UNIVERZITA KARLOVA V PRAZE
Filozofická fakulta
Ústav informačních studií a knihovnictví

Studijní program: informační studia a knihovnictví
Studijní obor: informační studia

Lukáš Marek

SYSTÉMY RELAČNÍCH DATABÁZÍ SE ZAMĚŘENÍM NA KNIHOVNÍ APLIKACE

DIPLOMOVÁ PRÁCE

Praha
Prosinec 2007

Vedoucí diplomové práce: Doc. RNDr. Jiří Souček, DrSc.

Oponent diplomové práce:

Datum obhajoby:

Hodnocení:

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje.

V Táboře, 5. prosince 2007

…….

Podpis diplomanta

Na tomto místě bych rád poděkoval vedoucímu své diplomové práce Doc. RNDr. Jiřímu Součkovi, DrSc. za odborné rady a námitky, které zásadním způsobem přispěly k dokončení této práce. Dále bych rád poděkoval Mgr. Aleně Otrubové, ředitelce Městské knihovny v Táboře a celému kolektivu pracovnic knihovny, za cenné rady týkající se provozu veřejné knihovny. V neposlední řadě patří poděkování mým rodičům a příbuzným za letitou oporu při studiích. Děkuji také Bc. Milanu Stejskalovi, který mě v září 2003 přivedl na myšlenku studovat na Ústavu informačních studií a knihovnictví.

Identifikační záznam

MAREK, Lukáš. *Systémy relačních databází se zaměřením na knihovní aplikace [Relational Database Systems with a view to library applications]*. Praha, 2007. 138 s. Diplomová práce. Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví 2007. Vedoucí diplomové práce Doc. RNDr. Jiří Souček, DrSc..

Abstrakt

Cílem této práce je překlenout propast mezi akademickou teorií o databázích a skutečným vytvářením databází v reálném knihovnickém světě. Práce je zaměřena na výklad informací, které je třeba vzít v úvahu, pokud se rozhodneme navrhovat databáze pro knihovní aplikace. V textu diplomové práci není vynecháno ani důležité vypracování analytické dokumentace k databázovým projektům. Stejně tak jsou popsány i možné návrhy grafických uživatelských rozhraní. Práce vychází z případů, které se vyskytují v reálném provozu knihovny, resp. v knihovnickém systému, jenž je v dnešní moderní době stěžněm každého provozu knihovny. Tyto případy se zakládají na běžných implementacích jazyka SQL.

Klíčová slova

databáze, relace, tabulka, návrh, SQL, entita, atribut, primární klíč, informační systém, databázový produkt, knihovnický systém, Z39.50

Obsah

Seznam zkratek

AACR2 - *Anglo-American cataloguing rules* - Anglo-americká katalogizační pravidla

ANSI - *American National Standards Institute* – americká standardizační organizace

AV - *Audio-Visual* - multimediální

DBMS - *Data-base Management System* - systém řízení báze dat

DFD - *Data flow Diagram* - diagram datových toků

E-R - *Entity-Relationship* - schématické znázornění množin entit, jejich atributů a vztahů mezi nimi grafickou formou

EAN - *European Article Number* - kód k označování jednotlivých druhů zboží

IS - *Information System* - informační systém

ISBN - *International Standard Book Numer* - mezinárodní standardní číslo knihy

ISO - 8859-2 - *znaková sada známá rovněž jako Latin-2*

MARC - *Machine readable cataloguing* – formát, který umožňuje podrobně popsat informační zdroj

MeSH - *Medical Subject Headings* - tezaurus termínů z lékařství a zdravotnictví

MS - *Microsoft* - softwarová firma

MS-DOS - *Microsoft Disk Operating System* - operační systém

MVS - *Meziknihovní Výpůjční Služba*

OLAP - *On-Line Analytical Processing* - technologie, která umožňuje pohlížet na data tradiční relační databáze jako na mnoharozměrnou strukturu

OPAC - *Online Public Access Catalogue* - veřejně přístupný on-line katalog

PC- *Personal Computer* - osobní počítač

PSH - *Polytematický Strukturovaný Heslář* - řízený slovník, určený pro věcný popis dokumentů v polytematických fondech

RDBMS - R*elation Data Base Management System* - relační systém řízení dat

SEQUEL -*Structured English Query* Language

SQL - *Structured Query Language* – dotazovací jazyk relačních databází

SŘBD - *Systém Řízení Báze Dat*

UNIX - *Unary Information and Computing Service* – operační systém

UTF-8 - *Unicode Transformation Format* - způsob kódování řetězců znaků do sekvencí bajtů

WINDOWS 1250 - *znaková sada používaná operačním systémem Microsoft Windows*

XML - *eXtensible Markup Languague* – značkovací jazyk

Předmluva

Při psaní této diplomové práce jsem vycházel z knižních zdrojů a z dostupných internetových zdrojů věnovaných této problematice. Neméně důležitým okruhem zdrojů byly mé vlastní zkušenosti z praxe.

Diplomová práce přináší primárně dva pohledy na problematiku relačních databází. První pohled je čistě teoretický, kdy jsou předneseny elementární vztahy, pravidla a pojmy, které se pojí s funkcí a návrhem databázových systémů. Druhý pohled se zaměřuje na praktické nasazení do praxe. Konkrétně se v této práci zaměřuji na návrh informačního systému pro malou knihovnu.

Rád bych také objasnil důvody výběru tohoto tématu. S knihovním systémem, tedy konkrétně se systémem LANIUS, jsem se poprvé setkal před 7 lety během mého rok a půl dlouhého působení v Městské knihovně v Táboře. Tehdy jsem se také začal hlouběji zajímat nejen o informační a knihovní vědu, ale především o princip fungování databázových systémů. Má potřeba prohloubení si znalostí v oblasti návrhu databází se také výrazně zvýšila při studiu na Ústavu informačních studií a knihovnictví. Během výuky jsem se nesčetněkrát setkal s termínem databáze. Není rozhodně na škodu, když je uživatel mnoha databázových center schopen nahlédnout hlouběji za hranici dialogového rozhraní a alespoň mít základní představy o tom, jak taková databáze vlastně pracuje.

K textu, jehož rozsah splňuje podmínky dané Studijním a zkušebním řádem FF UK je připojena elektronická verze práce ve formátu PDF. Použitá literatura je citována v souladu s normou ISO 690 a ISO 690-2. Citované zdroje jsou uvedeny v abecedním pořadí.

1. Úvod

Tato diplomová práce pojednává o návrhu relačních databázových systémů pro knihovní aplikace. Mým záměrem je přednést veškeré znalosti, které jsou zapotřebí pro přenesení složitých situací z reálného do efektivního, funkčního návrhu databáze.

V kapitole 2. bude popsána důležitá terminologie spojena s principy relačního modelu databáze. Budou popsány elementární tabulky, které jsou základním kamenem relačního modelu databáze pro knihovní aplikace. Tabulky sami o sobě databází nazývat nelze. Pokud však tyto tabulky provážeme konkrétními relacemi, stává se z takové architektury již relační databáze.

Pakliže známe jednotlivé relační vazby mezi tabulkami, je třeba tyto stavy umět i graficky prezentovat. Základní schématické značení sémantických vazeb mezi jednotlivými tabulkami resp. entitami, je popsáno v kapitole 3.

Navrhnout databázi, graficky znázornit vazby jednotlivých zúčastněných entit je ovšem jedna strana mince. Druhá strana mince je popsána v kapitole 4 a nese název Jazyk SQL. Zde bude představen dotazovací jazyk, který umí s daty, jenž jsou uložené v databázi, pracovat. Představena je základní terminologie relační algebry a vše je názorně předvedeno na příkladech.

Poznatky z předchozích kapitol jsou aplikovány v kapitole 5., kde je představen návrh databázového systému pro knihovnu. Tento model je pouze příkladovou studií. Nicméně vazby mezi entitami v databázi vycházejí z reálného chování knihovních systémů. Řada konkrétních logických vazeb v relačním modelu databáze bude použito v kapitole 7.

Nicméně kapitola 6. přináší přehled všech známých databázových produktů, které jsou vhodné pro různorodé požadavky uživatelů. Účelem této kapitoly není proto detailně popisovat jednotlivá databázová řešení, ale rozhodně je slušné představit alespoň jejich stručnou charakteristiku.

Kapitola 7. je zaměřena na postup při návrhu informačního knihovního systému. Beze studu mohu napsat, že se jedná o jádro celé práce. Nutno předeslat, že některé funkce systému budou jen zmíněny, neboť konkrétním návrhem, např. modulu pro generování čárových kódu knih, by se diplomová práce dostala až do příliš náročných popisů, které by mohly přesahovat rámec i několika set stran dlouhé práce.

Přehled známých automatizovaných knihovnických systémů, které se používají v České republice i v zahraničí, je popsán v kapitole 8.. Jedná se samozřejmě o přehled informačně knihovnických systémů, jejichž architektura je založena na relačním databázovém modelu.

V kapitole 9. bude zmíněn komunikační protokol Z39.50, který představuje sjednocující prvkem pro sdílení informací mezi knihovnickými systémy. Budou představeny jednotlivé komunikační kroky, atributy a formáty protokolu Z39.50.

I když v následujících, více jak sto stranách, bude hovořeno o otázkách implementace relačního databázového modelu, diplomová práce není v žádném případě nějakým manuálem pro to, jak programovat. Najdeme zde několik málo příkladů s programovým kódem. Ty jsou zde ovšem omezeny na minimum a měli by se snadno pochopit, třebaže čtenář této práce nikdy s žádným programovacím jazykem nepracoval.

1.1. Základní pojmy

Co je to tedy za záhadnou stvůru, ta relační databáze? Stručně řečeno, je to nástroj pro efektivní a spolehlivé ukládání informací a manipulace s nimi. Pod pojmem „efektivita“ a „spolehlivost“ zde přitom rozumíme ochranu dat před nahodilou ztrátou či poškozením; data nesmí dále spotřebovávat více prostředků (ať už počítačových, nebo i lidských), než kolik je nezbytně nutné a musíme je být schopni načíst rozumným způsobem při přijatelné rychlosti [RIORDAN, 2000, s. 3]. Samotná databáze je pak implementací relačního modelu, který představuje jistý způsob popisu určitých aspektů reálného světa a který vychází z množiny pravidel, zformulovaných poprvé Dr. E. F. Coddem na konci šedesátých let.

Teoreticky bychom mohli relační databázi naprogramovat „na zelené louce“ z ničeho. V reálu však budeme za normálních okolností využívat služeb vhodného systému pro správu databází (Data-base Management System, DBMS; česky se někdy uvádí pojem „systémy řízení báze dat“, SŘBD. Tomuto systému se dále často říká systém pro správu relačních databází (Relational Database Management System, RDBMS). Z technického hlediska však systém pro správu databází, který má být považován za relační, musí vyhovět souboru mnoha pravidel [RIORDAN, 200, s. 3].

Jak již bylo řečeno, relační databáze je fyzickou implementací relačního modelu (tedy datového modelu). Tyto dva pojmy je přitom velice důležité vzájemně rozlišovat. Ve fázi návrhu je téměř nemožné zcela ignorovat veškerá omezení konkrétního prostředí zvoleného pro implementaci, prvotní model by však měl být podle zásad správné praxe naopak co „nejčistší“. Při implementaci musí člověk poměrně často dělat různé kompromisy, například z důvodu rychlosti výsledného systému.

Během datového modelování však tato rozhodnutí můžeme, ba dokonce bychom je měli zcela pominout. Příkladem takovéhoto postupu je ukládání vypočtených polí (například SeznamTitulů) do základních tabulek. V návrhu relačního systému je to jeden z největších prohřešků, zatímco v praxi se jedná o zcela běžnou techniku.

2. Co je to databáze?

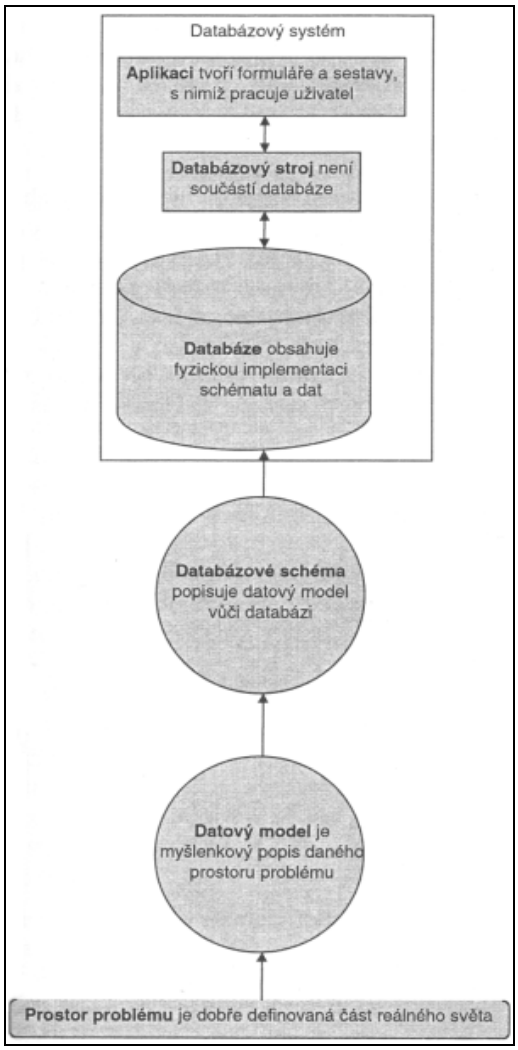
Databázová terminologie je skoro stejně ošemetná, jako pojem „objektově orientované programování“. Slovem „databáze“ můžeme totiž označit prakticky cokoliv - od jedné jednoduché množiny dat, jako je např. telefonní seznam, až po složitou množinu různých nástrojů, jako je třeba SQL Server - mezi těmito dvěma extrémy se pochopitelně nachází řada dalších možností. Tato určitá nepřesnost nemusí být nutně na závadu - takový je prostě náš přirozený jazyk [RIORDAN, 2000, s. 4]. Pro naše exaktní účely je však nevhodná a proto se pokusím naopak zavést do našeho vyjadřování jistou preciznost. Vztahy mezi jednotlivými pojmy, o kterých se bude dále hovořit, znázorňuje obr. 1 na straně 19.

Relační databáze samy o sobě sice v reálném světě nemají žádnou analogii, úkolem většiny z nich je ale modelovat jeho určitý aspekt. Tuto modelovanou část reálného světa nazýváme prostor problému. Každý prostor problému je ze své podstaty chaotický a komplikovaný - pokud by složitý nebyl, asi bychom si k němu nevytvářeli model [HERNANDEZ, 2006, s. 62]. Pro úspěch každého našeho projektu je ale životně důležité omezit navrhovaný databázový systém na konkrétní, dobře definovanou množinu objektů a jejich vzájemných vztahů. Jen tak můžeme provádět rozumná rozhodnutí ohledně záběru výsledného systému.

Pod pojmem datový model budeme rozumět myšlenkový (konceptuální) popis prostoru problému. Patří sem definice entit, jejich atributů (entitou může být např. Zákazník a jeho atributy bude tvořit Jméno a Adresa) a omezení entity (JménoZákazníka nemůže být například prázdné). Datový model zahrnuje také popis vztahů mezi

entitami a veškerá omezení platná pro tyto vztahy - jednomu nadřízenému tak např. nesmí přímo podléhat více než pět podřízených pracovníků. Datový model však ještě nemá žádnou souvislost s fyzickým rozložením výsledného systému.

Definice fyzického rozvržení systému tedy, seznam implementovaných tabulek a pohledů, představuje takzvané databázové schéma neboli stručně jen schéma. Vzniká z myšlenkového modelu a to převedením do fyzické reprezentace, kterou je již možné implementovat ve zvoleném systému pro správu databází. Všimněte si, že i schéma je stále myšlenkové (čili symbolické), nikoliv fyzické. Schéma není nic víc než jen opět datový model, tentokrát vyjádřený v pojmech, pomocí kterých jej popisujeme vůči databázovému stroji - tvoří ho tedy tabulky, spouště a další podobné entity. Jednou z výhod každého databázového stroje je, že se nikdy nemusíme zabývat fyzickou implementací. Jakmile databázovému stroji vysvětlíme, jak mají data podle naší představy vypadat (ať už pomocí programového kódu, nebo v nějakém interaktivním grafickém prostředí, jaké má například Microsoft Access), vytvoří stroj určité fyzické objekty (obvykle je umístí na určité místo pevného disku, nemusí to ale být pravidlem), do nichž posléze začneme ukládat data [RIORDAN, 200, s. 4]. Databází zde budeme nazývat sjednocení takto vytvořené struktury a vlastních dat. To znamená, že databázi tvoří fyzické tabulky, dále definované pohledy, dotazy a uložené procedury a konečně pravidla, jejichž „vynucováním“ bude databázový stroj zajišťovat ochranu dat.



Obr. 1: Schematicky znázorněná terminologie relačních databází [RIORDAN, 2000, s. 5]

Pojem „databáze“ přitom nezahrnuje samotnou aplikaci, která se skládá z formulářů a sestav s nimiž posléze pracuje uživatel, ani neobsahuje různé softwarové součásti, jež propojují obě části front-end a back-end, tedy např. různý middleware nebo Microsoft Transaction Server. Slovo „databáze“ ve svém významu neobsahuje také databázový stroj. Soubor mdb z Microsoft Accessu je tedy databází, zatímco Microsoft Jet je databázový stroj [RIORDAN, 2000, s. 5].

2.1. Dokumenty ve formě tabulek

Mnohé dokumenty obsahují data ve formě tabulek. Jako příklad je možné uvést seznam osob (čtenářů) ve formě klasické tabulky, tak jak ji můžeme vytvořit v programu MS Excel.

ID číslo čtenáře	Příjmení	Jméno	Datum nar.
7707	Novák	Karel	11.5.1972
7708	Urban	Ignác	12.12.1965
7709	Kocáb	Tibor	11.11.1969
7710	Kubeš	Jaroslav	4.4.1954

Tabulka I: Seznam čtenářů

Tabulka I obsahuje záhlaví s názvy jednotlivých sloupců. Pod záhlavím jsou v řádcích uložené jednotlivé položky, v našem konkrétním případě data o čtenářích.

Podobně také data v relačních databázích jsou z hlediska logické struktury uložena v dvojrozměrných databázových tabulkách, tedy v podobě známé z tzv. „papírové informatiky“. Struktura databázové tabulky je velmi jednoduchá - všechna data jsou uložena v jednotlivých polích tabulky. Pole tabulky jsou uspořádána do řádků, které budeme označovat jako záznamy a sloupců, jež se někdy označují také jako atributy.

2.1.1. Databázové tabulky v relační databázi

Pojem databázová tabulka je jednoduchý a snadný k pochopení. V hrubých rysech se podíváme na podmínky „relačnosti“, které definoval na přelomu šedesátých a sedmdesátých let minulého století Dr. E. F. Codd [LACKO, 2007, s. 15].

Minimální podmínky:

1. Všechna data v databázi jsou uložena v tabulkách.

2. Fyzická struktura dat a jejich uložení je nezávislá a od uživatele odstíněná (tedy neexistují pro uživatele viditelné přístupové cesty - včetně indexů).
3. Předpokládá se existence databázového jazyka, který umožňuje realizovat minimálně operace selekce, restrikce, projekce a spojení.

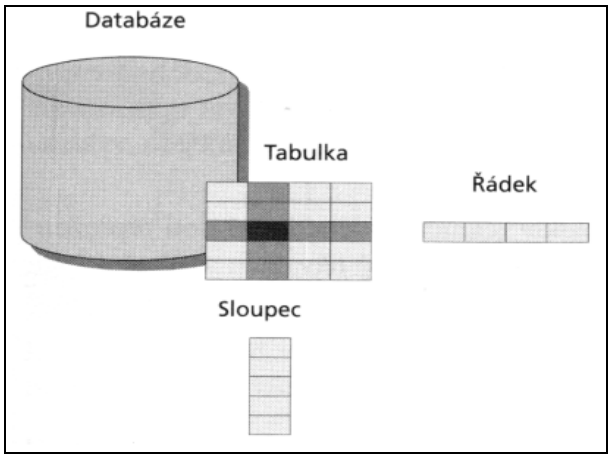
Vyjma těchto třech základních podmínek definoval Dr. E. F. Codd také **dvanáct pravidel pro relační databázové systémy**, které si uveďme ve zjednodušené podobě [LACKO, 2007, s. 15]:

1. Data v relační databázi musí být reprezentována explicitně na logické úrovni pomocí relačních tabulek.
2. Data uložená v relační databázi musí být přístupná pomocí kombinace názvu tabulky, názvu sloupce a hodnoty primárního klíče.
3. Musí existovat identifikátor chybějící hodnoty nebo neznámé hodnoty (odlišný od čísla „nula“ a prázdného řetězce). Tento identifikátor označujeme pojmem NULL.
4. Databáze musí umožňovat autorizovaným uživatelům přístup k datům i jejich popisům (metadatům) pomocí stejného databázového jazyka.
5. Databázový jazyk musí být jednoduchý a uživatelsky přívětivý, přičemž musí umožňovat definici dat, entitních omezení, manipulaci s daty, definici transakcí a přístupových práv.
6. Relační databázový systém musí umožňovat definování pohledů Musí pro tyto pohledy povolit či zakázat vkládání, mazání či aktualizaci záznamů v základních tabulkách, nad kterými jsou tyto pohledy definovány.

