



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

DATABÁZOVÝ SYSTÉM PRO ZÁVODNÍ JÍDELNU

DATABASE SYSTEM FOR THE CANTEEN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Lukáš Pokorný

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Luhan, Ph.D., MSc

BRNO 2022

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Lukáš Pokorný**
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**
Akademický rok: 2021/22
Studijní program: Systémové inženýrství a informatika

Garant studijního oboru 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:

Databázový systém pro závodní jídelnu

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 navrhnout a následně vytvořit databázi pro firmu z oblasti gastronomie pro efektivní práci s daty a podporu vybraných procesů.

Základní literární prameny:

BASL, J. a R. BLAŽÍČEK. Podnikové informační systémy - podnik v informační společnosti. 3. vyd. Praha: Grada Publishing a.s., 2012. 328 s. ISBN 978-80-247-4307-3.

CONOLLY, T., C. E. BEGG a R. HOLOWCZAK. Mistrovstvídatabáze: profesionální průvodce tvorbou efektivních databází. 1. vyd. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

KOCH, M. a B. NEUWIRTH. Datové a funkční modelování. 4. rozš. vyd. Brno: Akademické nakladatelství CERM, 2010. 139 s. ISBN 978-80-214-4125-5.

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

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2021/22

V Brně dne 28.2.2022

L. S.

doc. Ing. Miloš Koch, CSc.
garant

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Tato bakalářská práce se zabývá návrhem databázového systému pro jídelnu. Databáze bude umožňovat efektivní ukládání dat o objednávkách, jídlech a uživatelích. Databáze bude realizovaná v MySQL a jako vývojové rozhraní bude použito MySQL Workbench.

Klíčová slova

databázový systém, databáze, DBMS, ER-diagram, relace, SQL

Abstract

This bachelor thesis focuses on designing a database system for a canteen. The database will allow efficient storage of data of orders, meals and users. The database will be implemented in MySQL and MySQL Workbench will be used as a development interface.

Key words

database system, database, DBMS, ER-diagram, relation, SQL

Bibliografická citace

POKORNÝ, Lukáš. Databázový systém pro závodní jídelnu [online]. Brno, 2022 [cit. 2022-05-09]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/142303>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jan Luhan.

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 15.5.2022

Poděkování

Tímto bych chtěl poděkovat panu Ing. Janovi Luhanovi, Ph.D., MSc za odborné vedení bakalářské práce, jeho rady i čas. Taktéž bych rád poděkoval oponentovi práce Ing. Bernardovi Neuwirthovi, Ph.D., MSc, za věnovaný čas k ohodnocení práce. Nakonec bych chtěl poděkovat své rodině a známým za podporu při zpracovávání bakalářské práce.

Obsah

ÚVOD	8
CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ	9
1 TEORETICKÁ VÝCHODISKA PRÁCE	10
1.1 ZÁKLADNÍ POJMY	10
1.1.1 <i>Data</i>	10
1.1.2 <i>Informace</i>	10
1.2 DATABÁZOVÝ SYSTÉM	11
1.2.1 <i>Databáze</i>	12
1.2.2 <i>Systém řízení báze dat (DBMS)</i>	12
1.2.3 <i>Databázové aplikace</i>	13
1.2.4 <i>Uživatel</i>	14
1.3 DATOVÉ MODELY	14
1.3.1 <i>Lineární model</i>	14
1.3.2 <i>Relační model</i>	15
1.3.3 <i>Objektový model</i>	15
1.4 RELAČNÍ DATOVÝ MODEL	16
1.4.1 <i>Relace</i>	16
1.4.2 <i>Integrita relačního modelu</i>	16
1.5 NORMALIZACE	18
1.5.1 <i>První normální forma</i>	19
1.5.2 <i>Druhá normální forma</i>	19
1.5.3 <i>Třetí normální forma</i>	19
1.6 NÁVRH DATABÁZE	19
1.6.1 <i>Konceptuální návrh databáze</i>	20
1.6.2 <i>Logický návrh databáze</i>	20
1.6.3 <i>Fyzická návrh databáze</i>	20
1.7 JAZYK SQL	20
1.7.1 <i>Kategorie příkazů SQL</i>	20
1.8 SWOT ANALÝZA	21
2 ANALÝZA SOUČASNÉHO STAVU	23
2.1 ZÁKLADNÍ INFORMACE O SPOLEČNOSTI	23
2.2 ORGANIZAČNÍ STRUKTURA	23
2.3 POPIS SOUČASNÉ SITUACE	24
2.4 STRÁVNÍCI	24
2.5 PROCES SHÁNĚNÍ NOVÝCH KLIENTŮ	25
2.6 PROCES ODBĚRU JÍDLA	25
2.7 VYBAVENÍ JÍDELNY	25
2.8 HARDWAROVÉ VYBAVENÍ	25
2.9 SOFTWAREOVÉ VYBAVENÍ	26
2.10 SWOT ANALÝZA	26
2.11 SILNÉ STRÁNKY	26
2.12 SLABÉ STRÁNKY	26
2.13 PŘÍLEŽITOSTI	26
2.14 HROZBY	27
3 VLASTNÍ NÁVRH ŘEŠENÍ	28
3.1 FUNKCIONALITA	28
3.1.1 <i>Proces objednávky</i>	28

3.1.2	<i>Proces rezervace</i>	30
3.2	KONCEPTUÁLNÍ NÁVRH	32
3.2.1	<i>Identifikace entit</i>	32
3.2.2	<i>Identifikace relací</i>	34
3.3	LOGICKÝ NÁVRH	35
3.3.1	<i>Relační tabulky vztahující se k osobám</i>	35
3.3.2	<i>Relační tabulky vztahující se k objednávkám</i>	42
3.3.3	<i>Relační tabulky popisující jídlo</i>	44
3.3.4	<i>Relační tabulky vztahující se k rezervaci</i>	47
3.3.5	<i>ER – diagram</i>	48
3.4	FYZICKÝ NÁVRH	49
3.4.1	<i>Tvorba tabulek</i>	49
3.4.2	<i>Tvorba cizích klíčů a vazeb mezi tabulkami</i>	50
3.4.3	<i>Tvorba ostatních atributů</i>	51
3.4.4	<i>Pohledy</i>	52
3.5	ZHODNOCENÍ NÁVRHU A PŘÍNOS PRO FIRMU	54
	ZÁVĚR	55
	SEZNAM POUŽITÝCH ZDROJŮ	56
	SEZNAM POUŽITÝCH ZKRATEK	57
	SEZNAM POUŽITÝCH OBRÁZKŮ	58
	SEZNAM POUŽITÝCH TABULEK	59
	SEZNAM PŘÍLOH	60

ÚVOD

V dnešní době si jen málokdo dokáže představit život bez moderních technologií. Lidé každý den čerpají a zveřejňují nejrůznější data a je to nedílnou součástí jejich života. Tyto technologie nám pomohly značně zvýšit kvalitu každodenního života. Samozřejmě s daty nepracují jen lidé v běžném životě, ale hlavně firmy různých velikostí a zaměření.

Ukládání a zpracování dat do elektronických databází přináší firmám mnoho výhod. Databáze umožňuje efektivnější a pohodlnější práci s daty. Spoří čas zaměstnancům i zákazníkům a tím firma dokáže ušetřit nemalé peníze. Čím větší je objem dat, které firma zpracovává a užitkuje, tím větší je efektivita databáze. To ale neznamená, že pro firmy s malým objemem dat není databáze vhodné řešení. Správně vytvořená databáze umožňuje evidovat tržby, informace o zákaznících, zboží nebo dodavatelích.

Je na každé firmě, jestli se rozhodne databázi využívat. Databáze je unikátní pro každou firmu a její potřeby a je tedy potřeba ji řešit individuálně. Na trhu je v dnešní době mnoho databázových systémů různé cenové relace. Je tedy i důležité zvolit ten systém, který vyhovuje požadavkům dané firmy.

Tato práce se bude zabývat návrhem databáze pro konkrétní společnost a bude navržena tak, aby vyhovovala daným požadavkům firmy.

Cíle práce, metody a postupy zpracování

Cílem této práce je navrhnout databázi pro závodní jídelnu tak, aby byla firma schopná ukládat a využívat data, které potřebuje ke správnému chodu. Je tedy v první řadě potřeba analyzovat, jaká data jsou pro firmu relevantní a jakým způsobem chce s těmito daty dále pracovat. Po této analýze přijde na řadu samotný návrh databáze podle daných kritérií.

Dílčí cíle práce:

- Zpracování teoretického východiska práce
- Zpracování analýzy současného stavu
- Navržení vlastního řešení databáze

V první části práce se budeme zabývat vysvětlením teorie dané problematiky. Tato teorie bude potřebná pro pochopení současného stavu firmy, a hlavně k vytvoření správné a efektivní databáze. Teorie bude tedy sloužit jako podklad pro celou práci.

Ve druhé části práce analyzujeme současný stav firmy. Analyzujeme fungování firmy, její procesy a logiku práce. Tyto informace budou sloužit pro návrh funkční a efektivní databáze.

V poslední třetí části práce se zaměříme na návrh databáze, při které budeme využívat poznatků z prvních dvou částí.

Navržení databáze se bude skládat ze tří částí:

- Konceptuální návrh
- Logický návrh
- Fyzický návrh

V konceptuálním návrhu analyzujeme jednotlivé entity a relace, díky kterým vytvoříme entity-relationship diagram (ER diagram)

V logickém návrhu využijeme ER diagramu a vytvoříme správnou strukturu tabulek.

Ve fyzickém návrhu již vytvoříme samotnou databázi. Databáze bude obsahovat tabulky a pohledy.

1 TEORETICKÁ VÝCHODISKA PRÁCE

Tato kapitola se bude zabývat teoretickou částí práce, která tvoří základ pro pochopení požadavků dané společnosti a následný návrh databáze. Vysvětlíme si zde základní pojmy v oblasti databázových systémů, podíváme se na jednotlivé datové modely, normalizaci a v neposlední řadě také návrh databáze, jazyk SQL a jednotlivé analýzy. Tyto informace poté aplikujeme v návrhu vlastního řešení.

1.1 Základní pojmy

Důležitým předpokladem pro porozumění problematice datového modelování, je znát základní pojmy jako data, informace a znalosti (3, s. 4).

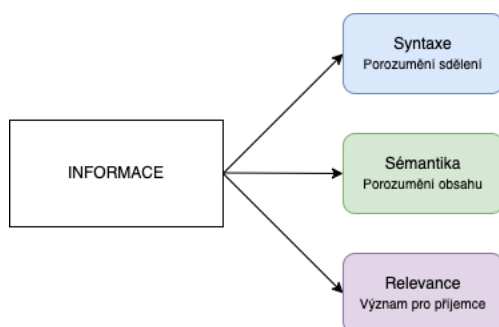
1.1.1 Data

Jednoduše se dá říct, že data jsou potenciálními informacemi. Stávají se informacemi v moment, kdy jsou používány k rozhodování. Data se dají uložit pro pozdější využití. Dají se zaznamenat například pomocí inkoustu na papír, nebo elektrických signálů. Můžeme je dekódovat na informace, případně informace kódovat zpět na data (3, s. 5).

Data se rozlišují na strukturované a nestrukturované. Strukturované data mají svůj řád a lze je alespoň částečně chápat. Strukturované data si můžeme představit jako data v tabulce uživatele. Nestrukturované data jsou ty data, kterým na první pohled nerozumím a potřebujeme nějaký nástroj pro jejich přečtení. Pod nestrukturovanými daty si lze představit nahrávky obrazu (5).

1.1.2 Informace

Informace je zpráva, která musí splňovat tři požadavky. Informace lze členit podle mnoha hledisek, např. informace operativní, informace strategické a taktické, informace krátkodobé a dlouhodobé.



Obrázek 1 Informace (Zdroj: Vlastní zpracování podle 3, s.4)

1.1.2.1 Syntaktická relevance

Prvním požadavkem je syntaktická relevance. Syntaktická relevance je dodržena tehdy, je-li příjemce schopen zprávu detekovat a porozumět ji.

1.1.2.2 Sémantická relevance

Druhý požadavek je sémantická relevance. Jestliže příjemce dostane zprávu, které dokáže porozumět a vypovídá o něm a okolí, je splněna sémantická relevance.

1.1.2.3 Pragmatická relevance

Poslední požadavek je pragmatická relevance, kdy zpráva pro příjemce musí mít nějaký význam (3, s. 4).

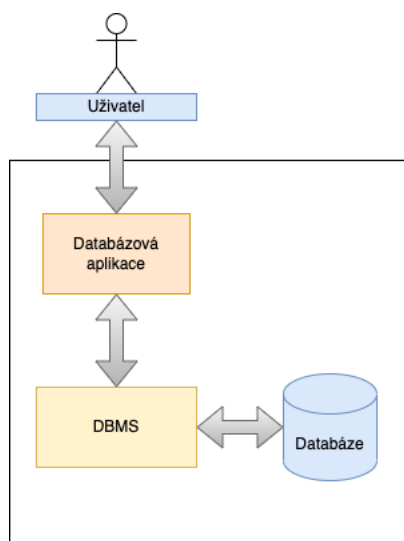
1.1.2.4 Znalosti

„Znalosti jsou výsledkem porozumění informací, která byla právě sdělena, a její integrace s dřívějšími informacemi“ (3, s. 5)

Znalost lze také chápat jako propojení více relevantních informací, potřebných pro řešení konkrétního problému. (3, s. 6)

1.2 Databázový systém

Databázový systém je tvořen čtyřmi hlavními komponenty. Jsou to uživatelé, databázová aplikace, systém řízení báze dat (DBMS) a vlastní databáze (6, s. 29).



Obrázek 2 Databázový systém (Zdroj: Vlastní zpracování podle 1, s.30)

1.2.1 Databáze

Databáze je soubor vzájemně souvisejících a strukturovaných dat, které mají obsahují vlastní popis. Konkrétně u relační databáze platí, že se jedná o skupinu souvisejících tabulek s vlastním popisem. Související tabulky a vlastní popis jsou klíčové pro tvorbu databázového systému. Související tabulky jsou ty, které mají mezi sebou vazbu pomocí klíče. Vlastní popis slouží k porozumění struktury databáze při jejím prozkoumávání. Všechny popis struktury databáze musí být její součástí a musí být popsán tak, abychom nemuseli hledat informace z jiných zdrojů. O popis struktury se starají metadata. Patří sem názvy tabulek a sloupců nebo vlastnosti tabulek a sloupců. (6, s. 30).

