



VYSOKÉ UČENÍ TECHNICKÉ
V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH SQL DATABÁZE PRO ELEKTRONICKÝ OBCHOD FIRMY

PROPOSAL OF SQL DATABASE FOR E-SHOP IN COMPANY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ SMEČKA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. JIŘÍ KŘÍŽ, PH.D.

BRNO 2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Smečka Jiří

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh SQL databáze pro elektronický obchod firmy

v anglickém jazyce:

Proposal of SQL Database for E-shop in Company

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

CONOLLY, T., C. BEGG a R. HALOWCZAK. Mistrovství – databáze: Profesionální průvodce tvorbou efektivních databází. 1. vyd. Brno: Computer Press, 2009, 584 s. ISBN 978-80-251-2328-7.

HOTEK, M. Microsoft SQL Server 2008: Krok za krokem. 1. vyd. Brno: Computer Press, 2009, 488 s. ISBN 978-80-251-2466-6.

KOCH, M. a B. NEUWIRTH. Datové a funkční modelování. 4. rozš. vyd. Brno: Akademické nakladatelství CERM, 2010, 142 s. ISBN 978-80-214-4125-5.

LACKO, L. 1001 tipů a triků pro SQL. 1. vyd. Brno: Computer Press, 2011, 416 s. ISBN 978-80-251-3010-0.

OPPEL, J. A. Databáze bez předchozích znalostí. 1. vyd. Brno: Computer Press, 2006, 316 s. ISBN 80-251-1199-7.

OPPEL, J. A. SQL bez předchozích znalostí. 1. vyd. Brno: Computer Press, 2012, 240 s. ISBN 978-80-251-1707-1.

Vedoucí bakalářské práce: Ing. Jiří Kříž, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/2013.

L.S.

doc. RNDr. Bedřich Půža, CSc.
Ředitel ústavu

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Děkan fakulty

V Brně, dne 19.05.2013

Abstrakt

Tato bakalářská práce obsahuje návrh a vytvoření SQL databáze pro elektronický obchod firmy. Teoretická část poskytuje základní informace pro úspěšné zvládnutí cílů této bakalářské práce. Údaje o firmě se nacházejí v kapitole analýza současného stavu. Praktická část se zaměřuje na vlastní návrh řešení a následně popisuje přínosy tohoto návrhu.

Abstract

This bachelor thesis contains proposal and creation of SQL database for e-shop in company. Theoretical part provides basic information to successfully accomplish goals of this bachelor thesis. Information about company can be found in chapter current state analysis. Practical part focuses on a proposal of solution and subsequently describes benefits of this proposal.

Klíčová slova

Databáze, SQL, normalizace, eshop, lékárna

Key words

Database, SQL, normalization, eshop, pharmacy

Bibliografická citace práce

SMEČKA, J. *Návrh SQL databáze pro elektronický obchod firmy*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2013. 64 s. Vedoucí bakalářské práce
Ing. Jiří 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 31. května 2013

.....

Jiří Smečka

Obsah

Úvod.....	10
1 Vymezení problému a cíle práce	11
2 Teoretická východiska práce	12
2.1 Terminologie	12
2.1.1 Informace	12
2.1.2 Data	13
2.1.3 Znalosti	13
2.1.4 DBMS.....	13
2.1.5 Databáze.....	13
2.1.6 Entita	14
2.1.7 Atribut	14
2.1.8 Vazby mezi entitami	14
2.1.9 Relace	15
2.1.10 Normalizace.....	15
2.1.11 Datové modely	15
2.1.12 Lineární datový model	16
2.1.13 Hierarchický datový model	16
2.1.14 Síťový datový model	16
2.1.15 Relační datový model	16
2.1.16 Objektový datový model	16
2.1.17 Relační databáze.....	17
2.1.18 Klíče relace	17
2.1.19 Integrita relačního modelu	17
2.1.20 ER - model	18
2.2 Proces návrhu databáze	18

2.2.1 Návrh databáze	18
2.2.2 Logický návrh databáze	19
2.2.3 Fyzický návrh databáze	19
2.3 Jazyk SQL	20
2.3.1 Historie jazyka SQL	20
2.4 Microsoft SQL Server	21
2.4.1 Historie SQL Serveru	21
2.4.2 Datové typy SQL Serveru	22
3 Analýza současného stavu	23
3.1 Základní informace o firmě	23
3.2 Historie firmy	23
3.3 Organizační struktura	24
3.4 Konkurence	25
3.5 Zákazníci	25
3.6 Hardware	25
3.7 Software	28
3.8 SWOT analýza	32
4 Návrh vlastního řešení, přínos návrhu řešení	33
4.1 Požadavky na databázi	33
4.1.1 Dílčí požadavky	33
4.2 Identifikace základních procesů v databázi	35
4.2.1 Evidence nového zboží	35
4.2.2 Registrace nového zákazníka	35
4.2.3 Vytvoření objednávky	35
4.3 Zvolené DBMS	39
4.4 Konceptuální návrh databáze	39
4.4.1 Identifikace základních entit a definování atributů entit	39

4.4.2 Určení vzájemných vztahů mezi entitami	41
4.4.3 Základní ER model	42
4.5 Logický návrh databáze	43
4.5.1 Rozklad vazeb M:N	43
4.5.2 Vytvoření tabulek	44
4.6 Fyzický návrh databáze.....	50
4.6.1 Vyhledávání zboží.....	50
4.6.2 Registrace nového zákazníka	51
4.6.3 Změna sazby DPH	53
4.6.4. Změna ceny zboží	53
4.6.5 Upozornění na novou objednávku.....	54
4.7 Postup po provedení návrhu	55
4.8 Přínosy řešení	57
5 Závěr	58
Seznam použitých zdrojů.....	59
Knižní zdroje.....	59
Internetové zdroje.....	60
Seznam obrázků a tabulek	61
Seznam obrázků	61
Seznam tabulek.....	62
Seznam použitých zkratk	63
Seznam příloh	64

Úvod

Je mnoho způsobů jak uchovávat data. Ať už se jedná o papírové tabulky, elektronické diáře nebo SQL databáze, jsou na tato úložiště kladeny určité požadavky. Tyto požadavky pak definují podobu, jak bude databáze zpracována. Elektronický způsob uchování dat má nespočet výhod (rychlý přístup k datům, přehledné, snadno zálohovatelné, atd.) oproti svému papírovému předchůdci (za předpokladu, že je databáze dobře navržena).

Elektronické obchodování zažívá v posledních letech značný vzestup. Existuje mnoho firem, které obchodují pouze elektronicky a nemají kamennou pobočku. Není pro ně výhodné platit výdaje spojené s provozem takové pobočky, a přesto jsou to funkční firmy. Relativně nízké náklady a potenciálně široká klientela jsou lákadlem pro mnoho lidí či firem.

SQL databáze pro elektronický obchod firmy bude zpracována k rozšíření činnosti firmy a zacílení na nové zákazníky.

Řešení návrhu SQL databáze pro elektronický obchod bude zpracováno v programu Microsoft SQL 2012.

1 Vymezení problému a cíle práce

Všechny důležité informace o produktech, objednávkách a registrovaných zákaznících v rámci elektronického obchodu je potřeba mít zaznamenané. Pro tento účel slouží SQL databáze. Ať už do ní data vložíme jakkoliv, je nutné se držet určitých zásad návrhu SQL databází.

Cílem práce je vytvoření SQL databáze, která bude základem pro budoucí elektronický obchod firmy. Ke splnění tohoto cíle je potřebné získat teoretické znalosti a použít je k návrhu. Zpracován bude také odhad ekonomického přínosu firmě. Databáze musí být funkční a použitelná.

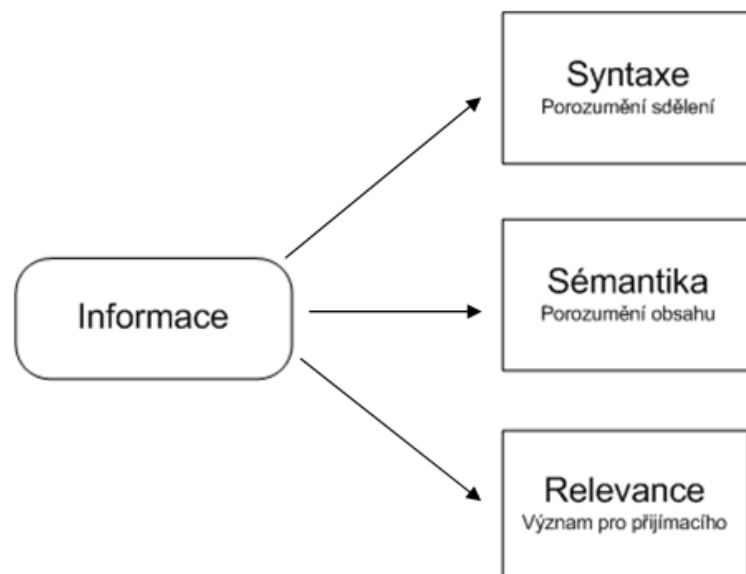
2 Teoretická východiska práce

V této části bakalářské práce je zpracováno teoretické pozadí potřebné k porozumění dané problematiky a úspěšnému zvládnutí praktické části práce.

2.1 Terminologie

2.1.1 Informace

„Informaci můžeme chápat jako zprávu, vjem, který splňuje tři požadavky. Prvním je syntaktická relevance. Subjekt, který zprávu přijímá, musí být schopen ji detekovat a rozumět jí. Druhým požadavkem je sémantická relevance. Subjekt musí vědět, co zpráva znamená, co vypovídá o něm a jeho okolí. Třetím požadavkem je pragmatická relevance. Zpráva musí mít pro přijímací subjekt nějaký význam.“ (3, s. 4)



Obr. 1: Požadavky na informaci (3, s. 4)

Informace nám pomáhají při rozhodování. Čím více informací máme k dispozici, tím snazší je pro nás učinit nějaké rozhodnutí.

2.1.2 Data

Pokud pro nás data mají nějaký význam a smysl, tzn. můžeme je použít k rozhodování, jsou tato data pro nás informací. Data můžeme uložit pro pozdější zpracování nebo je převést do jiné podoby (uložit do počítače, napsat na papír). V okamžiku, kdy zaznamenáme informaci na médium (tento proces se nazývá kódování informace) se z informace stávají data. Opačný proces se nazývá dekódování - z dat se stává opět informace. (3)

Procesy kódování a dekódování mohou být problematické, protože ten, komu jsou data určena, nemusí být schopen je dekódovat.

2.1.3 Znalosti

„Znalosti lze charakterizovat také jako informace o tom, jak využít jiné informace a data (a to i ve vzájemných kombinacích) v různých situacích. ” (3, str. 6)

Například pracujeme na technické podpoře a přijde za námi člověk s relevantním problémem. Pomocí znalostí, které máme uložené v mozku, identifikujeme problém, jeho možné příčiny a řešení. Zjistíme si další informace a navrhujeme řešení. (3)

2.1.4 DBMS

Je zkratka pro Database Management Systems. V české literatuře se můžeme setkat s názvem systém řízení báze dat (SŘBD). Databázový systém je software, který se stará o veškeré základní služby potřebné pro organizaci a chod databáze (správa současného přístupu více uživatelů, mechanismy pro zálohování, bezpečnostní mechanismy, podpora dotazovacího jazyku, správa transakcí, atd.). DBMS jsou např. MS Access, MS SQL Server, Oracle, MySQL. (7)

2.1.5 Databáze

Potřeba ukládání velkého množství dat vedla ke vzniku databází. Databáze slouží jako úložiště dat. Tato data mohou být užívána i více uživateli současně. Příkladem jednoduché databáze může být kartotéka pacientů u lékaře. Každá složka obsahuje informace o konkrétním pacientovi a lékař si může kdykoliv zjistit co potřebuje. Dnes

preferujeme kvůli jejich výhodám (rychlost, snazší zálohování dat, atd.) elektronické verze databází. (1)

„Teorie databázových systémů v takové podobě, v jaké ji známe dnes, byla definována až v sedmdesátých letech minulého století. Údaje jsou v databázích uloženy nezávisle na aplikačním programu. Databáze tak obsahují nejen vlastní údaje, ale i relační vazby mezi jednotlivými prvky a objekty databáze. Obsahují také popisy použitých struktur a mechanismy pro zaručení integrity dat. Komunikace mezi klientskými aplikacemi a databázovým serverem se uskutečňuje pomocí jazyka SQL.“ (4, s. 9)

2.1.6 Entita

Je množina reálných objektů se stejnými vlastnostmi. Každou entitu můžeme identifikovat podle jména a jedinečných vlastností. (1)

2.1.7 Atribut

Atribut je určitá charakteristika nebo vlastnost entity. Pomáhá nám identifikovat entitu. (7)

2.1.8 Vazby mezi entitami

Entity reprezentují objekty z reálného světa (člověk, auto) a často mají mezi sebou nějaký vztah. Pro korektní návrh databáze je potřeba určit, o jaký vztah se jedná. (7)

Vztah 1:1 - Jeden člověk vlastní jeden občanský průkaz.

Vztah 1:N - Jeden člověk může mít více počítačů (za předpokladu, že ostatní počítače nikdo další nepoužívá).