7. Relační databázový systém musí umožňovat množinové operace s celými tabulkami a to nejen při vyhledávání, ale také při vkládání, aktualizaci a mazání dat.
8. Aplikační logika nesmí vyžadovat modifikaci v případě změny interního uložení nebo způsobu přístupu k datům. Tento trend definuje fyzickou datovou nezávislost.
9. Aplikační logika nesmí vyžadovat modifikaci v případě změn základních tabulek, nezpůsobujících ztrátu informace (zrušení či přidání sloupce do tabulky). Tento bod definuje logickou datovou nezávislost.
10. Aplikační logika nesmí vyžadovat modifikaci v případě změn integritních omezení definovaných pomocí databázového jazyka a uložených v systémovém katalogu.
11. Aplikační logika nesmí vyžadovat modifikaci v případě, kdy jsou data distribuována na různých počítačích.
12. Pokud disponuje databázový systém nízkoúrovňovým (procedurálním) programovacím jazykem, nesmí být tomuto jazyku umožněno rušit či měnit omezení definovaná databázovým jazykem.

[LACKO, 2007, s. 15]

Ale dost již bylo teorie, pojďme se věnovat databázovým tabulkám. Mnohem výstižnější, než rozsáhlý odstavec s popisem organizace databázových tabulek, je názorný na obr. 2.



Obr. 2: Struktura dat v databázi [LACKO, 2003, s. 9]

Řádek představuje kombinaci hodnot sloupců v tabulce. Matematicky zaměření teoretici by ji dokonce definovali jako vektor. Při správném návrhu tabulky pro relační databázi by každý řádek měl být identifikovatelný pomocí primárního klíče. **Primární klíč** je tedy sloupec či skupina sloupců, která slouží pro jednoznačnou identifikaci každého řádku v tabulce. [18] Hodnota pole/polí primárního klíče musí být v rámci tabulky jedinečná. V našem příkladu (dle tabulky I) jsme použili identifikační číslo čtenáře. Bez primárního klíče vlastně ani není možné definovat relace mezi jednotlivými tabulkami.

Sloupec je množina dat jednoho typu v tabulce. Sloupce vlastně představují atributy objektů. Každý sloupec má svůj název a obsahuje hodnoty stejného datového typu.

Cizí klíč je sloupec či skupina sloupců, které jsou propojeny na primární klíč v jiné tabulce. Tím je provázána tabulka III na tabulku II.

ID číslo čtenáře	Příjmení	Jméno	Datum nar.	Kód oddělení
7707	Novák	Karel	11.5.1972	3
7708	Urban	Ignác	12.12.1965	1
7709	Kocáb	Tibor	11.11.1969	2
7710	Kubeš	Jaroslav	4.4.1954	3

Tabulka II: Seznam čtenářů rozšířený o další klíč

Kód oddělení	Název oddělení v knihovně
1	Dospělé
2	Dětské
3	Hudební

Tabulka III: Kódovník pro identifikaci oddělení v knihovně

Již v případě této jednoduché tabulky (tabulka III) vidíme, že zavedení relací mezi tabulkami minimalizuje objem dat uložených v databázi.

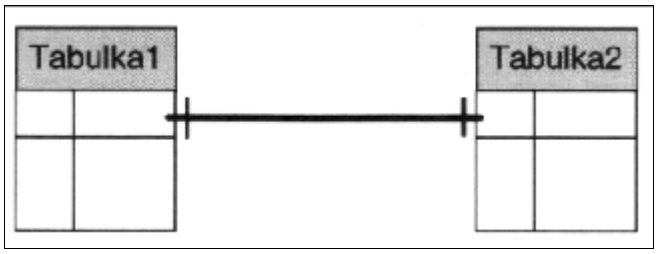
2.2. Relace mezi databázovými tabulkami

V následujících odstavcích se budeme věnovat trochu teorii. Mnoho čtenářů pravděpodobně v tento okamžik namítne, že to nemá žádný smysl a proto bychom se měli raději věnovat optimalizaci databázového serveru. Správně navržené databázové struktury jsou ale alfou a omegou výkonu a efektivnosti každé databázové aplikace.

Relace mezi tabulkami v podstatě popisují vztahy mezi objekty reálného světa, které jsou reprezentovány právě těmito tabulkami. Při návrhu databázových tabulek na které navazuje aplikační logika můžeme definovat několik druhů relací [LACKO, 2003, s. 10]. V následujících popisech je možné podle typu relace uvažovat také o povinném spojení, tedy „musí být svázán“.

2.2.1. Relace jedna ku jedné (1:1, one-to-one)

Každý řádek primární tabulky je možné svázat s právě jedním řádkem sekundární tabulky (viz obr. 3). Takovou relaci zajistíme pomocí jedinečných klíčů v obou tabulkách.

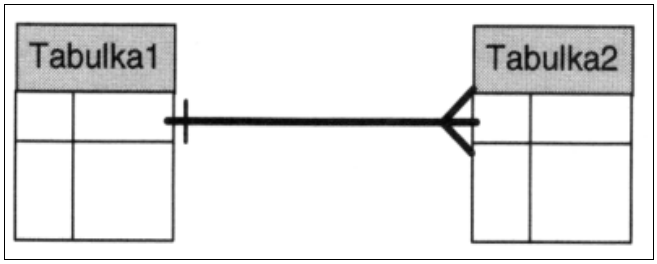


Obr. 3: Schématické znázornění vztahu 1:1 [LACKO, 2007, s. 17]

Relaci tohoto typu můžeme vysvětlit na příkladu vztahu řidič - automobil. Jeden řidič může současně řídit jen jeden automobil a právě jeden automobil může být v daném okamžiku řízen jen jedním řidičem. Dalším příkladem „skutečně ze života“ je manželství - v západní křesťanské kultuře je manželství definované jako vztah mezi jedním mužem a jednou ženou [LACKO, 2003, s. 10]. Posledním příkladem vztahu 1:1 nám poslouží vazba mezi čtenářem knihovny a exemplářem knihy v knihovně. Jeden konkrétní exemplář knihy může být zapůjčen pouze jedním čtenářem.

2.2.2. Relace jedna ku více (1:N, one-to-many)

Každý řádek primární tabulky je možné svázat s jedním či více řádky sekundární tabulky (viz obr. 4).



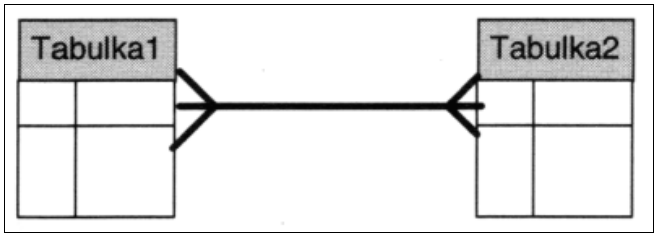
Obr. 4: Schématické znázornění vztahu 1:N [LACKO, 2007, s. 17]

Opět si uveďme jako příklad vztah z dopravy: autobus-cestující. V autobuse se může v jednom okamžiku nacházet více cestujících, nicméně jeden cestující nemůže být ve stejném okamžiku ve více autobusech [LACKO, 2003, s. 11]. Nebo jiný příklad - jeden čtenář si

může současně vypůjčit více knih, ale jednu knihu si v daný okamžik může vypůjčit jen jeden čtenář.

2.2.3. Relace více ku více (N:M many-to-many)

Více řádků primární tabulky může být svázáno s více řádky sekundární tabulky, jak ukazuje obr. 5.



Obr. 5: Schématické znázornění vztahu N:M [LACKO, 2003, s. 11]

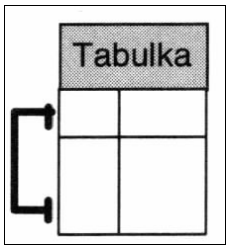
V praxi se tento vtah realizuje pomocí vazební (spojovací) tabulky, což znamená, že vztah N:M se rozloží na dva vztahy 1:N – konkrétně 1:N:M:1 (viz obr. 6)

Obr. 6: Schématické rozložení vztahu pomocí vazební tabulky [LACKO, 2003, s. 11]

Tento vztah je možné v praxi pozorovat například mezi výrobky a vlastnostmi výrobku. Jeden výrobek může mít více vlastností a jedna vlastnost může být přiřazena více výrobkům [LACKO, 2007, s. 18]. Nebo jiný příklad - relace N:M vznikne tak, že jeden autor může napsat více knih a na druhé straně jednu knihu může napsat více autorů současně (kolektiv autorů).

2.2.4. Unární relace

Doposud byly v přehledu uváděni jen vztahy mezi dvěma tabulkami. V praxi i v příkladech (tabulka II) se vyskytují také relace jedné tabulky se sama sebou. Na obr. 7 je pouze jedna tabulka s neurčeným počtem řádků.



Obr. 7: Unární relace [LACKO, 2003, s. 11]

Pomocí tohoto typu relace se velmi často vyjadřuje hierarchický vztah nadřízený - podřízený. Sloupec tabulky může obsahovat vazbu na primární klíč jeho nejbližšího nadřízeného. Jedno pole v tomto sloupci v některém řádku tabulky je prázdné. Jedná se o nejvyššího nadřízeného.

2.2.5. Normalizace databází - normální formy

Proces zjednodušování databáze se nazývá normalizace, která vede k odstranění redundancí a značně zefektivňuje práci s tabulkami. V případě relačních databází je možné konstatovat, že čím jsou tabulky ve vyšších normálních formách, tím lépe by se s nimi z hlediska aplikační logiky mělo pracovat. Ne vždy toto tvrzení musí platit. Existují situace, kdy se od normalizace vyšších forem upouští, např. za účelem zvyšování výkonu ve speciálních případech nebo při návrhu datových skladů [LACKO, 2007, s. 19]. Pro tabulky používané při analýzách OLAP ale tato pravidla nerespektujeme. Vzhledem k tomu, že se zatím pohybujeme v oblasti relačních databází, měli bychom dodržovat alespoň první a druhou normální formu.

2.2.6. Nultá normální forma (0NF)

Tato normální forma představuje z hlediska dotazů pomocí jazyka SQL prakticky neřešitelné rébusy. Tabulka IV je v nulté normální formě tehdy, pokud se skládá alespoň z jednoho pole, které obsahuje více než jednu hodnotu, tedy není atomické.

Jméno čtenáře	Adresa
Josef X. Drda	Husinecká 2112, 390 02 Tábor
Marta Nováková	Kpt. Nálepky 164, Tábor, 390 03
Koudelka Josef	Souběžná 9, 390 01 Tábor

Tabulka IV: Seznam čtenářů v knihovně

Na první pohled se zdá, že tabulka IV je navržena přehledně a celkem správně. Zkuste si například vybrat všechny Josefy. Nebude to triviální, protože sloupec *Jméno čtenáře* obsahuje jak jméno tak i příjmení, navíc v nedefinovaném pořadí. Ale co v případě, pokud bychom na základě dat ve sloupci Adresa měli tisknout adresní štítky s tím, že PSČ by mělo být na předem definovaném místě? Tento problém řeší následující kapitola.

2.2.7. První normální forma (1NF)

Tabulka splňuje podmínku první normální formy tehdy, pokud všechny sloupce (atributy) jsou atomické, tedy již dále nedělitelné. Jeden sloupec tak nesmí obsahovat více druhů dat. Matematicky řečeno - každý sloupec musí pro daný záznam obsahovat jen skalární hodnotu. Hodnota sloupce nesmí být relace. Správně navržená, než předchozí tabulka IV, je tabulka V, která by splňovala podmínky pro 1NF, bude mít tvar:

Příjmení	Jméno	Ulice	PSČ	Město
Drda	Josef X.	Husinecká 2112	390 02	Tábor
Nováková	Marta	Kpt. Nálepky 164	390 03	Tábor
Koudelka	Josef	Souběžná 9	390 01	Tábor

Tabulka V: Správný tvar Seznamu čtenářů v knihovně

Vidíme, že v této tabulce jsou všechny atributy z praktického pohledu již dále nedělitelné. Teoreticky by se ještě dalo rozdělit například jméno na dvě části, ale jedná se o málo běžný případ, který je možné zanedbat. Podobně bychom mohli rozdělit sloupec Ulice na Název ulice a Číslo popisné. Praktický význam ale ani toto dělení zpravidla nemá - číslo popisné píšeme v adrese takřka vždy za název ulice. Jiným příkladem tabulky v 1NF by mohl být seznam zapůjčených titulů knih (viz tabulka VI).

ID číslo čtenáře	Příjmení	Zápůjčky
4500	Drda	Zvukař amatér
9700	Nováková	Řeč těla, O smyslu života, Osobní údaje, Starověký Řím
3200	Koudelka	Pravidla českého pravopisu, Polovodičové součástky Revolvery a pistole

Tabulka VI: Seznam zápůjček čtenářů

Pokud bychom na tuto tabulku automaticky aplikovali postup, který se nám osvědčil v předchozím příkladě, získali bychom tabulku, která je teoreticky v naprostém pořádku a vyhovuje podmínkám 1NF (viz tabulka VII).

ID čtenáře	Příjmení	Zápůjčka 1	Zápůjčka 2	Zápůjčka 3	Zápůjčka 4
4500	Drda	Zvukař amatér			
9700	Nováková	Řeč těla	O smyslu života	Osobní údaje	Starověký Řím
3200	Koudelka	Pravidla českého pravopisu	Polovodičové součástky	Revolvery a pistole	

Tabulka VII: Omezení pouze na čtyři výpůjčky

Bystré oko čtenáře ale již určitě zaregistrovalo, kde je zakopaný pes. Takto navržená tabulka (tabulka VII) umožňuje zápůjčku jen čtyř položek. Pokud má čtenář v úmyslu zapůjčení více titulů, tak při takto navržené tabulce to není možné. Řešením je dekompozice tabulky na dvě jednodušší, mezi kterými je definovaný vztah master-detail.

Po dekompozici vznikne tabulka čtenářů (master)

ID čtenáře	Příjmení
4500	Drda
9700	Nováková
3200	Koudelka

Tabulka VIII: Vazba identifikačního čísla čtenáře a příjmení čtenáře

a tabulka detail (tabulka IX).

ID čtenáře	Zápůjčka
4500	Zvukař amatér
9700	Řeč těla
9700	O smyslu života
9700	Osobní údaje
9700	Starověký Řím
3200	Pravidla českého pravopisu
3200	Polovodičové součástky
3200	Revolvery a pistole

Tabulka IX: Vazba identifikačního čísla čtenáře a zápůjček

Při takto navržených strukturách si může každý čtenář půjčit tolik titulů, kolik sám chce. Vztah master-detail známe také z reálné knihovnické praxe - mnoho protokolů a formulářů o zápůjčkách (papírových i elektronických) je navržených tak, že v záhlaví formuláře jsou základní informace, např. datum a jméno čtenáře. Zbytek formuláře potom obsahuje seznam jednotlivých zápůjček. Z našich tabulek bychom dokázali vytisknout tyto doklady pro jednotlivé čtenáře velmi jednoduše. Příklady jsou v tabulkách X až XII.

Seznam výpůjček
Čtenář: 4500 Jméno: Drda
Zvukař amatér

Tabulka X: Jednoduchý výpis seznamu výpůjček čtenáře Drdy

Seznam výpůjček
Čtenář: 9700 Jméno: Nováková
Řeč těla
O smyslu života
Osobní údaje
Starověký Řím

Tabulka XI: Jednoduchý výpis seznamu výpůjček čtenářky Novákové

Seznam výpůjček Čtenář: 3200 Jméno: Koudelka
Pravidla českého pravopisu
Polovodičové součástky
Revolvery a pistole

Tabulka XII: Jednoduchý výpis seznamu výpůjček čtenáře Koudelky

2.3. Proces návrhu databáze

Struktura dat je pouze jednou z mnoha součástí databázového systému - je to samozřejmě velice důležitá součást, přesto však je jedna z mnoha. V této části se proto podíváme na některé další aspekty návrhu databázových systémů.

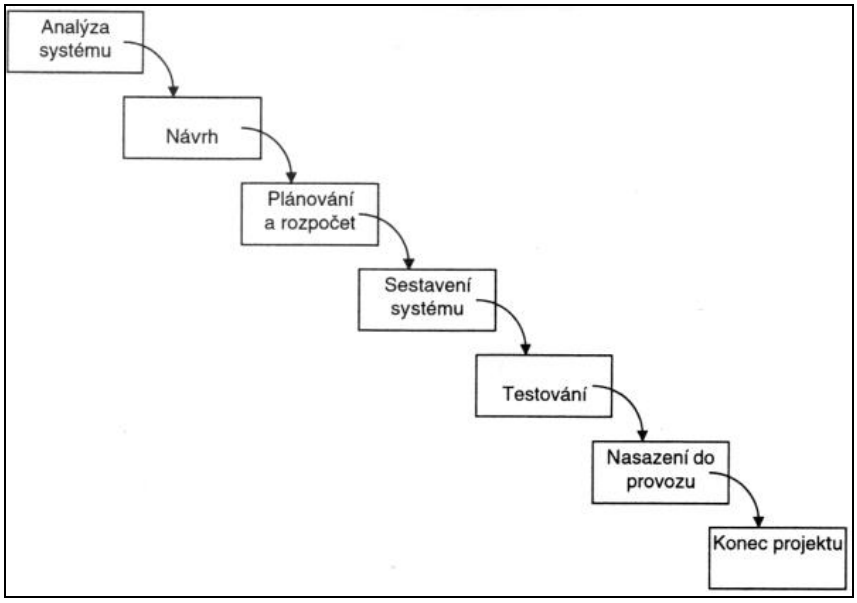
Tato část diplomové práce bude tedy rozebírat většinu činností spojených s analýzou a návrhem databázových systémů, tedy například definici systémových parametrů a pracovních procesů, myšlenkový databázový model a databázové schéma. Návrh uživatelského rozhraní je natolik složité téma, že by vystačilo na samostatnou diplomovou práci.

Zde se tedy bude hovořit pouze o analýze a návrhu databázových systémů. Analýza a návrh se ale nedají od zbytku celého procesu oddělit a nemohou existovat samy o sobě, izolovaně. Pro začátek si proto řekneme něco málo o životním cyklu projektů [HERNANDEZ, 2006, s. 58].

2.3.1. Modely životního cyklu

Kdysi dávno postupovali systémoví analytici při vývoji systémů metodou známou jako „vodopádový model“. Ten existuje v několika různých verzích. Jednu z nich, relativně jednoduchou, ukazuje obr. 8.

