda byla zakázka vyřízena, případně zaevidována v účetnictví.

Vzhledem k velikosti společnosti má tato společnost nyní ceny nastaveny paušálně – pro všechny zákazníky je cena jednotlivých úkonů shodná, což společnosti zjednodušuje účtování zakázek.

Jelikož se jedná o přepravní společnost, je také velmi důležitá pravidelná údržba jimi vlastněných vozidel. Proto kromě samotných zakázek bude v návrhu nutné myslet i na to, že společnost bude potřebovat evidovat informace o provedených servisních úkonech, bude je potřebovat plánovat a bude chtít v elektronické podobě mimo jiné uchovávat i doklady o vozidle.

Procesy spojené s účetními transakcemi jako jsou například evidence přijatých, respektive vydaných faktur, mzdové účetnictví a podobně jsou řešeny pomocí softwaru Money S3.



Obrázek č. 7: Vývojový diagram kontaktování zákazníkem

(Zdroj: vlastní zpracování)

3 VLASTNÍ NÁVRHY ŘEŠENÍ

V následující kapitole se budu věnovat řešení požadavků, které vyplynuly z předchozí kapitoly. Tato kapitola je rozdělena na konceptuální, logický a fyzický návrh. Tato kapitola obsahuje návrh databáze, která by měla poskytnout přepravní firmě vhodný nástroj pro organizaci a správu dat spojených s jejím fungováním dle požadavků vyplívajících z předchozí kapitoly.

3.1 Konceptuální návrh

V následující části práce se budu zabývat objekty reálného světa, se kterými je mnou navrhovaná databáze spojena, na jejich základě si definuji jednotlivé entity a jejich vzájemné vztahy.

3.1.1 Identifikace entit

Následující tabulka zobrazuje hlavní entity spojené s návrhem této databáze, jejich popisy a množství jejich výskytů v databázi. Tyto informace budou následně sloužit k určení jednotlivých tabulek a jejich atributů.

Tabulka č. 5: Identifikace entit

(Zdroj: vlastní zpracování)

Název entity	Popis entity	Alias entity	Počet výskytů
Pozice	Pozice pracovníka v rámci poskytování přepravních služeb		Jednotky
Řidičské oprávnění	Evidence řidičských oprávnění pracovníků		Jednotky
Pracovník	Evidence informací o jednotlivých pracovnících		Jednotky
PSC	Evidence jednotlivých poštovních směrovacích čísel a měst		Tisíce
Oblast	Evidence oblastí	Okres	Desítky
Doklady o vozidle	Evidence dokladů, které náležejí jednotlivým vozidlům		Desítky
Typ dokladu	Evidence jednotlivých typů dokladů		Jednotky

Vozidlo	Evidence informací spojených s jednotlivými vozidly	Auto, nákladní vozidlo	Jednotky
Typ vozidla	Evidence jednotlivých typů vozidel		Desítky
Značka	Evidence jednotlivých značek vozidel		Desítky
Parametry vozidla	Evidence parametrů jednotlivých vozidel		Desítky
Servis Informace	Evidence proběhlých oprav vozidel		Stovky
Typ servisní operace	Evidence jednotlivých typů servisních operací		Desítky
Dodavatel servisních služeb	Evidence dodavatelů servisních služeb		Desítky
Plánovaný servis	Evidence plánovaných oprav vozidel		Stovky
Kniha jízd	Evidence jízd jednotlivých vozidel		Tisíce
Typ jízdních operací	Evidence typů jízdních operací pro knihu jízd		Jednotky
Zákazník	Evidence potřebných informací spojených se zákazníkem		Stovky
Typ zákazníka	Evidence jednotlivých typů zákazníků		Jednotky
Zakázka	Evidence informací spojených se zakázkou		Tisíce
Poskytnutá služba	Evidence poskytovaných služeb		Desítky

3.2 Identifikace základních vztahů

V předchozí kapitole jsem si identifikoval jednotlivé entity, které jsou významné pro navrhovaný databázi. V této podkapitole se budu zabývat základními vazbami mezi těmito entitami.

Tabulka č. 6: Identifikace základních relací

(Zdroj: vlastní zpracování)

Název entity	Relace	Název entity;
Pracovník	Má	Pozici
Pracovník	Má	Řidičské oprávnění
Pracovník	Má	PSC
Pracovník	Zodpovídá za	Vozidlo
Pracovník	Zodpovídá za	Plánovaný servis
Vozidlo	Má	Značku

Vozidlo	Má	Doklady
Vozidlo	Má	Zodpovědnou osobu
Vozidlo	Má	Typ vozidla
Vozidlo	Má	Servisní informace
Vozidlo	Má	Plánovaný servis
Vozidlo	Má	Knihu jízd
Zakázka	Má přiděleného	Pracovníka
Zakázka	Má přiděleného	Pracovníka
Zakázka	Má přidělené	Vozidlo
Zakázka	Se týká	Služby
Zakázka	Se týká	Zákazníka
Zákazník	Má	Typ zákazníka

3.3 Logický návrh

V rámci logického návrhu si definuji jednotlivé tabulky, které bude obsahovat mnou navržená databáze. U každé tabulky definuji její atributy, určím jejich datové typy a primární a cizí klíče.

3.3.1 Tabulka 1: Pozice

Tato tabulka slouží k evidenci jednotlivých možných pozic pracovníků, slouží k jednoznačnému určení pozice, která náleží pracovníkovi.

Tabulka č. 7: Atributy tabulky Pozice

(Zdroj: vlastní zpracování)

Název atributu	Primární/ cizí klíč	Typ a délka	Not Null	Popis
Pozice_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 30	Ano	Název pozice pracovníka
Popis		Varchar 250	Ne	Popis pracovní náplně pracovníka

3.3.2 Tabulka 2: Řidičské oprávnění

Tato tabulka slouží k evidenci jednotlivých skupin řidičských oprávnění, kterých může pracovník nabývat.

Tabulka č. 8: Atributy tabulky Řidičské oprávnění

(Zdroj: vlastní zpracování)

Název atributu	Primární/ cizí klíč	Typ a délka	Not Null	Popis
Opravneni_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 3	Ano	Název skupiny řidičského oprávnění
Popis		Varchar 250	Ne	Zde je možné evidovat vozidla případně jízdní soupravy, které do této skupiny patří

3.3.3 Tabulka 3: Typ dokladu

Tato tabulka slouží jako číselník k evidování jednotlivých možných typů dokladů, které může držet vlastník vozidla. V tomto případě se jedná zejména o typy dokladu servisní kniha, osvědčení o registraci vozidla nebo například osvědčení o pojištění vozidla.

Tabulka č. 9: Atributy tabulky Typ dokladu

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Typ_dokladu_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název dokladu
Poznamka		Varchar 250	Ne	Doplňující informace k typu dokladu

3.3.4 Tabulka 4: Typ servisní operace

Tato tabulka slouží jako číselník k evidování jednotlivých možných typů servisních operací, které mohou být prováděny. Mezi servisní operace, které mohou být s vozidly prováděny, patří například pravidelná technická prohlídka, která je prováděna jedenkrát

za 2 až 4 roky dle stáří vozidla, nebo například výměna oleje, která se provádí každý rok, případně i častěji, pokud je s vozidlem najeto velké množství kilometrů.

Tabulka č. 10: Atributy tabulky Typ servisní operace

(Zdroj: vlastní zpracování)

Název atributu	Primární/ cizí klíč	Typ a délka	Not null	Popis
Typ_servisni_ operace_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název dané servisní operace
Poznamka		Varchar 250	Ne	Doplňující informace k typu servisní operace

3.3.5 Tabulka 5: Měrná jednotka

Tato tabulka slouží jako číselník měrných jednotek pro tabulky parametry vozidla a poskytnutá služba.

Tabulka č. 11: Atributy tabulky Měrná jednotka

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Merna_ jednotka_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název dané měrné jednotky
Poznamka		Varchar 250	Ne	Doplňující informace k měrné jednotce

3.3.6 Tabulka 6: Poskytnutá služba

V tabulce poskytnutá služba je evidován název služby, její měrná jednotka a cena za tuto měrnou jednotku služby. Z této tabulky čerpá informace tabulka řádky zakázky.

Tabulka č. 12: Atributy tabulky Poskytnutá služba

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Poskytnuta_ sluzba_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název poskytované služby
Merna_ jednotka	Cizí klíč	Int	Ano	Propojení s tabulkou měrná jednotka
Cena		Decimal (12,2)	Ano	Cena poskytnuté služby
Poznamka		Varchar 250	Ne	Doplňující informace k poskytnuté službě

3.3.7 Tabulka 7: Oblast

Tabulka oblast je číselníkem oblastí pro tabulku PSČ. Vycházíme z předpokladu, že určité skupiny poštovních směrovacích čísel, respektive měst náleží do určité oblasti.

Tabulka č. 13: Atributy tabulky Oblast

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Oblast_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název dané oblasti
Poznamka		Varchar 250	Ne	Doplňující informace k oblasti

3.3.8 Tabulka 8: Typ zákazníka

Tato tabulka je číselníkem typů zákazníka pro tabulku zákazník.

Tabulka č. 14: Atributy tabulky Typ zákazníka

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Typ_ zakaznika_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název typu zákazníka

3.3.9 Tabulka 9: Typ vozidla

Tato tabulka je číselníkem typů vozidel pro tabulku vozidlo.

Tabulka č. 15: Atributy tabulky Typ vozidla

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Typ_vozidla_ID	Primární klíč	Int	Ano	Automaticky generované číslo sloužící k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název typu vozidla

3.3.10 Tabulka 10: PSČ

V této tabulce jsou evidována poštovní směrovací čísla měst a oblastí do kterých tyto poštovní směrovací čísla, respektive města náleží.

Tabulka č. 16: Atributy tabulky PSČ

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Psc_ID	Primární klíč	Int	Ano	Automaticky generované číslo, sloužící k jednoznačné identifikaci záznamu
Psc		Varchar 6	Ano	Poštovní směrovací číslo
Mesto		Varchar 50	Ano	Město
Oblast	Cizí klíč	Int	Ano	Propojení s tabulkou oblast

3.3.11 Tabulka 11: Značka

V této tabulce jsou evidovány jednotlivé značky všech používaných vozidel, v rámci této tabulky je také evidován výrobce vozidla.

Tabulka č. 17: Atributy tabulky Značka

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Znacka_ID	Primární klíč	Int	Ano	Automaticky generované číslo, sloužící k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název značky vozidla
Vyrobce		Varchar 50	Ano	Výrobce značky vozidla

3.3.12 Tabulka 12: Dodavatel servisních služeb

Tato tabulka slouží k evidenci partnerů, u kterých společnost provádí jednotlivé servisní úkony, které se týkají jednotlivých vozidel. Jedná se o seznam vedením autorizovaných pracovišť.

Tabulka č. 18: Atributy tabulky Dodavatel

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Dodavatel_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Nazev		Varchar 50	Ano	Název dodavatele servisních služeb
K_tel1		Varchar 20	Ano	Primární telefonický kontakt na dodavatele servisních služeb
K_tel2		Varchar 20	Ne	Sekundární telefonický kontakt na dodavatele servisních služeb
K_mail		Varchar 50	Ne	Emailový kontakt na dodavatele servisních služeb
K_osoba		Varchar 100	Ano	Jméno kontaktní osoby dodavatele servisních služeb
K_ulice		Varchar 50	Ano	Název ulice
K_cp		Varchar 5	Ano	Číslo popisné
K_ce		Varchar 5	Ne	Číslo evidenční
K_psc	Cizí klíč	Int	Ano	Propojení s tabulkou psc

3.3.13 Tabulka 13: Pracovník

Účelem této tabulky je evidovat informace spojené s pracovníkem, které jsou potřebné v souvislosti s výkonem práce. V této tabulce nejsou evidovány podrobné osobní údaje, neboť se nejedná o účetní či mzdový systém, tudíž jsou obsahem této tabulky pouze nezbytně nutné informace.

Tabulka č. 19: Atributy tabulky Pracovník

(Zdroj: vlastní zpracování)

Název atributu	Primární/ cizí klíč	Typ a délka	Not null	Popis
Pracovník_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Titul_pred		Varchar 20	Ne	Tituly dosažené během vzdělání, které se píší před jméno
Jmeno		Varchar 30	Ano	Křestní jméno
Rodne_prijmeni		Varchar 30	Ne	Rodné příjmení
Prijmeni		Varchar 60	Ano	Příjmení
Titul_za		Vachar 20	Ne	Tituly dosažené během vzdělání, které se píší za jméno
Pozice	Cizí klíč	Int	Ano	Propojení s tabulkou pozice
Datum_narozeni		Date	Ano	Datum narození
Bezuhonnost		Bit	Ano	Evidence bezúhonnosti navázaná na externí systém
Bezuhonost_dok ad		Varchar 500	Ano	Odkaz na oskenovaný doklad
K_ulice		Varchar 50	Ano	Kontaktní ulice
K_cp		Varchar 5	Ano	Kontaktní číslo popisné domu
K_ce		Varchar 5	Ne	Kontaktní číslo evidenční domu
K_psc	Cizí klíč	Int	Ano	Propojení s tabulkou psč
K_tel1		Varchar 20	Ano	Primární telefonický kontakt
K_tel2		Varchar 20	Ne	Sekundární telefonický kontakt
K_mail		Varchar 50	Ne	Kontaktní emailová adresa

3.3.14 Tabulka 14: Řidič a řidičské oprávnění

Tato tabulka složí k propojení pracovníků s řidičskými oprávněními. Předpokladem je že jeden pracovník může být současně držitelem více skupin řidičského oprávnění.

Tabulka č. 20: Atributy tabulky Řidič a řidičské oprávnění

(Zdroj: vlastní zpracování)

Název atributu	Primární/ cizí klíč	Typ a délka	Not null	Popis
Cislo_dokladu	Primární klíč	Varchar 10	Ano	Číslo řidičského průkazu
Pracovník	Cizí klíč	Int	Ano	Propojení s tabulkou pracovník
Opravneni	Primární/cizí klíč	Int	Ano	Propojení s tabulkou řidičské oprávnění
Platnost		Date	Ano	Slouží k evidování platnosti

3.3.15 Tabulka 15: Parametry

V tabulce parametry jsou evidovány jednotlivé parametry, které mohou být u vozidel evidovány. Je zde evidován název parametru a jeho měrná jednotka, s tím, že měrná jednotka je provázána na tabulku měrná jednotka.