Vztah N:1 - Je stejný jako 1:N, jen na něj nahlížíme z opačného pohledu. Například více lidí vlastní jeden počítač.

Vztah N:M - V jednom filmu hraje více herců, ale jeden herec hraje také ve více filmech.

2.1.9 Relace

Je to množina spojení mezi entitami. Každé spojení můžeme identifikovat v rámci množiny a mělo by být jedinečné. (1)

2.1.10 Normalizace

Normalizaci můžeme chápat jako pravidla pro úpravu návrhů datových struktur. Jestliže datový model nesplňuje normalizační formy, není správně navržen. (3)

a) 1. normální forma

Každý atribut entity musí být jednoduchý. Nesmí být složený ani vícehodnotový. Například adresa je složený atribut, telefon pak vícenásobný. Složené atributy je nutné rozdělit. U vícenásobných atributů provedeme dekompozici. (3)

b) 2. normální forma

Relace musí být v 1. normální formě a všechny její atributy závisí na kandidátním, resp. primárním klíči. (3)

c) 3. normální forma

„Relace je ve třetí normální formě, pokud je ve druhé normální formě a navíc všechny její neklíčové atributy jsou vzájemně nezávislé.“ (3, str. 60)

d) Boyce - Coddova normální forma

Jestliže je relace v Boyce - Coddově normální formě, pak je také v 3. normální formě. Aby byla relace v Boyce - Coddově normální formě, nesmí mezi kandidátními klíči být žádná funkční závislost. (3)

2.1.11 Datové modely

Určují datovou strukturu a vazby mezi objekty. Ne všechny datové modely jsou aplikovatelné v současných databázových systémech. (3)

2.1.12 Lineární datový model

U lineárního datového modelu nenajdeme žádnou vazbu mezi jednotlivými objekty. Příkladem může být kartotéka občanů na policii. Složky mezi sebou nemají žádný vzájemný vztah. (3)

2.1.13 Hierarchický datový model

Hierarchický model, jak už název napovídá, je tvořen nadřazenými a podřazenými záznamy (rodič-potomek). Záznamy jsou propojeny ukazateli (resp. pointery). Výhodou je vysoká rychlost vyhledávání a přehlednost. Nevýhodou je časová náročnost při vkládání či rušení nových záznamů a složitost rekonstrukce dat. (3)

2.1.14 Síťový datový model

Síťový model je podobný hierarchickému modelu. Nicméně u síťového modelu vedou pointery i do vedlejších záznamů v různých směrech, ne pouze nadřazený-podřazený záznam. (3)

2.1.15 Relační datový model

V současné době patří k nejpoužívanějším datovým modelům. Vzniká spojením více lineárních modelů pomocí tzv. relačního klíče. Toto spojení vzniká při potřebě dat z propojených záznamů a zaniká při ukončení práce s modelem. (3)

RDBMS je relační systém řízení báze dat a jedná se o druhou generaci DBMS. Je založen na relačním datovém modelu, který se dnes řadí k nejpoužívanějším DBMS modelům. (1)

2.1.16 Objektový datový model

Nejnovější datový model. Základním prvkem je objekt, který má své atributy a definované metody určující chování objektu. Mezi objekty můžeme vést přímé vazby díky OID, což je unikátní identifikátor objektu. Relační vazby mohou v objektovém datovém modelu také existovat. (3)

2.1.17 Relační databáze

Relační databáze je založena na relačním datovém modelu. Fyzickou interpretací jsou normalizované tabulky. (1)

2.1.18 Klíče relace

Primární klíč - je složen z jednoho nebo více atributů. Je to jedinečný identifikátor pro každý záznam. Primární klíč musí být jednoznačný (v rámci jedné tabulky neexistují dva stejné primární klíče). Musí být minimální - nemůžeme vypustit žádný atribut primárního klíče bez toho, aby nebyla porušena jednoznačnost primárního klíče. (3)

Kandidátní klíč - je v podstatě primární klíč, avšak není vybrán jako primární klíč. Každá relace může mít více kandidátních klíčů, z nichž se určí jeden primární klíč. Kandidátní klíč, který není vybrán jako primární, se nazývá alternativní klíč. (3)

Cizí klíč - spojení mezi relacemi v relační databázi je možné díky cizímu klíči v jedné tabulce a primárnímu klíči v druhé. Všechny hodnoty cizího klíče jsou zadány nebo nezadány (nic mezitím). Ke každé zadané hodnotě cizího klíče jedné relace existuje stejná hodnota primárního klíče jiné relace. (3)

2.1.19 Integrita relačního modelu

Integrita relačního modelu zajišťuje, aby data uložená v relačním modelu reprezentovala vlastnosti objektů skutečného světa. Rozlišujeme tato integritní omezení: integritní omezení pro entity a integritní omezení pro vztahy entit. (3)

Integritní omezení pro entity jsou následující:

Doménová integrita - omezení pro hodnoty atributů relace. Například datový typ, povinné zadání položky, rozsah hodnot, výchozí hodnota, seznam přípustných hodnot atp. (3)

Entitní integrita - každá tabulka musí mít svůj primární klíč (3)

Referenční integrita - „Pokud existuje v tabulce cizí klíč, musí buď hodnota cizího klíče odpovídat hodnotě některého záznamu v domovské tabulce, nebo musí mít cizí klíč prázdnou hodnotu.“ (1, str. 71)

2.1.20 ER - model

U návrhu databáze je problematické, že programátoři, návrháři a koncoví uživatelé vidí data každý jiným způsobem. Společný pohled na provoz organizace je však velmi důležitý, protože by pak návrh nemohl odpovídat požadavkům zadavatelů. Proto používáme model, který není technický a neobsahuje víceznačnosti (entitně-relační model). ER model začíná identifikací důležitých entit a relací mezi entitami, které je třeba v modelu reprezentovat. (1)

2.2 Proces návrhu databáze

„Návrh databáze je proces vytvoření návrhu, který bude podporovat celkové poslání a dílčí cíle požadovaného databázového systému organizace.“ (1, s. 121)

2.2.1 Návrh databáze

„Během konceptuálního návrhu databáze se pokoušíme identifikovat důležité objekty, které je nutné reprezentovat v databázi, a relace mezi těmito objekty.“ (1, s. 115)

To znamená, že vytvoříme ER model, který reprezentuje úplné a přesné datové požadavky organizace. Kontrolou zajistíme, že model má co nejmenší redundanci dat a je schopen podporovat požadované transakce. (1)

„Konceptuální návrh databáze

Krok 1 Vytvoření ER modelu

- *Identifikace entit*
- *Identifikace relací*
- *Identifikace a spojení atributů s entitami nebo relacemi*
- *Určení domén atributů*
- *Určení atributů, které budou kandidátními, primárními a alternativními klíči*
- *Specializace/generalizace entit (volitelný krok)*

- *Kontrola redundance v modelu*
- *Kontrola, zda model podporuje uživatelské transakce*
- *Posouzení konceptuálního návrhu databáze s uživateli" (1, str. 208)*

2.2.2 Logický návrh databáze

Důležité objekty a relace mezi nimi, které jsme identifikovali v konceptuálním návrhu, jsou v logickém návrhu reprezentovány pomocí množiny tabulek. (1)

Jinými slovy převedeme ER model do množiny relačních tabulek, kde strukturu každé tabulky zkontrolujeme normalizací. Definuujeme požadovaná integritní omezení. (1)

„Logický návrh databáze

Krok 2 Mapování ER modelu do tabulek

- *Vytvoření tabulek*
- *Kontrola tabulek pomocí normalizace*
- *Kontrola, zda tabulky podporují uživatelské transakce*
- *Kontrola integritních omezení*
- *Posouzení logického návrhu databáze s uživateli " (1, str. 208)*

2.2.3 Fyzický návrh databáze

Ve fázi fyzického návrhu databáze rozhodujeme, jakým způsobem implementovat logický návrh do cílového DBMS (1)

Jedním z hlavních cílů fyzického návrhu je efektivní uložení dat, přičemž efektivitu můžeme měřit mnoha faktory (propustnost, doba odezvy, potřebný diskový prostor). Pro zlepšení výkonnosti je nutné porozumět interakci některých hardwarových složek (operační paměť, procesor, diskové I/O operace) a způsobu, jak ovlivňují výkonnost. (1)

„Fyzický návrh databáze

Krok 3 Převod logického návrhu databáze do cílového DBMS

- *Návrh podkladových tabulek*
- *Návrh reprezentace odvozených dat*

- *Návrh zbývajících integritních omezení*

Krok 4 Volba organizace souboru a indexu

- *Analýza transakcí*
- *Volba organizace souboru*
- *Volba indexu*

Krok 5 Návrh uživatelských pohledů

Krok 6 Návrh bezpečnostních mechanismů

Krok 7 Zvážení zavedení kontrolované redundance

Krok 8 Monitorování a doladění systému v provozu“ (1, str. 208)

2.3 Jazyk SQL

SQL (Structured Query Language) se používá při komunikaci s relačními databázemi. Český ekvivalent pro anglické slovo query je dotaz a je to požadavek, který se odesílá databázi. Na základě tohoto požadavku poskytne databáze uživateli odpověď. (8)

SQL je jazyk speciálně navržený pro databáze. SQL umožňuje lidem vytvářet databáze, přidávat do nich data, spravovat tato data a získat vybranou část dat. SQL se v průběhu let stal průmyslovým standardem pod záštitou mezinárodní organizace standardů ISO (International Standards Organisation). (9)

2.3.1 Historie jazyka SQL

„Databázový jazyk SQL (Structured Query Language) vznikl na základě projektu firmy IBM s názvem SEQUEL (Structured English Query Language), jehož cílem bylo vytvořit jazyk blízký angličtině pro práci s údaji v databázi. Postupně se k tomuto standardu přidávaly další firmy (Oracle, SyBase, Informix), a tak vznikl „nepsaný standard“ databázového jazyka s názvem SQL. Postupně byly přijaté vylepšené a upravené standardy jazyka SQL s názvem SQL-86 a později SQL-92. Pro verzi SQL-92 se vžil zkrácený název SQL-2. Ve vývoji jsou verze s pracovním názvem SQL-3 a dokonce už byly zahájeny přípravné práce na normě SQL-4.“ (6, str. 63)

Data Definition Language (DDL)

Pomocí příkazů DDL můžeme vytvářet, měnit, rušit a upravovat objekty (tabulky, indexy, procedury atp.) v databázi. Příkaz CREATE TABLE slouží k vytvoření tabulky. DROP TABLE tabulku zruší a ALTER TABLE tabulku upraví (přidání/odebrání sloupce atp.) (6)

Data Manipulation Language (DML)

Sem patří příkazy pro manipulaci s daty. SELECT pro výběr, INSERT pro vložení, UPDATE pro aktualizaci, DELETE pro smazání. (6)

Data Control Language (DCL)

DCL jsou příkazy pro správu databáze. GRANT - udělení práv, REVOKE - odebrání práv, ALTER USER - úprava uživatele, DROP USER - vymazání uživatele. (6)

Transaction Control Commands

Příkazy pro řízení transakcí jsou CREATE TRANSACTION - vytvoření transakce, COMMIT - ukončí transakci a zapíše změny do databáze. (6)

2.4 Microsoft SQL Server

2.4.1 Historie SQL Serveru

Počátek SQL Serveru se datuje na rok 1988. Tehdy byl SQL Server dodáván firmou Sybase pro operační systém OS/2. V roce 1993 firma Sybase uvedla SQL Server ve verzi 4.2 už pro operační systém Windows. O rok později byl SQL Server koupen společností Microsoft. Microsoft vydal SQL Server 6.05, primárně určený pro segment small business, v roce 1995. Pro platformu Windows NT byla určena verze 6.5 z roku 1996. SQL Server 7.0 byl označován přívlastkem „webová databáze“ a začínal být konkurencí pro databáze Oracle a IBM DB2. Podporu BI (Business Intelligence) přinesla verze SQL Server 2000. SQL Server 2005 přinesl změny v podobě podpory XML jako nativního formátu a filozofie UDM (univerzální dimenzionální modelování) pro BI projekty. Verze 2008 přišla s určitými novinkami, nicméně v porovnání se změnami ve verzi 2005 nebyly tak převratné. (5)

Dnes již existuje Microsoft SQL Server 2012. K vytvoření navrhnuté databáze pro elektronický obchod firmy jsem zvolil právě tuto, v současné době aktuální verzi.

2.4.2 Datové typy SQL Serveru

Datové typy definují, jaká data mohou být vložena pro konkrétní atribut. Máme například atribut věk s číselným datovým typem decimal, což znamená, že se nám nepodaří zadat věk slovy.

Datové typy dělíme na (12):

- Znakové datové typy - char, varchar, text, nchar, nvarchar, ntext
- Binární datové typy - bit, binary, varbinary, image
- Datové typy pro uložení čísel - integer (int), smallint, tinyint, bigint, decimal, smallmoney, money, numeric, float, real
- Datové typy pro uložení data a času - datetime, smalldatetime, date, timestamp, datetimeoffset
- Další datové typy - xml

3 Analýza současného stavu

V této kapitole provedeme analýzu současného stavu ve firmě. Seznámíme se s historií a organizační strukturou firmy. Dále se podíváme na používaný hardware a software. V neposlední řadě se zaměříme na zákazníky, klientelu a provedeme SWOT analýzu.

