UNIVERZITA PARDUBICE

FAKULTA EKONOMICKO - SPRÁVNÍ

BAKALÁŘSKÁ PRÁCE

2007 **Martina PACÁKOVÁ**

UNIVERZITA PARDUBICE

FAKULTA EKONOMICKO - SPRÁVNÍ
ÚSTAV SYSTÉMOVÉHO INŽENÝRSTVÍ A INFORMATIKY

VÝUKOVÁ APLIKACE PRO RELAČNÍ MODEL DAT

BAKALÁŘSKÁ PRÁCE

AUTOR PRÁCE: **Martina Pacáková**

VEDOUCÍ PRÁCE: Ing. Stanislava Šimonová, Ph.D.

2007

UNIVERSITY OF PARDUBICE
FACULTY OF ECONOMICS AND ADMINISTRATION
INSTITUTE OF SYSTEM ENGINEERING AND INFORMATICS

LEARNING APLICATION TO RELATIONAL MODEL DATA

BACHELOR WORK

AUTHOR: Martina Pacáková

SUPERVISOR: Ing. Stanislava Šimonová, Ph.D.

2007

Prohlašuji:

Tuto práci jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 10. 2. 2007

Martina Pacáková

Poděkování

Děkuji paní Ing. Stanislavě Šimonové, Ph.D. za vedení bakalářské práce, za cenné rady a poskytnutí materiálů.

Souhrn

Bakalářská práce je rozdělena na dvě části. V první části bakalářské práce jsou popsány teoretické principy datového modelování se zaměřením na relační model dat a v druhé části je uveden konkrétní příklad užití relačního modelu dat v databázovém systému Microsoft Access. V práci jsou vysvětleny pojmy databáze, entita, atribut, klíč a normalizace.

Summary

The bachelor work is divided into two parts. In the first part the bachelor work is writing theoretic principles data's models intent on relational model data and in the second part is mentioned example relational model data's in system of database Microsoft Access. In the bachelor work are explained concepts as a database, an entity, an attribute, key and normalization.

OBSAH

Souhrn....3
1. ÚVOD....9
2. DATABÁZOVÉ SYSTÉMY A DATOVÉ MODELY....10
2.1. Historie databázových systémů....10
2.2. Základní pojmy a typy databázových systémů....11
2.2.1. Relační databázové systémy....12
2.2.2. Objektové databázové systémy....13
2.2.3. Objektově-relační databáze....14
2.3. Základní typy datových modelů....14
2.3.1. Hierarchický model dat....14
2.3.2. Síťový model dat....14
2.3.3. Relační model dat....14
2.3.4. Objektový model dat....15
2.3.5. Objektově-relační model dat....16
2.4. Architektury databází....16
2.4.1. Centrální architektura....16
2.4.2. Architektura file-server....16
2.4.3. Architektura klient-server....17
2.4.4. Distribuovaná databáze....17
2.5. Transakce....17
3. RELAČNÍ MODEL DAT....18
3.1. Datový model ve fázi analýzy a návrhu....19
3.2. Datový model v implementaci – datové typy....20
3.2.1. Znak (text)....20
3.2.2. Číslo....20
3.2.3. Datum....20
3.2.4. Ano/Ne....20
3.2.5. Memo....20
3.2.6. Měna....20
3.2.7. Automatické číslo....21
3.2.8. Objekt OLE....21
3.2.9. Hypertextový odkaz....21
3.2.10. Hodnoty Null a řetězce s nulovou délkou....21
3.2.11. Indexy....21
4. DIAGRAM ENTIT A VZTAHŮ....22
4.1. Entity....22
4.2. Atributy....23
4.3. Domény....23
4.4. Vztahy....24
4.4.1. Kardinalita....25
4.4.2. Parcialita (volitelnost, členství ve vztahu)....25
4.5. Klíče....26
4.5.1. Kandidátní klíč....26
4.5.2. Primární klíč....27
4.5.3. Cizí klíč....28
4.5.4. Alternativní klíče....28
5. PŘEVOD ERD DO RELAČNÍHO MODELU DAT....29
5.1. Transformace E-R digramu do relačního modelu dat....29
5.2. Normalizace....30

5.2.1. První normální forma 30
5.2.2. Druhá normální forma 30
5.2.3. Třetí normální forma 30
5.2.4. Další normální formy 31
6. DATABÁZOVÝ SYSTÉM MICROSOFT ACCESS 32
6.1. Systém pro správu relační databáze – Microsoft Access 32
6.1.1. Tabulka 32
6.1.2. Dotaz 32
6.1.3. Formulář 32
6.1.4. Sestava 32
7. KONCEPTUÁLNÍ ANALÝZA VYBRANÉ REÁLIE 33
7.1. Určení entit 33
7.2. Určení atributů 33
7.3. Určení primárních klíčů 34
7.4. Určení vztahů mezi entitami 35
7.4.1. Ukázka dalších vztahů 36
7.5. ER diagram 37
7.6. Transformace ERD do RMD 38
7.6.1. Ukázka další transformace ERD do RMD 40
7.7. Normalizace 41
8. IMPLIKACE DO DATABÁZOVÉHO SYSTÉMU MICROSOFT ACCESS 45
8.1. Vytvoření databáze 45
8.2. Vytvoření tabulek 45
8.2.1. Vytvoření tabulky pomocí návrhového zobrazení 46
8.3. Definování vztahů 49
8.4. Dotazy 50
8.5. Formuláře 52
8.6. Sestavy 53
9. ZÁVĚR 55
10. SEZNAM LITERATURY A POUŽITÝCH ZDROJŮ 56
11. PŘÍLOHA 58

Seznam tabulek:

Tabulka 1 Určení atributů 33
Tabulka 2 Určení primárních klíčů 34
Tabulka 3 Zaměstnanci 1. NF 41
Tabulka 4 Zaměstnanci 2. NF 42
Tabulka 5 Zaměstnanci 3. NF 42
Tabulka 6 Dodavatelé 1. NF 42
Tabulka 7 Dodavatelé 2. NF 43
Tabulka 8 Dodavatelé 3. NF 43
Tabulka 9 Normalizace entity Odběratelé 44
Tabulka 10 Normalizace entity Léky 44
Tabulka 11 Přehled všech entit 44

Seznam obrázků:

Obrázek 1 Postup tvorby datového modelu [25] 19
Obrázek 2 Vztah Dodavatelé - Léky 35
Obrázek 3 Vztah Odběratelé – Léky 35
Obrázek 4 Vztah Zaměstnanci – Odběratelé 35
Obrázek 5 Vztah Zaměstnanci - Odběratelé 35
Obrázek 6 Vztah Dodavatelé – Léky 36
Obrázek 7 Vztah Dodavatelé – Léky 36
Obrázek 8 Vztah Odběratelé - Léky 36
Obrázek 9 ER diagram 37
Obrázek 10 Vztah Dodavatelé – Léky 38
Obrázek 11 RMD Dodavatelé – Léky 38
Obrázek 12 Vztah Odběratelé – Léky 38
Obrázek 13 RMD Odběratelé – Léky 38
Obrázek 14 Vztah Zaměstnanci – Dodavatelé 39
Obrázek 15 RMD Zaměstnanci – Dodavatelé 39
Obrázek 16 Vztah Zaměstnanci – Odběratelé 39
Obrázek 17 RMD Zaměstnanci - Odběratelé 39
Obrázek 18 Vztah Dodavatelé – Léky 40
Obrázek 19 RMD Dodavatelé – Léky 40
Obrázek 20 Vztah Dodavatelé – Léky 40
Obrázek 21 RMD Dodavatelé – Léky 40
Obrázek 22 Vztah Odběratelé – Léky 41
Obrázek 23 RMD Odběratelé - Léky 41
Obrázek 24 Vytvoření tabulky 45
Obrázek 25 Vytvoření tabulky pomocí návrhového zobrazení 46
Obrázek 26 Tabulka vytvořená pomocí průvodce 47
Obrázek 27 Tabulka Dodavatelé 47
Obrázek 28 Tabulka Zaměstnanci 48
Obrázek 29 Relace 49
Obrázek 30 Výběrový dotaz 50
Obrázek 31 Výsledek výběrového dotazu 51
Obrázek 32 Výsledek výběrového dotazu bez kritéria 51
Obrázek 33 Sestavení formuláře 52
Obrázek 35 Vytvoření sestavy 54
Obrázek 36 Sestava 54

Seznam zkratek:

MS	- Microsoft
RDBMS	- Systém správy relační databáze
RMD	- Relační model dat
SŘBD	- Systém řízení báze dat
ERD	- E-R diagram
NF	- Normální forma
ADT	- Abstraktní datové typy
OID	- Objektové identifikátory
SQL2	- Structure Query Language 2
LAN	- Local Area Network
WWW	- World Wide Web
AK/CAK	- Alternation Key

1. ÚVOD

Jako téma bakalářské práce byla zvolena Výuková aplikace pro relační model dat. Hlavním zdrojem informací při psaní práce byla literatura z univerzitní knihovny a dokumenty související s tímto tématem na Internetu.

Pro přehlednost je práce rozdělena do dílčích úseků, a to základní typy a rozdělení datových systémů, relační model dat, diagramy entit a vztahů, převod ER diagramu do relačního modelu dat a databázový systém Microsoft Access. Závěrečným bodem této práce je ucelený příklad. Práce je určena především pro zájemce o relační databáze v databázovém systému Microsoft Access.

Cílem práce bylo seznámení s teoretickými principy relačního modelu dat, se základními pojmy jako entita, atribut, klíče a implementace uceleného příkladu do relačního databázového systému Microsoft Access.

2. DATABÁZOVÉ SYSTÉMY A DATOVÉ MODELY

2.1. Historie databázových systémů

Databáze je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento systém se v české odborné literatuře nazývá systém řízení báze dat (SŘBD). Běžně se označením databáze myslí jak uložená data, tak i software (SŘBD). [9, 5]

Předchůdcem databází byly papírové kartotéky. Umožňovaly uspořádávání dat podle různých kritérií a zatřiďování nových položek. Veškeré operace s nimi prováděl přímo člověk. Správa takových kartoték byla v mnohém podobná správě dnešních databází. [9]

Dalším krokem bylo převedení zpracování dat na stroje. Za první velké strojové zpracování dat lze považovat sčítání lidu ve Spojených státech v roce 1890. Paměťovým médiem byl děrný štítek a zpracování sebraných informací probíhalo na elektromechanických strojích. Elektromechanické stroje se využívaly pro účely zpracování dat další půlstoletí. [9]

Velkým impulsem pro další rozvoj databází byl překotný vývoj počítačů v padesátých letech 20. století. Ukázalo se, že původně univerzální používání strojového kódu procesorů je pro databázové úlohy neefektivní, a proto se objevil požadavek na vyšší jazyk pro zpracování dat. [5, 9]

V roce 1959 se konala konference zástupců firem, jejímž závěrem byl požadavek na univerzální databázový jazyk. Výsledkem byla o rok později publikovaná první verze jazyka COBOL, který byl po mnoho dalších let nejrozšířenějším jazykem pro hromadné zpracování dat. [9, 22]

V roce 1965 na konferenci CODASYL byl vytvořen výbor Database Task Group (DBTG), který měl za úkol vytvořit koncepci databázových systémů. Začaly vznikat první síťové SŘBD na sálových počítačích. Jedním z prvních průkopníků databází byl Charles Bachman. V roce 1971 se objevily pojmy jako schéma databáze, jazyk pro definici schématu, subschéma a podobně. Byla zde popsána celá architektura síťového databázového systému. [9]

Ve stejné době byly vyvíjeny i hierarchické databáze. Jedním z prvních SŘBD byl IMS, který byl vyvinut firmou IBM. Systém IMS stále patří k nejrozšířenějším na sálových počítačích. V roce 1970 začínají zveřejněním článku E. F. Codda první relační databáze, které pohlížejí na data jako na tabulky. Kolem roku 1974 se vyvíjí první verze dotazovacího jazyka SQL. Vývoj této technologie po 10 letech přinesl výkonově použitelné systémy, srovnatelné se síťovými a hierarchickými databázemi.

V 90. letech 20. století se začaly objevovat první objektově orientované databáze, jejichž filozofie byla přebírána z objektově orientovaných jazyků. Tyto databáze měly podle předpokladů vytlačit relační systémy. Původní předpoklady se však v komerční sféře nenaplnily a vznikla kompromisní objektově-relační technologie. [9, 5]

2.2. Základní pojmy a typy databázových systémů

Databázový systém tvoří databáze a systém řízení báze dat. Databázový systém podle C. J. Date tvoří data, hardware, software a uživatelé. Databázový systém může být chápán i jako počítačový systém správy uložených záznamů. V tomto systému jsou nejdůležitější funkce [25]:

- přidání nového prázdného souboru do databáze
- vložení nových dat do existujícího souboru
- výběr dat z existujícího souboru
- oprava dat v existujícím souboru
- zrušení existujícího souboru z databáze

Data jsou údaje, které mají určitou vypovídací schopnost. Mohou být určitým způsobem uspořádána a jsou uživateli k dispozici v různých formách. Data jsou obvykle rozdělena na dílčí údaje (atributy) o dané množině objektů (entit).

Záznam je souhrn údajů (atributů) o dané části objektu, které jsou v polích charakterizovaných názvem a datovým typem. [6]

Systém řízení báze dat (SŘBD) je souhrn procedur a datových struktur, které zajistí nezávislost databázových aplikací na detailech vytváření, výběru, uchování, modifikaci a zabezpečení ochrany databází na fyzických paměťových strukturách počítače. [9]

2.2.1. Relační databázové systémy

V 70tých letech vznikají relační databázové systémy. Hlavním přínosem těchto systémů je, že jsou postavené na matematické teorii množin s využitím relační algebry a kalkulu. Termín relační pramení ze skutečnosti, že každý záznam v databázi obsahuje informace, vztažené k jedinému subjektu a pouze k tomuto subjektu. V systému správy relační databáze (RDBMS) pracuje systém se všemi daty v tabulkách. Tabulky uchovávají informace o subjektu a mají sloupce, které obsahují různé druhy informací o tomto subjektu. Systém správy relační databáze umožňuje kontrolu nad tím, jak definovat data, pracovat s nimi a sdílet je s ostatními. [12, 25]

Dr. Codd definoval 12 pravidel pro relační SŘBD [23]:

1) Informační pravidlo – Všechny informace v relační databázi jsou vyjádřeny explicitně na logické úrovni jediným způsobem – hodnotami v tabulkách.
2) Pravidlo jistoty – Všechna data v relační databázi jsou zaručeně přístupná kombinací jména tabulky s hodnotami primárního klíče a jménem sloupce.
3) Systematické zpracování nulových hodnot – Nulové hodnoty jsou plně podporovány relačním SŘBD pro reprezentaci informace, která není definována a to nezávisle na datovém typu.
4) Dynamický on-line katalog založený na relačním modelu – Popis databáze je vyjádřen na logické úrovni stejným způsobem jako zákaznická data, takže autorizovaný uživatel může aplikovat stejný relační jazyk ke svému dotazu jako uživatel při práci s daty.
5) Obsáhlý datový podjazyk – Relační systém může podporovat několik jazyků a různých módů použitých při provozu terminálu. Nicméně musí být nejméně jeden příkazový jazyk s dobře definovanou syntaxí, který obsáhle podporuje definici dat, definici pohledů, manipulaci s daty jak interaktivně, tak programem.

6) Pravidlo vytvoření pohledů – Všechny pohledy, které jsou teoreticky možné, jsou také systémem vytvořitelné.

7) Schopnost vkládání, vytvoření a mazání – Schopnost zachování relačních pravidel u základních i odvozených relací je zachována nejen při pohledu na data, ale i při operacích průniku, přidání a mazání dat.

8) Fyzická datová nezávislost – Aplikační programy jsou nezávislé na fyzické datové struktuře.

9) Logická datová nezávislost – Aplikační programy jsou nezávislé na změnách v logické struktuře databázového souboru.

10) Integritní nezávislost – Integritní omezení se musí dát definovat prostředky relační databáze nebo jejím jazykem a musí být schopna uložení v katalogu a nikoliv v aplikačním programu.

11) Nezávislost distribuce – Relační systém řízení báze dat musí být schopný implementace na jiných počítačových architekturách.

12) Pravidlo přístupu do databáze – Jestliže má relační systém jazyk nízké úrovně, pak tato úroveň nemůže být použita k vytváření integritních omezení a je nutno vyjádřit se v relačním jazyce vyšší úrovně.

2.2.2. Objektové databázové systémy

V devadesátých letech minulého století s mohutným nástupem objektově orientovaného programování začaly vznikat i objektově orientované databáze, které si kladou za cíl ulehčit a zrychlit práci s daty. Není to jednoduché, protože relační a objektový přístup je od základu rozdílný. Objektové databáze kombinují prvky objektově orientovaného programování s databázovými schopnostmi. Rozšiřují funkčnost objektových programovacích jazyků (C++, Smalltalk, Java) a poskytují plnou schopnost programování databáze. [18]

2.2.3. Objektově-relační databáze

Objektově-relační databáze představují jakýsi kompromis mezi objektovými databázemi a relačními databázemi. Existuje jak mnoho výhod, tak i mnoho nevýhod pro relační i objektové databáze.

2.3. Základní typy datových modelů

Datový model je souhrn pravidel pro reprezentaci logické organizace dat v databázi. Rozeznáváme pět základních modelů dat – hierarchický, síťový, relační, objektový a objektově-relační.

2.3.1. Hierarchický model dat

Data jsou organizovaná do stromové struktury, kdy každý záznam představuje uzel a vzájemný vztah mezi záznamy je typu rodič/potomek. Použití hierarchického modelu je vhodné, když i pozorovaná realita má hierarchickou strukturu. Nevýhodou hierarchického modelu jsou složité operace vkládání a rušení záznamů a občas nepřirozená organizace dat. [23]

2.3.2. Síťový model dat

Síťový model dat je zobecněním hierarchického modelu dat, který doplňuje o vztahy. Tyto vztahy jsou označovány jako C-množiny neboli Set. Tyto sety propojují záznamy různého či stejného typu, přičemž spojení může být realizováno na jeden nebo více záznamů. Nevýhodou síťového modelu je především nepružnost a obtížná změna její struktury. [20]

2.3.3. Relační model dat

Relační databázový model má jednoduchou strukturu. Data jsou organizovaná v tabulkách, které se skládají z řádků a sloupců. Všechny databázové operace jsou prováděny na těchto tabulkách. Relační model vychází z jistého souboru základních matematických principů odvozených z teorie množin a predikátové logiky. Relační model definuje způsob, jakým je možné data reprezentovat (tedy strukturu dat), způsoby jejich ochrany (neboli integritu dat) a dále operace, které se mohou nad daty provádět (manipulace s daty). [17]

Relační model dat odděluje data, která jsou chápána jako relace od jejich implementace. Při manipulaci s daty není nutné znát přístupové metody k datům v relacích. Pro práci s daty je k dispozici relační kalkul a algebra (matematické aparáty, jimiž lze popsat sémantiku, význam konstrukcí) relačních jazyků.
Pro omezení redundance dat je použita normalizace relací, což znamená vhodně navržené databázové struktury. [5, 25]

Pro porozumění idejím a vyvrácení mýtu, že relační model dat se jmenuje relační podle vztahů mezi daty, je nutné říci, že základním kamenem RMD je databázová relace (množina) obsahující data, která se od matematické relace poněkud liší. Je totiž vybavena pomocnou strukturou, které se říká schéma relace. Toto schéma obsahuje jméno relace, jména atributů (sloupců) a popisuje domény (integritní omezení). Každá buňka dat musí obsahovat pouze atomické hodnoty. [5]

Databázová relace je tabulka, pohled, výsledek dotazu, což dává možnost pracovat s výsledky dotazů stejně jako s tabulkou (není podporováno ve všech DBMS). Vzhledem k tomu, že relace je množina, která nesmí obsahovat duplicitní prvky a není uspořádána jinak než do sloupců a řádků, neexistuje první, druhý nebo n-tý řádek. Řádky v relaci nemají specifické pořadí, tudíž nejsou ani dosažitelné číslem řádku, musí existovat nějaká konstrukce, která umožní adresovat jednotlivé řádky, tato konstrukce se nazývá primární klíč. [5]

2.3.4. Objektový model dat

Objektové databáze využívají datového modelu, který má objektově orientované aspekty jako třídy s atributy, metodami a integritními omezeními; poskytují objektové identifikátory (OID) pro každou trvalou instanci třídy, podporují zapouzdření (encapsulation), násobnou dědičnost (multiple inheritance) a podporují abstraktní datové typy. Rozšiřují funkčnost objektových programovacích jazyků (C++, Smalltalk, Java) a poskytují plnou schopnost programování databáze. Datový model aplikace a datový model databáze se ve výsledku hodně shodují a výsledný kód se dá mnohem efektivněji udržovat. [18]