Tabulka č. 21: Atributy tabulky Parametry

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Parametr_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Merna_jednotka	Cizí klíč	Int	Ano	Propojení s tabulkou měrná jednotka
Nazev		Varchar 50	Ano	Název parametru

3.3.16 Tabulka 16: Vozidlo

Tato tabulka slouží k evidování informací potřebných k identifikaci konkrétního vozu a osoby, která za toto vozidlo zodpovídá. Konkrétní parametry jednotlivých vozidel jsou evidovány v tabulce parametry vozidla, protože pro jednotlivá vozidla se mohou požadavky na evidované parametry lišit.

Tabulka č. 22: Atributy tabulky Vozidlo

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Vozidlo_ID (VIN)	Primární klíč	Varchar 17	Ano	Kód vin u této tabulky slouží k jednoznačné identifikaci záznamu
SPZ		Varchar 7	Ano	Registrační značka vozidla
Znacka	Cizí klíč	Int	Ano	Propojení s tabulkou značka
Typ_vozidla	Cizí klíč	Int	Ano	Propojení s tabulkou typ vozidla
Zodpovedna_ osoba	Cizí klíč	Int	Ano	Propojení s tabulkou pracovník

3.3.17 Tabulka 17: Zákazník

Tato tabulka slouží k evidování informací o zákaznících. Z těch nejdůležitějších informací, které o zákazníkovi společnost potřebuje evidovat, můžeme vybrat například: fakturační adresu, kontaktní údaje, a především jeho typ a název. Zákazníci této firmy jsou zejména fyzické osoby poptávající jednorázovou dopravu případně služby spojené s touto dopravou, zejména se jedná o nakládku a vykládku.

Tabulka č. 23: Atributy tabulky Zákazník

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Zakaznik_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Typ_zakaznika	Cizí klíč	Int	Ano	Propojení s tabulkou typ zákazníka
Nazev		Varchar 250	Ano	Název zákazníka
F_ulice		Varchar 50	Ano	Název fakturační ulice
F_cp		Varchar 5	Ano	Fakturační číslo popisné
F_ce		Varchar 5	Ne	Fakturační číslo evidenční
F_psc	Cizí klíč	Int	Ano	Propojení s tabulkou psč
K_osoba		Varchar 100	Ano	Kontaktní osoba na straně zákazníka
K_tel1		Varchar 20	Ano	Primární telefonický kontakt na zákazníka
K_tel2		Varchar 20	Ne	Sekundární telefonický kontakt na zákazníka
K_mail		Varchar 50	Ne	Kontaktní mail zákazníka
Poznamka		Varchar 250	Ne	Doplňující informace o zákazníkovi

Na kartě zákazníka bude společnost evidovat pouze fakturační adresu zákazníka, neboť zákazníci mohou poptávat dopravu mezi různými zdrojovými a cílovými místy (pro jednotlivé zakázky mohou být různá místa nakládky a vykládky), a ty se zákazníkem nemusí být přímo spojeny. Proto k zákazníkovi přiřadíme přímo pouze jedinou adresu – kam bude fakturováno a adresy nakládek a vykládek budou přiřazeny až při vytváření konkrétní objednávky.

3.3.18 Tabulka 18: Zakázka

Tato tabulka slouží k evidování jednotlivých zakázek, zakázka je definována zákazníkem, který poptává služby, pracovníkem, který tuto zakázku vyřizuje a vozidlem které bylo této zakázce přiděleno. Ke každé zakázce je třeba evidovat místo naložení a vyložení přepravovaného nákladu. V této tabulce je rovněž evidována průběžná cena zakázky, která se v případě dokončení zakázky přenesení do externího systému pro fakturaci. Z tohoto externího systému přijde zpět potvrzení, že zakázka byla vyfakturovaná, na základě čehož se zakázka označí jako zaúčtovaná.

Tabulka č. 24: Atributy tabulky Zakázka

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Zakazka_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Zakaznik	Cizí klíč	Int	Ano	Propojení s tabulkou zákazník
Pracovnik	Cizí klíč	Int	Ano	Propojení s tabulkou pracovník
Vozidlo	Cizí klíč	Varchar 17	Ano	Propojení s tabulkou vozidlo
Datum_cas		Datetime	Ano	Datum a čas zakázky
N_ulice		Varchar 50	Ano	Název ulice naložení
N_cp		Varchar 5	Ano	Číslo popisné naložení
N_ce		Varchar 5	Ne	Číslo evidenční naložení
N_psc	Cizí klíč	Int	Ano	Propojení s tabulkou psč
V_ulice		Varchar 50	Ano	Název ulice vyložení
V_cp		Varchar 5	Ano	Číslo popisné vyložení
V_ce		Varchar 5	Ne	Číslo evidenční vyložení
V_psc	Cizí klíč	Int	Ano	Propojení s tabulkou psč
Cena_bez_DPH		Decimal (12,2)	Ano	Vypočítaná cena
DPH		Decimal (12,2)	Ano	Vypočítané DPH
Cena_s_SPH		Decimal (12,2)	Ano	Vypočítaná cena s DPH
Dokoncena		Bit	Ano	Zda je zakázka dokončená nebo ne
Zuctovana		Bit	Ano	Zda je zakázka zaúčtována nebo ne

Co se týče zadávání jednotlivých cen, ty bude systém kalkulovat automaticky – při zadání záznamu do databáze je uživatel zadávat nebude, aby bylo předejito chybám způsobeným

lidským faktorem – na úrovni databáze budou tyto ceny vypočteny při vložení řádku zakázky pomocí triggeru.

3.3.19 Tabulka 19. Řádky zakázky

V této tabulce jsou evidovány jednotlivé řádky zakázky. Každý řádek zakázky reprezentuje jednu službu poskytnutou zákazníkovi, což přepravní společnosti umožňuje jednotlivým zákazníkům v rámci jedné zakázky poskytnout libovolnou kombinaci poskytovaných služeb. Ke každé z poskytovaných služeb je v systému evidována její cena, která je závazná a shodná pro všechny zákazníky (což je způsobeno tím, že se jedná o malou společnost). Na řádku můžeme najít ceny, které ovšem uživatel při vytváření záznamu v tabulce nezadá ručně, ale jsou vypočteny systémem při vložení řádku.

Tabulka č. 25: Atributy tabulky Řádky zakázky

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Radky_ zakazky_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Zakazka	Cizí klíč	Int	Ano	Propojení s tabulkou zakázka
Sluzba	Cizí klíč	Int	Ano	Propojení s tabulkou poskytnutá služba
Pocet_m_je dnotek_na_ radku		Int	Ano	Celé číslo, které určuje počet měrných jednotek
Cena_na_ra dku		Decimal (12,2)	Ano	Cizí klíč z tabulky Ceník
Dph_na_rad ku		Decimal (12,2)	Ano	Výpočet celkové ceny
Cena_DPH _radek		Decimal (12,2)	Ano	Vypočítaná cena

3.3.20 Tabulka 20: Plánovaný servis

Tato tabulka slouží k evidování plánovaného servisu vozidla, je zde uvedeno datum a čas plánovaného servisu, dodavatel servisních služeb, typ servisní operace a osoba, které zodpovídá za provedení plánovaného servisu.

Tabulka č. 26: Atributy tabulky Plánovaný servis

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Planovany_ servis_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Vozidlo	Cizí klíč	Varchar 17	Ano	Propojení s tabulkou vozidlo
Datum		Datetime	Ano	Údaj určující datum a čas plánovaného servisu
Typ_operace	Cizí klíč	Int	Ano	Propojení s tabulkou typ servisní operace
Dodavatel	Cizí klíč	Int	Ano	Propojení s tabulkou dodavatel servisních služeb
Zodpovedna_ osoba	Cizí klíč	Int	Ano	Propojení s tabulkou pracovník

3.3.21 Tabulka 21: Parametry vozidla

Tato tabulka slouží k propojení vozidla s jeho jednotlivými parametry. Předpokladem je, že jedno vozidlo může mít více parametrů.

Tabulka č. 27: Atributy tabulky Parametry vozidla

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Parametry_ vozidla_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Vozidlo	Cizí klíč	Varchar 17	Ano	Propojení s tabulkou vozidlo
Parametr	Cizí klíč	Int	Ano	Propojení s tabulkou parametry
Hodnota_ parametru		Varchar 50	Ne	Skutečná hodnota parametru

3.3.22 Tabulka 22: Servis informace

Tabulka servis informace slouží k evidování informací spojených se servisními operacemi, které se týkají jednotlivých vozidel. Za servisní operace považujeme technické prohlídky, běžné servisní prohlídky a opravy vozidel. Tato tabulka je propojena s tabulkami dodavatel servisních služeb, typ servisní operace, vozidlo a pracovník

Tabulka č. 28: Atributy tabulky Servis informace

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Typ a délka	Not null	Popis
Servis_ info_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Zodpovedna _osoba	Cizí klíč	Int	Ano	Propojení s tabulkou pracovník
Dodavatel	Cizí klíč	Int	Ano	Propojení s tabulkou dodavatel servisních služeb
Typ	Cizí klíč	Int	Ano	Propojení s tabulkou typ servisní operace
Cena		Decimal (12,2)	Ano	Cena za konkrétní servisní operace
Vozidlo	Cizí klíč	Varchar 17	Ano	Propojení s tabulkou vozidlo
Datum		Date	Ano	Datum provedené servisního úkonu
Poznamka		Varchar 250	Ne	Doplňující informace k servisní informaci
Odkaz_ doklad		Varchar_50 0	Ano	Cesta k naskenovanému dokumentu
Zauctovano		Bit	Ano	Zda byla servisní operace zaúčtována

3.3.23 Tabulka 23: Doklady o vozidle

Tabulka doklady o vozidle slouží k propojení vozidla a typu dokladu. V této tabulce jsou evidovány informace o platnosti dokladu, odkaz na naskenovaný doklad a ověření, zda na základě konkrétního dokladu splňuje vozidlo náležitosti potřebné k provozu na pozemních komunikacích, zejména se jedná o technický průkaz, osvědčení o registraci vozidla a mezinárodní doklad o pojištění vozidla.

Tabulka č. 29: Atributy tabulky Doklady o vozidle

(Zdroj: vlastní zpracování)

Název atributu	Primární/ cizí klíč	Typ a délka	Not null	Popis
Doklady_ vozidlo_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Vozidlo	Cizí klíč	Varchar 17	Ano	Propojení s tabulkou vozidlo
Typ_dokladu	Cizí klíč	Int	Ano	Propojení s tabulkou typ dokladu

Odkaz_doklad		Varchar 500	Ano	Cesta k oskenovanému dokumentu
Platnost_od		Date	Ano	Den počátku platnosti dokumentu
Platnost_do		Date	Ne	Den konce platnosti dokumentu
Poznamka		Varchar 250	Ne	Přídavné informace, které souvisejí s dokumentem
Nalezitosti		Bit	Ano	Na základě dokumentu splňuje legislativní podmínky spojené s provozem na pozemních komunikacích

3.3.24 Tabulka 24: kniha jízd

Kromě potřeby evidovat výše popsané entity, projevila společnost rovněž potřebu evidovat v systému také konkrétní jízdy. Tedy především stavy kilometrů, které byly pro daná vozidla aktuální před začátkem jízdy a na jejím konci, a konkrétní časy, kdy bylo vyřizování započato a ukončeno. Zároveň však společnost také potřebuje vědět, v důsledku kterého vyřízení zakázky, či které servisní zakázky došlo k danému změnu stavů.

Mimo tyto informace je pro společnost kvůli transparentnosti nutné evidovat rovněž zda při dané jízdě bylo tankováno či nikoliv. Pokud je s danou jízdou spojené tankování, přála si společnost evidovat, jaké množství pohonných hmot bylo v rámci tankování doplněno do nádrže vozidla.

Tabulka č. 30: Atributy tabulky Kniha jízd

(Zdroj: vlastní zpracování)

Název	Primární/ cizí klíč	Délka a typ	Not null	Popis
Kniha_jezd_ID	Primární klíč	Int	Ano	Automaticky generované číslo, které slouží k jednoznačné identifikaci záznamu
Vozidlo	Cizí klíč	Varchar 17	Ano	Propojení s tabulkou vozidlo
Zakazka	Cizí klíč	Int	Ne	Propojení s tabulkou zakázka
Servis	Cizí klíč	Int	Ne	Propojení s tabulkou servis informace
Stav_na_zacatku		Int	Ano	Stav tachometru na začátku
Stav_na_konci		nt	Ano	Stav tachometru na konci

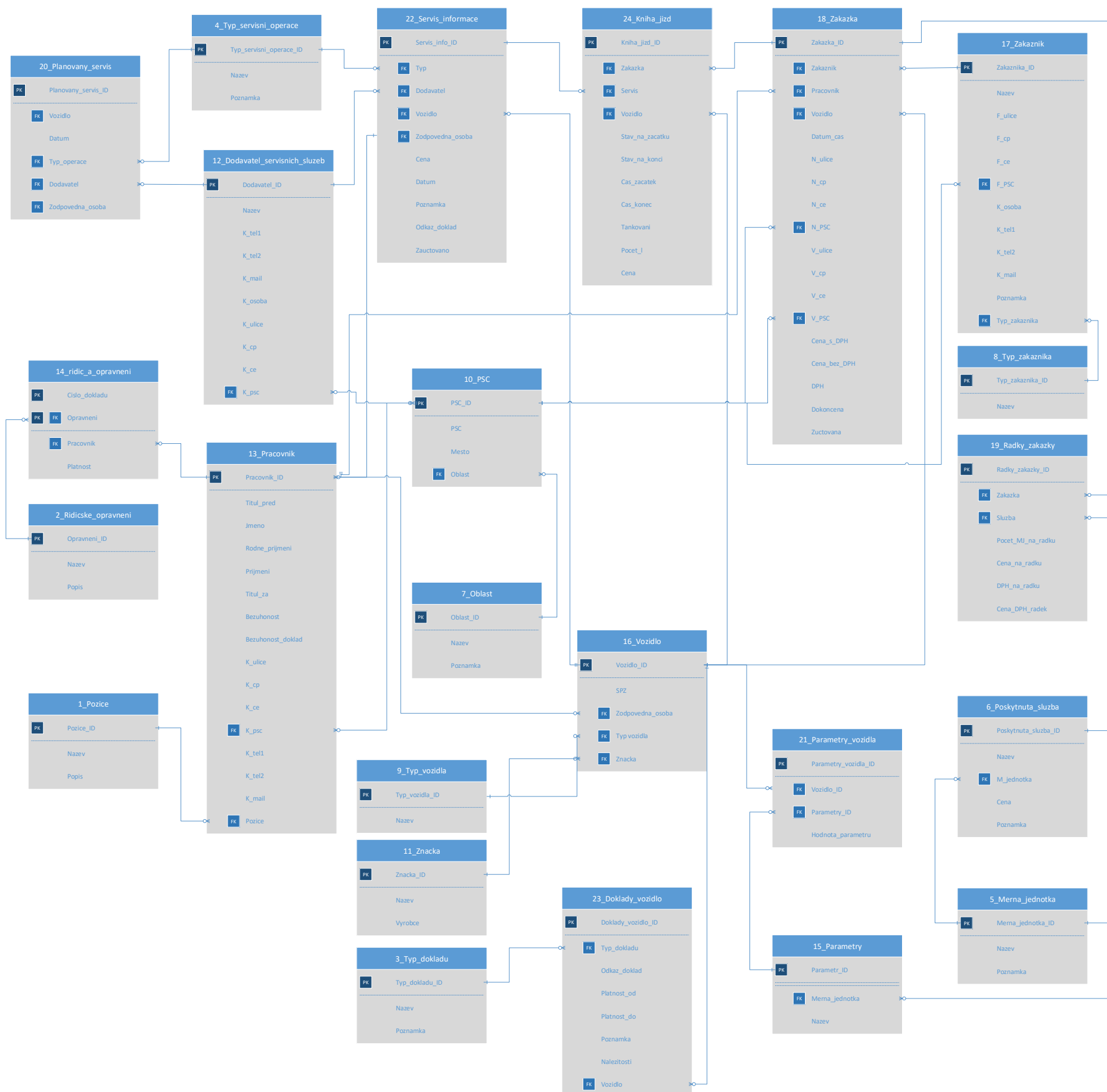
Cas_zacatku		Datetime	Ne	Datum a čas začátku
Cas_konce		Datetime	Ne	Datum a čas konce
Tankovani		Bit	Ano	Tankovani
Pocet_l		Real	Ne	Počet litrů
Cena		Decimal (12,2)	Ne	Cena tankovani

3.3.25 ER diagram

V předchozích podkapitolách jsem se zabýval identifikací entit a jejich vzájemných vazeb a následně jsem se zaměřil na stanovení jednotlivých atributů, které v rámci těchto entit budu potřebovat evidovat.