3.1 Základní informace o firmě

Název: ADONIA v.o.s.

Právní forma: Veřejná obchodní společnost

Sídlo: Brno, Palackého 90, okres Brno-město, PSČ 612 00

IČ: 253 07 746

Datum zápisu do obchodního rejstříku: 14. srpna 1996

Kontakt: www.lekarna-adonis.cz



Kompletní lékárenská péče
včetně poradenství o lécích

Obr. 2: Logo firmy (10)

3.2 Historie firmy

Lékárna ADONIS byla otevřena jako veřejná lékárna základního typu v září roku 1996. Lékárna sídlí v centru Králova Pole v Brně, na křižovatce ulic Palackého a Husitské. Název pochází z léčivé rostliny *Adonis vernalis* (hlaváček jarní). (10)

Počtem zaměstnanců se firma řadí mezi mikropodniky.

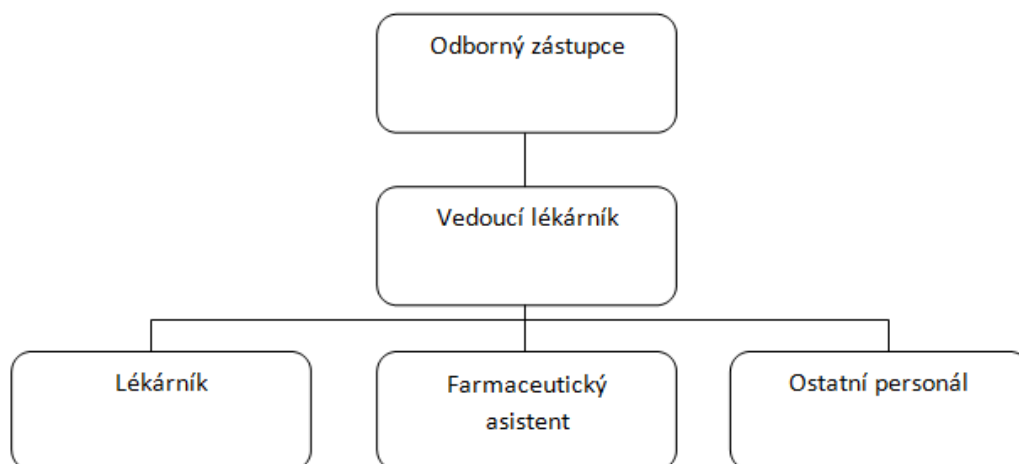
3.3 Organizační struktura

Jako odborný zástupce a vedoucí lékárník působí PharmDr. Roman Smečka, který je odpovědný za řádný chod lékárny a za koordinaci jejích činností. Vedoucí lékárník řídí a kontroluje práci všech pracovníků lékárny (lékárníků, farmaceutických asistentů a ostatních pracovníků lékárny, zejména uklízeček a administrativních pracovníků).

Lékařník vykonává odborné farmaceutické práce při přípravě, uchovávání, kontrole a při výdeji léčiv.

Farmaceutický asistent vydává léčiva, ostatní přípravky a zdravotnické prostředky, které je povoleno vydávat bez lékařského předpisu. Podává informace o jejich správném užívání a uchovávání. Samostatně připravuje léčivé přípravky (individuálně, nebo hromadně) s výjimkou přípravků obsahujících velmi silná a silně účinná léčiva. Objednává léčiva dle pokynů lékárníka. Sleduje a eviduje doby použitelnosti a expirace léčiv. Podle pokynů lékárníka kontroluje kvalitu uskladněných léčiv a vyřazuje nevyhovující léčiva. Farmaceutický asistent vykonává i další práce podle potřeb zaměstnavatele a na pokyn svého nadřízeného, pokud souvisí s jeho vlastní pracovní činností.

O ICT a ekonomické záležitosti se starají externí pracovníci.



Obr. 3: Organizační struktura

Zdroj: vlastní zpracování

3.4 Konkurence

Lékárenská činnost je prostředí s vysokou hustotou konkurenčních subjektů. Dá se říci, že proti sobě stojí soukromníci, státní lékárny a velké řetězce lékáren. Situace je pro všechny ztížena státní regulací marží pro lékárny. To znamená, že lékárny mají velmi limitované možnosti zasahovat do ceny zboží. V blízkém okolí lékárny Adonis se nachází několik dalších lékáren.

Snahou firmy je zvýšit konkurenceschopnost zavedením e-shopu. Primárně jde firmě o získání nových zákazníků, a tím pádem o zvýšení zisku.

3.5 Zákazníci

Pro každou firmu je důležité mít rozsáhlou klientelu, která se dlouhodobě vrací. Lékárna není výjimkou a snaží se proto oslovit co největší počet zákazníků. Nedá se říci, že by lékárna měla určitou cílovou skupinu. Nejvíce odběratelů je z řad fyzických osob, nicméně najde se i firemní klientela.

U elektronického obchodu bude primárně cíleno na české, resp. slovenské zákazníky, nicméně do budoucna je tu možnost prodeje v rámci Evropy.

3.6 Hardware

Co se týče technického vybavení, tak firma disponuje několika osobními počítači včetně periferních zařízení, které slouží k práci se skladovým softwarem, objednávkám zboží a komunikaci. Všechny počítače jsou v pracovní době neustále zapnuté. Počítače jsou sestavovány a spravovány firmou RR - SERVIS, s.r.o. Po určitém časovém intervalu jsou obměňovány, aby byla zachována efektivita práce.

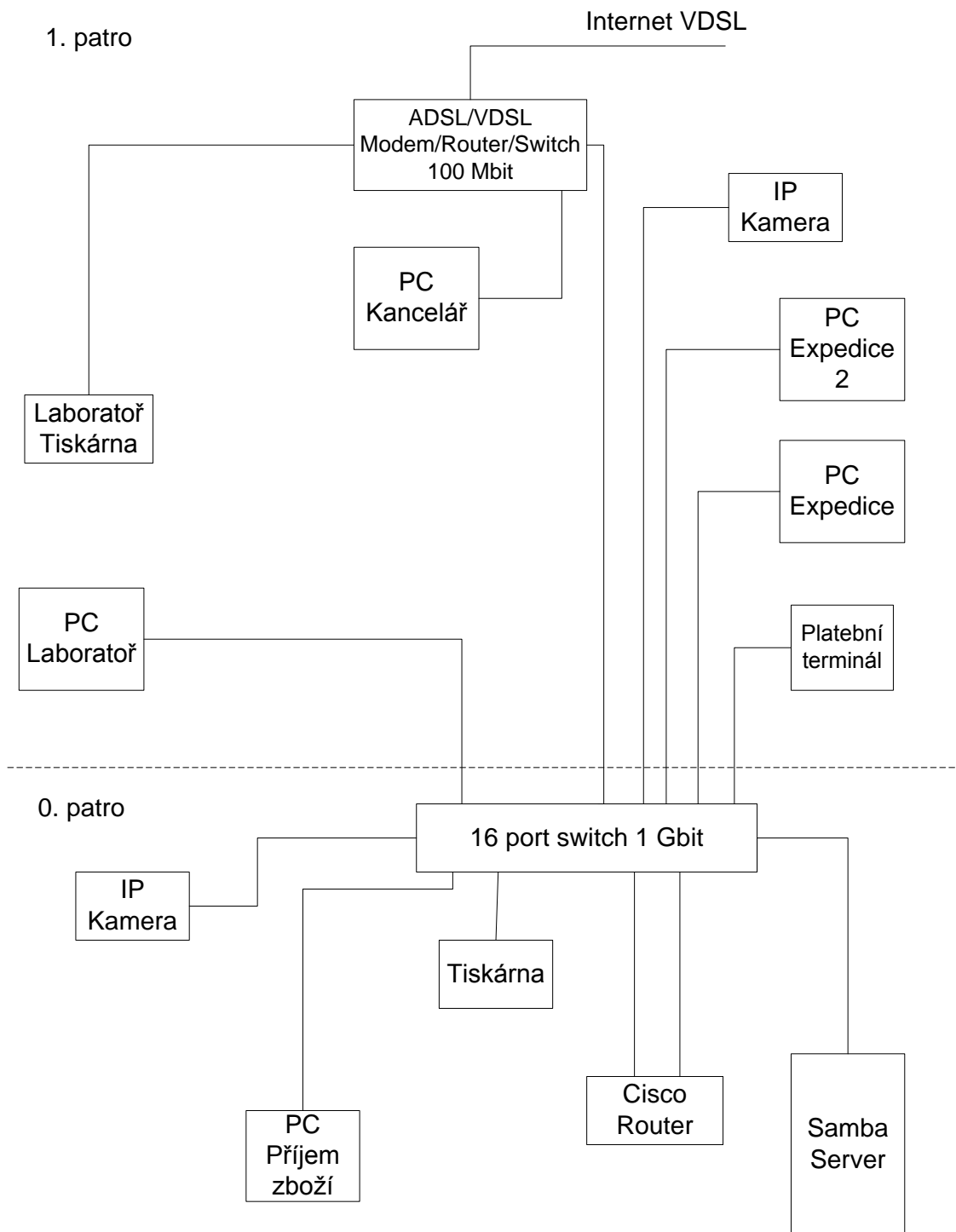
Kromě standardních periferních zařízení je firma vybavena laserovými tiskárnami, skenerem, čtečkou čárových kódů a pokladnami.

V prvním patře v kanceláři je umístěn 100 Mbit Modem/Router/Switch, do kterého je zapojena přípojka VDSL od poskytovatele internetu. Tento Modem/Router/Switch je propojen s tiskárnou v laboratoři, počítačem v kanceláři a vede do 0. patra do 16 portového switchu.

K gigabitovému switchi jsou připojena tato zařízení:

- Samba server, na kterém je uložena MySQL databáze programu Tara
- Cisco router sloužící pro komunikaci s úložištěm v SÚKL (Státní ústav pro kontrolu léčiv). Komunikace se SÚKL probíhá tak, že data jsou poslána do routeru, zašifrována, opatřena hlavičkou a pak odeslána do SÚKL
- Tiskárna
- 4 PC (jeden pro příjem zboží, dva pro expedici, jeden pro laboratoř)
- 2 IP kamery
- Platební terminál na kreditní karty

Při stavbě těchto prostor nebyla vhodně promyšlena budoucí infrastruktura sítě, proto muselo dojít při realizaci sítě k určitým kompromisům.



Obr. 4: Síťové schéma
Zdroj: vlastní zpracování

3.7 Software

Operační systém

Všechny počítače běží pod operačním systémem Windows Xp. Přívětivé uživatelské rozhraní, podpora některých periferií a softwaru, který je nutný pro chod firmy, rozhodlo pro výběr tohoto operačního systému.

Tara

Program Tara je stěžejní software lékárny a slouží nejen k práci se skladem. Původní verze fungovala ještě pod MS-DOS, podpora této verze byla ukončena. Aktuální verze je pro Microsoft Windows.

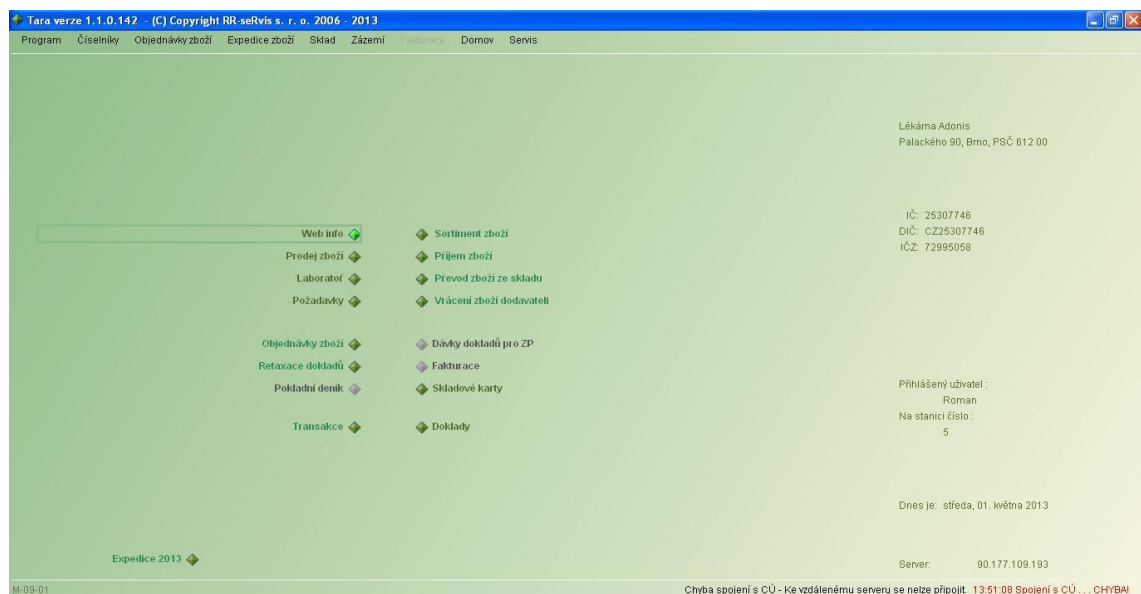
„Program TARA je postaven na nejnovějších technologiích

- platformě .NET

- SQL databázi

Program je postaven tak, aby splnil všechny požadavky kladené na dnešní lékárenský provoz. Tzn. že je umožněna veškerá podpora třetích stran --- AISLP, INTRAKCE atd.”