2.3.5. Objektově-relační model dat

Objektově-relační databázové modely dat využívají datový model tak, že "přidávají objektovost do tabulek". Všechny trvalé informace jsou stále v tabulkách, ale některé položky mohou mít bohatší datovou strukturu, nazývanou abstraktní datové typy (ADT). ADT je datový typ, který vznikne zkombinováním základních datových typů. Podpora ADT je atraktivní, protože operace a funkce asociované s novými datovými typy mohou být použity k indexování, ukládání a získávání záznamů na základě obsahu nového datového typu. Objektově-relační modely dat jsou nadmnožinou relačních modelů dat a pokud se nevyužijí žádné objektové rozšíření jsou ekvivalentní SQL2. Proto má objektově-relační model dat omezenou podporu dědičnosti, polymorfismu, referencí a integrace s programovacím jazykem. [18]

2.4. Architektury databází

Architektura databáze představuje všeobecný databázový koncept pro vysvětlení struktury databázového systému. [23]

2.4.1. Centrální architektura

Při této architektuře jsou data i systém řízení báze dat v centrálním počítači. Tato architektura je typická pro terminálovou síť, kdy se po síti přenáší vstupní údaje z terminálu na centrální počítač do příslušné aplikace, výstupy z této aplikace se přenáší na terminál. Vzhledem k tomu, že aplikační program i vlastní zpracování probíhá na centrálním počítači, který může zpracovávat více úloh, mají odezvy na dotazy určité zpoždění. [23]

2.4.2. Architektura file-server

Tato architektura souvisí s rozšířením osobních počítačů a sítí LAN. Systém řízení báze dat a příslušné databázové aplikace jsou provozovány na jednotlivých počítačích, data jsou umístěna na file-serveru a mohou být sdílena. Aby nedocházelo ke kolizím při přístupu více uživatelů k jedněm datům, musí SŘBD používat vhodný systém zamykání. [23]

2.4.3. Architektura klient-server

Architektura klient-server je založena na lokální síti (LAN), personálních počítačích a databázovém serveru. Na personálních počítačích běží program podporující například vstup dat. Dotaz se dále předává pomocí jazyka SQL na databázový server, který jej vykoná a vrátí výsledky zpět na personální počítač. Databázový server je tedy nejvíce zatíženým prvkem systému a musí být tvořen dostatečně výkonným počítačem. [23]

2.4.4. Distribuovaná databáze

Distribuovaná databáze je množina databází, která je uložena na několika počítačích. Uživateli se však jeví jako jedna velká databáze. Distribuovaná databáze se charakterizuje třemi vlastnostmi [23]:

- transparentnost (z pohledu klienta se zdá, že všechna data jsou zpracovávána na jednom serveru v lokální databázi)
- autonomnost (s každou lokální bází dat zapojenou do distribuované databáze je možno pracovat nezávisle na ostatních databázích)
- nezávislost na počítačové síti

2.5. Transakce

Jedním ze základních problémů pro udržení konzistence dat v databázi je situace, kdy během zpracování příkazu dojde k nestandardnímu ukončení práce s databází. Nastane-li chyba během příkazu, který aktualizuje hodnoty v databázi, stávají se data nekonzistentními. Z těchto důvodů byl vytvořen koncept transakčního zpracování. [10]

Transakce je nedělitelný logický celek, který bude buď proveden celý, nebo se neprovede žádná jeho část. Od začátku transakce až do jejího ukončení jsou všechny změny provedené v datech uschovány, aby bylo možné v libovolném okamžiku provést návrat do stavu, který byl na začátku transakce. Dojde-li před ukončením transakce k chybě, jsou při opětovném spuštění databáze všechna data uvedena do stavu, v jakém se nacházela před začátkem transakce. Procesu, který provede návrat ke stavu na počátku transakce, se říká „odrolování“ (z anglického rollback). Odrolování je možné vyvolat kdykoliv před ukončením transakce. Databázovému systému je nutné umožnit, aby dočasně ukládal informace o změnách v databázi. [10]

3. RELAČNÍ MODEL DAT

Obecně se dá říci, že relační databázové systémy vykazují následující charakteristické vlastnosti [5]:

- veškerá data se pomyslně dají reprezentovat v pravidelně uspořádaných strukturách s řádky a sloupci, kterým se říká relace
- všechny hodnoty v databázi jsou skalární, to znamená, že v každé konkrétní pozici řádku a sloupce dané relace se nachází právě jedna hodnota
- operace v databázi se provádějí vždy nad celou relací a jejich výsledkem je opět jiná celá relace

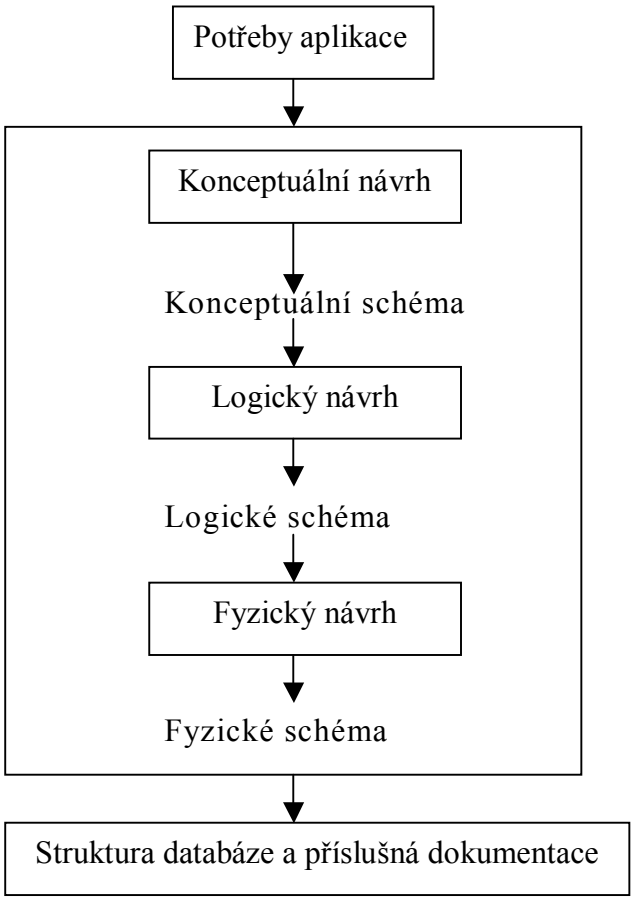
Pro relační model stačí pouze pomyslná reprezentace dat v relaci, relační model tedy vůbec neurčuje, jak mají být data fyzicky implementována. Toto oddělení myšlenkové (konceptuální) a fyzické reprezentace dat se dnes zdá být téměř samozřejmé, avšak před asi 30 lety, kdy programování databází znamenalo obvykle přímé zapisování strojového kódu pro fyzickou manipulaci se zařízeními pro ukládání dat, bylo důležitou, ba dokonce průlomovou inovací. [9, 5]

Relační model dat se jmenuje relační podle vztahů mezi daty, je nutné říci, že základním kamenem relačního modelu dat je databázová relace (množina) obsahující data, která se od matematické relace poněkud liší. Je vybavena pomocnou strukturou, které se říká schéma relace. Toto schéma obsahuje jméno relace, jména atributů (sloupců) a popisuje domény (integritní omezení). Každá buňka dat musí obsahovat pouze atomické hodnoty. [5]

Relace ve skutečnosti nepotřebují vůbec žádnou fyzickou reprezentaci. Určitá množina záznamů se tak může opravdu mapovat na reálnou, fyzickou tabulku, uloženou v určitém místě na disku, stejně tak ale může vyjadřovat pouze jisté sloupce, vybrané z deseti různých tabulek a doplněné o několik vypočtených polí - která se ovšem nikam fyzicky neukládají. Relací může být cokoliv, co je uspořádáno do struktury (formátu) řádků a sloupců a co obsahuje skalární hodnoty. [9, 5]

3.1. Datový model ve fázi analýzy a návrhu

Relační datový model je třeba navrhovat s ohledem na potřeby aplikace. Nejprve je nutné řešit konceptuální model reprezentovaný konceptuálním schématem, který je následně transformovaný do logického schématu. Logické schéma reprezentuje hodnotově-orientovaný datový model, v kterém jsou už vyjádřené vazby mezi objekty s ohledem na zabezpečení integrity dat, normalizace datového modelu a podobně. Implementace datového modelu je vytvořená v procesu fyzického návrhu, jehož výsledkem je fyzické schéma datového modelu. Toto schéma je implementačně závislé a obsahuje podrobnou specifikaci datových struktur, způsob implementace datových typů, organizaci dat a přístupové metody k datům. [25]



Obrázek 1 Postup tvorby datového modelu [25]

3.2. Datový model v implementaci – datové typy

I u velmi jednoduché tabulky je vidět, že data obsažená v různých sloupcích mohou být různého druhu. Odborně se těmto druhům hodnot říká datové typy.

3.2.1. Znak (text)

Znak (text) je základní datový typ. Sloupec tohoto typu může obsahovat libovolné znaky a to jak písmena, tak i čísla a některé další znaky jako čárky, pomlčky, středníky, závorky atd., používá se zejména pro názvy a dále pro uložení všech hodnot, které není možné zařadit do některého dalšího datového typu. [12]

3.2.2. Číslo

Číselný datový typ může obsahovat pouze číselné hodnoty určitého rozsahu. Rozlišuje se několik druhů číselných datových typů. Mezi základní rozlišení patří, jestli se jedná o celé nebo desetinné číslo. Číselné rozsahy jednotlivých datových typů závisí na používaném databázovém systému. [12]

3.2.3. Datum

Slouží pro uchování datumu v libovolném formátu (den-měsíc-rok). Ve většině databázových systémů bývá do datumu zahrnut i čas. [12]

3.2.4. Ano/Ne

Tento speciální datový typ slouží pro uchování logických hodnot Ano (pravda) a Ne (nepravda). [10]

3.2.5. Memo

Datový typ memo se používá pouze pro dlouhé řetězce textu, které mohou přesahovat 255 znaků nebo které mohou obsahovat formátovací znaky, jako například tabulátory nebo nové řádky. [12]

3.2.6. Měna

Datový typ měna se používá pro ukládání peněžních hodnot. Může se použít pro jakékoliv číselné pole, které musí obsahovat přesná setinná čísla s přesností až na čtyři desetinná místa. Datový typ měna má přesnost celých čísel, ale s pevným počtem desetinných míst. [12]

3.2.7. Automatické číslo

Datový typ automatické číslo je speciálně navržen pro automatické generování hodnot primárních klíčů. Do každé tabulky se může vložit pouze jedno pole, které používá datový typ automatické číslo. [12]

3.2.8. Objekt OLE

Datový typ objekt OLE umožňuje ukládat komplexní data, jako jsou například obrázky, grafy nebo zvuky, s nimiž pak lze pracovat pomocí dynamických vazeb s jinými aplikacemi běžících pod Windows. [12]

3.2.9. Hypertextový odkaz

Datový typ hypertextový odkaz umožňuje ukládání jednoduchých nebo složitějších odkazů na externí soubory nebo dokumenty. Takový odkaz může obsahovat adresu typu URL, která ukazuje na nějaké místo na World Wide Webu nebo na lokálním intranetu. [12]

3.2.10. Hodnoty Null a řetězce s nulovou délkou

Relační databáze podporují v polích zvláštní hodnotu, nazvanou Null, která označuje neznámou hodnotu. Hodnoty Null mají zvláštní vlastnosti. Hodnota Null nemůže být rovna žádné jiné hodnotě, ani další hodnotě Null. Nemohou se tedy spojit dvě tabulky podle hodnot Null. Do polí typu Text nebo Memo se může zadat řetězec s nulovou délkou, hodnota v tomto poli je známá, ale pole je prázdné. [10]

3.2.11. Indexy

Čím více dat bude obsaženo v tabulkách, tím více je potřeba indexů, které Microsoft Accessu pomáhají efektivně prohledávat data. Index je jednoduše řečeno nějaká vnitřní tabulka, která obsahuje dva sloupce: hodnotu v poli nebo polích, která jsou indexována, a umístění každého záznamu v tabulce, který tuto hodnotu obsahuje. Většina indexů, které je třeba definovat, bude pravděpodobně obsahovat hodnoty pouze z jednoho pole. MS Access tento typ indexu používá pro zúžení počtu záznamů, které musí prohledat při stanovení nějakého vyhledávacího kritéria nad tímto polem. Pokud jsou nadefinované indexy nad více poli a stanoveny vyhledávací kritéria pro více než jedno z polí, MS Access použije pro rychlé nalezení požadovaných řádků indexy dohromady. [12]

4. DIAGRAM ENTIT A VZTAHŮ

Model entit a vztahů, který popisuje data jako entity, atributy a relace (vztahy), zavedl Peter Pin Shan Chen. Současně navrhl metodu jeho zobrazení do diagramů, které nazval diagramy entit a vztahů (Entity Relationship Diagrams, E-R diagram), a které se záhy rychle rozšířily a staly se obecně uznávanými. V E-R diagramech se entity popisují pomocí obdélníků, atributy se vyjadřují jako elipsy (ovály) a vztahy se znázorňují pomocí kosočtverců. Povaha vztahů mezi entitami se vyjadřuje různými způsoby. Velkou výhodou E-R diagramů je, že se velice snadno nakreslí a přitom jsou dobře srozumitelné. [3, 5]

4.1. Entity

Entita je cokoliv, o čem je v systému třeba uchovávat nějaké informace. [9]

Při zahájení prací na návrhu datového modelu není sestavení prvotního seznamu entit nijak obtížné. Entity představují podstatná jména a slovesa. Sestavený prvotní seznam vhodných kandidátů na entity je nutno prověřit a zjistit, jestli je úplný a konzistentní. Vyhledávají se opakované entity a dále takové dvojice (nebo n-tice) různých entit, které jsou v seznamu zachyceny jako jediná entita. Většina entit modeluje objekty nebo události ve fyzickém světě. Těmto entitám se říká konkrétní entity. Některé entity však mohou modelovat také abstraktní pojmy (myšlenky). Těm se přirozeně říká abstraktní entity a zřejmě nejběžnějším příkladem je entita, která modeluje určitý vztah mezi jinými entitami. Někdy stačí namodelovat pouhou existenci takovéhoto vztahu. Jindy je potřeba k existujícímu vztahu evidovat různé doplňující informace a jeho charakteristické vlastnosti. Jestli se vztahy bez žádných atributů mají také modelovat jako samostatné entity je otázkou. [9]

4.2. Atributy

Navrhovaný systém o každé entitě zaznamenává a sleduje určité skutečnosti. Těmto údajům se říká atributy dané entity. Určování atributů daného modelu je sémantický proces. Rozhoduje se podle významu dat a podle způsobu jejich využití. Dokazuje se, že konkrétní návrh má určitou chybu, nepřítomnost chyb se však dokázat nedá. Návrh by tedy neměl být složitější než musí nezbytně být. Výsledný model musí být natolik flexibilní, aby dokázal odpovídat nejen na otázky, které mu uživatelé budou klást hned teď, ale také na otázky, jaké se dají do budoucna jistým způsobem předvídat. Jednou ze známek opravdu dobrého návrháře je důkladnost a tvořivost, s jakou se od uživatelů pokouší zjistit takovéto potenciální otázky. Uživatelé většinou nevědí, co vlastně chtějí. Cena za vysokou flexibilitu je většinou složitost výsledného systému. Čím více způsoby se data rozkládají a skládají, tím více různých výjimek se musí ošetřit. Je důležité umět identifikovat všechny výjimky a systém je třeba navrhnout tak, aby co nejvíce výjimek dokázal zvládnout bez zbytečného obtěžování uživatele. Flexibilita jde na úkor vyšší složitosti. Je velice důležité snažit se odchytit co nejvíce výjimek, naopak je velice rozumné některé z nich vypustit jako nepravděpodobné, protože jejich zpracování by bylo příliš nákladné a náročné. [9]

4.3. Domény

Specifikace pole (běžně označovaná jako doména) reprezentuje všechny prvky pole. Každá specifikace pole zahrnuje tři typy vlastností: obecné, fyzické a logické. V záhlaví relace je pro každý atribut uvedena dvojice údajů JménoAtributu:JménoDomény. Definice domény neboli oboru hodnot popisuje typ dat, která daný atribut reprezentuje. Obor hodnot tvoří množina všech přípustných platných hodnot, které smí atribut obsahovat. Obory hodnot se zaměňují s datovými typy. Datový typ je fyzický pojem, kdežto obor hodnot je logický. Pojem obor hodnot má také širší význam než pojem datový typ. [9]

4.4. Vztahy

Kromě atributů jednotlivých entit se musí v datovém modelu určit také vztahy, definované mezi různými entitami. Na myšlenkové (konceptuální) úrovní jsou vztahy jednoduše jistými asociacemi mezi entitami. Entity zapojené do určitého vztahu, se nazývají účastníci. Počet účastníků se označuje jako stupeň vztahu. Většina vztahů je binárních, tedy se dvěma účastníky. Běžné jsou i vztahy se třemi účastníky. Speciálním případem binárního vztahu je entita, která se účastní vztahu sama se sebou. [9]

Vztah mezi libovolnými dvěma entitami může být typu jedna k jedné, jedna k více, nebo více k více.

Účast dané entity v určitém vztahu může být úplná nebo částečná. Jestliže entita nemůže existovat bez účasti v nějaké relaci, pak musí být její účast vždy úplná, jinak je částečná. Podstatné je zde zajistit, aby specifikace částečné nebo úplné účasti ve vztahu platila pro veškeré instance dané entity v libovolném okamžiku. [9]

Vztah je zásadní složkou relační databáze [5, 8, 9]:

- vytváří spojení mezi dvěmi tabulkami, které spolu logicky souvisí; dvojice tabulek je logicky svázána daty, které tyto tabulky obsahují
- napomáhá dalšímu doladění struktur tabulek a minimalizaci redundantních dat, při zřizování vztahů mezi dvěmi tabulkami se dělají menší změny ve strukturách tabulek, toto doladění má za následek efektivnější struktury a minimalizaci redundantních dat, které mohou tabulky obsahovat
- jedná se o mechanismus, který umožní najednou zobrazit data z několika tabulek

4.4.1. Kardinalita

Důležité integritní omezení přibližující konceptuální schéma realitě je kardinalita vztahu resp. poměr. Existují celkem 3 obecné typy kardinality, a to jedna k jedné, jedna k více a více k více. [4]

Vztahy typu 1:1

Vztahy typu jedna k jedné jsou poměrně vzácné a nejčastěji představují vztah mezi entitou určitého nadtypu a podtypu. Dvojice tabulek je spolu svázána typem 1:1, pokud je každý záznam v první tabulce svázán s právě jedním záznamem druhé tabulky a zároveň každý záznam z druhé tabulky je svázán právě s jedním záznamem z první tabulky. Vztah typu 1:1 se obvykle (ale ne vždy) týká podmnožinových tabulek. Nejčastěji slouží ke snížení počtu atributů v relaci, případně modelují podtřídu určité entity. [4, 25]

Vztahy typu 1:N

Vztah typu 1:N se objevuje mezi dvěmi tabulkami v případě, kdy jeden záznam z první tabulky může být navázán na jeden, nebo více záznamů z druhé tabulky, ale každý záznam z druhé tabulky smí být navázán pouze na jediný záznam z první tabulky. Toto je nejčastější typ vztahu, který se v databázích mezi dvěmi tabulkami vyskytuje, také se nejjednodušeji rozpozná. Z pohledu integrity dat jde o zásadní druh vztahu, protože umožňuje eliminovat duplicitní data a udržet redundantní data na absolutním minimu. [4, 25]

Vztahy typu M:N

Dvě tabulky jsou propojeny vztahem typu M:N tehdy, když jeden záznam z první tabulky může být svázán s jedním, nebo více záznamy z druhé tabulky a zároveň jeden záznam z druhé tabulky může být svázán s jedním, nebo více záznamy z první tabulky. Toto je druhý nejčastější typ vztahů, který se v databázi mezi dvěmi tabulkami objevuje. Určit ho může být trochu obtížnější, než tomu bylo v případě typu 1:N. [4, 25]

4.4.2. Parcialita (volitelnost, členství ve vztahu)

Klasifikace účasti ve vztahu indikuje také volitelnost vztahu, tedy zda se daná entita musí nebo nemusí účastnit daného vztahu. Parcialita 0 znamená „musí“, parcialita 1 znamená „může ale nemusí“.

4.5. Klíče

Klíče jsou speciálními poli, která hrají v tabulce velmi významnou roli. Typ klíče určuje jeho význam v tabulce. Rozeznávají se čtyři typy klíčů: kandidáti na klíč, primární klíče, cizí klíče a neklíče. Nejvýznamnější jsou primární klíč a cizí klíč. Slouží jako prostředek pro vytvoření vztahů.