Celkem jsem v rámci identifikace entit dospěl k tomu, že společnost potřebuje pro evidenci požadovaných údajů vytvořit celkem 24 tabulek. V návrhu bylo začleněno také několik číselníků, které by měly zajistit, že při vyplňování údajů nebude docházet k chybám způsobených lidským faktorem.

Na následujícím obrázku můžeme vidět ER diagram, který slouží k přehlednému znázornění tabulek a jejich vzájemných vztahů. V rámci diagramu jsem si tabulky pro přehled očísloval stejně, jako jsem je čísloval v předchozí.



Obrázek č. 8: ER-diagram

(Zdroj: vlastní zpracování)

3.4 Fyzický návrh

V následující části práce se budu věnovat převedení předchozích částí této práce na fyzický návrh databáze, tato kapitola obsahuje vytvoření jednotlivých tabulek a jejich naplnění testovacími daty. Dále pak vytvořím několik uživatelských pohledů a triggerů.

3.4.1 Vytvoření tabulek a přidání primárních a cizích klíčů

Jako první krok logického návrhu databáze jsem vytvořil jednotlivé tabulky, které jsou popsány v logickém návrhu. Nejprve jsem vytvořil tabulky pomocí příkazu CREATE TABLE a následně jsem jim za pomoci příkazu ALTER TABLE přiřadil primární a cizí klíče.

Následující obrázek zobrazuje příklad kódu pro vytvoření tabulky zákazník. Kompletní kód pro vytvoření všech tabulek je uveden v příloze číslo 1.

```
CREATE TABLE ZAKAZNIK (  
  Zakaznik_ID int not null identity(1,1),  
  Typ_zakaznika int not null,  
  Nazev varchar(250) not null,  
  F_ulice varchar(50) not null,  
  F_cp varchar(5) not null,  
  F_ce varchar(5),  
  F_psc int not null,  
  K_osoba varchar(100) not null,  
  K_tell varchar(20) not null,  
  K_tel2 varchar(20),  
  K_mail varchar(50),  
  Poznamka varchar(250)  
);  
  
ALTER TABLE ZAKAZNIK  
ADD PRIMARY KEY (Zakaznik_ID),  
FOREIGN KEY (Typ_zakaznika) REFERENCES TYP_ZAKAZNIKA(Typ_zakaznika_ID),  
FOREIGN KEY (F_psc) REFERENCES PSC(Psc_ID);
```

Obrázek č. 9: Příklad kódu pro vytvoření tabulky a nastavení klíčů

(Zdroj: vlastní zpracování)

3.4.2 Naplnění tabulek testovacími daty

Pro testování funkcionality databáze jsem do databáze nahrál testovací data. Pro efektivní tvorbu příkazů na vložení těchto dat do databáze jsem využil tabulkový procesor Microsoft Excel. Následující obrázek se týká takto vytvořených příkazů INSERT INTO pro tabulku plánovaný servis.

ID	Vo	DTM	TYP	DOD	ZOS	
1	TMBD	2018-C	1	1	4	INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777','2018-07-24','1','1','4');
2	TMBD	2018-C	2	2	4	INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777','2018-08-11','2','2','4');
3	TMBD	2018-C	3	3	4	INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777','2018-07-15','3','3','4');
4	TMBD	2018-C	4	1	4	INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777','2018-07-09','4','1','4');
5	WFOG	2018-C	1	2	4	INSERT INTO PLANOVANY_SERVIS VALUES ('WFOGXXGBBG7Y42388','2018-09-04','1','2','4');
6	WFOG	2018-C	2	3	4	INSERT INTO PLANOVANY_SERVIS VALUES ('WFOGXXGBBG7Y42389','2018-09-03','2','3','4');
7	WFOG	2018-C	3	1	4	INSERT INTO PLANOVANY_SERVIS VALUES ('WFOGXXGBBG7Y42390','2018-08-02','3','1','4');
8	WFOG	2018-C	4	2	4	INSERT INTO PLANOVANY_SERVIS VALUES ('WFOGXXGBBG7Y42391','2018-07-22','4','2','4');
9	TMAD	2018-C	1	3	4	INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087957','2018-07-15','1','3','4');
10	TMAD	2018-C	2	1	4	INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087958','2018-08-26','2','1','4');
11	TMAD	2018-C	3	2	4	INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087959','2018-07-22','3','2','4');
12	TMAD	2018-C	4	3	4	INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960','2018-08-11','4','3','4');

Obrázek č. 10 Tvorba insertů

(Zdroj: vlastní zpracování)

Následně jsem díky takto vytvořeným příkazům naplnil jednotlivé tabulky testovacími daty. Následující ilustrační obrázek zobrazuje kód pro naplnění tabulky plánovaný servis z prostředí Management Studia. Kompletní kód pro naplnění všech tabulek je součástí přílohy číslo 1.

```

INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777', '2018-07-24', '1', '1', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777', '2018-08-11', '2', '2', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777', '2018-07-15', '3', '3', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777', '2018-07-09', '4', '1', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('WF0GXXGBBG7Y42388', '2018-09-04', '1', '2', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('WF0GXXGBBG7Y42388', '2018-09-03', '2', '3', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('WF0GXXGBBG7Y42388', '2018-08-02', '3', '1', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('WF0GXXGBBG7Y42388', '2018-07-22', '4', '2', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960', '2018-07-15', '1', '3', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960', '2018-08-26', '2', '1', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960', '2018-07-22', '3', '2', '4');
INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960', '2018-08-11', '4', '3', '4');

```

Obrázek č. 11: Příklad kódu pro naplnění tabulky testovacími daty

(Zdroj: vlastní zpracování)

3.4.3 Pohledy

V rámci této podkapitoly jsem vytvořil několik pohledů, které slouží k zobrazení testovacích dat uložených v databázi.

3.4.3.1 Pohled vozidlo a jeho doklady

Tento pohled využívá data uložená v tabulce doklady o vozidle a slouží k jejich přehlednému zobrazení. Díky propojení této tabulky s tabulkou vozidlo, respektive propojení tabulky vozidlo s tabulkou značka je možné zobrazit státní poznávací značku a název značky vozidla, kterého se konkrétní doklady týkají. Dále ve sloupci, který zobrazuje informaci o tom, zda na základě typu dokladu vozidlo splňuje náležitosti k provozu na pozemních komunikacích, využívá příkazu REPLACE a zobrazuje text ANO respektive NE což je uživatelsky přívětivější než zobrazování dat, která jsou přímo uložena v databázi. Následující obrázek zobrazuje skript pro vytvoření tohoto pohledu.

```

CREATE VIEW POHLED_DOKLADY_VOZIDLO AS
SELECT
    DOKLADY_VOZIDLO.Doklady_vozidlo_ID AS [ID záznamu],
    VOZIDLO.SPZ AS [SPZ vozidla],
    ZNACKA.Nazev AS [Název vozidla],
    TYP_DOKLADU.Nazev AS [Typ dokladu],
    DOKLADY_VOZIDLO.Platnost_od AS [Platnost od],
    DOKLADY_VOZIDLO.Platnost_do AS [Platnost do],
    DOKLADY_VOZIDLO.Odkaz_na_doklad AS [Odkaz],
    REPLACE(REPLACE(DOKLADY_VOZIDLO.Nalezitosti, 1, 'ANO'),0,'NE') AS [Vozidlo splňuje náležitosti]
FROM
    DOKLADY_VOZIDLO INNER JOIN VOZIDLO ON DOKLADY_VOZIDLO.Vozidlo = VOZIDLO.Vozidlo_ID
    INNER JOIN ZNACKA ON VOZIDLO.Znacka = ZNACKA.Znacka_ID
    INNER JOIN TYP_DOKLADU ON DOKLADY_VOZIDLO.Typ_dokladu = TYP_DOKLADU_ID

SELECT * FROM POHLED_DOKLADY_VOZIDLO
ORDER BY [SPZ vozidla]

```

Obrázek č. 12: Kód pro vytvoření pohledu Vozidlo a jeho doklady

(Zdroj: vlastní zpracování)

Následující obrázek se týká výše popsaného pohledu a zobrazuje jeho výstup výsledků seřazených dle státní poznávací značky.

ID záznamu	SPZ vozidla	Název vozidla	Typ dokladu	Platnost od	Platnost do	Odkaz	Vozidlo splňuje náležitosti
1	1B17890	Daily IV	Malý techničák	2017-01-01	1900-01-01	Zde bude odkaz	ANO
4	1B17890	Daily IV	Velký techničák	2017-01-01	1900-01-01	Zde bude odkaz	ANO
7	1B17890	Daily IV	Zelená karta	2018-02-01	2019-02-01	Zde bude odkaz	ANO
10	1B17890	Daily IV	Servisní kniha	2017-01-01	2019-01-01	Zde bude odkaz	ANO
2	1BU7654	Daily V	Malý techničák	2017-01-02	1900-01-01	Zde bude odkaz	ANO
5	1BU7654	Daily V	Velký techničák	2017-01-02	1900-01-01	Zde bude odkaz	ANO
8	1BU7654	Daily V	Zelená karta	2018-02-02	2019-02-02	Zde bude odkaz	ANO
11	1BU7654	Daily V	Servisní kniha	2017-01-02	2019-01-02	Zde bude odkaz	ANO
3	3BA9876	A31	Malý techničák	2017-01-03	1900-01-01	Zde bude odkaz	ANO
6	3BA9876	A31	Velký techničák	2017-01-03	1900-01-01	Zde bude odkaz	ANO

Obrázek č. 13: Výstup pohledu Vozidlo a jeho doklady

(Zdroj: vlastní zpracování)

3.4.3.2 Pohled pracovník a jeho zakázky

Tento pohled zobrazuje některá data uložená v tabulce zakázka, slouží k zobrazení ID zakázky, jména pracovníka, názvu zákazníka, vozidla a stavu této zakázky. V rámci tohoto pohledu jsou do sloupce pracovník spojena data z PRACOVNIK.Jmeno a PRACOVNIK.Prijmeni z tabulky pracovník. Tento pohled rovněž využívá příkazu REPLACE v souvislosti s daty, které se týkají stavu zakázky. Následující obrázek představuje skript pro vytvoření tohoto pohledu.

```

CREATE VIEW POHLED_PRACOVNIK_ZAKAZKY AS
SELECT
    ZAKAZKA.Zakazka_ID AS [ID Zakázky], (PRACOVNIK.Jmeno + ' ' + PRACOVNIK.Prijmeni) AS [Pracovník],
    ZAKAZNIK.Nazev AS [Název zákazníka], VOZIDLO.SPZ AS [SPZ vozidla], ZNACKA.Nazev AS [Vozidlo],
    REPLACE(REPLACE(ZAKAZKA.Dokoncena, 1, 'ANO'),0,'NE') AS [Dokončená zakázka],
    REPLACE(REPLACE(ZAKAZKA.Zuctovana, 1, 'ANO'),0,'NE') AS [Zaúčtovaná zakázka]
FROM
    ZAKAZKA INNER JOIN ZAKAZNIK ON ZAKAZKA.Zakaznik = ZAKAZNIK.Zakaznik_ID
    INNER JOIN PRACOVNIK ON ZAKAZKA.pracovnik = PRACOVNIK.Pracovnik_ID
    INNER JOIN VOZIDLO ON ZAKAZKA.Vozidlo = VOZIDLO.Vozidlo_ID
    INNER JOIN ZNACKA ON VOZIDLO.Znacka = ZNACKA.Znacka_ID

SELECT * FROM POHLED_PRACOVNIK_ZAKAZKY
WHERE Pracovník = 'Josef Šváb'

```

Obrázek č. 14: Kód pro vytvoření pohledu Pracovník a jeho zakázky

(Zdroj: vlastní zpracování)

Následující obrázek se týká výstupu výše popsaného pohledu pro zakázky, které vyřizoval pracovník, který se jmenuje Josef Šváb.