(11)



Obr. 5: Tara - úvodní obrazovka

Zdroj: vlastní zpracování

Při příjmu nového zboží je nejprve zboží vyhledáno, jestli má vytvořenou kartu. Pak je buď vytvořena nová karta nebo aktualizován stav skladu na novou hodnotu. V obou případech je zboží zaznamenáno v příjemce zboží. Kdykoliv je možné zkontrolovat, kolik je požadovaného zboží skladem, případně provést požadavek na objednávku.

The screenshot shows the 'Tara - Karta zboží' window. It features a header with three main sections: 'SUKL / EAN / PDK', 'Název / Doplněk / ATC skupina', and 'MJ / %DPH'. Below the header, there are two buttons: 'Sklad vedeno' and 'Regulace bez regulace'. The main area contains several input fields and dropdown menus: 'Cena fixní' (0,00), 'Marže' (-1), 'Minimum skladem' (0), 'Prodej' (V - Vyhrazený LP), 'Výrobce', 'Kategorie v E-shop', 'Skupina / podskupina zásob' (BS, ZBOZI), and 'kolik objednat'. There is also a checkbox for 'Publikovat na E-shop' and a button for finding an image. The window ends with 'OK' and 'Storno' buttons.

Obr. 6: Tara - karta zboží

Zdroj: vlastní zpracování

Obr. 7: Tara - příjem zboží

Zdroj: vlastní zpracování

Dále se zaznamenává expedované zboží i inventarizace. Přístup do programu je chráněn přihlašovacími údaji, které jsou vytvořeny každému zaměstnanci. Na základě ověření přihlašovacích údajů jsou přidělena práva. Je možné se přihlásit a pracovat i mimo pracoviště (například z domu).

Data týkající se převážně zboží (název zboží, popis zboží, aktuální počet kusů skladem, atd.) z Tary budou exportována do databáze e-shopu. Je proto důležité mít tento fakt v povědomí při navrhování databáze pro elektronický obchod.

Skupina	SKLKL	Název zboží	doplňk názvu	Mj	Dojino	Skladem	Cena SD	DPH v...	DPH v...	Cena...	Cena...	Sklad	Karta ID	Zboží ID	Expirace	Marže
BS		ALTERMED Superbronz SPF10 Sun spray 200ml		KS	1,00	1,00	259,00	21,0	20,0	184,87	0,00	SKLADI	138889	97188	17.02.2014	15,7
BS		DOZA KAPENÍ NA LEBNÝ		BAL	2,00	1,00	29,00	21,0	20,0	10,33	0,00	SKLADI	139322	99495	16.05.2013	31,8

Obr. 8: Tara - zboží na skladových kartách

Zdroj: vlastní zpracování

Výhody

- Úzká spolupráce s vývojáři. To znamená, že lékárny, které tento software používají, mají možnost se podílet na jeho vývoji a podobě
- Pokud se firma rozhodne pro Taru, pak k tomu dostane i kompletní servis (instalace, nastavení zálohování, řešení problémů, instrukce k používání atp.)

Nevýhody

- Lékárny používající Taru jsou závislé na podpoře ze stran vývojářů. Pokud by byl vývoj ukončen, pak by byl přechod na jiný software z časového a finančního hlediska problematický

OpenOffice

V podstatě žádná firma se dnes neobejde bez kancelářského softwaru. Důvodem, proč firma používá tento kancelářský balík, je, že zvládá všechny funkce co jiné kancelářské balíky a je zdarma použitelný i v komerční sféře.

AISLP

Pokud některý z pracovníků lékárny potřebuje zjistit určité informace o léčivých přípravcích, pak je naleznou v databázi AISLP. Databáze je určena pro odbornou veřejnost. Data jsou získávána od státních institucí a částečně z veřejně dostupných portálů.

3.8 SWOT analýza

SWOT představuje analýzu silných, slabých stránek, příležitostí a hrozeb firmy. Tato analýza nám pomáhá určit směr, jakým by se firma měla ubírat do budoucna. To znamená využít příležitosti, těžit ze silných stránek, eliminovat (potlačit) hrozby a slabé stránky.

SWOT analýza		
Vnitřní prostředí	Silné stránky (Strengths)	Slabé stránky (Weaknesses)
	<ul style="list-style-type: none">• Dobrá orientace na trhu• Schopnost přizpůsobit se• Dobrá poloha firmy• Výborné vztahy s partnery a zákazníky• Loajalita zaměstnanců	<ul style="list-style-type: none">• Absence E-shopu• Málo zaměstnanců• Slabá propagace firmy
Vnější prostředí	Příležitosti (Opportunities)	Hrozby (Threats)
	<ul style="list-style-type: none">• Intenzivnější zapojení zaměstnanců na zlepšení efektivity firmy• Expanze do dalších měst• Zajištění dlouhodobé přízně zákazníků• Založení E-shopu	<ul style="list-style-type: none">• Legislativní změny• Vstup nové konkurence na trh

Tabulka 1: SWOT analýza

Zdroj: vlastní zpracování

4 Návrh vlastního řešení, přínos návrhu řešení

V této části bakalářské práce si nejdříve ujasníme, jaké jsou kladeny požadavky na navrhovanou databázi. Poté se budeme věnovat samotnému návrhu, přičemž budeme postupovat od konceptuálního návrhu přes logický návrh po fyzický návrh. Nakonec zjistíme, jaký byl přínos návrhu databáze pro elektronický obchod firmy.

4.1 Požadavky na databázi

Pro získání více pohledů na problematiku byly požadavky na databázi konzultovány s odborným zástupcem firmy i dalšími zaměstnanci.

Hlavním požadavkem je navrhnutí databáze jako základu pro elektronický obchod. E-shop bude s daty uloženými v databázi pracovat (vytvářet, číst, ukládat, editovat, mazat).

4.1.1 Dílčí požadavky

Zálohování

Všechna data by měla jít snadno zálohovat. Nesmíme přijít o žádná data při poruše hardwaru, respektive chybě softwaru. Záloha bude prováděna automaticky, ideálně v ranních hodinách.

Zabezpečení

Databáze bude chráněna heslem. Běžní uživatelé nebudou mít z bezpečnostních důvodů přímý přístup do databáze e-shopu. Přístup bude umožněn přidělením práv pouze zaměstnancům.

Výkon

Odezva požadavků (vyhledávání, aktualizace) na databázi musí být v rozsahu nejvýše pár vteřin. Čím bude nižší odezva, tím lépe.

4.1.2 Požadavky na základní entity

Pobočka

Pobočky v databázi nebudou uváděny. V případě vybudování dalších poboček bude e-shop brán jako jeden centrální obchod.

Zboží

Při větším množství výrobků není možné ručně provádět aktualizace údajů (např. informace o počtu kusů skladem). Je proto důležité, aby bylo možné exportovat data z programu Tara a importovat je do databáze e-shopu.

Zákazník

Zákazník bude mít přístup ke změně svých kontaktních údajů, prohlížení obchodu, přidání zboží do košíku, smazání zboží z košíku, změně počtu položek v košíku a provedení objednávky.

Uživatel

Jako uživatel je chápán někdo, kdo bude mít přístup do administrátorské sekce e-shopu (pracovník lékárny). Uživateli jsou při registraci udělena práva. Ta určují, jaké úkony může provádět:

- Základní uživatel (například brigádník) má možnost pouze číst, nemůže nic editovat
- Zaměstnanec je oprávněn spravovat objednávky, editovat/mazat kategorie a zboží
- Administrátor má práva ke všemu. Provádí nové registrace a uděluje práva ostatním uživatelům

U každé objednávky bude uvedeno, kdo ji vyřizuje. Jednu objednávku bude vyřizovat pouze jeden zaměstnanec.

4.2 Identifikace základních procesů v databázi

Hlavní procesy, které byly zjištěny, jsou evidence nového zboží, registrace nového zákazníka, vytvoření objednávky.

4.2.1 Evidence nového zboží

Zboží uložené v databázi programu Tara exportujeme do externího dokumentu. Dokument importujeme do databáze. Při importu je zjištěno, zda se zboží v databázi již nachází. V případě výskytu se informace aktualizují na nové hodnoty. V opačném případě se přidá nový záznam a uloží se do databáze.

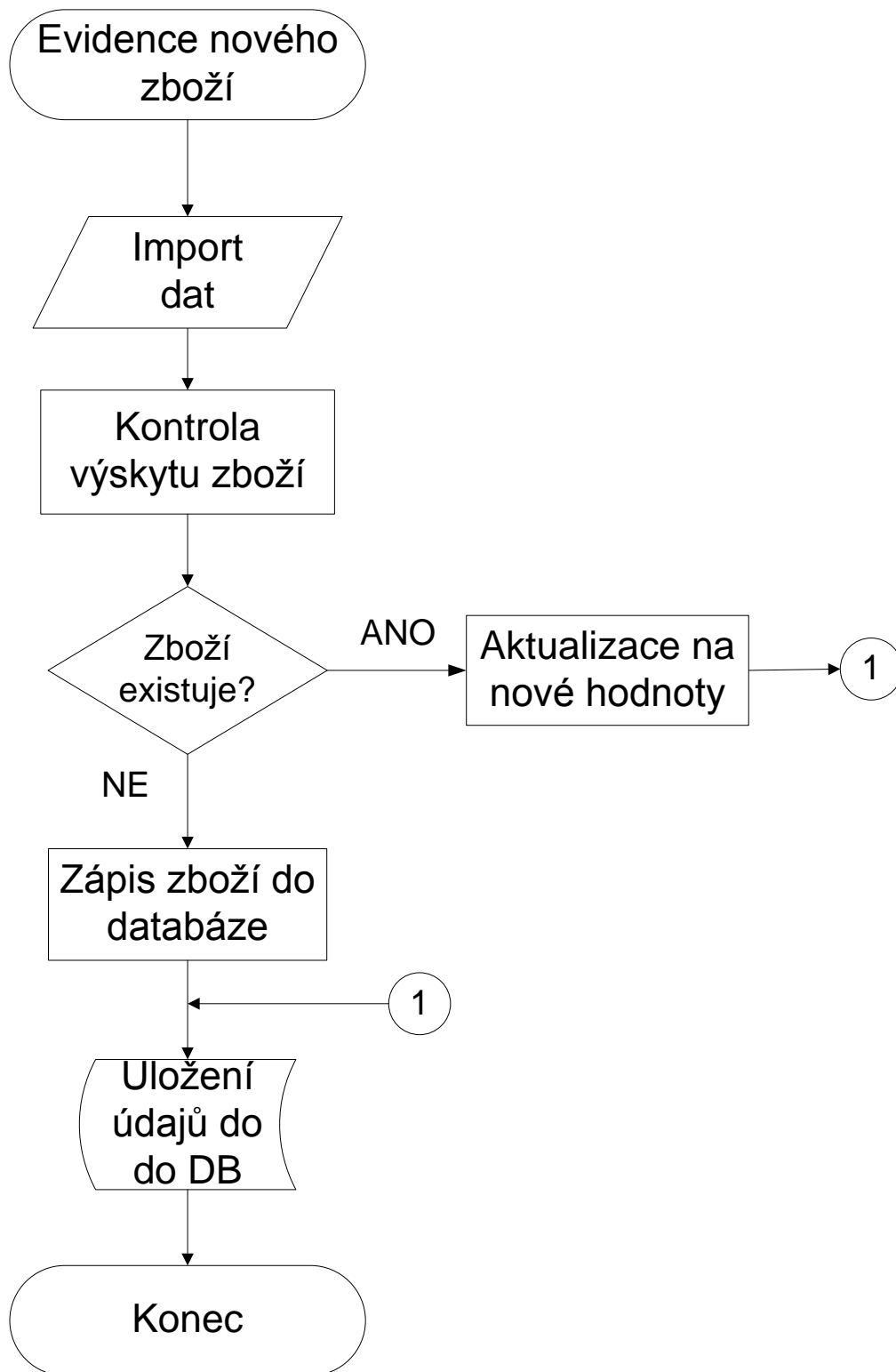
4.2.2 Registrace nového zákazníka

Máme nového zákazníka, který se chce registrovat do databáze. Poté co vyplní všechny údaje o sobě, je ověřeno správné zadání údajů a výskyt tohoto zákazníka v databázi. Při nesprávném zadání údajů (například číslo ve jménu) je vyzván k opravě. Jestliže se již v databázi nachází, pak je požádán, jestli chce zadat jiné údaje. Pokud se v databázi nenachází jsou jeho údaje uloženy.

Registrace nového uživatele je obdobná registraci nového zákazníka. S tím rozdílem, že tuto registraci provádí administrátor.

