



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH DATABÁZE SQL PRO DROGISTICKOU FIRMU

PROPOSAL OF SQL DATABASE FOR A DRUGSTORE COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

IVO ZEGZULKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ KŘÍŽ, Ph.D.

BRNO 2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Zegzulka Ivo

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 databáze SQL pro drogistickou firmu

v anglickém jazyce:

Proposal of SQL Database for a Drugstore 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:

DOSTÁL, P.; KŘÍŽ, J. Databázové systémy. 2006. ISBN: 80-214-3064-8

KOCH, M.; NEUWIRTH, B. Datové a funkční modelování. Brno: Cerm, 2010. ISBN: 978-80-214-4125- 5

MOLINARO, A. SQL Kuchařka programátora. 1. vydání. Brno: Computer Press., 2009. s. 572. ISBN: 978-80-251-2617-2

VIERA, R. SQL Server 2000 Programujeme profesionálně. 1. vydání. Praha: Computer Press., 2001. s. 1170. ISBN: 80-7226-506-7

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

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

L.S.

Ing. Jiří Kříž, Ph.D.
Ředitel ústavu

doc. RNDr. Anna Putnová, Ph.D., MBA
Děkan fakulty

V Brně, dne 20.05.2012

Abstrakt

Tato bakalářská práce se zabývá návrhem databáze pro firemní využití v jazyku SQL. Výsledek by měl být použitelný pro podnik Iveta Šťastníková – výroba a distribuce drogerie, ve kterém se vyrábí drogistické zboží. Primárně by měla zjednodušit operace jako naskladnění, fakturace, příjem a výdej zboží. Nejdříve se seznámíme s teoretickými východisky práce, poté přejdeme na současný stav a samotnou analýzu. Výsledkem je návrh řešení, který by měl odpovídat potřebám majitele.

Abstract

This thesis deals with corporate database applications in SQL. The result should be applicable to business Iveta Šťastníková - production and distribution of toiletries, which is produced in dry goods. Primarily, should simplify operations such as stocking, invoicing, receipt and distribution of goods. First, we introduce the theoretical bases of work, and then move to the current state and the analysis itself. The result is a design solution that would meet the needs of the owner.

Klíčová slova

Databáze, normalizace, SQL, diagram, data, operace, proces, SŘBD, model, relace, atributy, vazby

Key words

A database, normalization, SQL, chart, data, operations, process, DBMS, model, relations, attributes, relationships

Bibliografická citace práce

ZEGZULKA, I. *Návrh databáze SQL pro drogistickou firmu*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2012. 68 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

.....

Zegzulka Ivo

Poděkování

Děkuji vedoucímu práce Ing. Jiřímu Křížovi, Ph.D za jeho připomínky, rady a příjemný přístup k mým nápadům a návrhům. Dále děkuji Iveta Šťastníková – výroba a distribuce drogerie za přístup k informacím a možnosti vytvořit tuto práci pro jejich účely.

Obsah

ÚVOD.....	11
1. VYMEZENÍ PROBLÉMU A CÍLE PRÁCE.....	12
1.1 Vymezení problému	12
1.2 Cíl práce.....	12
2. TEORETICKÁ VÝCHODISKA PRÁCE	13
2.1 Základní pojmy a definice	13
2.1.1 Databáze	13
2.1.2 SŘBD.....	13
2.1.3 Jazyk SQL	13
2.1.4 Data a informace	14
2.2 Základy datového modelu.....	14
2.2.1 Základní pojmy	14
2.2.2 Datová položka.....	14
2.3 Datové modely	15
2.3.1 Lineární datový model.....	15
2.3.2 Hierarchický datový model	15
2.3.3 Síťový datový model.....	16
2.3.4 Relační datový model.....	16
2.3.5 Objektový datový model	16
2.4 Relační databáze	17
2.4.1 Klíče	17
2.4.2 Integrita.....	17
2.4.3 Normalizace.....	18
2.5 Metodologie návrhu.....	18
2.5.1 Konceptuální návrh	18
2.5.2 Logický návrh.....	19
2.5.3 Fyzický návrh	19
2.6 Microsoft SQL Server.....	20
2.6.1 Možnosti produktu.....	20
2.6.2 Používané datové typy	21
3. Analýza problému a současné situace	24

3.1	Základní údaje o firmě.....	24
3.1.1	Výrobní sortiment	24
3.1.2	Konkurence	25
3.2	Popis současné situace.....	26
3.2.1	Stav zboží ve skladě a jeho výroba	26
3.2.2	Prodej zboží.....	26
3.2.3	Vozový park.....	27
3.2.4	Prodejci	27
3.3	Analýza problému	27
3.3.1	Skladové operace	27
3.3.2	Prodejní operace	28
3.3.3	Prodejci a jejich automobily.....	28
3.4	Popis alternativních komerčních databázových skladů	28
3.4.1	TRELL	28
3.4.2	TROP.....	30
3.4.3	WareHouse EU	32
4.	Vlastní návrhy řešení, přínos návrhů řešení.....	34
4.1	Zachycení požadavků a procesů.....	34
4.1.1	Prodej zboží.....	34
4.1.2	Registrace prodejce.....	36
4.1.3	Údržba vozu.....	37
4.2	Konceptuální model	38
4.3	Logický návrh.....	38
4.3.1	Prodejce, automobil, teritorium	40
4.3.2	Zboží a jeho detaily, sklad a umístění zboží	41
4.3.3	Objednávka, faktura a metody placení	43
4.3.4	Objednání zboží ze skladu	43
4.4	Fyzický návrh	44
4.4.1	Procedura registrace prodejce.....	44
4.4.2	Procedura přidělení automobilu	46
4.4.3	Procedura nová karta zboží.....	47
4.4.4	Procedura údržba vozu	48
4.4.5	Trigger nízký stav zboží.....	49

4.4.6	Triggery pro mazání dat	49
4.4.7	Přínosy databáze	50
5.	Závěr.....	51
6.	Seznam literatury	52
7.	Seznam obrázků a tabulek	54
7.1	Seznam obrázků	54
7.2	Seznam tabulek.....	54
8.	Seznam příloh.....	55
8.1	Datový slovník	55
8.2	Zdrojový kód.....	58

ÚVOD

V dnešním světě prolínají informační a komunikační technologie veškerý průmysl, obchodování a přenos informací mezi lidmi. I přes veškeré výhody, zjednodušení práce, zrychlení firemních procesů a centralizaci můžeme stále nacházet firmy, podniky či samostatné podnikatele, kteří zastávají konzervativní přístup k podnikání.

Majitel vlastní několik počítačů – především přenosných notebooků, které však slouží pouze k vyřizování e-mailů. Na jednom z nich se pak provádí účtování pomocí účetního programu. Nicméně jsem nenašel žádný software, který by zajišťoval data o zboží, cenách, naskladnění, vyskladnění apod.

Proto jsem se rozhodl pro tuto práci – návrh databáze šité na míru majiteli. Měla by být jednoduchá, kvalitní a postihnout všechny aspekty potřebné k úspěšnému podnikání. Proto si v první části vymezíme základní cíle a problémy.

Druhá část jsou teoretická východiska, nutná k řešení. Pokusím se probrat ty nejdůležitější informace, které v plné výši využiju při samotném návrhu. Řekneme si, co jsou to SQL databáze, jaké prvky obsahují a zahrneme zde i postup při tvorbě.

Třetí a neposlední část je analýza současného problému, zde shrneme současný stav v podniku, jakým způsobem probíhají jednotlivé procesy, kde vidíme nedostatky a chyby. Pokusím se Vám přiblížit problémy podniku, z nichž vyplynou návrhy na zlepšení.

Poslední čtvrtá část je samotný návrh vlastního řešení. Vytvoření fyzického a logického návrhu, vztahy mezi relacemi (tabulkami) a několik skriptů jako příklad. Nakonec zhodnotíme přínosy této práce, jak pro mou osobu, tak pro samotný podnik.

1. VYMEZENÍ PROBLÉMU A CÍLE PRÁCE

1.1 Vymezení problému

Iveta Šťastníková – výroba a distribuce drogerie využívá stále ve velké míře papírové záznamy o zboží, prodejích, stavu skladu a objednávkách. Vyúčtování se zákazníky probíhá klasicky pomocí papírových faktur. Spousta informací, které by mohly pomoci v rozvoji podnikání, není vůbec uchováváno. Existují databázové aplikace, které při zakoupení mohou být okamžitě uvedeny do provozu, problém je v tom, že nemusí přesně vyhovovat představám a potřebám. Taktéž vyšší cena může být problémem a kritériem při rozhodování.

Proto jsem se rozhodl vytvořit návrh databáze, vyhovující podnikateli. Vymezit správná data, která chce uchovávat a usnadnit tak práci s informacemi. Měla by přispět k formě klasických papírů, dokladů a mluveného slova. Efektivnost, přehlednost a zrychlení procesů je nejdůležitějším měřítkem pro samotný návrh.

1.2 Cíl práce

Hlavním cílem je tedy databáze pro Iveta Šťastníková – výroba a distribuce drogerie. Evidovat bude výrobky na skladě, objednávky, naskladňování a prodej. Dále pak materiál na výrobu zboží, informace o výrobě, prodejce, kteří výrobky distribuují, a nakonec informace o vozovém parku (počet najetých kilometrů, výměna oleje a jiné).

Tento návrh by mohl posloužit pro následné naprogramování aplikace a zavedení do praxe. Dále zde popíšeme základy SQL jazyka a jeho vlastnosti.

2. TEORETICKÁ VÝCHODISKA PRÁCE

2.1 Základní pojmy a definice

2.1.1 Databáze

Rozumíme kolekci datových položek vzájemně souvisejících a spravovaných jako jedna jediná jednotka. Přesný význam se liší podle známých výrobců softwaru, poskytujících databázové systémy (Oracle Corporation vs. Microsoft SQL server). Na databázový objekt, který je uložen v databázi jako pojmenovaná datová struktura, můžeme nahlížet více způsoby – například jako tabulka, index nebo pohled. U některých výrobců databází se data ukládají do více souborů, u firmy Microsoft pouze do jednoho. Hlavní výhodou relačních databází je oddělení fyzické implementace od logické definice, tudíž uživatel nemusí v průběhu práce vůbec přijít do styku s uloženými soubory (1).

2.1.2 SŘBD

Neboli systém řízení báze dat, na které pohlížíme jako na entitu informací, o kterých chceme mít přehled, zajímají nás a hlavně jsou definovatelné. Toto je základ každé databáze. Za entitu lze považovat zboží ve skladu, zaměstnance podniku, služby apod.

Entitou databáze zboží je výrobek, každý záznam obsahuje informace o jednotlivém výrobku (detaily jako cena, barva, popis, balení).

SŘBD taktéž zajišťuje prostředky, jak data uchovávat na disku, třídít, provádět údržby (aktualizace, přidávání a rušení záznamů). Uživatel musí mít k dispozici všechny služby, které jsou nutné k manipulaci s daty. Dále dokáže zobrazovat data na obrazovce, terminálu nebo tištěnou formou. Důležitou funkcí je zajištění integrity dat (2).

2.1.3 Jazyk SQL

Mezi SŘBD a koncovým uživatelem musí figurovat i databázová aplikace, která zajistí přístup k datům. Takové aplikace píší v drtivé většině programátoři pomocí specializovaných programovacích jazyků. Trend uživatelsky orientovaných nástrojů pro práci v databázích jsou stále více využívány. Zjednodušují a zrychlují práci – eliminace programování (MS Access a jiné).

SQL – structured query language neboli strukturovaný dotazovací jazyk, původně navržen pro relační modely, které komunikovali se SŘBD. Historicky se objevila první verze pod názvem SEQUEL v 70. letech, vyvinuta firmou IBM.

Hlavní použití je pro interaktivní dotazy nad databázemi (dynamické SQL), dále pak jako části aplikací tvořených v procedurálních jazycích (vložené SQL). Přímé použití – nejpoužívanější příkaz SELECT. SQL dokáže zpracovat i jiné než relační databáze. Podmínkou je však dodržení již zmiňované integrity dat. Vlastnosti jako podpora procedur, vzdálených procedur, triggerů, ODBC jsou velice důležité a zlepšují podmínky pro vývoj databáze (2).

2.1.4 Data a informace

Můžeme chápat jako technický záznam, který lze přenést a zpracovat. Toto lze realizovat pomocí různých médií a technik – papírový výtisk, CD-Rom, textový soubor v počítači, hlas v mobilním telefonu.

Data jsou kódována. Není to však forma zašifrování, jedná se pouze o zaznamenání informace. Samotnou informaci můžeme chápat jako vjem, splňující tři základní požadavky – syntaktická relevance (schopnost dekodovat informaci), sémantická relevance (pochopit její obsah), pragmatická relevance (musí mít pro příjemce nějaký význam. Znalost, kterou získáme z dané informace, vznikne na základě zkušenosti uložené v mozku (4) (5).

2.2 Základy datového modelu

2.2.1 Základní pojmy

Datový objekt (reálný) je reprezentovaný entitou. Definovaný je pomocí atributů, entit, položek, které uchováváme. Struktura objektu (věta) je dána konečnou množinou prvků.

2.2.2 Datová položka

Údaj, který je atomický (dále nedělitelný), za předpokladu, že jako položku neuvažujeme skupinu údajů – adresa (město, ulice, PSČ). Položky jsou definovány typem a délkou (počet znaků, které do nich můžeme uložit).

- Libovolné znaky

- Pouze číslice
- Datum a čas

Zmínil jsem již skupinové položky neboli složené atributy. Zde se rozhodujeme, jak důležité je mít tyto informace uložené, jestli v celku nebo rozdělené na jednotlivé části. Taktéž existují tzv. číselníky, které využijeme při konečném a relativně malém počtu záznamů. Vhodné například při vyplňování formulářů – rodinný stav, muž/žena, apod. Tyto informace jsou důležité pro další kapitolu o datových modelech (4).

2.3 Datové modely

Při projektování informačních systémů a databází vytváříme obrazy, které musí odpovídat realitě tak, že data do nich vložená zcela odpovídají skutečnosti. V dnešní době známe 5 hlavních typů těchto modelů, 2 z nich současné databázové systémy nepodporují (4).

- Lineární
- Hierarchický
- Síťový
- Relační
- Objektový

2.3.1 Lineární datový model

V lineárním modelu neexistují žádné vazby (výjimkou je pouze předchůdce vs. následovník) mezi tabulkami databáze. Jedinou výhodou tohoto dnes poměrně zastaralého typu je možnost implementace na libovolné médium, například na papír, do kartotéky, na optické médium a jiné (4).

2.3.2 Hierarchický datový model

Data hierarchicky uspořádaná do podoby stromu (obráceného), kde tabulka která je na vrcholu se považuje za „kořen“, ostatní tabulky pod ní označujeme za větve. Správná

terminologie je však rodič a potomek. Výhodou tohoto typu je rychlost vyhledávání dat ve stromu. Nejvíce využíván v dobách magnetických pásek (7).