	ID Zakázky	Pracovník	Název zákazníka	SPZ vozidla	Vozidlo	Dokončená zakázka	Zaúčtovaná zakázka
1	1	Josef Šváb	Karel Ježek	1BU7654	Daily V	NE	NE
2	7	Josef Šváb	Hejlik, a.s.	1BU7654	Daily V	NE	NE

Obrázek č. 15: Výstup pohledu pracovník a jeho zakázky

(Zdroj: vlastní zpracování)

3.4.3.3 Pohled kniha jízd vozidla

Tento pohled slouží k zobrazení dat, která jsou uložena v tabulce kniha jízd. Tato tabulka má jako cizí klíč z tabulky vozidlo vin kód vozidla, nicméně z důvodu přehlednosti tento pohled zobrazuje státní poznávací značku vozidla a název vozidla. V rámci tohoto pohledu jsou u dat, pro které to má smysl přidány k zobrazovaným datům i jejich jednotky. Následující obrázek se týká skriptu pro vytvoření tohoto pohledu.

```

CREATE VIEW POHLED_VOZIDLO_KNIHA_JIZD AS
SELECT
    KNIHA_JIZD.Kniha_jizd_ID AS [ID záznamu],
    VOZIDLO.SPZ AS [SPZ vozidla],
    ZNACKA.Nazev AS [Název vozidla],
    KNIHA_JIZD.Zakazka AS [ID zakázky],
    KNIHA_JIZD.Servis AS [ID servisu],
    cast(KNIHA_JIZD.Stav_na_zacatku as varchar) + ' km' AS [Stav km na začátku],
    cast(KNIHA_JIZD.Stav_na_konci as varchar) + ' km' AS [Stav km na konci],
    KNIHA_JIZD.Cas_zacatku AS [Čas začátku],
    KNIHA_JIZD.Cas_konce AS [Čas konce],
    REPLACE(REPLACE(KNIHA_JIZD.Tankovani, 1, 'ANO'),0,'NE') AS [Doplnění paliva],
    cast(KNIHA_JIZD.Pocet_l as varchar) + ' l' AS [Počet litrů],
    cast(KNIHA_JIZD.Cena as varchar) + ' Kč' AS [Cena tankování]
FROM
    KNIHA_JIZD INNER JOIN VOZIDLO ON KNIHA_JIZD.Vozidlo = VOZIDLO.Vozidlo_ID
    INNER JOIN ZNACKA ON VOZIDLO.Znacka = ZNACKA.Znacka_ID
SELECT* FROM POHLED_VOZIDLO_KNIHA_JIZD
WHERE [SPZ vozidla] = '3BA9876'

```

Obrázek č. 16: Kód pro vytvoření pohledu Kniha jízd vozidla

(Zdroj: vlastní zpracování)

Následující obrázek představuje využití výše popsaného pohledu pro zobrazení záznamů, které se týkají vozidla, které má státní poznávací značku 3BA 9876.

ID záznamu	SPZ vozidla	Název vozidla	ID zakázky	ID servisu	Stav km na začátku	Stav km na konci	Čas začátku	Čas konce	Doplnění paliva	Počet litrů	Cena tankování	
1	12	3BA9876	A31	NULL	2	32102 km	32123 km	2018-05-15 11:00:00.000	2018-05-15 15:00:00.000	ANO	211	630.00 Kč
2	15	3BA9876	A31	NULL	5	12345 km	12987 km	2018-05-18 11:00:00.000	2018-05-18 18:00:00.000	ANO	241	720.00 Kč
3	18	3BA9876	A31	NULL	8	12987 km	13234 km	2018-05-22 11:00:00.000	2018-05-22 19:00:00.000	ANO	271	810.00 Kč
4	26	3BA9876	A31	6	NULL	29305 km	31293 km	2018-05-09 11:00:02.000	2018-05-09 16:00:02.000	ANO	151	450.00 Kč
5	29	3BA9876	A31	9	NULL	31293 km	32102 km	2018-05-12 11:00:00.000	2018-05-12 21:00:00.000	ANO	181	540.00 Kč
6	32	3BA9876	A31	NULL	2	32102 km	32123 km	2018-05-15 11:00:00.000	2018-05-15 15:00:00.000	ANO	211	630.00 Kč
7	35	3BA9876	A31	NULL	5	12345 km	12987 km	2018-05-18 11:00:00.000	2018-05-18 18:00:00.000	ANO	241	720.00 Kč
8	38	3BA9876	A31	NULL	8	12987 km	13234 km	2018-05-22 11:00:00.000	2018-05-22 19:00:00.000	ANO	271	810.00 Kč

Obrázek č. 17: Výstup pohledu Kniha jízd vozidla

(Zdroj: vlastní zpracování)

3.4.3.4 Pohled zakázka a její řádky

Tento pohled slouží k zobrazení jednotlivých řádků zakázky, které patří konkrétní zakázce. Pohled čerpá data z tabulky řádky zakázky. U zobrazovaných dat jsou ve sloupcích, u kterých to má smysl zobrazovány i jejich jednotky pomocí příkazu CAST. Následující obrázek představuje skript pro vytvoření tohoto pohledu.

```

CREATE VIEW POHLED_ZAKAZKA_A_JEJI_RADKY AS
SELECT
    RADKY_ZAKAZKY.Radky_zakazky_ID AS [ID záznamu],
    ZAKAZKA.Zakazka_ID [ID zakázky],
    ZAKAZNIK.Nazev AS [Zákazník],
    ZAKAZKA.Datum_cas AS [Datum řešení zakázky],
    POSKYTNUTA_SLUZBA.Nazev AS [Poskytnutá služba],
    RADKY_ZAKAZKY.Pocet_m_jednotek_na_radku AS [Počet měrných jednotek],
    MERNA_JEDNOTKA.Nazev AS [Měrná jednotka],
    cast(RADKY_ZAKAZKY.Cena_na_radku as varchar) + ' Kč' AS [Cena],
    cast(RADKY_ZAKAZKY.DPH_na_radku as varchar) + ' Kč' AS [DPH],
    cast(RADKY_ZAKAZKY.Cena_DPH_radek as varchar) + ' Kč' AS [Cena s DPH]
FROM
    RADKY_ZAKAZKY INNER JOIN ZAKAZKA ON RADKY_ZAKAZKY.Zakazka = ZAKAZKA.Zakazka_ID
    INNER JOIN ZAKAZNIK ON ZAKAZKA.Zakaznik = ZAKAZNIK.Zakaznik_ID
    INNER JOIN POSKYTNUTA_SLUZBA ON RADKY_ZAKAZKY.Sluzba = POSKYTNUTA_SLUZBA.Poskytnuta_sluzba_ID
    INNER JOIN MERNA_JEDNOTKA ON POSKYTNUTA_SLUZBA.Merna_jednotka = MERNA_JEDNOTKA.Merna_jednotka_ID
SELECT * FROM POHLED_ZAKAZKA_A_JEJI_RADKY
WHERE Zákazník = 'Hejlík, a.s.'

```

Obrázek č. 18: Kód pro vytvoření pohledu Zakázka a její řádky

(Zdroj: vlastní zpracování)

Následující obrázek se týká výše popsaného pohledu a jeho využití pro zobrazení zakázek, které souvisí se zákazníkem jménem Hejlík.

ID záznamu	ID zakázky	Zákazník	Datum řešení zakázky	Poskytnutá služba	Počet měrných jednotek	Měrná jednotka	Cena	DPH	Cena s DPH
20	7	Hejlík, a.s.	2018-07-15 00:00:00.000	Přeprava po ČR	86	KM	543.52 Kč	144.48 Kč	688.00 Kč
21	7	Hejlík, a.s.	2018-07-15 00:00:00.000	Naložení nákladu	17	KS	134.30 Kč	35.70 Kč	170.00 Kč
22	7	Hejlík, a.s.	2018-07-15 00:00:00.000	Vyložení nákladu	17	KS	134.30 Kč	35.70 Kč	170.00 Kč
23	7	Hejlík, a.s.	2018-07-15 00:00:00.000	Výnos do patra	8	KS	126.40 Kč	33.60 Kč	160.00 Kč
24	8	Hejlík, a.s.	2018-08-24 00:00:00.000	Přeprava po ČR	96	KM	606.72 Kč	161.28 Kč	768.00 Kč
25	8	Hejlík, a.s.	2018-08-24 00:00:00.000	Čekání	35	M	82.95 Kč	22.05 Kč	105.00 Kč
26	8	Hejlík, a.s.	2018-08-24 00:00:00.000	Naložení nákladu	37	KS	292.30 Kč	77.70 Kč	370.00 Kč
27	8	Hejlík, a.s.	2018-08-24 00:00:00.000	Vyložení nákladu	37	KS	292.30 Kč	77.70 Kč	370.00 Kč

Obrázek č. 19: Výstup pohledu Zakázka a její řádky

(Zdroj: vlastní zpracování)

3.4.4 Triggery

Tato kapitola obsahuje triggery, neboli spouště. Trigger obsahuje posloupnost příkazů, které se provedou na základě předem definované události, může jít například o změnu souvisejících záznamů v případě změny některého záznamu.

3.4.4.1 Trigger pro mazání zákazníka

Tento trigger se používá pro mazání záznamů, které nelze smazat jednoduše pomocí příkazu DELETE, protože jsou použity jako cizí klíče v jiných záznamech. Nejprve jsem se tedy zamyslel nad tím, kterých tabulek se mazání zákazníka dotkne. Předtím, než mohu konkrétního zákazníka smazat, musím nejprve odstranit všechny jeho zakázky, řádky

jeho zakázek a záznamy z knihy jízd, které jsou těmito zakázkami propojeny. V prvním kroku jsem si definoval všechny proměnné, které budu pro správné fungování triggeru potřebovat.

```
CREATE TRIGGER MAZANI_ZAKAZNIKA ON ZAKAZNIK
INSTEAD OF DELETE AS

BEGIN
DECLARE @POMOCNA AS varchar
DECLARE @POMOCNA_INT AS int
DECLARE @ZAKAZNIK_ID AS int

SET @ZAKAZNIK_ID = (SELECT TOP 1 Zakaznik_ID FROM deleted);

DECLARE NAJDIZAKAZKY CURSOR FOR
SELECT
ZAKAZKA.Zakazka_ID FROM ZAKAZKA WHERE ZAKAZKA.Zakaznik =@ZAKAZNIK_ID

OPEN NAJDIZAKAZKY
FETCH NAJDIZAKAZKY INTO @POMOCNA
WHILE @@FETCH_STATUS = 0
BEGIN

SET @POMOCNA_INT = CAST (@POMOCNA AS INT)
SELECT * FROM KNIHA_JIZD WHERE KNIHA_JIZD.Zakazka = @POMOCNA_INT
DELETE FROM KNIHA_JIZD WHERE KNIHA_JIZD.Zakazka = @POMOCNA_INT
SELECT * FROM RADKY_ZAKAZKY WHERE RADKY_ZAKAZKY.Zakazka = @POMOCNA_INT
DELETE FROM RADKY_ZAKAZKY WHERE RADKY_ZAKAZKY.Zakazka = @POMOCNA_INT
SELECT * FROM ZAKAZKA WHERE ZAKAZKA.Zakazka_ID = @POMOCNA_INT
DELETE FROM ZAKAZKA WHERE Zakazka_ID = @POMOCNA_INT

FETCH NEXT FROM NAJDIZAKAZKY INTO @POMOCNA
END

CLOSE NAJDIZAKAZKY
DEALLOCATE NAJDIZAKAZKY

DELETE FROM ZAKAZNIK WHERE Zakaznik_ID = @ZAKAZNIK_ID
END

DELETE FROM ZAKAZNIK WHERE Zakaznik_ID = 3
```

Obrázek č. 20: Kód pro vytvoření triggeru na mazání zákazníka

(Zdroj: vlastní zpracování)

V druhém kroku jsem si z tabulky deleted zjistil ID záznamu, který chce uživatel odstranit.

Jelikož předpokládám, že zákazník může poptávat služby přepravní společnosti opakovaně, je nutné s tím počítat i v triggeru. Tudíž jsem musel použít CURSOR, který

mi umožní proházet po jednom určitou sadu záznamů. Kurzor tedy vždy vezme nalezenou zakázku daného zákazníka a odstraní záznamy z knihy jízd, řádky této zakázky a nakonec zakázku samotnou.

Následující obrázek znázorňuje záznamy, kterých se smazání zákazníka týká, a jsou odstraněny výše popsaným triggerem.

Kniha_jezd_ID	Vozidlo	Stav_na_zacatku	Stav_na_konci	Cas_zacatku	Cas_konce	Tankovani	Pocet_J	Cena	Zakazka	Servis						
1	25	WFOGXXGBBG7Y42388	10234	10566	2018-05-08 11:00:00.000	2018-05-08 15:00:00.000	1	14	420.00	5	NULL					
Radky_zakazky_ID	Zakazka	Sluzba	Pocet_m_jednotek_na_radku	Cena_na_radku	DPH_na_radku	Cena_DPH_radek										
1	13	5	1	38	240.16	63.84	304.00									
2	14	5	2	19	45.03	11.97	57.00									
3	15	5	3	7	55.30	14.70	70.00									
4	16	5	4	7	55.30	14.70	70.00									
5	17	5	5	9	142.20	37.80	180.00									
Zakazka_ID	Zakaznik	pracovnik	Vozidlo	Datum_cas	N_ulice	N_cp	N_ce	N_psc	V_ulice	V_cp	V_ce	V_psc	Cena_bez_DPH	DPH	Cena_s_DPH	
1	5	3	2	TMADB51CAAJ087960	2018-08-04 00:00:00.000	Žlutý kopec	389	25	5	U Koupaliště	986	87	15	3142.62	835.38	3978.00

Obrázek č. 21: Výstup triggeru pro mazání zákazníka

(Zdroj: vlastní zpracování)

3.4.4.2 Trigger pro nastavení cen

Tento trigger má za účel provést při vložení nového řádku do tabulky řádky zakázky, novou kalkulaci ceny na tomto řádku a ceny celé zakázky. Nejprve tedy dojde ke vložení řádku do tabulky řádky zakázky se všemi cenami rovnými 0 Kč, následně pak systém automaticky provede dopočítání cen dle aktuálního nastavení ceny služby.