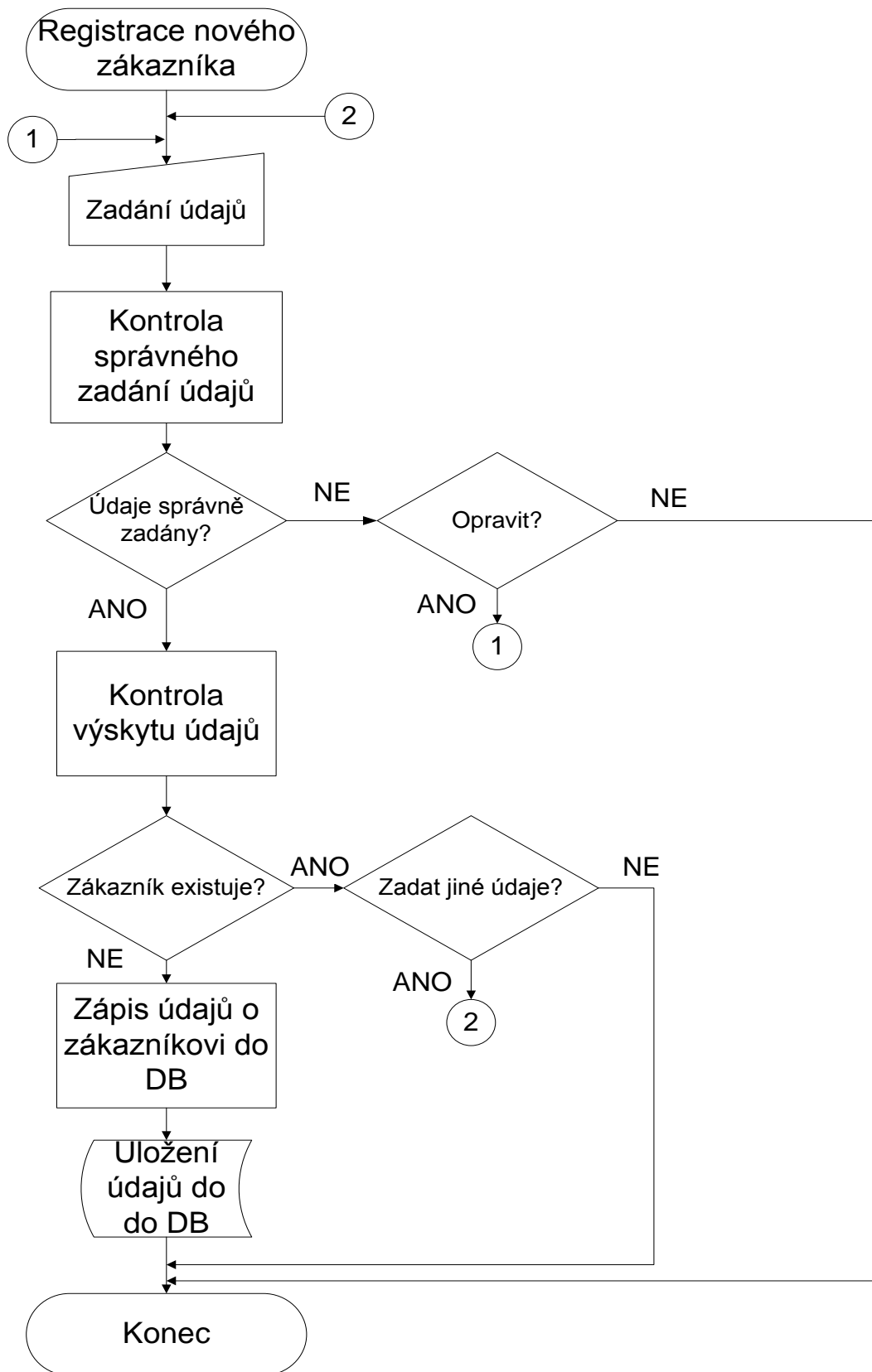
4.2.3 Vytvoření objednávky

Tato činnost popisuje provedení objednávky určitého zboží zákazníkem. Zákazník si vybere zboží a přejde k objednávce. Zákazník je dotázán, zda je registrovaný. Jestliže není registrován, pak je mu nabídnuta registrace. Registrovanému zákazníkovi je nabídnuto přihlášení. Po zadání přihlašovacích údajů je ověřena jejich platnost. Pro neplatné přihlašovací údaje je vznesen požadavek na jejich opětovné zadání. Následuje výběr upřesňujících informací ohledně typu dopravy, platby atp. Poté zbývá poslední dotaz na správnost všech údajů. Pokud jsou údaje v nepořádku, pak je vznesen požadavek na opravení. U korektních údajů následuje uložení objednávky do databáze.



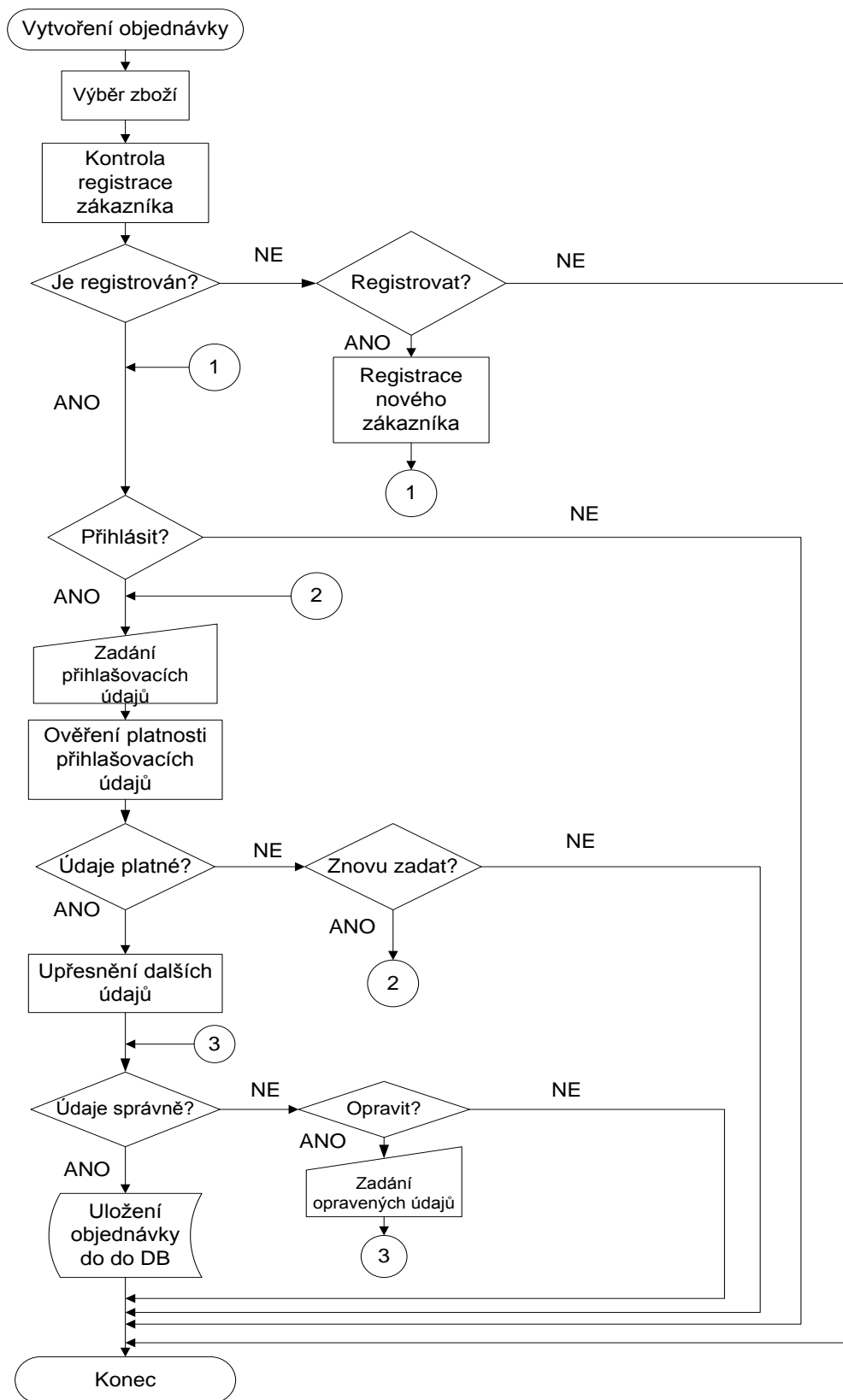
Obr. 9: Evidence nového zboží

Zdroj: vlastní zpracování



Obr. 10: Registrace nového zákazníka

Zdroj: vlastní zpracování



Obr. 11: Vytvoření objednávky

Zdroj: vlastní zpracování

4.3 Zvolené DBMS

Při rozhodování, jaký DBMS použijeme, si můžeme vybrat z nabídky komerčních produktů nebo produktů s open source licencí. Dá se říci, že žádná volba není špatná, pokud DBMS splňuje naše požadavky.

Já jsem vybíral mezi MySQL a MS SQL Server. I když databáze programu Tara běží na MySQL, nakonec jsem zvolil MS SQL Server 2012. Dříve jsem s ním pracoval, má příjemné uživatelské rozhraní, je stabilní, má vysoký výkon a dostatečnou úroveň zabezpečení. Také rozhodlo to, že Microsoft je dlouhodobou zaběhnutou firmou, což zajišťuje SQL Serveru dlouhodobou podporu.

MySQL a MS SQL Server nejsou až tak odlišné. To znamená, že s jistými úpravami jsme schopni převést návrh databáze provedený v MS SQL Serveru do MySQL, respektive obráceně. Vkládání dat z MYSQL do SQL Serveru lze provést přímo v SQL Serveru (po nainstalování příslušných ovladačů) nebo pomocí programů třetích stran. Podstatné je, aby obě databáze měly stejnou strukturu. My pak určíme, jaká data jsou importována kam.

4.4 Konceptuální návrh databáze

V konceptuálním návrhu databáze jde především o identifikaci základních entit, definování atributů entit a určení vzájemných vztahů mezi entitami. Pomocí těchto informací vytvoříme základní ER model.

4.4.1 Identifikace základních entit a definování atributů entit

Zboží

Zboží je to, co budeme nabízet zákazníkovi. Z prodaného zboží má obchod tržby, respektive zisk. Zákazníkovi musíme poskytnout všechny důležité údaje o zboží, aby měl dostatečné množství informací při rozhodování o koupi. Atributy zboží jsou:

- Název zboží
- Popis zboží - k čemu zboží slouží, pro koho je určeno atp.
- Měrná jednotka - kus, balení, atd.

- Cena za měrnou jednotku - cena za balení, kus, atd.
- Počet měrné jednotky skladem - počet kusů, balení, atd. skladem
- SÚKL - identifikační kód Státního úřadu pro kontrolu léčiv
- EAN - čárový kód
- Obrázek - obrázek výrobku dává zákazníkovi představu o tom, co kupuje
- DPH - daň z přidané hodnoty
- Výrobce
- Kategorie zboží - do jaké kategorie zboží patří (vitamíny, homeopatika, atd.)
- Dostupnost - zboží skladem, není skladem

Zákazník

Zákazník o sobě při registraci poskytuje e-shopu informace. Tyto informace jsou důležité pro komunikaci se zákazníkem, pro vyřízení objednávky a zároveň zákazníkovi šetří čas při jeho dalším nákupu. Atributy zákazníka jsou:

- Adresa - město, ulice, číslo popisné, psč, stát
- Jméno, příjmení
- Název firmy
- IČ - identifikační číslo
- DIČ - daňové identifikační číslo
- E-mail - e-mail bude zároveň plnit funkci přihlašovacího jména
- Heslo - heslo pro přihlášení
- Telefon
- Newsletter - Souhlas/nesouhlas se zasíláním informačního letáku
- Datum registrace
- Datum aktualizace - datum provedení změn v účtu

Objednávka

Objednávka je požadavek konkrétního člověka na koupi specifického zboží v nějakém množství za určitých podmínek. Objednávka má následující atributy:

- Zboží - údaje o vybraném zboží (co, kolik, cena, atd.)

- Zákazník - informace o nakupujícím
- Stav objednávky - přijata, vyřízena, atd.
- Způsob doručení - poštou, osobní převzetí, atd.
- Způsob platby - bankovním převodem, hotově, atd.
- Datum objednávky
- Datum změny v objednávce
- Uživatel - informace o tom, kdo objednávku vyřizuje

Faktura

Na základě údajů z objednávky je sestavena faktura. Faktura má také archivační úlohu. Podává přehled o prodeji zboží v čase. Atributy faktury jsou:

- Objednávka - informace o objednávce
- Datum vytvoření - datum vytvoření faktury
- Datum splatnosti - kdy má být faktura zaplacená
- Celková cena poštovného
- Celková cena

Uživatel

Zaměstnancům jsou vytvořeny uživatelské účty pro přístup do administrace elektronického obchodu. Uživatelským účtům jsou přidělena práva pro práci s databází. Entita uživatel má tyto atributy:

- Práva - přidělena práva pro uživatele/zaměstnance/administrátora
- E-mail - e-mail slouží jako přihlašovací jméno
- Heslo - heslo pro přihlášení
- Jméno příjmení

4.4.2 Určení vzájemných vztahů mezi entitami

Zákazník - Objednávka

Zákazník provádí objednávku. Zákazník může provést více objednávek. Konkrétní objednávka může být provedena jedním zákazníkem. Vazba mezi entitami je typu 1:N.