2.3.3 Sít'ový datový model

Hlavním cílem tohoto modelu mělo být vyřešení problému, který se vyskytoval u výše zmiňovaného hierarchického modelu. Ve struktuře se objevují uzly a množinové struktury. Vazby už nejsou jen z rodiče na potomka, ale obecně v jakémkoli směru a k libovolné tabulce, což znamená okamžitý přístup k potřebným datům. Výhodou je opět rychlost vyhledávání dat (8).

2.3.4 Relační datový model

Relační model tvoří více tabulek, které obsahují funkční vztahy, indexy a další součásti – relační databázi. Výhodami je často velice přirozená reprezentace zpracovaných dat, snadné definování vazeb (11). Položky, které spojují tabulky, nazýváme relační klíče. Taková spojení nejsou trvalá, vznikají při potřebě zobrazení dat s více tabulek najednou. Zanikají při ukončení práce s nimi (4).

2.3.5 Objektový datový model

Nejnovější model z výše popsaných. Hlavním prvkem je zde – OBJEKT. Oproti jiným modelům obsahuje objekt jak jednotlivé atributy, tak i definované metody, podle kterých se poté chová. Typově stejné objekty tvoří třídy objektů, konkrétní záznam z objektu nazýváme instance. Objekty mají své identifikátory – OID, které zajišťují přímé vazby podobné vazbám síťového modelu. Existovat zde však mohou i relační vazby.

Dalším rysem objektového modelu je „zapouzdření objektu“. Pokud tedy s objektem chceme pracovat, je nutné volat metody objektu – datová abstrakce a nezávislost dat (1).

2.4 Relační databáze

Relační databáze musí splňovat vždy určitá pravidla. Prvním z nich je to, že každou informaci, kterou uložíme, můžeme vždy nalézt jako hodnotu v tabulce. Všechny data lze jednoznačně najít pomocí názvu tabulky, sloupce a hodnoty primárního klíče.

Často se v tabulkách objevují prázdná pole bez hodnot, v takovýchto případech je vhodné ošetření za pomoci hodnoty NULL. Jazyk, kterým ovládáme chod databáze, musí podporovat potřebné funkce, nezbytné k práci. O nezávislosti fyzické a logické jsem se již zmínil u obecného pojmu databáze. Existuje ještě několik dalších pravidel, jako nezávislost integrity, distribuce a jiné (1).

2.4.1 Klíče

V databázi není příliš vhodná duplicita záznamů, vede ke zpomalení výkonu, ztrátě integrity, či jiným škodlivým faktorům. Proto vytváříme klíče, které poté přiřazujeme jednotlivým entitám a jednoznačně je tím identifikujeme (10).

Existují 3 typy klíčů:

- **Kandidátní klíč** – je jeden nebo více sloupců daného záznamu, navíc je tento klíč jednoznačný a dále neredukovatelný.
- **Primární klíč** – jeden vybraný klíč z množiny kandidátních.
- **Cizí klíč** – je klíč, který poukazuje na primární klíč v jiné tabulce a tudíž slouží k propojení mezi oběma tabulkami.

2.4.2 Integrita

Entitní integrita – primární klíč nesmí být prázdný nebo obsahovat nulu (hodnota NULL)

Referenční integrita – jde o vytvoření záznamu, kde do hodnoty cizího klíče zapisujeme existující hodnotu primárního klíče, který se nachází v jiné tabulce. Pokud tento záznam nenajdeme, ukládáme hodnotu NULL.

2.4.3 Normalizace

Normalizace je proces, při kterém se databáze upravuje do výsledné podoby tak, aby splňovala určité požadavky. Odstranění nedostatků, zabránění redundance či zamezení ztráty dat. Normalizace má několik kroků, kterým se říká normální formy. Ve většině dnešních databází stačí první tři normální formy. Ve skutečnosti je jich 5 (1).

1. normální forma – atomičnost, neboli obsahuje dále nedělitelné atributy. Pokud však vyžadujeme složený nebo vícehodnotový atribut (adresa), záleží na uvážení, je-li to v rozporu s touto formou.

2. normální forma – funkční závislost, což znamená, že každý neklíčový atribut je plně závislý na celém primárním klíči, dále musí splňovat i podmínku 1. NF.

3. normální forma – tranzitivní závislost, zajišťuje vzájemnou nezávislost mezi neklíčovými atributy, taktéž musí být i v 2. NF.

BC. normální forma – Boyce-Coddova variace třetí normální formy, kde relace obsahuje více než jeden kandidátní klíč (podmínkou je složený klíč). Tyto kandidátní klíče se v některých z atributů navzájem překrývají. Dále musí splňovat požadavky z 2. NF.

4. normální forma – multizávislost znamená, že veškeré vícehodnotové závislosti jsou i funkčními závislostmi z kandidátních klíčů. Taktéž je v Boyce-Coddově formě.

5. normální forma – cykličnost a omezení, kde je nutné spojení všech relací, jinak nelze dospět k správným výsledkům. Navíc relaci nelze rozložit bezztrátově.

2.5 Metodologie návrhu

2.5.1 Konceptuální návrh

Nejdůležitějším výstupem této části návrhu je konceptuální schéma, které výstižně popisuje uživatelské požadavky na data, datové typy, vztahy a integritní podmínky. Výhodou této části je především srozumitelnost pro obyčejné lidi, bez technických znalostí. Zde neřešíme implementační prvky – programování, znalost SQL jazyka či serveru (1).

1. část: Identifikace entit a relací mezi nimi – hlavní objekty pro ukládání dat, vztahy mezi nimi (většinou mezi dvěma – relace binární).

2. část: Identifikace a určení atributů – údaje uložené v entitách a jejich přípustné hodnoty, velikost nebo formát.

3. část: Nalezení kandidátních klíčů a určení primárních – v každé entitě nejvhodněji vybereme atributy pro vytvoření primárního klíče.

4. část: Kontrola redundance, uživatelských transakcí, posouzení správnosti – nesmí obsahovat vazby 1:1, podporuje a splňuje veškeré uživatelské požadavky.

Výstupem obvykle bývá Entito-Relační diagram, který shrnuje dosavadní zjištění.

2.5.2 Logický návrh

Čerpá z konceptuálního návrhu. Je taktéž rozdělen do několika kroků, pomocí kterých vytvoříme výslednou podobu tabulek (1).

1. část: Z výstupu ER modelu vytvoříme tabulky pro entity, všechny jejich atributy, primární a cizí klíče a integritní omezení.

2. část: Normalizace tabulek – všechny tabulky v této části návrhu se nacházejí nejméně ve 3. normální formě, pokud tomu tak není, je nutné provést vhodné úpravy.

3. část: Kontrola požadavků transakcí – při převodu z konceptuálního ER diagramu nesmí dojít k chybě

4. část: Kontrola integrity – Not NULL, multiplicita, entitní a referenční integrita, omezení domén a další.

2.5.3 Fyzický návrh

Popisuje proces implementace databáze, organizaci souborů, indexování a zabezpečení dat. Veškeré činnosti se reálně provádějí na vnějších paměťových zařízeních. Implementace se odvíjí na základě cílového DBMS (1).

1. část: Převod logického návrhu databáze – návrh tabulek pro konkrétní DBMS, reprezentace odvozených dat a ostatní integritní omezení.

2. část: Organizace souborů – u menších databází není obvykle potřeba, slouží především ke zvýšení výkonnosti databáze. Analyzují se operace a transakce, soubory jsou vhodně organizovány. Výhodné je i použití indexů (dodatečných klíčů).

3. část: Uživatelské pohledy – návrh podle předem stanovených požadavků uživatelů (spojení vybraných tabulek a vhodných sloupců pro efektivní zobrazení dat).

4. část: Bezpečnostní mechanismy – důležitým aspektem při práci s databází je bezpečnost uchovávání dat, nastavení pravomocí pro uživatele, přihlašovací údaje, spouště a jiné.

5. část: Redundance – u některých databází se může jevit redundance jako zbytečná a zpomalující, na druhou stranu porušení některých pravidel může přinést jisté vylepšení výkonu. Tato praktika je však využívána pouze u velkých databází náročných na výpočet.

2.6 Microsoft SQL Server

Pro provoz databází je na trhu spousta produktů, které se liší svými funkcemi, cenovou politikou, uživatelským rozhraním a komplexností nabízených služeb. Firmy jako Oracle, Microsoft či IBM jsou velkými hráči na poli databázových aplikací, jednou z nich je MS SQL Server 2008 (3).

2.6.1 Možnosti produktu

Spolehlivá škálovatelnost databázového provozu a zlepšení efektivity vývoje informačních technologií jsou základním měřítkem tohoto produktu. Nechybí podpora business intelligence, analytická řešení a vše dokáže zobrazit jako srozumitelný přehled.

Hlavním modulem je modul úložiště. Zpřístupňuje aplikacím data a řídí i jejich ukládání. Jedna z interních komponent MS SQL Serveru, obsahující i dílčí doplňky. Systém zabezpečení je zajišťován buď pomocí Windows autentizace, nebo přes interní účty serveru.

Výhodou programovacího rozhraní je možnost rozšíření základního Transact-SQL (T-SQL) o jiné jazyky podporující CLR moduly – C++, C#, Microsoft Visual Basic a jiné.

Od verze 2005 je integrován systém řazení zpráv do fronty (Service Broker), plánování a výstrahy (SQL Server Agent), umožňuje kopírování dat a jejich synchronizaci s hlavní datovou sadou (3).

2.6.2 Používané datové typy

MS SQL Server 2008 podporuje širokou paletu datových typů v jazyce SQL. Zde vypíšeme ty nejpoužívanější z nich (6).

Numerické typy:

- **TINYINT** – celé číslo, rozsah hodnot **0 až 255**
- **INT** – celé číslo, rozsah hodnot **-2^{31} až $2^{31}-1$**
- **SMALLINT** – celé číslo, rozsah hodnot **-32,768 až 32,767**
- **BIGINT** – celé číslo, rozsah hodnot **-2^{63} až $2^{63}-1$**
- **DECIMAL** (n, m) – peněžní data, rozsah hodnot **$-10^{38}+1$ až $10^{38}-1$**
- **NUMERIC** (n, m) – peněžní data, rozsah hodnot **$-10^{38}+1$ až $10^{38}-1$**
- **MONEY** – peněžní data (omezení maximálně na 4 desetinná čísla), rozsah hodnot **$\pm 922,337,203,685,477.5808$**
- **FLOAT** (n), **REAL** – přibližné datové typy s plovoucí desetinnou čárkou, což znamená, že čísla uložená v databázi nejsou zcela přesná a závisí na architektuře procesoru. Rozsah u typu
- **REAL** má rozsah **-3.4^{38} až -1.18^{38} , **0**, **1.18^{38} až 3.4^{38}****
- **FLOAT** má rozsah **-1.79^{308} až -2.23^{308} , **0**, **2.23^{308} až 1.79^{308}****

Znakové typy (textové):

Definují se s pevnou nebo proměnnou délkou n.

- **CHAR** (n) – maximálně 8 000 znaků, pokud je řetězec menší než n, je doplněn prázdnými znaky.

- **VARCHAR** (n) – maximálně 8 000 znaků, pokud je řetězec menší než n, není doplněn prázdnými znaky.
- **TEXT** – maximální velikost až 2 GB
- **NCHAR** (n) – stejné jako u CHAR, pouze pro kódování UNICODE
- **NVARCHAR** (n) – stejné jako u VARCHAR, pouze pro kódování UNICODE
- **NTEXT** – stejné jako u TEXT, pouze pro kódování UNICODE

Typy pro datum a čas

- **DATETIME** – umožňuje ukládat data jako jedinou hodnotu, rozsah je **01/01/1753 až 12/31/9999**
- **DATE** – rozsah je **01/01/0001 až 12/31/9999**
- **TIME** – rozsah je **00:00:00.0000000 až 23:59:59.9999999**
- **DATETIMEOFFSET** – slouží k ukládání časových pásem, nutný požadavek na lokalizaci času a data, rozsah je **01/01/0001 až 12/31/9999**

Binární datové typy

- **BIT** – hodnota NULL nebo 0/1
- **BINARY** – pevná šířka, maximálně 8000 bajtů
- **IMAGE** – proměnná šířka, velikost až 2 GB

Další datové typy

- **XML** – pro manipulaci s dokumenty XML o maximální velikosti 2GB anebo do 128 úrovní.
- **FILESTREAM** – ukládání nestrukturovaných dat (BLOB).
- **GEOMETRY** – datový typ sloužící s ukládání geometrických objektů, jako jsou body, čáry, křivky a polygony.

- **GEOGRAPHY** – datový typ, do kterého lze ukládat hodnoty zeměpisné šířky a délky
- **HIERARCHYID** – dokáže strukturovat hierarchická data, tudíž můžeme ukládat vývojové diagramy, organizační schémata a jiné.

3. Analýza problému a současné situace

Hlavním cílem kapitoly analýza je zhodnocení stávajících procesů, jak méně častých, tak i těch které se každodenně opakují. Poté je zde navrženo a porovnáno i několik databázových aplikací, vhodných pro podobné potřeby firem jako je firma Iveta Šťastníková – Výroba a distribuce drogerie.

3.1 Základní údaje o firmě

Iveta Šťastníková – výroba a distribuce drogerie

- Poštovní 533, 749 01 Vítkov
- Tel./Fax: 556 300 153, Mobilní telefon: 603 265 529
- IČ: 46107930
- DIČ: CZ6152091539

Právní forma: Fyzická osoba podnikající dle živnostenského zákona nezapsaná v obchodním rejstříku

- Datum vzniku: 1. prosinec 1997

Předmět podnikání (činnosti) dle statistického úřadu ČR

- Výroba ostatních chemických výrobků
- Nezařazeno
- Opravy a údržba motorových vozidel, kromě motocyklů

3.1.1 Výrobní sortiment

Tato firma se od založení v roce 2007 zabývá výrobou a prodejem pracích prášků (kvalitních profesionálních prášků používaných i v prádelnách), jarů, aviváží, pěn do koupelen, univerzálních čističů, pracích gelů a ostatního drogistického zboží (balzámy, šampóny, dílenská mýdla, WC čističe, sava, odstraňovače skvrn, písky, toaletní papíry). Sortiment je flexibilní a odvíjí se podle potřeb zákazníku.

Obr. 1 Výrobní sortiment



Zdroj: Arix. *Katalog*. [online]. 2007 [cit. 2012-1-25]. Dostupné z:
<http://www.arix.cz/images/TMP7.JPG>

3.1.2 Konkurence

Dnešními největšími konkurenty jsou Flipper, In a Sonet. Tyto firmy však nemají zdaleka takové zázemí, ceny jejich výrobků jsou předražené a hlavně kvalita je velice nízká.

Expanze podniku dopomohla k větším ziskům, nákupu dalších firemních automobilů a příjmu více prodejců.

Portfolio zákazníků je především:

- Restaurační zařízení
- Autoservisy
- Domovní prodej
- Zdravotnické zařízení,
- Domovy důchodců,
- Ústavy soc. péče apod.

Konkurenci se podnik snaží porazit pomocí různých obchodních postupů, jako akce a zvýhodnění cen při nákupu více zboží nebo pravidelném odběru. V této chvíli zavedla tzv. dumpingové ceny a tím velice ztížila ostatním firmám situaci na trhu. Dalším

motivačním činem je i podpora prodejců, aby více prodávali a tím zvýšili své obraty, jelikož jejich plat se odvíjí od provize.