Nejprve jsem si tedy deklaroval hlavičku triggeru, který bude reagovat na vložení řádku do tabulky řádky zakázky a provede se po vložení záznamu. Následně jsem si deklaroval potřebné proměnné. Tento trigger bude aktualizovat 3 druhy cen: cenu bez DPH, cenu s DPH a DPH.

```
CREATE TRIGGER NASTAVENI_CENY ON RADKY_ZAKAZKY
AFTER INSERT
AS
BEGIN
    DECLARE @J_CENA AS DECIMAL(12,2)
    DECLARE @J_CENA_DPH AS DECIMAL(12,2)
    DECLARE @J_DPH AS DECIMAL(12,2)
    DECLARE @CENA AS DECIMAL(12,2)
    DECLARE @CENA_DPH AS DECIMAL(12,2)
    DECLARE @DPH AS DECIMAL(12,2)
    DECLARE @SLUZBA AS INT
    DECLARE @MNOZSTVI AS INT
    DECLARE @RADEK AS INT
    DECLARE @POMOCNA AS DECIMAL(12,2)
    DECLARE @ZAKAZKA AS INT
```

Obrázek č. 22: Deklarace proměnných v triggeru pro nastavení cen

(Zdroj: vlastní zpracování)

Dále jsem definoval všechny potřebné proměnné. Jejich definici můžeme vidět na předcházejícím obrázku.

```
SELECT Radky_zakazky_ID FROM inserted
SET @ZAKAZKA = (SELECT ZAKAZKA FROM inserted)
SET @RADEK = (SELECT Radky_zakazky_ID FROM inserted)
SELECT Pocet_m_jednotek_na_radku FROM inserted
SET @MNOZSTVI = (SELECT Pocet_m_jednotek_na_radku FROM inserted)
SELECT Sluzba FROM inserted
SET @SLUZBA = (SELECT Sluzba FROM inserted)

SELECT Cena FROM POSKYTNUTA_SLUZBA WHERE POSKYTNUTA_SLUZBA.Poskytnuta_sluzba_ID = @SLUZBA
SET @J_CENA_DPH = (SELECT Cena FROM POSKYTNUTA_SLUZBA WHERE POSKYTNUTA_SLUZBA.Poskytnuta_sluzba_ID = @SLUZBA)
SET @J_CENA = 0.79 * @J_CENA_DPH
SET @J_DPH = 0.21 * @J_CENA_DPH
SET @CENA = @MNOZSTVI * @J_CENA
SET @DPH = @MNOZSTVI * @J_DPH
SET @CENA_DPH = @MNOZSTVI * @J_CENA_DPH
```

Obrázek č. 23: Kód pro nastavení hodnot proměnných

(Zdroj: vlastní zpracování)

V dalším kroku jsem potřeboval zjistit, ke které zakázce byl vložen nový řádek. To můžu zjistit z právě vloženého záznamu, který byl dočasně uložen do systémové tabulky INSERTED. Dále potřebuji znát číslo právě vloženého řádku, množství jednotek dané služby, pro které je potřeba cenu spočítat a samozřejmě potřebuji zjistit ID služby, abych si mohl správně nastavit cenu. Jak již bylo řečeno v předchozích částech práce, pro zjednodušení je u služby cena uvedena pouze s DPH. Následně jsem provedl výpočet zbylých jednotkových cen (tedy ceny bez DPH a DPH) a následně jsem si vypočetl ceny, které mají být na řádek nastaveny.

Poté už stačí pouze provést aktualizaci právě vloženého záznamu pomocí příkazu UPDATE. Tento příkaz se má ovšem provést pouze pro právě vložený řádek, protože je nutné aktualizované řádky omezit na základě ID řádku.

```
UPDATE RADKY_ZAKAZKY
SET Cena_DPH_radek = @CENA_DPH, Cena_na_radku = @CENA, DPH_na_radku = @DPH
WHERE Radky_zakazky_ID = @RADEK;
```

Obrázek č. 24: Kód pro vložení vypočtených hodnot do řádku zakázky

(Zdroj: vlastní zpracování)

Jelikož budu používat pro další práci stejné proměnné, musím z bezpečnostních důvodů nejprve provést anulování jejich hodnot, aby nedošlo k vložení chybné hodnoty. Následně

potřebuji aktualizovat ceny ještě na hlavičce zakázky, jelikož se změnil počet řádků a vypočtená cena by tudíž nemusela být aktuální. K tomuto výpočtu použiji tři kurzory, které budou nyní popsány.

První kurzor má za účel vypočíst součet cen s DPH na jednotlivých řádcích. Kurzor si tedy na základě ID zakázky najde k ní příslušující řádky. Poté prochází tyto řádky řádek po řádku a sčítá cenu na nich uvedenou. Na následujícím obrázku můžeme vidět kurzor, který slouží pro výpočet ceny s DPH. Ostatní kurzory pro výpočet Ceny bez DPH a DPH jsou součástí přílohy číslo 2 jakožto i kompletní kód tohoto triggeru.

```
SET @CENA = 0
SET @CENA_DPH = 0
SET @DPH = 0

DECLARE AKTUALIZUJ_HLAVICKU CURSOR FOR
SELECT Cena_DPH_radek FROM RADKY_ZAKAZKY WHERE Zakazka=@ZAKAZKA

OPEN AKTUALIZUJ_HLAVICKU
FETCH AKTUALIZUJ_HLAVICKU INTO @POMOCNA
WHILE @@FETCH_STATUS=0 BEGIN
SET @CENA_DPH = @CENA_DPH + @POMOCNA
FETCH NEXT FROM AKTUALIZUJ_HLAVICKU INTO @POMOCNA
END
CLOSE AKTUALIZUJ_HLAVICKU
DEALLOCATE AKTUALIZUJ_HLAVICKU
```

Obrázek č. 25: Kód pro kurzor pro výpočet cen

(Zdroj: vlastní zpracování)

Poté, co jsem pomocí kurzorů vypočetl jednotlivé ceny, musím ještě tyto ceny vložit do hlavičky zakázky, k čemuž používám příkaz UPDATE, který nahradí původní hodnoty hodnotami nově vypočtenými.

```
UPDATE ZAKAZKA
SET Cena_s_DPH = @CENA_DPH, Cena_bez_DPH = @CENA, DPH = @DPH
WHERE Zakazka_ID = @ZAKAZKA;
```

Obrázek č. 26: Kód pro aktualizaci údajů na hlavičce zakázky

(Zdroj: vlastní zpracování)

Na následujícím obrázku můžeme vidět průběh celého triggeru. Nejprve si systém našel ID nově vytvořeného řádku zakázky (v tomto případě rovné 64). V dalším kroku si systém zjistil, kolik měrných jednotek je na daném řádku (v našem případě je to 132 jednotek). Následně si systém zjistil, jaká služba je na řádek vkládána (v tomto případě se jedná o službu s ID rovným 1). Následně systém provedl výpočet a uložení cen pro daný řádek číslo 64 a hodnoty aktualizoval na příslušném řádku a následně došlo k přepočtu sumy na hlavičce zakázky.

Radky_zakazky_ID						
1	64					
Pocet_m_jednotek_na_radku						
1	132					
Sluzba						
1	1					
Cena						
1	8.00					
Radky_zakazky_ID	Zakazka	Sluzba	Pocet_m_jednotek_na_radku	Cena_na_radku	DPH_na_radku	Cena_DPH_radek
1	34	2	1	76	480.32	127.68
2	35	2	2	15	35.55	9.45
3	64	2	1	132	834.24	221.76
Zakazka_ID	CENA_bez_DPH	DPH	Cena_s_DPH			
1	2	1350.11	358.89	1709.00		

Obrázek č. 27: Výstup triggeru pro aktualizaci cen

(Zdroj: vlastní zpracování)

ZÁVĚR

Cílem této práce bylo navrhnout databázi pro přepravní firmu, nejdříve jsem zpracoval teoretická východiska práce, následně jsem formou osobního rozhovoru získal podklady pro analýzu současného stavu. V rámci vlastního návrhu řešení jsem využil podklady získané pro analýzu současného stavu a navrhl databázi, která poskytuje požadovanou funkcionalitu.

Mnou navržená databáze umožňuje evidovat informace spojené se zákazníky a zakázkami které se těchto zákazníků týkají, informace spojené s pracovníky a vozidly. Dále je pak možné evidovat jednotlivé doklady, které náleží vozidlům, plánované servisní operace spojené s vozidlem a informace o proběhlých servisních operacích.

SEZNAM POUŽITÝCH ZDROJŮ

- (1) CONOLLY, Thomas, Carolyn BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Vyd. 1. Brno: Computer Press, 2009, 584 s. : il. ISBN 978-80-251-2328-7.
- (2) KOCH, Miloš a Bernard NEUWIRTH. *Datové a funkční modelování*. Vyd. 4., rozšířené. Brno: Akademické nakladatelství CERM, 2010, 142 s. : il., grafy, tab. ISBN 978-80-214-4125-5.
- (3) KROENKE, David, David J AUER a Jakub GONER. *Databáze*. 1. vyd. Brno: Computer Press, 2015, 496 s. : il., grafy, tab. ISBN 978-80-251-4352-0.
- (4) OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Vyd. 1. Brno: Computer Press, 2008, 240 s. : il. ; 23 cm. ISBN 978-80-251-1707-1.
- (5) KOCH, Miloš a Viktor ONDRÁK. *Informační systémy a technologie*. Vyd. 3. Brno: Akademické nakladatelství CERM, 2008, 166 s. : il., grafy, tab. ISBN 978-80-214-3732-6.
- (6) STEPHENS, Ryan K, Ronald R PLEW, Arie JONES a Lukáš KREJČÍ. *Naučte se SQL za 28 dní*. Vyd. 1. Brno: Computer Press, 2010, 728 s. : il. ISBN 978-80-251-2700-1.
- (7) KŘÍŽ, Jiří. *Databázové systémy*. Brno, 2016. Přednáška. VUT v Brně, Fakulta podnikatelská.
- (8) 10. díl: *MS-SQL Krok za krokem: Změna struktury, transakce a pohledy* [online]. ITNetwork.com: Žůrek, 2018 [cit. 2018-05-17]. Dostupné z: <https://www.itnetwork.cz/ms-sql/mssql-database-tutorial-zmena-struktury-database-transakce-pohledy>
- (9) 12.díl MS SQL Trigger (DML) [online]. ITNetwork.com: Gallas, 2018 [cit. 2018-05-17]. Dostupné z: <https://www.itnetwork.cz/ms-sql/ms-sql-trigger-dml>

SEZNAM POUŽITÝCH OBRÁZKŮ

Obrázek č. 1 Schéma databázového systému	15
Obrázek č. 2 Základní značky vývojového diagramu.....	19
Obrázek č. 3 Vztah 1:1	22
Obrázek č. 4 Vztah 1:N.....	22
Obrázek č. 5 Vztah N:M.....	23
Obrázek č. 6 Dekompozice vztahu N:M.....	23
Obrázek č. 7: Vývojový diagram kontaktování zákazníkem	32
Obrázek č. 8: ER-diagram	50
Obrázek č. 9: Příklad kódu pro vytvoření tabulky a nastavení klíčů	51
Obrázek č. 10 Tvorba insertů.....	52
Obrázek č. 11: Příklad kódu pro naplnění tabulky testovacími daty	53
Obrázek č. 12: Kód pro vytvoření pohledu Vozidlo a jeho doklady	54
Obrázek č. 13: Výstup pohledu Vozidlo a jeho doklady	54
Obrázek č. 14: Kód pro vytvoření pohledu Pracovník a jeho zakázky.....	55
Obrázek č. 15: Výstup pohledu pracovník a jeho zakázky.....	55
Obrázek č. 16: Kód pro vytvoření pohledu Kniha jízd vozidla.....	56
Obrázek č. 17: Výstup pohledu Kniha jízd vozidla	56
Obrázek č. 18: Kód pro vytvoření pohledu Zakázka a její řádky	57
Obrázek č. 19: Výstup pohledu Zakázka a její řádky	57
Obrázek č. 20: Kód pro vytvoření triggeru na mazání zákazníka	58
Obrázek č. 21: Výstup triggeru pro mazání zákazníka	59
Obrázek č. 22: Deklarace proměnných v triggeru pro nastavení cen	59
Obrázek č. 23: Kód pro nastavení hodnot proměnných.....	60
Obrázek č. 24: Kód pro vložení vypočtených hodnot do řádku zakázky	60
Obrázek č. 25: Kód pro kurzor pro výpočet cen.....	61
Obrázek č. 26: Kód pro aktualizaci údajů na hlavičce zakázky	61
Obrázek č. 27: Výstup triggeru pro aktualizaci cen.....	62

SEZNAM POUŽITÝCH TABULEK

Tabulka č. 1 Datové typy pro celá čísla.....	27
Tabulka č. 2 Datové typy pro čísla s pohyblivou desetinou čárkou	27
Tabulka č. 3 Datové typy spojené s časem	28
Tabulka č. 4 Datové typy pro text.....	28
Tabulka č. 5: Identifikace entit	33
Tabulka č. 6: Identifikace základních relací	34
Tabulka č. 7: Atributy tabulky Pozice	35
Tabulka č. 8: Atributy tabulky Řidičské oprávnění	36
Tabulka č. 9: Atributy tabulky Typ dokladu.....	36
Tabulka č. 10: Atributy tabulky Typ servisní operace.....	37
Tabulka č. 11: Atributy tabulky Měrná jednotka.....	37
Tabulka č. 12: Atributy tabulky Poskytnutá služba	38
Tabulka č. 13: Atributy tabulky Oblast	38
Tabulka č. 14: Atributy tabulky Typ zákazníka	38
Tabulka č. 15: Atributy tabulky Typ vozidla.....	39
Tabulka č. 16: Atributy tabulky PSČ.....	39
Tabulka č. 17: Atributy tabulky Značka	39
Tabulka č. 18: Atributy tabulky Dodavatel	40
Tabulka č. 19: Atributy tabulky Pracovník.....	41
Tabulka č. 20: Atributy tabulky Řidič a řidičské oprávnění.....	41
Tabulka č. 21: Atributy tabulky Parametry	42
Tabulka č. 22: Atributy tabulky Vozidlo	42
Tabulka č. 23: Atributy tabulky Zákazník	43
Tabulka č. 24: Atributy tabulky Zakázka	44
Tabulka č. 25: Atributy tabulky Řádky zakázky	45
Tabulka č. 26: Atributy tabulky Plánovaný servis.....	46
Tabulka č. 27: Atributy tabulky Parametry vozidla.....	46
Tabulka č. 28: Atributy tabulky Servis informace.....	47
Tabulka č. 29: Atributy tabulky Doklady o vozidle	47
Tabulka č. 30: Atributy tabulky Kniha jízd	48

SEZNAM PŘÍLOH

Příloha č. 1: Skript pro vytvoření databáze.....	I
Příloha č. 2: Trigger pro nastavení ceny.....	XXIX