Klíče jsou důležité, protože zabezpečují, že každý záznam v tabulce je přesně identifikován. Tabulka reprezentuje soubor podobných událostí či objektů. Soubor tvoří všechny záznamy v tabulce a každý záznam reprezentuje jedinečnou instanci entity tabulky v rámci celého souboru. K přesné identifikaci jednotlivé instance, je třeba klíč. [5]

Napomáhají stanovení a dodržení různých druhů integrity. Klíče jsou zásadní součástí integrity na úrovni tabulek a na úrovní vztahů. Umožňují zabezpečit, že tabulka obsahuje jedinečné záznamy a že hodnoty polí, která se použijí pro vytvoření vztahu mezi tabulkami, si vždy navzájem odpovídají. [5]

4.5.1. Kandidátní klíč

Má-li být splněna podmínka, že každý člen množiny je jedinečný, musí u každé relace existovat určitá kombinace atributů, která jednoznačně identifikuje každý jednotlivý vektor souřadnic. Tato množina jednoho nebo více atributů se nazývá kandidátní klíč. Kandidátní klíč musí být schopen v jakémkoli daném okamžiku jednoznačně identifikovat všechny možné vektory hodnot, nikoli pouze vektory hodnot z jisté vybrané množiny. Kandidátní klíč nelze zjistit pohledem na konkrétní množinu dat. Pokud je určité pole pro danou množinu vektorů hodnot jedinečné, neznamená to, že je vhodným kandidátním klíčem, není žádná záruka, že bude jedinečné pro všechny vektory hodnot. Každá relace musí mít alespoň jeden kandidátní klíč. Triviálním kandidátním klíčem je množina všech atributů, ze kterých se skládá vektor souřadnic. Kandidátní klíče se skládají buď z jediného atributu (jednoduchý klíč), nebo z více atributů (složený klíč). [5]

4.5.2. Primární klíč

Primární klíč je atribut, nebo soustava atributů, jejichž hodnoty tvoří jednoznačnou identifikaci řádku relace. Každá relace musí obsahovat primární klíč, v nejhorším případě jím jsou všechny atributy. Je-li primární klíč složen ze dvou, nebo více polí, tak se označuje jako složený primární klíč. Primární klíč je absolutně nejdůležitějším klíčem v celé tabulce. [9]

Hodnota primárního klíče identifikuje daný záznam v celé databázi. Pole primárního klíče identifikuje danou tabulku v rámci celé databáze. Primární klíč vynucuje integritu na rovni tabulky a pomáhá zřizovat vztahy s ostatními tabulkami v databázi.

Vlastnosti primárního klíče [9]:

- nesmí to být vícesložkové pole
- musí obsahovat jedinečné hodnoty
- nesmí obsahovat hodnoty null
- jeho hodnota nesmí zapříčinit únik informací, které jsou dány bezpečnostními pravidly organizace
- hodnota primárního klíče není ani částečně, ani jako celek volitelná
- skládá se z nejmenšího možného počtu polí, která zajišťují jedinečnost
- jeho hodnota musí být v rámci celé tabulky jedinečná
- hodnota primárního klíče musí jednoznačně identifikovat hodnoty všech polí daného záznamu
- všechny primární klíče v databázi musí být jedinečné – žádné dvě tabulky nesmí sdílet primární klíč, pokud se nejedná o podmnožinové tabulky
- jeho hodnota se změní jen ve velmi výjimečných případech

4.5.3. Cizí klíč

Zjistí-li se, že dvě tabulky by k sobě měly mít vztah, zřídí se mezi nimi vztah většinou tak, že se vezme kopie primárního klíče první tabulky a začlení se do druhé tabulky, kde se stane cizím klíčem. Jméno „cizí klíč“ je odvozeno od faktu, že druhá tabulka už má svůj primární klíč a že primární klíč první tabulky je z pohledu druhé tabulky „cizí“. Kromě pomoci se zřizováním vztahu mezi dvojicemi tabulek pomáhá cizí klíč také implementovat a zajišťovat integritu na úrovni vztahů. Tedy záznamy v obou tabulkách jsou vždy ve správném vztahu, protože hodnota cizího klíče musí odpovídat hodnotě primárního klíče, na který se odkazuje. Integrita na úrovni vztahů také předchází obávaným sirotčím záznamům. [9]

4.5.4. Alternativní klíče

Když se u jednotlivých tabulek označí některý kandidát na klíč za primární klíč, stanou se zbylí kandidáti na klíč alternativními klíči. Použije-li se některý z alternativních klíčů, ve struktuře tabulky se označí značkou „AK“ nebo „CAK“, v opačném případě typ tohoto pole se převede z alternativního klíče na běžné pole. [9]

5. PŘEVOD ERD DO RELAČNÍHO MODELU DAT

5.1. Transformace E-R digramu do relačního modelu dat

Proces návrhu diagramu toků dat se dá chápat jako transformace diagramů od počátečního ke koncovému. Přitom se transformují jak jména použitá v počátečním diagramu, tak i logické spoje definované pro logické objekty použité v počátečním diagramu. Použití transformací je řízeno zásadou docílit zjemnění, ale současně i funkční nezávislosti jednotlivých objektů diagramu toků dat. [4, 5]

Kardinalita 1:1

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 1;1 – 1;1, z těchto dvou entit se stane jedna entita s jedním primárním klíčem. V tomto případě je jedno, který z primárních klíčů to bude. [4]

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 0;1 – 0;1, k těmto dvěma entitám přibude třetí entita s dvěma atributy, původními primárními klíči. Jeden z těchto cizích klíčů bude primárním klíčem. V tomto případě je jedno, který z původních primárních klíčů to bude. [4]

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 0;1 – 1;1 zůstávají nám tyto dvě entity. K entitě s parcialitou 0 přidá se jako cizí klíč primární klíč druhé entity. [4]

Kardinalita 1:N

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 0;1 – x;N, ke dvěma entitám přibude třetí entita s jedním primárním klíčem. Primární klíč této nové entity bude primární klíč entity s parcialitou 0, jako cizí klíč nové entity bude primární klíč entity s parcialitou x. [4]

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 1;1 – x;N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá jako cizí klíč primární klíč druhé entity. [4]

Kardinalita N:M

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou x;N – x;M vzniká třetí entita. Tato entita má složený primární klíč, složený z obou primárních klíčů. [4]

5.2. Normalizace

Normalizace je proces rozkladu velkých tabulek na menší, což slouží k eliminaci redundantních a duplicitních dat a k eliminaci budoucích problémů s vkládáním, rušením a aktualizací dat. Během procesu normalizace jsou tabulky porovnávány s tzv. normálními formami a poté, je-li nalezen některý z výše zmíněných problémů, modifikovány. Normální forma je specifická množina pravidel, která se používají k otestování struktury tabulek a k ověření, že tato struktura je solidní a bezchybná. Existuje celá řada normálních forem a každá z nich se používá k testování určitých specifických problémů. V současné době se používá především první, druhá, třetí a dále čtvrtá a pátá normální forma, Boyce-Coddova normální forma a doménově klíčová normální forma. [9]

5.2.1. První normální forma

Relace je v první normální formě, jestliže jsou všechny její atributy definovány nad doménami (skalárními obory hodnot). Každý atribut vektoru hodnot musí obsahovat jen jednu jednotlivou, skalární hodnotu. Jinými slovy, každý atribut musí obsahovat pouze jednu hodnotu. [5]

5.2.2. Druhá normální forma

Relace je ve druhé normální formě, právě když je v první normální formě a zároveň jsou všechny její atributy závislé na celém primárním klíči. Jde o požadavek nevyjadřovat dvě různé entity v jedné relaci. Pokud se obě tyto entity od sebe oddělí, zbaví se tím nejen nežádoucí redundance, ale definuje se také mechanismus pro uložení informací. [5]

5.2.3. Třetí normální forma

Relace je ve třetí normální formě, právě když je ve druhé normální formě a zároveň jsou všechny její neklíčové atributy vzájemně nezávislé. To, zda a jak třetí normální formu implementovat závisí na sémantice daného modelu. Samostatná entita se vytvoří jen tehdy, když je nová entita pro datový model významná nebo pokud se její data budou často měnit. [5]

5.2.4. Další normální formy

První tři normální formy byly součástí původní Coddovy formulace relační teorie. Boyce/Coddova, čtvrtá a pátá normální forma slouží pro speciální případy.

Boyce/Coddova normální forma se pokládá za jistou variaci třetí normální formy. Používá se pro speciální případy relací s více kandidátními klíči. Boyce/Coddovu normální formu lze přitom aplikovat pouze při splnění následujících podmínek [5]:

- relace musí mít dva nebo více kandidátních klíčů
- nejméně dva z kandidátních klíčů musí být složené
- kandidátní klíče se v některých atributech musí překrývat
- mezi kandidátními klíči nesmí být žádná funkční závislost

Čtvrtá normální forma znamená, že v jedné relaci se nesmí spojovat nezávislé opakované skupiny.

Pátá normální forma se týká spojené závislosti vyjadřující cyklické omezení.

6. DATABÁZOVÝ SYSTÉM MICROSOFT ACCESS

6.1. Systém pro správu relační databáze – Microsoft Access

Systém pro správu relační databáze (RDBMS) umožňuje úplnou kontrolu nad tím, jak definovat data, pracovat s nimi a sdílet je s ostatními. Takovýto systém zabezpečuje i důmyslné funkce, které usnadňují uspořádání a správu rozsáhlých objemů dat v mnoha tabulkách. RDBMS má tři hlavní typy funkcí: definice dat, manipulace s daty a řízení dat. V Microsoft Access jsou obsaženy všechny tři funkce.

Vše, co může mít nějaké jméno se nazývá objektem. V rámci databáze Access jsou hlavními objekty tabulky, dotazy, formuláře a sestavy. [13]

6.1.1. Tabulka

Objekt, který se definuje a používá pro ukládání dat. Každá tabulka obsahuje konkrétní informace o předmětu. Tabulky obsahují pole (neboli sloupce), které uchovávají různé typy dat, a záznamy (neboli řádky), které uchovávají informace o konkrétní instanci daného předmětu. Nad každou tabulkou se může definovat primární klíč (jedno nebo více polí) a jeden nebo více indexů, které přispívají k rychlejšímu výběru dat. [13]

6.1.2. Dotaz

Objekty, který zajišťuje uživatelský pohled na data z jedné nebo více tabulek. Dotaz můžeme definovat pro vytvoření nové tabulky z existujících tabulek, aktualizaci, výběr, vkládání nebo odstraňování dat. [13]

6.1.3. Formulář

Objekt, který je navržen hlavně pro vstup nebo zobrazení dat. Pomocí formulářů se může uživatelsky přizpůsobit pohled na data, která aplikace vybírá z dotazů nebo tabulek. [13]

6.1.4. Sestava

Objekt navržený pro formátování, výpočty, tisk a souhrny vybraných dat. [13]

7. KONCEPTUÁLNÍ ANALÝZA VYBRANÉ REÁLIE

Obchodní firma s licencí získanou od Ministerstva zdravotnictví ČR se zabývá distribucí léčiv. Dodavatelé pochází jak z tuzemska, tak i ze zahraničí. Firma má řadu zaměstnanců, kteří se zabývají účetnictví, skladováním, dopravou a dalšími činnostmi souvisejícími s touto podnikatelskou činností. To znamená, že ne všichni zaměstnanci jednají s dodavateli nebo odběrateli. Odběratelé jsou tuzemské lékárny, nemocnice a zdravotní a sociální ústavy. Účetnictví je vedeno odděleně s možností použití dat z databáze.

7.1. Určení entit

Nejprve je nutné určit entity,které popíší jednotlivé objekty reality (ty objekty, o kterých má smysl uchovávat nějakou informaci):

- Zaměstnanci
- Dodavatelé
- Odběratelé
- Léky

7.2. Určení atributů

Navrhovaný systém o každé entitě zaznamenává a sleduje určité skutečnosti, těmto údajům se říká atributy dané entity.

Tabulka 1 Určení atributů

Zaměstnanci	(číslo zaměstnance, datum narození, jméno, příjmení, ulice a čp, PSČ, oddělení, telefon)
Dodavatelé	(číslo dodavatele, kód země dodavatele, název dodavatele, ulice a čp, PSČ, bankovní spojení)
Odběratelé	(IČ odběratele, název odběratele, ulice a čp, PSČ, bankovní spojení)
Léky	(kód léku, skupina léčiv, název, šarže, balení, cena léku)

K daným entitám je nutné přiřadit atributy, samozřejmě by se daly další atributy přidat nebo naopak některé vynechat. Vždy záleží na konkrétních požadavcích uživatele, dle toho co ho bude zajímat.

7.3. Určení primárních klíčů

Primární klíč je atribut, nebo soustava atributů, jejichž hodnoty tvoří jednoznačnou identifikaci řádku relace. Každá relace musí obsahovat primární klíč. Je-li primární klíč složen ze dvou, nebo více polí, označuje se jako složený primární klíč. Primární klíč je nejdůležitějším klíčem v celé relaci.

U entity zaměstnanci je zvolen jako primární klíč atribut číslo zaměstnance. Toto číslo bude vždy unikátní. Každý nově příchozí zaměstnanec získá své osobní číslo odpovídající pořadí nástupu do firmy.

U entity léky je zvolen jako primární klíč atribut kód léku. Kód léku je číslo mezinárodně platné pro označení konkrétního léčiva, proto je nejvhodnějším kandidátem na primární klíč.

U entity odběratelé je stanoven jako primární klíč atribut IČ odběratele. Všichni odběratelé mají své identifikační číslo, které je jedinečné a dohledatelné v rejstříku.

U entity dodavatelé je zvolen složený primární klíč z atributů číslo dodavatele a kód země. Kód země dodavatele představuje zkrácené označení pro zemi dodavatele. Číslo dodavatele představuje pořadí dodavatele v dané zemi.

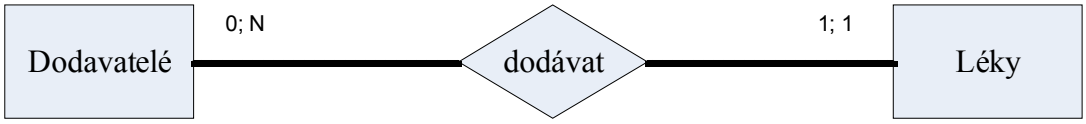
Tabulka 2 Určení primárních klíčů

Zaměstnanci	(číslo zaměstnance, datum narození, jméno, příjmení, ulice a čp, PSČ, oddělení, telefon)
Dodavatelé	(číslo dodavatele, kód země, název dodavatele, ulice a čp, PSČ, bankovní spojení)
Odběratelé	(IČ odběratele, název odběratele, ulice a čp, PSČ, bankovní spojení)
Léky	(kód léku, skupina léčiv, název, šarže, balení, cena léku)

7.4. Určení vztahů mezi entitami

V datovém modelu je nutné určit vztahy definované mezi různými entitami. Na konceptuální úrovni jsou vztahy určitými asociacemi mezi entitami.

Dodavatel může (tzn. v databázi bude uveden i dodavatel, který ještě žádné léky nedodal) dodávat jeden nebo i více druhů léků. Jeden konkrétní lék musí být vždy dodán jedním konkrétním dodavatelem.

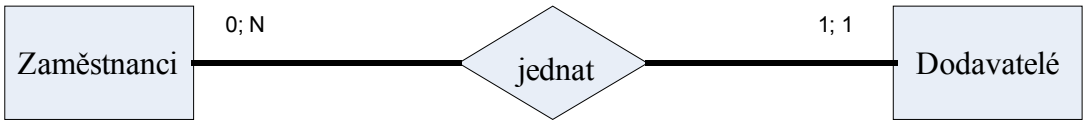


Obrázek 2 ERD Dodavatelé - Léky

Odběratel může odebírat jeden nebo i více druhů léků. Jeden konkrétní lék může být odebírán jedním nebo i více odběrateli.

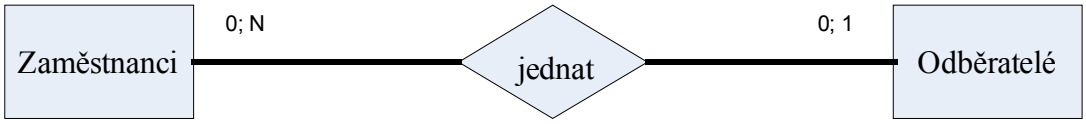
Obrázek 3 ERD Odběratelé – Léky

Zaměstnanec může jednat s jedním nebo i více dodavateli. Jeden konkrétní dodavatel musí jednat právě s jedním zaměstnancem. Jednat znamená objednávat, dojednávat obchod s léky.



Obrázek 4 ERD Zaměstnanci – Odběratelé

Zaměstnanec může jednat s jedním nebo i více odběrateli. Jeden konkrétní odběratel může jednat právě s jedním konkrétním zaměstnancem.

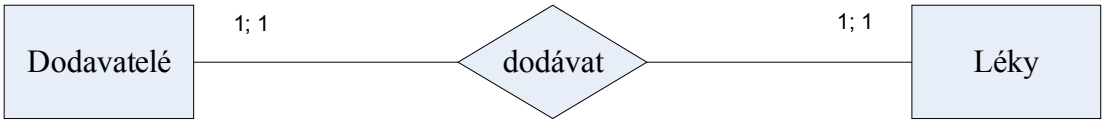


Obrázek 5 ERD Zaměstnanci - Odběratelé

7.4.1. Ukázka dalších vztahů

Vztah 1; 1 - 1; 1 by byl vyjádřen v případě, kdy by každý dodavatel mohl dodávat pouze jeden lék a byla sjednána smlouva na pravidelné dodávky, proto parcialita 1.

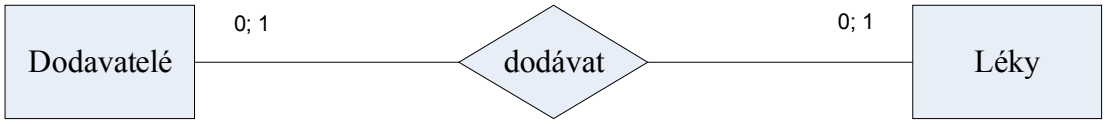
Dodavatel musí dodávat právě jeden konkrétní lék. Lék musí dodávat právě jeden dodavatel.



Obrázek 6 ERD Dodavatelé – Léky

Vztah 0; 1 – 0; 1 by byl vyjádřen v případě, kdy by dodavatel mohl dodávat právě jeden lék, to znamená že jiné léky by nedodával. Lék může být dodán vždy jen jedním dodavatelem, ale připouští se, že lék nemusí být vyráběn pouze jedním dodavatelem, například při různých licenčních smlouvách.

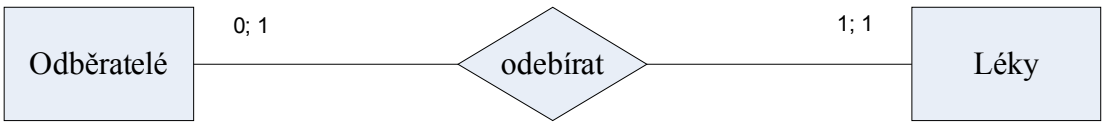
Dodavatel může dodávat právě jeden konkrétní lék. Lék může být dodán právě jedním dodavatelem.



Obrázek 7 ERD Dodavatelé – Léky

Vztah 0; 1 – 1; 1 by byl vyjádřen v případě, kdy by odběratelem bylo specializované pracoviště, které by potřebovalo pouze jeden speciální lék, který není nikam jinam prodáván. Lék jako takový je jinak nedostupný.

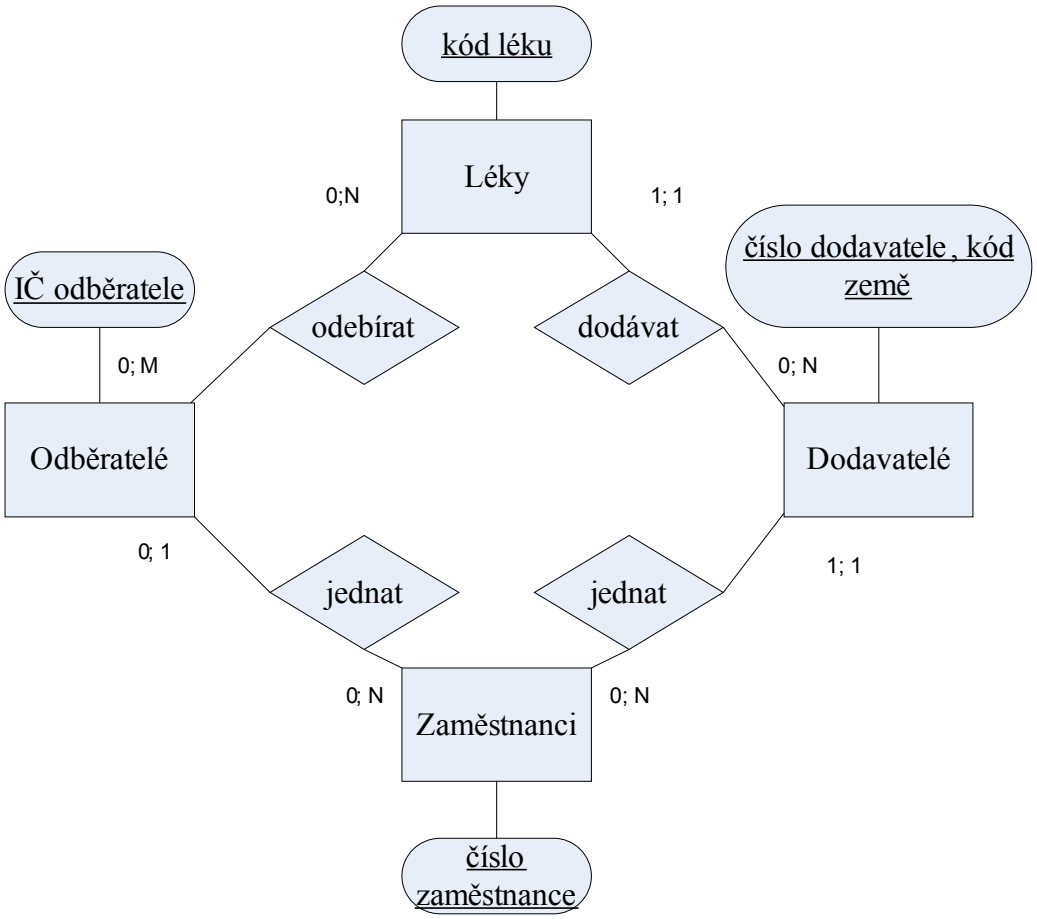
Odběratel může odebírat pouze jeden lék. Lék musí být odebírán právě jedním konkrétním odběratelem.