Celý proces začíná tedy analýzou systému, které se někdy říká analýza požadavků, protože se zaměřuje na to, co zákaznická organizace a její uživatelé od systému potřebují. Jakmile je analýza systému dokončena a odsouhlasena, vytvoří se podrobný návrh celého systému. Po této fázi následuje plánování a tvorba rozpočtu projektu. Poté již zbývá systém sestavit, otestovat a nasadit do zkušebního provozu. [17]



Obr. 8: Vodopádový model [RIORDAN, 2000, s. 98]

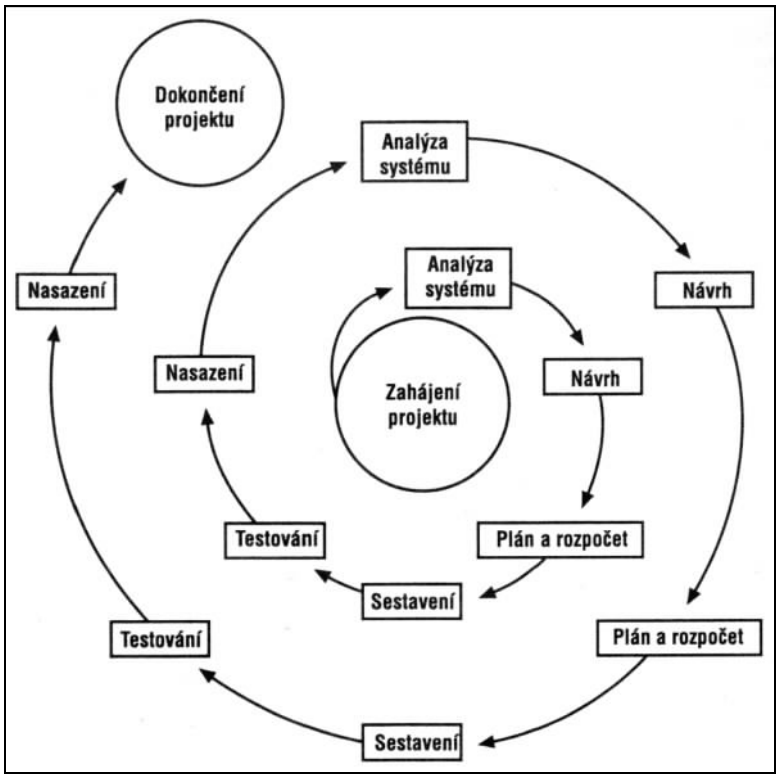
Vodopádový model našemu oku zajisté lahodí. Každá činnost se pěkně dokončí a odsouhlasí ještě před započetím další, přičemž model sleduje rozumnou kontrolu nad rozpočtem, nasazením pracovníků a časem. Jestliže se vám podaří vodopádový model nasadit včas s dodržením původního finančního plánu, bude vše v pořádku.

Tento model v sobě ale skrývá jeden zásadní problém - realita se většinou nedá tak pěkně „nalinkovat“. Model předpokládá, že během zpracování určitého úkolu máme pro něj k dispozici veškeré potřebné informace a nedovoluje pozdější vstup žádných dalších informací do projektu. Tato situace je, snad jen s výjimkou opravdu malých systémů, mimořádně nepravděpodobná.

Vodopádový model neuvažuje také v průběhu řešení projektu žádné změny aplikačních požadavků. Předpokládat, že systém, jež vyhovuje potřebám organizace někdy na začátku projektu, bude vyhovovat i na konci dvouletého nebo tříletého vývoje, je ale čiré bláznovství [RIORDAN, 2000, s. 98].

Na tomto místě si ale musíme říci, že veškeré činnosti, které jsme identifikovali ve vodopádovém modelu, jsou stanoveny naprosto správně. Vynecháním libovolné z nich si ve vývojovém projektu zadělváme na téměř jistý problém. Problémem tohoto modelu je tedy jeho linearita, neboli česky řečeno chybný předpoklad, podle něhož do jednou ukončené fáze vývoje nebudeme muset nikdy vstupovat znovu.

Pro řešení uvedených problémů vodopádového modelu bylo navrženo několik alternativních modelů životního cyklu projektu. Spirálový model tak v podstatě předpokládá několik iterací (opakování) činností z vodopádu; záběr každé další iterace se oproti předchozí rozšiřuje, až se dostaneme k dokončení projektu [RIORDAN, 2000, s. 99]. Schéma spirálového modelu ukazuje obr. 9.

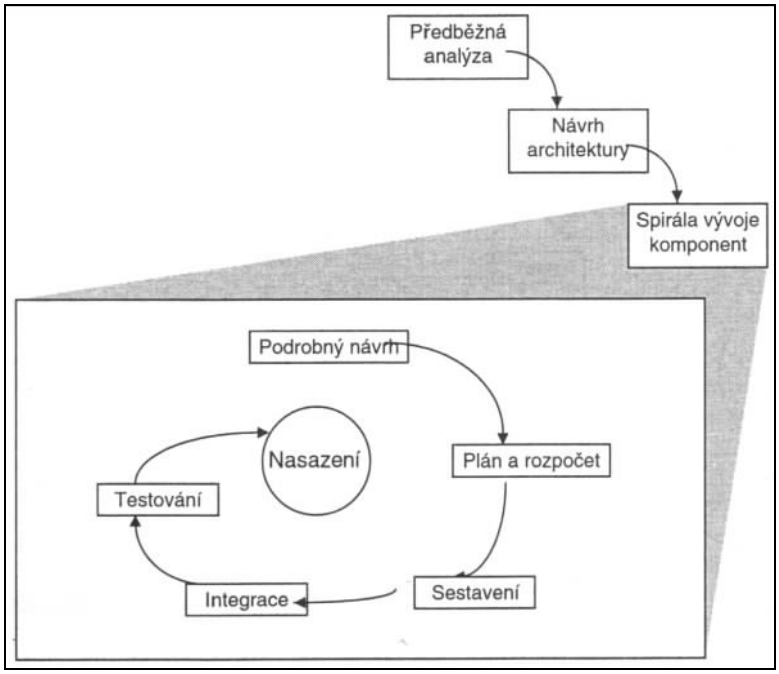


Obr. 9: Spirálový model [RIORDAN, 2000, s. 99]

Problém spirálového modelu spočívá v tom, že pokud se jím budeme striktně řídit, vezmeme celkovou strukturu a záběr projektu v úvahu až příliš pozdě, takže hrozí určité (a to nikoli bezvýznamné) riziko, že při pozdějších iteracích přijdou výsledky dřívější práce vniveč.

U větších systémů se doporučuje jiný model, kterému se říká inkrementální vývojový model nebo evoluční vývojový model; vidíme jej na obr. 10. V tomto modelu, který je do značné míry jistou variací na předcházející spirálový model, se nejprve provede předběžná analýza celého systému a nikoli pouze jeho části. Po této etapě následuje návrh architektury a to opět celého systému. [17] Cílem návrhu architektury je přitom zejména definovat jednotlivé komponenty systému, které se již budou implementovat víceméně nezávisle a popsat komunikace a vzájemné závislosti mezi těmito komponentami. Podrobný návrh a implementace každé z komponent se pak provádí podle libovolného

vhodného vybraného modelu. Proto je do obr. 10 zanesen v této fázi spirálový model, protože znamená pro návrh a implementaci vyšší flexibilitu.



Obr. 10: Inkrementální vývojový model [RIORDAN, 2000, s. 100]

Všimněme si, že spirála obsahuje v tomto schématu jeden úkol navíc, a sice integraci. Ve spirálovém modelu je integrace komponent samozřejmě také obsažena (implicitně), podle mých vlastních zkušeností je ale právě v inkrementálním vývojovém modelu integrace o něco složitější. To je také jeden z důvodů, proč se při vývoji komponent používá raději spirálový model. Jestliže podrobný návrh komponenty odložíme až na okamžik, kdy se skutečně pustíme do jejího vývoje, pak do ní můžeme zapracovat veškeré poznatky, které jsme získali během integrace předcházejících komponent. Snad se tak můžeme vyhnout i problémům, s jakými jsme se při integraci setkali.

Inkrementální vývojový model předpokládá, že se každý rozsáhlejší systém dá dekomponovat neboli rozložit do určitých samostatných komponent.

Analýza systému a návrh architektury se provádí již na začátku projektu a proto je zde určité riziko, že později zastarají a budou neaktuální; to je ovšem, jednou z hlavních nevýhod vodopádového modelu. Z toho důvodu je důležité uvedené dva kroky -a zejména pak analýzu požadavků - protože u té je větší „šance“ na případné změny - ještě před zahájením podrobného návrhu jednotlivých komponent opravdu důkladně prověřit a zrevidovat. Často se takto můžeme dočkat velice příjemného překvapení, jak snadno se dají změny v požadavcích promítnout do změn ještě nenavržených komponent; někdy dokonce stačí změnit jen pořadí, v jakém budeme nové komponenty vyvíjet, takže žádná část naší dříve provedené práce nepřijde vniveč [RIORDAN, 2000, s. 100].

Inkrementální vývojový model má ale několik důležitých výhod. Na začátku projektu nadefinujeme totiž jen „celkový pohled“ na nový systém, takže riziko promarněné práce je v něm opravdu minimální. Protože rozsáhlé projekty se rozkládají clo menších komponent, dají se projekty těchto menších jednotlivých komponent lépe udržovat a spravovat. Díky rozdělení systému do komponent můžeme některé základní funkce systému nabídnout uživatelům již v časných stádiích vývoje projektu. Takto začíná systém dosti brzo přinášet kýžené ovoce a současně se do něj dostává mechanismus, pomocí něhož do dalších vývojových prací zapojíme přirozenou cestou i samotné uživatele.

2.3.2. Postup návrhu databáze

V každém zvoleném modelu obecného vývoje musíme provést určité činnosti spojené s analýzou a návrhem. Nezáleží na tom, jestli je provedeme postupně nebo opakovaně, jestli pracujeme nad celým systémem nebo v jediné jednotlivé komponentě, jestli používáme

formální nebo neformální postupy; v každém projektu musíme každým z těchto kroků alespoň jednou projít.

2.3.3. Definice parametrů systému

V ideálním případě by měly práce na každém projektu začít jasnou definicí cílů, kterých chceme dosáhnout, a stanovením, proč jich chceme dosáhnout. Pak už by nám měl být souzen pouze úspěch. Většina projektů ale žádnou takovouto definici před zahájením prací nemá; k tomu tedy slouží právě tato první fáze procesu návrhu [RIORDAN, 2000, s. 104]. Cíle projektu říkají, „proč“ jej řešíme. Na základě cílů si uvědomíme, „co“ máme udělat a stanovíme tak záběr budoucího systému. Jakmile porozumíme cílům projektu a jeho záběru, dokážeme již reálně určit kritéria návrhu a víme tedy, „jak“.

2.3.4. Definice pracovních procesů

Databázové systémy zabezpečují na první pohled jen operace spojené s ukládáním a načítáním dat, ve skutečnosti však většina z nich podporuje jeden nebo více pracovních procesů. Uživatelé neukládají data jen tak pro nic za nic; chtějí je nějakým způsobem využívat [RIORDAN, 2000, s. 119]. Správně porozumět pracovním procesům, které mají naše data podporovat, je životně důležité pro správné pochopení sémantiky datového modelu.

3. Návrh konceptuálního schématu

Od počátku 70. let se objevují pokusy o zachycení sémantiky (významu) dat ukládaných v informačních systémech. Vznik prakticky použitelného sémantického modelu umožňujícího popsat konceptuální úroveň databáze je svázán s tzv. Entity-Relationship modelem, který se také zkráceně označuje jako E-R model. [28] Jde o schématické znázornění množin entit, jejich atributů a vztahů mezi nimi grafickou formou. Používané grafické symboly ukazuje obr. 11.

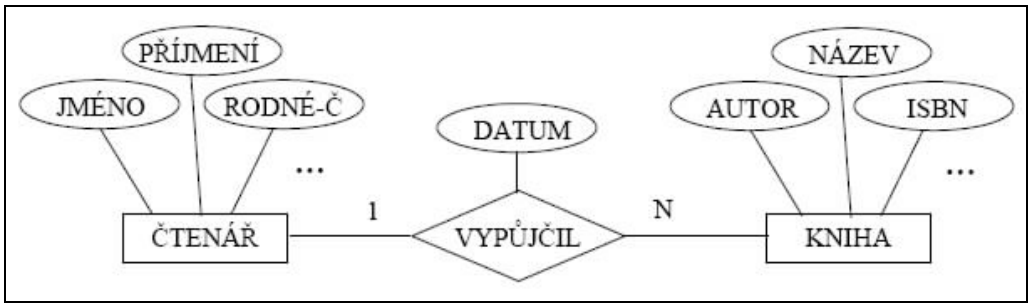
Grafický symbol	Význam symbolu
	obdélník reprezentuje entity nebo množiny entit
	kosočtverec znázorňuje vztahy mezi entitami nebo množinami entit
	oválné uzly reprezentují atributy
	čáry jsou použity pro spojení symbolů
1	znak 1 je použit pro označení jednoduchého výskytu ve vztahu
M, N, …	M, N, atd. označují vícenásobný výskyt ve vztahu

Obr. 11: Grafické symboly E-R modelů

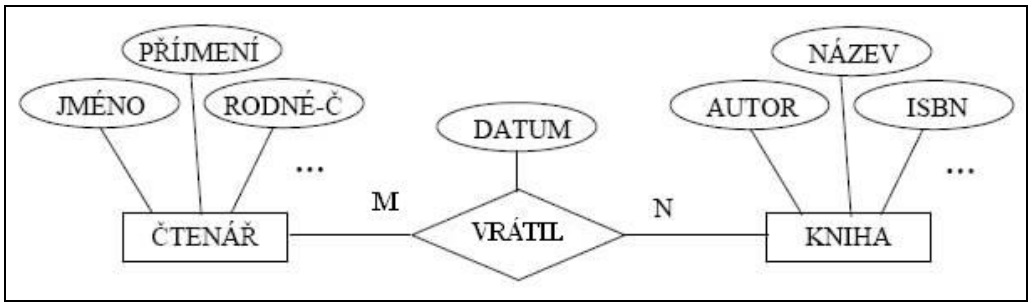
Postup při vytváření E-R modelu je následující [FARANA, 2003]:

1. Určí se a pojmenují zobrazované objekty z reality a vztahy mezi nimi, které nás budou zajímat.
2. Rozhodne se o rozdělení objektů na entity a hodnoty.
3. Definičním oborům vztahů mezi entitami a atributům se podle potřeby přiřadí nová jména.
4. Stanoví se identifikátory entit.
5. U vztahů mezi entitami se určí jejich typ (1:1, 1:N, M:N).
6. Sestrojí se grafické zobrazení modelu.

Na obr. 12 je vyjádřen vztah mezi čtenáři a knihami v knihovně. Tento vztah je typu 1 : N, protože každý čtenář může mít vypůjčeno více knih a naopak každá kniha může být v každém okamžiku vypůjčena nejvýše jedním čtenářem. Na obr. 13 E-R diagramu jde o vztah M : N mezi čtenářem a knihou, kterou čtenář vrací do knihovny. Jeden čtenář může vracet více knih a přitom jeden knižní titul může být v knihovně ve více exemplářích. Vidíme, že vztah může mít svůj vlastní atribut, v našem případě, povinnost čtenáře vracet knihy do knihovny.



Obr. 12: E-R model se vztahem 1 : N

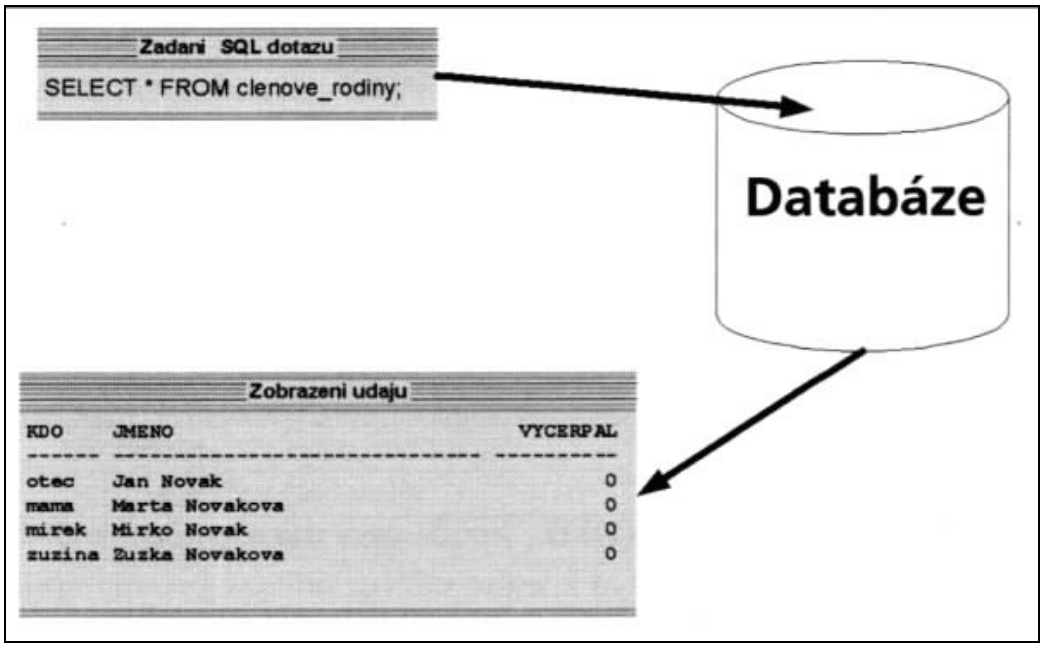


Obr. 13: E-R model se vztahem M : N

4. Jazyk SQL

Databázový jazyk SQL (Structured Query Language) vznikl na základě projektu společnosti IBM, tehdy se ovšem ještě jmenoval SEQUEL (Structured English Query Language). Cílem projektu bylo vytvořit jazyk blízký angličtině pro práci s daty v databázích. Později na vývoji databázového jazyka SQL spolupracovaly také další firmy, přičemž komerčně jej uvedla do praxe společnost Oracle. Postupem času se ujaly vylepšené a upravené standardy tohoto jazyka s označeními SQL 86 a SQL 92. Pro verzi SQL 92 se vžilo zkrácené označení SQL 2. V současné době se již několik let pracuje na novém standardu s názvem SQL 3 [LACKO, 2003, s. 77].

SQL je dotazovací jazyk, přičemž základní princip komunikace s relačním databázovým serverem je možné nejjednodušeji vyjádřit obrázkem č. 14.



Obr. 14: Princip zpracování příkazu jazyka SQL [LACKO, 2003, s. 77]

Řečeno jednou větou - klient zformuluje a položí svůj dotaz a databázový server na něj odpoví, obvykle tím, že vygeneruje nějakou množinu výstupních dat. Tento princip komunikace s databázovým serverem je velmi jednoduchý a efektivní [LACKO, 2003, s. 77]. Samozřejmě ale jen z pohledu uživatele. Jazyk SQL totiž připomíná klasický přirozený jazyk (anglický), má ale přesně definovaná syntaktická a lexikální pravidla. Z pohledu serveru se příkaz SQL přenese, dekóduje, optimalizuje a provede. Podrobné schéma tohoto mechanismu by ale bylo velmi složité. Také probírat jednotlivou syntaxi příkazů jazyka SQL by bylo tématem spíše pro studenty informatiky.

4.1. Relační algebra

Nicméně je třeba s ohledem na následující příklady uvést alespoň několik základních vztahů. V relační algebře, resp. pro praktické užití v předvedeném návrhu knihovního systému, je třeba znát alespoň pět níže uvedených vztahů. Základem je Relační datový model, který odděluje data chápaná jako relace od jejich implementace. Přístup k datům je symetrický, tj. při manipulaci s daty se nezajímáme o jejich přístupové mechanizmy. Pro manipulaci využíváme dva silné prostředky – relační kalkul a relační algebru. Pro omezení redundance dat v relační databázi jsou navrženy postupy umožňující normalizaci relace [KALUŽA, 2004].

Definice Relačního datového modelu říká: mějme množiny D1,D2...Dn. Z každé vybereme 1 prvek. Tím vytvoříme uspořádanou n-tici. Kartézský součin $D1 \times D2.... \times ....Dn$ je množina všech posloupností. Z hlediska databázového systému se množiny D označují jako množiny hodnot atributů, tzv. domény.

Od matematické relace se databázová poněkud liší:

- relace je vybavena pomocnou strukturou, které se říká schéma relace; schéma relace se skládá ze jména relace a jmen atributů a domén
- prvky domén, ze kterých se berou jednotlivé komponenty prvků relace jsou atomické (dále nedělitelné) hodnoty – jde o tzv. 1 normální formu

[KALUŽA, 2004]

Relační algebra je nástrojem pro manipulaci s relacemi Je to jazyk, který pracuje s celými relacemi, operátory relační algebry se aplikují na relace a výsledkem jsou opět relace. Pojďme se podívat na některé další pojmy z této oblasti.

Projekce umožňuje potlačit atributy v relaci, umožňuje přejít z relace o n sloupcích na relaci o p sloupcích. Tento algebraický výraz představuje v jazyce SQL příkaz *PROJECT*.

Selekcí *(restrikcí)* vznikne vybrání nových řádků z původní relace pomocí logických spojek. Tento algebraický výraz představuje v jazyce SQL příkaz *RESTRICT*.