Příloha č. 1: Skript pro vytvoření databáze, tabulek, určení primárních a cizích klíčů a naplnění tabulek testovacími daty (Zdroj: vlastní zpracování)

```
create database final3;
```

```
use final3;
```

```
--1.
```

```
CREATE TABLE POZICE (  
Pozice_ID int not null identity(1,1),  
Nazev varchar(30) not null,  
Popis varchar(250)  
);
```

```
ALTER TABLE POZICE  
ADD PRIMARY KEY (Pozice_ID);
```

```
INSERT INTO POZICE VALUES('Řidič','Doprava dle požadavků zákazníka');  
INSERT INTO POZICE VALUES('Servisní technik','Má na starosti údržbu vozidel');  
INSERT INTO POZICE VALUES('Vedoucí','Má na starosti zpracování zakázek');
```

```
--2.
```

```
CREATE TABLE RIDICKSKE_OPRAVNENI (  
Opravneni_ID int not null identity(1,1),  
Nazev varchar(3) not null,  
Popis varchar(250)  
);
```

```
ALTER TABLE RIDICKSKE_OPRAVNENI  
ADD PRIMARY KEY (Opravneni_ID);
```

```

INSERT INTO RIDICSKE_OPRAVNENI VALUES('AM','Motorka');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('A1','Motorka');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('A2','Motorka');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('A','Motorka');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('B1','Osobní vozidlo');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('B','Osobní vozidlo');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('C1','Nákladní vozidlo');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('C','Nákladní vozidlo');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('D1','Autobusy');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('D','Autobusy');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('BE','Osobní auto s přívěsem');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('C1E','Nákladní auto s přívěsem');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('CE','Velké nákladní auto');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('D1E','Autobus s přívěsem');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('DE','Autobus s přívěsem');
INSERT INTO RIDICSKE_OPRAVNENI VALUES('T','Traktor');

```

--3.

```

CREATE TABLE TYP_DOKLADU (
Typ_dokladu_ID int identity(1,1) not null,
Nazev varchar(50) not null,
Poznamka varchar(250)
);

ALTER TABLE TYP_DOKLADU
ADD PRIMARY KEY (TYP_DOKLADU_ID);

```

```
INSERT INTO TYP_DOKLADU VALUES('Malý techničák','Osvědčení o registraci vozidla');
```

```
INSERT INTO TYP_DOKLADU VALUES('Velký techničák','Technický průkaz vozidla');
```

```
INSERT INTO TYP_DOKLADU VALUES('Zelená karta','Osvědčení o pojištění vozidla');
```

```
INSERT INTO TYP_DOKLADU VALUES('Servisní kniha','Pravidelné servisní prohlídky');
```

--4.

```
CREATE TABLE TYP_SERVISNI_OPERACE (  
Typ_servisni_operace_ID int identity(1,1) not null,  
Nazev varchar(50) not null,  
Poznamka varchar(250)  
);
```

```
ALTER TABLE TYP_SERVISNI_OPERACE  
ADD PRIMARY KEY (Typ_servisni_operace_ID);
```

```
INSERT INTO TYP_SERVISNI_OPERACE VALUES('Technická prohlídka','Pravidelně dle stáří vozidla');
```

```
INSERT INTO TYP_SERVISNI_OPERACE VALUES('Výměna oleje','1x ročně či dle kilometrů');
```

```
INSERT INTO TYP_SERVISNI_OPERACE VALUES('Servis klimatizace','1x ročně na jaře');
```

```
INSERT INTO TYP_SERVISNI_OPERACE VALUES('Pravidelná servisní prohlídka','Dle najetých kilometrů');
```

--5.

```
CREATE TABLE MERNA_JEDNOTKA (  

```

```
Merna_jednotka_ID int identity(1,1) not null,  
Nazev varchar(50) not null,  
Poznamka varchar(250)  
);
```

```
ALTER TABLE MERNA_JEDNOTKA  
ADD PRIMARY KEY (Merna_jednotka_ID);
```

```
INSERT INTO MERNA_JEDNOTKA VALUES('H','Hodina');  
INSERT INTO MERNA_JEDNOTKA VALUES('M','Minuta');  
INSERT INTO MERNA_JEDNOTKA VALUES('KM','Kilometry');  
INSERT INTO MERNA_JEDNOTKA VALUES('KS','Kusy');  
INSERT INTO MERNA_JEDNOTKA VALUES('T','Tuny');  
INSERT INTO MERNA_JEDNOTKA VALUES('KG','Kilogramy');  
INSERT INTO MERNA_JEDNOTKA VALUES('CZK','Koruny');  
INSERT INTO MERNA_JEDNOTKA VALUES('EUR','Euro');  
INSERT INTO MERNA_JEDNOTKA VALUES('MM','Milimetry');  
INSERT INTO MERNA_JEDNOTKA VALUES('CM','Centimetry');  
INSERT INTO MERNA_JEDNOTKA VALUES('M','Metry');  
INSERT INTO MERNA_JEDNOTKA VALUES('M3','Metr krychlový');  
INSERT INTO MERNA_JEDNOTKA VALUES('L','Litr');  
INSERT INTO MERNA_JEDNOTKA VALUES('KMH','Kilometry za hodinu');  
INSERT INTO MERNA_JEDNOTKA VALUES('CCM','Kubický centimetr');  
INSERT INTO MERNA_JEDNOTKA VALUES('L/100','Litr na 100 km');  
INSERT INTO MERNA_JEDNOTKA VALUES('KWH','Kilowatt hodina');
```

--6.

```
CREATE TABLE POSKYTNUTA_SLUZBA (  
Poskytnuta_sluzba_ID int identity(1,1) not null,  
Nazev varchar(50) not null,  
Merna_jednotka int not null,  
Cena decimal(12,2) not null,  
Poznamka varchar(250)  
);
```

```
ALTER TABLE POSKYTNUTA_SLUZBA  
ADD PRIMARY KEY (Poskytnuta_sluzba_ID),  
FOREIGN KEY (Merna_jednotka) REFERENCES  
MERNA_JEDNOTKA(Merna_jednotka_ID);
```

```
INSERT INTO POSKYTNUTA_SLUZBA VALUES('Přeprava po  
ČR','3','8.00','Doprava po ČR bez dalších služeb');
```

```
INSERT INTO POSKYTNUTA_SLUZBA VALUES('Čekání','2','3','Cena za minutu  
čekání');
```

```
INSERT INTO POSKYTNUTA_SLUZBA VALUES('Naložení  
nákladu','4','10.00','Cena za naložení jednoho kusu');
```

```
INSERT INTO POSKYTNUTA_SLUZBA VALUES('Vyložení  
nákladu','4','10.00','Cena za vyložení jednoho kus');
```

```
INSERT INTO POSKYTNUTA_SLUZBA VALUES('Výnos do patra','4','20.00','Cena  
za výnos 1 kusu');
```

--7.

```
CREATE TABLE OBLAST (  
Oblast_ID int identity (1,1)not null,  
Nazev varchar(50) not null,  
Poznamka varchar(250)
```

);

```
ALTER TABLE OBLAST  
ADD PRIMARY KEY (Oblast_ID);
```

```
INSERT INTO OBLAST VALUES('BK','Blansko');  
INSERT INTO OBLAST VALUES('BM','Brno-město');  
INSERT INTO OBLAST VALUES('BI','Brno-venkov');  
INSERT INTO OBLAST VALUES('BV','Břeclav');  
INSERT INTO OBLAST VALUES('HO','Hodonín');  
INSERT INTO OBLAST VALUES('VY','Vyškov');  
INSERT INTO OBLAST VALUES('ZN','Znojmo');  
INSERT INTO OBLAST VALUES('SY','Svitavy');
```

--8.

```
CREATE TABLE TYP_ZAKAZNIKA (  
Typ_zakaznika_ID int identity(1,1) not null,  
Nazev varchar(50) not null  
);
```

```
ALTER TABLE TYP_ZAKAZNIKA  
ADD PRIMARY KEY (Typ_zakaznika_ID);
```

```
INSERT INTO TYP_ZAKAZNIKA VALUES('Fyzická osoba');  
INSERT INTO TYP_ZAKAZNIKA VALUES('Právnícká osoba');  
INSERT INTO TYP_ZAKAZNIKA VALUES('Živnostník');
```

--9.

```
CREATE TABLE TYP_VOZIDLA (  
Typ_vozidla_ID int identity(1,1) not null,  
Nazev varchar(50) not null  
);
```

```
ALTER TABLE TYP_VOZIDLA  
ADD PRIMARY KEY (Typ_vozidla_ID);
```

```
INSERT INTO TYP_VOZIDLA VALUES('Osobní automobil');  
INSERT INTO TYP_VOZIDLA VALUES('Nákladní automobil');  
INSERT INTO TYP_VOZIDLA VALUES('Dodávka');  
INSERT INTO TYP_VOZIDLA VALUES('Valník');  
INSERT INTO TYP_VOZIDLA VALUES('Sklopka');  
INSERT INTO TYP_VOZIDLA VALUES('Cisterna');
```

--10.

```
CREATE TABLE PSC (  
Psc_ID int identity(1,1) not null,  
Psc varchar(6) not null,  
Mesto varchar(50) not null,  
Oblast int not null  
);
```

```
ALTER TABLE PSC
```

```
ADD PRIMARY KEY (Psc_ID),  
FOREIGN KEY (Oblast) REFERENCES OBLAST(Oblast_ID);
```

```
INSERT INTO PSC VALUES('67961','Letovice','1');  
INSERT INTO PSC VALUES('68001','Boskovice','1');  
INSERT INTO PSC VALUES('67801','Blansko','1');  
INSERT INTO PSC VALUES('61400','Brno-město','2');  
INSERT INTO PSC VALUES('68201','Vyškov','6');  
INSERT INTO PSC VALUES('66902','Znojmo','7');  
INSERT INTO PSC VALUES('66501','Rosice u Brna','3');  
INSERT INTO PSC VALUES('69002','Břeclav','4');  
INSERT INTO PSC VALUES('69201','Mikulov','4');  
INSERT INTO PSC VALUES('67201','Moravský Krumlov','7');  
INSERT INTO PSC VALUES('69603','Dubňany','5');  
INSERT INTO PSC VALUES('69102','Velké Bílovice','4');  
INSERT INTO PSC VALUES('69642','Vracov','5');  
INSERT INTO PSC VALUES('56802','Svitavy','8');  
INSERT INTO PSC VALUES('57101','Moravská Třebová','8');  
INSERT INTO PSC VALUES('56943','Jevíčko','8');
```

--11.

```
CREATE TABLE ZNACKA (  
Znacka_ID int IDENTITY(1,1) not null,  
Nazev varchar(50) not null,  
Vyrobce varchar(50) not null  
);
```

```
ALTER TABLE ZNACKA
ADD PRIMARY KEY (Znacka_ID);
```

```
INSERT INTO ZNACKA VALUES('Daily IV','IVECO');
INSERT INTO ZNACKA VALUES('Daily V','IVECO');
INSERT INTO ZNACKA VALUES('A31','AVIA');
```

--12.

```
CREATE TABLE DODAVATEL (
Dodavatel_ID int identity(1,1) not null,
Nazev varchar(50) not null,
K_tel1 varchar(20) not null,
K_tel2 varchar(20),
K_mail varchar(50),
K_osoba varchar(100) not null,
K_ulice varchar(50) not null,
K_cp varchar(5) not null,
K_ce varchar(5),
K_psc int not null
);
```

```
ALTER TABLE DODAVATEL
ADD PRIMARY KEY (Dodavatel_ID),
FOREIGN KEY (K_psc) REFERENCES PSC(Psc_ID);
```

```
INSERT INTO DODAVATEL
VALUES('1.SERVIS','606123456','','servis@servis.cz','Josef Servis','Servisní','49','','1');
```

```
INSERT INTO DODAVATEL
VALUES('2.SERVIS','774123456','','servis2@servis.cz','Karel
Servis','Oprávérenská','50','2');
```

```
INSERT INTO DODAVATEL
VALUES('3.SERVIS','728123456','','servisš@servis.cz','Zdeněk
Servis','Oprávrů','51','3');
```

--13.

```
CREATE TABLE PRACOVNIK (
Pracovnik_ID int identity(1,1) not null,
Titul_pred varchar(20),
Jmeno varchar(30) not null,
Rodne_prijmeni varchar(30),
Prijmeni varchar(60) not null,
Titul_za varchar(20),
Pozice int not null,
Datum_narozeni date not null,
Bezuhonost bit not null,
Bezuhonost_doklad varchar(500) not null,
K_ulice varchar(50) not null,
K_cp varchar(5) not null,
K_ce varchar(5),
K_psc int not null,
K_tel1 varchar(20) not null,
K_tel2 varchar(20),
K_mail varchar(50)
);
```

```
ALTER TABLE PRACOVNIK
```

```
ADD PRIMARY KEY (Pracovnik_ID),
FOREIGN KEY (Pozice) REFERENCES Pozice(Pozice_ID),
FOREIGN KEY (K_psc) REFERENCES PSC(Psc_ID);
```

```
INSERT INTO PRACOVNIK VALUES('Ing.','Josef','Šváb','Šváb','1','1976-03-17','1','Zde bude odkaz','Švábí','43','13','775123456','svab@seznam.cz');
```

```
INSERT INTO PRACOVNIK VALUES('','Karel','Moucha','Moucha','1','1956-05-09','1','Zde bude doklad','Muší','47','15','779123456','moucha@seznam.cz');
```

```
INSERT INTO PRACOVNIK VALUES('Mgr.','Cecil','Motýl','Motýl','3','1979-03-27','1','Zde bude doklad','Motýlí','58','12','779654321','motyl@seznam.cz');
```

```
INSERT INTO PRACOVNIK VALUES('Ing.','Zdeněk','Liška','Liška','Ph.D.','2','1959-08-14','0','Zde bude doklad','Liščí','78','11','776654321','liška@seznam.cz');
```

```
INSERT INTO PRACOVNIK VALUES('','Martin','Medojed','Medojed','1','1945-09-06','1','Zde bude doklad','Medojedí','88','10','778654321','medojed@seznam.cz');
```

--14.

```
CREATE TABLE RIDIC_A_OPRAVNENI (
Cislo_dokladu varchar(10) not null,
Pracovnik int not null,
Opravneni int not null,
Platnost date not null
);
```

```
ALTER TABLE RIDIC_A_OPRAVNENI
ADD PRIMARY KEY (Cislo_dokladu, Opravneni),
FOREIGN KEY (Pracovnik) REFERENCES Pracovnik(Pracovnik_ID),
FOREIGN KEY (Opravneni) REFERENCES
RIDICSKE_OPRAVNENI(Opravneni_ID);
```

```
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EA120520','1','1','2025-01-02');
```

```

INSERT INTO RIDIC_A_OPRAVNENI VALUES('EA120520','1','5','2025-01-02');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EA120520','1','8','2025-01-02');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EB123521','2','1','2027-03-02');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EB123521','2','2','2027-03-04');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EB123521','2','5','2027-03-04');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EB123521','2','8','2027-03-04');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EC167213','5','1','2028-08-10');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EC167213','5','2','2028-08-10');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EC167213','5','3','2028-08-10');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EC167213','5','4','2028-08-10');
INSERT INTO RIDIC_A_OPRAVNENI VALUES('EC167213','5','6','2028-08-10');