Obrázek 8 ERD Odběratelé - Léky

7.5. ER diagram

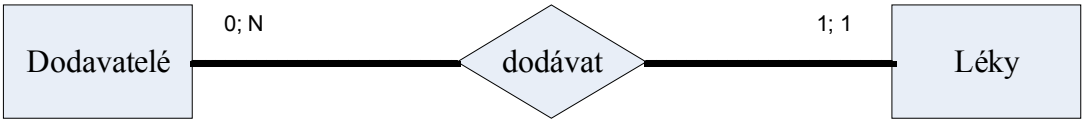
Model entit a vztahů popisující data jako entity, atributy a vztahy. V ER diagramech se entity popisují pomocí obdélníků, atributy se vyjadřují jako elipsy (ovály) a vztahy se znázorňují pomocí kosočtverců. [4, 5]



Obrázek 9 ER diagram

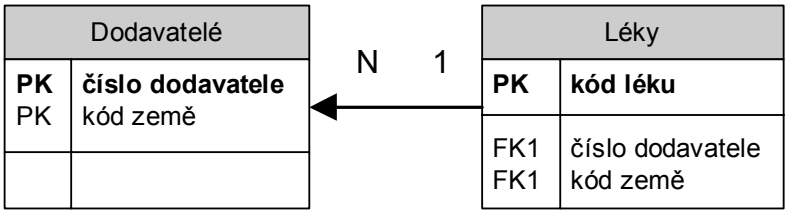
7.6. Transformace ERD do RMD

Proces návrhu diagramu toků dat se dá chápat jako transformace diagramů od počátečního ke koncovému. Transformují se jednak jména použitá v počátečním diagramu, jednak logické spoje pro logické objekty použité v počátečním diagramu. Použití jednotlivých transformací je řízeno zásadou docílit zjemnění, ale současně i funkční nezávislosti jednotlivých objektů diagramu toků dat. [9]



Obrázek 10 ERD Dodavatelé – Léky

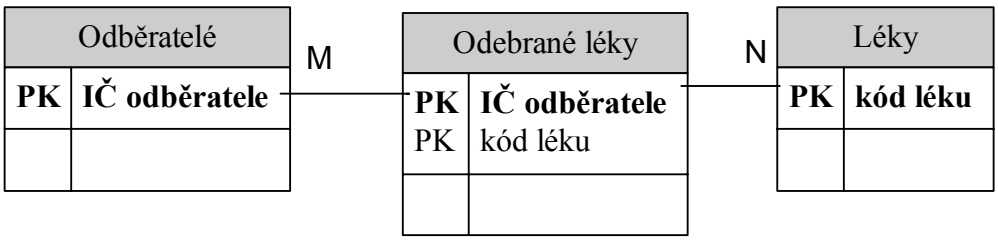
Pokud je dán vztah mezi dvěma entitami s kardinalitou 1; 1 – x; N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá jako cizí klíč primární klíč druhé entity.



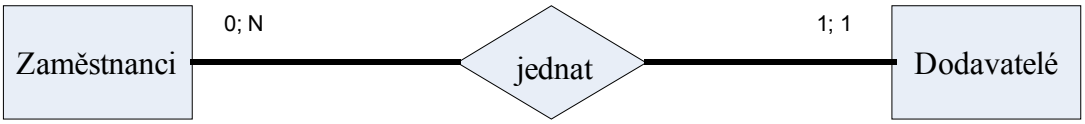
Obrázek 11 Návrh RMD Dodavatelé – Léky

Obrázek 12 ERD Odběratelé – Léky

Pokud je dán vztah mezi dvěma entitami s parcialitou a kardinalitou x; M – x; N vzniká třetí entita. Tato entita má spojený primární klíč, složený z obou primárních klíčů.

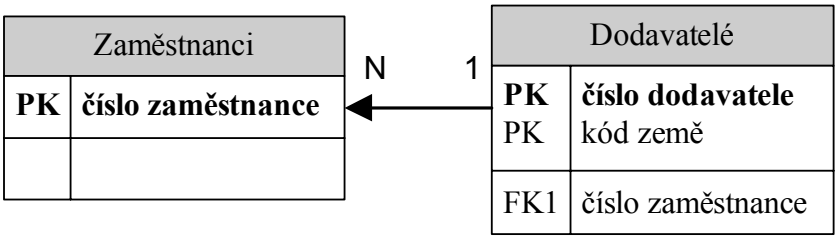


Obrázek 13 Návrh RMD Odběratelé – Léky

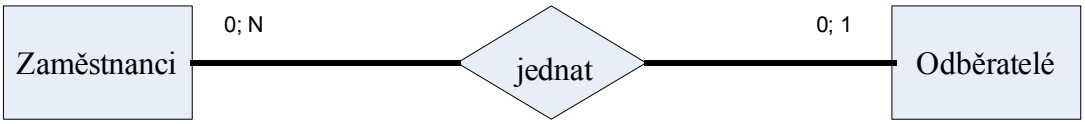


Obrázek 14 ERD Zaměstnanci – Dodavatelé

Pokud je dán vztah mezi dvěma entitami s parcialitou a kardinalitou 1; 1 – x; N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá jako cizí klíč primární klíč druhé entity.

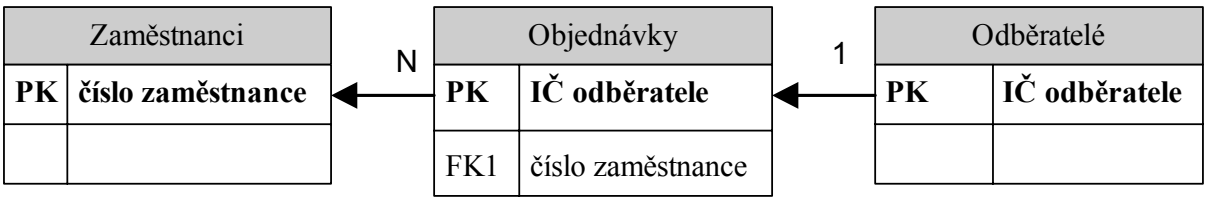


Obrázek 15 Návrh RMD Zaměstnanci – Dodavatelé



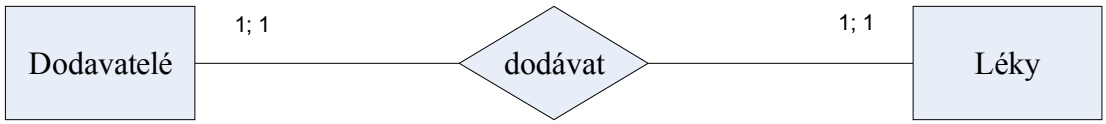
Obrázek 16 ERD Zaměstnanci – Odběratelé

Pokud je dán vztah mezi dvěma entitami s parcialitou a kardinalitou 0; 1 – x; N, ke dvěma entitám přibude třetí entita s dvěma cizími klíči, původně primárními klíči. Primární klíč této entity bude primární klíč entity s parcialitou 0, cizí klíč bude primární klíč entity s parcialitou x.



Obrázek 17 Návrh RMD Zaměstnanci – Odběratelé

7.6.1. Ukázka další transformace ERD do RMD

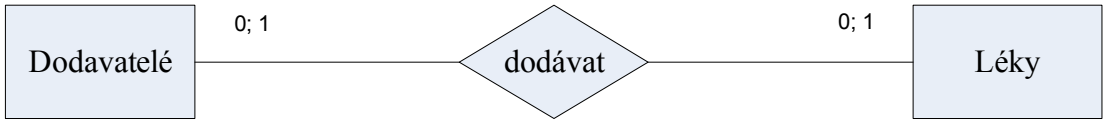


Obrázek 18 ERD Dodavatelé – Léky

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 1; 1 – 1; 1, z těchto dvou entit se stane jedna entita s jedním primárním klíčem a jedním cizím klíčem. V tomto případě je jedno, který z primárních klíčů to bude.

Dodavatelé léků	
PK PK	**číslo dodavatele** kód země
	kód léku

Obrázek 19 Návrh RMD Dodavatelé – Léky



Obrázek 20 ERD Dodavatelé – Léky

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 0; 1 – 0; 1, k těmto dvěma entitám přibude třetí entita s dvěma cizími klíči, původními primárními klíči. Jeden z těchto cizích klíčů bude primárním klíčem. V tomto případě je jedno, který z původních primárních klíčů to bude.

Dodavatelé	
PK PK	**číslo dodavatele** kód země

1

Dodavatelé léků	
PK	**kód léku**
FK1 FK1	číslo dodavatele kód země

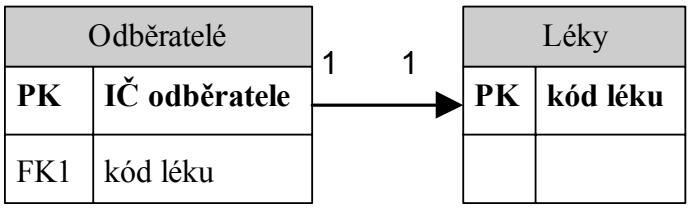
1

Léky	
PK	**kód léku**

Obrázek 21 Návrh RMD Dodavatelé – Léky

Obrázek 22 ERD Odběratelé – Léky

Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 0; 1 – 1; 1 zůstávají tyto dvě entity. K entitě s parcialitou 0 se přidá jako cizí klíč primární klíč druhé entity.



Obrázek 23 Návrh RMD Odběratelé - Léky

7.7. Normalizace

Relace je v první normální formě, jestliže jsou všechny její atributy definovány nad doménami (skalárními obory hodnot). Každý atribut vektoru hodnot musí obsahovat jen jednu jednotlivou, skalární hodnotu. Jinými slovy, každý atribut musí obsahovat pouze jednu hodnotu. [5]

Tabulka 3 Zaměstnanci 1. NF

	Zaměstnanci (číslo zaměstnance, datum narození, jméno zaměstnance, adresa zaměstnance, oddělení, telefon)
1NF	Zaměstnanci (číslo zaměstnance, datum narození, jméno, příjmení, ulice, čp, město, PSČ, oddělení, telefon)

Atribut adresa zaměstnance se rozpadá na atributy ulice a čp, město a PSČ.

Relace je ve druhé normální formě, právě když je v první normální formě a zároveň jsou všechny její atributy závislé na celém primárním klíči. Jde o požadavek nevyjadřovat dvě různé entity v jedné relaci. Pokud se obě tyto entity od sebe oddělí, zbaví se tím nejen nežádoucí redundance, ale definuje se také mechanismus pro uložení informací. [5]

Tabulka 4 Zaměstnanci 2. NF

	Zaměstnanci (číslo zaměstnance, datum narození, jméno zaměstnance, adresa zaměstnance, oddělení, telefon)
1NF	Zaměstnanci (číslo zaměstnance, datum narození, jméno, příjmení, ulice, čp, město, PSČ, oddělení, telefon)
2NF	Zaměstnanci (číslo zaměstnance, datum narození, jméno, příjmení, ulice, čp, město, PSČ, oddělení, telefon)

Relace je ve třetí normální formě, právě když je ve druhé normální formě a zároveň jsou všechny její neklíčové atributy vzájemně nezávislé. To, zda a jak třetí normální formu implementovat závisí na sémantice daného modelu. Samostatná entita se vytvoří jen tehdy, když je nová entita pro datový model významná nebo pokud se její data budou často měnit. [5]

Tabulka 5 Zaměstnanci 3. NF

	Zaměstnanci (číslo zaměstnance, datum narození, jméno zaměstnance, adresa zaměstnance, oddělení, telefon)
1NF	Zaměstnanci (číslo zaměstnance, datum narození, jméno, příjmení, ulice, čp, město, PSČ, oddělení, telefon)
2NF	Zaměstnanci (číslo zaměstnance, datum narození, jméno, příjmení, ulice, čp, město, PSČ, oddělení, telefon)
3NF	Zaměstnanci (číslo zaměstnance, datum narození, jméno, příjmení, ulice, čp, PSČ, oddělení, telefon) Pošta (PSČ, město)

Ukazuje se, že atribut město je závislý na atributu PSČ. PSČ proto v entitě zaměstnanci zůstává a naopak atribut město se přesune do nově vzniklé relace Pošta, kde PSČ představuje primární klíč.

Relace je v první normální formě, jestliže jsou všechny její atributy definovány nad doménami (skalárními obory hodnot). Každý atribut vektoru hodnot musí obsahovat jen jednu jednotlivou, skalární hodnotu. Jinými slovy, každý atribut musí obsahovat pouze jednu hodnotu. [5]

Tabulka 6 Dodavatelé 1. NF

	Dodavatelé (číslo dodavatele, kód země, název dodavatele, adresa dodavatele, bankovní spojení, číslo zaměstnance)
1NF	Dodavatelé (číslo dodavatele, kód země, název dodavatele, ulice, čp, město, PSČ, země, bankovní spojení, číslo zaměstnance)

Atribut adresa se rozpadá na atributy ulice a čp, město a PSČ.

Relace je ve druhé normální formě, právě když je v první normální formě a zároveň jsou všechny její atributy závislé na celém primárním klíči. Jde o požadavek nevyjadřovat dvě různé entity v jedné relaci. Pokud se obě tyto entity od sebe oddělí, zbaví se tím nejen nežádoucí redundance, ale definuje se také mechanismus pro uložení informací. [5]

Tabulka 7 Dodavatelé 2. NF

	Dodavatelé (číslo dodavatele, kód země, název dodavatele, adresa dodavatele, bankovní spojení, číslo zaměstnance)
1NF	Dodavatelé (číslo dodavatele, kód země, název dodavatele, ulice, čp, město, PSČ, země, bankovní spojení, číslo zaměstnance)
2NF	Dodavatelé (číslo dodavatele, kód země, název dodavatele, ulice, čp, město, PSČ, bankovní spojení, číslo zaměstnance) Stát (kód země dodavatele, země)

Ukazuje se, že atribut země není závislý na celém primárním klíči, ale pouze na kódu země dodavatele. Atribut země se proto přesune do nově vzniklé relace Stát, kde bude primárním klíčem kód země dodavatele.

Relace je ve třetí normální formě, právě když je ve druhé normální formě a zároveň jsou všechny její neklíčové atributy vzájemně nezávislé. To, zda a jak třetí normální formu implementovat závisí na sémantice daného modelu. Samostatná entita se vytvoří jen tehdy, když je nová entita pro datový model významná nebo pokud se její data budou často měnit. [5]

Tabulka 8 Dodavatelé 3. NF

	Dodavatelé (číslo dodavatele, kód země, název dodavatele, adresa dodavatele, bankovní spojení, číslo zaměstnance)
1NF	Dodavatelé (číslo dodavatele, kód země, název dodavatele, ulice, čp, město, PSČ, země, bankovní spojení, číslo zaměstnance)
2NF	Dodavatelé (číslo dodavatele, kód země, název dodavatele, ulice, čp, město, PSČ, bankovní spojení, číslo zaměstnance) Stát (kód země dodavatele, země)
3NF	Dodavatelé (číslo dodavatele, kód země, název dodavatele, ulice, čp, PSČ, bankovní spojení, číslo dodavatele) Stát (kód země, země) Pošta (PSČ, město)

Atribut město je závislý na atributu PSČ. PSČ proto v entitě Zaměstnanci zůstává a odstraní se atribut město. Zároveň vzniká nová entita Pošta, kde PSČ představuje primární klíč a přesouvá se sem atribut město.

Tabulka 9 Normalizace entity Odběratelé

	Odběratelé (IČ odběratele, název odběratele, adresa odběratele, bankovní spojení)
1NF	Odběratelé (IČ odběratele, název odběratele, ulice, čp, město, PSČ, bankovní spojení)
2NF	Odběratelé (IČ odběratele, název odběratele, ulice, čp, město, PSČ, bankovní spojení)
3NF	Odběratelé (IČ odběratele, název odběratele, ulice, čp, PSČ, bankovní spojení) Pošta (PSČ, město)

Tabulka 10 Normalizace entity Léky

	Léky (kód léku, skupina léčiv, název, balení, cena léku, číslo dodavatele, kód země)
1NF	Léky (kód léku, skupina léčiv, název, balení, cena léku, číslo dodavatele, kód země)
2NF	Léky (kód léku, skupina léčiv, název, balení, cena léku, číslo dodavatele, kód země)
3NF	Léky (kód léku, skupina léčiv, název, balení, cena léku, číslo dodavatele, kód země)

Tabulka 11 Přehled všech entit

Zaměstnanci	(číslo zaměstnance, datum narození, jméno, příjmení, ulice, čp, PSČ, oddělení, telefon)
Dodavatelé	(číslo dodavatele, kód země, název dodavatele, ulic, čp, PSČ, bankovní spojení, číslo zaměstnance)
Odběratelé	(IČ odběratele, název odběratele, ulice, čp, PSČ, bankovní spojení)
Léky	(kód léku, skupina léčiv, název, balení, cena léku, číslo dodavatele, kód země)
Pošta	(PSČ, město)
Stát	(kód země, země)
Objednávka	(IČ odběratele, číslo zaměstnance)
Odebrané léky	(kód léku, IČ odběratele)

8. IMPLIKACE DO DATABÁZOVÉHO SYSTÉMU MICROSOFT ACCESS

Nyní je nutné entity s atributy importovat do databázového systému a vytvořit strukturu databáze.

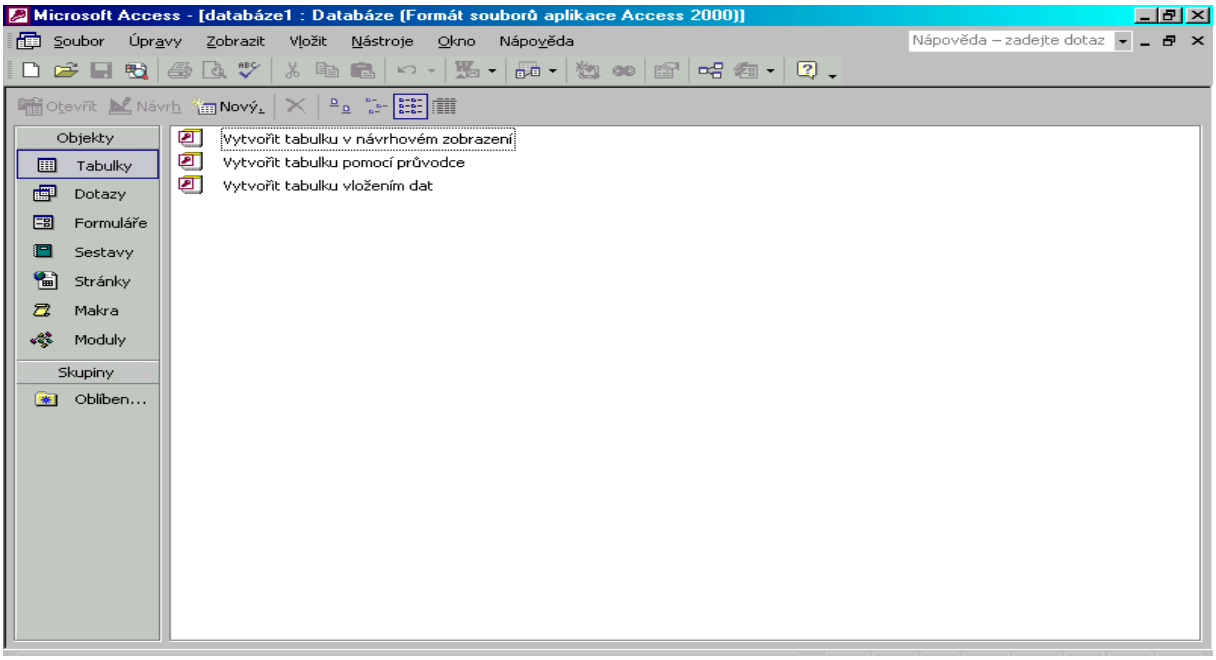
8.1. Vytvoření databáze

Po otevření Microsoft Access je nabídnuto otevřít stávající databázi, nebo vytvořit novou databázi. Nejprve se musí nově vytvořená databáze pojmenovat.