Databáze obsahuje řadu dalších typů dat. Mimo uživatelská data a metadata se jedná také o indexy a jiné struktury, díky kterým se zvyšuje výkon databáze (6, s. 31).

1.2.2 Systém řízení báze dat (DBMS)

DBMS je softwarový systém, který umožňuje vytváření, spravování a údržbu dat pomocí dotazů. Tyto systémy se pořizují pomocí licence od různých dodavatelů. Mezi nejznámější systémy patří mimo jiné: Microsoft Access, MySQL, Microsoft SQL Server nebo Oracle Database (6, s. 31).

Jednotlivé funkce DBMS můžeme vidět v následujícím obrázku.

Funkce DBMS
Vytvoření databáze
Vytvoření tabulek
Vytvoření podpůrných struktur
Čtení dat z databáze
Úpravy databázových dat
Údržba databázových struktur
Vynucování pravidel
Kontrola souběžnosti
Zajištění bezpečnosti
Zálohování a obnovení

Obrázek 3 Funkce DBMS (Zdroj: Vlastní zpracování podle 1, s.31)

DBMS mimo vytváření databází a tabulek umí také vytvořit další pomocné struktury, které zvyšují výkon databáze (např. indexy). Další funkcí je čtení a úprava dat. Tyto akce se spouštějí pomocí příkazů v dotazovacím jazyku SQL. O změnu struktury se stará údržba databázových struktur, která dokáže změnit např. formát tabulky (6, s. 31).

Pomocí vynucování pravidel systém pomáhá udržovat vztahy mezi tabulkami a zajišťovat správný chod. Tento proces se nazývá omezení referenční integrity. Kontrola souběžnosti má

za úkol neomezovat činnost jednoho uživatele činností druhého. Bezpečnostní systém hlídá oprávnění uživatelů a povoluje operaci s daty pouze uživatelům s požadovaným oprávněním. Poslední funkcí je zálohování a obnovení dat. Databáze bývají velmi důležitým aktivem firmy a její ztráta by mohla mít katastrofální dopad na celou firmu, proto je důležité pravidelné zálohování (6, s. 32).

1.2.2.1 Architektury DBMS

Architekturu DBMS dělíme na:

- Jednovrstvá architektura
- Dvojvrstvá architektura
- Trojvrstvá architektura

Jednovrstvá architektura (centralizovaná)

Tato architektura se již nepoužívá. Počítač se může připojit lokálně či vzdáleně pomocí terminálu a všechny operace jsou provedeny na centrálním serveru (7, s. 5).

Dvojvrstvá architektura

Tato architektura, která se také nazývá klient – server, má data uložena na straně serveru a databázová aplikace (všechny aplikace a uživatelská rozhraní) běží na straně klienta (7, s. 8).

Přednosti této architektury spočívají v rozdělení práce mezi server a klienta. Díky tomu se snižuje náročnost přenášení dat po síti a je snadnější údržba integrity (7, s. 9).

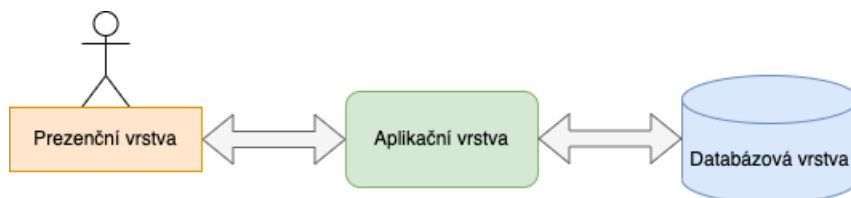
Trojvrstvá architektura

Trojvrstvá architektura je nejpoužívanější a je rozdělena na aplikační, prezenční a databázovou vrstvu (9).

Prezenční vrstva je například html soubor webové stránky a nachází se na straně klienta.

Aplikační vrstva je prostředník mezi prezenční a databázovou vrstvou. Tato vrstva dostane dotaz od prezenční vrstvy, zpracuje požadavek a dotáže se databázové vrstvy na požadovaná data. Tyto data dále zpracovává a posílá zpět prezenční vrstvě.

Databázová vrstva naslouchá aplikační vrstvě a posílá ji požadovaná data (8).



Obrázek 4 Trojvrstvá architektura (Zdroj: Vlastní zpracování)

1.2.3 Databázové aplikace

Databázová aplikace je program, který funguje jako prostředník mezi uživatelem a DBMS. S databází komunikuje pomocí SQL jazyka (6, s. 29).

1.2.4 Uživatel

Uživatele mohou pomocí formuláře načítat data, vkládat data, nebo se na ně dotazovat. Díky databázové aplikaci si udržují přehled o určitých skutečnostech.

1.3 Datové modely

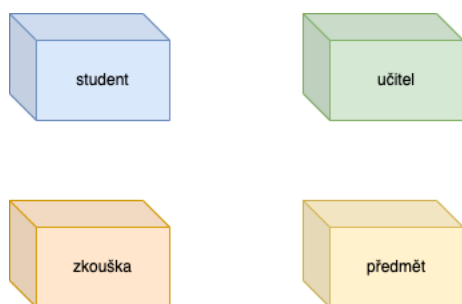
Každý reálný datový objekt – zvíře, zaměstnanec, automobil – je v datovém modelu prezentovaný jako datový objekt (entita). Současně se pro každý datový objekt musí definovat atributy. Tyto atributy uchovávají informace o datovém objektu (3, s. 11).

Při vytváření informačního systému si většinou nevystačíme s jedinou strukturou věty. Pro každý typ datového objektu, který je částí informačního systému, je potřeba navrhnout samostatnou datovou strukturu věty. Data vložená do systému musí odpovídat obrazu realitě. Pokud budeme mít systém správy zaměstnanců, musíme navrhnout datové struktury například pro zaměstnanec, mzda, smlouva. Tyto objekty se poté musí propojit, jelikož spolu souvisí (3, s. 20).

V dnešní době je momentálně na výběr ze tří datových modelů: lineární, relační, objektový (3, s. 20).

1.3.1 Lineární model

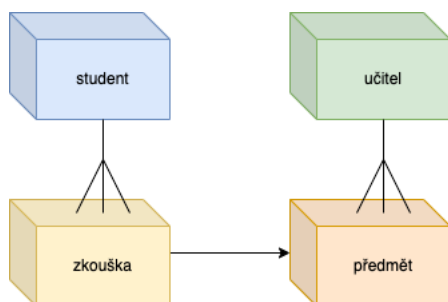
V lineárním modelu nenajdeme žádnou vazbu mezi jednotlivými tabulkami. Nelze tím pádem zjistit, jaký zaměstnanec uzavřel jakou smlouvu, nebo jaký má plat. Příklad lineárního modelu je například kartotéka klientů, kdy jsou jednotlivé karty s informacemi o klientech seřazeny v krabici a není mezi nimi žádný vztah (3, s. 21).



Obrázek 5 Lineární model (Zdroj: Vlastní zpracování dle: 3, s 20)

1.3.2 Relační model

Relační model patří k nepoužívanějším modelům dnešní doby. Vzniká z vícero lineárních modelů, které mají mezi sebou vazbu. Tato vazba vzniká pomocí relačního klíče. Toto propojení není trvalé, ale vzniká v momentě potřeby dat z více propojených tabulek a zaniká, když práci s modelem ukončíme. (3, s. 21).

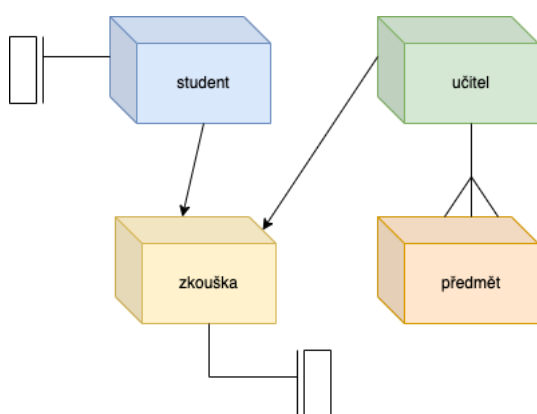


Obrázek 6 Relační model (Zdroj: Vlastní zpracování dle: 3, s 21)

1.3.3 Objektový model

Objektový model je nejnovějším datovým modelem. Tyto modely jsou vystavěny na jednom základním prvku – objektu, kde má tento objekt mimo atributy definované i metody, které určují jeho chování. (3, s. 21).

Jestliže máme například objekt zkouška, atributy tohoto objektu budou datum, známka, číslo studenta, termín, předmět a zkoušející. Tento objekt pak může mít nadefinované různé metody jako například metodu vytvoř záznam o zkoušce. Tato metoda poté kontroluje, zda daný student nemá již tuto zkoušku hotovu, nebo jestli nevyčerpal maximální počet termínů (3, s. 22).



Obrázek 7 Datový model (Zdroj: Vlastní zpracování dle: 3, s 22)

1.4 Relační datový model

Relační datový model je založen na teorii relací. Jednoduše lze říct, že relační datový model nám umožňuje zachytit v modelu kromě dat o zkoumaných objektech, vzájemné vztahy těchto objektů. To má za efekt větší přiblížení reálnému světu (3, s. 23).

Relační model dat obsahuje pět hlavních položek:

Relace – tabulka se sloupci a řádky

Atribut – pojmenovaný sloupec relace

Doménová n-tice – řádek relace

Doména – množina přístupných hodnot pro atribut.

Relační databáze – kolekce normalizovaných tabulek (2, s. 63)

1.4.1 Relace

Databázové produkty ukládají data ve formě relací. Tím je myšlena dvojrozměrná tabulka, která se skládá ze schématu (záhlaví) a těla relace. Obsahuje řádky (záznamy), sloupce (atributy) a vyznačuje se určitými vlastnostmi. (6, s. 78)

Vlastnosti relace
Každý řádek obsahuje data o entitě
Každý sloupec obsahuje data o attributech entity
Buňka v tabulce uchovává pouze jednu hodnotu
Položky ve stejném sloupci jsou jednoho druhu
Všechny položky mají jedinečný název
Na pořadí sloupců nezáleží
Na pořadí řádků nezáleží
Žádné dva řádky nesmí obsahovat identické sady datových hodnot

Obrázek 8 Vlastnosti relace (Zdroj: Vlastní zpracování dle: 6, s. 78)

1.4.2 Integrita relačního modelu

Při tvorbě teoretického modelu musíme mít na vědomí, že modelování dat z reality sebou přináší i určitá omezení. Stav, při kterém data uložena v modelu odpovídají vlastnostem objektů reálného světa můžeme chápat jako integritu modelu (3, s. 27)

Integritní omezení dělíme na:

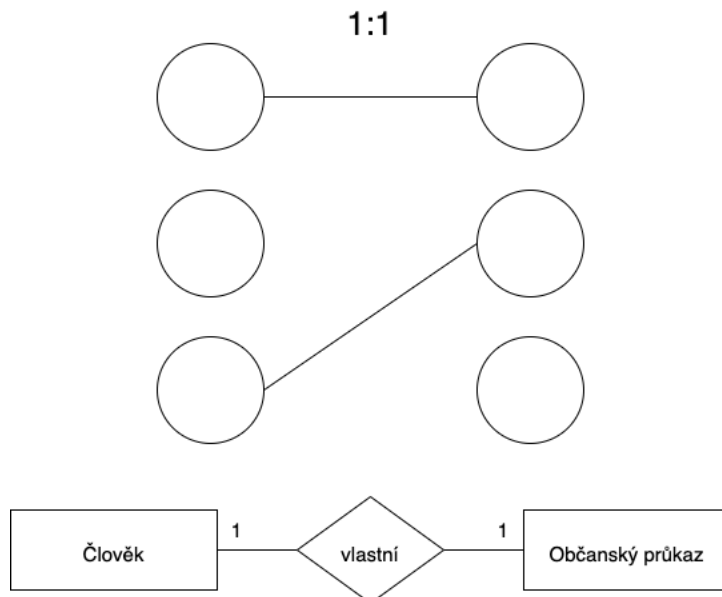
- Omezení samotných entit
- Omezení vzájemných vztahů mezi entitami (3, s. 27)

1.4.2.1 Integritní omezení pro entity

- **Doménová integrita** – Hodnoty atributů relace musí být z domén přístupných pro tyto hodnoty. Každý atribut musí mít definovanou doménu a specifikované povolené hodnoty (3, s. 28).
- **Entitní integrita** – Každá relace musí mít definovaný **primární klíč (PK)**. Tento klíč jednoznačně identifikuje všechny záznamy v relaci. Může být reprezentován jedním atributem, nebo složen z více atributů. Primární klíč musí splňovat vlastnosti jednoznačnosti (pro každou n-tici relace existuje unikátní primární klíč) a minimality (nelze vypustit žádný atribut primárního klíče bez toho, aby byla porušena podmínka jednoznačnosti) (3, s. 28).
- **Referenční integrita** – Pomáhá udržet vztahy pomocí primárního a cizího klíče. Oba klíče musí mít definovanou stejnou doménu (3, s. 29)
- **Cizí klíč (FK)** – Je atribut, jehož hodnota může být plně zadaná nebo nezadaná. Cizí a primární klíč je propojen relací totožnou hodnotou. (3, s. 29)

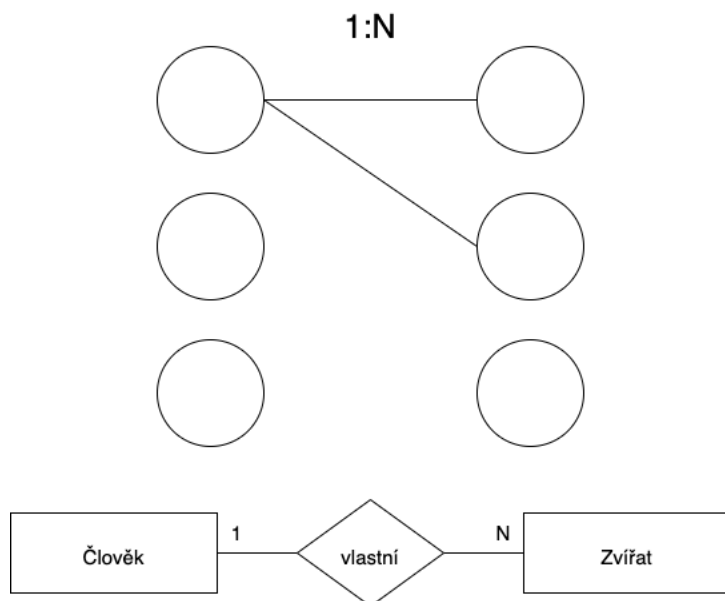
1.4.2.2 Integritní omezení pro vztahy entit

- 1:1 – Jedna n-tice jedné relace odpovídá jedné nebo žádné n-tici druhé relace. Příklad: Člověk vlastní jeden nebo žádný občanský průkaz (3, s. 30).