Spojení dvou relací vytvoří třetí relaci, výsledná relace obsahuje všechny kombinace, které vyhovují zadané podmínce; podmínka vyjadřuje vztah mezi dvěma relacemi. Spojení mohou být na rovnost, nerovnost, inkluzi (spojení s tím rozdílem že do výsledné relace se přidají i nespojené řádky z první, ze druhé nebo z obou relací). Příslušné atributy pak nabývají hodnoty NULL. Tento algebraický výraz představuje v jazyce SQL příkaz *JOIN* [KALUŽA, 2004].

Symetrický rozdíl – nová relace bude obsahovat n-tice obou relací, s výjimkou těch, které se vyskytují v obou relacích. Tento algebraický výraz představuje v jazyce SQL příkaz *MINUS* [KALUŽA, 2004].

Kartézský součin spojuje každou n-tici s každou, kardinalita (počet n-tic) je součin kardinalit vstupních n-tic. Tento algebraický výraz představuje v jazyce SQL příkaz *CARTESIAN* [KALUŽA, 2004].

4.2. Typy dotazovacích jazyků

Máme dva typy dotazovacích jazyků – prvním typem jsou navigační (procedurální, algebraické); při formulaci dotazu je třeba zadat algoritmus jako posloupnost operací prováděných nad relacemi, který zajistí výběr příslušných dat. Navigační jazyky jsou založeny na relační algebře [LACKO, 2007, s. 126].

Druhým typem jsou specifikační (neprocedurální, deskriptivní, deklarativní) jazyky – požadavky na výběr se zadávají jako predikát charakterizující výslednou relaci. Výsledek výběru dat je relace, jejíž n-tice splňují podmínky výběru uvedené ve formuli. Specifikační jazyky jsou založeny na relačním kalkulu [LACKO, 2003, s. 78].

Relační kalkul vychází z predikátové logiky 1. řádu a v relačních databázích se vyskytuje ve dvou formách. Jedná se o n-maticový a doménový relační kalkul [LACKO, 2007, s. 126].

Seznam hodnot formule WHERE představuje v relačním kalkulu především seznam atributů, spojených n-ticovou proměnnou s danou relací; dále může obsahovat agregační funkce, resp. aritmetické výrazy. Formule se skládá z atomických formulí s unárním predikátem (jména

relací s příslušnými n-ticovými proměnnými). Kvalifikátory v relačním kalkulu jsou EXISTS a FORALL [LACKO, 2007, s. 132].

4.2.1. Příklady

Následující uvedené příklady z praxe nevycházejí z žádného zatím předkládaného relačního modelu. Slouží však k přednesu základní příkazové syntaxe jazyka SQL.

1) Z relace CENA_KNIHA vyberme všechna přírůstková čísla knih, jejich název a rok vydání, jejichž pořizovací cena je větší než 1000,- Kč.

a)

RESTRICT CENA_KNIHA WHERE cena>1000)
OVER prirust_c, nazev_k, rok_v, cena giving VYSL

b)

SELECT prisust_c, nazev_k, rok_v, cena FROM cena_kniha into table vysl
WHERE cena>1000

Tento dotaz má význam tehdy, kdy potřebujeme, aby nás knihovní systém upozornil na to, že čtenář si půjčuje drahou knihu a její zapůjčení lze pouze proti podpisu čtenáře.

2) Proveďme spojení relací KNIHA(isbn,autor,titul) a REZERV(isbn,c_ct,d_rez) včetně projekce všech neduplicitních atributů v relační algebře. Mějme relaci KNIHA, která se vyznačuje základními parametry jako je kód ISBN knihy, autorské a názvové údaje.

KNIHA [isbn=isbn] REZERV [isbn,autor,c_ct,d_rez]
PROJECT
(JOIN KNIHA and REZERV OVER isbn)
OVER isbn,autor,titul,c_ct,d_rez

```
SELECT a.isbn, a.autor, a.titul, b.c_ct, b.d_rez
FROM kniha a, rezerv b WHERE a.isbn=b.isbn
```

Tento dotaz má pochopitelně upotřebení v případě, že knihovník provede rezervaci dané knihy konkrétnímu čtenáři.

4.2.2. Další příklady

Je dáno schéma relační databáze pro automatizaci provozu veřejné knihovny :

Kniha (isbn, nazev, autor, zeme) klíč isbn

Exemplar (id, isbn, koupeno, cena)klíč id

Ctenar (cislo, jmeno, prijmeni, ulice, mesto, psc, rc, telefon, vzdelani) klíč cislo

Vypujcka(id, cislo, pujceno, vraceno) klíče id, cislo

Rezervace (isbn, cislo, rezervovano) klíče isbn,cislo

kde položky koupeno, pujceno, rezervovano jsou datumové, isbn je číslo titulu knihy (bez ohledu na počet kopií v knihovně), id je přírůstkové číslo exempláře knihy, cislo je číslo průkazu čtenáře.

Nyní si vypišme seznam jmen a adres čtenářů.

```
SELECT prijmeni, jmeno, ulice, mesto, psc
FROM Ctenar ;
```

Vypišme seznam čtenářů setříděný abecedně podle jmen.

```
SELECT prijmeni, jmeno, ulice, mesto, psc
FROM Ctenar
ORDER BY prijmeni,jmeno ;
```

Vypišme seznam knih nakoupených po 20.3.2002.

a)

```
SELECT *
FROM Exemplar INTO TABLE exemplar_20_03_02
```

```
WHERE koupeno>'20.03.02' ;
```

b)

```
SELECT K.isbn, K.nazev, E.koupeno, E.cena
FROM Exemplar E, Kniha K
WHERE E.isbn=K.isbn AND E.koupeno>’20.03.2002‘ ;
```

Najděme autory, na jejichž některé tituly je rezervace před 31.12.2006.

```
SELECT DISTINCT K.autor
FROM Kniha K, Rezervace R
WHERE K.isbn=R.isbn AND R.rezervovano<'31.12.2006' ;
```

Najděme dvojice čtenářů se stejnou adresou.

```
SELECT A.jmeno AS jmeno_1, A.prijmeni AS prijmeni_1,
B.jmeno AS jmeno_2, B.prijmeni AS prijmeni_2
FROM Ctenar A, Ctenar B
WHERE A.mesto=B.mesto AND A.ulice=B.ulice
AND A.cislo=B.cislo;
```

Najděme všechny exempláře vydané na Slovensku a jejich cenu v SK.

```
SELECT E.isbn, E.id, K.nazev, E.cena *(10/8) AS Cena_ve_SK
FROM Exemplar E, Kniha K
WHERE E.isbn=K.isbn AND E.zeme='SK' ;
```

Zjistěme celkový počet čtenářů knihovny.

```
SELECT COUNT(*) AS pocet_ctenaru
FROM Ctenar ;
```

(nepojmenovaná unární relace s jedním prvkem).

Určeme počet čtenářů, kteří mají rezervovanou nějakou knihu.

```
SELECT COUNT(DISTINCT cislo) AS počet čtenářů s rezervací
FROM Rezervace ;
```

Určeme počet titulů knihovny, které byly vydány v Německu.

```
SELECT COUNT(*)
FROM Kniha
WHERE zeme='DE' ;
```

Jaká je celková cena českých knih v knihovně ?

```
SELECT SUM(E.cena)
FROM Exemplar E, Kniha K
WHERE E.isbn=K.isbn AND K.zeme='CZ' ;
```

Určeme pro každého čtenáře aktuální počet vypůjčených knih.

a)

```
SELECT cislo,COUNT(*) AS pocet
FROM Vypujcka WHERE ISNULL(vraceno)
GROUP BY cislo ;
```

b)

```
SELECT C.jmeno,COUNT(*) AS pocet
FROM Vypujcka V, Ctenar C
WHERE V.cislo=C.cislo
GROUP BY C.cislo ;
```

Najděme čtenáře, kteří mají půjčeno více než 5 knih.

```
SELECT C.jmeno,COUNT(*)
FROM Vypujcka V, Ctenar C
WHERE V.cislo=C.cislo
GROUP BY C.cislo
HAVING COUNT(*)>5 ;
```

Najděme všechny tituly vydané v ČR, Rusku a SR.

```
SELECT isbn, nazev, zeme
FROM Kniha
WHERE zeme IN ('CZ','RU','SK') ;
```

Najděme čísla čtenářů, kteří mají současně zapůjčeny nějaké knihy a rezervovány nějaké tituly do konce roku 2007.

a)

```
SELECT C.cislo, C.jmeno, C.prijmeni
FROM Ctenar C, Rezervace R, Vypujcka V
WHERE C.cislo = R.cislo AND
C.cislo = V.cislo AND
R.rezervovano <'31.12.07' ;
```

b)

```
SELECT cislo, jmeno, prijmeni FROM ctenar
WHERE cislo IN (SELECT cislo FROM vypujcka)
AND cislo IN (SELECT cislo FROM rezervace WHERE
rezervovano<’31.12.2007‘)
```

Najděme jména čtenářů, kteří mají rezervovánu knihu Babička.

```
SELECT C.prijmeni,C.jmeno
FROM Ctenar C, Rezervace R, Kniha K
WHERE C.cislo = R.cislo AND K.isbn  = R.isbn AND
K.nazev = 'Babička' ;
```

5. Příklad návrhu databáze KNIHOVNA

Databáze je soubor souvisejících dat. Systém pro zprávu databáze je určen ke dvěma hlavním účelům:

- přidávání, odstraňování a aktualizace dat v DB
- zabezpečení dat

Jsou-li data jednoduchá a je-li jich málo, stačí jedna tabulka (viz. tabulka XIII) a aplikace MS WORD nebo MS EXCEL. Např. chcete-li vytvořit databázi pro knihy ve vaší knihovně, předpokládejme, že máte 14 knih.

ISBN	Název	KódAut	JménoAut	KódNakl	NázevNakl	TelNakl	Cena
1-1111-1111-1	C++	4	Roman	1	Big House	123-456-7890	129
0-99-999999-9	Emma	1	Austen	1	Big House	123-456-7890	120
0-91-335678-7	Fairie Queene	7	Spencer	1	Big House	123-456-7890	150
0-91-045678-5	Hamlet	5	Shakespeare	2	Alpha Press	999-999-9999	200
0-103-45678-9	Iliad	3	Homer	1	Big House	123-456-7890	490
0-12-345678-6	Jane Eyre	1	Austen	3	Small House	714-000-0000	250
0-99-777777-7	King Lear	5	Shakespeare	2	Alpha Press	999-999-9999	340
0-555-55555-9	Macbeth	5	Shakespeare	2	Alpha Press	999-999-9999	200
0-11-345678-9	Moby Dick	2	Melville	3	Small House	714-000-0000	490
0-12-333433-3	On Liberty	8	Mill	1	Big House	123-456-7890	250
0-321-32132-1	Balloon	13	Sleepy	3	Small House	714-000-0000	340
0-321-32132-1	Balloon	11	Snoopy	3	Small House	714-000-0000	340
0-321-32132-1	Balloon	12	Grumpy	3	Small House	714-000-0000	340
0-55-123456-9	Main Street	10	Jones	3	Small House	714-000-0000	225
0-55-123456-9	Main Street	9	Smith	3	Small House	714-000-0000	225
0-123-45678-0	Ulysses	6	Joyce	2	Alpha Press	999-999-9999	340
1-22-233700-0	Visual Basic	4	Roman	1	Big House	123-456-7890	250

Tabulka XIII: Jednoduchý výpis seznamu knih vytvořený programem MS Excel

Aktualizace takovéto tabulky nepředstavuje žádný problém, znáte-li např. MS EXCEL.

5.1. Proč používat relační databázový návrh?

Skutečné databáze, např. databáze knih v univerzitní knihovně v Praze jsou mnohem složitější. Těžko si můžeme představit, že by se systém vyhledávání a ukládání dat realizoval přes systém tabulkového procesoru. První problém, který může potenciálního uživatele i správce této databáze potkat je problém s nadbytečností (redundancí). Jedná se o zbytečné opakování stejných údajů, např. v tabulce XIII se **NázevNakl** Big House opakuje 6-krát a **JménoAut** Shakespeare 3x. Proto, abychom

odstranili tuto redundanci musíme tabulku XIII upravit. Udělejme si základní dvě dělení:

- rozdělit do více tabulek, ve kterých se údaje nebudou opakovat
- provázat tabulky tak, abychom se na příslušné údaje mohli odkazovat (přidáním nové položky)

Rozdělení tabulky XIII v našem případě.

tabulka XIV: **KNIHA** má atributy **ISBN, Název, KódNakl, Cena**

tabulka XV: **AUTOR** má atributy **KódAut, JménoAut**

tabulka XVI: **NAKLADATELSTVÍ** má atributy **KódNakl, NázevNakl, TelNakl**

Navíc ještě zavedeme tabulku XVII **KNIHA/AUTOR**.

KNIHA

ISBN	Název	KódNakl	Cena
1-1111-1111-1	C++	1	129
0-99-999999-9	Emma	1	120
0-91-335678-7	Fairie Queene	1	150
0-91-045678-5	Hamlet	2	200
0-103-45678-9	Iliad	1	490
0-12-345678-6	Jane Eyre	3	250
0-99-777777-7	King Lear	2	340
0-555-55555-9	Macbeth	2	200
0-11-345678-9	Moby Dick	3	490
0-12-333433-3	On Liberty	1	250
0-321-32132-1	Balloon	3	340
0-55-123456-9	Main Street	3	225
0-123-45678-0	Ulysses	2	340
1-22-233700-0	Visual Basic	1	250

Tabulka XIV: Atributy tabulky KNIHA

AUTOR

KódAut	JménoAut
4	Roman
1	Austen
7	Spencer
5	Shakespeare
3	Homer
2	Melville
8	Mill
13	Sleepy
11	Snoopy
12	Grumpy
10	Jones
9	Smith
6	Joyce

Tabulka XV: Atributy tabulky AUTOR

KNIHA/AUTOR

ISBN	KódAut
1-1111-1111-1	4
0-99-999999-9	1
0-91-335678-7	7
0-91-045678-5	5
0-103-45678-9	3
0-12-345678-6	1
0-99-777777-7	5
0-555-55555-9	5
0-11-345678-9	2
0-12-333433-3	8
0-321-32132-1	13
0-321-32132-1	11
0-321-32132-1	12
0-55-123456-9	10
0-55-123456-9	9
0-123-45678-0	6
1-22-233700-0	4

Tabulka XVII: Atributy tabulky KNIHA/AUTOR

NAKLADATELSTÍ

KódNakl	NázevNakl	TelNakl
1	Big House	123-456-7890
2	Alpha Press	999-999-9999
3	Small House	714-000-0000

Tabulka XVI: Atributy tabulky NAKLADATELSTVÍ

Nyní se již v tabulkách neopakují uvedené údaje (Big House a Shakespeare), některé údaje se ale stále opakují, tomu však nelze zabránit. (např. KódNakl)

5.1.1. Problémy s násobnými hodnotami

Např. jedna kniha je napsaná více autory. Možnost řešení:

- pro danou knihu zvolit tolik řádků, kolik autorů ji napsalo
- pro daného autora zvolit tolik sloupců, kolik autorů ji napsalo (problém kolik)
- jména všech autorů umístit do jednoho sloupce

5.1.2. Anomálie při aktualizaci

Máme-li např. změnit u některého nakladatelství telefonní číslo, musíme důkladně prohledat celou tabulku, vynechání některého čísla by vedlo k problémům (tzv. sirotčí či slepý záznam).

5.1.3. Anomálie při vkládání

Chceme-li vložit do tabulky nové nakladatelství nastane problém, protože k němu nemáme knihu. U názvu či autora se to dá vyřešit tzv. hodnotou NULL, ale nedá se to vyřešit u ISBN, což je jedinečné číslo a má jedinečnou hodnotu.

5.1.4. Anomálie při odstraňování

Zrušíme některé nakladatelství, ale tím zrušíme vazby některých knihy.

5.1.5. Zabránění ztrátě dat

Při navrhování relační databáze je poměrně složité nalézt způsob, jak vhodně navrhnout tabulky, aby při zápisu dat nemohlo dojít ke ztrátě některých údajů. Proto se zavádějí další tabulky. V našem případě by to byla tabulka XVI (KNIHA/AUTOR) s atributy ISBN a KódAut**.** Kdyby nebyla tato tabulka nebudeme mít vazbu mezi knihami a autory.

5.1.6. Zachování relační integrity

Pokud v tabulkách dojde k nějakým změnám, musíme důsledně udržovat integritu existujících vztahů mezi tabulkami. To znamená, pokud chceme zrušit nějaké nakladatelství z tabulky XVI (NAKLADATELSTVÍ), musíme zajistit, aby se odkaz na toto nakladatelství zrušilo i v tabulce XIV (KNIHA). Jinak se jedná opět o slepý či sirotčí záznam.

5.1.7. Vytváření dotazů

Dotaz můžeme směřovat např. na sloučení dat z různých tabulek do jedné. Např. chceme zobrazit všechny knihy jednoho nakladatelství, které stojí méně než 200,- Kč.

5.1.8. Stručný slovníček

Dříve, než se pustíme ve výkladu a popisu jednotlivých vztahů mezi atributy a objekty, připomeneme si význam jednotlivých slov používaných v databázové terminologii.

Entita

Objekt pro uložení jehož informací je databáze určena. Příklad: *Kniha*, to znamená informace o ISBN, Názvu, Ceně apod.

Atribut

Vlastnost, která popisuje každou entitu.

Třída entit

Abstraktní skupina entit se společným popisem. Příklad: třída entit *Kniha* představuje všechny knihy na Zemi.

Množina entit

Skupina konkrétních entit. Příklad: Obsah tabulky KNIHA (tabulka XIV).

Nadklíč

Množina určitých atributů entit, která slouží pro jednoznačnou identifikaci každé entity. Příklad: Název, ISBN, v případě třídy entit *Kniha.*

Klíč

Minimální nadklíč, to znamená takový nadklíč, ze kterého když odstraníme libovolný atribut, tak výsledná množina přestává být nadklíčem. Příklad: ISBN v případě třídy entit *Kniha.*

Tabulka

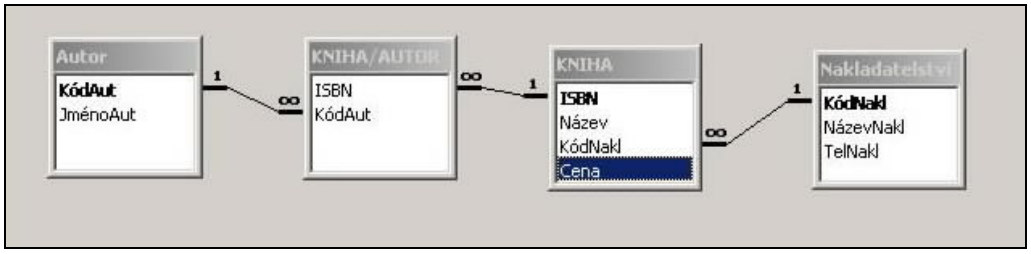
Obdélníkové pole hodnot atributů, jehož sloupce obsahují všechny hodnoty daného atributu a jehož řádky obsahují hodnoty všech atributů dané entity. Tabulky se používají pro implementaci množin entit.

Definice tabulky

Množina všech názvů atributů pro danou třídu entit. Příklad: ISBN, Název, Cena

Relační databáze

Konečná množina tabulek, které představují databázi - viz obr. 15.



Obr. 15: Znázornění vytvořených relačních vazeb pro databázi KNIHOVNA

5.2. Model entit a vztahů pro popis databáze

5.2.1. Entity a jejich atributy

Entita je typ objektu v našem případě např. Kniha, Autor, Nakladatelství. V daném okamžiku naše databáze KNIHOVNA obsahuje 14 entit knih. Okruh všech možných entit, které může databáze obsahovat se označuje jako třída entit. Tzn. okruh všech možných knih tvoří třídu entit *Kniha,* okruh všech možných autorů tvoří třídu entit *Autor.* Třída entit je abstraktní popis objektu, entita je konkrétní případ objektu, Množina entit je souhrn entit příslušné třídy. Např. tabulka XIV (KNIHA): entita – kniha obecně objekt, třída entit – možné knihy, množina entit – konkrétní knihy (v našem případě množina entit obsahuje 14 knih obsažených v tabulce XIV (KNIHA). Množina entit se stále mění, třída entit je neměnná. (U objektově orientovaného programování třída entit je třída a entita objekt) Entity z každé třídy entit obsahují určité vlastnosti, které se nazývají atributy. Tyto atributy odpovídají jednotlivým polím v tabulkách.