8.2. Vytvoření tabulek

Tabulky se dají vytvořit trojím způsobem:

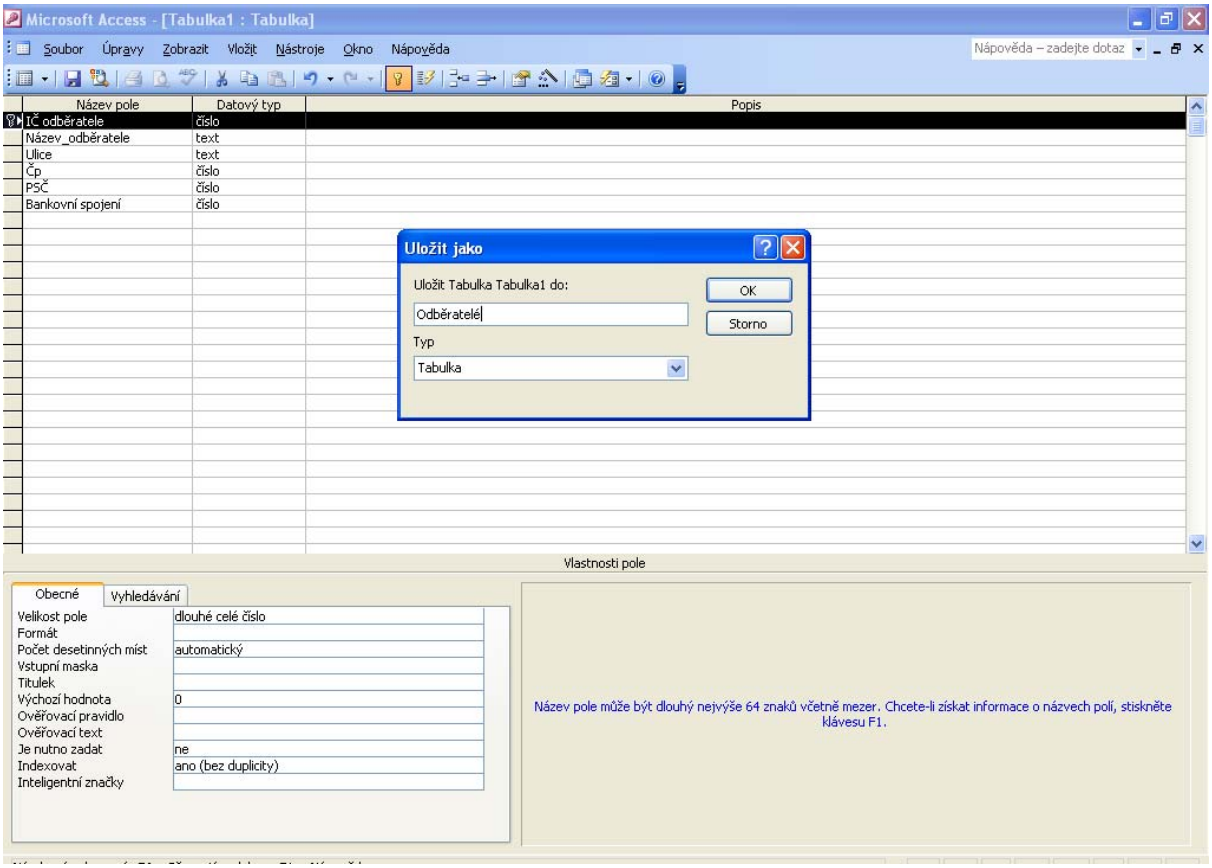
- pomocí průvodce
- vložením dat
- v návrhovém zobrazení



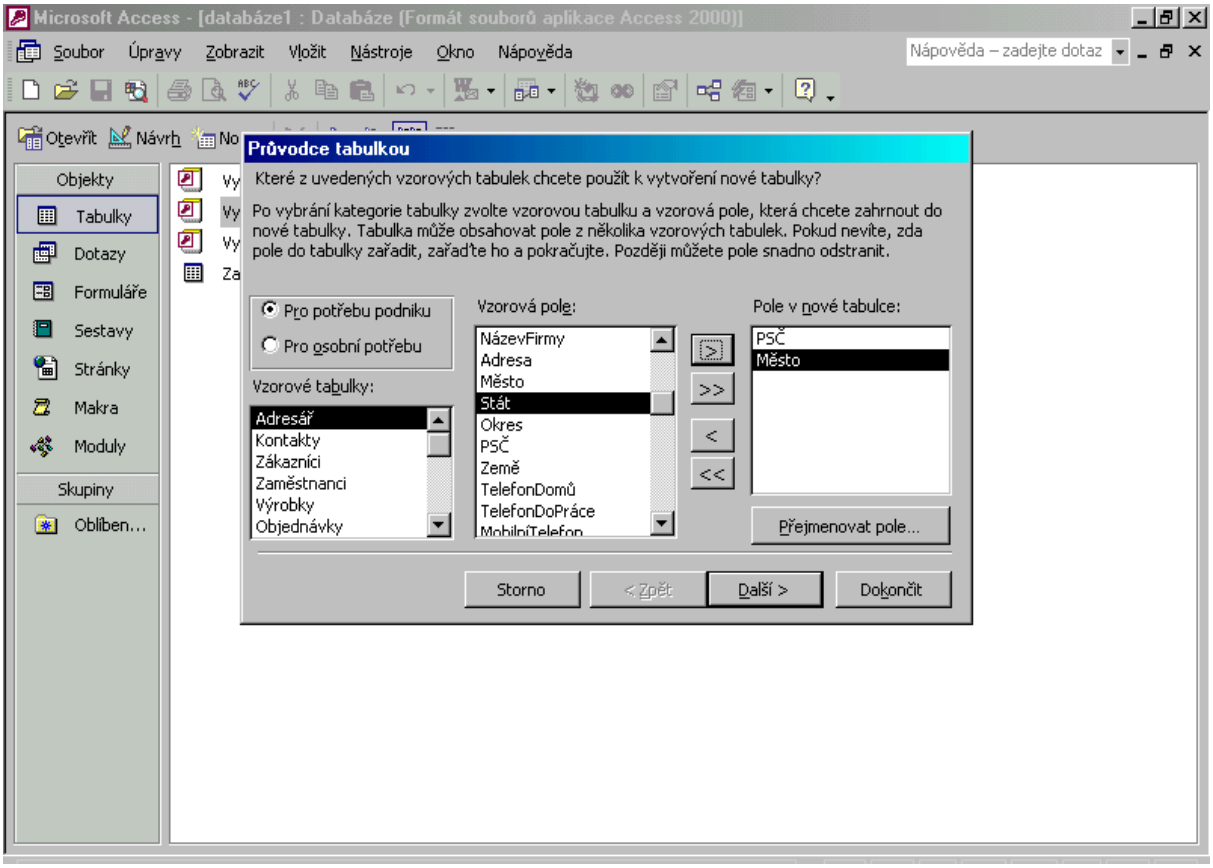
Obrázek 24 Vytvoření tabulky

8.2.1. Vytvoření tabulky pomocí návrhového zobrazení

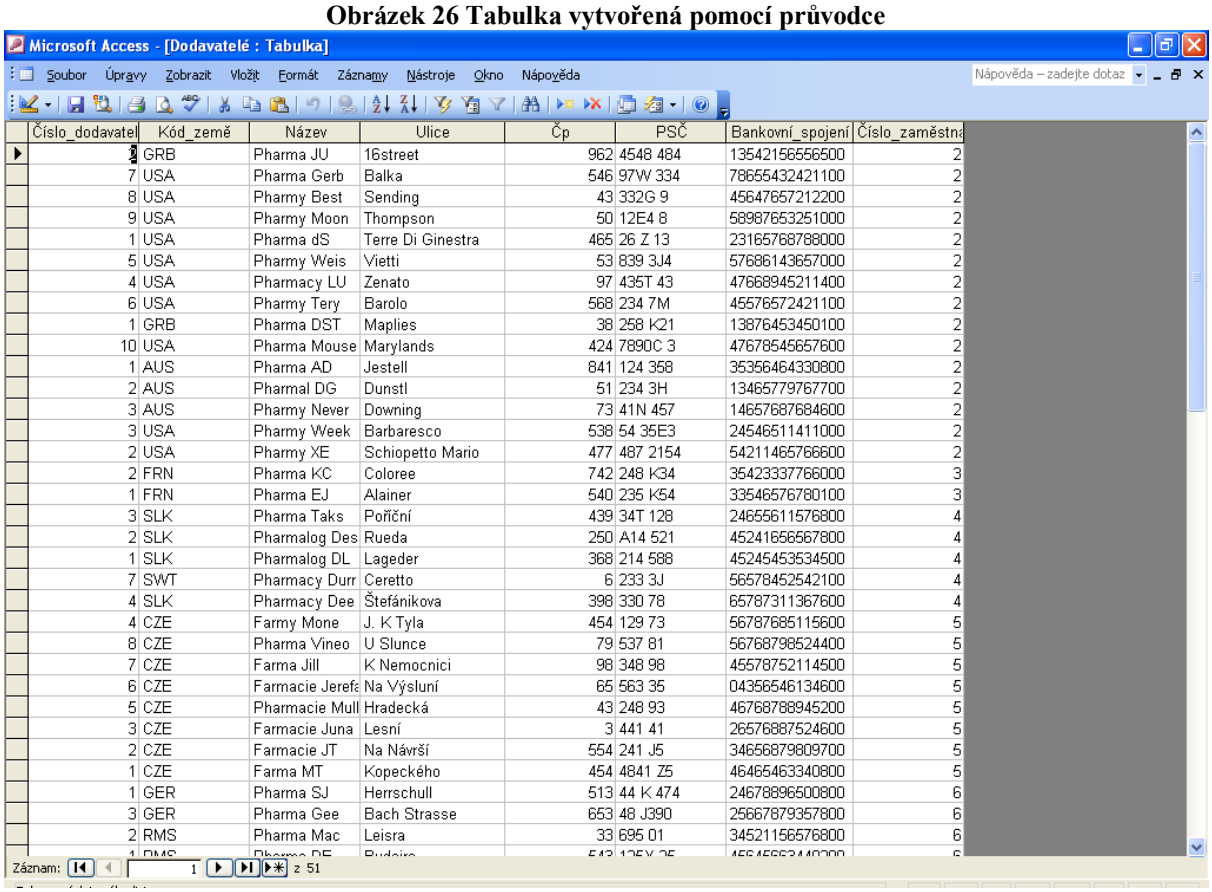
U každého pole tabulky je nutné určit jeho datový typ. Datový typ může být Text, Memo, Číslo, Datum a čas, Měna, Automatické číslo, Ano/ne, Objekt OLE nebo Hypertextový odkaz. V obecných vlastnostech se nechá určit velikost pole podle zvoleného datového typu, dále využít Formát, Vstupní maska, Titulek nebo Indexovat.



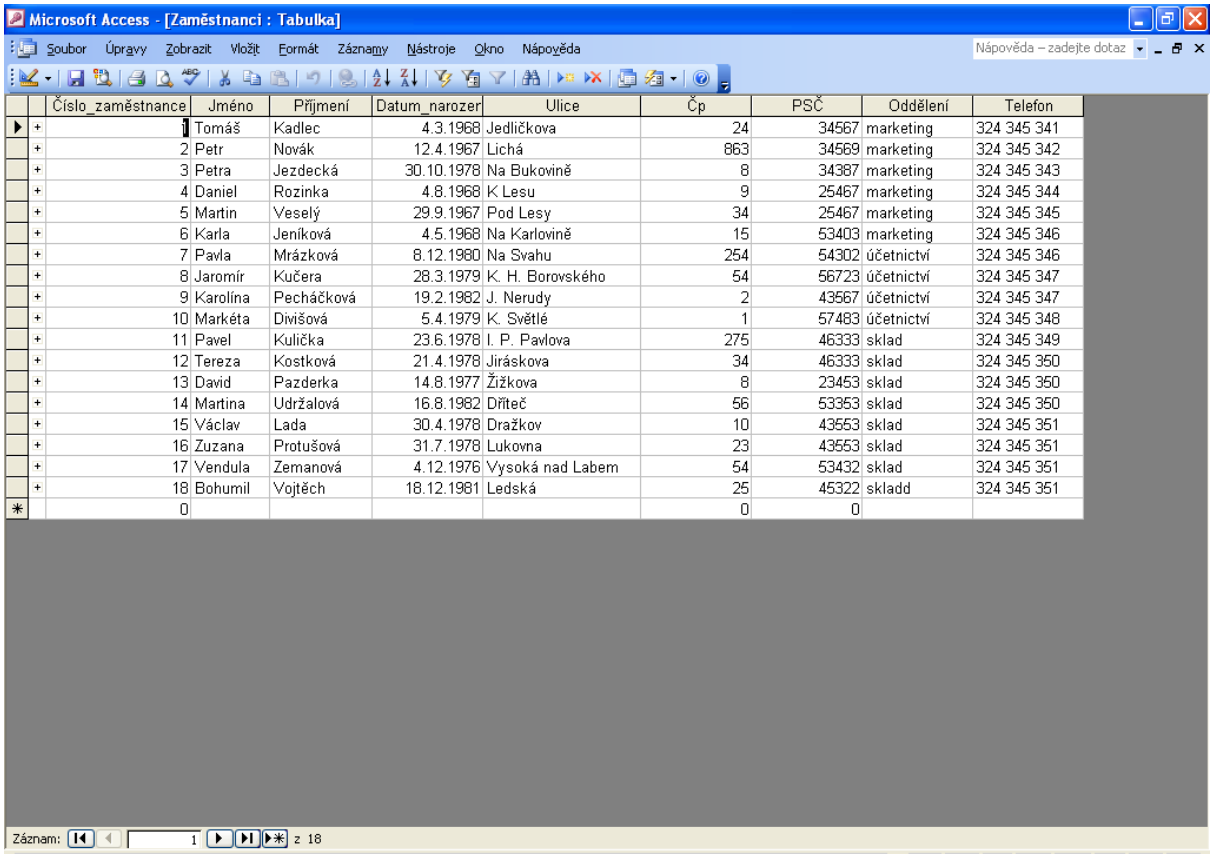
Obrázek 25 Vytvoření tabulky pomocí návrhového zobrazení



Obrázek 26 Tabulka vytvořená pomocí průvodce



Obrázek 27 Tabulka Dodavatelé



Číslo_zaměstnance	Jméno	Příjmení	Datum_narozer	Ulice	Čp	PSČ	Oddělení	Telefon
1	Tomáš	Kadlec	4.3.1968	Jedličkova	24	34567	marketing	324 345 341
2	Petr	Novák	12.4.1967	Lichá	863	34569	marketing	324 345 342
3	Petra	Jezdecká	30.10.1978	Na Bukovině	8	34387	marketing	324 345 343
4	Daniel	Rozinka	4.8.1968	K Lesu	9	25467	marketing	324 345 344
5	Martin	Veselý	29.9.1967	Pod Lesy	34	25467	marketing	324 345 345
6	Karla	Jeníková	4.5.1968	Na Karlovině	15	53403	marketing	324 345 346
7	Pavla	Mrázková	8.12.1980	Na Svahu	254	54302	účetnictví	324 345 346
8	Jaromír	Kučera	28.3.1979	K. H. Borovského	54	56723	účetnictví	324 345 347
9	Karolína	Pecháčková	19.2.1982	J. Nerudy	2	43567	účetnictví	324 345 347
10	Markéta	Divišová	5.4.1979	K. Světlé	1	57483	účetnictví	324 345 348
11	Pavel	Kulička	23.6.1978	I. P. Pavlova	275	46333	sklad	324 345 349
12	Tereza	Kostková	21.4.1978	Jiráskova	34	46333	sklad	324 345 350
13	David	Pazderka	14.8.1977	Žižkova	8	23453	sklad	324 345 350
14	Martina	Udržalová	16.8.1982	Dříteč	56	53353	sklad	324 345 350
15	Václav	Lada	30.4.1978	Dražkov	10	43553	sklad	324 345 351
16	Zuzana	Protušová	31.7.1978	Lukovna	23	43553	sklad	324 345 351
17	Vendula	Zemanová	4.12.1976	Vysoká nad Labem	54	53432	sklad	324 345 351
18	Bohumil	Vojtěch	18.12.1981	Ledská	25	45322	skladd	324 345 351
0					0	0		

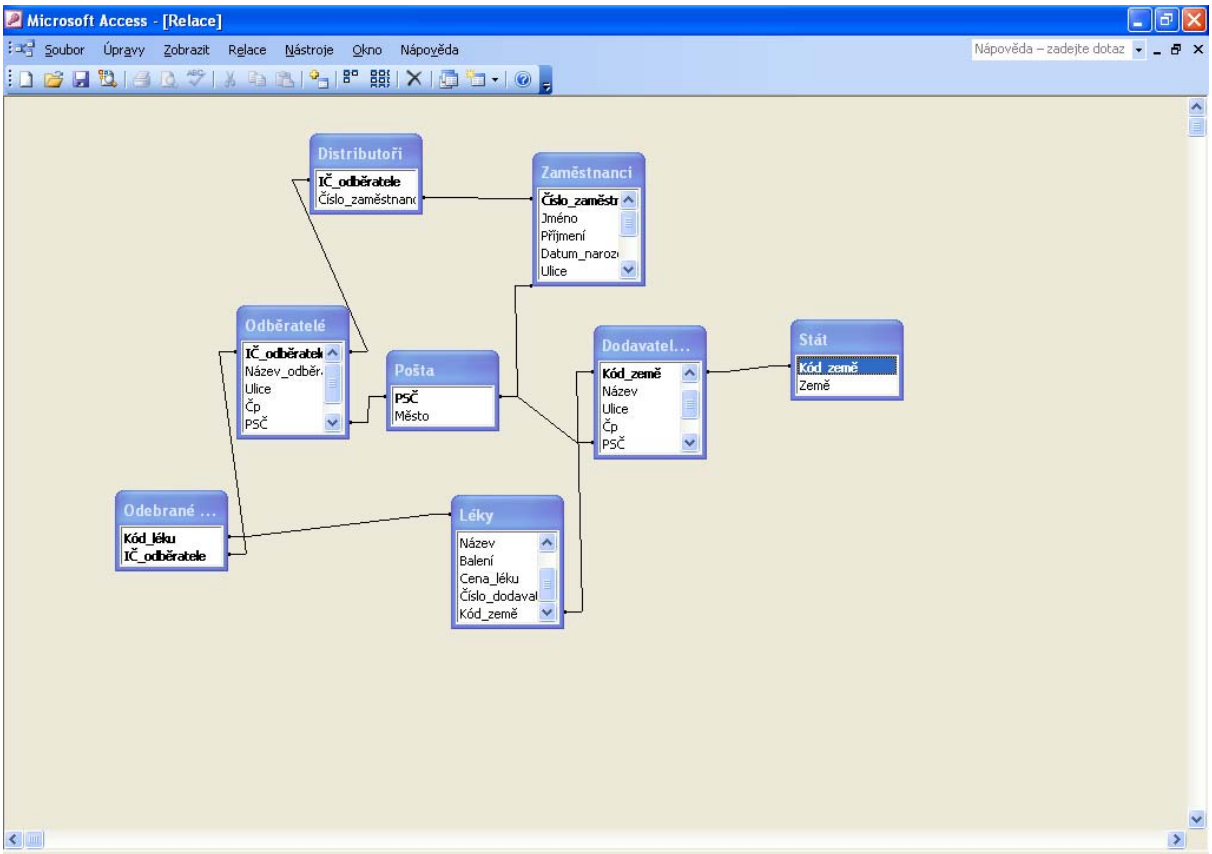
Obrázek 28 Tabulka Zaměstnanci

8.3. Definování vztahů

Po nadefinování souvisejících tabulek, je nutné MS říci, v jakých vztazích jsou tyto tabulky. Pro nadefinování vztahů je nutné se vrátit do okna Databáze a vybrat z nabídky Nástroje příkaz Relace. Access otevře prázdné okno Relace a otevře dialogové okno Zobrazit tabulku. Je třeba označit postupně všechny tabulky a kliknout na tlačítko přidat. [13]

Vztahy mezi tabulkami se vytvoří tak, že se vezme pole s primárním klíčem z jedné tabulky a tahem myši ho přetáhne do druhé tabulky na cizí klíč, který obě tyto tabulky propojuje.

Toto je nutné provést u všech souvisejících tabulek, které je třeba propojit.