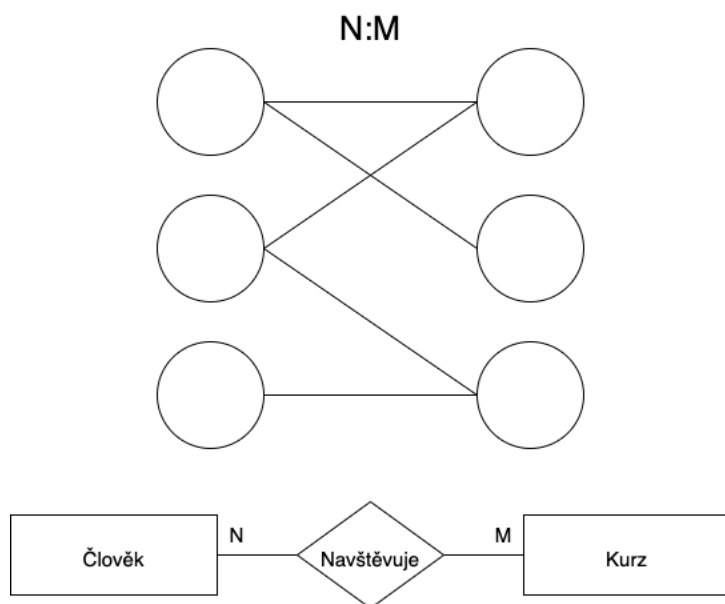
Obrázek 9 relace 1:1 (Zdroj: vlastní zpracování podle 3, s. 31-32)

- 1:N – Jedna n-tice relace odpovídá jedné nebo více n-ticím jiné relace. Příklad: Člověk může vlastnit jedno nebo více zvířat (3, s. 31).



Obrázek 10 relace 1:N (Zdroj: vlastní zpracování podle 3, s. 31-32)

- N:M – Několika n-ticím jedné relace odpovídá více n-tic druhé relace. Př. člověk může navštěvovat více kurzů a současně do jednoho kurzu může chodit více lidí (3, s. 31).



Obrázek 11 relace N:M (Zdroj: vlastní zpracování podle 3, s. 31-32)

1.5 Normalizace

Normalizace je technika používaná při tvorbě databáze. Normalizace má za úkol vytvořit takovou sadu tabulek, která bude splňovat datové požadavky organizace a zároveň bude obsahovat co nejmenší počet redundance (2, s. 188).

Redundantní data plýtvají místem na disku a způsobují potíže s údržbou (10).

Při normalizaci určujeme, zda jednotlivé tabulky vyhovují všem normálním formám. V současné době existuje šest druhů normálních forem. Nejčastěji používané jsou 1. normální forma, 2. normální forma a 3. normální forma (2, s. 188).

1.5.1 První normální forma

Relace je v první normální formě, pokud průsečík sloupce a záznamu obsahuje pouze jednu hodnotu. To znamená, že relace nemá žádné složené nebo vícehodnotové atributy (2, s. 191).

- Eliminace opakujících se skupiny v jednotlivých tabulkách.
- Tvorba samostatné tabulky pro každou sadu souvisejících dat.
- V každé sadě souvisejících dat je nutná definice identifikace prostřednictvím primárního klíče (10).

1.5.2 Druhá normální forma

Relace je v druhé normální formě, pokud splňuje první normální formu a zároveň jsou její neklíčové atributy závislé na primárním klíči (2, s. 192).

- Tvorba oddělených tabulek pro sady hodnot, které se týkají více záznamů.
- Tvorba relací na tyto tabulky pomocí cizího klíče. (10)

1.5.3 Třetí normální forma

Relace je v třetí normální formě, pokud splňuje druhou normální formu a zároveň jsou její neklíčové atributy závislé pouze na primárním klíči (2, s. 195).

- Eliminace polí, která nezávisí na klíči (10).

1.6 Návrh databáze

Při návrhu databáze je vhodné zvolit systematický přístup. Zvláště u větších databází je velmi důležité, aby byla databáze navržena co nejlépe. Toho docílíme strukturovaným přístupem k analýze a modelování souboru požadavků, který bude sloužit k plánování, správě, kontrole a zhodnocení. (2, s. 206).

Návrh databáze se skládá ze tří hlavních fází:

- Konceptuální návrh databáze
- Logický návrh databáze
- Fyzický návrh databáze (2, s. 206)

1.6.1 Konceptuální návrh databáze

V první fázi vytvoříme ER model, který je součástí dokumentace a obsahuje datový slovník a ER diagram. Slouží jako zdroj informací pro další fázi návrhu (2, s. 206).

1.6.2 Logický návrh databáze

V druhé fázi převedeme ER model do množiny relačních tabulek. Použitím normalizace poté zkontrolujeme strukturu tabulek a schopnost plnit požadované operace. Následně definujeme integritní omezení (2, s. 207).

1.6.3 Fyzická návrh databáze

Poslední fáze má za úkol převést návrh databáze do DBMS. Dále musí vytvořit popis implementace databáze na vnějších paměťových zařízeních (2, s. 208).

1.7 Jazyk SQL

Jazyk SQL je strukturovaný dotazovací jazyk, který se používá pro správu relačních databází a provádění různých operací s daty uvnitř databáze. Byl vytvořen v roce 1970 a je pravidelně využíván nejen správci databází, ale také programátory, kteří píší skripty pro integraci dat (11).

SQL se používá pro:

- úpravu databázových tabulek a indexových struktur
- přidávání, mazání a úpravu řádků s daty
- získávání podmnožin informací ze systémů správy relačních databází (RDBMS) (11).

Jak již bylo řečeno, SQL je dotazovací jazyk, a proto obsahuje pouze konstrukce, které umožňují definovat a zpracovávat databázi. Tím pádem neposkytuje plně programátorské možnosti. Jestli tedy chceme databázi celou naprogramovat, musíme použít některý ze skriptovacích jazyků jako je např. Java nebo C#, ve kterých voláme příkazy SQL (6, s. 127).

1.7.1 Kategorie příkazů SQL

SQL příkazy jsou rozděleny do několika kategorií a mají stejnou základní syntaxi (11).

1.7.1.1 Data Definition Language (DDL)

Umožňuje vytvářet databázové objekty nebo je upravovat. Jsou to příkazy CREATE, ALTER a DROP (11).

1.7.1.2 Data Manipulation Language (DML)

Umožňuje manipulaci s daty přes příkazy INSERT, UPDATE a DELETE. Tyto příkazy vkládají, upravují a mažou data z databáze.

1.7.1.3 Data Query Language (DQL)

Tento jazyk obsahuje pouze jeden příkaz SELECT, který se používá k získání specifických dat (11).

1.7.1.4 Data Control Language (DCL)

Umožňuje používat příkazy pro udělení či odebrání uživatelského oprávnění. Spadají sem příkazy GRANT a ALTER (11).

1.7.1.5 Transaction Control Language

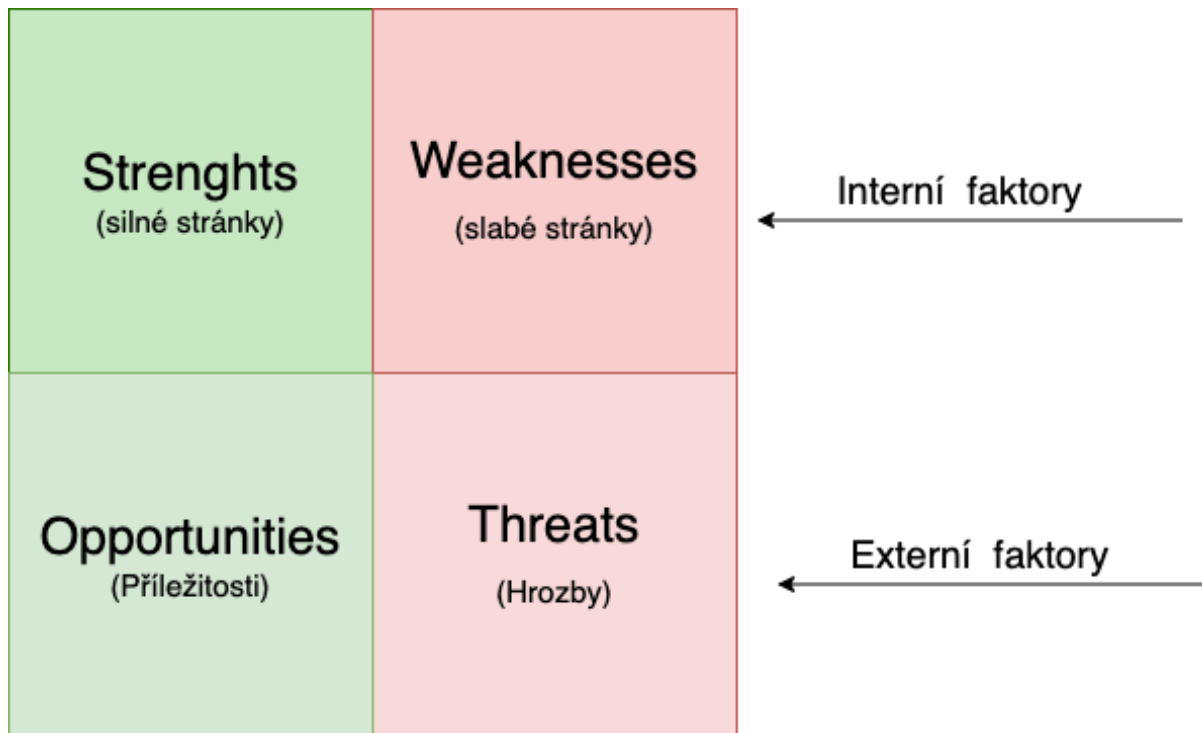
Umožňuje změnit stav dat například příkazem COMMIT pro uložení změn, nebo ROLLBACK pro vrácení změn (11).

1.8 SWOT analýza

SWOT analýza je jednou z nejpoužívanějších a nejuniverzálnějších analýz a používá se pro zhodnocení vnitřních a vnějších faktorů ovlivňující úspěšnost firmy.

Analýza hodnotí ve vnitřním prostředí silné a slabé stránky, ve vnějším prostředí zase příležitosti a hrozby.

Hotová analýza se následně využívá jako podklad pro podpoření silných stránek, omezení těch slabých, hledání nových příležitostí a poznání možných hrozeb. (12)



Obrázek 12 SWOT (Zdroj: vlastní zpracování podle 12)

2 ANALÝZA SOUČASNÉHO STAVU

V této kapitole se budu věnovat analýze současného stavu závodní jídelny. Prvně si popíšeme základní informace o společnosti, její fungování a struktuře. Následně analyzujeme současný stav webových stránek a zpracování dat.

2.1 Základní informace o společnosti

Název společnosti: Jídelna Centrum s.r.o.

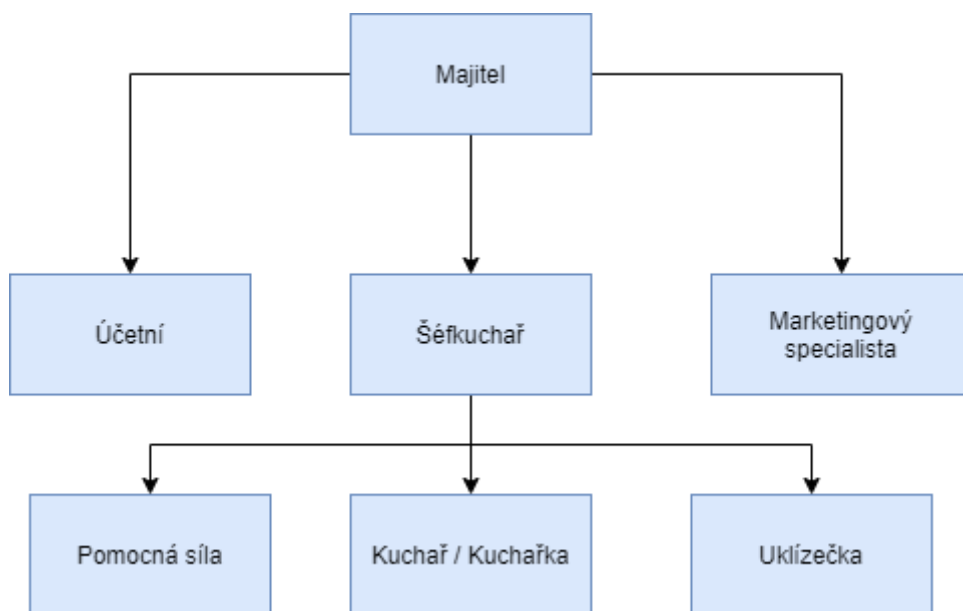
Sídlo: Pekařská 7602 00 Brno-střed-Staré Brno

Založení: 3.2 2021

Jedná se o imaginární firmu, která působí na poli gastronomie. Firma sídlí v Brně a věnuje se závodnímu stravování. Zakládá si hlavně na kvalitě za rozumnou cenu a vaří převážně česká jídla. Cílí hlavně na okolní firmy a jejich zaměstnance. Za jeden rok, co působí na trhu, si dokázala i přes nepříznivou situaci ve světě najít místo mezi konkurencí a má již stabilní strávníky, které získala hlavně přes spolupráce s okolními podniky, kdy jejich zaměstnanci mohou čerpat zaměstnanecké slevy právě v jejich jídelně. Firma má ve svém portfoliu již několik spoluprací a na dalších usilovně pracuje.

2.2 Organizační struktura

Ve firmě pracuje dohromady 10 zaměstnanců. Vaří tu celkem 6 kuchařů a z toho jeden šéfkuchař, který se stará o chod jídelny a vytváří denní menu. Dále má pomocné síly do kuchyně a jednu uklízečku. O finance se stará externí účetní. Firma má i jednoho marketingového specialistu, který komunikuje s firmami v okolí a snaží se uzavřít nové spolupráce. Firmu vlastní jeden majitel, který úzce spolupracuje se všemi zaměstnanci.