Atributy třídy entit slouží k:

- uložení informace, které chceme do databáze ukládat
- jednoznačné identifikaci entit v rámci třídy entit
- popisu vztahu mezi entitami z různých jiných tříd entit

5.2.2. Atributy jednotlivých tříd entit z databáze KNIHOVNA

Atributy třídy *Kniha*:

- Název
- ISBN
- KódNakl
- Cena

Atributy třídy *Autor*:

- JménoAut
- KódAut

Atributy třídy *Nakladatelství*:

- NázevNakl
- TelNakl
- KódNakl

poznámka:

- Primární klíče jsou podtržené.
- Z uvedených atributů nelze říct kdo je autor dané knihy, není uveden ve třídě entit *Kniha*.
- Jedinečné číslo knihy ISBN (International Standard Book Number) složí pro jednoznačné určení knihy.

5.2.3. Klíče a nadklíče

Skupina atributů, která jednoznačně identifikuje každou entitu mezi všemi možnými entitami ze třídy entit, jež se mohou v databázi

vyskytnout, se nazývá nadklíč této třídy entit. Množina [ISBN] je tedy nadklíčem pro třídu entit *Kniha* a množiny [Kód Nakl] a [NázevNakl, TelNakl] jsou obě nadklíči pro třídu entit *Nakladatelství.* Nadklíč se týká tříd entit nikoli množin entit. Klíč je takový nadklíč, ze kterého, když odstraníme libovolný atribut, pak výsledkem již není nadklíč. Stručněji klíč je minimální nadklíč.

5.2.4. Typy vztahů

Vztahy mohou být řazeny do jednoho ze tří typů, které byly popsány v předchozích kapitolách:

- typ **1:1**
- typ **1:n**
- typ **m:n**

5.3. Implementace modelů entit a vztahů: Relační databáze

E-R model databáze je abstraktní model, může být znázorněn pomocí E-R diagramu. Když každý aspekt modelu popíšeme konkrétními pojmy dostaneme model konkrétní.Říkáme, že E-R implementujeme. K implementaci patří:

- implementace entit
- implementace tříd entit
- implementace množin entit
- implementace vztahů mezi množinami entit

Výsledkem této implementace je relační databáze (implementovat něco, znamená popsat něco konkrétními pojmy)

5.3.1. Implementace entit

Entita je implementována, když se jejím atributům přiřadí konkrétní hodnoty. Následující zápis může být implementací entity *Kniha*:

Název = King Lear
ISBN = 0-99-777777-7
KódNakl = 2
Cena = 340

5.3.2. Implementace tříd entit – definice tabulek

Entity jsou implementovány hodnotami svých atributů, proto je vhodné celou třídu entit implementovat množinou názvu těchto atributů. Např. třída entit *Kniha* by byla identifikována touto množinou (ISBN, Název, KódNakl, Cena). Množina atributů tabulky se zapisuje do záhlaví sloupců a nazývá se definice tabulky. Používá se notace *Kniha* (ISBN, Název, KódNakl, Cena), z této notace vyčteme název entity i názvy atributů.

5.3.3. Implementace množin entit – tabulky

V relační databázi je každá množina entit modelována tabulkou. Např. tabulka XIV (KNIHA)

- První řádek tabulky představuje definici tabulky pro třídu entit *Kniha.*
- Všechny ostatní řádky tabulky implementují konkrétní entity *Kniha.*
- Všechny řádky této tabulky, kromě prvního, implementují samotnou množinu entit.

Formálně je libovolná tabulka obdélníkové pole s vlastnostmi:

- začátek každého sloupce má záhlaví (atribut).

- všechny prvky jednoho sloupce pocházejí z jediné množiny, která se nazývá doménou. Doména je množina přípustných hodnot atributu
- žádné dva řádky v tabulce nejsou identické

Poznámky:

- Tabulka má název, např. KNIHA.
- Počet řádků tabulky se nazývá velikost tabulky a počet sloupců se nazývá stupeň. Např. jméno tabulky KNIHA, velikost 14 stupeň 4, atributy ISBN, Název, KódNakl a Cena.
- Na pořadí řádků v tabulce nezáleží.

5.3.4. Implementace vztahů v relační databázi

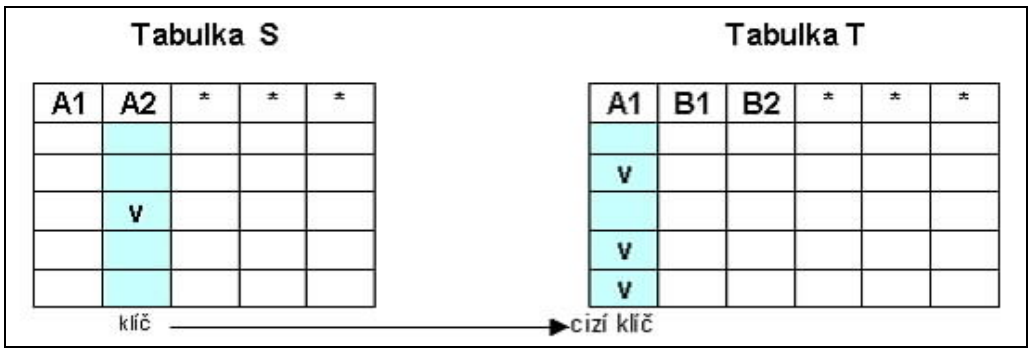
5.3.4.1. Vztah typu 1:n – cizí klíče

Tato implementace (např. **Vydalo)** je poměrně jednoduchá. Protože **KódNakl** je klíčem pro třídu entit *Nakladatelství* a cizím klíčem pro třídu entit *Kniha.* Třída entit *Kniha* je tabulka XVIII.

ISBN	Název	KódNakl	Cena
1-1111-1111-1	C++	1	129
0-99-999999-9	Emma	1	120
0-91-335678-7	Fairie Queene	1	150
0-91-045678-5	Hamlet	4	200
0-103-45678-9	Iliad	1	490
0-12-345678-6	Jane Eyre	3	250
0-99-777777-7	King Lear	2	340
0-555-55555-9	Macbeth	2	200
0-11-345678-9	Moby Dick	3	490
0-12-333433-3	On Liberty	1	250
0-321-32132-1	Balloon	3	340
0-55-123456-9	Main Street	3	225
0-123-45678-0	Ulysses	2	340
1-22-233700-0	Visual Basic	1	250

Tabulka XVIII: Atributy v tabulce KNIHA

Hodnota cizího klíče KódNakl v tabulce XVIII vytváří odkaz na odpovídající hodnotu klíče v tabulce NAKLADATELSTVÍ (tabulka XVI), ta se tam vždy nalezne a tím se definuje vztah ***Vydalo,*** který je typu **1:n.** Tento vztah lze znázornit podle obr. 16.



Obr. 16: Vztah typu 1:n na příkladu tabulek S a T

5.3.5. Implementace vztahu typu m:n - nové třídy entit

Je to vztah **Napsal** mezi třídami *Kniha* a *Autor* Nestačí přidat cizí klíč, museli bychom duplikovat řádky v tabulce. Kdybychom přidali do definice tabulky *Autor* klíč ISBN a do definice tabulky *Kniha* klíč KódAut, pak by každá kniha, kterou napsali 2 autoři, měla 2 různé řádky v tabulce. Správný postup je ten, že do databáze přidáme novou definici tabulky a s její pomocí rozdělíme vztah **m:n** na dva vztahy **1:n** a **1:n**. V našem případě do tabulky přidáme definici tabulky KNIHA/AUTOR (viz tabulka XVII), jejíž atributy se skládají z cizích klíčů ISBN a KódAut: KNIHA/AUTOR (ISBN,KódAut*)*.

5.3.6. Referenční integrita

Každá hodnota cizího klíče musí mít odpovídající hodnotu v odkazovaném klíči (jinak slepý či sirotčí záznam). Požadavek, že každá hodnota v cizím klíči musí být hodnotou odkazovaného klíče se nazývá **Referenční omezení**. Zajištění tohoto požadavku se nazývá **Referenční integrita**. Jako příklad lze uvést úmyslné zavedení chyby do tabulky

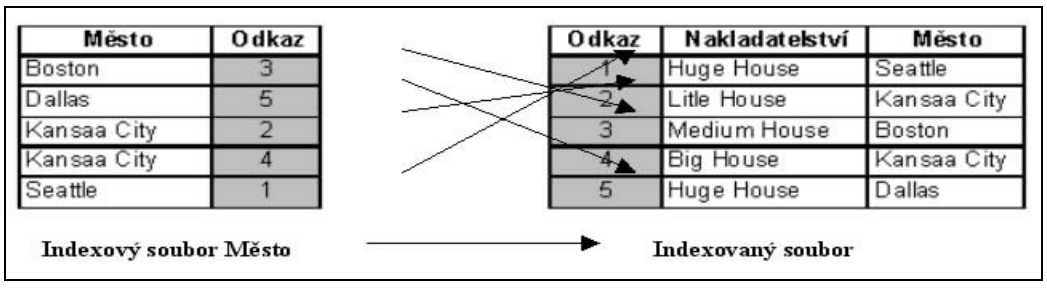
XVIII, kde u knihy Hamlet je úmyslně pozměněno pole KódNakl na hodnotu 4. Vědomě jsme tak vytvořili slepý odkaz. Avšak při vytváření relace v programu Microsoft Access se objeví chybové hlášení o porušení referenční integrity. Porušení spočívá v tom, že byla zadána do tabulky XVIII do pole KódNakl hodnota 4. Ovšem nakladatelství s kódem 4 nebylo definováno v tabulce XVI (NAKLADATELSTVÍ).

5.3.7. Kaskádová aktualizace a kaskádové odstraňování

Zachovat **referenční integritu** nám umožňuje **kaskádová aktualizace** (změna odkazovaného klíče automaticky znamená změnu cizího klíče). Podobně funguje **kaskádové odstraňování** (když v odkazované tabulce zrušíme klíč, pak se automaticky zruší všechny řádky s tímto klíčem v odkazující tabulce). To znamená, že tímto je udržována databáze „v souladu“.

5.3.8. Indexové soubory

Řekněme, že jistá tabulka je uložena v počítači na pevném disku. Toto zhmotnění se označuje jako soubor. Řádek takové tabulky se označuje jako záznam a sloupec jako pole. Záznamy na pevném disku se mohou prohlížet postupně jeden za druhým nebo-li sekvenčně (tento postup bývá pomalým). Nebo záznamy prohlížíme přímo, ale to musí být prostřednictvím indexovaného souboru. Princip indexovaného souboru je na obr. 17.



Obr. 17: Indexovaný soubor Město pro soubor Nakladatelství

Vysvětleme si princip vztahu souborů, tak jak jsou vidět na obr. 17. Levý soubor indexuje pravý soubor podle položky Město, která se označuje jako indexové pole. Levý soubor se někdy také nazývá index do tabulky. Tento soubor obsahuje pouze dvě položky. Položku Město, kde jsou zapsaná a abecedně setříděná všechna města z pravého souboru a ukazatele na odpovídající záznamy pravého souboru. Tohoto MS Access využívá např. v případě, když chceme nalézt všechna nakladatelství v Kansas City. Microsoft Access nalezne v indexovém souboru město Kansas City, tím nalezne odkaz na záznam indexovaného souboru. Soubor může být indexován podle více polí. Index nad primárním klíčem se nazývá **primární klíč**, ostatní klíče se nazývají **sekundární.**

5.3.9. Hodnoty NULL

NULL je speciální hodnota, kterou používáme když hodnota chybí nebo není známá. Hodnota NULL není povolena v primárním klíči. Hodnota NULL v cizím klíči neporušuje referenční identitu [HERNANDEZ, 2006, s. 68].

6. Databázové produkty

Databáze je možné rozdělit podle několika kritérií. Časté je dělení podle velikosti, které samozřejmě souvisí i s dalšími možnostmi, jež databáze poskytuje.

Malé databáze - PC Fand, Paradox, dBase apod. Databáze určené převážně pro použití na lokálním PC a vhodné pro návrh jednodušších, obvykle jednouživatelských databázových úloh.

Střední databáze - Typickými představiteli jsou MS Access a Visual FoxPro apod. Lze je použít na jednom PC a pro rychlý a jednoduchý vývoj databázových aplikací, ale mohou pracovat i jako síťové a obsahují rovněž některé charakteristické znaky velkých databází. [2]

Velké databáze - Oracle, Sybase, MS SQL Server, Caché, IBM DB2, Informix apod. Mohutné databáze s důrazem na výkon, bezpečnost a zpracování velkého množství dat. V této souvislosti se o nich hovoří jako o RDBMS (Relation Data Base Management System), neboli systému báze řízení dat. U velkých databází se vždy jedná o funkcionalitu na principu klient/server, což znamená, že vlastní databáze (databázový stroj) běží na samostatném výkonném počítači (tzv. databázovém serveru). Přístup k tomuto serveru se děje pomocí klientských aplikací, které k němu přistupují z jednotlivých PC (klientů) přes počítačovou síť. U velkých databází je rovněž vždy určen princip licencování, tzn. kolik klientů může přistupovat k databázovému serveru. Možnosti jsou většinou buď podle předem definovaných (pojmenovaných) klientů, nebo podle maximálního počtu v jednom okamžiku přistupujících klientů. [2]

Vlastní databáze nedílně provází řada vývojářských nástrojů pro tvorbu databázových aplikací. Tyto nástroje umožňují rychlý návrh struktury databáze, formulářů pro vstup dat, tiskových výstupů a vlastní logiky aplikace. Mohou být přímo od dodavatelů databází nebo od nezávislých poskytovatelů.

6.1. Přehled vybraných databázových řešení

Není posláním této práce detailně popisovat jednotlivá databázová řešení. Je však třeba uvést alespoň názvy hlavních výrobců (podle abecedního pořadí) a jejich velmi stručnou charakteristiku.

6.1.1. Caché

Postrelační databáze Caché umožňuje objektově orientovaný přístup i práci pomocí SQL, přičemž data jsou ukládána ve vícerozměrném databázovém stroji. Moderní databáze s možností komunikace s klasickými relačními databázemi [STEJSKAL, 2006].

6.1.2. IBM DB/2

Hlavní databázový produkt IBM s dlouhou historií vývoje. Doporučený pro vysoce výkonné, škálovatelné a nepřetržitě dostupné databázové aplikace na nejrůznějších platformách operačních systémů. [2]

6.1.3. Informix

Původně jedna z nejvíce rozšířených velkých databázových platforem. K datu 1. 7. 2001 přešla databázová společnost Informix celosvětově pod IBM, která se zavázala k dalšímu vývoji i podpoře produktů Informix tak, aby byla zajištěna ochrana investic zákazníků a jejich obchodních partnerů. Některé klíčové technologie budou postupně zabudovány do produktů a řešení IBM. [2]

6.1.4. MS Access

Desktopová databáze pro operační systém Windows, která je součástí vyšších verzí Microsoft Office. Mimořádně uživatelsky příjemná, velmi vhodná pro praktické seznámení s databázovou problematikou.

6.1.5. MS FoxPro

Jednoznačně nejrozšířenější desktopová databáze v České republice, které se pak díky nelegálním instalacím tohoto produktu a nepřebernému množství vyvinutých aplikací dostalo poněkud nelichotivého přídomku Foxland. I po odkoupení společností Microsoft a jednom období nevyjasněné budoucnosti (přímý konkurent Accessu) má stále svůj neotřesitelný okruh příznivců [STEJSKAL, 2006].

6.1.6. MS SQL Server

Hlavní databázový produkt firmy Microsoft pro operační systém Windows. Dodáván s celou řadou administrátorských a klientských nástrojů.

6.1.7. MySQL

Populární databáze pro internetová řešení ve spojení s webovým serverem Apache. Pro některá nekomerční řešení zdarma. Postrádá transakční zpracování, což ji předurčuje k prohlížení dat (www stránky). Provozovatelná na Unixu i Windows. [2]

6.1.8. Oracle

Světová špička v databázových technologiích. Podle řady nezávislých výzkumů drží dlouhodobě největší podíl na trhu. Vysoký výkon, podpora moderních technologií, řada souvisejících produktů. Dostupná na většině operačních systémů.

6.1.9. Progress RDBMS

Mimořádně stabilní a přitom škálovatelná databáze s řadou technologických novinek. Určená pro operační systémy Windows i Unix [STEJSKAL, 2006].

6.1.10. Sybase

Jeden z průkopníků v oblasti distribuovaných databází. Robustní platforma, vysoký výkon, důraz na bezpečnost. Běží na Windows i Unixu. Podporována řadou souvisejících produktů. [2]

7. Projekt pro malou knihovnu

Tato část diplomové práce bude příkladovou studií o tom, jak by se mohlo postupovat při návrhu databázového, resp. informačního systému (dále jen IS), který bude postaven na relačním databázovém modelu. Tento systém by měl podporovat vypůjčování knih, nákup nových knih od vydavatele a vyřazování poškozených nebo ztracených knih. Knihy bude možné rezervovat. Nejedná se o příliš složitý systém a některé funkce systému budou jen zmíněny, neboť konkrétním návrhem například modulu generování čárových kódu knih, by se studie dostala až do příliš náročných popisů. Proto bude studie zaměřena na popis modelových situací, dále pak popis diagramů a obrázků u takto navrhovaného systému.

Hlavní část knihovního systému je vždy tvořena databází všech knih, které se nacházejí v knihovně. Knihy jsou v dnešní době ve velké části knihoven vybaveny čárovým kódem. To pro snadnou identifikaci. Do databáze jsou zaneseny při dodání knih do knihovny. Tuto akci provede v našem případě správce knihovny. Naopak, v případě zničení nebo ztráty knihy, vymaže správce knihovny nebo knihovník příslušný záznam.

Návštěvníci knihovny budou registrováni v IS (tím se stanou členy knihovny). Při registraci jim bude přidělen průkaz s osobním číslem, jméno a heslo pro autorizovaný přistup přes Internet. Členové si mohou volně vypůjčovat knihy na dobu jednoho měsíce. Každá výpůjčka bude zanesena knihovníkem do IS. Po překročení výpůjční doby bude možno po požádání člena (ústně nebo přes Internet) výpůjční dobu prodloužit. Výpůjční dobu bude možno prodloužit maximálně na tři měsíce. Pokud člen nepožádá o prodloužení výpůjční doby, nebo tato doba přesáhne tři

měsíce, vygeneruje IS na podnět správce knihovny upomínku, která bude poslána členovi. Upomínky budou generovány jednou týdně. Po třetí upomínce ztrácí člen možnost půjčovat si další knihy, dokud příslušnou knihu nevrátí nebo nezaplatí. Na překročení výpůjční doby u některé z výpůjček upozorní systém automaticky.

Do IS knihovny se u moderních variant přistupuje pomocí Internetu. Neregistrovaní návštěvníci internetových stránek knihovny si mohou pouze prohlížet seznam všech knih. Registrovaní členové si mohou navíc rezervovat vypůjčení knihy, prohlédnout seznam jimi vypůjčených knih a případně požádat o prodloužení výpůjční doby. V seznamu knih získaném přes Internet se uvádí i informace, zda se příslušná kniha nachází právě v knihovně, nebo je vypůjčená.

V IS bude obsažena tabulka zaměstnanců (knihovníci, správce knihovny) kvůli autorizaci přístupu k IS. S tabulkou zaměstnanců má většinou právo manipulovat pouze správce knihovny.

7.1. Návrh realizace

IS bude postaven na architektuře klient-server. Centrem bude výkonný server s databázovým strojem. Tento server bude připojen přímo k Internetu.

V dnešní době má prakticky každá knihovna k dispozici připojení na Internet. V reálné praxi se doporučuje, aby pracoviště knihovníka, pracoviště správce a terminály byly se serverem spojeny vnitřní sítí knihovny (tzv. intranet)

7.1.1. Moduly IS

Jak už bylo zmíněno, knihovnický systém má tři základní varianty. Každá se s ohledem na daného uživatele definuje odlišně.