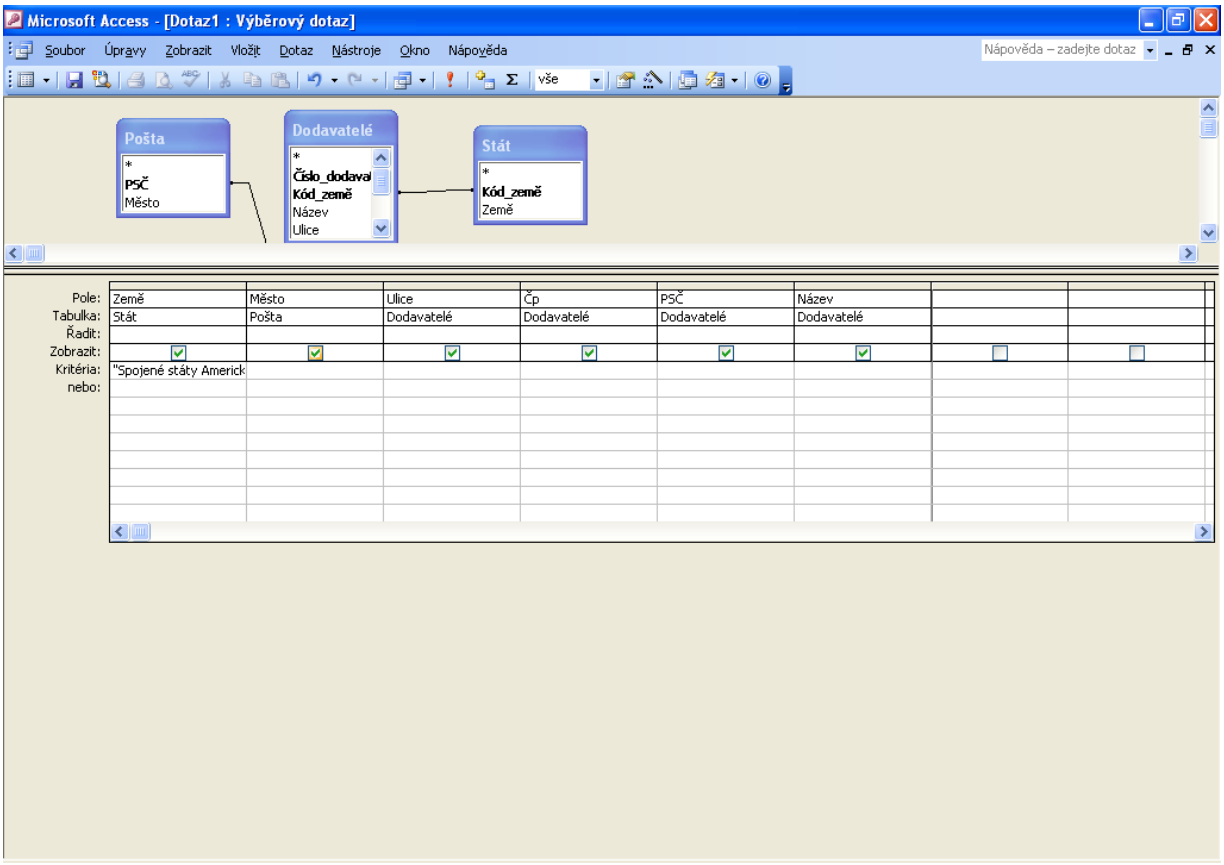


Obrázek 29 Relace

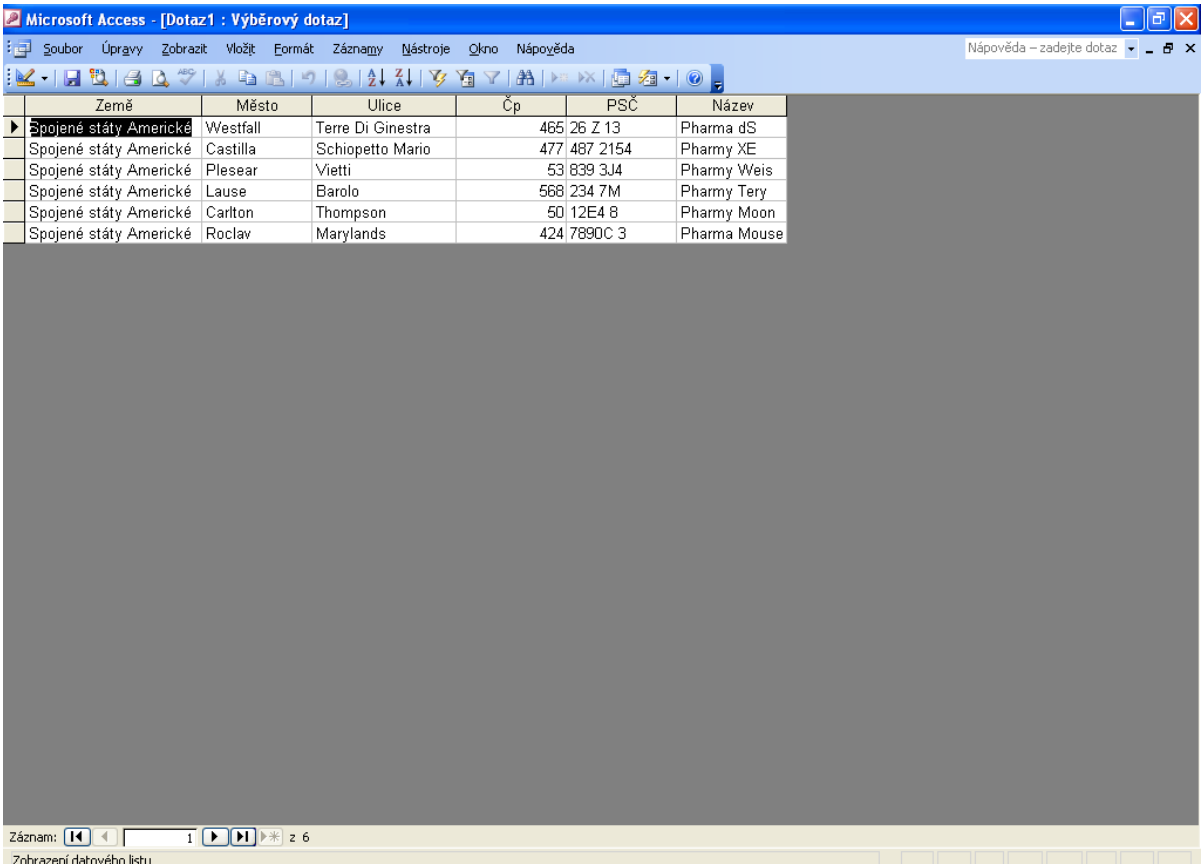
8.4. Dotazy

Jednou z výhod využití dotazů je to, že umožňují snadné vyhledávání dat ve více spojených tabulkách. Dotazy jsou však užitečné i pro zpracování dat v jediné tabulce. Nejjednodušším způsobem, jak začít vytvářet dotaz nad nějakou tabulkou je otevřít okno Databáze, vybrat požadovanou tabulku a z rozbalovacího seznamu tlačítka Nový objekt v panelu nástrojů vybrat možnost Nový dotaz. Okno dotaz v návrhovém zobrazení má dvě hlavní části. V horní části jsou seznamy polí z tabulek nebo dotazů, které se pro dotaz vybírají. Ve spodní části okna je návrhová mřížka, ve které se provádí veškerá činnost při navrhování. [13]

V nabídce pole se vyberou hodnoty, které se mají zobrazit, to je PSČ z tabulky Adresa, Země z tabulky stát, Ulice, ČP, Město a Číslo dodavatele z tabulky Dodavatelé. Zjišťuje se celá adresu dodavatelů. Jako kritérium je zadáno, že země musí být Francie. Jinými slovy, zobrazí se jenom dodavatelé z Francie, resp. jejich adresy.

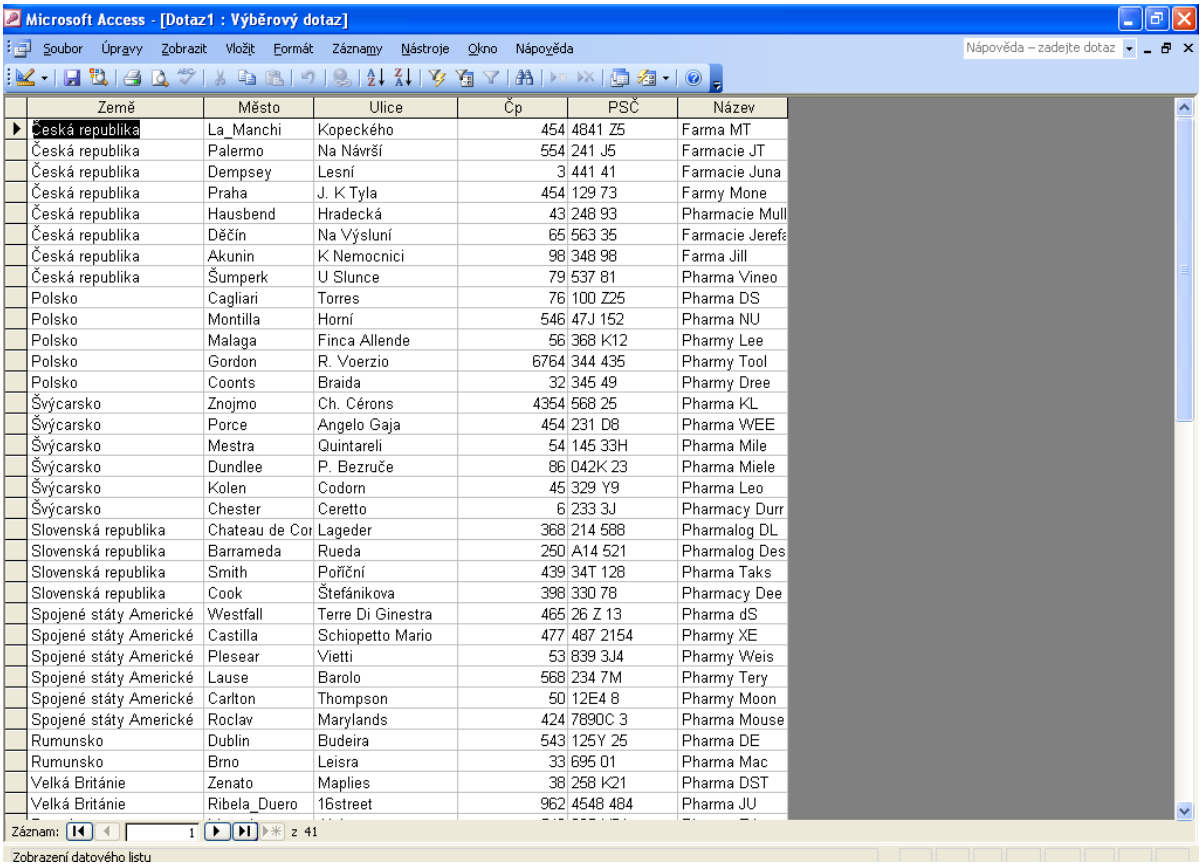


Obrázek 30 Výběrový dotaz



Obrázek 31 Výsledek výběrového dotazu

Pokud by nebylo zadáno žádné kritérium, výběrový dotaz by vypadal takto:



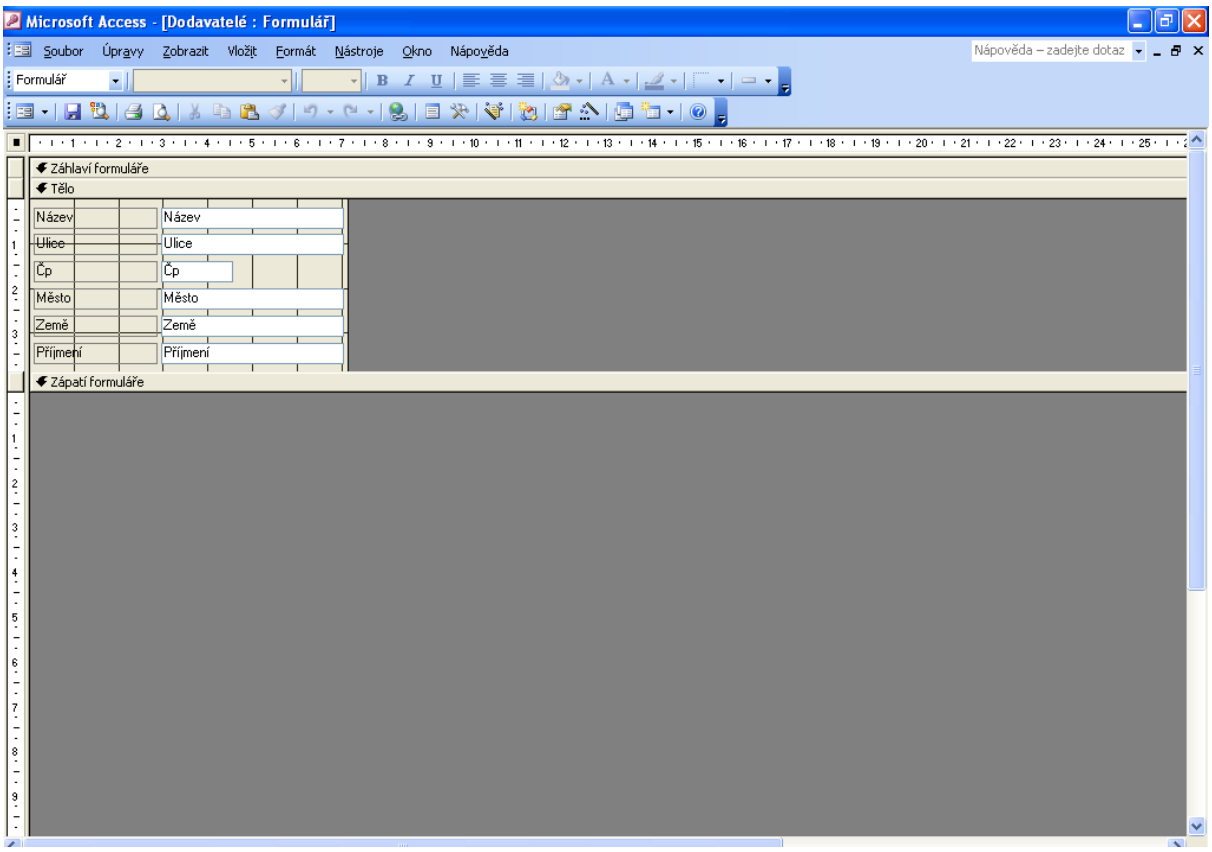
Obrázek 32 Výsledek výběrového dotazu bez kritéria

8.5. Formuláře

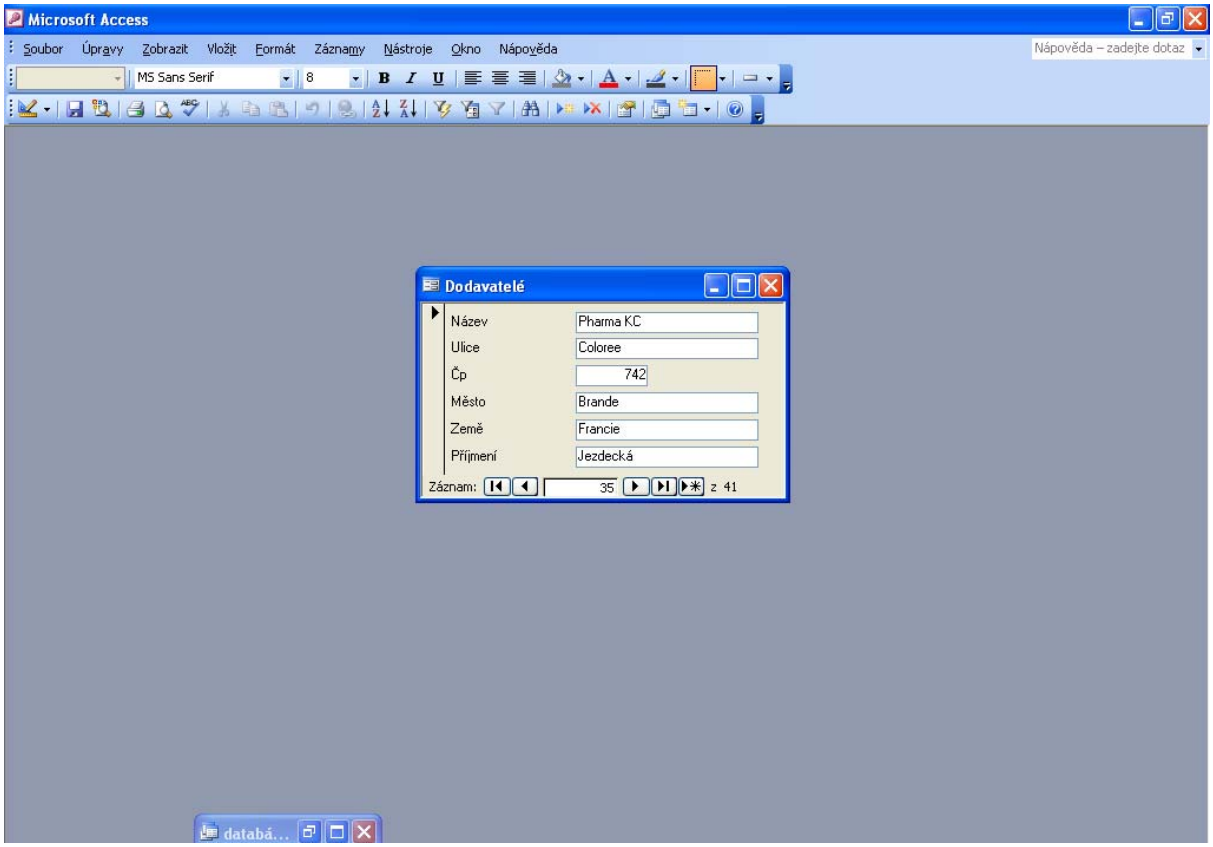
Formuláře tvoří prvotní rozhraní mezi uživateli a aplikací MS Access.

Je třeba otevřít databázi a kliknout na objekt Formulář. Z nabídky vybrat Průvodce formulářem. V seznamu Dostupná pole se může vybrat libovolné pole a kliknout na tlačítko s jednoduchou šipkou vpravo, které zkopíruje toto pole do seznamu Vybraná pole. Pokud jsou v databázi nadefinované mezi tabulkami nějaké vztahy, Průvodce formulářem dokáže určit, v jakých vztazích jsou data z více zdrojů, a umí nabídnout buď vytvoření jednoduchého formuláře, který bude zobrazovat všechna data, nebo složitějšího formuláře, který některá data zobrazuje v hlavní části formuláře a vztažená data zobrazí ve vloženém podformuláři. Kliknutím na tlačítko další, v následujícím okně průvodce poskytne možnosti pro uspořádání formuláře. Průvodce zobrazí další okno, ve kterém je možné vybrat styl formuláře. Nakonec se průvodce zeptá na název formuláře. [13]

Sestavení formuláře, kde bude uveden název, ulice, čp, město, země dodavatele a příjmení zaměstnance, kterého má dodavatel přidělen.