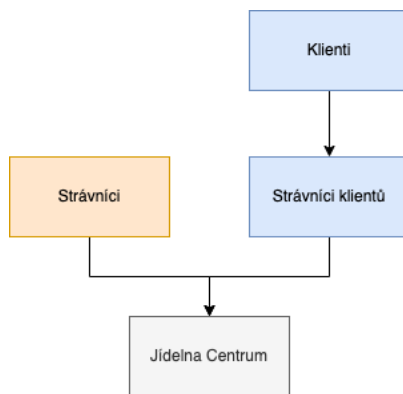
Obrázek 13 struktura zaměstnanců (Zdroj: vlastní zpracování)

2.3 Popis současné situace

Jak již bylo řečeno, firma působí na trhu jeden rok a již od začátku zaznamenává rostoucí trend i přes nepříznivou situaci ve světě v době korona krize. Majitel se proto rozhodl, že je na čase posunout firmu o kus dál a zaměřit se na slabiny firmy. Hlavní slabinu vidí v technologii. Firma používá jednoduchý web, který slouží pouze pro dohledání základních informací o jídelně, jako je otevírací doba a jídelníček. Stránky nejsou uzpůsobené budoucímu vývoji a rozšíření působnosti firmy. Firma by chtěla například začít rozvážet obědy strávnickům do práce i domů, díky čemuž by se mohla vyhnout dalším budoucím omezením, které jsme zažili v době pandemie. Vytvořením databáze a předěláním webu budou mít klienti v budoucnu plno nových možností, jako například objednání obědu, hodnocení, rozvoz obědů domů či práce a mnoho dalšího. V neposlední řadě budou nové stránky a funkcionalita stránek sloužit i k lepšímu postavení ve vyjednávání spoluprací u potencionálních klientů.

2.4 Strávníci

Strávníci jsou rozděleni do dvou skupin. První skupina jsou zaměstnanci firem, s kterými má jídelna spolupráci. Tito strávníci mohou čerpat zaměstnanecké slevy. Druhá skupina jsou lidé z okolí, kteří nejsou zaměstnanci klientů a mají základní ceník jídel.



Obrázek 14 Typ strážníků (Zdroj: vlastní zpracování)

2.5 Proces shánění nových klientů

Klienti jsou firmy, které mají domluvenou spolupráci a jejich zaměstnanci využívají služeb jídelny. Marketingový specialista sjednává schůzky s vedením firem v okolí a snaží se získat nové klienty. Nabízí jim především výhodnější ceny pro zaměstnance. Díky tomu si firma během krátkého působení získala stálé strážníky a drží krok s konkurencí.

2.6 Proces odběru jídla

Odběr jídla funguje na klasickém způsobu, kdy strážník přijde do jídelny, vybere si jídlo a zaplatí za něj. Strážník, který využívá zaměstnaneckých slev, před placením přiloží čip ke čtečce, která zkontroluje platnost a poté může čerpat slevu při placení. Tento způsob odběru má několik nedostatků a není optimální pro chod firmy. Jakmile proběhne digitalizace, bude možné objednávat i mimo jídelnu z domu nebo kanceláře. Díky databázi pak můžeme i ukládat informace o jednotlivých objednávkách. Toho pak může firma využít například při odhadu počtu jídel na jednotlivý den.

2.7 Vybavení jídelny

Firma vlastní velký prostor, kde je rozmístěno 21 stolů, které momentálně stačí největšímu dennímu náporu v odpoledních hodinách. Firma využívá nejrůznějších spotřebičů a vybavení nutné pro provoz jídelny.

2.8 Hardwarové vybavení

Firma vlastní jeden stolní počítač a tiskárnu v kanceláři majitele. Marketingový specialista má firemní laptop a mobilní telefon. Pro kontrolu zaměstnanecké slevy slouží terminál, ke kterému

strávníci přiloží čip a při platné kontrole mohou odebírat jídlo se slevou. V budoucnu by se hardwarové vybavení mělo rozšířit i pro některé ostatní zaměstnance pro efektivnější chod firmy.

2.9 Softwarové vybavení

Firma nevlastní velké množství softwaru, jelikož v této oblasti podnikání není tolik potřeba. Pro zaslání mailu slouží emailový klient Mozilla Thunderbird.

Marketér pro prezentaci firmy využívá převážně Microsoft PowerPoint a Microsoft Excel pro tvorbu grafů.

Účetní pracuje v účetním systému Money S3, který je ideální volba pro menší firmy.

2.10 SWOT analýza

Na základě SWOT analýzy nejprve zhodnotíme silné a slabé stránky firmy. Následně si popíšeme největší příležitosti a hrozby.

2.11 Silné stránky

Jídlna si za pár let činnosti vytvořila velmi dobrou pověst a image. Díky příjemnému prostředí, obsluze a skvělému jídlu si zaslouženě vydobila místo na trhu. Prostory jsou nejen příjemně zařízeny, ale také mají dostatečnou kapacitu pro velký nápor lidí v dopoledních hodinách. Jednou z nejsilnějších stránek je chytré vymyšlený marketing, díky kterému firma získává klienty (firmy) a jejich zaměstnanci následně využívají pravidelně jídelní prostory.

2.12 Slabé stránky

Jednou z nejslabších stránek jsou zastaralé statické webové stránky, které nepůsobí na první pohled vůbec dobře a mohou odradit případné hosty. Firma má také problémy s dodavatelem, který není poslední měsíce schopen dodávat suroviny včas. Uvnitř kolektivu panuje taky špatná organizace, kterou způsobuje právě absence databázového systému vytvořeného na míru.

2.13 Příležitosti

Nejzásadnější příležitost, která je schopna markantně zvýšit výkonnost firmy spočívá v efektivním ukládání dat. Tyto data mohou sloužit k lepší organizaci uvnitř firmy, nebo tvorbě různých grafů a statistik pro prezentaci klientům. Další příležitost je umožnit rozvoz obědů přímo ke strážníkům do práce či domů. V období kdy ještě není korona krize zcela za námi, je

toto důležité opatření pro možný negativní budoucí vývoj ve světě. Poslední příležitost se naskytuje ve změně dodavatele potřebných surovin.

2.14 Hrozby

Jídlna sídlí v centru města, a proto má v okolí velkou konkurenci. Nesmí tedy polevit ve své kvalitě a nadále investovat své příjmy zpátky do firmy. Ekonomická situace ve světě není v dnešní době nejlepší a hrozí další vládní opatření.

s	w
<ul style="list-style-type: none"> - Pověst a image firmy - Příjemné prostředí - Velké prostory - Propracovaný marketing - Klientela 	<ul style="list-style-type: none"> - Zastaralé webové stránky - Příjemné prostředí - Nevhodný dodavatel - Špatná organizace - Absence relevantních dat na jednom místě
o	t
<ul style="list-style-type: none"> - Ukládání dat do databáze - Příjemné prostředí - Tvorba grafu a statistik - Změna dodavatele - Rozvoz obědů 	<ul style="list-style-type: none"> - Velká konkurence - Vládní opatření - Ekonomická situace

Obrázek 15 SWOT analýza (Zdroj: vlastní zpracování)

3 VLASTNÍ NÁVRH ŘEŠENÍ

V této kapitole si popíšeme vlastní návrh řešení. Začneme procesy v rámci kterých si ukážeme vývojový diagram a pohled ze strany uživatele. Dále bude vytvořen konceptuální, logický a fyzický návrh databáze. Tento návrh zjednoduší fungování firmy a firemních procesů.

3.1 Funkcionalita

Databáze je navržena tak, aby mohla být napojena na nové webové stránky a zajišťovala všechny potřebné procesy, které firma vyžaduje. Webové stránky budou používány hlavně zákazníky. Zákazníci si budou moci například zobrazit přehled svých objednávek, nebo se podívat na konkrétní menu, případně provést rezervaci a objednávku online. Web může mít díky databázi i modul pro zaměstnance, převážně účetní a marketéra, kteří si budou moci díky uloženým datům vytvářet nejrůznější formuláře, faktury, grafy apod.

3.1.1 Proces objednávky

Objedávka je možná s dovozem online anebo přímo na místě.

Pokud si zákazník objednává jídlo na místě, řeší vše pokladní, která na terminálu zadá objednávku a zákazník provede platbu. Jedná-li se o zákazníka, který má možnost uplatnit zaměstnaneckou slevu, identifikuje se před platbou svoji čipovou kartou. Poté se ověří platnost čipu a vypočítá se cena objednávky.

V případě online objednávky domů, nebo do práce, musí zákazník provést objednávku online, kde vyplní potřebný formulář a následně ho odešle. Aplikace po každé úspěšné objednávce zapíše potřebné záznamy do databáze.

Pavel Dvořák

Čas

dd.mm.rrrr 

Adresa

Náměstí 124/1 -> Kancelář

Polní 83 -> Domů

Dnešní menu

- Zelňačka
- Smažený vepřový řízek, brambor, okurka
- Plněné bramborové knedlíky uzeným, zelí

[Přidat jídlo](#)

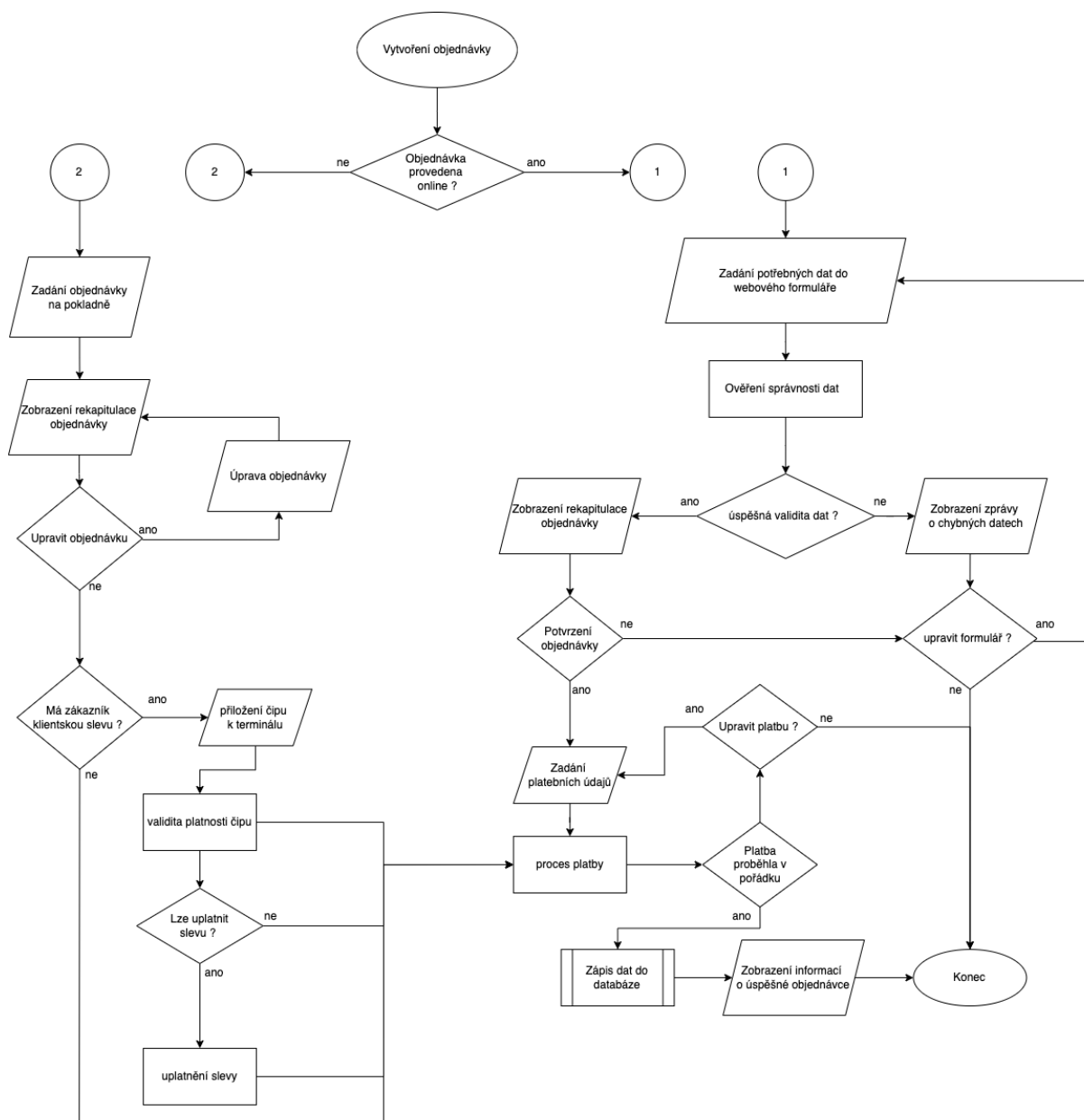
Košík

Zelňačka -> 1x

Smažený vepřový řízek, brambor, okurka -> 2x

[Objednat](#)

Obrázek 16 Uživatelský pohled procesu objednávek (Zdroj: vlastní zpracování)



Obrázek 17 Vývojový diagram procesu objednávek (Zdroj: vlastní zpracování)

3.1.2 Proces rezervace

Rezervaci lze provést telefonicky, nebo přes webové stránky. Na webových stránkách bude provádět rezervaci jak přihlášený uživatel, tak host. Přihlášený uživatel nemusí vyplňovat některé položky, protože je již máme uložené v databázi. Systém si před zobrazením formuláře načte data o rezervacích a po zadání dne rezervace ukáže uživateli všechny dostupné hodiny, které je možné rezervovat.