- **modul knihovník** - aplikační program pro knihovníka a správce knihovny
- **modul terminál** - program běžící na terminálech rozmístěných po knihovně
- **modul správce** – práce s databází, dohled nad všemi procedurami

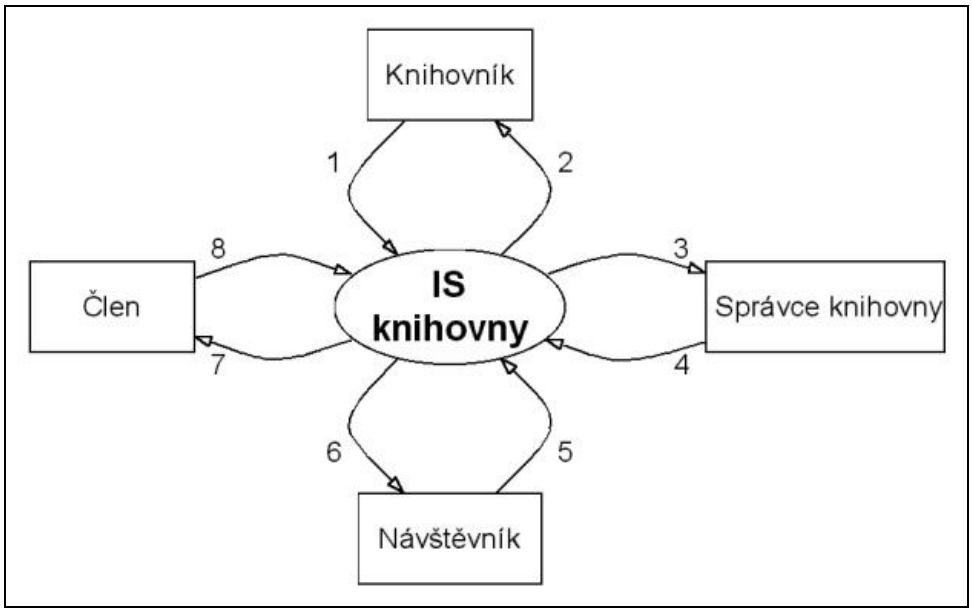
7.1.2. Seznam aktérů

Moduly navrhovaného resp. popisovaného systému jsou tedy podmíněny typům uživatelů. Tito uživatelé jsou rozděleni do čtyř skupin:

- **knihovník** - registruje nové členy, eviduje vypůjčení a vrácení knih, může vyřadit knihu z databáze, prodlužuje výpůjční dobu
- **správce knihovny** - eviduje nové knihy, může vyřadit knihu z databáze, tiskne upomínky, udržuje tabulku zaměstnanců
- **člen** - registrovaný člen si prohlíží seznam knih na terminálu nebo prostřednictvím Internetu, rezervuje si půjčení knihy (ústně nebo z Internetu), prohlíží si seznam jím vypůjčených knih
- **návštěvník** - prohlíží si seznam knih

7.1.3. Popis událostí

Na obr. 18 jsou znázorněny vztahy mezi knihovnickým systémem a jednotlivými vazebními prvky na knihovnu, což je skupina knihovníků, správců knihovny, členů a návštěvníků knihovny. Vazby mezi nimi a systémem jsou očíslované události a reakce systému. Nazývejme je datové toky.



Obr. 18: Diagram kontextu znázorňující jednotlivé události

Popišme si tedy nejprve vstupní události, tak jak k nim v reálném chodu knihovny dochází. Tyto události jsou označeny čísly na obr. 18.

Datový tok č. 1

- **vypůjčení knihy** - systém zaznamená do databáze vypůjčení knihy
- **vrácení knihy** - systém zaznamená do databáze vrácení knihy
- **vyřazení knihy** - vymazání knihy z databáze
- **registrace člena** - vložení nového člena do databáze registrovaných čtenářů
- **prodloužení výpůjčky** – možnost prodloužit si dobu výpůjčky po uplynutí 30-ti denní lhůty
- **dotaz na knihy** - výběr záznamu o knihách podle zadaných kritérií
- **dotaz na výpůjčky** - výběr záznamu o výpůjčkách podle zadaných kritérií

Datový tok č.4

- **nové knihy** - vložení informací o nové knize

- **vyřazení knihy** – tento úkon může provést jen knihovník a správce systému
- **dotaz na knihy** – vyhledání knih v knihovní databázi
- **dotaz na výpůjčky** - výběr záznamu o výpůjčkách podle zadaných kritérií
- **dotaz na zaměstnance** - výběr záznamů o zaměstnancích podle zadaných kritérií
- **požadavek upomínky** - požadavek na tisk upomínek
- **úprava tabulky zaměstnanců** - vložení, zrušení nebo úprava záznamu o zaměstnanci

Datový tok č.5

- **dotaz na knihu** – přes uživatelské rozhraní systému neregistrovaný návštěvník vyhledává knihu

Datový tok č.8

- **dotaz na knihu** – přes uživatelské rozhraní systému registrovaný čtenář vyhledává knihu
- **rezervace** – čtenář má možnost rezervace vypůjčené knihy jiným čtenářem
- **žádost** - žádost o prodloužení výpůjčky, kterou si podá čtenář
- **vypůjčené knihy** - čtenářův požadavek na zobrazení jím vypůjčených knih

Vstupní události jsme si popsali a následuje seznam reakcí systému na tyto požadavky. Opět můžeme říci, že tyto reakce knihovního systému jsou tak, jak je zná praxe. Tyto reakce jsou označeny čísly v obr. 18.

Datový tok č.2

- **potvrzení akce** – následuje reakce knihovníka, jenž provede požadovanou akci, kterou je:
  - vypůjčení knihy čtenáři

- **vrácení knihy** od čtenáře
- **vyřazení knihy** z fondu knihovny
- **prodloužení výpůjčky** čtenáři

- **seznam knih** – tento dotaz provede knihovník, jenž dostane odpověď systému na:
  - **dotaz na knihy** vyhledání knih v databázi knihovny
- **seznam výpůjček** - takto odpoví systém knihovníkovi na:
  - **dotaz na výpůjčky** a knihovník má tak možnost vidět seznam zapůjčených knih čtenářem

Datový tok č.3

- **potvrzení akce** – knihovnický systém tak reaguje na zadání od správce knihovny a následuje:
  - **vyřazení knihy** tj. systém vymaže údaje o knize z databáze
  - **úprava tabulky zaměstnanců**, což je právo jen správce systému
- **seznam knih** - tento dotaz provádí správce systému, jenž dostane odpověď systému na:
  - **dotaz na knihy,** což je vyhledání knih v databázi knihovny a správce tak dostává výběr záznamu o knihách podle zadaných kritérií
- **seznam výpůjček** – správce systému obdrží reakci systému na:
  - **dotaz na výpůjčky**, což je výběr záznamu o výpůjčkách podle zadaných kritérií
- **informace o zaměstnanci** – správce systému (neboli knihovny) obdrží odpověď systému na:
  - **dotaz na zaměstnance** - výběr záznamů o zaměstnancích podle zadaných kritérií
- **tisk upomínky** – správce systému reaguje na:
  - **požadavek upomínky** a správce knihovny má oprávnění tisknout upomínky

- **ID knih** – správce knihovny při zavádění **nových knih** do fondu knihovny přiřadí každé z nich jedinečné číslo
- **upozornění na upomínky** – tuto reakci vyvolá systém automaticky a správce tak zaznamená, že čtenář překročil dobu zapůjčení knihy

Datový tok č.6

- **seznam knih** – systém takto reaguje na dotaz návštěvníka (neregistrovaného čtenáře):

  dotaz na knihy což je vyhledání knih v databázi knihovny a neregistrovaný čtenář tak dostává výběr záznamu o knihách podle zadaných kritérií

Datový tok č.7

- **potvrzení rezervace** – člen resp. registrovaný čtenář knihovny dostává zprávu od systému o uskutečnění **rezervace**
- **seznam knih** – systém nabízí odpověď registrovanému čtenáři na:

  - **dotaz na knihy**, což je vyhledání knih v databázi knihovny a člen tak dostává výběr záznamu o knihách podle zadaných kritérií
- **seznam vypůjčených knih** – člen knihovny dostává reakci knihovního systému na dotaz:

  - **vypůjčené knihy**, což je zobrazení čtenářem vypůjčených knih
- **osobní číslo** – každému novému členovi knihovny je přiděleno osobní číslo při:

  - **registraci člena**, což je vložení údajů o novém čtenáři do databáze registrovaných čtenářů
- **potvrzení akce** - následuje reakce knihovníka, či člena nebo správce jenž provede akci na nespecifikovanou **žádost**

7.1.4. Diagram datových toků (Data Flow Diagram)

V předchozí kapitole jsme si tedy definovali a představili jednotlivé datové toky. Pro názornější sémantické vazby je dobré v návrhu představit i grafické představy těchto vazeb. K tomu slouží tzv. Diagramy datových toků známé pod zkratkou DFD (Data flow Diagram).

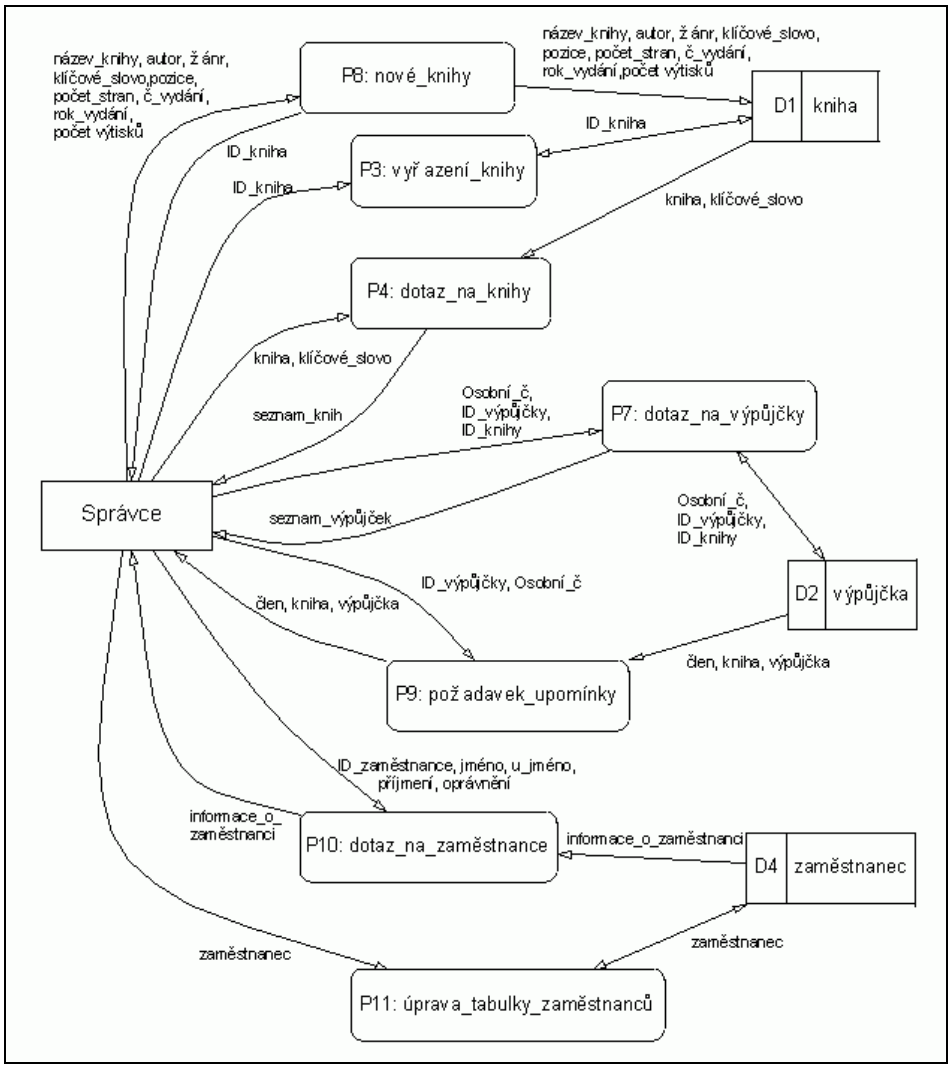
V průběhu akcí, které jsou v systému definovány na základě Entit, jejich atributů a relačních vazeb mezi těmito entitami, se ve vzniklých akcích musí definovat datové zásobníky. Pro náš záměr si definujme následující:

- D1 - kniha
- D2 - výpůjčka
- D3 - rezervace
- D4 - zaměstnanec
- D5 – člen

Tyto datové zásobníky si konkrétně představíme na modelech chování knihovního systému z pohledu správce, knihovníka, registrovaného člena knihovny a pouhého neregistrovaného návštěvníka.

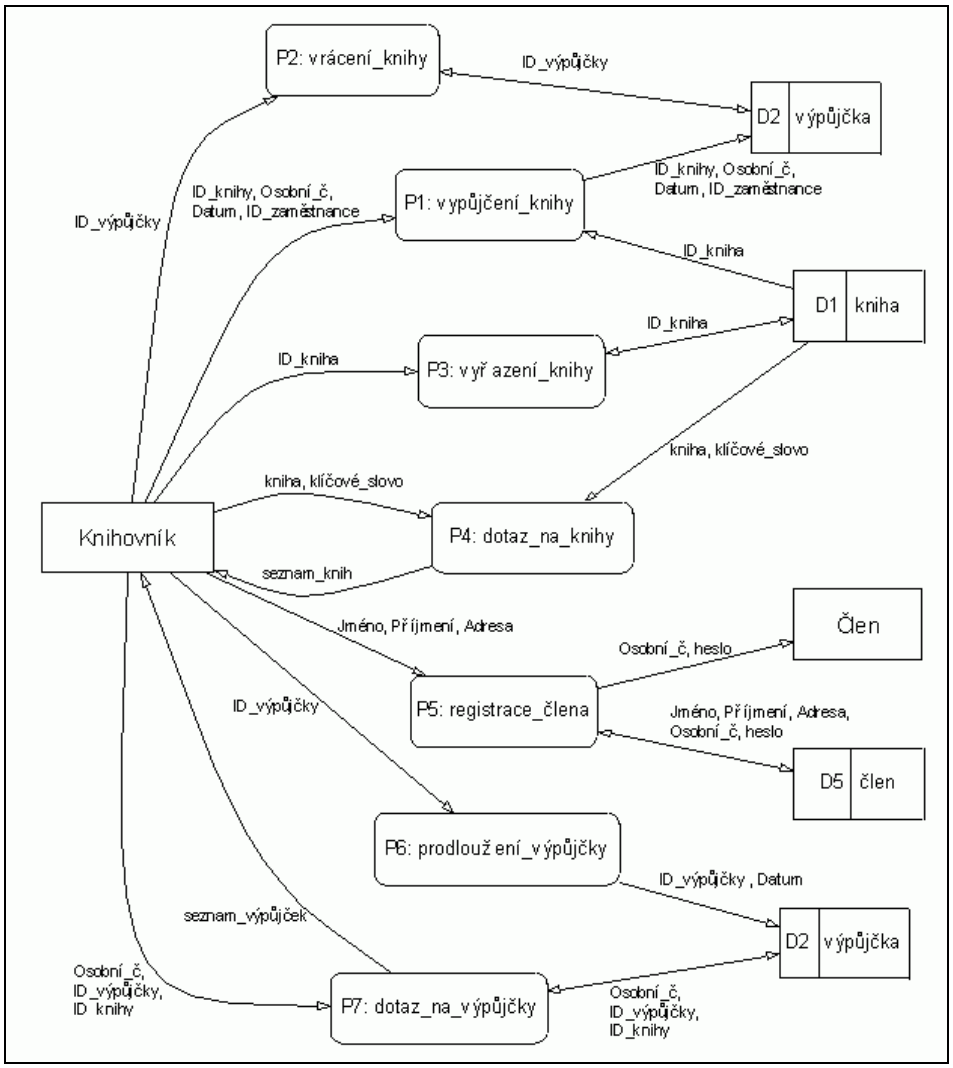
7.1.5. DFD na straně správce

Jednotlivé modely a nastavení systému jsou postaveny nejen z datových toků a datových zásobníků, ale hlavně i z různých procesů. Tyto procesy budou popsány v následující kapitole (ve formě tabulek) až po představení jednotlivých modelů.



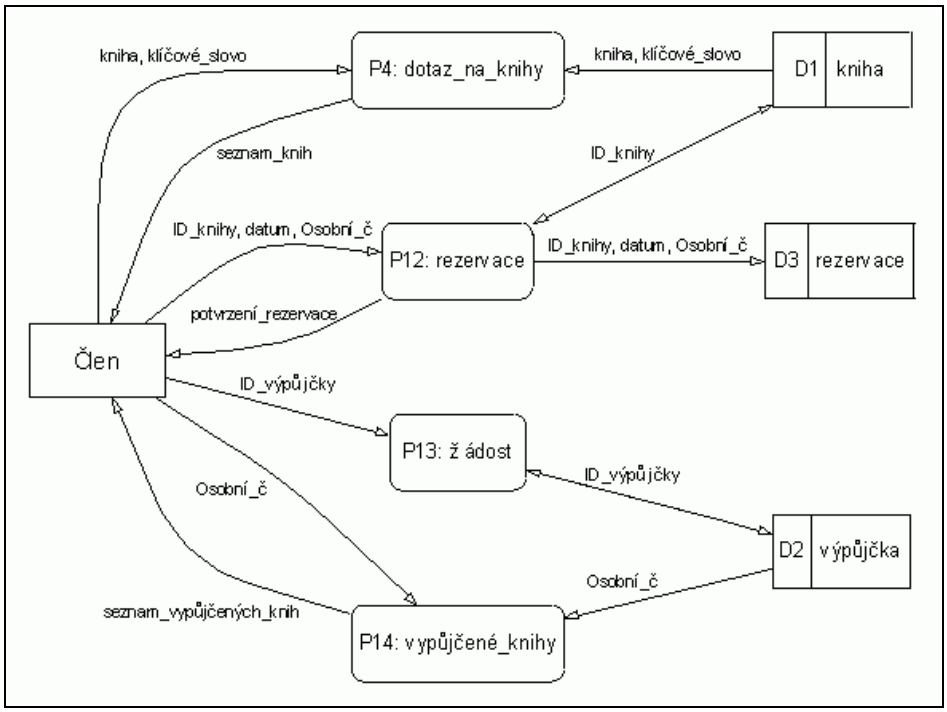
Obr. 19: Diagram datových toků pro správce systému

Na obr. 19 je patrné, že veškeré akce uskutečněny správce systému mají přístup ke všem definovaným datovým tokům. Tyto datové toky vedou i ke všem definovaným datovým zásobníkům. Co z toho plyne? Správce systému má oprávnění ke všem akcím spojených s vyřizováním upomínek na knihy, vkládání a odebíraní knih z fondu knihovny, úpravě dat v databázi zaměstnanců, půjčování knih atd.



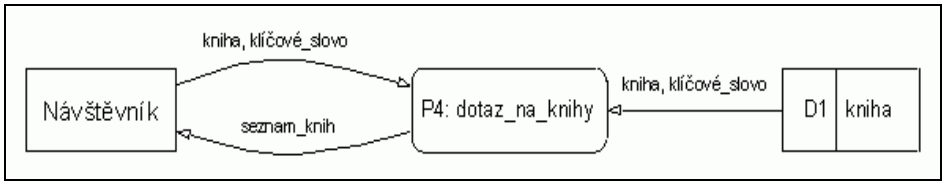
Obr. 20: Diagram datových toků definovaných pro knihovníka

Porovnáním obrázků 19 a 20 zjistíme, že řádný zaměstnanec knihovny resp. knihovník má možnost uskutečnit libovolné akce spojené s registrací nových členů, s vypůjčováním knih, rezervací knih čtenářům. Avšak dle nastavení systému již nemá oprávnění plnit akce spojené s vkládáním nových knih do fondu knihovny, úpravě dat v databázi zaměstnanců či vyřizováním upomínek. Na to má právo jen správce systému.



Obr. 21: Diagram datových toků definovaných pro registrovaného člena

Z obr. 21 je patrné, že došlo opět k redukci akcí spojených s přístupem řádného člena knihovny. Čtenář má nárok na rezervaci knih, na výpůjčku knih. Avšak v žádném případě nemůže čtenář zasahovat do databáze knih za účelem vkládání a vyřazování knih z fondu knihovny. Čtenář pochopitelně nemá nárok přístupu do databáze čtenářů a zaměstnanců knihovny.



Obr. 22: Diagram datových toků definovaných pro neregistrovaného člena

Neregistrovaný čtenář nemá v knihovně prakticky žádnou možnost využít služeb knihovny. Systém je však nastaven tak, aby měl neregistrovaný návštěvník alespoň možnost nahlédnout do databáze knih. Tuto funkci reprezentuje diagram na obr. 22.