3.2 Popis současné situace

Všeobecně nejvíce využívaná forma papírového zapisování veškerých údajů o prodeji, kusé informace o svých odběratelích, téměř žádný obraz o stavu naskladněného zboží (pokud vynecháme optickou kontrolu samotného majitele) může být pro podnikání překážkou a může bránit samotnému rozvoji firmy.

3.2.1 Stav zboží ve skladě a jeho výroba

Výroba zboží probíhá na jiném místě, než je umístěn centrální sklad, ve kterém se nachází suroviny pro výrobu. Suroviny se převážejí do výroby (zde probíhá míchání chemikálií, obarvování, parfémování, stáčení do nádob, štítkování a ukládání do palet), odkud poté zpět putují do skladu, kde jsou přístupné prodejcům, i běžným zákazníkům. Tato metoda není příliš vhodná, a může se zdát zbytečně zdlouhavá a nákladově náročná. Nicméně prostory nedovolují jinou organizaci.

Největším problémem je častý nedostatek některého zboží. Příčinou je nezkontrolování stavu zásob. Určité zboží je kupováno častěji než jiné, což znamená, že oblíbené je vykoupeno téměř okamžitě, zatímco jiné leží na skladu i více než měsíc. Majitel musí často chodit přímo do skladu a kontrolovat, co dochází a čeho je zatím dostatek.

3.2.2 Prodej zboží

Hlavními odběrateli zboží této firmy jsou prodejci, kteří zprostředkovávají distribuci po České republice (především Morava a Slezsko). Většina z nich nakupuje sortiment drogistických výrobků na papírovou fakturu. Zbytek kupuje zboží hotově.

Obyčejný zákazník, který přijde do skladu, může zboží zakoupit pouze za peníze. Jeho volbou je, jestli si vyžádá doklad o zaplacení či ne. Zbytek zboží, které je prodáváno do výše zmiňovaných institucí, jako jsou domovy důchodců, zdravotnické zařízení a jiné, je možné zaplatit hotově nebo elektronicky na bankovní účet firmy.

Základním nedostatkem takovýchto fakturací je skladnost – spousta papírových dokladů různých typů, které zabírají místo. Hledání v nich je zdlouhavé a nepřehledné,

často dochází k jejich ztrátě či poškození. U prodeje, který není doložen žádným dokladem, je nemožné se po pár dnech orientovat.

3.2.3 Vozový park

Majitel firmy nakoupil od svého vzniku již několik automobilů pro rozvoz zboží až do domu. Jednalo se převážně o vozy značky Citroen (typy C15 a Berlingo), které jsou velice spolehlivé a vhodné pro převoz výrobků tohoto typu.

S narůstajícím počtem automobilů ovšem rostly i náklady na chod vozidel – pneumatiky, výměna oleje a opravy všeho druhu. Proto se rozhodl podnikatel rozšířit svou činnost i o údržbu a opravy. Vozy jsou propůjčeny bezplatně do užívání prodejčům, kteří s nimi ovšem ne vždy zacházejí opatrně. Tím přichází podnikatel do styku s častými poruchami, kterým by šlo předcházet pravidelnou kontrolou základních prvků, jako stav vody či oleje. Do některých vozů byla instalována GPS navigace pro hlídání polohy, došlo k tomu po problémech s nepřizpůsobivými lidmi, kteří je využívali k jiným účelům, než byl prodej.

3.2.4 Prodejci

Ve spoustě kapitol jsme se již zmínili o prodejcích. Nicméně u všech majitel nemá ucelené informace (bydliště, lokality rozvozu zboží, průměrné tržby a jiné), které by pomohli organizovat prodej centrálně.

3.3 Analýza problému

Na základě zhodnocení dosavadních problémů v této kapitole lze připravit návrh pro relační databázi, která by je dokázala odstranit. Ušetření papírových prostředků, zpřehlednění a vhodná struktura je cílem této práce.

3.3.1 Skladové operace

Vytvoření záznamu o výrobě či nákupu od subdodavatele určitého typu zboží by mohlo proběhnout pomocí jednoduchého formuláře, na kterém lze vybrat základní informace – typ, barva, vůně, cena atd. a popis produktu – detailnější.

Databáze by měla uchovávat aktuální stav skladu, kolik kusů různých výrobků je právě naskladněno, pokud bude některý výrobek docházet, vytvoří se požadavek na jeho vyrobení či objednání, který bude odeslán do výroby, čímž docílíme kompaktnosti a

integrity zásob. Při odebrání prodejcem či zakoupení zákazníkem bude patřičné zboží ze skladu odečteno. Na základě těchto operací bude možné tisknout faktury či účtenky, které by mohli nahradit klasické „ruční“ zapisování.

3.3.2 Prodejní operace

Při prodeji by do naší databáze mělo být zaznamenáno, kdy byl uskutečněn (to pro případ zpětné kontroly, nespokojenosti či reklamaci), dále pak jaké zboží figuruje na faktuře, komu je účtováno a jakým způsobem (hotově, na fakturu, bankovním převodem).

Databáze propojená s webovou aplikací – elektronický obchod firmy by mohl sloužit pro oslovení nových potencionálních zákazníků, kde by si zákazník jednoduše vybral, jaké zboží chce do košíku uložit, okamžitě by viděl, jestli je tohoto výrobku dostatek a kdy bude popřípadě doručeno.

3.3.3 Prodejci a jejich automobily

U každého prodejce bude požadavek na uchování jeho iniciály, adresy, telefonních čísel a jiných informací, která jsou pro podnikatele důležité. Dalším bodem je pak uložení teritorií a měst, do kterých jednotliví prodejci rozvázejí zboží.

Každý prodejce vlastní automobil, ty které patří do firemního vozového parku, budou uloženy v databázi. Důležitou položkou bude počet najetých kilometrů, od kterého se bude odvíjet servis, a to především výměny kapalin, rozvodů, filtrů apod. Pokud se bude blížit stav kilometrů dané hodnotě, databáze upozorní, popřípadě pokud uvede prodejce i svůj email odešle zprávu o dopravení vozu do servisu.

3.4 Popis alternativních komerčních databázových skladů

V této kapitole se pokusíme nastínit možné systémy, které jsou nabízeny firmami pro podobné účely, jaké má splňovat naše databáze. Cena a složitost takovýchto produktů se liší a je odvozena od obsáhlosti a možností, které nabízí.

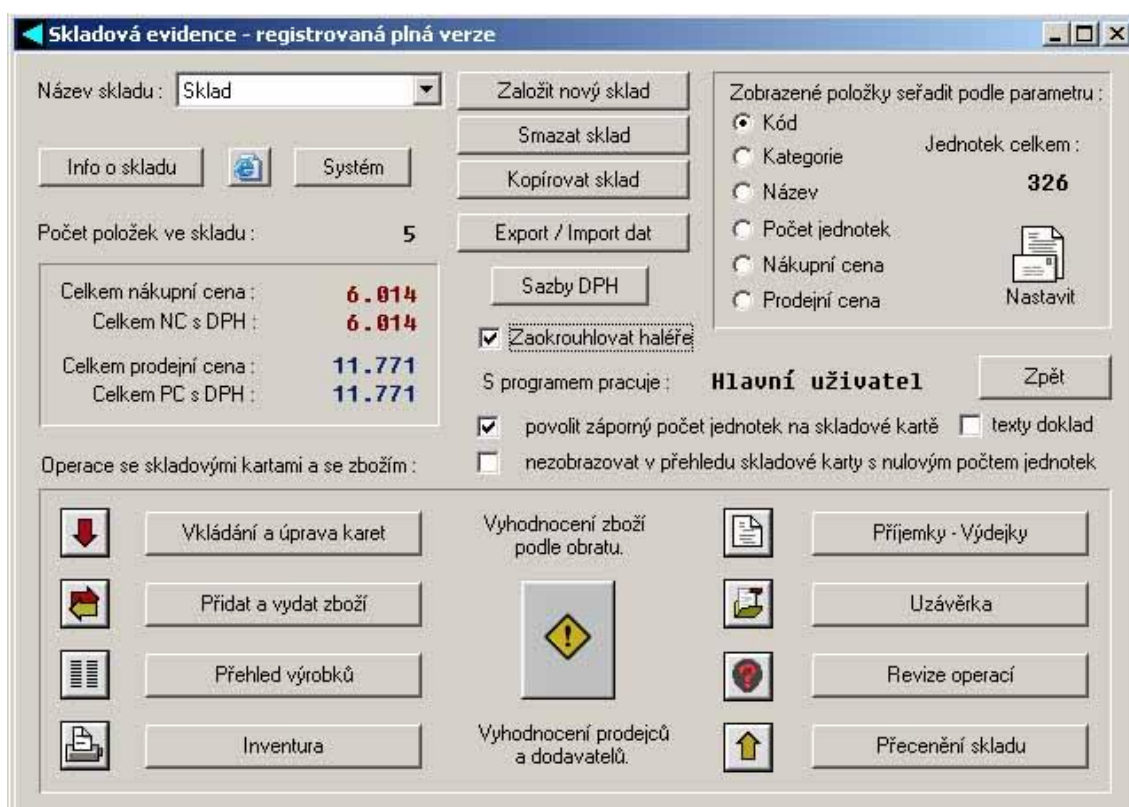
3.4.1 TREL

Tato skladová evidence nabízí širokou podporu všech potřebných funkcí, které jsou ve skladech denně prováděny. Funkce jako neomezený počet skladů, příjem a výdej zboží, slevy, generování ceníků, inventury, zobrazení minimálních zásob, plátcí a neplátcí

DPH, export do různých forem (e-mail, disketa, internet). Velkou výhodou tohoto systému je cena, která startuje na 3.990,- Kč za instalaci. Dále pak pravidelná aktualizace, o níž se starají programátoři. Ti dokážou přizpůsobit software i na netypické konfigurace.

Při spuštění se zobrazí základní obrazovka, kde nalezneme základní tlačítka a nejužívanější funkce pro práci ve skladu. Okno pro vkládání karet je přehledné, lze přiřazovat čárové kódy, množstevní slevy, poznámky (dodací lhůty, kvalita, upozornění). Velice zajímavou funkcí je „Kritické zásoby“: umožňuje sledovat, zda zboží nekleslo pod stanovenou mez. Ta se jednoduše vloží jako parametr skladové karty do kolonky "Minimum". Při kliknutí se zobrazí přehled všech výrobků daného skladu, kde je počet nižší než nastavené minimum.

Obr. 2 TRELl úvodní obrazovka



Zdroj: Ekonomický systém TRELl. *Sklady*. [online]. 2012 [cit. 2012-1-25].

Dostupné z: <http://trell.hyperlink.cz/sklady.htm>

Produkt je nabízen v demoverzi, která je bezplatná. Je však funkcionálně omezena (především počet možných položek v databázi). Stáhnout lze bez registrace. Nejčastěji prodávaná licence je pro 3 PC: 1PC pokladna, 1PC sklad a kontrolní pracoviště či notebook majitele, která stojí 4.990,- Kč. Doplnkové moduly jako je dotyková obrazovka, e-shop, který generuje webové stránky na základě vytvořené databáze, či systém pro evidování elektronických plateb je možné zakoupit v cenách od 1.000,- Kč.

3.4.2 TROP

Tento odbytový software je koncipován především jako pomůcka při fakturaci, dobírkách a tvoření předplatných. Hlavními soubory, které jsou nutné pro plynulý běh programu: katalog zboží, adresář odběratelů a objednávky.

Výhodami jsou různé možnosti expedice jako hotově, dobírkou, platbou z bankovního konta a fakturou. Účetní doklady lze kdykoli vytisknout a to i zpětně, taktéž se automaticky číslují. Ochrana proti zneužití systému je pomocí hesla, navíc jsou spolu se změnami dat zapisovány i informace o osobě, která je prováděla.

Obr. 3 TROP okno opravy objednávky

The screenshot shows a window titled "Oprava objednávky č.71291 od odběratele". The interface includes several input fields and buttons. At the top, there are fields for "Datum:" (17.03.2008), "Skupina:" (9966), "Rabat:" (0.00 %), "Splatnost:" (0 dní), and "Odběratel:" (1042623). Below this, there are fields for "Datum uskut.zdan.pln.:" (17.03.2008), "Stará objednávka", "Firma:", "Cena obj.zboží:" (303.00 Kč), "Status:" (0 objednáno), "DPH" (checked), "Doklady", "Jméno:" (Klára Urushadze), "Dlužné doklady", "Firma:" (Nad Školou 218, 25073 Radonice), "Platby vyrovnány", "Nový odběratel", "Faktura" (1042623), "Firma:" (Klára Urushadze), "Jméno:" (Klára Urushadze), "Firma:" (Nad Školou 218, 25073 Radonice). On the left side, there are buttons for "Platba" (H Hotově), "Sklad" (1 Hlavní sklad), "Zpracování" (21 vlastní odvoz), "Zásilka" (balík-Česká pošta), "Azyzo", "Výběr řady", "Přidání položky", "Přidat celý sklad", "Objednání", and "Poznámka". At the bottom, there are buttons for "Opravu", "Zrušit", and "Uložit".

Zdroj: AbaComp s.r.o. *Odbytový software TROP*. [online]. 2010 [cit. 2012-1-25].

Dostupné z: <http://www.abacom.cz/odbytovy-skladovy-software-trop/>

Obr. 4 TROP katalog zboží

The screenshot displays the 'Katalog zboží - prohlížení záznamu' window. It contains the following data:

- Artikl: 001518, Kód: , Kód2: D15, Poznámka:
- Název: MERKUR 010 Formule
- Název2: 001518.D15, Heslo: 001518.D15
- Místo: od 8 let
- Edice: 724000, Skupina: 72400, Třída: 080300
- Anotace: 10000110, Váha: 0.7000 kg, Balení: 7, MJ: ks
- Na všech skladech je: Množství: 0 ks, Objednáno: 0 ks, Blokováno: 0 ks
- Cena skladová: 199.49 Kč, Cena 2: 324.00 Kč (s DPH), Cena 4: 0.00 Kč (bez DPH)
- Cena 1: 324.00 Kč (s DPH), Cena 3: 0.00 Kč (bez DPH), Cena 5: 0.00 Kč (bez DPH)
- Druh zboží: Aktivita: Ano Ne, První příjem: plánovaný: 20.01.2006, Sazba DPH: Svybozeno, Výška: 2.550 m
- Merkur Toys, skutečný: , Sazba DPH: 19.00 %, Snížená, Délka: 0.500 m
- Datum posledního pohybu: 13.12.2007, Zisk: , Základní, Šířka: 1.800 m
- katalogový: 0.00 %, kalkulovaný: 800 Kč, Hloubka: 0.000 m
- dealerský: 1.20 %, skutečný: 9900 Kč, Tloušťka: 0.000 m
- Obrázek: d2_me001518.jpg
- Zvuk: Stavebnice obsahuje 223 dílků.
- Vídeo:
- Htm text: Kovová stavebnice MERKUR - Formule.