Rezervace

Jméno
Michal

Příjmení
Opluštil

telefon
342 552 492

Den
10.05.2022 📅

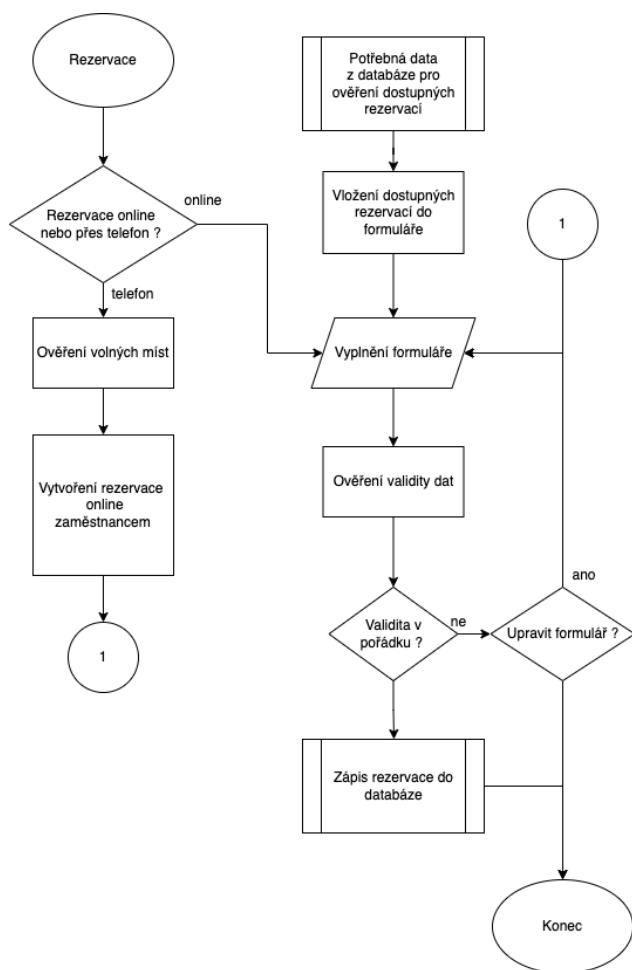
čas
13:00 - 14:00 ▾

Počet osob
3

Poznámka

Rezervovat

Obrázek 18 Uživatelský pohled procesu rezervace (Zdroj: vlastní zpracování)



Obrázek 19 Vývojový diagram procesu rezervace (Zdroj: vlastní zpracování)

3.2 Konceptuální návrh

V části konceptuálního návrhu si popíšeme jednotlivé entity a jejich vzájemné vztahy. Výstup toho návrhu bude sloužit jako podklad pro další část návrhu.

3.2.1 Identifikace entit

V této kapitole si popíšeme všechny entity databáze. U každé entity si určíme její název, popíšeme, k čemu je určena a kolik se v nich bude vyskytovat záznamů.

Tabulka 1 Identifikace entit (Zdroj: vlastní zpracování)

Název entity	Popis entity	Počet výskytů
osoby		
address	Evidence všech adres	Stovky
osoby - zaměstnanec		
employee	Informace o zaměstnanci	Desítky
employee_position	Pozice zaměstnance	Desítky
employee_address	Přiřazení adresy zaměstnance	Desítky
bank	Bankovní údaje zaměstnance	Desítky
salary	Mzdy zaměstnanců	Desítky
osoby - klient		
client	Informace o klientech	Desítky
client_address	Přiřazení adresy klienta	Desítky
contract	Smlouvy klientů	Desítky
discount_type	Typ slevy na jídlo	Desítky
osoby - user		
user	Informace o registrovaném uživateli	Stovky
user_address	Přiřazuje adresy uživatelům	Tisíce
user_role	Přiřazení oprávnění	Jednotky
role	Číselník oprávnění	Jednotky
objednávky		
orders	Objednávky	Tisíce
order_item	Záznamy jednotlivých objednaných položek	Tisíce
transaction	Informace o transakci	Tisíce
delivery	Informace o doručení objednávky	Stovky
jídlo		
lunch	Evidence jednotlivých jídel	Desítky
lunch_allergen	Přiřazuje jednotlivé alergeny k jídlu	Jednotky
allergen	Číselník alergenů	Desítky
label	Štítek jídla	Desítky
menu	Zaznamenává vytvořená menu	Stovky
rating	Hodnocení jídla	Stovky
rezervace		
reservation	Záznam rezervace	Tisíce
reservation_info	Popis rezervace	Tisíce
tables	Evidence jednotlivých stolů	Desítky

3.2.2 Identifikace relací

V této kapitole si popíšeme vytvořené relace mezi jednotlivými entitami.

Tabulka 2 Identifikace relací (Zdroj: vlastní zpracování)

Název entity	Relace	Název entity
employee	Má	employee_address
employee	Má	bank
employee	Pracuje jako	employee_position
client	Má	client_address
client	Podepsal	contract
contract	Upřesňuje	discount_type
user	Spadá pod	user_role
user	Má	user_address
user	Vytváří	orders
user	Zaznamenává	rating
user	Spadá pod	client
reservation	Vytváří	user
tables	Se vztahuje	reservation
reservation_info	Upřesňuje	reservation
delivery	Upřesňuje	orders
delivery	Má	address
transaction	Upřesňuje	orders
orders	Se skládá	order_item
menu	Má	rating
menu	Upřesňuje	order_item
lunch	Upřesňuje	menu
lunch	Má	label
lunch	Obsahuje	lunch_allergen
lunch_allergen	tvoří	allergen
client_address	Upřesňuje	address
employee_address	Upřesňuje	address
user_address	Upřesňuje	address

3.3 Logický návrh

Další důležitou částí je logický návrh. V této kapitole si tedy popíšeme jednotlivé tabulky a definujeme jejich atributy. U atributu identifikujeme datový typ, jestli je atribut NULL nebo NOT NULL a stručný popis. V neposlední řadě si také určíme primární a cizí klíče. Primární klíče jsou označeny žlutou barvou a cizí klíče zelenou. Jedná-li se o atribut, který reprezentuje primární a cizí klíč, identifikujeme ho pomocí oranžové barvy.

3.3.1 Relační tabulky vztahující se k osobám

V této části si popíšeme všechny tabulky, které se vztahují k zaměstnancům, klientům a registrovaným uživatelům.

3.3.1.1 client

Tabulka client slouží k ukládání dat o všech klientech, kteří mají podepsanou smlouvu s kantýnou. Evidují se zde základní informace jako například název podniku, ico nebo dic. Pomocí primárního klíče je tabulka propojena s tabulkami contract, client_address a user.

Tabulka 3 client

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	ano	Identifikační číslo	Auto increment
name	varchar (64)	ano	Název klienta	
ico	varchar (10)	ano	Ičo klienta	
dic	varchar (12)	ano	Dič klienta	
email	varchar (64)	ano	Email klienta	
tel	varchar (15)	ano	Telefonní číslo klienta	
created_at	datetime	ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.1.2 client_address

Tabulka client_address je dekompozicí mezi vazbou client a address. Evidujeme zde klíče client_id a address_id. Tyto klíče jsou primární klíče tabulky a zároveň cizí klíče do tabulek client a address.

Tabulka 4 client_address (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
client_id	int	ano	Identifikace klienta	Primární klíč tabulky a cizí klíč do tabulky client
address_id	int	ano	Identifikace adresy klienta	Primární klíč tabulky a cizí klíč do tabulky address

3.3.1.3 address

Tabulka eviduje všechny možné adresy, které firma potřebuje uchovávat. Momentálně to jsou adresy klientů, zaměstnanců, a hlavně registrovaných uživatelů. Každý z těchto subjektů má vlastní dekompoziční tabulku, ve které uchováváme přiřazené adresy.

Tabulka 5 address (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	ano	Identifikace adresy	Primární klíč
street	varchar(64)	ano	ulice	
city	varchar(45)	ano	město	
zip	varchar(10)	ano	Psč	
address_type	varchar(15)	ano	Typ adresy	Lehčí identifikace adresy například „domů“ nebo „práce“
created_at	datetime	ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.1.4 contract

Tabulka eviduje jednotlivé podepsané smlouvy s klienty. Najdeme zde informace jako název smlouvy, jestli je smlouva aktivní nebo zrušená, období smlouvy a jiné. V této tabulce jsou i dva cizí klíče client_id a discount_type_id. Client_id nám odkazuje do tabulky client a propojuje nám tedy smlouvu s klientem. Discount_type_id nám určuje typ slevy z číselníku discount_type. (např. procento, množství)

Tabulka 6 contract (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
----------------	--------------------	----------	-------	----------

id	int	ano	Identifikace smlouvy	Primární klíč
discount_type_id	int	ano	Typ slevy	Cizí klíč do tabulky discount_type
client_id	int	ano	Přiřazený klient dané smlouvy	Cizí klíč do tabulky client
name	varchar(25)	ano	Název smlouvy	
discount	tinyint	ano	Hodnota slevy	
valid_from	date	ano	Začátek platnosti	
valid_to	date	ano	Konec platnosti	
active	tinyint(1)	ano	Status smlouvy	Automatické nastavení na 1
created_at	datetime	ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.1.5 discount_type

Tabulka slouží jako číselník pro zjištění typu slevy dané smlouvy. Sleva může být například 15 % na jídlo nebo 30kč sleva ze základní ceny.

Tabulka 7 discount_type (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	ano	Identifikace typu slevy	
type	Varchar(25)	ano	Typ slevy	

3.3.1.6 employee

Tabulka eviduje základní informace o všech zaměstnancích jako je jméno, mail a telefonní číslo. Tato tabulka je dále napojena na banku klienta, jeho plat, adresu a v neposlední řadě typ pozice, kterou zastává.

Tabulka 8 employee (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace zaměstnance	
employee_position_id	int	Ano	Pozice zaměstnance	Cizí klíč do číselníku pozic zaměstnanců
firstname	varchar(45)	Ano	Jméno	
lastname	varchar(45)	Ano	Příjmení	
tel	varchar(15)	Ano	Telefonní číslo	
email	varchar(64)	Ano	email	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.1.7 employee_position

Tabulka, která slouží jako číselník pracovních pozic zaměstnanců.

Tabulka 9 employy_position (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace pozice	
name	varchar(45)	ano	Název pozice	

3.3.1.8 employee_address

Tabulka employee_address slouží pro evidenci přidělených adres konkrétnímu zaměstnanci. Jsou zde pouze dva atributy, které jsou zároveň primární a cizí klíče.

Tabulka 10 employee_address (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
employee_id	int	ano	Identifikace zaměstnance	Primární klíč tabulky a cizí klíč do tabulky employee
address_id	int	ano	Identifikaci adresy	Primární klíč tabulky a cizí klíč do tabulky address

3.3.1.9 bank

Tabulka eviduje jednotlivé banky zaměstnanců pro zaslání výplaty. Jeden zaměstnanec má evidovanou pouze jednu banku, která je evidovaná na podepsané pracovní smlouvě.

Tabulka 11 bank (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	ano	Identifikace banky	
account_number	varchar(16)	ano	Číslo účtu	
bank_code	varchar(8)	ano	Číslo banky	
created_at	datetime	ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.1.10 salary

Tabulka pro evidenci platu zaměstnanců. Cizí klíč employee_id nám určuje zaměstnance, ke kterému je připojen konkrétní záznam. Platy zaměstnanců se mohou časem měnit a kvůli tomu je zde zaveden atribut active, který odděluje platné a neplatné mzdy.

Tabulka 12 salary (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace mzdy	
employee_id	int	Ano	Identifikace zaměstnance	Cizí klíč identifikující zaměstnance
amount	Decimal(8,2)	Ano	Výše mzdy	
valid_from	date	Ano	Platí od	
valid_to	date	Ano	Platí do	
active	tinyint(1)	Ano	Status platnosti	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.1.11 user

Tabulka user eviduje registrované uživatele v systému. Kromě základních informací jako jméno, příjmení a telefonní číslo, se zde eviduje zašifrované heslo účtu a kód čipu, který má přidělen každý uživatel spadající pod některého z klientů. Vztah mezi uživatelem a klientem nám určuje cizí klíč client_id.

Tabulka 13 user (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace uživatele	
client_id	int	Ne	Zaměstnavatel uživatele	
firstname	varchar(45)	Ano	Jméno	
lastname	varchar(45)	Ano	Příjmení	
tel	varchar(12)	Ne	Telefonní číslo	
email	varchar(64)	Ano	Mail uživatele	
password	varchar(45)	Ano	Heslo pro přihlášení	Zašifrované heslo
code	varchar(25)	Ne	Kód čipu	Kód pro uplatnění slevy
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.1.12 user_address

Tabulka evidující adresy registrovaných uživatelů. V tomto případě je potřeba ukládat pro každého uživatele jednu až více adres, kvůli dovozu.

Tabulka 14 user_address (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
address_id	int	Ano	Identifikace adresy	Primární klíč tabulky a cizí do tabulky address

user_id	int	ano	Identifikace uživatele	Cizí klíč do tabulky user
---------	-----	-----	------------------------	---------------------------

3.3.1.13 user_role

Každý uživatel musí mít přidělenou nějakou roli. Tato role bude sloužit pro povolení/zakázání jednotlivých modulů webových stránek. Tím můžeme rozdělit webovou aplikaci na interní (pro zaměstnance) a externí (pro zákazníky). Jsou zde dva cizí klíče, které definují uživatele a jemu přidělené role.

Tabulka 15 user_role (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace role uživatele	
user_id	int	Ano	Identifikace uživatele	Cizí klíč do tabulky user
role_id	int	ano	Identifikace role	Cizí klíč do tabulky role

3.3.1.14 role

Tabulka sloužící jako číselník pro evidenci jednotlivých rolí.

Tabulka 16 role (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace role	
name	varchar(32)	Ano	Název role	
description	varchar(128)	Ano	Popis role	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.2 Relační tabulky vztahující se k objednávkám

V této části si popíšeme tabulky, které se vztahují k objednávkám.

3.3.2.1 Orders

Tabulka objednávky eviduje záznamy o úspěšné objednávce. Pokud objednávku provede registrovaný uživatel online, přiřadí se objednávka k danému uživateli. Atribut delivery určuje, zda se jedná o objednávku s dovoz.

Tabulka 17 orders (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace objednávky	
user_id	int	Ne	Identifikace uživatele	Cizí klíč pro propojení uživatele s objednávkou
delivery	tinyint(1)	Ano	Doprava objednávky	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	Ne	Datum poslední úpravy záznamu	

3.3.2.2 order_item

Každá objednávka může mít jednu až více položek. Tyto položky jsou evidovány v tabulce order_item. Jednotlivé položky evidují jídlo z jídelníčku a jsou propojeny s konkrétní objednávkou v tabulce orders.

Tabulka 18 order_item (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace položky objednávky	
menu_id	int	Ano	Identifikace položky v menu	Ke každé položce je

orders_id	int	Ano	Identifikace objednávky	Propojuje všechny položky s objednávkou
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.2.3 transaction

Tato tabulka eviduje transakce, které byly provedeny online přes některou z dostupných platebních služeb. (např. stripe)

Tabulka 19 transaction (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace online transakce	
ticket	varchar(64)	ano	Identifikace online platby	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.2.4 delivery

Pokud si zákazník objednal jídlo s dovozem, zaevidujeme tyto informace do tabulky delivery. Primární klíč nám v tomto případě vytváří relaci s konkrétní objednávkou a address_id eviduje místo doručení.