7.1.6. Popis jednotlivých procesů

V diagramech datových toků se objevují v notaci články P1 až P14. Pojďme si nyní představit tyto procesy a formou tabulek definovat předpoklady, které daná funkce vyžaduje, stejně tak jsou představeny i požadavky na ostatní funkce při provádění a ukončení dané funkce a v neposlední řadě také popis akcí, které funkce provede.

P1	**vypůjčení_knihy**
Předpoklady	Správně vyplněná výpůjčka.
Požadavky	Výpůjčka čtenáři
Vstupní data	ID_knihy, Osobní_č, Datum, ID_zaměstnance
Výstupní data	ID_knihy, Osobní_č, Datum, ID_zaměstnance
Provedené akce	Zapíše se nová výpůjčka.

Tabulka XIX.: Definice procesu jenž se provede v modulu Knihovník

P2	**vrácení_knihy**
Předpoklady	
Požadavky	
Vstupní data	ID_výpůjčky
Výstupní data	ID_výpůjčky
Provedené akce	Výpůjčka se zruší.

Tabulka XX: Definice procesu jenž se provede v modulu Knihovník

P3	**vyřazení_knihy**
Předpoklady	Správně zadaná kniha.
Požadavky	
Vstupní data	ID_kniha
Výstupní data	ID_kniha
Provedené akce	Daná kniha je vyřazena z databáze.

Tabulka XXI: Definice procesu jenž se provede v modulu Správce

P4	dotaz_na_knihy
Předpoklady	
Požadavky	
Vstupní data	kniha, klíčové_slovo
Výstupní data	seznam_knih
Provedené akce	Funkce vrátí seznam knih, které vyhovují zadaným vstupním datům.

Tabulka XXII: Definice procesu jenž se provede v modulu Správce, Knihovník, registrovaný člen knihovny a návštěvník

P5	registrace_člena
Předpoklady	
Požadavky	
Vstupní data	Jméno, Příjmení, Adresa
Výstupní data	Jméno, Příjmení, Adresa, Osobní_č, heslo
Provedené akce	Funkce přidá do databáze nového člena a vygeneruje mu osobní číslo a heslo.

Tabulka XXIII: Definice procesu jenž se provede v modulu Knihovník

P6	prodloužení_výpůjčky
Předpoklady	Správně vyplněná výpůjčka.
Požadavky	
Vstupní data	ID_výpůjčky
Výstupní data	ID_výpůjčky, Datum
Provedené akce	Funkce změní u dané výpůjčky datum.

Tabulka XXIV: Definice procesu jenž se provede v modulu Knihovník

P7	dotaz_na_výpůjčky
Předpoklady	
Požadavky	
Vstupní data	Osobní_č, ID_výpůjčky, ID_knihy
Výstupní data	Osobní_č, ID_výpůjčky, ID_knihy
Provedené akce	Knihovník se dozví, co si půjčila daná osoba, co je předmětem dané výpůjčky, nebo kdo si půjčil danou knihu.

Tabulka XXV: Definice procesu jenž se provede v modulu Správce a Knihovník

P8	nové_knihy
Předpoklady	
Požadavky	
Vstupní data	název_knihy, autor, žánr, klíčové_slovo, pozice, počet_stran, č_vydání, rok_vydání, počet_výtisků
Výstupní data	název_knihy, autor, žánr, klíčové_slovo, pozice, počet_stran, č_vydání, rok_vydání, počet_výtisků, ID_kniha
Provedené akce	Funkce zapíše do databáze novou knihu a přiřadí jí ID.

Tabulka XXVI: Definice procesu jenž se provede v modulu Správce

P9	požadavek_upomínky
Předpoklady	Existence výpůjčky.
Požadavky	
Vstupní data	ID_výpůjčky, Osobní_č, člen, kniha, výpůjčka
Výstupní data	člen, kniha, výpůjčka
Provedené akce	Systém automaticky vypíše správci po uplynutí výpůjční doby údaje o knize, výpůjčce a „hříšníkovi“, který knihu nevrátil.

Tabulka XXVII: Definice procesu jenž se provede v modulu Správce

P10	**dotaz_na_zaměstnance**
Předpoklady	
Požadavky	
Vstupní data	informace_o_zaměstnanci
Výstupní data	informace_o_zaměstnanci
Provedené akce	Funkce vypíše informace o zaměstnancích, vyhovujících danému klíči.

Tabulka XXVIII: Definice procesu jenž se provede v modulu Správce

P11	**úprava_tabulky_zaměstnanců**
Předpoklady	
Požadavky	
Vstupní data	zaměstnanec
Výstupní data	zaměstnanec
Provedené akce	Funkce zahrnuje operace s databází zaměstnanců - jsou podrobně popsány jako funkce Q1, Q2, Q3.

Tabulka XXIX: Definice procesu jenž se provede v modulu Správce

P12	**rezervace**
Předpoklady	Správně zadaná kniha a člen.
Požadavky	
Vstupní data	ID_knihy, Datum, Osobní_č
Výstupní data	ID_knihy, Datum, Osobní_č, potvrzení_rezervace
Provedené akce	Funkce provede rezervaci dané knihy danému členovi.

Tabulka XXX: Definice procesu jenž se provede v modulu Člen

P13	žádost
Předpoklady	Správně zadaná výpůjčka.
Požadavky	Člen si knihu již půjčil.
Vstupní data	ID_výpůjčky
Výstupní data	ID_výpůjčky
Provedené akce	Člen zažádá o prodloužení výpůjčky.

Tabulka XXXI: Definice procesu jenž se provede v modulu Člen

P14	vypůjčené_knihy
Předpoklady	
Požadavky	
Vstupní data	Osobní_č
Výstupní data	seznam_vypůjčených_knih
Provedené akce	Funkce vrátí členovi seznam knih, které si půjčil.

Tabulka XXXII: Definice procesu jenž se provede v modulu Člen

7.1.7. Datový slovník

Datový slovník slouží jako průběžný obraz informačního systému. Je tvořen soustavou metadatových entit popisujících jednotlivé prvky IS pomocí jejich atributů a vzájemných vazeb. Vzniká většinou aktivitou projektanta.

Víme tedy, jaké akce reprezentují jednotlivé interaktivní vazby mezi člověkem a knihovním systémem. Pojďme si však jednotlivé akce představit a převést do matematických modelů. Pro takovou konverzi zavedeme následující notaci:

- = skládá se
- + a
- (...) může chybět

- {...} opakování
- [...|...] jeden z možných
- *....* komentář
- @ klíčová položka

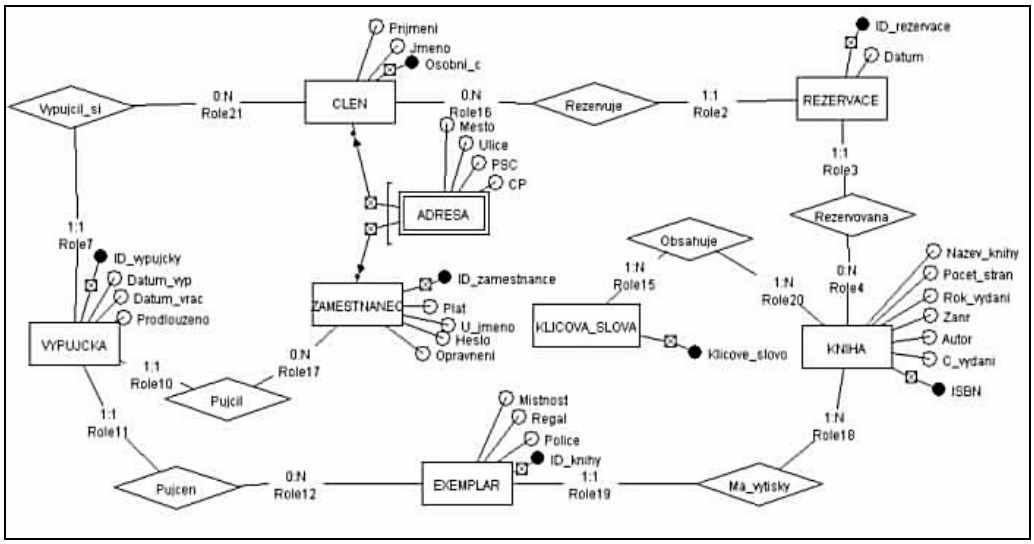
S touto základní notací lze tedy převést všechny již představené akce do podoby rovnic ve tvaru:

- vypůjčení_knihy = ID_knihy + Osobní_č + Datum + ID_zaměstnance
- vrácení_knihy = ID_výpůjčky
- vyřazení_knihy = ID_knihy
- registrace_člena = Jméno + Příjmení + Adresa
- prodloužení_výpůjčky = ID_výpůjčky
- dotaz_na_knihy = (název_knihy) + (autor) + (žánr) + (ID_knihy) + {klíčové_slovo} + (rok_vydání)
- dotaz_na_výpůjčky = [Osobní_č | ID_výpůjčky | ID_knihy]
- nové_knihy = název_knihy + autor + žánr + { klíčové_slovo } + pozice + počet_stran + č_vydání + rok_vydání + počet výtisků
- dotaz_na_zaměstnance = [ID_zaměstnance | jméno | u_jméno | příjmení | oprávnění]
- úprava_tabulky_zaměstnanců = [vložení_zaměstnance | vyřazení_zaměstnance | úprava_zaměstnance]
- vložení_zaměstnance = jméno + příjmení + u_jméno + heslo + oprávnění
- vyřazení_zaměstnance = ID_zaměstnance
- úprava_zaměstnance = ID_zaměstnance + (jméno) + (příjmení) + (u_jméno) + (heslo) + (oprávnění)
- rezervace = ID_knihy + datum + Osobní_č
- žádost = ID_výpůjčky
- vypůjčené_knihy = Osobní_č

- potvrzení_akce = potvrzení akce systémem
- seznam_knih = { kniha + přítomna }
- přítomna = [ANO | NE]
- ID_knihy = číslo
- seznam_výpůjček = { výpůjčka }
- informace_o_zaměstnanci = { ID_zaměstnance + jméno + příjmení + u_jméno + oprávnění }
- tisk_upomínky = * text upomínky s vyplněnými údaji o členovi, výpůjčce a knize*
- upozornění_na_upomínky = { ID_výpůjčky + Osobní_č }
- potvrzení_rezervace = * potvrzení rezervace výpůjčky knihy daným členem na dané datum *
- seznam_vypůjčených_knih= { kniha + datum + ID_výpůjčky}
- Osobní_číslo = Osobní_č
- kniha = @ID_knihy + název_knihy + autor + žánr + pozice + počet_stran + č_vydání +rok_vydání
- výpůjčka = @ID_výpůjčky + ID_knihy + Osobní_č + Datum + ID_zaměstnance
- ID_zaměstnace = číslo
- Osobní_č = číslo
- ID_výpůjčky = číslo

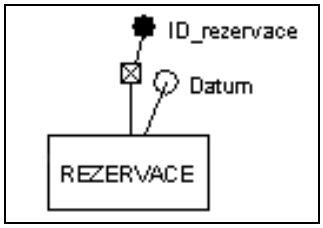
7.2. Analytická dokumentace

Projekt knihovního systému pro malou knihovnu se musí řídit určitým schématem. Pro databázový systém je nejlépe použít konceptuální schéma, jenž zachycuje sémantický model a umožňuje popsat konceptuální úroveň systému. Tento model nazýván E-R a jeho schématické značky byly již popsány v kapitole 3.. Přejděme však už ke konkrétnímu schématu navrhovaného systému pro malou knihovnu na obr. 23.



Obr. 23: E-R diagram znázorňující množiny entit.

Pro tento datový model jsme použili již známé objekty z reality a vztahy mezi nimi, které nás budou zajímat. Jednotlivé entity v diagramu mají i své atributy. Nyní si toto schéma rozebereme. Pro začátek nám poslouží entita *REZERVACE*. Podívejme se na obr. 24, na němž je tato entita vybrána z celého schématu.



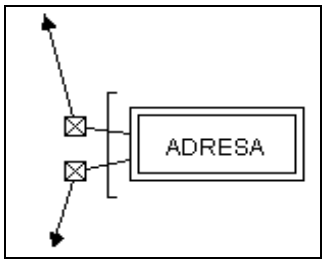
Obr. 24: Schématická značka entity REZERVACE

Všimněme si rozdílné vazby mezi entitou a atributy Datum a atributem ID_rezervace. Druhý zmíněný atribut je totiž klíčový. Podle schématu na obr. 23 je snadno rozeznat i relační vazby mezi entitou *REZERVACE* a relací rezervuje. Do schématu jsou tyto vazby popsány jako role. Tedy entita *REZERVACE* v relaci Rezervuje (role č.2) je ve vztahu 1:1. To tedy znamená a loga to podporuje, že pokud člen knihovny si rezervuje knihu, pak je tento dotaz možný zodpovědět jen jedním možným způsobem. Analogicky to platí i pro relaci Rezervovana

(role č.4), kdy je sytém opět tak nastaven, aby vykonal pouze rezervaci jedné knihy. Tedy jiný relační vztah než-li 1:1, nelze použít.

Cílem dalšího popisu bude entita *KNIHA*. Klíčovým atributem této entity je kód knihy ISBN. Jedná se o jedinečné číslo a nehrozí, že by se v databázi objevilo pro více různých knih duplicitně. Jako neklíčové atributy této entity jsou použity atributy, které se běžně v knihovnické praxi používají k identifikačnímu popisu díla: Nazev_knihy, Počet_stran, Rok_vydani, Zanr, Autor, Vydani. Tentokrát v roli č.18 relace *Ma_vytisky*, je vztah 1:N. Pochopitelně že systém předpokládá, že jedna kniha může být v systému a fyzicky na regále v několika exemplářích. Stejným způsobem lze popsat i relaci Obsahuje (role č.20), kde je opět vazba 1:N, neboť kniha se v systému identifikuje i na základě několika klíčových slov. Těžko si lze připustit vztah 1:1, protože žádné dílo nelze charakterizovat pouze jedním klíčovým slovem. Nezvyklý vztah je v relaci Rezervovana (role č.4). Vztah 0:N je vlastně jakousi značkou, která pro řidiče znamená Zákaz vjezdu jedním směrem. Tato relace je propustná pouze z entity *REZERVACE*, neboť v této relaci musí reagovat jen entita *KNIHA* na entitu *REZERVACE* a nesmí tak být opačně.

Takto lze ve vysvětlování postupovat od relace k relaci a od entity k entitě. Ovšem zvláštní pozornost si zaslouží entita *ADRESA*. Podívejme se na obr. 25, kde je v detailu tato entity zobrazena z celkového schématu.

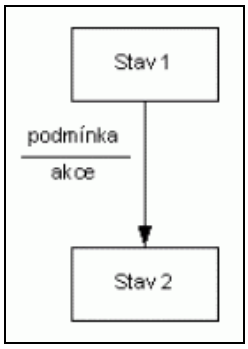


Obr. 25: Schématická značka výlučného vztahu

Na obr. 23 je patrné odlišné znázornění relačních vztahů mezi entitou *ADRESA* dvěma dalším entitám. Jedná se o tzv. výlučný vztah. Na celkovém E-R schématu jsou sousedícími entitami *CLEN* a ZAMESTNANEC. Znamená to tedy, že v databázi je jednotný systém ukládání pro zaměstnance i členy knihovny. Avšak na základě atributů jednotlivých entit se tyto seznamy navzájem nepřekrývají. Znamená to hlavně softwarovou úsporu při programování. Také z praxe lze uvést příklad, kdy i zaměstnanec knihovny je vlastně jejím řádným členem a vztahují se na něj pravidla nastavená v knihovním, resp. informačním systému.

7.2.1. Popis stavu systému

Diagram stavů a přechodů, jak je někdy tento prostředek nazýván, slouží pro modelování životního cyklu části systému svým rozsahem odpovídající jednomu objektu [ŠEŠERA, 2001, s. 14]. Přechod mezi jednotlivými stavy bývá vyvolán podnětem z vnějšího okolí, nejčastěji ve formě informace vložené přes uživatelské rozhraní, nebo jinou externí událostí. Na obr. 26 je předvedena základní notace použita ve stavovém diagramu.



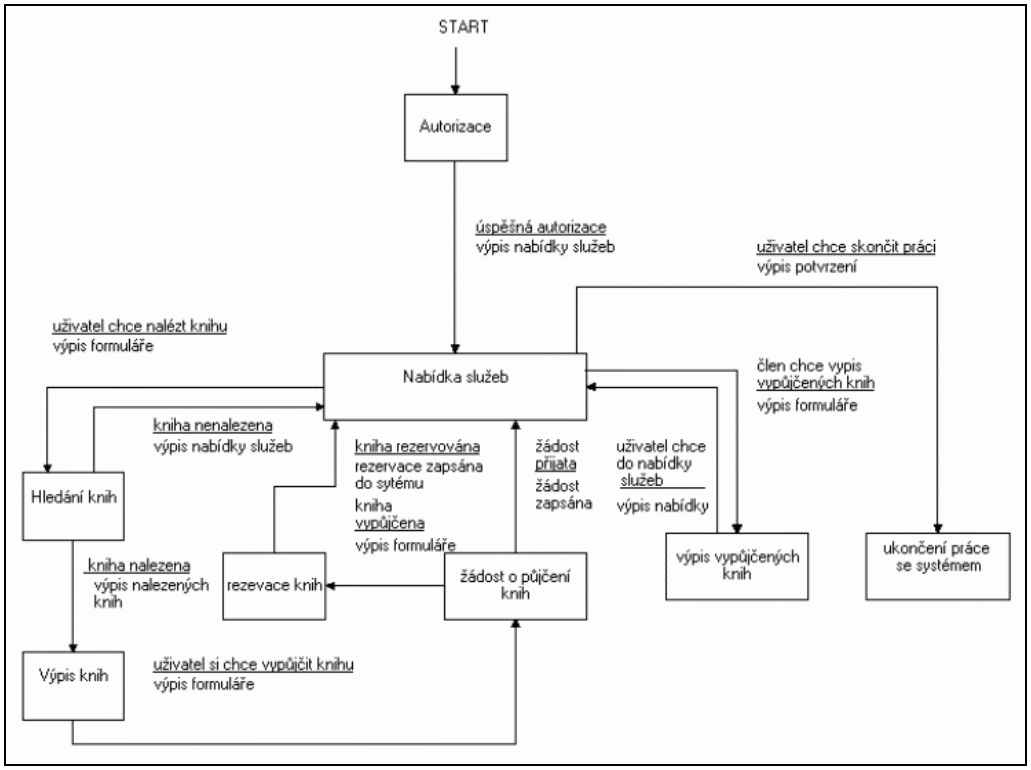
Obr. 26: Základní notace ve stavovém diagramu

Jak je asi zřejmé, životní cyklus objektu nějak začíná a zpravidla i nějak končí. Ovšem nemusí být již zcela samozřejmé, že tzv. start stav by měl být v diagramu vždy pouze jeden, zatímco stop stavů, tj. stavů

ukončujících životní cyklus sledovaného objektu může být více, pochopitelně většinou alespoň jeden. Nyní si představíme stavový diagram, tak jak ho představuje předkládaný návrh informačního knihovního systému. Z pohledu člena knihovny, knihovníka a správce systému.