Buttons at the bottom: Na začátek, Na konec, Další, Předchozí, Ukončit.

Zdroj: AbaComp s.r.o. *Odbytový software TROP*. [online]. 2010 [cit. 2012-1-25].

Dostupné z: <http://www.abacomp.cz/odbytovy-skladovy-software-trop/>

Zajímavým modulem je KLUB, sloužící pro vyhodnocování speciálních agend. Především pak evidencí stálých „klubových“ zákazníků, dále rozšířenými datovými strukturami pro evidování smluvních partnerů zajišťujících prodeje zboží a výpočty a evidování jejich příjmů a srážkové daně z příjmu. Neposlední součástí je i katalog distribuovaných artiklů za speciální ceny.

Cena licencí je u všech modulů konstantní, pokud však chceme při podnikání využívat více modulů současně, ceny se snižují. Dále jsou uvedeny ceny pro upgrade, vytvoření e-shopu a jeho provozu či odlehčené verze LITE, která stojí 2.990,- Kč.

Tab. 1 Ceník modulů TROP

Verze	modul Faktury	modul Dobírky	modul Předplatné	modul Klub	současně 2 moduly	současně 3 moduly	současně 4 moduly
SUMÁŘ	2.990	2.990	2.990	2.990	4.990	6.990	8.990
PLNÁ	11.990	11.990	11.990	11.990	19.990	26.990	32.990

Zdroj: AbaComp s.r.o. *Odbytový software TROP*. [online]. 2010 [cit. 2012-1-25].

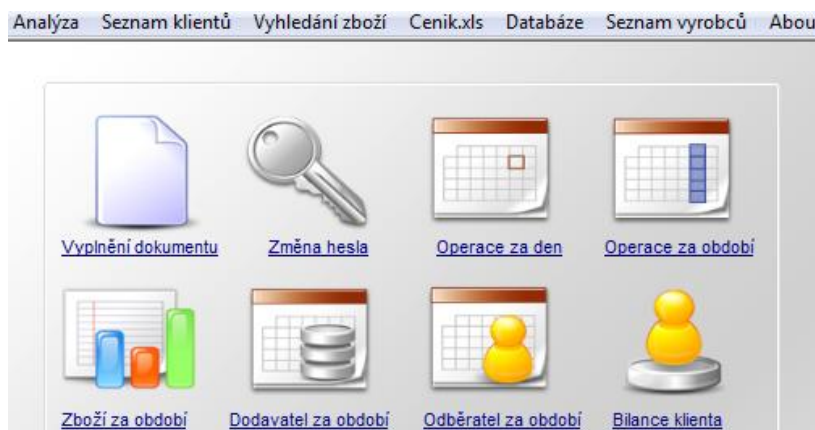
Dostupné z: <http://www.abacomp.cz/odbytovy-skladovy-software-trop/cenik-licenci-trop/>

3.4.3 WareHouse EU

Tento program byl vytvořen pro účely společnosti Greg Computers s.r.o., později byl rozšířen pro veřejnost a to především malé a střední podniky. Umí pracovat jak na lokálním počítači, tak i přes síť. Taktéž podporuje čtení čárkových kódů a ukládání jejich sériových čísel.

Výhodou je uživatelsky jednoduchá obsluha, u které není potřeba vysoké kvalifikace, a multijazyčná verze (čeština, angličtina a ruština).

Obr. 5 Část menu WareHouse EU



Zdroj: Trade-Soft.cz. *Warehouse EU :: Ukázky obrazovek*. [online]. 2012 [cit. 2012-1-25]. Dostupné z: <http://www.trade-soft.cz/7-2-warehouse-ukazky-obrazovek-sklad-evidence-hospodarstvi-program.aspx>

Dostatek funkcí jako propracovaná karta pro zadávání klientů, zboží (kde lze přidávat fotky, obrázky, podrobný popis produktu, marže, váhu či objem), generování stavu skladů do souborů typu Excel, tisk faktur s podporou vložení vlastního loga a spousta dalších se jeví jako velice kvalitní.

Software umožňuje i nastavování práv uživatelů, kteří s ním pracují a lze tak zamezit nevhodným úpravám, či zásahům do databáze. Pravidelné aktualizace rozšiřují funkcionalitu a podporují práci ve skladech. Zkušební verze je poskytována zdarma, je však omezena na dobu 30 pracovních dnů. Cena základní verze, která není omezena je 4.990,- Kč. Dražší verze jsou rozšířeny o programy Fakturace či obsahuje internetový obchod.

4. Vlastní návrhy řešení, přínos návrhů řešení

Díky analýzy provedené v podniku, popisu základních nedostatků a shrnutí poznatků lze přikročit k samotnému návrhu databáze. Samotný návrh bude rozdělen na logické kapitoly, které na sebe navazují.

4.1 Zachycení požadavků a procesů

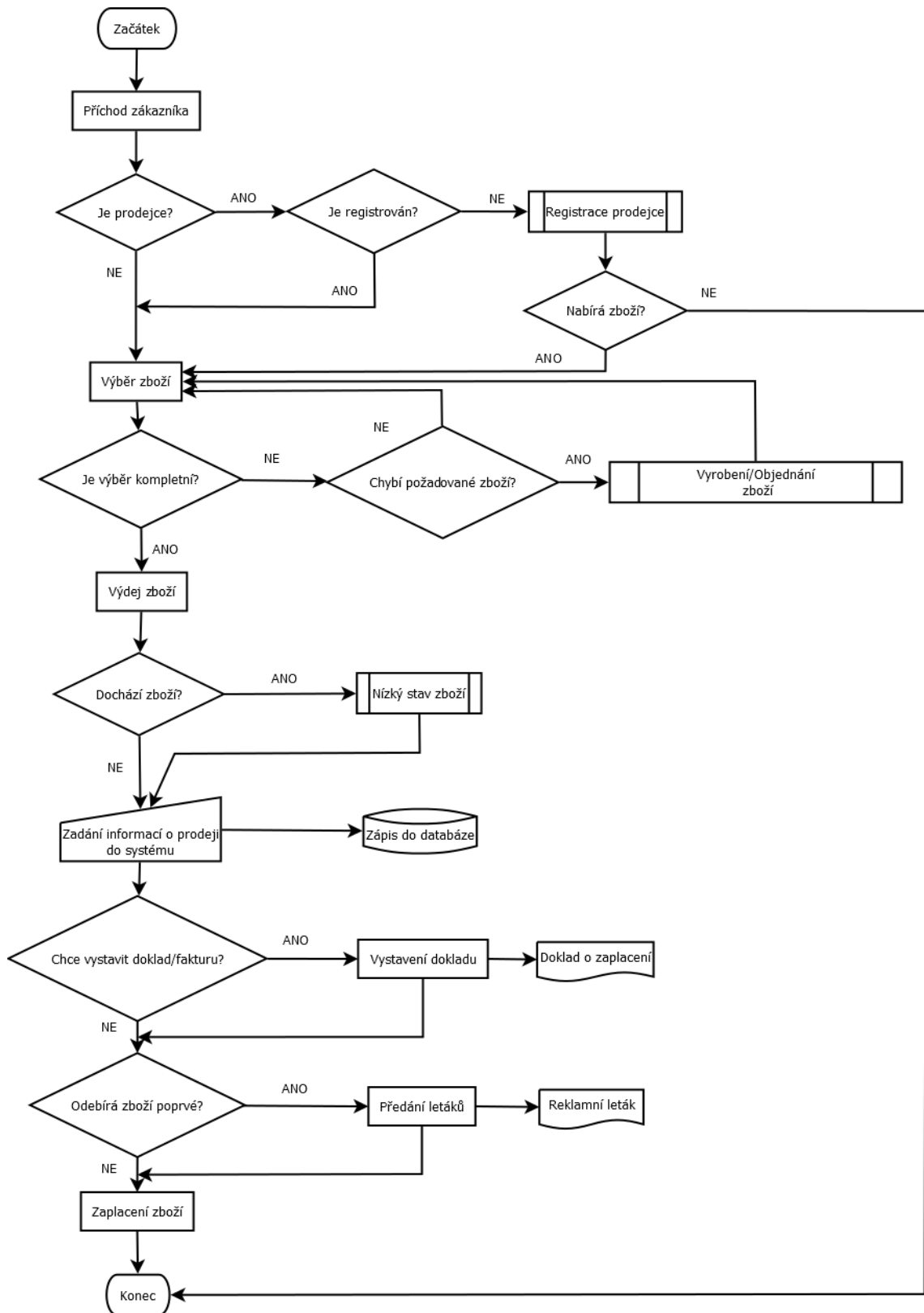
V této kapitole popíši a zachytím pomocí vývojových diagramů základní činnosti. Nastíním základní požadavky a procesy kladené na databázi. Mezi prioritní části jsem zařadil Prodej zboží, subproces Registrování prodejce a Údržba automobilu.

4.1.1 Prodej zboží

V procesu prodeje zboží nacházejícího se ve skladě je zachyceno několik činností, které se týkají zákazníka, od samotného příchodu a rozhodnutí o jaký typ člověka se jedná, přes výběr zboží až po zaplacení.

V podmínkách je ošetřeno několik případů, které mohou nastat. Zda jde o prodejce, a je-li registrován, či rozhodnutí o vydání dokladu či faktury. Obsahuje i několik podprocesů, které ošetřují stavy, jako je samotná registrace prodejce, hlášení o nízkém stavu zboží a objednání chybějícího zboží.

Obr. 6 Vývojový diagram – Prodej zboží



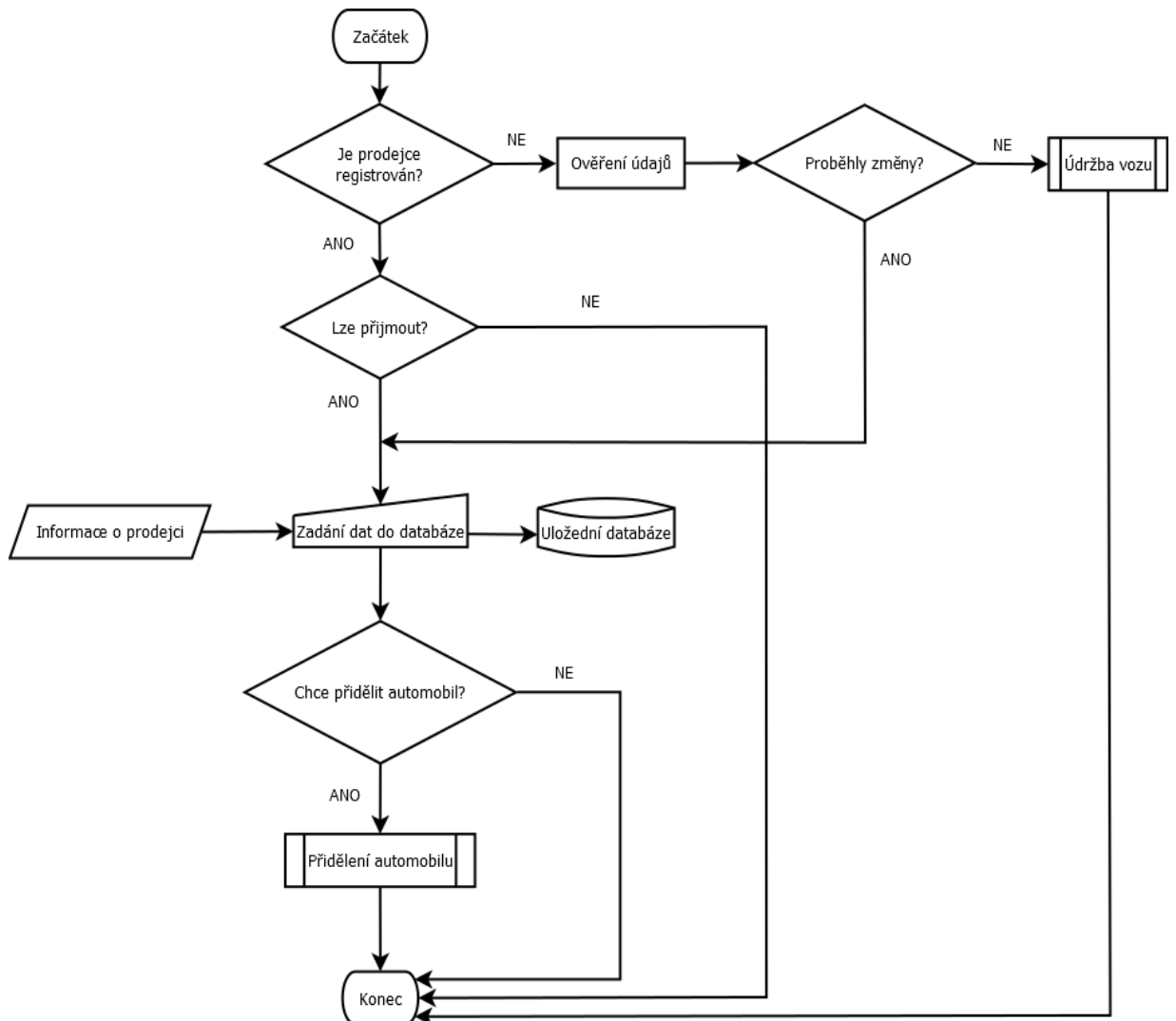
Zdroj: Vlastní zpracování

4.1.2 Registrace prodejce

Subproces registrace prodejce obsahuje několik významných podmínek, podle kterých se podnikatel rozhoduje, zda přijmout dalšího prodejce. V první řadě se ověřuje, jestli nedošlo ke změnám (adresa, lokalita rozvozu atd.) – provedeme aktualizaci databáze, pokud se jedná o zcela nového prodejce, který by chtěl nastoupit, ptáme se na přidělení firemního automobilu.

Pokud vyhovuje všem podmínkám, může nastoupit a začít distribuovat zboží. Výběr lokalit prodeje záleží na něm.