Tabulka 20 delivery (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace dovozu	
address_id	int	Ano	Identifikace adresy dovozu	
delivery_time	datetime	Ano	Čas doručení	

3.3.3 Relační tabulky popisující jídlo

3.3.3.1 Lunch

Tabulka lunch eviduje jednotlivá jídla, které firma nabízí nebo bude nabízet. Z této tabulky se pak skládá tabulka menu, kterou si popíšeme o pár řádku dole. Každé jídlo může mít svůj štítek a přiřazené alergen.

Tabulka 21 lunch (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace jídla	
label_id	int	ne	Štítek jídla	
name	varchar(128)	ano	Název jídla	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.3.2 lunch_allergen

Tabulka slouží k přiřazení jednotlivých alergenů ke konkrétnímu jídlu. Každé jídlo může mít jeden až více alergenů, a proto použijeme dekompoziční tabulku mezi vazbou lunch a allergen.

Tabulka 22 lunch_allergen (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace alergenu jídla	
lunch_id	int	Ano	Identifikace jídla	
allergen_id	int	Ano	Identifikace alergenu	

3.3.3.3 allergen

Tabulka slouží jako číselník evidující názvy jednotlivých alergenů a jejich číslo.

Tabulka 23 allergen (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace alergenu	
name	varchar(32)	Ano	Název alergenu	
number	tinyint	Ano	Číslo alergenu	

3.3.3.4 label

V tabulce label evidujeme jednotlivé štítky, které slouží jako identifikátor jídla v denním menu.

Tabulka 24 label (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace značky	
name	varchar(32)	Ano	Název štítku	

3.3.3.5 menu

Tabulka menu eviduje informace potřebné k sestavení menu. Pro každou položku se musí přiřadit konkrétní jídlo, cena a celkové množství, které bude kontrolováno na základě počtu objednávek. Atribut date pak slouží k definování dne, kdy má být konkrétní položka v menu vystavena.

Tabulka 25 menu (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	ano	Identifikace menu	
lunch_id	int	ano	Identifikace jídla	

date	date	ano	Datum vystavení položky v menu	
price	decimal(6,2)	ano	cena	
quantity	int	ano	Celkové dostupné množství	
created_at	datetime	ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.3.6 rating

Tabulka rating slouží k evidenci hodnocení jídel. Hodnotit může jak přihlášený uživatel, tak host. Pokud provedl hodnocení přihlášený uživatel, zaznamenáme si ho pomocí cizího klíče v user_id. Druhý primární klíč odkazuje do tabulky menu na konkrétní položku.

Tabulka 26 rating (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace hodnocení	
menu_id	int	Ano	Hodnocená položka z menu	
user_id	int	ne	Uživatel, který hodnotil jídlo	
positive_rating	tinyint(1)	Ano	Typ hodnocení	
comment	varchar(128)	ne	Poznámka k hodnocení	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.4 Relační tabulky vztahující se k rezervaci

V této části si popíšeme tabulky, které se vztahují k rezervaci.

3.3.4.1 Reservation

Tabulka reservation eviduje všechny vytvořené rezervace. Pokud rezervaci založil přihlášený uživatel, přiřadíme si jeho ID k cizímu klíči user_id. Druhý cizí klíč tables_id slouží k propojení konkrétního stolu s rezervací.

Tabulka 27 reservation (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace rezervace	
tables_id	int	Ano	Identifikace rezervovaného stolu	
user_id	int	ne	Identifikace uživatele	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.4.2 reservation_info

Reservation_info eviduje podrobné informace o rezervaci. Jsou zde atributy pro název rezervace, čas rezervace a poznámku k rezervaci.

Tabulka 28 reservation_info

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace popisu rezervace	
reservation_name	varchar(45)	Ano	Jméno rezervace	
reservation_from	datetime	ano	Rezervace od	

reservation_to	datetime	ano	Rezervace do	
comment	varchar(128)	ne	Poznámka k rezervaci	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.4.3 tables

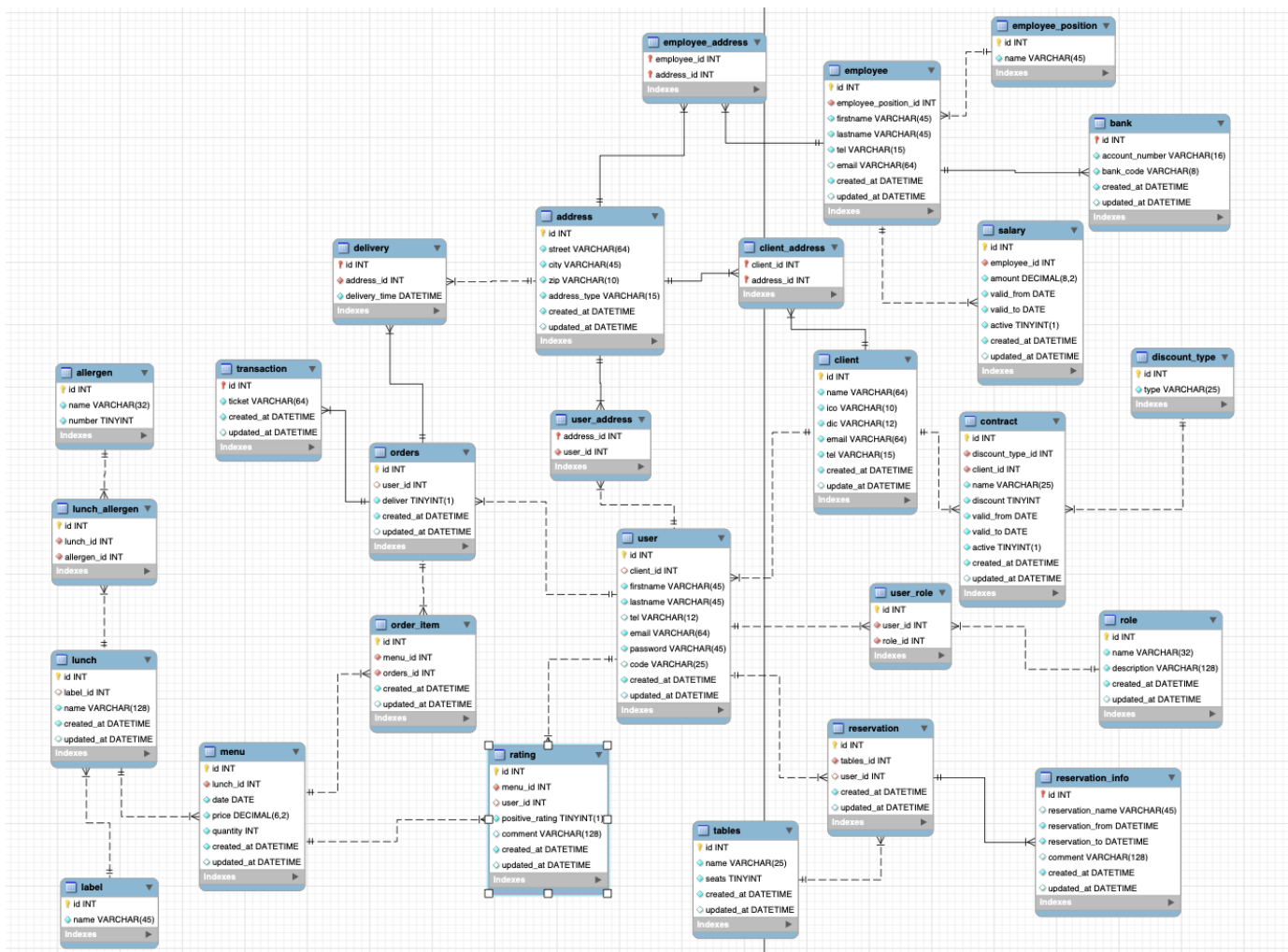
Tabulka 29 tables (Zdroj: vlastní zpracování)

Název atributu	Datový typ a délka	NOT NULL	Popis	Poznámka
id	int	Ano	Identifikace stolů	
name	varchar(25)	Ano	Jméno rezervace	
seats	tinyint	ano	Rezervace od	
created_at	datetime	Ano	Datum záznamu	Automaticky vložen
updated_at	datetime	ne	Datum poslední úpravy záznamu	

3.3.5 ER – diagram

Tato část zobrazuje Entity Relationship Diagram (ERD) navrženého databázového systému. ERD zahrnuje zobrazení všech tabulek databázového systému a identifikuje jejich primární/cizí klíče. Dále zobrazuje všechny atributy tabulky a jejich specifikace. V neposlední řadě vyznačuje i jednotlivé vazby mezi tabulkami.

V diagramu jsou rozmístěny tabulky, které mají v záhlaví svůj název. Pod názvem jsou již v tělu tabulky přiřazeny jednotlivé atributy a jejich nastavení. Primární klíč je označen symbolem klíče. Tento symbol může mít buď žlutou barvu a je to tedy pouze primární klíč a nebo barvu červenou, která značí, že je atribut klíčem primárním a zároveň cizím. Atributy, které nejsou primární klíč, mají barvu červenou, modrou nebo bílou. Bílá barva identifikuje atribut jako NULL, modrá barva naopak značí NOT NULL. Poslední červená barva označuje atribut jako cizí klíč. Vazby 1:N jsou vyznačené přerušovanou čarou. Plnou čarou jsou značeny vazby 1:1.



Obrázek 20 ER - diagram (Zdroj: vlastní zpracování)

3.4 Fyzický návrh

V této části budeme převádět navržený ERD do fyzického návrh. Fyzický návrh bude rozdělen do několika částí. Prvně si vytvoříme fyzické tabulky pomocí SQL a přiřadíme jim primární klíče. Následně vytvoříme cizí klíče a přiřadíme vazby. Poslední krok vytvoření databáze zahrnuje přiřazení jednotlivým tabulkám atributy a integritní omezení. V dalších krocích následně vytvoříme několik pohledů.

3.4.1 Tvorba tabulek

V první řadě si tedy vytvoříme pomocí příkazu CREATE TABLE všechny tabulky vyznačené v ER – diagramu. Při vytváření nastavíme také primární klíče a přiřadíme vlastnost AUTO_INCREMENT, díky které zajistíme jedinečnost atributu.

Obrázek níže zobrazuje vytvoření některých tabulek. Script pro vytvoření všech tabulek nalezneme v příloze č. 1.

```

1 • ⊖ CREATE TABLE client (
2     id int NOT NULL AUTO_INCREMENT,
3     PRIMARY KEY (id)
4 );
5
6 • ⊖ CREATE TABLE user (
7     id INT NOT NULL AUTO_INCREMENT,
8     PRIMARY KEY (id)
9 );
10
11 • ⊖ CREATE TABLE role (
12     id int NOT NULL AUTO_INCREMENT,
13     PRIMARY KEY (id)
14 );
15
16 • ⊖ CREATE TABLE user_role (
17     id int NOT NULL AUTO_INCREMENT,
18     PRIMARY KEY (id)
19 );
20
21 • ⊖ CREATE TABLE lunch (
22     id INT NOT NULL AUTO_INCREMENT,
23     PRIMARY KEY (id)
24 );
25
26 • ⊖ CREATE TABLE lunch_allergen (
27     id INT NOT NULL AUTO_INCREMENT,
28     PRIMARY KEY (id)
29 );
30

```

Obrázek 21 Vytvoření tabulek (Zdroj: vlastní zpracování)

3.4.2 Tvorba cizích klíčů a vazeb mezi tabulkami

Po vytvoření tabulek musíme zajistit relaci mezi nimi. Toho docílíme prostřednictvím atributů, kterým přiřadíme vlastnost `FOREIGN_KEY`, tedy cizí klíč. Podle potřeby přiřadíme navíc vlastnost `NULL` nebo `NOT NULL`. Cizí klíč je vždy pojmenovaný ve formátu: `nazevTabulky + '_id'`. Pokud tedy chceme propojit zaměstnance s jeho platem, vytvoříme cizí klíč `employee_id`.

Obrázek níže zobrazuje vytvoření některých cizích klíčů. Script pro tvorbu všech cizích klíčů nalezneme v příloze č. 2.

- `ALTER TABLE salary ADD employee_id INT NOT NULL;`
- `ALTER TABLE salary ADD FOREIGN KEY (employee_id) REFERENCES employee(id);`

- `ALTER TABLE client_address ADD FOREIGN KEY (address_id) REFERENCES address(id);`
- `ALTER TABLE client_address ADD FOREIGN KEY (client_id) REFERENCES client(id);`

- `ALTER TABLE employee ADD employee_position_id INT NOT NULL;`
- `ALTER TABLE employee ADD FOREIGN KEY (employee_position_id) REFERENCES employee_position(id);`

- `ALTER TABLE employee_address ADD FOREIGN KEY (address_id) REFERENCES address(id);`
- `ALTER TABLE employee_address ADD FOREIGN KEY (employee_id) REFERENCES employee(id);`

- `ALTER TABLE user_address ADD FOREIGN KEY (address_id) REFERENCES address(id);`
- `ALTER TABLE user_address ADD user_id INT NOT NULL;`
- `ALTER TABLE user_address ADD FOREIGN KEY (user_id) REFERENCES user(id);`

Obrázek 22 Vytvoření cizích klíčů (Zdroj: vlastní zpracování)

3.4.3 Tvorba ostatních atributů

V posledním kroku dojde k tvorbě všech zbylých atributů a přiřazení potřebných vlastností jako typ, délka nebo omezení NULL. Většina tabulek obsahuje atributy `created_at` a `updated_at`. Tyto atributy slouží k lepší přehlednosti databáze při editaci záznamů.

Obrázek níže zobrazuje vytvoření některých atributů. Script pro tvorbu všech atributů nalezneme v příloze č. 3.

```

200
207
208 • ALTER TABLE client
209   ADD name varchar (64) NOT NULL,
210   ADD ico varchar (10) NOT NULL,
211   ADD dic varchar (12) NOT NULL,
212   ADD email varchar (64) NOT NULL,
213   ADD tel varchar (15) NOT NULL,
214   ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
215   ADD update_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP,
216   ADD UNIQUE (email);
217
218 • ALTER TABLE user
219   ADD firstname varchar (45) NOT NULL,
220   ADD lastname varchar (45) NOT NULL,
221   ADD tel varchar (15) NULL,
222   ADD email varchar (64) NOT NULL,
223   ADD password varchar (45) NOT NULL,
224   ADD code varchar (25) NULL,
225   ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
226   ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
227

```

Obrázek 23 Vytvoření atributů (Zdroj: vlastní zpracování)

3.4.4 Pohledy

Před vytvořením pohledů si nejprve vytvoříme testovací data. Pro vkládání záznamů se používá příkaz INSERT INTO.