Faktura - Objednávka

Faktura vychází z objednávky. Faktura může mít více objednávek. Objednávka má jednu fakturu. Relace 1:N.

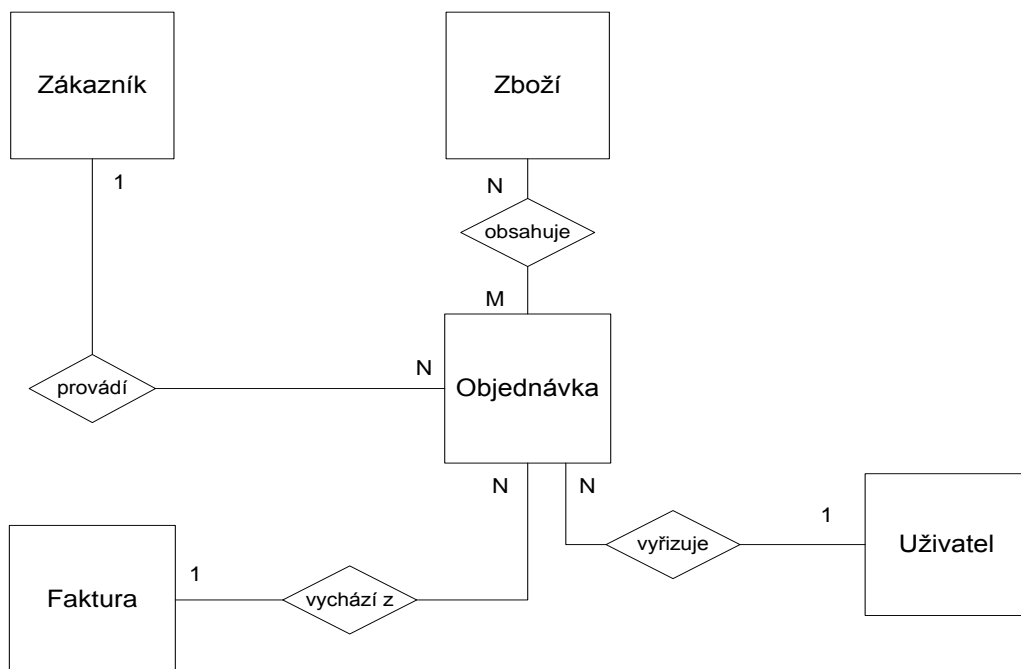
Uživatel - Objednávka

Uživatel vyřizuje jednu nebo více objednávek. Objednávka je vyřizována pouze jedním uživatelem. Vztah mezi entitami je 1:N.

Objednávka - Zboží

Objednávka obsahuje zboží. Jedna objednávka může obsahovat více zboží. Jedno zboží může být obsaženo ve více objednávkách. Vazba M:N.

4.4.3 Základní ER model



Obr. 12: Základní ER model

Zdroj: vlastní zpracování

4.5 Logický návrh databáze

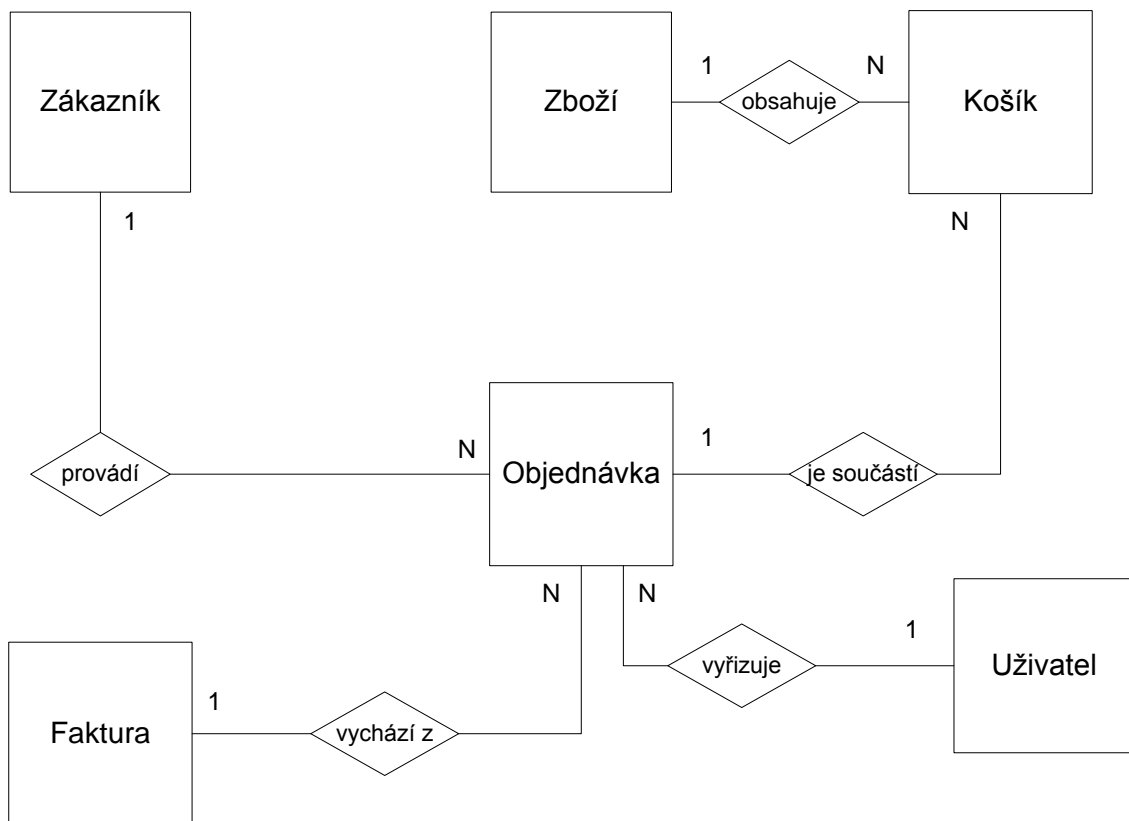
V logickém návrhu použijeme poznatky z konceptuálního modelu k vytvoření tabulek. Musíme zajistit integritní omezení, provést rozklad vazeb M:N a normalizaci tabulek.

4.5.1 Rozklad vazeb M:N

Vazby M:N je nutné rozložit. Rozklad probíhá tak, že se vytvoří další tabulka, která obsahuje primární klíče obou tabulek, jež je nutné dekomponovat.

Vazba mezi entitami Zboží a Objednávka

Vytvoříme novou tabulku s názvem Košík. Košík bude mimo jiné obsahovat primární klíč tabulky Zboží a primární klíč tabulky Objednávka.



Obr. 13: ER model po dekompozici

Zdroj: vlastní zpracování

4.5.2 Vytvoření tabulek

Identifikovali jsme základní entity, určili vazby mezi nimi a provedli dekompozici. Pomocí těchto poznatků převedeme entity na tabulky.

Úprava do 1. normální formy

Rozdělíme složené atributy u všech entit. Adresu rozdělíme na město, ulici, číslo popisné, psč. Jméno, příjmení rozložíme na položky jméno a příjmení.

Zjistíme, které atributy jsou vícenásobné a uděláme jim zvláštní tabulky. Nově vytvořenou tabulku spojíme s původní tabulkou pomocí primárního a cizího klíče. Toto provedeme pro atributy obrázek, kategorie zboží, adresa zákazníka (zákazník může mít fakturační a dodací adresu). Telefon ke kontaktování zákazníka nám stačí jeden. E-mail taktéž není třeba více než jeden, zároveň plní funkci přihlašovacího jména.

Úprava do 2. normální formy

Druhá normální forma se týká jen tabulek se složenými primárními klíči - tedy Košík. Všechny atributy tabulky Košík jsou v 1. normální formě a jsou funkčně závislé na celém primárním klíči. Splňují tak podmínky 2. normální formy.

Úprava do 3. normální formy

Všechny tabulky jsou ve 3. normální formě, protože všechny neklíčové atributy jsou závislé pouze na primárním klíči.

Číselníky

Některé položky tabulky je nutné převést do zvláštních tabulek jako tzv. číselníky. Důvodem je usnadnění práce s databází. Příkladem může být položka DPH u tabulky Zboží. Kdybychom ponechali daň z přidané hodnoty tak, jak je, tak při změně DPH jsme nuceni přepisovat ručně každou hodnotu. Pokud vytvoříme speciální tabulku DPH, která je s tabulkou Zboží spojená primárním klíčem DPH (id_dph), pak stačí změnit sazbu DPH jednoho řádku u nově vytvořené tabulky DPH. To znamená pokud v tabulce DPH změním u id_dph 1 sazbu z 20 na 21, pak v tabulce Zboží se pro všechny id_dph 1 změní hodnota na 21.

Tabulky vypadají následovně:

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
kosik	id_objednavka	int	fk	not null	
	id_zbozi	int	fk	not null	
	cena_za_mernou_jednotku_bez_dph	decimal(8,2)		not null	Aktuální cena za měrnou jednotku
	sazba_dane	decimal(5,3)		not null	Aktuální sazba DPH
	pocet_mj	decimal(5,2)		not null	Počet měrných jednotek zboží

Tabulka 2: Košík

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
zbozi	id_zbozi	int	pk	identity(1, 1)	
	nazev	varchar(80)		not null	Název zboží
	popis	text			Popis zboží
	id_merna_jednotka	int	fk	not null	
	cena_za_mernou_jednotku_bez_dph	decimal(8,2)		not null	Cena za balení, kus bez dph
	pocet_merne_jednotky_skladem	decimal(5,2)		not null	Počet měrných jednotek skladem
	sukl	varchar(7)			Identifikační kód SÚKL
	ean	varchar(13)		not null	Čárový kód
	id_obrazek	int	fk		
	id_dph	int	fk	not null	
	datum_pridani	date			Datum přidání zboží
	id_vyrobce	int	fk		
	id_zbozi_kategorie	int	fk	not null	
	dostupnost	tinyint		not null	Skladem, není skladem 0/1

Tabulka 3: Zboží

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
zbozi_kategorie	id_zbozi_kategorie	int	pk	identity(1, 1)	
	nazev	varchar(64)		not null	Název kategorie zboží
	popis	text			Popis kategorie zboží

Tabulka 4: Kategorie zboží

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
obrazek	id_obrazek	int	pk	identity(1, 1)	
	obrazek_nazev	varchar(50)		not null	Název obrázku
	obrazek	image		not null	Uložený obrázek

Tabulka 5: Obrázek

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
vyrobce	id_vyrobce	int	pk	identity(1, 1)	
	nazev	varchar(64)		not null	Název výrobce

Tabulka 6: Výrobce

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
dph	id_dph	int	pk	identity(1, 1)	
	sazba	decimal(5,3)		not null	Sazba DPH

Tabulka 7: DPH

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
merna_jednotka	id_merna_jednotka	int	pk	identity(1, 1)	
	nazev	varchar(5)		not null	Bal, ks, atd.

Tabulka 8: Měrná jednotka

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
zakaznik	id_zakaznik	int	pk	identity(1, 1)	
	newsletter	tinyint		not null	Chce zasílat newsletter? 0/1
	email	varchar(128)		not null	Email slouží jako login
	heslo	varchar(32)		not null	Heslo pro přihlášení
	zakaznik_telefon	varchar(15)		not null	Telefon na zákazníka
	datum_registrace	date		not null	Datum registrace
	datum_aktualizace	date			Datum aktualizace
	id_zakaznik_adresa	int	fk	not null	

Tabulka 9: Zákazník

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
zakaznik_adresa	id_zakaznik_adresa	int	pk	identity(1, 1)	
	jmeno	varchar(35)		not null	Jméno zákazníka
	prijmeni	varchar(50)		not null	Příjmení zákazníka
	firma	varchar(64)			Firma
	mesto	varchar(64)		not null	Město
	ulice	varchar(80)		not null	Ulice
	cislo_popisne	varchar(5)		not null	Číslo popisné
	psc	varchar(12)		not null	Poštovní směrovací číslo
	ic	varchar(13)			Identifikační číslo
	dic	varchar(14)			Daňové identifikační číslo
	id_stat	int	fk	not null	
	typ_adresy	tinyint		not null	Fakturační nebo dodací adresa 0/1

Tabulka 10: Adresa zákazníka

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
stat	id_stat	int	pk	identity(1, 1)	
	nazev	varchar(64)		not null	Název státu

Tabulka 11: Stát

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
objednavka	id_objednavka	int	pk	identity(1, 1)	
	id_zakaznik	int	fk	not null	
	id_zpusob_doruceni	int	fk	not null	
	id_zpusob_platby	int	fk	not null	
	objednavka_datum	date		not null	Datum objednávky
	objednavka_zmena	date		not null	Datum změny v objednávce
	id_uzivatel	int	fk	not null	
	id_objednavka_stav	int	fk	not null	

Tabulka 12: Objednávka

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
zpusob_platby	id_zpusob_platby	int	pk	identity(1, 1)	
	nazev	varchar(30)		not null	Hotově, bankovní převod, atd.