Obr. 7 Vývojový diagram – Registrace prodejce



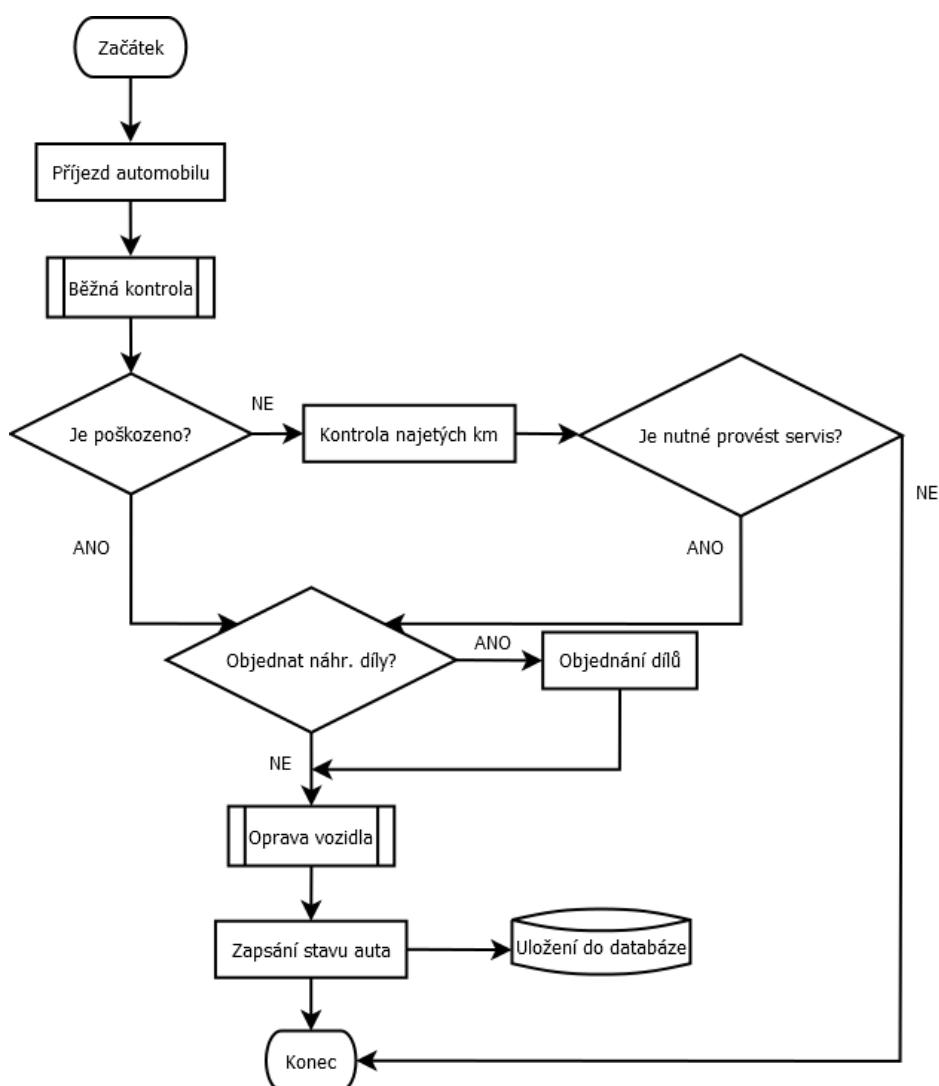
Zdroj: Vlastní zpracování

4.1.3 Údržba vozu

Při příjezdu prodejce na sklad ověřujeme povrchovou a běžnou kontrolu automobilu, pokud je vůz v pořádku, přečte se počet najetých kilometrů a porovná se s informací z databáze, od tohoto se odvíjí následný servis (výměna oleje, rozvodů, filtrů).

Pokud dochází k opravám či výměnám, jsou nutné náhradní díly. Ty jsou buď naskladněny nebo se objednávají. Po proběhnutí veškerých procesů se opět data zapíše do databáze.

Obr. 8 Vývojový diagram – Údržba vozu



Zdroj: Vlastní zpracování

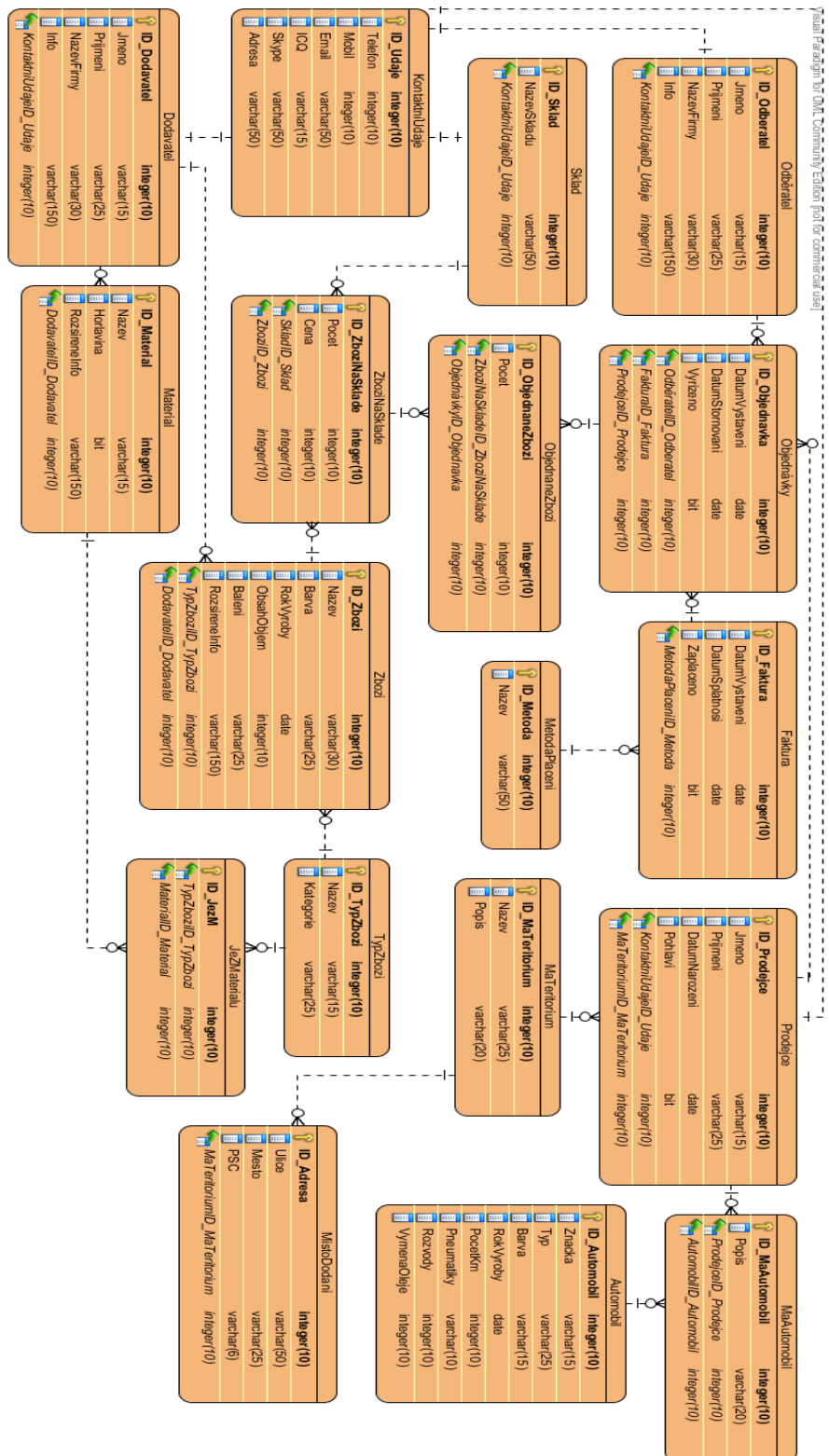
4.2 Konceptuální model

Konceptuálním modelem myslíme zachycení entit databáze, vzájemných vztahů a atributů, které obsahují. Z prvotního schématu provedeme transformaci na plnohodnotné databázové schéma. Dále musíme zajistit vhodnou strukturu dat a správně specifikovat integritní omezení.

4.3 Logický návrh

Týká se vytvoření tabulek, normalizace a ověření integrity a transakcí. Ve vytvořeném schématu vidíme jednotlivé relace, u každé z nich je zvýrazněn primární a cizí klíč, spolu se všemi atributy.

Obr. 9 Logický návrh databáze



Zdroj: Vlastní zpracování

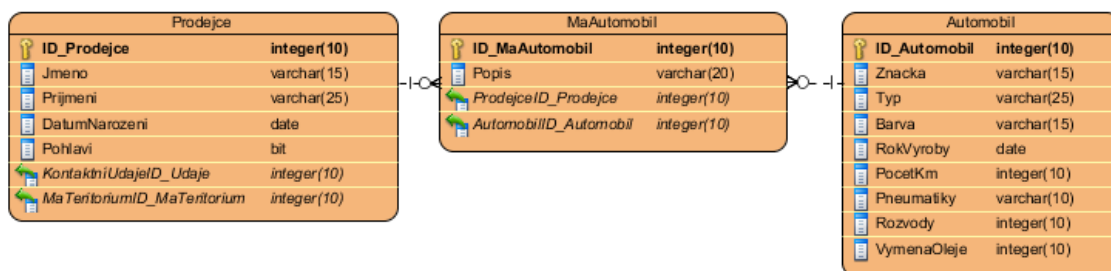
4.3.1 Prodejce, automobil, teritorium

Tabulka prodejce je primárně určena pomocí klíče ID_Prodejce, dále se v ní nacházejí základní údaje – jméno, příjmení, datum narození a pohlaví (řešeno pomocí 0 a 1, kde 0 je muž a 1 je žena).

Tabulka automobil obsahuje primární klíč ID_Automobil, veškeré základní údaje o vozidle. Název, typ, barva, rok výroby a typ pneumatik (zimní, letní). Hlavními atributy této tabulky jsou počet najetých km, dále kilometry, při kterých je nutné vyměnit olej a rozvody. Tyto dva atributy mohou být automaticky dopočteny při zápisu o výměně do systému na základě aktuálního stavu tachometru.

Tabulka MaAutomobil určená klíčem ID_MaAutomobil, obsahuje především odkazy na dvě předešlé tabulky – slouží jako cizí klíče a spojují tyto tabulky dohromady. Pomocí těchto 3 tabulek lze jednoduše zjistit, kteří prodejci právě využívají firemních vozů, jaký vůz mají a nakonec i detailní informace o každém z nich.

Obr. 10 Vazba prodejce s automobilem

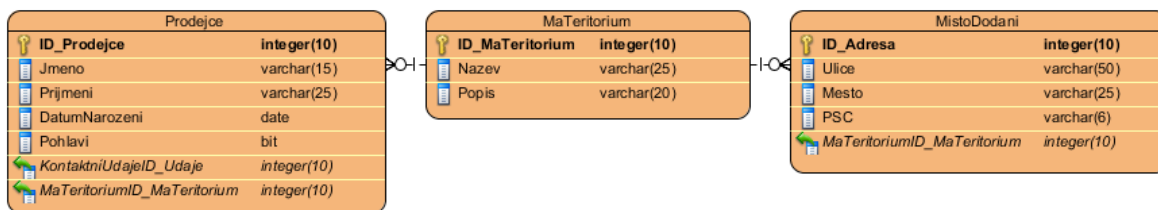


Zdroj: Vlastní zpracování

Tabulka MaTeritorium, kde je primárním klíčem atribut ID_MaTeritorim, popisuje teritorium, které prodejce zásobuje zbožím. Základním údajem je název (např. Olomouc a okolí, Ostrava, apod.) a popis (stupnice – výborné, dobré, slabé tržby a jiné).

Na tabulku MistoDodani, určenou klíčem ID_Adresa, vážeme tabulku MaTeritorium – pomocí cizího klíče. Tabulka slouží především k uchování adres, které se nacházejí v daném teritoriu, kam mohou prodejci distribuovat výrobky. Není nutné uchovávat informace o tom, kdo se na dané adrese nachází. Slouží pouze k výpisu, který můžeme předat novému prodejci, které dané území nezná – což vede k zjednodušení orientace.

Obr. 11 Vazba prodejce k teritoriu



Zdroj: Vlastní zpracování

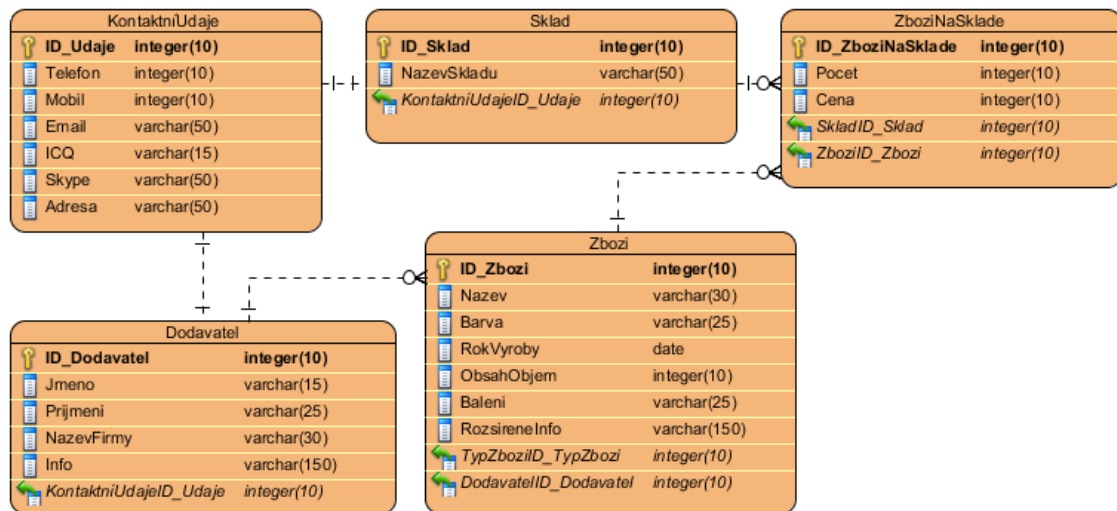
4.3.2 Zboží a jeho detaily, sklad a umístění zboží

V tabulce Sklad je primárním klíčem ID_Sklad, který v databázi jednotlivé sklady exaktně rozlišuje. Dále zde nalezneme atribut název skladu a cizí klíč do tabulky KontaktniUdaje, v níž uchováváme informace o telefonních číslech, emailech, adrese a dalších podrobnostech.

Tabulka Zboží je základním prvkem každé databáze obsahující výrobky, zboží či materiál určený k dalšímu prodeji. Primárním klíčem je ID_Zboží. Informace o produktu samotném jsou rozepsány do několika atributů, jimiž jsou např. název, barva, obsah/objem, rok výroby a jiné. Cizími klíči jsou ID_TypZbozi, pomocí kterého můžeme lehce roztřídit veškeré zboží do kategorií (aviváže, prací prášky, WC prostředky atd.), a ID_Dodavatel, kde tento klíč určuje zboží, které firma nevyrábí a nakupuje je od externích dodavatelů.

Tabulka ZboziNaSklade nám spojuje tabulky Zboží a Sklad, kde jasně určíme veškeré zboží v každém skladu, který tato databáze bude spravovat. Primárním klíčem je ID_ZboziNaSklade. Nachází se zde i atributy cena, za kterou budeme zboží prodávat a taktéž počet kusů k prodeji. V této tabulce budeme hlídat stav (počet) zboží a klesne-li na předem stanovenou úroveň, dojde k upozornění.