Obrázek 33 Sestavení formuláře

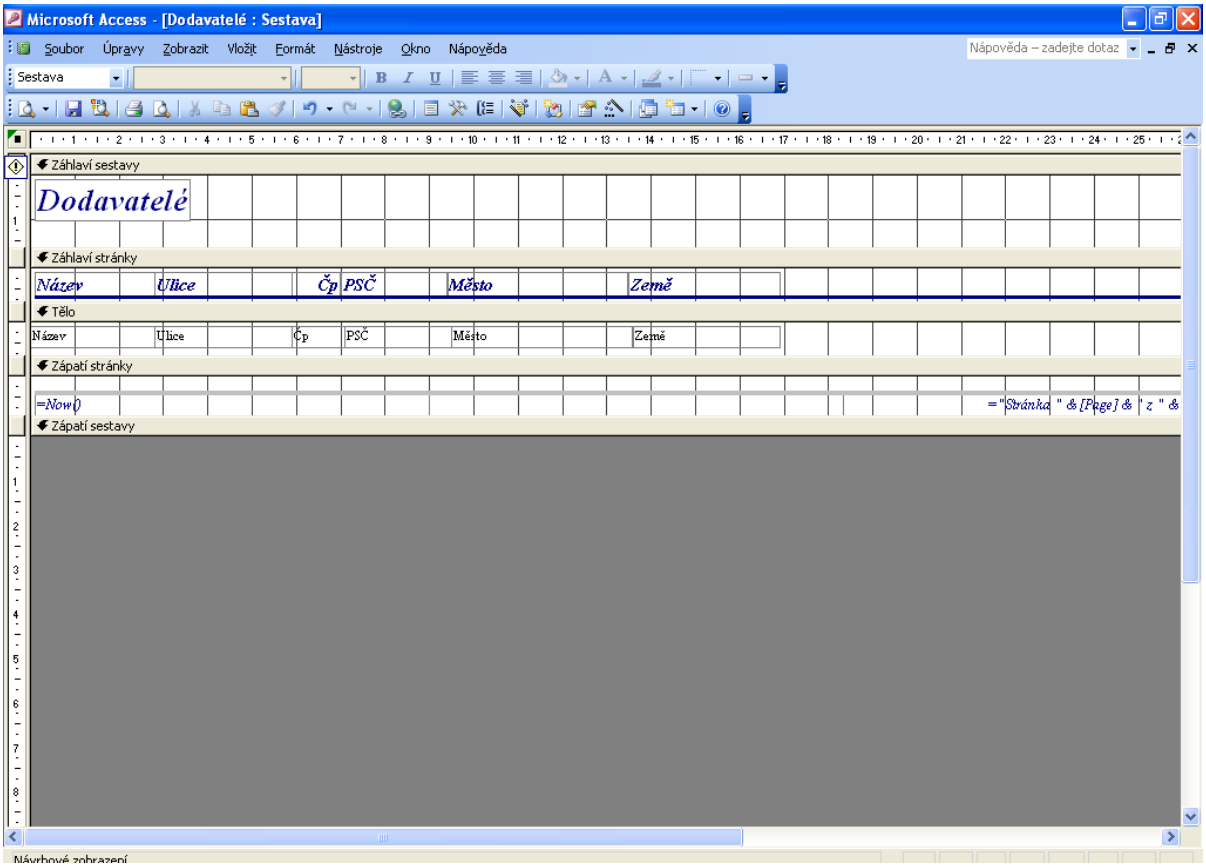


Obrázek 34 Výsledný formulář

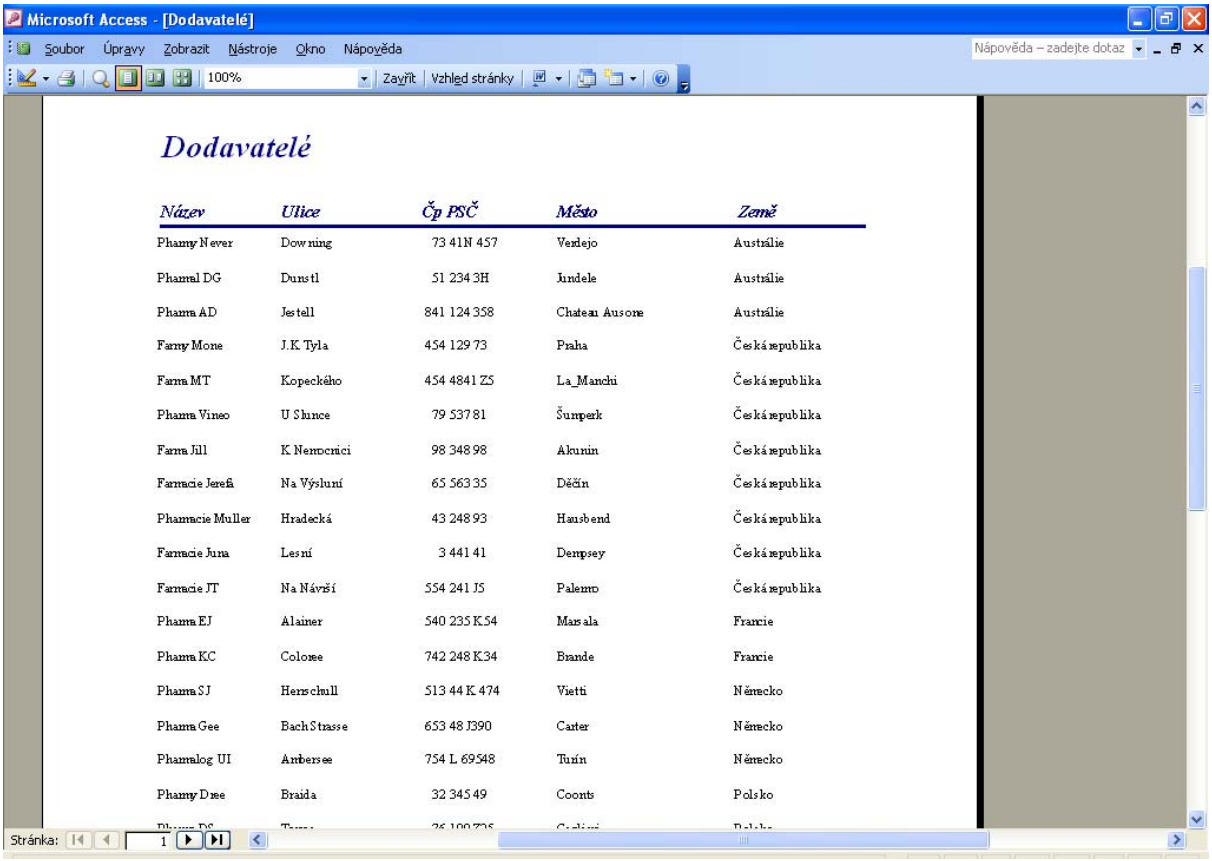
8.6. Sestavy

V okně databáze se vybere objekt Sestava. Vybere se návrhové zobrazení a kliknutím na tlačítko OK se otevře nové okno Sestava v návrhovém zobrazení. Prázdná sestava má oddíly Záhlaví sestavy a Zápatí sestavy, mezi nimiž je tělo sestavy. V horní, resp. dolní části každé stránky se v sestavě bude tisknout záhlaví, resp. zápatí stránky. Je možné přidat záhlaví sestavy, které se vytiskne jednou na začátku sestavy, a zápatí sestavy, které se vytiskne jednou na konci sestavy. [13]

Klíčovou činností, kterou se sestavy liší od formulářů, je skutečnost, že u sestav se můžou informace pro zobrazení seskupovat za použití okna Řazení a seskupování. [13]



Obrázek 35 Vytvoření sestavy



Obrázek 36 Sestava

9. ZÁVĚR

Cílem práce je podat určité skupině uživatelů základní informace o databázích, relačním modelu dat a poskytnout srozumitelný návod pro vytvoření relačního modelu dat a následně databázové struktury v databázovém systému MS Access.

Výsledná práce je rozdělena do dílčích částí. V první části práce je uvedena charakteristika databázových systémů a datových modelů.

V další části je věnována pozornost přímo relačnímu modelu dat. Jsou uvedeny vlastnosti relačního modelu dat, charakteristika datového modelu ve fázi analýzy a návrhu a základní datové typy.

Další kapitola seznamuje s diagramem entit a vztahů (E-R diagram). E-R diagram je velmi důležitou částí návrhu databáze. Je zde popsána charakteristika entit, domén, atributů, vztahů mezi entitami, které jsou dané parcialitou a kardinalitou, a klíčů.

Další část je věnována převodu E-R diagramu do relačního modelu dat. Dochází zde k transformaci a následné normalizaci, kdy dochází k odstraňování chyb.

Následuje stručný popis Microsoft Access a jeho základních objektů.

Následná kapitola se již věnuje konceptuální analýze a návrhu vybrané reálie. Další varianty řešení příkladu dané reálie jsou uvedeny v příloze. Určují se entity, atributy, klíče a vztahy mezi entitami. Zachycuje se E-R diagram, jeho transformace a následná normalizace.

Závěrečnou část práce tvoří implementace do databázového systému Microsoft Access a konkrétní ukázky vytvořené databáze.

Pro práci byla využita řada příslušných publikací, internetových stránek a dílčích studií, přesto přispívá k přehlednému postupu vytvoření relačního modelu dat a jeho konkrétní implementaci do databázového systému.

Přínosem této práce je ukázka uceleného příkladu, protože dostupné materiály obsahovaly vždy jen dílčí ukázky. Příklad pokrývá zjištění entit ve vybrané reálii, stanovení atributů a primárních klíčů, definování vztahů mezi entitami, vytvoření E-R diagramu, transformace E-R diagramu do relačního modelu dat, normalizaci až po vlastní realizaci v databázovém systému MS Access.

10. SEZNAM LITERATURY A POUŽITÝCH ZDROJŮ

[1] DATE, C. J.: *Introduction to databáze systems*, 8th edition, Boston:Pearson Education Limited, 2004, 983 s., ISBN: 0-321-19784-4

[2] ŠEŠERA, Lubor, MIČOVSKÝ, Aleš, ČERVEŇ Juraj: *Datové modelování v příkladech*, 1. vydání, Praha: Grada, 2001, 151 s., ISBN 80-247-0049-2

[3] POKORNÝ, Jaroslav: *Konstrukce databázových systémů*, 2. vydání, Praha: Vydavatelství ČVUT, 2004, 116 s., ISBN 80-01-02898-4

[4] POKORNÝ, Jaroslav, HALAŠKA Ivan: *Databázové systémy*, 1. vydání, Praha: Vydavatelství ČVUT, 1998, 146 s., ISBN 80-01-01724-9

[5] RIORDAN, Rebecca M.: *Vytváříme relační databázové aplikace*, 1. vydání, Praha: Computer Press, 2000, 280 s., ISBN

[6] OPPEL, Andrew J.: *Databáze bez předešlých znalostí*, 1. vydání, Brno: Computer Press, 2006, 319 s., ISBN 80-251-1199-7

[7] RIESSLER, Petr: *Databáze a programování*, 1. vydání, Zlín:Vysoké učení technické, 2000, 102 s., ISBN 80-214-1778-1

[8] PÍSEK, Slavoj: *Databáze v Accessu*, 1. vydání, Praha: Grada, 2003,

ISBN 80-247-0572-9

[9] HERNANDEZ, Michael J.: *Návrh databází*, 1. vydání, Praha: Grada, 2006, 408 s., ISBN: 980-247-0900-7

[10] ŠIMŮNEK, Milan: *SQL kompletní kapesní průvodce*, 1. vydání, Praha: Grada Publishing, 1999, 247 s., ISBN 80-7169-692-7

[11] HAVLÍČEK, Zdeněk: *Databázové systémy Paradox*, 1. vydání, Praha: Grada, 1992, 132 s., ISBN 80-85424-97-5

[12] VIESCAS, John L.: *Mistrovství v Microsoft Access 97*, 1. vydání, Praha: Computer Press, 1998, 750 s., ISBN 80-7226-107-X

[13] *Relační databáze*, 29. 12. 2006. [online] [citováno 2007-01-20] Dostupné na WWW: <http://cs.wikipedia.org/wiki/Rela%C4%8Dn%C3%AD_datab%C3%A1ze>

[14] Fakulta informatiky Masarykova univerzita, *Relační vs. objektově-relační vs. objektové databáze*, [online] [citováno 2007-01-20] Dostupné na WWW: <http://www.fi.muni.cz/~xbatko/oracle/compare.htm>

[15] POKORNÝ, Josef, *Objektově-relační databáze*, 2002, [online] [citováno 2007-01-19] Dostupné na WWW: <http://www.datakon.cz/datakon02/d02_pokorny.pdf>

[16] VONDRA, Tomáš, *Datový model*, 2004, [online] [citováno 2007-01-19] Dostupné na WWW: <http://www.fuzzy.cz/databaze/>

[17] *Relační datový model*, [online] [citováno 2007-01-20] Dostupné na WWW: <http://www.owebu.cz/databaze/vypis.php?clanek=334>

[18] ŠVEC, Martin, *Objektové databáze*, 2003, [online] [citováno 2007-01-18] Dostupné na WWW: <http://fit.fuckin.rulez.sk/study/courses/VPD/public/0203VPD-Svec.pdf>

[19] *Databáze*, 2006, [online] [citováno 2007-01-18] Dostupné na WWW: <http://cs.wikipedia.org/wiki/Datab%C3%A1ze>

[20] *Síťová databáze*, 2007, [online] [citováno 2007-01-18] Dostupné na WWW: <http://cs.wikipedia.org/wiki/S%C3%AD%C5%A5ov%C3%A1_datab%C3%A1ze>

[21] *Relační databáze*, 2006, [online] [citováno 2007-01-18] Dostupné na WWW: <http://cs.wikipedia.org/wiki/Rela%C4%8Dn%C3%AD_datab%C3%A1ze>

[22] PROCHÁZKA, Tomáš, *Databázové systémy*, 2001, [online] [citováno 2007-01-18] Dostupné na WWW: <http://www.umel.feec.vutbr.cz/%7Edanekl/Archiv.PCK/!!!2001>

[23] FARANA, Radim, Databázové systémy Microsoft Access 2.0, 1995, [online] [citováno 2007-01-18] Dostupné na WWW: <http://www.fs.vsb.cz/books/dbacc20/dbacc01.htm>

[24] ZELENÝ, Jindřich, Metody pro přístup k datům v databázových systémech, 2002, [online] [citováno 2007-01-14] Dostupné na WWW: <http://nb.vse.cz/%7Ezelnyj/it380/eseje/jzeleny/indexy/indexy.htm>

[25] MATIAŠKO, Karol, Databázové systémy,1. vydání, Žilina: Žilinská univerzita, 2002, 367 s., ISBN 80-7100-968-1

11. PŘÍLOHA

Obchodní firma s licencí získanou od Ministerstva zdravotnictví ČR se zabývá distribucí léčiv. Dodavatelé pochází jak z tuzemska, tak i ze zahraničí. Firma má řadu zaměstnanců, kteří se zabývají účetnictví, skladováním, dopravou a dalšími činnostmi souvisejícími s touto podnikatelskou činností. Odběratelé jsou tuzemské lékárny, nemocnice a zdravotní a sociální ústavy. Databáze je částečně spjata s účetnictvím.

11.1. Určení entit

Nejprve je nutné určit entity,které popíší jednotlivé objekty reality (ty objekty, o kterých má smysl uchovávat nějakou informaci):

- Zaměstnanci
- Dodavatelé
- Odběratelé
- Přijaté faktury
- Vydané faktury
- Léky

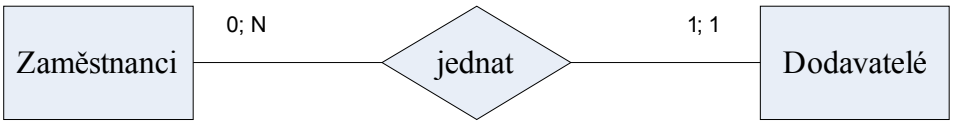
11.2. Určení atributů

Navrhovaný systém o každé entitě zaznamenává a sleduje určité skutečnosti, těmto údajům se říká atributy dané entity. Atributy se získají na základě prostudování dokladů. Atributy představují jednotlivé položky účetních dokladů, entit.

Primární klíč je atribut, nebo soustava atributů, jejichž hodnoty tvoří jednoznačnou identifikaci řádku relace. Primární klíč je absolutně nejdůležitějším klíčem v celé tabulce. [5]

.

11.3. Určení vztahů



ERD Zaměstnanci - Dodavatelé

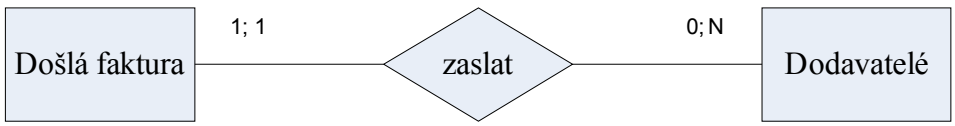
Zaměstnanec může jednat s jedním nebo více dodavateli. Dodavatel musí jednat s jedním zaměstnancem.

ERD Zaměstnanci - Došlá faktura

Zaměstnanec může zpracovat více došlých faktur a došlá faktura musí být zpracována jedním zaměstnancem.

ERD Došlá faktura - Léky

Došlá faktura musí obsahovat jeden nebo více léků a lék může být obsažen na jedné nebo více došlých fakturách.



ERD Došlá faktura - Dodavatelé

Došlá faktura musí být zaslána od jednoho dodavatele a dodavatel může zaslat více došlých faktur.

ERD Zaměstnanci - Vydaná faktura

Zaměstnanec může vyúčtovat jednu nebo více faktur a vydaná faktura musí být vyúčtována jedním zaměstnancem.

ERD Vydaná faktura - Léky

Vydaná faktura musí obsahovat jeden nebo více léků a léky mohou být na více vydaných fakturách.

ERD Vydaná faktura - Odběratelé

Vydaná faktura musí být zaslána jednomu odběrateli a odběratel může obdržet více vydaných faktur.

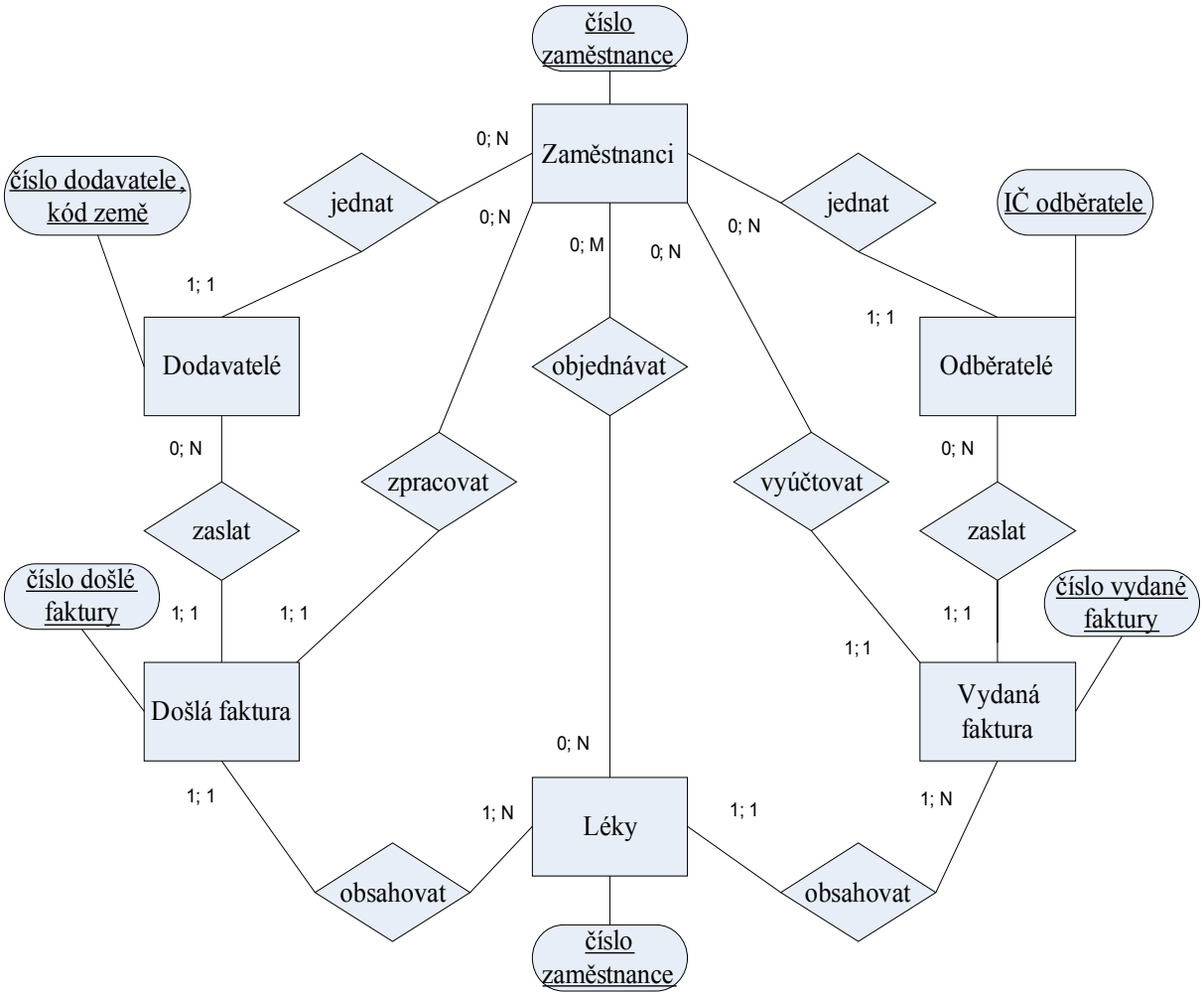
ERD Zaměstnanci - Léky

Zaměstnanci mohou objednávat více léků. Léky musí objednávat jeden nebo více zaměstnanců.

ERD Zaměstnanci - Odběratelé

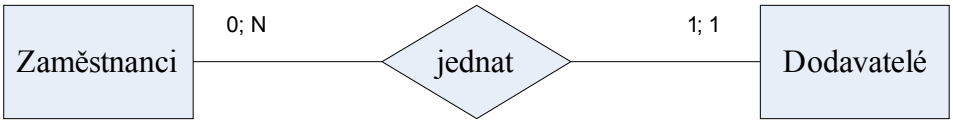
Zaměstnanci mohou jednat s více odběrateli. Odběratelé musí jednat s jedním zaměstnancem.

11.4. ER diagram



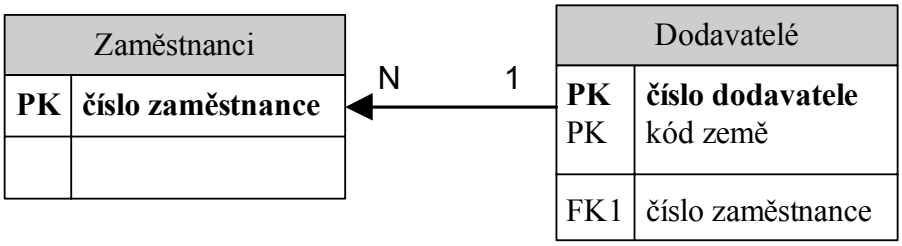
ER diagram

11.5. Transformace ER diagramu do relačního modelu dat



ERD Zaměstnanci - Dodavatelé

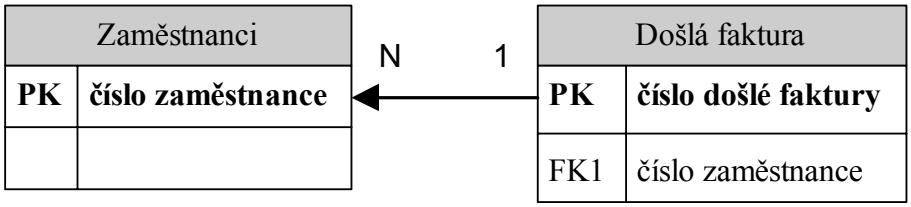
Pokud je dán vztah mezi dvěma entitami s kardinalitou 1;1 – x;N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá cizí klíč.