Tabulka 13: Způsob platby

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
zpusob_doruceni	id_zpusob_doruceni	int	pk	identity(1, 1)	
	nazev	varchar(64)		not null	Osobní odběr, PPL, Česká Pošta, atd.
	cena_doruceni	decimal(6,2)		not null	Cena za doručení

Tabulka 14: Způsob doručení

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
objednavka_stav	id_objednavka_stav	int	pk	identity(1, 1)	
	nazev	varchar(20)		not null	Přijatá, expedovaná, zpracovává se

Tabulka 15: Stav objednávky

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
faktura	id_faktura	int	pk	identity(1, 1)	
	id_objednavka	int	fk	not null	
	datum_vytvoreni	date		not null	Datum vytvoření faktury
	datum_splatnosti	date		not null	Datum splatnosti faktury
	celkova_cena	decimal(10,2)		not null	Cena celkem
	celkova_cena_postovne	decimal(7,2)		not null	Poštovné celkem

Tabulka 16: Faktura

Zdroj: vlastní zpracování

Název tabulky	Atributy tabulky	Datový typ	Klíč	Další omezení	Popis
uzivatel	id_uzivatel	int	pk	identity(1, 1)	
	prava	tinyint		not null	Uživatelská práva. 0 pro uživatele, 1 pro zaměstnance. 2 pro administrátora
	email	varchar(128)		not null	Přihlašovací jméno. Slouží jako login
	heslo	varchar(32)		not null	Heslo pro přihlášení
	jmeno	varchar(35)		not null	Jméno
	prijmeni	varchar(50)		not null	Příjmení

Tabulka 17: Uživatel

Zdroj: vlastní zpracování

4.6 Fyzický návrh databáze

Hlavní náplní fyzického návrhu je naprogramování návrhu databáze na základě znalostí, které jsme doposud získali. Zdrojový kód je uveden v příloze. Vytvoříme také několik příkladů praktického využití databáze.

4.6.1 Vyhledávání zboží

Podstatnou funkcí každého elektronického obchodu je možnost vyhledávání. Představme si, že máme zájem koupit například Aspirin. Nevíme, ve které kategorii zboží najdeme, tak použijeme vyhledávání. Zde je důležité zajistit vyhledávání všech produktů, i když zadáme jen jedno slovo. Název zboží se většinou skládá z více slov. Tato procedura současně umožňuje vyhledávat podle čárového kódu.

```
use eshop
```

```
go
```

```
-- Procedura vyhledávání zboží
```

```
create procedure vyhledat_zbozi
```

```
@nazevzbozi varchar(80),
```

```
@kodean varchar(13)
```

```
as
```

```
select zbozi.nazev as 'Název zboží', zbozi.popis as 'Popis zboží', zbozi.ean as 'Čárový  
kód' from zbozi where nazev like '%'+@nazevzbozi+'%' or @kodean=ean
```

```
go
```

```
-- Spuštění procedury
```

```
exec vyhledat_zbozi 'aspirin',null
```

```
-- Smazání procedury
```

```
drop procedure vyhledat_zbozi
```

4.6.2 Registrace nového zákazníka

Při registraci nového zákazníka plníme daty 2 tabulky. Tabulku Zákazník a Zákazník adresa. Po zadání údajů je ověřeno, zda se údaje v databázi vyskytují, či nikoliv. Jestliže není zjištěn výskyt, pak jsou nové údaje uloženy.

```
use eshop
```

```
go
```

```
-- Procedura registrace zákazníka
```

```
create procedure registrace_zakaznika
```

```
@newsletter tinyint,
```

```
@email varchar(128),
```

```
@heslo varchar(32),
```

```
@telefon varchar(15),
```

```
@registrace date,
```

```
@aktualizace date,
```

```
@jmeno varchar(35),
```

```
@prijmeni varchar(50),
```

```
@firma varchar(64),
```

```
@mesto varchar(64),
```

```
@ulice varchar(80),
```

```
@cislo_popisne varchar(5),
```

```
@psc varchar(12),
```

```
@ic varchar(13),
```

```
@dic varchar(14),
```

```
@id_stat int,
```

```
@typ_adresy tinyint
```

```
as
```

```
begin
```

```
declare
```

```
@id_zakaznik int
```

```
select @id_zakaznik = id_zakaznik
```

```

from zakaznik where @email = email and @telefon = zakaznik_telefon
declare
@id_zakaznik_adresa int
select @id_zakaznik_adresa = IDENT_CURRENT('zakaznik_adresa')
if @id_zakaznik is null
begin
insert into zakaznik_adresa (jmeno, prijmeni, firma, mesto, ulice, cislo_popisne, psc, ic,
dic, id_stat, typ_adresy)
values (@jmeno, @prijmeni, @firma, @mesto, @ulice, @cislo_popisne, @psc, @ic,
@dic, @id_stat, @typ_adresy)
insert into zakaznik (newsletter, email, heslo, zakaznik_telefon, datum_registrace,
datum_aktualizace, id_zakaznik_adresa)
values (@newsletter, @email, @heslo,@telefon, @registrace, @aktualizace,
@id_zakaznik_adresa)
end
else
begin
print 'Zákazník se v databázi již nachází!'
end
end
go

-- Spuštění procedury
exec registrace_zakaznika 0,'zkouska@email.com', 'heslo5', '732123589', '2012-3-10',
'2012-4-20', 'Jan','Dvořák','DataEO', 'Praha', 'Bzenecká',
'85a','65200','987654321','CZ987654321',1,0

-- Smazání procedury
drop procedure registrace_zakaznika

-- Zobrazení výsledků
select * from zakaznik

```

```
select * from zakaznik_adresa
```

4.6.3 Změna sazby DPH

Výchozí hodnota sazby DPH je 21. Dejme tomu, že příští rok se sazba DPH zvedne na 25. Pomocí jednoduché transakce jsme schopni změnit hodnoty DPH u veškerého zboží.

```
use eshop
```

```
go
```

```
-- Transakce změna sazby dph
```

```
begin tran zmena_sazby_dph
```

```
update dph set sazba=25 where sazba=21
```

```
-- Potvrzení transakce
```

```
commit tran zmena_sazby_dph
```

```
-- Odvolání transakce
```

```
rollback tran zmena_sazby_dph
```

```
-- Kontrola změn
```

```
select * from dph
```

4.6.4. Změna ceny zboží

Zboží je sice exportováno z Tary a importováno do databáze elektronického obchodu, nicméně abychom zaujali zákazníky, tak snížíme cenu za měrnou jednotku zboží.

```
use eshop
```

```
go
```

```
-- Transakce změna ceny zboží
```

```
begin tran zmena_ceny_zbozi
```

```

update zboží set cena_za_mernou_jednotku_bez_dph =
cena_za_mernou_jednotku_bez_dph*0.95
-- Potvrzení transakce
commit tran zmena_ceny_zbozi
-- Odvolání transakce
rollback tran zmena_ceny_zbozi

-- Kontrola změn
select * from zboží

```

4.6.5 Upozornění na novou objednávku

Pomocí spouští (triggerů) jsme schopni sledovat změny v konkrétních tabulkách. Triggery spouštějí příkazy insert, update, delete.

```

use eshop
go

-- Spoušť upozorňující na novou objednávku
create trigger nova_objednavka
on objednavka
after insert
as
begin
print 'Vytvořena nová objednávka'
end

insert into objednavka (id_zakaznik, id_zpusob_doruceni, id_zpusob_platby,
objednavka_datum, objednavka_zmena, id_uzivatel, id_objednavka_stav)
values (1,1,1,'2013-05-03','2013-05-07',2,1)

--Smazání spouště
drop trigger nova_objednavka

```

4.7 Postup po provedení návrhu

Po provedení návrhu máme více možností pro vybudování kompletního elektronického obchodu.

Použijeme již vytvořenou šablonu e-shopu

Na internetu je spousta zdarma dostupných nebo placených šablon právě pro tyto účely. Výběr správné šablony včetně následné editace do požadovaného stavu může být časově náročný.

Výhody

- Příjemné grafické zpracování
- Funkčnost ověřena spoustou dalších uživatelů
- Relativně levné

Nevýhody

- Nutné přizpůsobení šablony pro náš návrh
- Potřeba úpravy databáze konkrétnímu návrhu
- Bez dalších úprav designu existuje spousta dalších obchodů se stejným vzhledem

Zbytek obchodu naprogramujeme sami

Informace, které jsme získali při navrhování databáze, nám zde také pomohou.

Výhody

- Možnost absolutního přizpůsobení
- Problémy a chyby jsme schopni sami vyřešit

Nevýhody

- Případné nedostatky v zabezpečení, což může mít za následky v krajních mezích například vznik chyb, únik dat, atd.
- Nutná znalost programování e-shopů, SEO optimalizace

Necháme si obchod naprogramovat

Různorodá cena v závislosti na komplexnosti řešení. Obchod je zpracován na míru, často spolu se SEO optimalizací pro vyhledávače.

Výhody

- Zpracování přesně podle našeho zadání
- Hotové řešení je téměř bez chyb
- Unikátní grafické zpracování
- Zabezpečení
- Optimalizace pro vyhledávače

Nevýhody

- Počáteční náklady mohou být vysoké
- Některé problémy nejsme schopni sami vyřešit

Zvolíme pronájem e-shopu

Pro velmi malý objem prodaného zboží nemusí být výhodné (tržby jsou menší než cena pronájmu). V tomto případě je ovšem potřeba si ujasnit, zda e-shop opravdu chceme provozovat.

Výhody

- Nemusíme se starat o zajištění domény, hosting
- Velmi rychlý způsob, jak uvést elektronický obchod do provozu

Nevýhody

- Nejsme fyzickými vlastníky obchodu, po ukončení pronájmu náš obchod zaniká
- Velmi omezené možnosti přizpůsobení vzhledu

4.8 Přínosy řešení

Firma doposud neprovozuje elektronický obchod, ale v rámci vylepšení své pozice na trhu se rozhodla e-shop založit. Navrhnutá databáze je prvním krokem k uskutečnění tohoto cíle. Již hotový e-shop používá databázi ke čtení, ukládání, úpravě a mazání dat. Některá data jsou přímo viditelná na webových stránkách obchodu (například název zboží, cena, popis atd.). Jiná zase nejsou přímo viditelná, ale jsou neméně důležitá (informace o zaměstnancích). Nesporná výhoda databáze tkví v tom, že umožňuje rychlou práci s daty při zachování nízkého objemu dat na disku.

Databáze je navržena tak, aby bylo možné vkládat a aktualizovat zboží, včetně jeho vlastností z databáze programu Tara. Tímto způsobem je zajištěn aktuální stav skladových zásob, aby nenastala situace, kdy v jedné databázi už zboží není skladem a ve druhé stále je. Firem, které používají Taru, je více, takže je možné použít návrh i v dalších lékárnách.

Ekonomický přínos je složité spočítat, protože se jedná pouze o návrh. Databáze není naplněna finálními daty a do doby, než bude vyřešen zbytek elektronického obchodu, je bez reálného využití. Po spuštění e-shop je očekávaný ekonomický přínos v podobě zvýšení objemu prodaného zboží a tudíž i tržeb. V tuto chvíli je tedy vytvořený návrh přínosem v podobě úspory času a lidských zdrojů.

Náklady na vytvoření návrhu databáze můžeme odhadnout pomocí nákladů ušlé příležitosti. Při sazbě 100 Kč za hodinu a pracovní době 6 hodin denně po dobu 23 dní dostaneme částku 13 800 Kč. Délka zpracování návrhu zahrnuje i důležité získání teoretických znalostí včetně analýzy firmy. Mohli bychom porovnat cenu návrhu databáze od profesionální firmy a cenu tohoto provedení. Nicméně zde je nutné podotknout, že firmy většinou zpracovávají kompletní řešení elektronického obchodu. Software SQL Server 2012 je v základní verzi pro studijní účely a vývojáře zdarma, dále je zpoplatněn podle konkrétní verze.

5 Závěr

Cílem této bakalářské práce bylo navrhnout databázi pro elektronický obchod firmy ADONIA v.o.s.. Nejdříve jsem studiem příslušných literárních zdrojů získal důležité teoretické znalosti. Poté jsem lékárně představil, zmínil používaný software, hardware a popsal, jaká je situace na trhu. Nakonec jsem zužitkoval teoretické znalosti při vytvoření návrhu.

Cíl byl splněn, databáze je navržena a funkční. Dílčí požadavky na výkon a zabezpečení jsou většinou pod správou DBMS. Korektně navržená databáze má ovšem taky svůj podíl na výkonu. Zálohování může být nastaveno poskytovatelem serveru nebo manuálně v SQL Serveru.

Databáze obecně, ač splňuje požadavky na ni kladené, není vhodná, aby fungovala samostatně. Je potřeba nějaká nástavba (například webová aplikace), která dokáže s databází pracovat a usnadní práci koncovým uživatelům. Proto se v současné době pracuje na řešení pro dokončení elektronického obchodu.