```

--15.

```

CREATE TABLE PARAMETRY (
Parametr_ID int identity(1,1) not null,
Merna_jednotka int not null,
Nazev varchar(50) not null
);

```

```

ALTER TABLE PARAMETRY
ADD PRIMARY KEY (Parametr_ID),
FOREIGN KEY (Merna_jednotka) REFERENCES
MERNA_JEDNOTKA(Merna_jednotka_ID);

```

```

INSERT INTO PARAMETRY VALUES('6','Hmotnost vozidla');
INSERT INTO PARAMETRY VALUES('6','Nosnost');
INSERT INTO PARAMETRY VALUES('4','Barva');
INSERT INTO PARAMETRY VALUES('4','Počet náprav');
INSERT INTO PARAMETRY VALUES('4','Počet kol');

```

```
INSERT INTO PARAMETRY VALUES('12','Objem prostoru');
INSERT INTO PARAMETRY VALUES('4','Počet sedadel');
INSERT INTO PARAMETRY VALUES('13','Objem nádrže');
INSERT INTO PARAMETRY VALUES('14','maximální rychlost');
INSERT INTO PARAMETRY VALUES('15','Objem motoru');
INSERT INTO PARAMETRY VALUES('16','Spotřeba');
INSERT INTO PARAMETRY VALUES('4','Koně');
INSERT INTO PARAMETRY VALUES('17','Kilowaty');
```

--16.

```
CREATE TABLE VOZIDLO (
Vozidlo_ID varchar(17) not null,
SPZ varchar(7) not null,
Znacka int not null,
Typ_vozidla int not null,
Zodpovedna_osoba int not null
);
```

```
ALTER TABLE VOZIDLO
ADD PRIMARY KEY (Vozidlo_ID),
FOREIGN KEY (Znacka) REFERENCES ZNACKA(Znacka_ID),
FOREIGN KEY (Typ_vozidla) REFERENCES TYP_VOZIDLA(Typ_vozidla_ID),
FOREIGN KEY (Zodpovedna_osoba) REFERENCES Pracovnik(Pracovnik_ID);
```

```
INSERT INTO VOZIDLO VALUES('TMBDA41Z7C2102777','1B17890','1','3','4');
INSERT INTO VOZIDLO VALUES('WF0GXXGBBG7Y42388','1BU7654','2','3','4');
INSERT INTO VOZIDLO VALUES('TMADB51CAAJ087960','3BA9876','3','4','4');
```

--17.

```
CREATE TABLE ZAKAZNIK (  
Zakaznik_ID int not null identity(1,1),  
Typ_zakaznika int not null,  
Nazev varchar(250) not null,  
F_ulice varchar(50) not null,  
F_cp varchar(5) not null,  
F_ce varchar(5),  
F_psc int not null,  
K_osoba varchar(100) not null,  
K_tel1 varchar(20) not null,  
K_tel2 varchar(20),  
K_mail varchar(50),  
Poznamka varchar(250)  
);
```

```
ALTER TABLE ZAKAZNIK
```

```
ADD PRIMARY KEY (Zakaznik_ID),
```

```
FOREIGN KEY (Typ_zakaznika) REFERENCES  
TYP_ZAKAZNIKA(Typ_zakaznika_ID),
```

```
FOREIGN KEY (F_psc) REFERENCES PSC(Psc_ID);
```

```
INSERT INTO ZAKAZNIK VALUES('1','Karel Ježek','Ježčí','56','12','Karel  
Ježek','327899899','jezek@seznam.cz','Poznámka');
```

```
INSERT INTO ZAKAZNIK VALUES('1','Vlastimil  
Motyka','Zemědělská','12','5','Vlastimil  
Motyka','608Z776778','motyka@seznam.cz','Poznámka');
```

```
INSERT INTO ZAKAZNIK VALUES('2','Babočka,s.r.o','Můří','19','7','Antonín  
Babočka','776554334','babocka@babocka.cz','Poznámka');
```

```
INSERT INTO ZAKAZNIK VALUES('2','Hejlík, a.s.','Ptačí','32','','2','Jaromír  
Hejlík','609786543','','hejlik@hejlik.cz','Poznámka');
```

```
INSERT INTO ZAKAZNIK VALUES('3','Helena Rýčová,  
advokátka','Polnohospodářská','69','','3','Helena  
Rýčová','775671295','','rycova@seznam.cz','Poznámka');
```

--18.

```
CREATE TABLE ZAKAZKA (  
Zakazka_ID int identity(1,1) not null,  
Zakaznik int not null,  
pracovnik int not null,  
Vozidlo varchar(17) not null,  
Datum_cas datetime not null,  
N_ulice varchar(50) not null,  
N_cp varchar(5) not null,  
N_ce varchar(5),  
N_psc int not null,  
V_ulice varchar(50) not null,  
V_cp varchar(5) not null,  
V_ce varchar(5),  
V_psc int not null,  
Cena_bez_DPH decimal(12,2) not null,  
DPH decimal(12,2) not null,  
Cena_s_DPH decimal(12,2) not null,  
Dokoncena bit not null,  
Zuctovana bit not null  
);
```

```

ALTER TABLE ZAKAZKA
ADD PRIMARY KEY (Zakazka_ID),
FOREIGN KEY (Zakaznik) REFERENCES ZAKAZNIK(Zakaznik_ID),
FOREIGN KEY (Pracovnik) REFERENCES Pracovnik(Pracovnik_ID),
FOREIGN KEY (Vozidlo) REFERENCES VOZIDLO(Vozidlo_ID),
FOREIGN KEY (N_psc) REFERENCES PSC(Psc_ID),
FOREIGN KEY (V_psc) REFERENCES PSC(Psc_ID);

INSERT INTO ZAKAZKA VALUES('1','1','WF0GXXGBBG7Y42388','2018-07-01','Absolonova','23','','1','Veletřní','456','11','11','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('1','2','TMADB51CAAJ087960','2018-07-03','Arménská','41','','2','Vaníčkova','7654','23','12','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('2','5','TMBDA41Z7C2102777','2018-07-09','Barvířská','872','23','3','Urxova','654','43','13','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('2','1','WF0GXXGBBG7Y42388','2018-08-06','Bezejmenná','987','43','4','U Lesíčka','432','27','14','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('3','2','TMADB51CAAJ087960','2018-08-04','Žlutý kopec','389','25','5','U Koupaliště','986','87','15','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('3','5','TMBDA41Z7C2102777','2018-07-02','Zubatého','290','34','6','U Hájovny','765','23','16','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('4','1','WF0GXXGBBG7Y42388','2018-07-15','Zákoutí','29','','7','Tovární','753','34','1','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('4','2','TMADB51CAAJ087960','2018-08-24','Wintrova','2901','12','8','Štompil','8976','32','2','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('5','5','TMBDA41Z7C2102777','2018-07-30','Vyhlídka','290','32','9','Škroupova','657','12','3','0','0','0','0','0');
INSERT INTO ZAKAZKA VALUES('5','5','WF0GXXGBBG7Y42388','2018-08-31','Vlhká','876','54','10','Šafaříkova','568','13','4','0','0','0','0','0');

```

--19.

```

CREATE TABLE RADKY_ZAKAZKY (
Radky_zakazky_ID int identity(1,1) not null,

```

Zakazka int not null,
Sluzba int not null,
Pocet_m_jednotek_na_radku int not null,
Cena_na_radku decimal(12,2) not null,
DPH_na_radku decimal(12,2) not null,
Cena_DPH_radek decimal(12,2) not null
);

ALTER TABLE RADKY_ZAKAZKY
ADD PRIMARY KEY (Radky_zakazky_ID),
FOREIGN KEY (Zakazka) REFERENCES ZAKAZKA(Zakazka_ID),
FOREIGN KEY (Sluzba) REFERENCES
POSKYTNUTA_SLUZBA(Poskytnuta_sluzba_ID);

INSERT INTO RADKY_ZAKAZKY VALUES('1','1','132','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('1','2','11','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('1','5','4','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('2','1','76','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('2','2','15','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('3','1','89','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('3','2','17','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('3','3','4','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('3','4','4','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('4','1','54','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('4','3','10','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('4','4','10','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('5','1','38','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('5','2','19','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('5','3','7','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('5','4','7','0','0','0');

```

INSERT INTO RADKY_ZAKAZKY VALUES('5','5','9','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('6','1','209','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('6','2','47','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('7','1','86','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('7','3','17','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('7','4','17','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('7','5','8','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('8','1','96','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('8','2','35','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('8','3','37','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('8','4','37','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('9','1','12','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('9','2','25','0','0','0');
INSERT INTO RADKY_ZAKAZKY VALUES('10','1','178','0','0','0');

```

--20.

```

CREATE TABLE PLANOVANY_SERVIS (
Planovany_servis_ID int identity(1,1) not null,
Vozidlo varchar(17) not null,
Datum datetime not null,
Typ_operace int not null,
Dodavatel int not null,
Zodpovedna_osoba int not null
);

```

```

ALTER TABLE PLANOVANY_SERVIS
ADD PRIMARY KEY (Planovany_servis_ID),
FOREIGN KEY (Vozidlo) REFERENCES VOZIDLO(Vozidlo_ID),
FOREIGN KEY (Typ_operace) REFERENCES
TYP_SERVISNI_OPERACE(Typ_servisni_operace_ID),

```

FOREIGN KEY (Dodavatel) REFERENCES DODAVATEL(Dodavatel_ID),
FOREIGN KEY (Zodpovedna_osoba) REFERENCES Pracovnik(Pracovnik_ID);

INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777','2018-07-24','1','1','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777','2018-08-11','2','2','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777','2018-07-15','3','3','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('TMBDA41Z7C2102777','2018-07-09','4','1','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('WF0GXXGBBG7Y42388','2018-09-04','1','2','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('WF0GXXGBBG7Y42388','2018-09-03','2','3','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('WF0GXXGBBG7Y42388','2018-08-02','3','1','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('WF0GXXGBBG7Y42388','2018-07-22','4','2','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960','2018-07-15','1','3','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960','2018-08-26','2','1','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960','2018-07-22','3','2','4');

INSERT INTO PLANOVANY_SERVIS VALUES ('TMADB51CAAJ087960','2018-08-11','4','3','4');

--21.

CREATE TABLE PARAMETRY_VOZIDLA (
Parametry_vozidla_ID int not null identity(1,1),

```
Vozidlo varchar(17) not null,  
Parametr int not null,  
Hodnota_parametru varchar(50) not null  
);
```

```
ALTER TABLE PARAMETRY_VOZIDLA  
ADD PRIMARY KEY (Parametry_vozidla_ID),  
FOREIGN KEY (Vozidlo) REFERENCES VOZIDLO(Vozidlo_ID),  
FOREIGN KEY (Parametr) REFERENCES PARAMETRY(Parametr_ID);
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('TMBDA41Z7C2102777','1','2078');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('WF0GXXGBBG7Y42388','1','1786');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('TMADB51CAAJ087960','1','1300');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('TMBDA41Z7C2102777','2','950');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('WF0GXXGBBG7Y42388','2','1100');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('TMADB51CAAJ087960','2','750');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('TMBDA41Z7C2102777','3','BÍLÁ');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('WF0GXXGBBG7Y42388','3','BÍLÁ');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('TMADB51CAAJ087960','3','MODRÁ');
```

```
INSERT INTO PARAMETRY_VOZIDLA VALUES('TMBDA41Z7C2102777','4','2');
```

```
INSERT INTO PARAMETRY_VOZIDLA  
VALUES('WF0GXXGBBG7Y42388','4','2');
```

INSERT INTO PARAMETRY_VOZIDLA VALUES('TMADB51CAAJ087960','4','2');
INSERT INTO PARAMETRY_VOZIDLA VALUES('TMBDA41Z7C2102777','5','4');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','5','4');
INSERT INTO PARAMETRY_VOZIDLA VALUES('TMADB51CAAJ087960','5','6');
INSERT INTO PARAMETRY_VOZIDLA VALUES('TMBDA41Z7C2102777','6','8');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','6','10');
INSERT INTO PARAMETRY_VOZIDLA VALUES('TMADB51CAAJ087960','6','5');
INSERT INTO PARAMETRY_VOZIDLA VALUES('TMBDA41Z7C2102777','7','3');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','7','3');
INSERT INTO PARAMETRY_VOZIDLA VALUES('TMADB51CAAJ087960','7','3');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMBDA41Z7C2102777','8','80');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','8','85');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMADB51CAAJ087960','8','60');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMBDA41Z7C2102777','9','150');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','9','153');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMADB51CAAJ087960','9','123');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMBDA41Z7C2102777','10','2500');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','10','2500');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMADB51CAAJ087960','10','1900');
INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMBDA41Z7C2102777','11','9');

```
INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','11','10');

INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMADB51CAAJ087960','11','15');

INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMBDA41Z7C2102777','12','100');

INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','12','100');

INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMADB51CAAJ087960','12','70');

INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMBDA41Z7C2102777','13','70');

INSERT INTO PARAMETRY_VOZIDLA
VALUES('WF0GXXGBBG7Y42388','13','70');

INSERT INTO PARAMETRY_VOZIDLA
VALUES('TMADB51CAAJ087960','13','50');
```

--22.

```
CREATE TABLE SERVIS_INFO (
Servis_info_ID int identity(1,1) not null,
Zodpovedna_osoba int not null,
Dodavatel int not null,
Typ int not null,
Cena decimal(12,2) not null,
Vozidlo varchar(17) not null,
Datum date not null,
Poznamka varchar(250),
Odkaz_doklad varchar(500) not null,
Zauctovano bit not null
);
```

```
ALTER TABLE SERVIS_INFO
ADD PRIMARY KEY (Servis_info_ID),
FOREIGN KEY (Zodpovedna_osoba) REFERENCES Pracovnik(Pracovnik_ID),
FOREIGN KEY (Dodavatel) REFERENCES DODAVATEL(Dodavatel_ID),
FOREIGN KEY (Typ) REFERENCES
TYP_SERVISNI_OPERACE(Typ_servisni_operace_ID),
FOREIGN KEY (Vozidlo) REFERENCES VOZIDLO(Vozidlo_ID);
```

```
INSERT INTO SERVIS_INFO
VALUES('4','1','1','1000','TMBDA41Z7C2102777','2018-03-12','Bez poznámky','odkaz
na složku','1');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','2','2','3500','WF0GXXGBBG7Y42388','2018-04-11','Bez
poznámky','odkaz na složku','1');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','3','3','7600','TMADB51CAAJ087960','2018-03-07','Bez poznámky','odkaz
na složku','1');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','1','4','3250','TMBDA41Z7C2102777','2018-03-26','Bez poznámky','odkaz
na složku','1');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','2','1','1765','WF0GXXGBBG7Y42388','2018-04-22','Bez
poznámky','odkaz na složku','0');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','3','2','2435','TMADB51CAAJ087960','2018-03-17','Bez poznámky','odkaz
na složku','0');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','1','3','3902','TMBDA41Z7C2102777','2018-04-21','Bez poznámky','odkaz
na složku','0');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','2','4','7698','WF0GXXGBBG7Y42388','2018-02-19','Bez
poznámky','odkaz na složku','0');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','3','1','9087','TMADB51CAAJ087960','2018-03-18','Bez poznámky','odkaz
na složku','0');
```

```
INSERT INTO SERVIS_INFO
VALUES('4','2','4','1200','TMBDA41Z7C2102777','2018-04-02','Bez poznámky','odkaz
na složku','0');
```