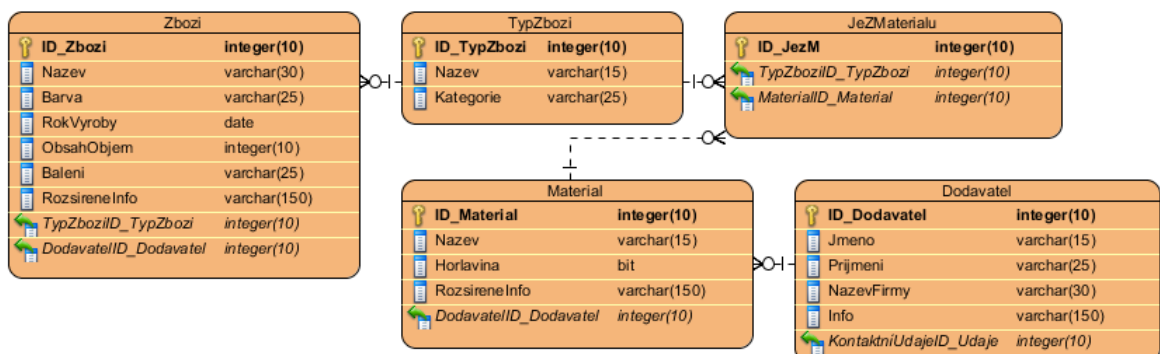
Obr. 12 Vazba skladu a zboží



Zdroj: Vlastní zpracování

Tabulka TypZbozi slouží jako seznam kategorií s popisem, které přiřazujeme každému zboží nacházejícímu se ve skladech. Každý typ je složen z několika druhů materiálů, pro tyto účely je v databázi vytvořena tabulka JeZMaterialu, na kterou vážeme pomocí cizího klíče daný typ zboží a tabulku Materiál, kde jsou uloženy materiály s popisem a záznamem o tom, jestli se jedná o hořlavinu – tato položka je zde především kvůli bezpečnosti uchovávání. Materiál pro výrobu je dodáván taktéž od různých dodavatelů, proto opět vidíme vazbu do této tabulky. Všechny tabulky jsou určeny podle primárních klíčů, které jsou vždy složeny s předpony ID_ a jejich názvu.

Obr. 13 Vazba zboží a jeho specifikace



Zdroj: Vlastní zpracování

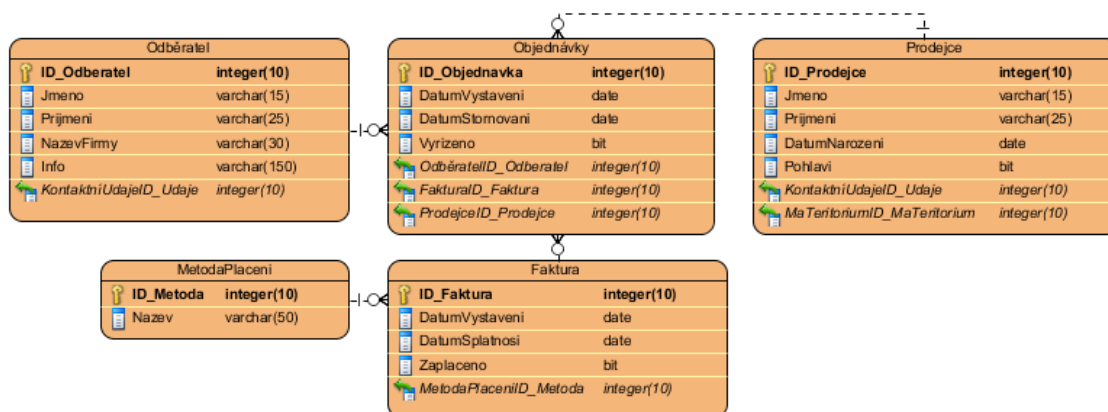
4.3.3 Objednávka, faktura a metody placení

Tabulka Objednávka, která je jednou z nejdůležitějších tabulek v celé databázi, je určena klíčem ID_Objednávka. Do této tabulky se zapisují objednávky na zboží, které buď přijdou od prodejců, nebo od jiných odběratelů, kteří o něj mají zájem – opět vazby pomocí cizích klíčů. Ukládá se datum vystavení objednávky, dále pak její storno. Není nutné držet v evidenci objednávky, které se neuskuteční. Pokud vyzvednutí zboží neproběhne do 7 dnů od vystavení automaticky je objednávka stornována. Posledním atributem je Vyrizeno (0 – nevyřízeno, 1 - vyřízeno).

U tabulky Faktura je primárním klíčem ID_Faktura. U faktur je klasickými informacemi datum vystavení a datum splatnosti, které se zde nachází. Dále atribut Zaplaceno, který nám říká, jestli byl prodej zboží zaplacen.

Na Fakturu se váže i metoda placení, kde si můžeme vybrat ze seznamu, o jaký typ se jedná – hotově, bankovním převodem atd. Toto je zastoupeno poslední tabulkou MetodaPlaceni, u níž je klíčem ID_Metoda.

Obr. 14 Vazba objednávek a faktur



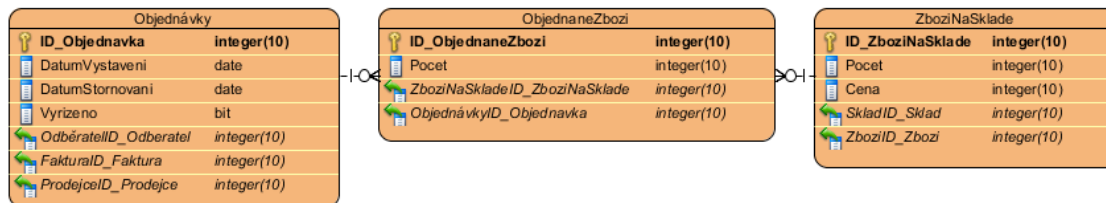
Zdroj: Vlastní zpracování

4.3.4 Objednání zboží ze skladu

Pokud vystavíme nějakou objednávku, musíme do ní zapsat i o jaké zboží se jedná, z jakého skladu je a jaký je počet – jeden z atributů. To provedeme pomocí tabulky ObjednaneZbozi, kde odkazujeme na patřičnou objednávku a dále pak na zboží na

skladě – tabulky Objednavka a ZboziNaSklade. Ty jsme popsali výše, proto je zde není nutné znovu vypisovat.

Obr. 15 Vazba objednávky a objednaného zboží



Zdroj: Vlastní zpracování

4.4 Fyzický návrh

Vytvořil jsem několik procedur, které usnadní práci v databázi. U každé z nich je přiložen zdrojový kód a popis, jak pracují. Některé události probíhající při operacích jako INSERT, DELETE nebo UPDATE je třeba hlídat a zajistit neustálou integritu, proto jsem se rozhodl uvést několik triggerů, které zabezpečují správnost dat.

Zbytek kódů napsaných v jazyce SQL je přiloženo v přílohách – především creaty (vytvoření) všech tabulek databáze, inserty a několik jednoduchých selectů, pro zobrazení dat z tabulek.

4.4.1 Procedura registrace prodejce

Zde je uvedena celá řada vstupních parametrů, které se ukládají do dvou tabulek – Prodejce a Kontaktní údaje. První podmínkou k uložení prodejce je, že se ověřuje přiřazené teritorium, pokud neexistuje, uložení neproběhne a vypíše se kritické hlášení. Poté proběhnou další dvě podmínky. Jedna z nich ověřuje existenci stejných kontaktních údajů, které už mohou být v databázi uloženy, druhá zase existenci prodejce se stejnými iniciály. Pokud je vše v pořádku, proběhne úspěšná registrace.

```

go
create procedure registrace_prodejce
@jmeno varchar(15),
@prijmeni varchar(25),
@adresa varchar(50) = NULL,
@telefon int = NULL,
@mobil int = NULL,
@skype varchar(50) = NULL,
@icq varchar(15) = NULL,

```

```

@email varchar(50) = NULL,
@datum_narozeni date = NULL,
@pohlavi bit = NULL,
@teritorium varchar(35)
as
begin
    declare @id_teritorium int
    select @id_teritorium = ID_MaTeritorium
    from MaTeritorium where Nazev = @teritorium
    if (@id_teritorium IS NULL)
    begin
        print 'Dané teritorium neexistuje - prosím nejdříve jej zadejte'
    end
    else
    begin
        declare @id_udaje int
        select @id_udaje = ID_Udaje
        from KontaktniUdaje where Telefon = @telefon AND Mobil = @mobil
and Email = @email

        declare @id_prodejce int
        select @id_prodejce = ID_Prodejce
        from Prodejce where Jmeno = @jmeno and Prijmeni = @prijmeni and
DatumNarozeni = @datum_narozeni

        if (@id_udaje IS NULL) and (@id_prodejce IS NULL)
        begin
            insert into KontaktniUdaje (Telefon, Mobil, Email, ICQ, Skype,
Adresa)
            values (@telefon, @mobil, @email, @icq, @skype, @adresa)
            set @id_udaje = IDENT_CURRENT('KontaktniUdaje')

            insert into Prodejce (Jmeno, Prijmeni, DatumNarozeni, Pohlavi,
ID_MaTeritorium, ID_Udaje)
            values (@jmeno, @prijmeni, @datum_narozeni, @pohlavi,
@id_teritorium, @id_udaje)
        end
        else
        begin
            print 'Překontrolujte správnost dat - existence podobných
záznamů!'
        end
    end
end
go

begin tran RegistrujProdejce
execute registrace_prodejce 'Pavel', 'Paska', 'Vítkov, Selská
942', '723634527', '556300153', 'Pajosh080', '247-160-421',
'pavel.paska@gmail.com', '1990-07-10', '1', 'Olomoucko'

select * from Prodejce
select * from KontaktniUdaje

ROLLBACK TRAN RegistrujProdejce

```

4.4.2 Procedura přidělení automobilu

V proceduře přidělení automobilu se ověřuje, zda jméno a příjmení souhlasí s údaji některého prodejce uloženého v databázi. Druhou podmínkou je, že značka a samotný vůz existují. Poté se do tabulky MaAutomobil zapisují ID nalezených záznamů spolu s krátkým popisem, který upřesňuje podmínky přidělení.

```
go
create procedure prideleni_automobilu
@jmeno varchar(15),
@prijmeni varchar(25),
@znacka varchar(15),
@popis varchar(20) = NULL,
as
begin
    declare @id_prodejce int
    select @id_prodejce = ID_Prodejce
    from Prodejce where Jmeno = @jmeno and Prijmeni = @prijmeni

    declare @id_automobil int
    select @id_automobil = ID_Automobil
    from Automobil where Znacka = @znacka

    if (@id_prodejce IS NULL)
    begin
        print 'Daný prodejce neexistuje - prosím nejdříve jej
registrujte'
    end
    else if (@id_automobil IS NULL)
    begin
        print 'Daný automobil neexistuje - prosím nejdříve jej
zadejte'
    end
    else
    begin
        insert into MaAutomobil (Popis, ID_Prodejce, ID_Automobil)
        values (@popis, @id_prodejce, @id_automobil)
    end
end
go

begin tran PridelAutomobil
execute prideleni_automobilu 'Pavel', 'Peska', '1A7 45-52', 'Půjčeno
na 2 měsíce'

select * from MaAutomobil

ROLLBACK TRAN PridelAutomobil
```

4.4.3 Procedura nová karta zboží

Tato procedura je velice podobná té předešlé, smyslem je vytvořit kartu zboží, přidělit ji do kategorie, a pokud je zboží dodáváno externím dodavatelem, tak uložit i toto. Opět vidíme podmínky ověřující existenci dané kategorie a dodavatele. Každou proceduru jsem vložil do transakce (jednoduchá práce s daty – potvrzení, odvolání transakce).

```
go
create procedure karta_zbozi
@nazev varchar(30),
@barva varchar(25),
@rok_vyroby date,
@obsah_objem int,
@baleni varchar(25),
@rozsirene_info varchar(150) = NULL,
@typ_zbozi varchar(15),
@dodavatel varchar(30) = NULL
as
begin
    declare @id_dodavatel int
    select @id_dodavatel = ID_Dodavatel
    from Dodavatel where NazevFirmy = @dodavatel

    declare @id_typ_zbozi int
    select @id_typ_zbozi = ID_TypZbozi
    from TypZbozi where Nazev = @typ_zbozi

    if (@id_dodavatel IS NULL)
    begin
        print 'Daný dodavatel neexistuje - prosím nejdříve jej
registrujte'
    end
    else if (@id_typ_zbozi IS NULL)
    begin
        print 'Daný typ zboží neexistuje - prosím nejdříve jej
zadejte'
    end
    else
    begin
        insert into Zbozi (Nazev, Barva, RokVyroby, ObsahObjem,
Baleni, RozsireneInfo, ID_TypZbozi, ID_Dodavatel)
        values (@nazev, @barva, @rok_vyroby, @obsah_objem, @baleni,
@rozsirene_info, @id_typ_zbozi, @id_dodavatel)
    end
end
go

begin tran VytvorZbozi
execute karta_zbozi 'Prací gel', 'Tmavě modrá', '2012-03-30', '5 l',
'Plastová nádoba', 'Síla na jakoukoli špínu', 'Prací prostředky',
'Ariel'

select * from Zbozi

ROLLBACK TRAN VytvorZbozi
```

4.4.4 Procedura údržba vozu

Pokud přijede automobil na pravidelnou kontrolu a provede se jistá údržba, je potřeba zaznamenat nová data. Je zde i obligátní podmínka, zdali je značka vozu uložena v databázi. Při zadávání typu údržby, rozlišujeme, jestli se upraví data o další výměně oleje či rozvodů.

```
go
create procedure udrzba_vozu
@znacka varchar(15),
@rok_vyroby date,
@stav_tachometru int,
@typ_uzrby varchar(15)
as
begin
    declare @id_automobil int
    select @id_automobil = ID_Automobil
    from Automobil where Znacka = @znacka and RokVyroby = @rok_vyroby

    if (@id_automobil IS NULL)
    begin
        print 'Daný automobil neexistuje - prosím nejdříve jej
registrujte'
    end
    else if (@typ_uzrby = 'Olej')
    begin
        declare @olej int
        set @olej = @stav_tachometru + 10000
        update Automobil set PocetKm = @stav_tachometru where
ID_Automobil = @id_automobil
        update Automobil set VymenaOleje = @olej where ID_Automobil =
@id_automobil
    end
    else
    begin
        declare @rozvody int
        set @rozvody = @stav_tachometru + 30000
        update Automobil set PocetKm = @stav_tachometru where
ID_Automobil = @id_automobil
        update Automobil set Rozvody = @rozvody where ID_Automobil =
@id_automobil
    end
end
go

begin tran ProvedUdrzbu
execute udrzba_vozu 'BRC-60-90', '2005-01-01', 143260, 'Olej'

select * from Automobil

ROLLBACK TRAN ProvedUdrzbu
```

4.4.5 Trigger nízký stav zboží

Tento trigger je vytvořen nad tabulkou ZboziNaSklade. Pokud proběhne vyskladnění zboží ze skladu a počet tohoto zboží klesne pod požadovanou hodnotu, vytvoří se doporučení na výrobu.

```
create trigger nizky_stav_zbozi on ZboziNaSklade
for update
as
begin
  if((SELECT Pocet FROM INSERTED) < 20)
  begin
    print 'Nízký stav zboží - požadavek na vyrobení!'
  end
  else
  begin
    print 'Dostatek zboží na skladu'
  end
end
go

select * from ZboziNaSklade

begin tran Kontrola
update ZboziNaSklade set Pocet = 15 where ID_Zbozi = 7

ROLLBACK TRAN Kontrola

DROP TRIGGER nizky_stav_zbozi
```

4.4.6 Triggery pro mazání dat

Oba triggery zajišťují konzistenci dat, slouží k mazání v tabulkách Automobil, Prodejce, KontaktniUdaje a MaAutomobil. Podobným způsobem můžeme vytvořit triggery i pro jiné tabulky v databázi.

```
create trigger mazani_prodejce on Prodejce
for delete
as
  delete from MaAutomobil where ID_Prodejce = (select ID_Prodejce
from deleted)
  delete from KontaktniUdaje where ID_Udaje = (select ID_Udaje
from deleted)
go

create trigger mazani_automobilu on Automobil
for delete
as
  delete from MaAutomobil where ID_Automobil = (select
ID_Automobil from deleted)
go
```

4.4.7 Přínosy databáze

V teoretické rovině má tato relační databáze spoustu přínosů. Identifikujeme ty nejužitečnější a jasně viditelné.