Seznam použitých zdrojů

Knižní zdroje

1. CONOLLY, T., BEGG, C., HOLOWCZAK, R. *Mistrovství – databáze: Profesionální průvodce tvorbou efektivních databází*. 1. vydání. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.
2. HOTEK, M. *Microsoft SQL Server 2008: Krok za krokem*. 1. vydání. Brno: Computer Press, 2009. 488 s. ISBN 978-80-251-2466-6.
3. KOCH, M., NEUWIRTH, B. *Datové a funkční modelování*. 3. přepracované vydání. Brno: Akademické nakladatelství Cerm, 2008. 121 s. ISBN 978-80-214-3731-9.
4. LACKO, L. *SQL Kapesní přehled*. 1. vydání. Brno: Computer Press, 2005. 96 s. ISBN 80-251-0788-4.
5. LACKO, L. *Jak vyzrát na SQL Server 2008*. 1. vydání. Brno: Computer Press, 2009. 469 s. ISBN 978-80-251-2101-6.
6. LACKO, L. *1001 tipů a triků pro SQL*. 1. vydání. Brno: Computer press, 2011. 416 s. ISBN 978-80-251-3010-0
7. OPPEL, J. A. *Databáze bez předchozích znalostí*. 1. vydání. Brno: Computer Press, 2006. 316 s. ISBN 80-251-1199-7.
8. OPPEL, J. A. *SQL bez předchozích znalostí*. 1. vydání. Brno: Computer Press, 2012. 240 s. ISBN 978-80-251-1707-1.
9. TAYLOR, A. G. *SQL for dummies*. 7th edition. Indianapolis: Wiley, 2010. 456 s. ISBN 978-0-470-55741-9

Internetové zdroje

10. Lékárna Adonis [online]. 2001 - 2003 [cit. 2013-04-25]. Dostupné z:
<http://www.lekarna-adonis.cz/>
11. RR - SERVIS. *Lékařský software pro WINDOWS* [online]. 2008 [cit. 2013-04-27]. Dostupné z:
<http://www.rr-servis.cz/?page=lekarensky-software-win>
12. W3schools.com [online]. 1999, 2010 [cit. 2013-04-25]. *SQL Data types*.
Dostupné z: http://www.w3schools.com/sql/sql_datatypes.asp

Seznam obrázků a tabulek

Seznam obrázků

Obr. 1: Požadavky na informaci	12
Obr. 2: Logo firmy.....	23
Obr. 3: Organizační struktura	24
Obr. 4: Síťové schéma	27
Obr. 5: Tara - úvodní obrazovka.....	28
Obr. 6: Tara - karta zboží.....	29
Obr. 7: Tara - příjem zboží	30
Obr. 8: Tara - zboží na skladových kartách	31
Obr. 9: Evidence nového zboží.....	36
Obr. 10: Registrace nového zákazníka	37
Obr. 11: Vytvoření objednávky	38
Obr. 12: Základní ER model.....	42
Obr. 13: ER model po dekompozici	43

Seznam tabulek

Tabulka 1: SWOT analýza.....	32
Tabulka 2: Košík.....	45
Tabulka 3: Zboží.....	45
Tabulka 4: Kategorie zboží.....	46
Tabulka 5: Obrázek.....	46
Tabulka 6: Výrobce	46
Tabulka 7: DPH	46
Tabulka 8: Měrná jednotka	46
Tabulka 9: Zákazník	47
Tabulka 10: Adresa zákazníka.....	47
Tabulka 11: Stát.....	48
Tabulka 12: Objednávka.....	48
Tabulka 13: Způsob platby	48
Tabulka 14: Způsob doručení	48
Tabulka 15: Stav objednávky	49
Tabulka 16: Faktura.....	49
Tabulka 17: Uživatel.....	49

Seznam použitých zkratk

ER model - Entity-Relationship model, entitně-relační model

FK - Foreign key, cizí klíč

ICT - Information and Communication Technologies, informační a komunikační technologie

OID - Object identifier, objektový identifikátor

PK - Primary key, primární klíč

RDBMS - Relational DataBase Management System, relační databázový systém

VDSL - Very high speed dsl

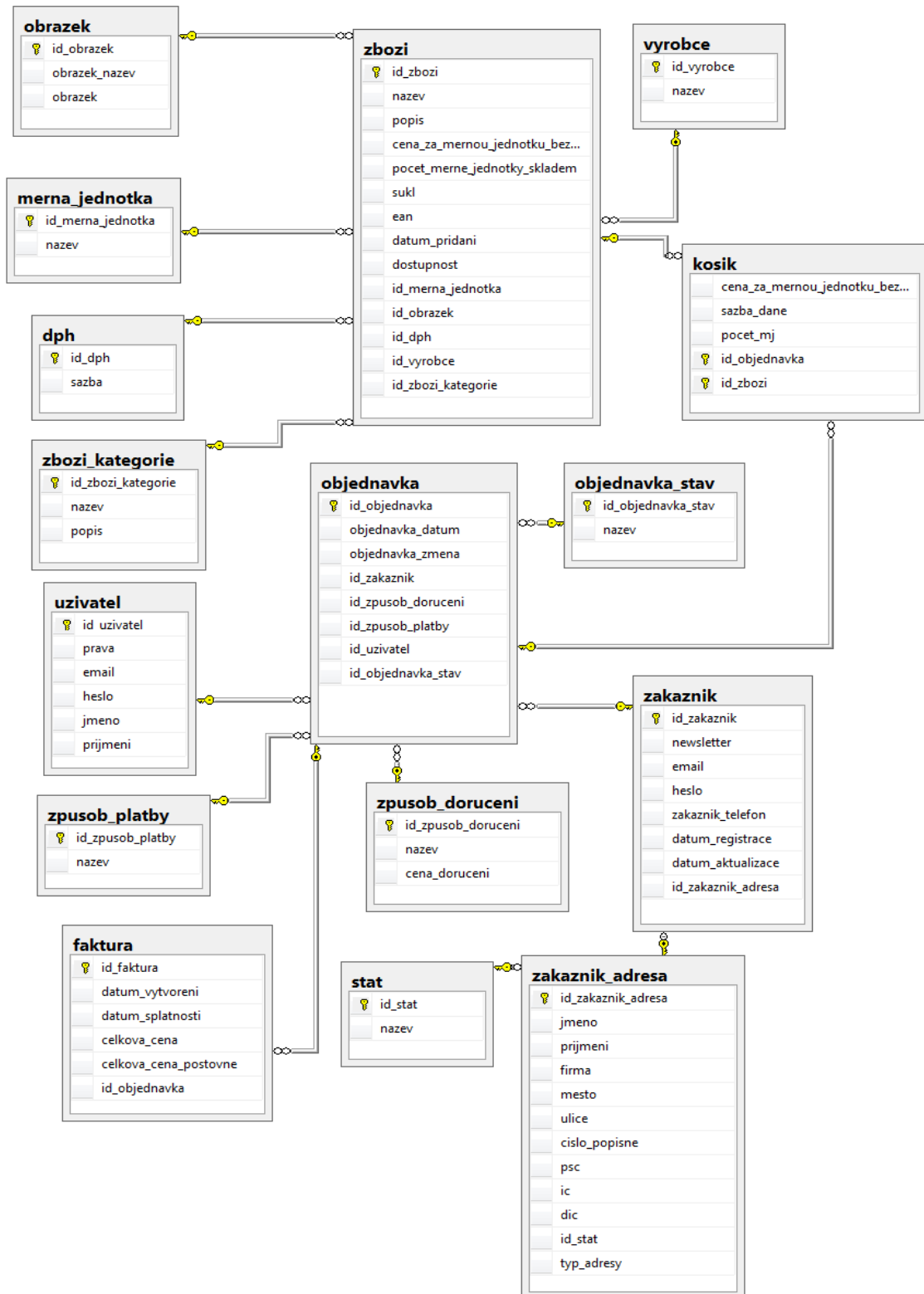
Seznam příloh

Příloha 1 - ER model

Příloha 2 - Zdrojový kód

Přílohy

Příloha 1 - ER model



Příloha 2 - Zdrojový kód

```
create database eshop
```

```
go
```

```
use eshop
```

```
go
```

```
-- Tvorba tabulek
```

```
-- Tabulka Uživatel
```

```
create table uzivatel
```

```
(
```

```
id_uzivatel int identity (1, 1) primary key,
```

```
prava tinyint not null,
```

```
email varchar (128) not null,
```

```
heslo varchar (32) not null,
```

```
jmeno varchar (35) not null,
```

```
prijmeni varchar (50) not null
```

```
)
```

```
go
```

```
-- Tabulka Objednávka stav
```

```
create table objednavka_stav
```

```
(
```

```
id_objednavka_stav int identity (1, 1) primary key,
```

```
nazev varchar (20) not null
```

```
)
```

```
go
```

```
-- Tabulka Způsob doručení
```

```
create table zpusob_doruceni
```

```
(
```

```
id_zpusob_doruceni int identity (1, 1) primary key,
```

```
nazev varchar (64) not null,  
cena_doruceni decimal (6, 2) not null  
)  
go
```

```
-- Tabulka Způsob platby  
create table zpusob_platby  
(  
id_zpusob_platby int identity (1, 1) primary key,  
nazev varchar (30) not null  
)  
go
```

```
-- Tabulka Stát  
create table stat  
(  
id_stat int identity (1, 1) primary key,  
nazev varchar (64) not null  
)  
go
```

```
-- Tabulka Zákazník adresa  
create table zakaznik_adresa  
(  
id_zakaznik_adresa int identity (1, 1) primary key,  
jmeno varchar (35) not null,  
prijmeni varchar (50) not null,  
firma varchar (64),  
mesto varchar (64) not null,  
ulice varchar (80) not null,  
cislo_popisne varchar (5) not null,  
psc varchar (12) not null,
```

```
ic varchar (13),
dic varchar (14),
id_stat int foreign key references stat (id_stat) not null,
typ_adresy tinyint
)
go
```

```
-- Tabulka Měrná jednotka
create table merna_jednotka
(
id_merna_jednotka int identity (1, 1) primary key,
nazev varchar (5) not null
)
go
```

```
-- Tabulka DPH
create table dph
(
id_dph int identity (1, 1) primary key,
sazba decimal (5, 3) not null
)
go
```

```
-- Tabulka Výrobce
create table vyrobce
(
id_vyrobce int identity (1, 1) primary key,
nazev varchar (64) not null
)
go
```

```
-- Tabulka Obrázek
```

```
create table obrazek
```

```
(  
id_obrazek int identity (1, 1) primary key,  
obrazek_nazev varchar (50) not null,  
obrazek image not null  
)  
go
```

```
-- Tabulka Zboží kategorie
```

```
create table zbozi_kategorie
```

```
(  
id_zbozi_kategorie int identity (1, 1) primary key,  
nazev varchar (64) not null,  
popis text  
)  
go
```

```
-- Tabulka Zákazník
```

```
create table zakaznik
```

```
(  
id_zakaznik int identity (1, 1) primary key,  
newsletter tinyint not null,  
email varchar (128) not null,  
heslo varchar (32) not null,  
zakaznik_telefon varchar (15) not null,  
datum_registrace date not null,  
datum_aktualizace date,  
id_zakaznik_adresa int foreign key references zakaznik_adresa (id_zakaznik_adresa)  
not null  
)  
go
```

-- Tabulka Zboží

create table zbozi

(

id_zbozi int identity (1, 1) primary key,

nazev varchar (80) not null,

popis text,

cena_za_mernou_jednotku_bez_dph decimal (8, 2) not null,

pocet_merne_jednotky_skladem decimal (5, 2) not null,

sukl varchar (7),

ean varchar (13) not null,

datum_pridani date,

dostupnost tinyint not null,

id_merna_jednotka int foreign key references merna_jednotka (id_merna_jednotka) not null,

id_obrazek int foreign key references obrazek (id_obrazek),

id_dph int foreign key references dph (id_dph) not null,

id_vyrobce int foreign key references vyrobce (id_vyrobce),

id_zbozi_kategorie int foreign key references zbozi_kategorie (id_zbozi_kategorie) not null

)

go

-- Tabulka Objednávka

create table objednavka

(

id_objednavka int identity (1, 1) primary key,

objednavka_datum date not null,

objednavka_zmena date not null,

id_zakaznik int foreign key references zakaznik (id_zakaznik) not null,

id_zpusob_doruceni int foreign key references zpusob_doruceni (id_zpusob_doruceni) not null,

id_zpusob_platby int foreign key references zpusob_platby (id_zpusob_platby) not null,

```
id_uzivatel int foreign key references uzivatel (id_uzivatel) not null,  
id_objednavka_stav int foreign key references objednavka_stav (id_objednavka_stav)  
not null  
)  
go
```

```
-- Tabulka Košík
```

```
create table kosik
```

```
(  
cena_za_mernou_jednotku_bez_dph decimal (8, 2) not null,  
sazba_dane decimal (5, 3) not null,  
pocet_mj decimal (5,2) not null,  
id_objednavka int foreign key references objednavka (id_objednavka) not null,  
id_zbozi int foreign key references zbozi (id_zbozi) not null,  
primary key (id_objednavka, id_zbozi)  
)  
go
```

```
-- Tabulka Faktura
```

```
create table faktura
```

```
(  
id_faktura int identity (1, 1) primary key,  
datum_vytvoreni date not null,  
datum_splatnosti date not null,  
celkova_cena decimal (10, 2) not null,  
celkova_cena_postovne decimal (7, 2) not null,  
id_objednavka int foreign key references objednavka (id_objednavka) not null  
)  
go
```