--23.

```
CREATE TABLE DOKLADY_VOZIDLO (
Doklady_vozidlo_ID int not null identity(1,1),
Vozidlo varchar(17) not null,
Typ_dokladu int not null,
Odkaz_na_doklad varchar(500) not null,
Platnost_od date not null,
Platnost_do date,
Poznamka varchar(250),
Nalezitosti bit not null
);
```

```
ALTER TABLE DOKLADY_VOZIDLO
ADD PRIMARY KEY (Doklady_vozidlo_ID),
FOREIGN KEY (Vozidlo) REFERENCES VOZIDLO(Vozidlo_ID),
FOREIGN KEY (Typ_dokladu) REFERENCES TYP_DOKLADU(Typ_dokladu_ID);
```

```
INSERT INTO DOKLADY_VOZIDLO VALUES ('TMBDA41Z7C2102777','1','Zde
bude odkaz','2017-01-01','','Bez problémů','1');
```

```
INSERT INTO DOKLADY_VOZIDLO VALUES ('WF0GXXGBBG7Y42388','1','Zde
bude odkaz','2017-01-02','','Bez problémů','1');
```

```
INSERT INTO DOKLADY_VOZIDLO VALUES ('TMADB51CAAJ087960','1','Zde
bude odkaz','2017-01-03','','Bez problémů','1');
```

```

INSERT INTO DOKLADY_VOZIDLO VALUES ('TMBDA41Z7C2102777','2','Zde
bude odkaz','2017-01-01','','Bez problémů','1');

INSERT INTO DOKLADY_VOZIDLO VALUES ('WF0GXXGBBG7Y42388','2','Zde
bude odkaz','2017-01-02','','Bez problémů','1');

INSERT INTO DOKLADY_VOZIDLO VALUES ('TMADB51CAAJ087960','2','Zde
bude odkaz','2017-01-03','','Bez problémů','1');

INSERT INTO DOKLADY_VOZIDLO VALUES ('TMBDA41Z7C2102777','3','Zde
bude odkaz','2018-02-01','2019-02-01','Bez problémů','1');

INSERT INTO DOKLADY_VOZIDLO VALUES ('WF0GXXGBBG7Y42388','3','Zde
bude odkaz','2018-02-02','2019-02-02','Bez problémů','1');

INSERT INTO DOKLADY_VOZIDLO VALUES ('TMADB51CAAJ087960','3','Zde
bude odkaz','2018-02-03','2019-02-03','Bez problémů','1');

INSERT INTO DOKLADY_VOZIDLO VALUES ('TMBDA41Z7C2102777','4','Zde
bude odkaz','2017-01-01','2019-01-01','Bez problémů','1');

INSERT INTO DOKLADY_VOZIDLO VALUES ('WF0GXXGBBG7Y42388','4','Zde
bude odkaz','2017-01-02','2019-01-02','Bez problémů','1');

INSERT INTO DOKLADY_VOZIDLO VALUES ('TMADB51CAAJ087960','4','Zde
bude odkaz','2017-01-03','2019-01-03','Bez problémů','1');

```

--24.

```

CREATE TABLE KNIHA_JIZD (
Kniha_jizd_ID int identity(1,1) not null,
Vozidlo varchar(17) not null,

Stav_na_zacatku int not null,
Stav_na_konci int not null,
Cas_zacatku datetime,
Cas_konce datetime,
Tankovani bit not null,
Pocet_l Real,
Cena decimal(12,2)

```

);

```
ALTER TABLE KNIHA_JIZD
ADD PRIMARY KEY (Kniha_jizd_ID);
ALTER TABLE KNIHA_JIZD ADD Zakazka INT null;
ALTER TABLE KNIHA_JIZD ADD Servis INT null;
ALTER TABLE KNIHA_JIZD ADD CONSTRAINT FK_VOZIDLO FOREIGN KEY
(Vozidlo) REFERENCES VOZIDLO(Vozidlo_ID);
ALTER TABLE KNIHA_JIZD ADD CONSTRAINT FK_ZAKAZKA FOREIGN
KEY (Zakazka) REFERENCES ZAKAZKA(Zakazka_ID);
ALTER TABLE KNIHA_JIZD ADD CONSTRAINT FK_SERVIS FOREIGN KEY
(Servis) REFERENCES SERVIS_INFO(Servis_info_ID);
```

```
INSERT INTO KNIHA_JIZD
VALUES('TMBDA41Z7C2102777','19789','20789','2018-05-04 11:00:00','2018-05-04
12:00:00','1','10','300','1',NULL);
```

```
INSERT INTO KNIHA_JIZD
VALUES('WF0GXXGBBG7Y42388','9876','10234','2018-05-05 11:00:00','2018-05-05
16:00:00','1','11','330','2',NULL);
```

```
INSERT INTO KNIHA_JIZD
VALUES('TMADB51CAAJ087960','28938','29305','2018-05-06 11:00:00','2018-05-06
17:00:00','1','12','360','3',NULL);
```

```
INSERT INTO KNIHA_JIZD
VALUES('TMBDA41Z7C2102777','20789','21908','2018-05-07 11:00:00','2018-05-07
12:00:00','1','13','390','4',NULL);
```

```
INSERT INTO KNIHA_JIZD
VALUES('WF0GXXGBBG7Y42388','10234','10566','2018-05-08 11:00:00','2018-05-
08 15:00:00','1','14','420','5',NULL);
```

```
INSERT INTO KNIHA_JIZD
VALUES('TMADB51CAAJ087960','29305','31293','2018-05-09 11:00:02','2018-05-09
16:00:02','1','15','450','6',NULL);
```

```

INSERT INTO KNIHA_JIZD
VALUES('TMBDA41Z7C2102777','21908','22123','2018-05-10 11:00:02','2018-05-10
12:00:02','1','16','480','7',NULL);

INSERT INTO KNIHA_JIZD
VALUES('WF0GXXGBBG7Y42388','10566','11234','2018-05-11 11:00:00','2018-05-
11 20:00:00','1','17','510','8',NULL);

INSERT INTO KNIHA_JIZD
VALUES('TMADB51CAAJ087960','31293','32102','2018-05-12 11:00:00','2018-05-12
21:00:00','1','18','540','9',NULL);

INSERT INTO KNIHA_JIZD
VALUES('TMBDA41Z7C2102777','22123','23222','2018-05-13 11:00:00','2018-05-13
22:00:00','1','19','570','10',NULL);

INSERT INTO KNIHA_JIZD
VALUES('WF0GXXGBBG7Y42388','11234','12345','2018-05-14 11:00:00','2018-05-
14 00:00:00','1','20','600',NULL,'1');

INSERT INTO KNIHA_JIZD
VALUES('TMADB51CAAJ087960','32102','32123','2018-05-15 11:00:00','2018-05-15
15:00:00','1','21','630',NULL,'2');

INSERT INTO KNIHA_JIZD
VALUES('TMBDA41Z7C2102777','32123','32345','2018-05-16 11:00:00','2018-05-16
17:00:00','1','22','660',NULL,'3');

INSERT INTO KNIHA_JIZD
VALUES('WF0GXXGBBG7Y42388','23222','24536','2018-05-17 11:00:00','2018-05-
17 17:00:00','1','23','690',NULL,'4');

INSERT INTO KNIHA_JIZD
VALUES('TMADB51CAAJ087960','12345','12987','2018-05-18 11:00:00','2018-05-18
18:00:00','1','24','720',NULL,'5');

INSERT INTO KNIHA_JIZD
VALUES('TMBDA41Z7C2102777','32345','32475','2018-05-20 11:00:00','2018-05-20
15:00:00','1','25','750',NULL,'6');

INSERT INTO KNIHA_JIZD
VALUES('WF0GXXGBBG7Y42388','24536','23456','2018-05-21 11:00:00','2018-05-
21 13:00:00','1','26','780',NULL,'7');

INSERT INTO KNIHA_JIZD
VALUES('TMADB51CAAJ087960','12987','13234','2018-05-22 11:00:00','2018-05-22
19:00:00','1','27','810',NULL,'8');

```

```
INSERT INTO KNIHA_JIZD
VALUES('TMBDA41Z7C2102777','32475','33657','2018-05-23 11:00:00','2018-05-23
23:00:00','1','28','840',NULL,'9');
```

```
INSERT INTO KNIHA_JIZD
VALUES('WF0GXXGBBG7Y42388','23456','24553','2018-05-24 11:00:00','2018-05-
24 20:00:00','1','29','870',NULL,'10');
```

Příloha č. 2: Trigger pro nastavení cen (Zdroj: vlastní zpracování)

```
DROP TRIGGER NASTAVENI_CENY
CREATE TRIGGER NASTAVENI_CENY ON RADKY_ZAKAZKY
AFTER INSERT
AS
BEGIN
    DECLARE @J_CENA AS DECIMAL(12,2)
    DECLARE @J_CENA_DPH AS DECIMAL (12,2)
    DECLARE @J_DPH AS DECIMAL (12,2)
    DECLARE @CENA AS DECIMAL(12,2)
    DECLARE @CENA_DPH AS DECIMAL(12,2)
    DECLARE @DPH AS DECIMAL(12,2)
    DECLARE @SLUZBA AS INT
    DECLARE @MNOZSTVI AS INT
    DECLARE @RADEK AS INT
    DECLARE @POMOCNA AS DECIMAL(12,2)
    DECLARE @ZAKAZKA AS INT

    SELECT Radky_zakazky_ID FROM inserted
    SET @ZAKAZKA = (SELECT ZAKAZKA FROM inserted)
    SET @RADEK = (SELECT Radky_zakazky_ID FROM inserted)
    SELECT Pocet_m_jednotek_na_radku FROM inserted
    SET @MNOZSTVI = (SELECT Pocet_m_jednotek_na_radku FROM inserted)
    SELECT Sluzba FROM inserted
    SET @SLUZBA = (SELECT Sluzba FROM inserted)

    SELECT Cena FROM POSKYTNUTA_SLUZBA WHERE
    POSKYTNUTA_SLUZBA.Poskytnuta_sluzba_ID = @SLUZBA

    SET @J_CENA_DPH = (SELECT Cena FROM POSKYTNUTA_SLUZBA WHERE
    POSKYTNUTA_SLUZBA.Poskytnuta_sluzba_ID = @SLUZBA)
```

```
SET @J_CENA = 0.79 * @J_CENA_DPH
SET @J_DPH = 0.21 * @J_CENA_DPH
SET @CENA = @MNOZSTVI * @J_CENA
SET @DPH = @MNOZSTVI * @J_DPH
SET @CENA_DPH = @MNOZSTVI * @J_CENA_DPH
```

```
UPDATE RADKY_ZAKAZKY
```

```
SET Cena_DPH_radek = @CENA_DPH, Cena_na_radku = @CENA, DPH_na_radku
= @DPH
```

```
WHERE Radky_zakazky_ID = @RADEK;
```

```
SET @CENA = 0
```

```
SET @CENA_DPH = 0
```

```
SET @DPH = 0
```

```
DECLARE AKTUALIZUJ_HLAVICKU CURSOR FOR
```

```
SELECT Cena_DPH_radek FROM RADKY_ZAKAZKY WHERE
Zakazka=@ZAKAZKA
```

```
OPEN AKTUALIZUJ_HLAVICKU
```

```
FETCH AKTUALIZUJ_HLAVICKU INTO @POMOCNA
```

```
WHILE @@FETCH_STATUS=0 BEGIN
```

```
SET @CENA_DPH = @CENA_DPH + @POMOCNA
```

```
FETCH NEXT FROM AKTUALIZUJ_HLAVICKU INTO @POMOCNA
```

```
END
```

```
CLOSE AKTUALIZUJ_HLAVICKU
```

```
DEALLOCATE AKTUALIZUJ_HLAVICKU
```

```
DECLARE AKTUALIZUJ_HLAVICKU2 CURSOR FOR
SELECT Cena_na_radku FROM RADKY_ZAKAZKY WHERE
Zakazka=@ZAKAZKA
```

```
OPEN AKTUALIZUJ_HLAVICKU2
FETCH AKTUALIZUJ_HLAVICKU2 INTO @POMOCNA
WHILE @@FETCH_STATUS=0 BEGIN
SET @CENA = @CENA + @POMOCNA
FETCH NEXT FROM AKTUALIZUJ_HLAVICKU2 INTO @POMOCNA
END
CLOSE AKTUALIZUJ_HLAVICKU2
DEALLOCATE AKTUALIZUJ_HLAVICKU2
```

```
DECLARE AKTUALIZUJ_HLAVICKU3 CURSOR FOR
SELECT DPH_na_radku FROM RADKY_ZAKAZKY WHERE Zakazka =
@ZAKAZKA
```

```
OPEN AKTUALIZUJ_HLAVICKU3
FETCH AKTUALIZUJ_HLAVICKU3 INTO @POMOCNA
WHILE @@FETCH_STATUS=0 BEGIN
SET @DPH = @DPH + @POMOCNA
FETCH NEXT FROM AKTUALIZUJ_HLAVICKU3 INTO @POMOCNA
END
CLOSE AKTUALIZUJ_HLAVICKU3
DEALLOCATE AKTUALIZUJ_HLAVICKU3
```

```
UPDATE ZAKAZKA
SET Cena_s_DPH = @CENA_DPH, Cena_bez_DPH = @CENA, DPH = @DPH
WHERE Zakazka_ID = @ZAKAZKA;
```

```
SELECT * FROM RADKY_ZAKAZKY WHERE Zakazka = @ZAKAZKA
SELECT Zakazka_ID, CENA_bez_DPH, DPH, Cena_s_DPH
FROM ZAKAZKA WHERE Zakazka_ID=@ZAKAZKA
END
INSERT INTO RADKY_ZAKAZKY VALUES('2','1','132','0','0','0');
```