Návrh RMD Zaměstnanci – Dodavatelé

ERD Zaměstnanci - Došlá faktura

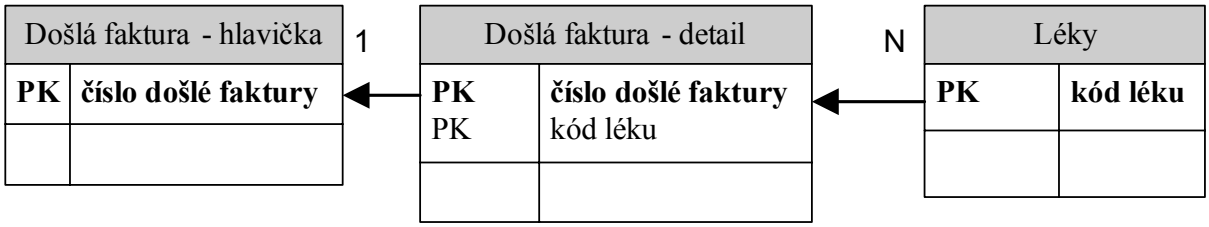
Pokud je dán vztah mezi dvěma entitami s kardinalitou 1;1 – x;N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá cizí klíč.



Návrh RMD Zaměstnanci - Došlá faktura

Vztah Došlá faktura - Léky

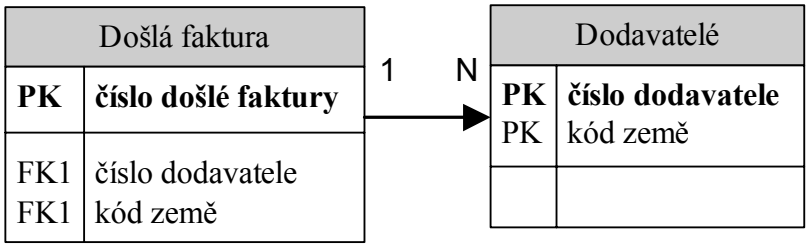
Pokud je dán vztah mezi dvěma entitami s parcialitou a kardinalitou 1;1 – x; N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá jako cizí klíč primární klíč druhé entity.



Návrh RMD Došlá faktura – Léky

ERD Došlá faktura - Dodavatelé

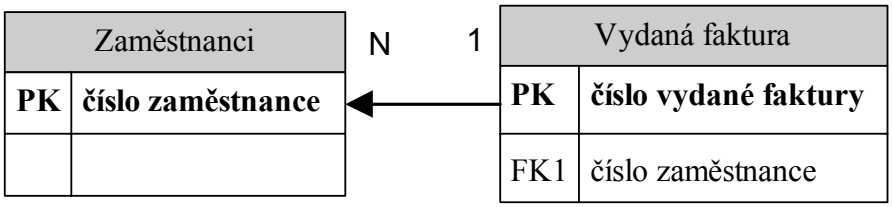
Pokud je dán vztah mezi dvěma entitami s parcialitou 1;1 a x; N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá jako cizí klíč primární klíč druhé entity.



Návrh RMD Došlá faktura – Dodavatelé

ERD Zaměstnanci - Vydaná faktura

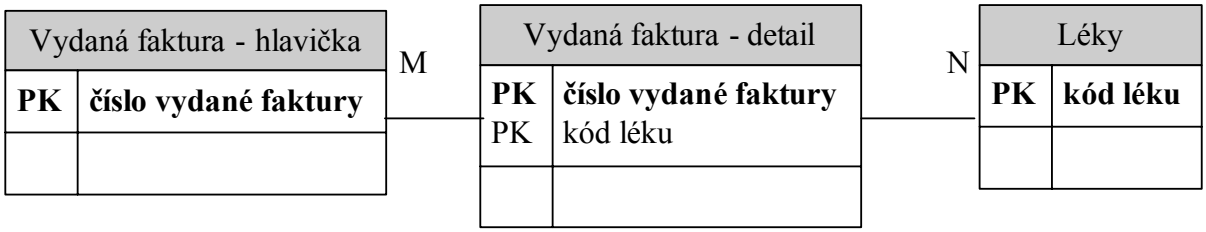
Pokud je dán vztah mezi dvěma entitami s parcialitou a kardinalitou 1;1 – x; N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá cizí klíč primární klíč druhé entity.



Návrh RMD Zaměstnanci - Vydaná faktura

ERD Vydaná faktura - Léky

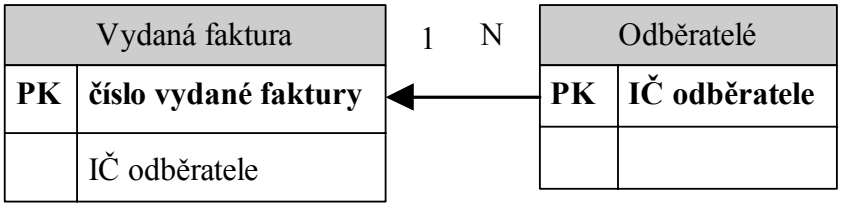
Pokud je dán vztah mezi dvěma entitami s parcialitou a kardinalitou x; M – x; N vzniká třetí entita. Tato entita má spojený primární klíč, složený z obou primárních klíčů.



Návrh RMD Vydaná faktura – Léky

ERD Vydaná faktura - Odběratelé

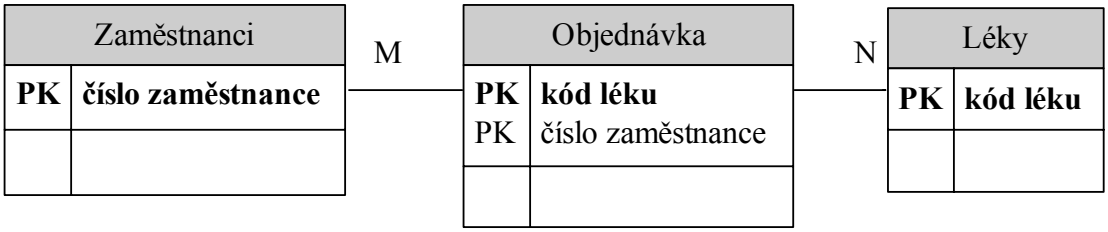
Pokud je vztah mezi dvěma entitami s parcialitou a kardinalitou 1;1 – x; N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá jako cizí klíč primární klíč druhé entity.



Návrh RMD Vydaná faktura – Odběratelé

ERD Zaměstnanci - Léky

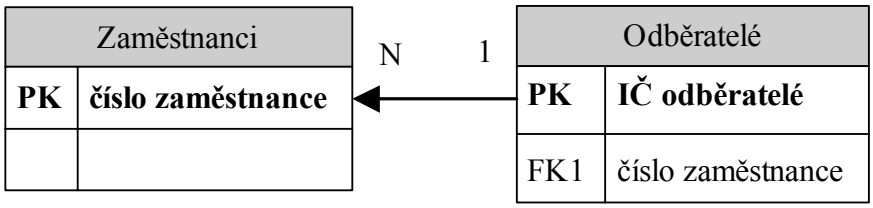
Pokud je vztah mezi dvěma entitami s parcialitou x; M – x; N vzniká třetí entita. Tato entita má spojený primární klíč, složený z obou primárních klíčů.



Návrh RMD Zaměstnanci – Objednávka

ERD Zaměstnanci - Odběratelé

Pokud je dán vztah mezi dvěma entitami s parcialitou a kardinalitou 1;1 – x; N zůstávají pouze tyto dvě entity. K entitě s parcialitou 1 se přidá cizí klíč primární klíč druhé entity.



Návrh RMD Zaměstnanci – Odběratelé

11.6. Proces normalizace

Normalizace je proces rozkladu velkých tabulek na menší, což slouží k eliminaci redundantních a duplicitních dat a k eliminaci budoucích problémů s vkládáním, rušením a aktualizací dat. [5]

Karta zaměstnance
Číslo zaměstnance: 00023
Jméno a příjmení: Petr Novák
Adresa: Vinohradská 24, 120 00 Praha 2
Oddělení: účetní
Telefon: 254 063 023
Datum narození: 5. 3. 1970

1 Karta zaměstnance

V této podobě není možné data použít, musíme je proto upravit pomocí normálních forem.

Relace je v první normální formě, jestliže jsou všechny její atributy definovány nad doménami (skalárními obory hodnot). Každý atribut vektoru hodnot musí obsahovat jen jednu jednotlivou, skalární hodnotu. Jinými slovy, každý atribut musí obsahovat pouze jednu hodnotu. [5]

Zaměstnanci 1. NF

	Zaměstnanci (číslo zaměstnance, jméno, adresa, oddělení, telefon, datum narození)
1 NF	Zaměstnanci (číslo zaměstnance, jméno, příjmení, ulice a čp, PSČ, město, oddělení, telefon, datum narození)

Relace je ve druhé normální formě, právě když je v první normální formě a zároveň jsou všechny její atributy závislé na celém kandidátním klíči. Jde o požadavek nevyjadřovat dvě různé entity v jedné relaci. Pokud se obě tyto entity od sebe oddělí, ztratí se tím nejen nežádoucí redundance, ale definuje se také mechanismus pro uložení informací. [5]

Zaměstnanci 2. NF

	Zaměstnanci (číslo zaměstnance, jméno, adresa, oddělení, telefon, datum narození)
1 NF	Zaměstnanci (číslo zaměstnance, jméno, příjmení, ulice a čp, PSČ, město, oddělení, telefon, datum narození)
2 NF	Zaměstnanci (číslo zaměstnance, jméno, příjmení, ulice a čp, PSČ, město, oddělení, telefon, datum narození)

Relace je ve třetí normální formě, právě když je ve druhé normální formě a zároveň jsou všechny její neklíčové atributy vzájemně nezávislé. Samostatná entita se vytvoří jen tehdy, když je nová entita pro datový model významná nebo pokud se její data budou často měnit. [5]

Zaměstnanci 3. NF

	Zaměstnanci (<u>číslo zaměstnance</u>, jméno, adresa, oddělení, telefon, datum narození)
1 NF	Zaměstnanci (<u>číslo zaměstnance</u>, jméno, příjmení, ulice a čp, PSČ, město, oddělení, telefon, datum narození)
2 NF	Zaměstnanci (<u>číslo zaměstnance</u>, jméno, příjmení, ulice a čp, PSČ, město, oddělení, telefon, datum narození)
3 NF	Zaměstnanci (<u>číslo zaměstnance</u>, jméno, příjmení, ulice a čp, PSČ, oddělení, telefon, datum narození) Pošta (<u>PSČ</u>, město)

Atribut Město je závislý na atributu PSČ, protože podle konkrétního PSČ se nechá zjistit o jaké město se jedná. Vzniká tedy nová entita PSČ.

Dodavatel

Číslo dodavatele: 00025 Kód země: CZE
Název dodavatele: Pharma, s. r. o.
Adresa: Za Bránou 257, 250 32 Brno, Česká republika
Bankovní spojení: 1288745262/2200
Kontaktní osoba: František Toman Telefonní číslo: 547 845 123
Kód a název léku: 11025 Paralen, 11011 Nolicin, 11008 Ipobrufen
Jméno a příjmení zaměstnance: Petr Novák

Karta dodavatele

Dodavatelé 1. NF

	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, adresa, bankovní spojení, kontaktní osoba, telefonní číslo, kód a název léku, jméno a příjmení, číslo zaměstnance)
1 NF	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, ulice a čp, PSČ, město, země, bankovní spojení, kontaktní osoba, telefonní číslo, kód léku, název léku, jméno, příjmení, číslo zaměstnance)

Dodavatelé 2. NF

	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, adresa, bankovní spojení, kontaktní osoba, telefonní číslo,kód a název léku, jméno a příjmení, číslo zaměstnance)
1 NF	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, ulice a čp, PSČ, město, země, bankovní spojení, kontaktní osoba, telefonní číslo, kód léku, název léku, jméno, příjmení, číslo zaměstnance)
2 NF	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, ulice a čp, PSČ, město, bankovní spojení, kontaktní osoba, telefonní číslo, kód léku, název léku, jméno, příjmení, číslo zaměstnance) Stát (<u>kód země</u>, země)

Dodavatelé 3. NF

	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, adresa, bankovní spojení, kontaktní osoba, telefonní číslo, kód léku, název léku, jméno a příjmení, číslo zaměstnance)
1 NF	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, ulice a čp, PSČ, město, země, bankovní spojení, kontaktní osoba, telefonní číslo, kód léku, název léku, jméno, příjmení, číslo zaměstnance)
2 NF	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, ulice a čp, PSČ, město, bankovní spojení, kontaktní osoba, telefonní číslo, kód léku, název léku, jméno, příjmení, číslo zaměstnance) Stát (<u>kód země</u>, země)
3 NF	Dodavatelé (<u>číslo dodavatele, kód země</u>, název dodavatele, ulice a čp, PSČ, bankovní spojení, kontaktní osoba, telefonní číslo, kód léku, číslo zaměstnance) Pošta (<u>PSČ</u>, město) Léky (<u>kód léku</u>, název léku) Zaměstnanci (<u>číslo zaměstnance</u>, jméno, příjmení)

Zkrácená ukázka entity Dodavatelé

Číslo dodavatele	Název dodavatele	Ulice a čp	PSČ	Kontaktní osoba	Číslo zaměstnance	Kód léku
0025	Pharma s. r. o.	Za Bránou 257	25032	František Toman	00023	11025
0025	Pharma s. r. o.	Za Bránou 257	25032	František Toman	00023	11008
0025	Pharma s. r. o.	Za Bránou 257	25032	František Toman	00023	11011

Na první pohled je jasné, že toto řešení není optimální, protože údaje dodavatele se tady opakují podle počtu výrobků, které dodává.

Odběratel

Číslo odběratele: 25478962
Název odběratele: Lékárna U Dubu
Adresa: U Dubu 68, 152 00 Praha 5
Bankovní spojení: 4525555541/0800
Kontaktní osoba: Petra Mrštíková
Telefonní číslo: 254 332 125
Jméno a příjmení zaměstnance: Tomáš Riegel

Karta odběratele

Normalizace entity Odběratelé

	Odběratelé (IČ odběratele, název odběratele, adresa, bankovní spojení, kontaktní osoba, telefonní číslo, jméno a příjmení, číslo zaměstnance)
1 NF	Odběratelé (IČ odběratele, název odběratele, ulice a čp, PSČ, město, bankovní spojení, kontaktní osoba, telefonní číslo, jméno, příjmení, číslo zaměstnance)
2 NF	Odběratelé (IČ odběratele, název odběratele, ulice a čp, PSČ, město, bankovní spojení, kontaktní osoba, telefonní číslo, jméno, příjmení, číslo zaměstnance)
3 NF	Odběratelé (IČ odběratele, název odběratele, ulice a čp, PSČ, bankovní spojení, kontaktní osoba, telefonní číslo, číslo zaměstnance) Pošta (PSČ, město) Zaměstnanci (číslo zaměstnance, jméno, příjmení)

Zkrácená ukázka entity Odběratelé

Číslo odběratele	Název odběratele	Ulice a čp	PSČ	Číslo zaměstnance	Kontaktní osoba
25478962	Lékárna	U Dubu 68	152 00	00014	Petra Mrštíková

FAKTURA (daňový doklad)

Dodavatel
Pharma, s. r. o.
Za Bránou 257
250 32 BRNO
Číslo účtu:

IČ: 018759

Pořadové číslo: 3
Konstantní symbol: 0008
Datum zdanitelného plnění: 3. 1. 2007
Datum splatnosti: 24. 1. 2007

Odběratel
Farmacie PP, s. r. o.
Mezi Mosty 24
245 02 Kolín

IČ: 254759

Označení dodávky	Množství	Cena
12124 Trinordior 21	500	100.000,--
11011 Nolicin	1.000	145.000,--
11025 Paralen	10.000	80.000,--
		325.000,--

Došlá faktura

Normalizace entity Došlá faktura

	Došlá faktura (<u>číslo faktury</u>, datum, název dodavatele, adresa, množství, název léku, cena, číslo zaměstnance, <u>kód léku</u>, číslo dodavatele, kód země)
1 NF	Došlá faktura (<u>číslo faktury</u>, datum, název, ulice, PSČ, město, země, název léku, cena, množství, číslo zaměstnance, <u>kód léku</u>, číslo dodavatele, kód země)
2 NF	Došlá faktura – detail (<u>číslo faktury</u>, <u>kód léku</u>, název léku, cena, množství) Došlá faktura – hlavička (<u>číslo faktury</u>, datum, číslo dodavatele, kód země, název, ulice, PSČ, město, země, číslo zaměstnance)
3 NF	Došlá faktura – detail (<u>číslo faktury, kód léku</u>, cena, množství) Došlá faktura – hlavička (<u>číslo faktury</u>, datum, číslo dodavatele, kód země, číslo zaměstnance) Dodavatelé (<u>číslo dodavatele, kód země</u>, název, ulice, PSČ) Stát (<u>kód země</u>, země) Pošta (<u>PSČ</u>, město)

FAKTURA (daňový doklad)

Dodavatel
Farmacie PP, s. r. o.
Mezi Mosty 24
245 02 Kolín
Číslo účtu: 1243456/0800

IČ: 254759

Pořadové číslo: 12/2007
Konstantní symbol: 0008
Datum zdanitelného plnění: 1. 2. 2007
Datum splatnosti: 25. 2. 2007

Odběratel
Lékárna U Dubu
U Dubu 68
152 02 Praha

IČ: 25478962

Označení dodávky	Množství	Cena
11011 Nolicin	25	4.000,--
11025 Paralen	10	150,--
		4.150,--

Vydaná faktura

Normalizace entity Vydaná faktura

	Vydaná faktura (<u>číslo faktury</u>, datum, IČ odběratele, název odběratele, adresa, <u>kód léku</u>, název léku, množství, číslo zaměstnance, jméno a příjmení)
1 NF	Vydaná faktura (<u>číslo faktury</u>, řádek, datum, IČ odběratele, název, ulice, PSČ, město, <u>kód léku</u>, název léku, množství, číslo zaměstnance, jméno, příjmení)
2NF	Vydaná faktura – detail (<u>číslo faktury</u>, <u>kód léku</u>, název léku, množství) Vydaná faktura – hlavička (<u>číslo faktury</u>, datum, IČ odběratele, název, ulice, PSČ, město, číslo zaměstnance, jméno, příjmení)
3NF	Vydaná faktura – detail (<u>číslo faktury</u>, <u>kód léku</u>, množství) Vydaná faktura – hlavička (<u>číslo faktury</u>, datum, IČ odběratele, číslo zaměstnance) Zaměstnanci (<u>číslo zaměstnance</u>, jméno, příjmení) Odběratelé (<u>IČ odběratele</u>, název, ulice, PSČ) Pošta (<u>PSČ</u>, město)

Lék

Kód léku: 11025
Název léku: Paralen
ATC skupina: Paracetamolum
Balení: 1 x 10
Cena: 8,00
Dodavatel: Pharma s. r. o., Za Bránou 257, 350 32 Brno

Karta léku

Normalizace entity Léky

	Léky (<u>kód léku</u>, název léku, ATC skupina, balení, cena, dodavatel)
1 NF	Léky (<u>kód léku</u>, název léku, ATC skupina, balení, cena, číslo dodavatele, kód země, název dodavatele, ulice a čp, PSČ, město, země)
2 NF	Léky (<u>kód léku</u>, název léku, ATC skupina, balení, cena, číslo dodavatele, kód země, název dodavatele, ulice a čp, PSČ, město, země)
3 NF	Léky (<u>kód léku</u>, název léku, ATC skupina, balení, cena, číslo dodavatele, kód země) Dodavatelé (<u>číslo dodavatele, kód země</u>, ulice a čp, PSČ) Pošta (<u>PSČ</u>, město) Stát (<u>kód země</u>, země)

Přehled všech entit

Léky	(kód léku, název léku, ATC skupina, balení, cena, číslo dodavatele, kód země)
Stát	(kód země, země)
Pošta	(PSČ, město)
Dodavatelé	Dodavatelé (číslo dodavatele, kód země, název dodavatele, ulice a čp, PSČ, bankovní spojení, kontaktní osoba, telefonní číslo, kód léku, číslo zaměstnance)
Odběratelé	Odběratelé (IČ odběratele, název odběratele, ulice a čp, PSČ, bankovní spojení, kontaktní osoba, telefonní číslo, číslo zaměstnance)
Objednávky	kód léku, číslo zaměstnance
Vydaná faktura	Vydaná faktura – detail (číslo faktury, kód léku, množství) Vydaná faktura – hlavička (číslo faktury, datum, IČ odběratele, číslo zaměstnance)
Došlá faktura	Došlá faktura – detail (číslo faktury, řádek, kód léku, cena, množství) Došlá faktura – hlavička (číslo faktury, datum, číslo dodavatele, kód země, číslo zaměstnance)
Zaměstnanci	Zaměstnanci (číslo zaměstnance, jméno, příjmení, ulice a čp, PSČ, oddělení, telefon, datum narození)

ÚDAJE PRO KNIHOVNICKOU DATABÁZI

Název práce	Výuková aplikace pro relační model dat
Autor práce	Martina Pacáková
Obor	Informatika ve veřejné zprávě
Rok obhajoby	2008
Vedoucí práce	Ing. Stanislava Šimonová, Ph.D.
Anotace	Práce je rozdělena na dvě části. Na teoretické principy datového modelování se zaměřením na relační model dat a na konkrétní příklad využití relačního modelu dat v databázovém systému Microsoft Access.
Klíčová slova	Relační model dat Entita Atribut Klíče Normalizace