Obrázek níže zobrazuje vytvoření záznamu do tabulky address. Script pro tvorbu záznamů nalezneme v příloze č. 4.

```
INSERT INTO address (street, city, zip, address_type)
VALUES
('Náměstí 211/2', 'Brno', '104 21', 'kancelář'),
('Svobodova 12/2', 'Brno', '104 21', 'kancelář'),
('Otrokova 211/4', 'Brno', '104 21', 'Domů'),
('Náměstí 11/55', 'Brno', '104 21', 'Práce'),
('Hodonínská 51/2', 'Brno', '104 21', 'Práce')
;
```

Obrázek 24 Vložení testovacích dat (Zdroj: vlastní zpracování)

3.4.4.1 Pohled klientů

Tento pohled má za úkol zobrazí všechny klienty, kteří uzavřeli smlouvu s kantýnou. Pohled připojí potřebné tabulky, které mají společnou vazbu a vybere příkazem SELECT požadované údaje. Sloupec 'Počet strážníků' zobrazuje pomocí příkazu COUNT celkový počet strážníků přiřazených k jednotlivým klientům. Vedení firmy tak může mít jednoduchý přehled o všech klientech.

```
create view client_info as
SELECT cl.name as jméno, cl.ico as ico, co.name as kontrakt, co.discount as sleva,
dt.type as 'typ slevy',
COUNT(us.client_id) as 'Počet strážníků'
FROM client cl
INNER JOIN contract co on cl.id = co.client_id
INNER JOIN discount_type dt on dt.id = co.discount_type_id
INNER JOIN user us on us.client_id = cl.id
GROUP BY us.client_id
```

Výstup pohledu:

<input type="checkbox"/> Změnit	jméno	ico	kontrakt	sleva	typ slevy	Počet strážníků
<input type="checkbox"/> upravit	Soft company	42325623	Soft smlouva - VIP	20	procento	1
<input type="checkbox"/> upravit	Stavebniny Novotný	56228923	basic-stavby	15	množství	2

Obrázek 25 Výstup pohledu klient (Zdroj: vlastní zpracování)

3.4.4.2 Pohled menu

Pohled názvem `menu_overview` nám díky podmínce `WHERE` vyhledá všechny položky v menu, kde je atribut `date` roven dnešnímu dni. Poté se připojí potřebné tabulky a vypíše se požadované údaje. Tento pohled bude implementován na webové stránky, aby se mohli strážníci každý den podívat na aktuální menu jídelny.

```
create view menu_overview as
SELECT la.name as štítek, lu.name as jídlo, me.date as den, me.price as cena,
me.quantity as Množství
FROM menu me
INNER JOIN lunch lu on lu.id = me.lunch_id
INNER JOIN label la on lu.label_id = la.id
WHERE me.date = curdate()
```

Výstup pohled:

štítek	jídlo	den	cena	Množství
P	Čočková polévka	2022-05-09	35.00	180
M1	Řízek s bramborem	2022-05-09	119.00	250

Obrázek 26 Výstup pohledu menu

3.4.4.3 Pohled objednávky

Poslední vytvořený pohled slouží pro zobrazení všech objednávek. Objednávky jsou řazeny v sestupném pořadí. Pohled bude zase sloužit vedení firmy pro analýzu prodeje.

```
create view orders_overview as
SELECT lu.name as jídlo, m.date as datum, m.price as cena,
replace(replace(o.deliver, 0, 'NE'),1,'ANO') as
'Dovoz'
FROM order_item oi
INNER JOIN menu m on m.id = oi.menu_id
INNER JOIN lunch lu on lu.id = m.lunch_id
INNER JOIN orders o on o.id = oi.orders_id
order by date desc
```


jídlo	datum	cena	Dovoz
Řízek s bramborem	2022-05-09	119.00	NE
Řízek s bramborem	2022-05-09	119.00	NE
Čočková polévka	2022-05-09	35.00	NE
Čočková polévka	2022-05-09	35.00	NE
Rizoto, okurek	2022-04-20	99.00	NE
Rizoto, okurek	2022-04-20	99.00	NE

Obrázek 27 Výstup pohledu objednávky (Zdroj: vlastní zpracování)

3.5 Zhodnocení návrhu a přínos pro firmu

V rámci této bakalářské práce byl analyzován současný stav firmy, její procesy a požadavky. Z této analýzy byl vypracován databázový systém, který pokrývá všechny potřebné procesy a požadavky vedení. Hlavní myšlenkou vedení bylo začít efektivně a přehledně ukládat data pro účely nových webových stránek. Databázový systém je tedy připraven pro implementaci k aplikačnímu rozhraní a je navržen tak, aby se v případě potřeby mohly vytvořit další komponenty, které budou napojeny na databázi.

Zatím není možné analyzovat přínosy databázového systému, protože se nejdříve musí naprogramovat nové webové stránky, které budou napojeny na vytvořenou databázi. Jelikož firma doposud nedisponovala žádnou databází a dynamickou webovou stránkou, můžeme usoudit, že přínos pro firmu bude významný.

Možné funkcionality díky vytvořené databázi:

- **Objednávky** – Firma bude mít možnost uchovávat nejrůznější data o všech objednávkách a díky tomu bude moct v interním rozhraní stránek sestavovat nejrůznější statistické grafy a pohledy.
- **Rezervace** – Díky tabulkám spadající pod modul rezervace bude možnost vést data o rezervacích a vytvářet nové rezervace online.
- **Dovoz** – Další důležitou funkcionalitou je dovoz obědů do práce či domů, díky online objednávce.
- **Přehled subjektů** – Databáze ukládá data o zaměstnancích, klientech a strážnících. Díky tomu bude mít firma detailní přehled o všech relevantních subjektech.
- **Jiné** – Tvorba nejrůznějších grafů, report a sledování výkonnosti firmy.

Závěr

Cílem této bakalářské práce bylo navržení a vytvoření databáze pro nově vznikající webové stránky firmy působící na poli gastronomie. Návrh byl vytvořen podle analýzy současného stavu a kritériích zadaných vedením a reflektuje všechny potřebné funkce.

V první části práce se věnujeme teorii, která vysvětluje všechny potřebné informace nutné k porozumění principů tvorby databázového systému. Kromě databázové teorie vysvětluje i použité analýzy.

V druhé části se věnujeme analýze současného stavu, kde je prvně představena firma a organizační struktura. Následně je popsána současná situace a jednotlivé procesy ve firmě. Jako poslední byla provedena SWOT analýza.

V poslední části je sestaven vlastní návrh řešení podle předchozích provedených analýz. V první řadě byly sepsány některé požadované procesy, které budou implemetovány na webové stránky. Následně je popsán celý návrh systému ve třech fázích. V první fázi již došlo k vytvoření konceptuálního návrhu, kde jsou identifikované entity a jejich relace. V druhé fázi byl řešen logický návrh databáze. Zde byly popsány jednotlivé tabulky s definovanými parametry a následně ukázan ER - diagram. V poslední fázi došlo už k fyzickému tvoření jednotlivých částí databáze. Prvně byly vytvořeny tabulky s primárním klíčem, poté byly přidány cizí klíče a tím se vytvořila relace. Poslední krok tvorby databáze bylo vytvoření zbylých potřebných atributů a konkrétních pohledů. Návrh databáze je připraven pro implementaci k webové aplikaci.

SEZNAM POUŽITÝCH ZDROJŮ

- (1) BASL, J. a R. BLAŽÍČEK. Podnikové informační systémy - podnik v informační společnosti. 3. vyd. Praha: Grada Publishing a.s., 2012. 328 s. ISBN 978-80-247-4307-3.
- (2) CONOLLY, T., C. E. BEGG a R. HOLOWCZAK. Mistrovství databáze: profesionální průvodce tvorbou efektivních databází. 1. vyd. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.
- (3) KOCH, M. a B. NEUWIRTH. Datové a funkční modelování. 4. rozš. vyd. Brno: Akademické nakladatelství CERM, 2010. 139 s. ISBN 978-80-214-4125-5.
- (4) NOVOTNÝ, O., J. POUR a D. SLÁNSKÝ. Business Intelligence: jak využít bohatství ve vašich datech. 1. vyd. Praha: Grada, 2005. 256 s. ISBN 80-247-1094-3.
- (5) Data, informace a cesta ke znalostem | Informační gramotnost | Tipy & triky ze světa informací. Portál Informační gramotnost | Data, informace a znalosti [online]. Copyright © Vydavatel on [cit. 09.05.2022]. Dostupné z: <https://www.informacnigramotnost.cz/data-informace-znalosti/>
- (6) KROENKE, David a David J. AUER. Databáze. 1. vyd. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.
- (7) KŘÍŽ, Jiří a Petr DOSTÁL. Databázové systémy. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3064-8.
- (8) Lekce 3 - Třívrstvá architektura a další vícevrstvé architektury. itnetwork.cz - Učíme národ IT [online]. Copyright © 2022 itnetwork.cz. Veškerý obsah webu [cit. 09.05.2022]. Dostupné z: <https://www.itnetwork.cz/navrh/architektury-a-dependency-injection/trivrstva-architektura-a-dalsi-vicevrstve-architektury>
- (9) [online]. Dostupné z: <https://www.scaler.com/topics/dbms/dbms-architecture/>
- (10) Popis normalizace databáze - Office | Microsoft Docs. [online]. Copyright © Microsoft 2022 [cit. 09.05.2022]. Dostupné z: <https://docs.microsoft.com/cscz/office/troubleshoot/access/database-normalization-description>
- (11) What is Structured Query Language (SQL)?. Purchase Intent Data for Enterprise Tech Sales and Marketing - TechTarget [online]. Dostupné z: <https://www.techtarget.com/searchdatamanagement/definition/SQL>
- (12) SWOT analýza - ManagementMania.com. [online]. Copyright © 2011 [cit. 09.05.2022]. Dostupné z: <https://managementmania.com/cs/swot-analyza>

Seznam použitých zkratk

DBMS	Database Management System
DCL	Data Query Language
DDL	Data Definition Language
DQL	Data Query Language
DML	Data Manipulation Language
PK	Primary key
FK	Foreign Key
SQL	Structured Query Language
ID	Identifikační číslo
DB	Databáze
ERD	Entity Relationship Diagram

SEZNAM POUŽITÝCH OBRÁZKŮ

Obrázek 1 Informace (Zdroj: Vlastní zpracování podle 3, s.4)	10
Obrázek 2 Databázový systém (Zdroj: Vlastní zpracování podle 1, s.30)	11
Obrázek 3 Funkce DBMS (Zdroj: Vlastní zpracování podle 1, s.31)	12
Obrázek 4 Trojvrstvá architektura (Zdroj: Vlastní zpracování)	13
Obrázek 5 Lineární model (Zdroj: Vlastní zpracování dle: 3, s 20).....	14
Obrázek 6 Relační model (Zdroj: Vlastní zpracování dle: 3, s 21)	15
Obrázek 7 Datový model (Zdroj: Vlastní zpracování dle: 3, s 22)	15
Obrázek 8 Vlastnosti relace (Zdroj: Vlastní zpracování dle: 6, s 78)	16
Obrázek 9 relace 1:1 (Zdroj: vlastní zpracování podle 3, s. 31-32)	17
Obrázek 10 relace 1:N (Zdroj: vlastní zpracování podle 3, s. 31-32)	18
Obrázek 11 relace N:M (Zdroj: vlastní zpracování podle 3, s. 31-32).....	18
Obrázek 12 SWOT (Zdroj: vlastní zpracování podle 12)	22
Obrázek 13 struktura zaměstnanců (Zdroj: vlastní zpracování).....	24
Obrázek 14 Typ stránek (Zdroj: vlastní zpracování)	25
Obrázek 15 SWOT analýza (Zdroj: vlastní zpracování)	27
Obrázek 16 Uživatelský pohled procesu objednávek (Zdroj: vlastní zpracování).....	29
Obrázek 17 Vývojový diagram procesu objednávek (Zdroj: vlastní zpracování).....	30
Obrázek 18 Uživatelský pohled procesu rezervace (Zdroj: vlastní zpracování).....	31
Obrázek 19 Vývojový diagram procesu rezervace (Zdroj: vlastní zpracování).....	32
Obrázek 20 ER - diagram (Zdroj: vlastní zpracování)	49
Obrázek 21 Vytvoření tabulek (Zdroj: vlastní zpracování).....	50
Obrázek 22 Vytvoření cizích klíčů (Zdroj: vlastní zpracování)	51
Obrázek 23 Vytvoření atributů (Zdroj: vlastní zpracování)	51
Obrázek 24 Vložení testovacích dat (Zdroj: vlastní zpracování)	52
Obrázek 25 Výstup pohledu klient (Zdroj: vlastní zpracování)	52
Obrázek 26 Výstup pohledu menu	53
Obrázek 27 Výstup pohledu objednávky (Zdroj: vlastní zpracování).....	54

Seznam použitých tabulek

Tabulka 1 Identifikace entit (Zdroj: vlastní zpracování)	33
Tabulka 2 Identifikace relací (Zdroj: vlastní zpracování)	34
Tabulka 3 client	35
Tabulka 4 client_address (Zdroj: vlastní zpracování)	35
Tabulka 5 address (Zdroj: vlastní zpracování)	36
Tabulka 6 contract (Zdroj: vlastní zpracování)	36
Tabulka 7 discount_type (Zdroj: vlastní zpracování).....	37
Tabulka 8 employee (Zdroj: vlastní zpracování).....	37
Tabulka 9 employy_position (Zdroj: vlastní zpracování)	38
Tabulka 10 employee_address (Zdroj: vlastní zpracování).....	38
Tabulka 11 bank (Zdroj: vlastní zpracování)	39
Tabulka 12 salary (Zdroj: vlastní zpracování).....	39
Tabulka 13 user (Zdroj: vlastní zpracování)	40
Tabulka 14 user_address (Zdroj: vlastní zpracování)	40
Tabulka 15 user_role (Zdroj: vlastní zpracování)	41
Tabulka 16 role (Zdroj: vlastní zpracování).....	41
Tabulka 17 orders (Zdroj: vlastní zpracování)	42
Tabulka 18 order_item (Zdroj: vlastní zpracování).....	42
Tabulka 19 transaction (Zdroj: vlastní zpracování).....	43
Tabulka 20 delivery (Zdroj: vlastní zpracování)	43
Tabulka 21 lunch (Zdroj: vlastní zpracování)	44
Tabulka 22 lunch_allergen (Zdroj: vlastní zpracování)	44
Tabulka 23 allergen (Zdroj: vlastní zpracování)	45
Tabulka 24 label (Zdroj: vlastní zpracování)	45
Tabulka 25 menu (Zdroj: vlastní zpracování)	45
Tabulka 26 rating (Zdroj: vlastní zpracování).....	46
Tabulka 27 reservation (Zdroj: vlastní zpracování)	47
Tabulka 28 reservation_info.....	47
Tabulka 29 tables (Zdroj: vlastní zpracování).....	48