- Zrychlení práce při skladových operacích (naskladnění, inventura)
- Hlídaní nízkého stavu zásob
- Vytvoření a editace zboží (název, specifikace, a jiné)
- Přehledný soupis dodavatelů, prodejců a zákazníků s tabulkou kontaktních údajů
- Vytvoření dodavatelské sítě – adresy
- Informace o technickém stavu firemních automobilů
- Objednávky, fakturace zboží
- Souhrné informace o celé firmě

Jak již samotný název práce – návrh databáze neřeší aplikační část, jedná se pouze o čisté vytvoření tabulek, vložení dat a ověření funkčnosti, tudíž přímo nejde změřit hmatatelné výsledky a přínosy pro firmu.

Vhodnou by mohla být webová aplikace, která poskytuje komfortní přístup k databázi odkudkoli.

Pro mě celá tato práce měla spoustu pozitivních zkušeností a zážitků. Ve firmě jsem si mohl odzkoušet všechny práce, které jsou potřeba k úspěšnému prodeji a spokojenosti zákazníků. Každodenní kontakt s lidmi, výroba zboží, doprava až do domu.

Práce s databázemi jsou pro mě lákavou možností jak rozvíjet dál své schopnosti. Samotný návrh od základů – co chceme ukládat, až po tvoření procedur, které jednoduše ovládají práci s nimi.

5. Závěr

Cílem této práce bylo vytvoření databáze pro firmu Iveta Šťastníková – výroba a distribuce drogerie. Tento návrh měl být šitý na míru potřebám skladu a důležitým procesům, které jsou pro fungování firmy zásadní. Evidence zboží na skladě, jejich prodej, doplňování nebo klasické objednávky jsou na denním pořádku. Zboží dále distribuují prodejci, kteří vlastní firemní automobily, které je nutné podrobovat klasickým servisním prohlídkám podle stanovených kritérií. I toto se v návrhu objevilo. Ostatní dílčí procesy či požadavky jsou zahrnuty v práci.

Analyzoval jsem současný stav firmy, a shrnul situaci do několika bodů, ze kterých se vycházelo při návrhu. Samotným řešením a vytvořením výsledné databáze tato bakalářská práce splnila veškeré požadavky firmy a povede ke zrychlení a zefektivnění procesů.

Nicméně proto, aby byla úspěšně zavedená do užívání, musíme vytvořit uživatelsky příjemnou a intuitivní aplikaci (nejlépe webovou), nelze efektivně pracovat bez uživatelského prostředí, pouze s klasickým databázovým serverem a příkazy v SQL zadávanými ručně.

6. Seznam literatury

- [1] CONNOLY, T., BEGG, C., HOLOWCZAK, R. *Mistrovství: Databáze. Profesionální průvodce tvorbou efektivních databází*. 1. vydání. Brno: Computer Press, 2009. s. 573. ISBN: 978-80-251-2328-7.
- [2] DOSTÁL, P., KŘÍŽ, J. *Databázové systémy*. 1. vydání. Brno: Cerm, 2006. s. 111. ISBN: 80-214-3064-8.
- [3] HOTEK, M. *Microsoft SQL Server 2008*. 1. vydání. Brno: Computer Press, 2009. s. 481. ISBN: 978-80-251-2466-6.
- [4] KOCH, M. *Datové a funkční modelování*. 1. vydání. Brno: Cerm, 2004. s. 108. ISBN: 978-80-214-4125-5.
- [5] KOCH, M. *Informační systémy a technologie – Část I*. 1. vydání. Brno: VUT, 2001. s. 103. ISBN: 80-214-1970-9.
- [6] LACKO, L. *SQL 1001 tipů a triků pro SQL*. 1. vydání. Brno: Computer Press, 2011. s. 407. ISBN: 978-80-251-3010-0.
- [7] MANAGED DEDICATED SERVERY. *Hierarchický databázový model* [online]. © 2009 [cit. 2011-12-13]. Dostupné z: <http://www.managed-dedicated-servery.net/hierarchicky-databazovy-model.html>
- [8] MANAGED DEDICATED SERVERY. *Síťový databázový model* [online]. © 2009 [cit. 2011-12-13]. Dostupné z: <http://www.managed-dedicated-servery.net/sitovy-databazovy-model.html>
- [9] OPPEL, A. *SQL bez předchozích znalostí*. 1. vydání. Brno: Computer Press, 2012. s. 233. ISBN: 978-80-251-1707-1.

[10] STEPHENS, R., PLEW, R., JONES, A. *Naučte se SQL za 28 dní*. 1. vydání. Brno: Computer Press, 2010. s. 715. ISBN: 978-80-251-2700-1.

Normalizace databáze

[11] WEB4COMPANY. *Databázové platformy* [online]. 2005-2011 [cit. 2011-12-13].

Dostupné z: <http://www.web4company.cz/technologie-databazove-platformy/>

7. Seznam obrázků a tabulek

7.1 Seznam obrázků

Obr. 1 Výrobní sortiment	25
Obr. 2 TRELLE úvodní obrazovka.....	29
Obr. 3 TROP okno opravy objednávky	30
Obr. 4 TROP katalog zboží.....	31
Obr. 5 Část menu Warehouse EU	32
Obr. 6 Vývojový diagram – Prodej zboží	35
Obr. 7 Vývojový diagram – Registrace prodejce	36
Obr. 8 Vývojový diagram – Údržba vozu.....	37
Obr. 9 Logický návrh databáze.....	39
Obr. 10 Vazba prodejce s automobilem	40
Obr. 11 Vazba prodejce k teritoriu.....	41
Obr. 12 Vazba skladu a zboží.....	42
Obr. 13 Vazba zboží a jeho specifikace	42
Obr. 14 Vazba objednávek a faktur.....	43
Obr. 15 Vazba objednávky a objednaného zboží.....	44

7.2 Seznam tabulek

Tab. 1 Ceník modulů TROP.....	32
-------------------------------	----

8. Seznam příloh

- Datový slovník
- Zdrojový kód

8.1 Datový slovník

Datový slovník přehledně zobrazuje jednotlivé tabulky databáze, jednotlivé položky, datové typy a jejich délky, spolu s omezeními jako je PK a FK či not null.

Tabulka	Položka	Typ	Delka	PK, FK, CHECK	Další omezení
Sklad	ID_Sklad	int	identity (1,1)	PK	
	NazevSkladu	varchar	50		not null
	ID_Udaje	int		FK	
MistoDodani	ID_Adresa	int	identity (1,1)	PK	
	Ulice	varchar	50		not null
	Město	varchar	25		not null
	PSC	varchar	6		not null
	ID_MaTeritorium	int		FK	not null
KontaktniUdaje	ID_Udaje	int		PK	
	Telefon	int			
	Mobil	int			
	Email	varchar	50		
	ICQ	varchar	15		
	Skype	varchar	50		
	Adresa	varchar	50		
Prodejce	ID_Prodejce	int	identity (1,1)	PK	
	Jmeno	varchar	15		not null
	Prijmeni	varchar	25		not null
	DatumNarozeni	date			not null
	Pohlavi	bool			
	ID_MaTeritorium	int		FK	
	ID_Udaje	int		FK	
MaAutomobil	ID_MaAutomobil	int	identity (1,1)	PK	
	Popis	varchar	20		
	ID_Prodejce	int		FK	not null
	ID_Automobil	int		FK	not null
Automobil	ID_Automobil	int	identity (1,1)	PK	
	Znacka	varchar	15		not null

	Typ	varchar	25		not null
	Barva	varchar	15		not null
	RokVýroby	date			not null
	PocetKm	int			not null
	Pneumatiky	varchar	10		not null
	Rozvody	int			not null
	VymenaOleje	int			not null
MaTeritorium	ID_MaTeritorium	int	identity (1,1)	PK	
	Název	varchar	35		not null
	Popis	varchar	20		
Dodavatel	ID_Dodavatel	int	identity (1,1)	PK	
	Jmeno	varchar	15		
	Prijmeni	varchar	25		
	NazevFirmy	varchar	30		not null
	Info	varchar	150		
	ID_Udaje	int		FK	
Odběratel	ID_Odběratel	int	identity (1,1)	PK	
	Jmeno	varchar	15		
	Prijmeni	varchar	25		
	NazevFirmy	varchar	30		not null
	Info	varchar	150		
	ID_Udaje	int		FK	
Zboží	ID_Zboží	int	identity (1,1)	PK	
	Nazev	varchar	30		not null
	ID_TypZbozi	int		FK	not null
	ID_Dodavatel	int		FK	
	Barva	varchar	25		not null
	RokVýroby	date			not null
	ObsahObjem	int			not null
	Balení	varchar	25		not null
	RozsireneInfo	varchar	150		
TypZboží	ID_TypZbozi	int	identity (1,1)	PK	
	Nazev	varchar	15		not null
	Kategorie	varchar	25		not null
JeZMaterialu	ID_JezM	int	identity (1,1)	PK	
	ID_TypZbozi	int		FK	not null
	ID_Material	int		FK	not null
Materiál	ID_Materiál	int	identity (1,1)	PK	
	Nazev	varchar	15		not null
	Horlavina	bool			not null
	RozsireneInfo	varchar	150		
	ID_Dodavatel	int		FK	not null

Faktura	ID_Faktura	int	identity (1,1)	PK	
	ID_Metoda	int		FK	
	DatumVystaveni	date			not null, akt. Datum
	DatumSplatnosti	date			akt. Datum + 14 dní
	Zaplaceno	bool			
MetodaPlaceni	ID_Metoda	int	identity (1,1)	PK	
	Nazev	varchar	50		not null
ZboziNaSklade	ID_ZboziNaSklade	int	identity (1,1)	PK	
	Pocet	int			not null
	Cena	int			not null
	ID_Zbozi	int		FK	
	ID_Sklad	int		FK	
Objednávka	ID_Objednavka	int	identity (1,1)	PK	
	ID_Odběratel	int		FK	
	ID_Prodejce	int		FK	
	ID_Faktura	int		FK	
	DatumVystaveni	date			not null, akt. Datum
	DatumStornovani	date			akt. Datum + 3 dny
	Vyrizeno	bool			
ObjednaneZbozi	ID_ObjednaneZbozi	int	identity (1,1)	PK	
	Pocet	int			not null
	ID_ZboziNaSklade	int		FK	
	ID_Objednavka	int		FK	

8.2 Zdrojový kód

Obsahuje kódy pro vytvoření všech tabulek databáze, dále pak procedury a triggerů, které s nimi operují a několik selectů, které poslouží k zobrazení dat.

```
CREATE TABLE MaTeritorium
(
    ID_MaTeritorium int identity (1,1) primary key,
    Nazev varchar(25) NOT NULL,
    Popis varchar(20),
);

CREATE TABLE MistoDodani
(
    ID_Adresa int identity (1,1) primary key,
    Ulice varchar(50) NOT NULL,
    Mesto varchar(25) NOT NULL,
    PSC varchar(6) NOT NULL,
    ID_Materitorium int foreign key references Materitorium
(ID_Materitorium)
);

CREATE TABLE KontaktniUdaje
(
    ID_Udaje int identity (1,1) primary key,
    Telefon int,
    Mobil int,
    Email varchar(50),
    ICQ varchar(15),
    Skype varchar(50),
    Adresa varchar(50),
);

CREATE TABLE Sklad
(
    ID_Sklad int identity (1,1) primary key,
    NazevSkladu varchar(50) NOT NULL,
    ID_Udaje int foreign key references KontaktniUdaje (ID_Udaje)
);

CREATE TABLE Prodejce
(
    ID_Prodejce int identity (1,1) primary key,
    Jmeno varchar(15) NOT NULL,
    Prijmeni varchar(25) NOT NULL,
    DatumNarozeni date NOT NULL,
    Pohlavi bool,
    ID_Udaje int foreign key references KontaktniUdaje (ID_Udaje),
    ID_MaTeritorium int foreign key references MaTeritorium
(ID_MaTeritorium)
);
```

```

CREATE TABLE Automobil
(
    ID_Automobil int identity (1,1) primary key,
    Znacka varchar(15) NOT NULL,
    Typ varchar(25) NOT NULL,
    Barva varchar(15) NOT NULL,
    RokVyroby date NOT NULL,
    PocetKm int NOT NULL,
    Pneumatiky varchar(10) NOT NULL,
    Rozvody int NOT NULL,
    VymenaOleje int NOT NULL,
);

CREATE TABLE Odberatel
(
    ID_Odberatel int identity (1,1) primary key,
    Jmeno varchar(15),
    Prijmeni varchar(25),
    NazevFirmy varchar(30) NOT NULL,
    Info varchar(150),
    ID_Udaje int foreign key references KontaktniUdaje (ID_Udaje)
);

CREATE TABLE Dodavatel
(
    ID_Dodavatel int identity (1,1) primary key,
    Jmeno varchar(15),
    Prijmeni varchar(25),
    NazevFirmy varchar(30) NOT NULL,
    Info varchar(150),
    ID_Udaje int foreign key references KontaktniUdaje (ID_Udaje)
);

CREATE TABLE MaAutomobil
(
    ID_MaAutomobil int identity (1,1) primary key,
    Popis varchar(20),
    ID_Prodejce int foreign key references Prodejce (ID_Prodejce),
    ID_Automobil int foreign key references Automobil
(ID_Automobil)
);

CREATE TABLE TypZbozi
(
    ID_TypZbozi int identity (1,1) primary key,
    Nazev varchar(15) NOT NULL,
    Kategorie varchar(25) NOT NULL,
);

CREATE TABLE Material
(
    ID_Material int identity (1,1) primary key,
    Nazev varchar(15) NOT NULL,
    Horlavina bool NOT NULL,
    RozsireneInfo varchar(150),
    ID_Dodavatel int foreign key references Dodavatel
(ID_Dodavatel)
);