Seznam příloh

Příloha č. 1: Tvorba tabulek

Příloha č. 2: Tvorba relací

Příloha č. 3: Tvorba atributů

Příloha č. 4: Tvorba testovacích dat pro pohledy

Příloha č. 5: Tvorba pohledů

Příloha č. 1 Tvorba tabulek (vlastní zpracování)

```
CREATE TABLE client (  
    id int NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE user (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE role (  
    id int NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE user_role (  
    id int NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE lunch (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE lunch_allergen (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE label (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id));  
  
CREATE TABLE employee (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE reservation (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE orders (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE order_item (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id));  
  
CREATE TABLE rating (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE address (  
    id INT NOT NULL AUTO INCREMENT,  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE delivery (  
    id INT NOT NULL,  
    PRIMARY KEY (id));  
  
CREATE TABLE client_address(  
    client_id INT NOT NULL,
```



```

address_id INT NOT NULL,
PRIMARY KEY (client_id, address_id)
);

CREATE TABLE employee address (
employee_id INT NOT NULL,
address_id INT NOT NULL,
PRIMARY KEY (employee_id, address_id)
);

CREATE TABLE user address (
address_id INT NOT NULL,
PRIMARY KEY (address_id)
);

CREATE TABLE allergen (
id INT NOT NULL AUTO INCREMENT,
PRIMARY KEY (id)
);

CREATE TABLE bank (
id INT NOT NULL,
PRIMARY KEY (id)
);

CREATE TABLE transaction (
id INT NOT NULL,
PRIMARY KEY (id)
);

CREATE TABLE menu (
id INT NOT NULL AUTO INCREMENT,
PRIMARY KEY (id)
);

CREATE TABLE salary (
id INT NOT NULL AUTO INCREMENT,
PRIMARY KEY (id)
);

CREATE TABLE tables (
id INT NOT NULL AUTO INCREMENT,
PRIMARY KEY (id)
);

CREATE TABLE employee position (
id INT NOT NULL AUTO INCREMENT,
PRIMARY KEY (id)
);

CREATE TABLE contract (
id INT NOT NULL AUTO INCREMENT,
PRIMARY KEY (id)
);

CREATE TABLE discount type (
id INT NOT NULL AUTO INCREMENT,
PRIMARY KEY (id)
);

CREATE TABLE reservation info (
id INT NOT NULL AUTO INCREMENT,
PRIMARY KEY (id)
);

```

Příloha č. 2: Tvorba relací (vlastní zpracování)

```
ALTER TABLE salary ADD employee_id INT NOT NULL;
ALTER TABLE salary ADD FOREIGN KEY (employee_id) REFERENCES employee(id);

ALTER TABLE client_address ADD FOREIGN KEY (address_id) REFERENCES address(id);
ALTER TABLE client_address ADD FOREIGN KEY (client_id) REFERENCES client(id);

ALTER TABLE employee ADD employee_position_id INT NOT NULL;
ALTER TABLE employee ADD FOREIGN KEY (employee_position_id) REFERENCES
employee_position(id);

ALTER TABLE employee_address ADD FOREIGN KEY (address_id) REFERENCES address(id);
ALTER TABLE employee_address ADD FOREIGN KEY (employee_id) REFERENCES employee(id);

ALTER TABLE user_address ADD FOREIGN KEY (address_id) REFERENCES address(id);
ALTER TABLE user_address ADD user_id INT NOT NULL;
ALTER TABLE user_address ADD FOREIGN KEY (user_id) REFERENCES user(id);

ALTER TABLE menu ADD lunch_id INT NOT NULL;
ALTER TABLE menu ADD FOREIGN KEY (lunch_id) REFERENCES lunch(id);

ALTER TABLE user ADD client_id INT DEFAULT NULL;
ALTER TABLE user ADD FOREIGN KEY (client_id) REFERENCES client(id);

ALTER TABLE delivery ADD address_id INT NOT NULL;
ALTER TABLE delivery ADD FOREIGN KEY (address_id) REFERENCES address(id);
ALTER TABLE delivery ADD FOREIGN KEY (id) REFERENCES orders(id);

ALTER TABLE order_item ADD menu_id INT NOT NULL;
ALTER TABLE order_item ADD FOREIGN KEY (menu_id) REFERENCES menu(id);
ALTER TABLE order_item ADD orders_id INT NOT NULL;
ALTER TABLE order_item ADD FOREIGN KEY (orders_id) REFERENCES orders(id);

ALTER TABLE orders ADD user_id INT DEFAULT NULL;
ALTER TABLE orders ADD FOREIGN KEY (user_id) REFERENCES user(id);

ALTER TABLE transaction ADD FOREIGN KEY (id) REFERENCES orders(id);

ALTER TABLE bank ADD FOREIGN KEY (id) REFERENCES employee(id);

ALTER TABLE reservation_info ADD FOREIGN KEY (id) REFERENCES reservation(id);

ALTER TABLE lunch ADD label_id INT DEFAULT NULL;
ALTER TABLE lunch ADD FOREIGN KEY (label_id) REFERENCES label(id);

ALTER TABLE lunch_allergen ADD lunch_id INT NOT NULL;
ALTER TABLE lunch_allergen ADD FOREIGN KEY (lunch_id) REFERENCES lunch(id);
ALTER TABLE lunch_allergen ADD allergen_id INT NOT NULL;
ALTER TABLE lunch_allergen ADD FOREIGN KEY (allergen_id) REFERENCES allergen(id);

ALTER TABLE rating ADD menu_id INT NOT NULL;
ALTER TABLE rating ADD FOREIGN KEY (menu_id) REFERENCES menu(id);
ALTER TABLE rating ADD user_id INT DEFAULT NULL;
ALTER TABLE rating ADD FOREIGN KEY (user_id) REFERENCES user(id);

ALTER TABLE user_role ADD user_id INT NOT NULL;
ALTER TABLE user_role ADD FOREIGN KEY (user_id) REFERENCES user(id);
ALTER TABLE user_role ADD role_id INT NOT NULL;
ALTER TABLE user_role ADD FOREIGN KEY (role_id) REFERENCES role(id);

ALTER TABLE reservation ADD tables_id INT NOT NULL;
ALTER TABLE reservation ADD FOREIGN KEY (tables_id) REFERENCES tables(id);
```

```
ALTER TABLE reservation ADD user_id INT DEFAULT NULL;  
ALTER TABLE reservation ADD FOREIGN KEY (user_id) REFERENCES user(id);  
  
ALTER TABLE contract ADD discount_type_id INT NOT NULL;  
ALTER TABLE contract ADD FOREIGN KEY (discount_type_id) REFERENCES  
discount_type(id);  
ALTER TABLE contract ADD client_id INT NOT NULL;  
ALTER TABLE contract ADD FOREIGN KEY (client_id) REFERENCES client(id);
```

Příloha č. 3: Tvorba atributů (vlastní zpracování)

```
ALTER TABLE client
ADD name varchar (64) NOT NULL,
ADD ico varchar (10) NOT NULL,
ADD dic varchar (12) NOT NULL,
ADD email varchar (64) NOT NULL,
ADD tel varchar (15) NOT NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT TIMESTAMP,
ADD update_at DATETIME NULL ON UPDATE CURRENT TIMESTAMP,
ADD UNIQUE (email);

ALTER TABLE user
ADD firstname varchar (45) NOT NULL,
ADD lastname varchar (45) NOT NULL,
ADD tel varchar (15) NULL,
ADD email varchar (64) NOT NULL,
ADD password varchar (45) NOT NULL,
ADD code varchar (25) NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT TIMESTAMP;

ALTER TABLE employee
ADD firstname varchar (45) NOT NULL,
ADD lastname varchar (45) NOT NULL,
ADD tel varchar (15) NOT NULL,
ADD email varchar (64) NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT TIMESTAMP;

ALTER TABLE employee position
ADD name varchar (45) NOT NULL;

ALTER TABLE bank
ADD account_number VARCHAR (16) NOT NULL,
ADD bank_code VARCHAR (8) NOT NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT TIMESTAMP;

ALTER TABLE role
ADD name VARCHAR (32) NOT NULL,
ADD description VARCHAR (128) NOT NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT TIMESTAMP;

ALTER TABLE lunch
ADD name varchar (128) NOT NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT TIMESTAMP;

ALTER TABLE allergen
ADD name varchar (32) NOT NULL,
ADD number tinyint NOT NULL;

ALTER TABLE tables
ADD name varchar (25) NOT NULL,
ADD seats tinyint NOT NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT TIMESTAMP;

ALTER TABLE contract
ADD name varchar (25) NOT NULL,
ADD discount tinyint NOT NULL,
ADD valid_from DATE NOT NULL,
ADD valid_to DATE NOT NULL,
ADD active BOOLEAN NOT NULL DEFAULT true,
```

```
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE salary

```
ADD amount decimal(8,2) NOT NULL,
ADD valid_from DATE NOT NULL,
ADD valid_to DATE NOT NULL,
ADD active BOOLEAN NOT NULL DEFAULT true,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE address

```
ADD street varchar(64) NOT NULL,
ADD city varchar(45) NOT NULL,
ADD zip varchar(10) NOT NULL,
ADD address_type varchar (15) NOT NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE menu

```
ADD date DATE NOT NULL,
ADD price decimal(6, 2) NOT NULL,
ADD quantity INT NOT NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE orders

```
ADD deliver BOOLEAN NOT NULL DEFAULT false,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE delivery

```
ADD delivery_time DATETIME NOT NULL;
```

ALTER TABLE order item

```
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE rating

```
ADD positive_rating BOOLEAN NOT NULL,
ADD comment varchar (128) NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE reservation

```
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE label

```
ADD name VARCHAR (45) NOT NULL;
```

ALTER TABLE reservation info

```
ADD reservation_name varchar (45) NULL,
ADD reservation_from DATETIME NOT NULL,
ADD reservation_to DATETIME NOT NULL,
ADD comment varchar (128) NULL,
ADD tel varchar (15) NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL ON UPDATE CURRENT_TIMESTAMP;
```

ALTER TABLE transaction

```
ADD ticket VARCHAR (64) NOT NULL,
ADD created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
ADD updated_at DATETIME NULL;
```

ALTER TABLE discount type

```
ADD type varchar(25) NOT NULL;
```

Příloha č. 4: Tvorba testovacích dat pro pohledy (vlastní zpracování)

```
INSERT INTO address (street, city, zip, address_type)
VALUES
('Náměstí 211/2', 'Brno', '104 21', 'kancelář'),
('Svobodova 12/2', 'Brno', '104 21', 'kancelář'),
('Otrokova 211/4', 'Brno', '104 21', 'Domů'),
('Náměstí 11/55', 'Brno', '104 21', 'Práce'),
('Hodonínská 51/2', 'Brno', '104 21', 'Práce')
;

INSERT INTO client (name, ico, dic, email, tel)
VALUES
('Soft company', '42325623', 'cz42325623', 'soft.eral@gmail.com', '432445325'),
('Stavebniny Novotný', '56228923', 'cz56228923', 'stavby.nov@gmail.com',
'432278345')
;

INSERT INTO client_address (client_id, address_id)
VALUES
(1, 1),
(2,2)
;

INSERT INTO discount_type (type)
VALUES
('procento'),
('množství')
;

INSERT INTO contract (discount_type_id, client_id, name, discount, valid_from,
valid to, active)
VALUES
(1, 1, 'Soft smlouva - VIP', 20, '2021-02-02', '2023-02-02', 1),
(2, 2, 'basic-stavby', 15, '2021-05-02', '2023-05-02', 1)
;

INSERT INTO user (client_id, firstname, lastname, email, password, code)
VALUES
(1, 'Pavel', 'Dvorak', 'pavel@seznam.cz', 'sad324dsa123f', '12sq4123ffds2'),
(2, 'Honza', 'Navratil', 'nara2@seznam.cz', 'rwerw214314', '13q41qwe23fds2'),
(2, 'Michal', 'Simek', 'michall.2@seznam.cz', '41sadd3423', '112453ffds2')
;

INSERT INTO user_address (address_id, user_id)
VALUES
(3,1),
(4,2),
(5,2)
;

INSERT INTO label (name)
VALUES
('P'),
('M1'),
('M2')
;

INSERT INTO lunch (label_id, name)
VALUES
(1, 'Čočková polévka'),
(2, 'Řízek s bramborem'),
(3, 'Rizoto, okurek')
;
```

```
INSERT INTO menu(lunch_id, date, price, quantity)
VALUES
(1, '2022-04-20', 35.00, 180),
(2, '2022-04-21', 119.00, 250),
(3, '2022-04-20', 99.0, 110);

INSERT INTO orders (user_id, deliver)
VALUES
(1, 0),
(1, 0),
(2, 0)
;

INSERT INTO order item (menu_id, orders_id)
VALUES
(1, 1),
(2, 1),
(3, 1),
(1, 1),
(2, 2),
(3, 2)
;
```

Příloha č. 5: Tvorba pohledů (vlastní zpracování)

```
create view client_info as
SELECT cl.name as jméno, cl.ico as ico, co.name as kontrakt, co.discount as sleva,
dt.type as 'typ slevy',
COUNT(us.client_id) as 'Počet strážníků'
FROM client cl
INNER JOIN contract co on cl.id = co.client_id
INNER JOIN discount type dt on dt.id = co.discount_type_id
INNER JOIN user us on us.client_id = cl.id
GROUP BY us.client_id

create view menu_overview as
SELECT la.name as štítek, lu.name as jídlo, me.date as den, me.price as cena,
me.quantity as Množství
FROM menu me
INNER JOIN lunch lu on lu.id = me.lunch_id
INNER JOIN label la on lu.label_id = la.id
WHERE me.date = curdate()

create view orders_overview as
SELECT lu.name as jídlo, m.date as datum, m.price as cena,
replace(replace(o.deliver, 0, 'NE'),1,'ANO') as
'Dovoz'
FROM order item oi
INNER JOIN menu m on m.id = oi.menu_id
INNER JOIN lunch lu on lu.id = m.lunch_id
INNER JOIN orders o on o.id = oi.orders_id
order by date desc
```