```

```

CREATE TABLE JeZMaterialu
(
    ID_JezM int identity (1,1) primary key,
    ID_TypZbozi int foreign key references TypZbozi (ID_TypZbozi),
    ID_Material int foreign key references Material (ID_Material)
);

CREATE TABLE Zbozi
(
    ID_Zbozi int identity (1,1) primary key,
    Nazev varchar(30) NOT NULL,
    Barva varchar(25) NOT NULL,
    RokVyroby date NOT NULL,
    ObsahObjem int NOT NULL,
    Baleni varchar(25) NOT NULL,
    RozsireneInfo varchar(150),
    ID_TypZbozi int foreign key references TypZbozi (ID_TypZbozi),
    ID_Dodavatel int foreign key references Dodavatel
(ID_Dodavatel)
);

CREATE TABLE MetodaPlaceni
(
    ID_Metoda int identity (1,1) primary key,
    Nazev varchar(50) NOT NULL,
);

CREATE TABLE ZboziNaSklade
(
    ID_ZboziNaSklade int identity (1,1) primary key,
    Pocet int NOT NULL,
    Cena int NOT NULL,
    ID_Sklad int foreign key references Sklad (ID_Sklad),
    ID_Zbozi int foreign key references Zbozi (ID_Zbozi)
);

CREATE TABLE Faktura
(
    ID_Faktura int identity (1,1) primary key,
    DatumVystaveni date NOT NULL,
    DatumSplatnosti date NOT NULL,
    Zaplaceno bool NOT NULL,
    ID_Metoda int foreign key references MetodaPlaceni (ID_Metoda)
);

CREATE TABLE Objednavky
(
    ID_Objednavka int identity (1,1) primary key,
    DatumVystaveni date NOT NULL,
    DatumStornovani date NOT NULL,
    Vyrizeno bool NOT NULL,
    ID_Odberatel int foreign key references Odberatel
(ID_Odberatel),
    ID_Faktura int foreign key references Faktura (ID_Faktura),
    ID_Prodejce int foreign key references Prodejce (ID_Prodejce)
);

```

```

CREATE TABLE ObjednaneZbozi
(
    ID_ObjednaneZbozi int identity (1,1) primary key,
    Pocet int NOT NULL,
    ID_ZboziNaSklade int foreign key references ZboziNaSklade
(ID_ZboziNaSklade),
    ID_Objednavka int foreign key references Objednavky
(ID_Objednavka)
);

go
create procedure registrace_prodejce
@jmeno varchar(15),
@prijmeni varchar(25),
@adresa varchar(50) = NULL,
@telefon int = NULL,
@mobil int = NULL,
@skype varchar(50) = NULL,
@icq varchar(15) = NULL,
@email varchar(50) = NULL,
@datum_narozeni date = NULL,
@pohlavi bit = NULL,
@teritorium varchar(35)
as
begin
    declare @id_teritorium int
    select @id_teritorium = ID_MaTeritorium
    from MaTeritorium where Nazev = @teritorium
    if (@id_teritorium IS NULL)
    begin
        print 'Dané teritorium neexistuje - prosím nejdříve jej zadejte'
    end
    else
    begin
        declare @id_udaje int
        select @id_udaje = ID_Udaje
        from KontaktniUdaje where Telefon = @telefon AND Mobil = @mobil
and Email = @email
        if (@id_udaje IS NULL)
        begin
            insert into KontaktniUdaje (Telefon, Mobil, Email, ICQ, Skype,
Adresa)
            values (@telefon, @mobil, @email, @icq, @skype, @adresa)
            set @id_udaje = IDENT_CURRENT('KontaktNiUdaje')
        end
        declare @id_prodejce int
        select @id_prodejce = ID_Prodejce
        from Prodejce where Jmeno = @jmeno and Prijmeni = @prijmeni and
DatumNarozeni = @datum_narozeni
        if (@id_prodejce IS NULL)
        begin
            insert into Prodejce (Jmeno, Prijmeni, DatumNarozeni, Pohlavi,
ID_MaTeritorium, ID_Udaje)
            values (@jmeno, @prijmeni, @datum_narozeni, @pohlavi,
@id_teritorium, @id_udaje)
        end
    end
end

```

```

        print 'Tento prodejce již pracuje pro naši firmu!'
    end
end
end
go

begin tran RegistrujProdejce
execute registrace_prodejce 'Pavel', 'Peska', 'Vítkov, Selská
942', '723634527', '556300153', 'Pajosh080', '247-160-421',
'pavel.paska@gmail.com', '1989-12-15', '1', 'Olomoucko'

select * from Prodejce

ROLLBACK TRAN RegistrujProdejce

go
create procedure prideleni_automobilu
@jmeno varchar(15),
@prijmeni varchar(25),
@znacka varchar(15),
@popis varchar(20) = NULL,
as
begin
    declare @id_prodejce int
    select @id_prodejce = ID_Prodejce
    from Prodejce where Jmeno = @jmeno and Prijmeni = @prijmeni

    declare @id_automobil int
    select @id_automobil = ID_Automobil
    from Automobil where Znacka = @znacka

    if (@id_prodejce IS NULL)
    begin
        print 'Daný prodejce neexistuje - prosím nejdříve jej
registrujte'
    end
    else if (@id_automobil IS NULL)
    begin
        print 'Daný automobil neexistuje - prosím nejdříve jej
zadejte'
    end
    else
    begin
        insert into MaAutomobil (Popis, ID_Prodejce, ID_Automobil)
        values (@popis, @id_prodejce, @id_automobil)
    end
end
end
go

begin tran PridelAutomobil
execute prideleni_automobilu 'Pavel', 'Peska', '1A7 45-52', 'Půjčeno
na 2 měsíce'

select * from MaAutomobil

ROLLBACK TRAN PridelAutomobil

```

```

go
create procedure karta_zbozi
@nazev varchar(30),
@barva varchar(25),
@rok_vyroby date,
@obsah_objem int,
@baleni varchar(25),
@rozsirene_info varchar(150) = NULL,
@typ_zbozi varchar(15),
@dodavatel varchar(30) = NULL
as
begin
declare @id_dodavatel int
select @id_dodavatel = ID_Dodavatel
from Dodavatel where NazevFirmy = @dodavatel

declare @id_typ_zbozi int
select @id_typ_zbozi = ID_TypZbozi
from TypZbozi where Nazev = @typ_zbozi

if (@id_dodavatel IS NULL)
begin
print 'Daný dodavatel neexistuje - prosím nejdříve jej
registrujte'
end
else if (@id_typ_zbozi IS NULL)
begin
print 'Daný typ zboží neexistuje - prosím nejdříve jej
zadejte'
end
else
begin
insert into Zbozi (Nazev, Barva, RokVyroby, ObsahObjem,
Baleni, RozsireneInfo, ID_TypZbozi, ID_Dodavatel)
values (@nazev, @barva, @rok_vyroby, @obsah_objem, @baleni,
@rozsirene_info, @id_typ_zbozi, @id_dodavatel)
end
end
go

begin tran VytvorZbozi
execute karta_zbozi 'Prací gel', 'Tmaví modrá', '2012-03-30', '5 l',
'Plastová nádoba', 'Síla na jakoukoli špínu', 'Prací prostředky',
'Ariel'

select * from Zbozi

ROLLBACK TRAN VytvorZbozi

```

```

go
create procedure udrzba_vozu
@znacka varchar(15),
@rok_vyroby date,
@stav_tachometru int,
@typ_udrzby varchar(15)
as
begin
    declare @id_automobil int
    select @id_automobil = ID_Automobil
    from Automobil where Znacka = @znacka and RokVyroby = @rok_vyroby

    if (@id_automobil IS NULL)
    begin
        print 'Daný automobil neexistuje - prosím nejdříve jej
registrujte'
    end
    else if (@typ_udrzby = 'Olej')
    begin
        declare @olej int
        set @olej = @stav_tachometru + 10000
        update Automobil set PocetKm = @stav_tachometru
        update Automobil set VymenaOleje = @olej
    end
    else
    begin
        declare @rozvody int
        set @rozvody = @stav_tachometru + 30000
        update Automobil set PocetKm = @stav_tachometru
        update Automobil set Rozvody = @rozvody
    end
end
go

begin tran ProvedUdrzbu
execute udrzba_vozu '1A7 45-52', '2000-03-30', 143260, 'Olej'

select * from Automobil

ROLLBACK TRAN ProvedUdrzbu

```

```

insert into KontaktniUdaje (Telefon, Mobil, Email, ICQ, Skype, Adresa)
values ('723634527', '556300120', 'i.zigi@seznam.cz', '247-160-
421', 'teowall', 'Vitkov, Opavská 30')

insert into KontaktniUdaje (Telefon, Mobil, Email, ICQ, Skype, Adresa)
values ('723158527', '554200120', 'palo@seznam.cz', '221-160-
381', 'palo20', 'Opava, Česká 3')

insert into KontaktniUdaje (Telefon, Mobil, Email, ICQ, Skype, Adresa)
values ('777158874', '556100165', 'svihovsky@gmail.com', '', '', 'Brno,
Matièní 332')

insert into KontaktniUdaje (Telefon, Mobil, Email, ICQ, Skype, Adresa)
values ('771758244', '245100165', 'Dvorakova@gmail.com', '', '', 'Bolatice,
Slovanská 78')

select * from KontaktniUdaje

insert into MaTeritorium (Nazev, Popis)
values ('Opavsko', 'Sever Moravy')

insert into MaTeritorium (Nazev, Popis)
values ('Olomoucko', 'Jih Moravy')

insert into MaTeritorium (Nazev, Popis)
values ('Vysočina', 'Střední Čechy')

select * from MaTeritorium

insert into Sklad (NazevSkladu, ID_Udaje)
values ('Vítkov', '1')

select * from Sklad

insert into MistoDodani (Ulice, Mesto, PSC, ID_MaTeritorium)
values ('Dukelská 12', 'Hradec n. Moravici', '74502', 1)

insert into MistoDodani (Ulice, Mesto, PSC, ID_MaTeritorium)
values ('Hlavní 789', 'Vítkov', '74901', 1)

select * from MistoDodani

insert into
Prodejce (Jmeno, Prijmeni, DatumNarozeni, Pohlavi, ID_MaTeritorium, ID_Udaje
)
values ('Radek', 'Adámek', '1990-07-10', '1', '1', '2')

select * from Prodejce

insert into
Automobil (Znacka, Typ, Barva, RokVyroby, PocetKm, Pneumatiky, Rozvody, Vymena
Oleje)
values ('BRC-60-90', 'Citroen
Berlingo', 'Zelená', '2005', '87000', 'Letní', '95000', '90000')

```

```

insert into
Automobil (Znacka, Typ, Barva, RokVyroby, PocetKm, Pneumatiky, Rozvody, Vymena
Oleje)
values ('1A5-75-62', 'Citroen
Berlingo', 'Bílá', '2001', '135000', 'Letní', '150000', '140000')

select * from Automobil

insert into MaAutomobil (Popis, ID_Prodejce, ID_Automobil)
values ('Půjčeno na 5 měsíců', '1', '1')

select Prodejce.Jmeno, Automobil.Typ from Prodejce, Automobil,
MaAutomobil where Prodejce.Jmeno = 'Radek'
and Prodejce.ID_Prodejce = MaAutomobil.ID_Prodejce and
Automobil.ID_Automobil = MaAutomobil.ID_Automobil

insert into Dodavatel (Jmeno, Prijmeni, NazevFirmy, Info, ID_Udaje)
values ('Jaroslav', 'Švihovský', 'Sonet', 'Primární', '3')

insert into Odberatel (Jmeno, Prijmeni, NazevFirmy, Info, ID_Udaje)
values ('Antonie', 'Dvořáková', 'DsPS', '', '4')

select * from Dodavatel
select * from Odberatel

insert into TypZbozi (Nazev, Kategorie)
values ('Prací prášky', 'Prací prostředky')

insert into TypZbozi (Nazev, Kategorie)
values ('Prací gely', 'Prací prostředky')

insert into TypZbozi (Nazev, Kategorie)
values ('Aviváže', 'Prací prostředky')

select * from TypZbozi

insert into
Zbozi (Nazev, Barva, RokVyroby, ObsahObjem, Baleni, RozsireneInfo, ID_TypZboz
i)
values ('Jednička', 'Žlutá', '2012', '5', 'PET', 'Svěží vůně a
měkkost', '3')

insert into
Zbozi (Nazev, Barva, RokVyroby, ObsahObjem, Baleni, RozsireneInfo, ID_TypZboz
i)
values ('Jednička', 'Modrá', '2012', '5', 'PET', 'Svěží vůně a
měkkost', '3')

insert into
Zbozi (Nazev, Barva, RokVyroby, ObsahObjem, Baleni, RozsireneInfo, ID_TypZboz
i)
values ('Arix', 'Bílá', '2012', '8', 'Pytel', 'Prací síla a účinnost v
jednom', '1')

```

```

insert into
Zbozi (Nazev, Barva, RokVyroby, ObsahObjem, Baleni, RozsireneInfo, ID_TypZbozi,
ID_Dodavatel)
values ('Soret', 'Zelená', '2012', '3', 'PET', 'Vyhledávač skvrn', '2', '1')

select * from Zbozi

insert into Material (Nazev, Horlavina, ID_Dodavatel)
values ('Lih', '1', '1')

insert into Material (Nazev, Horlavina, ID_Dodavatel)
values ('Sul', '0', '1')

insert into Material (Nazev, Horlavina, ID_Dodavatel)
values ('Parfém', '1', '1')

select * from Material

insert into JeZMaterialu (ID_TypZbozi, ID_Material)
values ('3', '1')

insert into JeZMaterialu (ID_TypZbozi, ID_Material)
values ('3', '2')

insert into JeZMaterialu (ID_TypZbozi, ID_Material)
values ('3', '3')

select * from JeZMaterialu

insert into MetodaPlaceni (Nazev)
values ('Hotově')

insert into MetodaPlaceni (Nazev)
values ('Platební kartou')

insert into MetodaPlaceni (Nazev)
values ('Bankovním převodem')

insert into MetodaPlaceni (Nazev)
values ('Složenkou')

select * from MetodaPlaceni

insert into
Faktura (DatumVystaveni, DatumSplatnosti, Zaplaceno, ID_Metoda)
values ('2012-2-3', '2012-2-17', '0', '3')

insert into
Faktura (DatumVystaveni, DatumSplatnosti, Zaplaceno, ID_Metoda)
values ('2012-4-3', '2012-4-17', '0', '1')

select * from Faktura

insert into ZboziNaSklade (Pocet, Cena, ID_Zbozi, ID_Sklad)

```

```

values ('320','75','7','1')

insert into ZboziNaSklade (Pocet,Cena, ID_Zbozi, ID_Sklad)
values ('270','75','8','1')

insert into ZboziNaSklade (Pocet,Cena, ID_Zbozi, ID_Sklad)
values ('150','250','9','1')

select * from ZboziNaSklade

insert into
Objednavky (DatumVystaveni, DatumStornovani, Vyrizeno, ID_Odberatel, ID_Fak
tura)
values ('2012-2-1', '2012-2-8', '1', '1', '1')

insert into
Objednavky (DatumVystaveni, DatumStornovani, Vyrizeno, ID_Prodejce, ID_Fakt
ura)
values ('2012-4-1', '2012-4-8', '0', '1', '1')

select * from Objednavky

insert into ObjednaneZbozi (Pocet, ID_ZboziNaSklade, ID_Objednavka)
values ('20', '1', '3')

insert into ObjednaneZbozi (Pocet, ID_ZboziNaSklade, ID_Objednavka)
values ('10', '2', '3')

insert into ObjednaneZbozi (Pocet, ID_ZboziNaSklade, ID_Objednavka)
values ('50', '3', '4')

select * from ObjednaneZbozi

```