7.2.1.1. Stavy systému z pohledu registrovaného člena

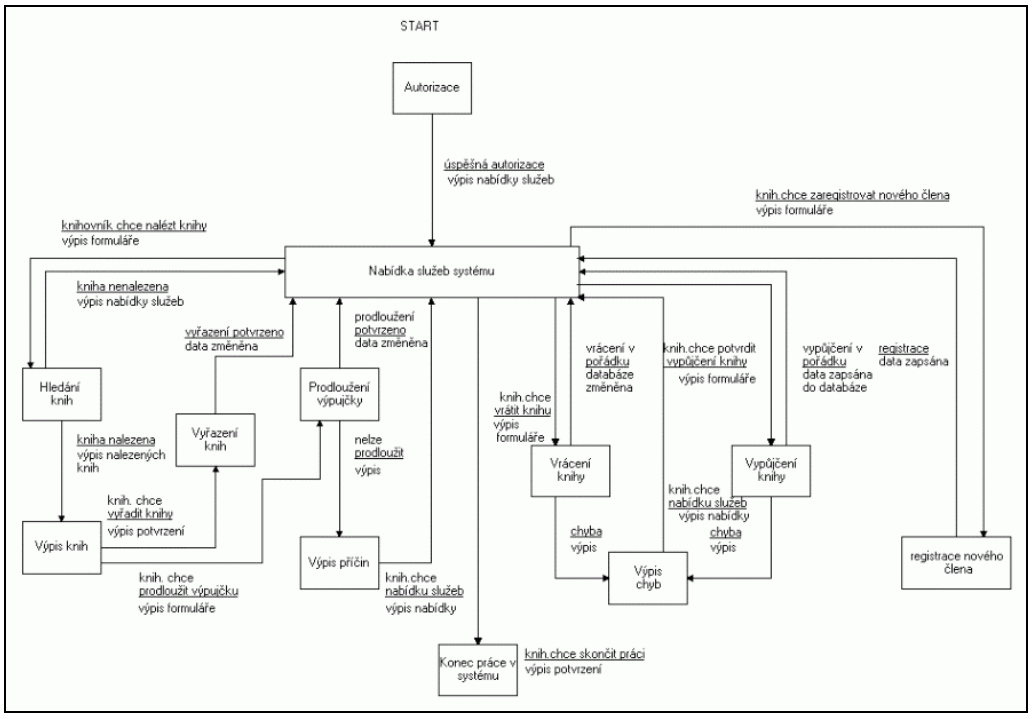
Na obr. 27 je návrh stavového diagramu informačního systému knihovny tak, jak odpovídá chování systému vůči registrovanému členovi knihovny. Startovací stavy jsou v souladu s definovanými datovými toky. Stejně tak navrhované uživatelské rozhraní nabízí veškeré vstupní a výstupní informace při dialogu s uživatelem (viz obr. 31 na straně 91).



Obr. 27: Stavový diagram pro registrovaného člena

7.2.1.2. Stavy systému z pohledu knihovníka

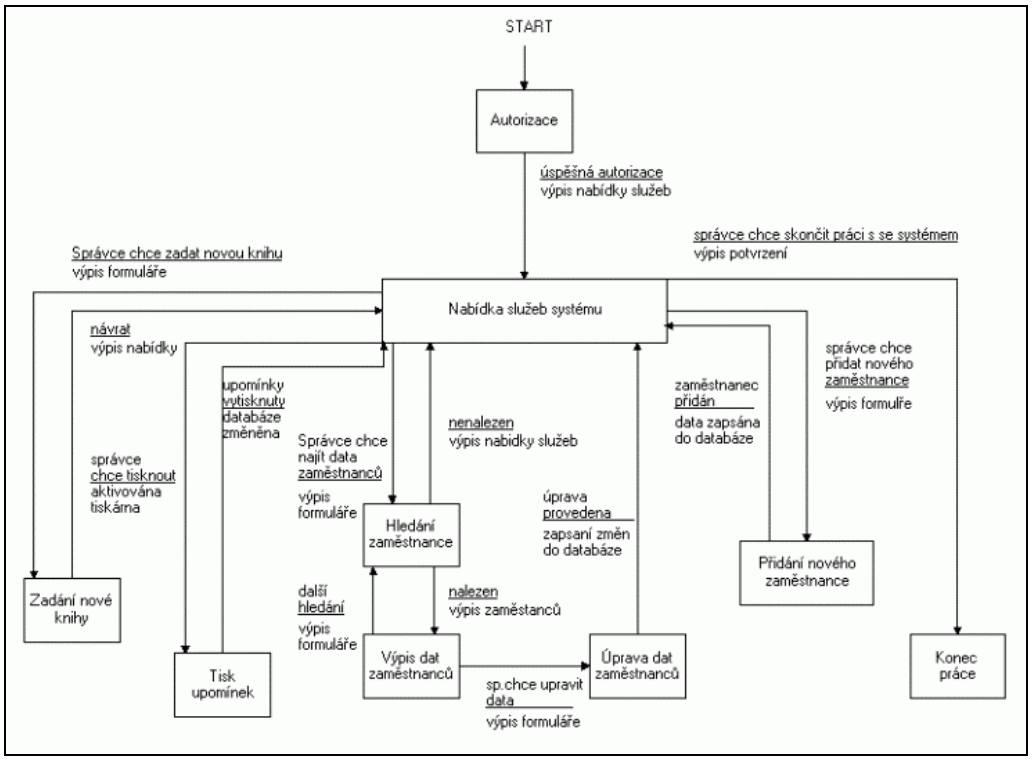
Na obr. 28 je další stavový diagram informačního systému knihovny tak, jak odpovídá chování systému vůči knihovníkovi. Jednotlivé startovací stavy jsou v souladu s definovanými datovými toky. Stejně jako v předešlém případě, navrhované uživatelské rozhraní nabízí opět veškeré vstupní a výstupní informace při dialogu s uživatelem (viz obr. 32 na straně 92).



Obr. 28: Stavový diagram pro knihovníka

7.2.1.3. Stavy systému z pohledu správce

Stavový diagram znázorněn na obr. 29 opět odpovídá požadavkům správce systému na informační systém knihovny. To znamená, že máme možnost opět vidět navržené chování systému a dle předložených stavových diagramů lze snadno naprogramovat libovolné uživatelské grafické rozhraní. Předkládané návrhy uživatelského rozhraní, jak budou prezentovány v následující kapitole jsou pouze ilustrativní a vznikly jen jako grafický doprovod právě k popisovaným stavovým diagramům.



Obr. 29: Stavový diagram pro správce systému

7.3. Uživatelská rozhraní

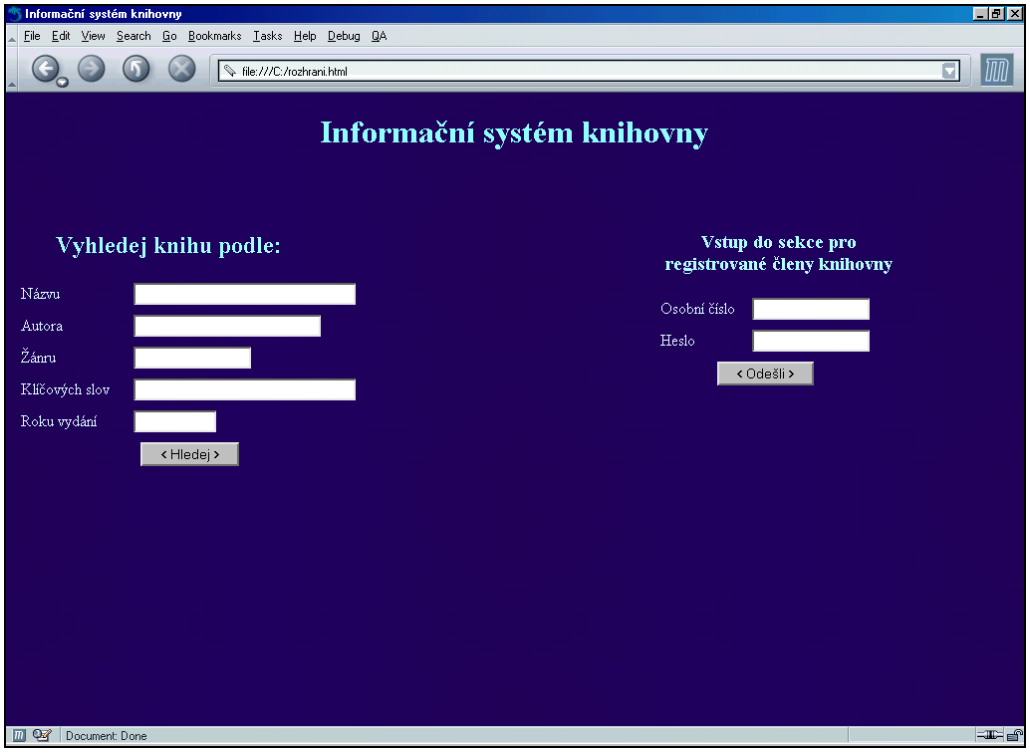
Uživatel považuje za knihovní systém přímo uživatelské rozhraní systému; všechno ostatní jsou prostě maličkosti, které můžeme s ledovým klidem ignorovat.Dobrý návrh uživatelského rozhraní je tudíž pro úspěch nebo pád projektu životně důležitý. Můžeme v tichosti předpokládat, že v dobře navrženém uživatelském rozhraní nám uživatelé odpustí občasnou nešťastně zvolenou implementaci. Naopak v nevhodně navrženém uživatelském rozhraní, nebude uživatele systému zajímat, nakolik efektivně je systém projektovaný.

Nejdůležitější princip při rozhodování o struktuře budoucího rozhraní můžeme vyjádřit zhruba takto: konkrétní zvolená architektura musí vycházet z pracovních procesů, které má systém podporovat, nikoli ze struktury dat. Sledujme, jaké úkoly se uživatelé snaží plnit a strukturu systému přizpůsobme s ohledem na podporu těchto aktivit. Podíváme-li

se do diagramu entit a vztahů (E-R diagramu na straně 84, obr. 23) knihovního systému, najdeme v něm entitu *CLEN*. Vytvoříme tedy formulář Registrovaní členové. Protože se snažíme vytvořit uživatelsky příjemný systém, umožníme registrovanému čtenáři nejenom knihu vyhledat, ale i v případě absence knihy provést její rezervaci, či vykonat prodloužení výpůjční doby na knihu čtenářem zapůjčenou.

7.3.1. Neregistrovaní návštěvníci

Na obrázku 30 je navržený formulář uživatelského rozhraní vytvořený ve formátu html stránky.

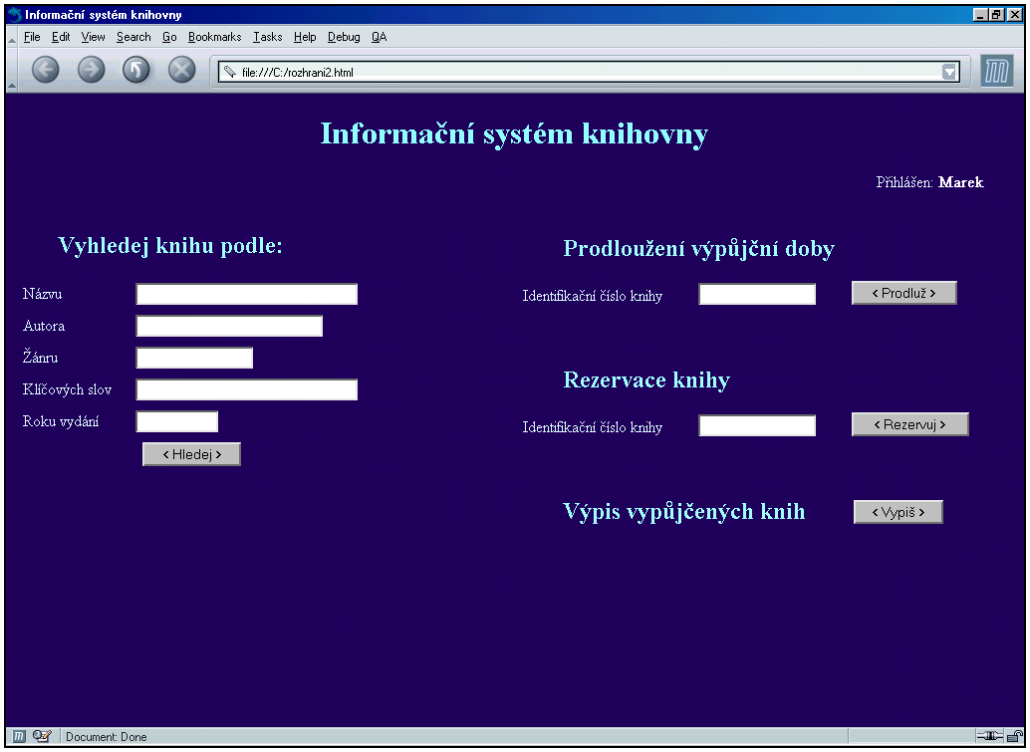


Obr. 30: Příklad rozhraní pro neregistrované návštěvníky.

Pomocí tohoto rozhraní má návštěvník knihovny, nikoli však registrovaný člen, možnost pohodlně nahlížet do databáze knihovního fondu a dostává výběr záznamu o knihách podle zadaných kritérií.

7.3.2. Registrovaní čtenáři

Na obrázku 31 je navržený formulář uživatelského rozhraní pro registrované členy knihovny. Jak již bylo řečeno, formulář nabízí uživateli nejen prohlížet data v režimu pouze pro čtení, ale čtenář má pomocí tohoto formuláře přímo zasáhnout do systému s požadavky na rezervaci knihy resp. zažádat o prodloužení výpůjčky.



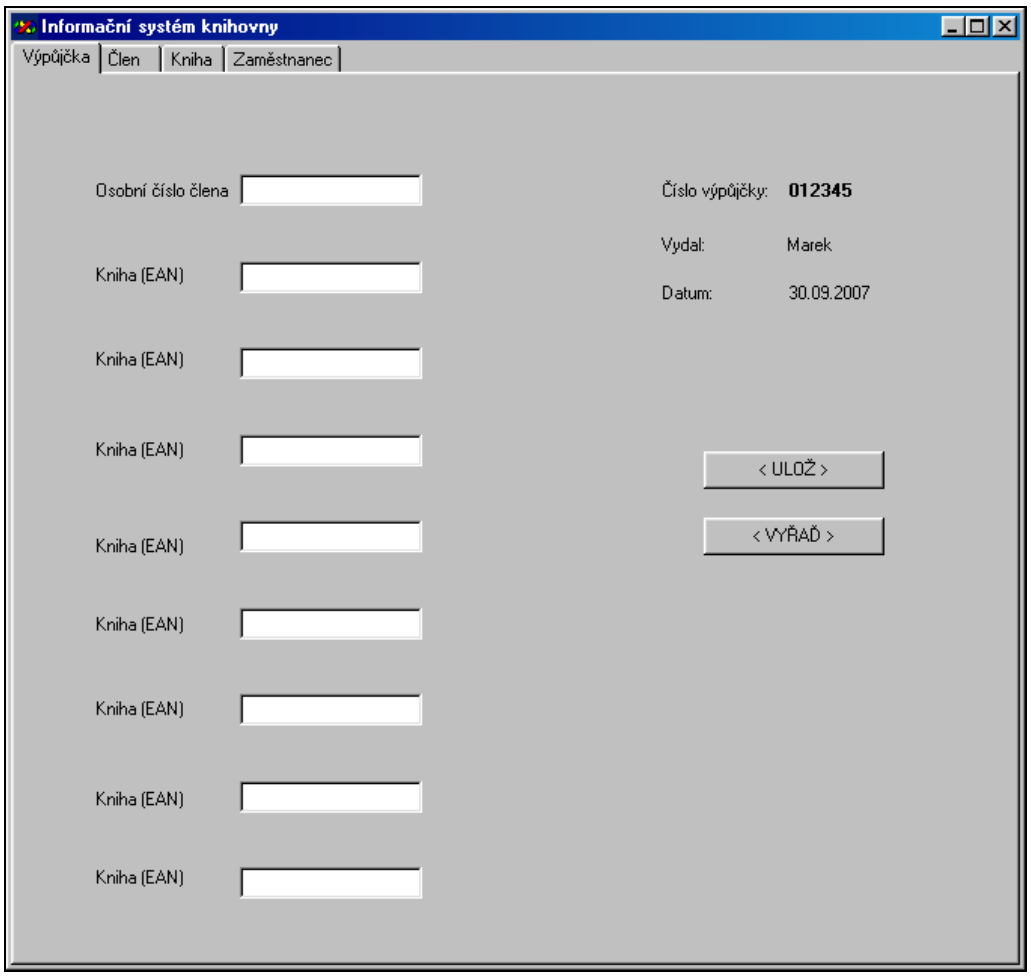
Obr. 31: Příklad rozhraní pro registrované čtenáře.

Čtenář však v žádném případě nemůže zasahovat do databáze knih za účelem vkládání a vyřazování knih z fondu knihovny. Čtenář pochopitelně nemá nárok přístupu do databáze čtenářů a zaměstnanců knihovny.

7.3.3. Návrh aplikace pro knihovníka

Pro účely spolupráce knihovníka se systémem bylo navrženo rozhraní tak, aby knihovník měl možnost uskutečnit libovolné akce spojené s registrací nových členů, s vypůjčováním knih, rezervací knih

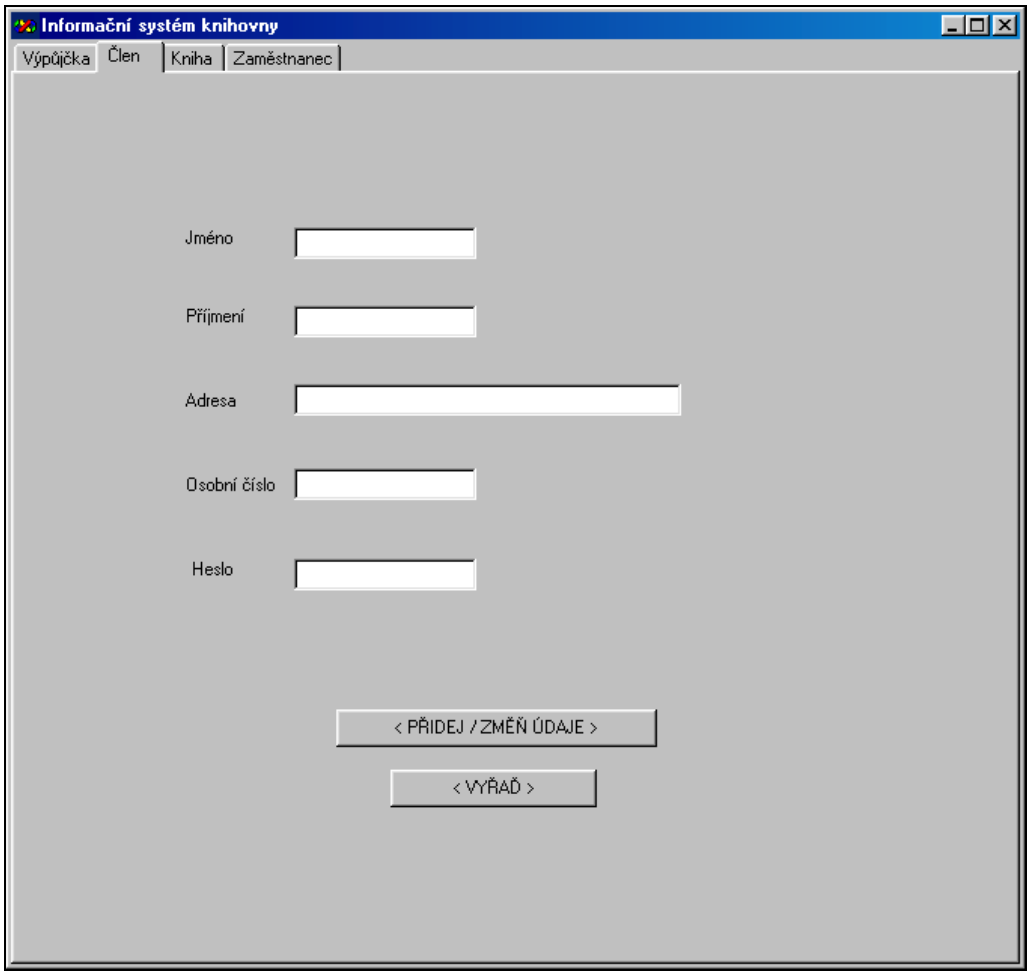
čtenářům. Dle navrženého rozhraní na obr. 32 je patrné, že knihovník může využít jednoduchý dialog pro manipulaci s výpůjčkami registrovaného člena knihovny. Dialog je navržen tak, že čtenář je limitován maximálním počtem osmi výpůjček. K dispozici je jednoduché vkládání EAN kódů jednotlivých výpůjček, ať už ročním zadáním, nebo pomocí čtečky čárových kódů. Pomocí tlačítek Ulož a Vyřaď má knihovník možnost manipulovat s výpůjčkami, které čtenář buď vrací do knihovny a nebo si je naopak z knihovny půjčuje.



Obr. 32: Příklad grafického rozhraní aplikace pro knihovníka

Na obr. 33. u je navržené rozhraní pro dialog knihovníka se systémem, pro účely změny údajů o registrovaném členovi knihovny. Dialogový formulář obsahuje všechny podstatné údaje, které slouží

k identifikaci čtenáře. Pokud zaměstnanec, resp. knihovník potřebuje změnit osobní údaje čtenáře, má možnost po vyplnění Osobního čísla čtenáře možnost data ve formuláři měnit. Dialogový formulář je doplněn o nepovinný blok se zadáním hesla.

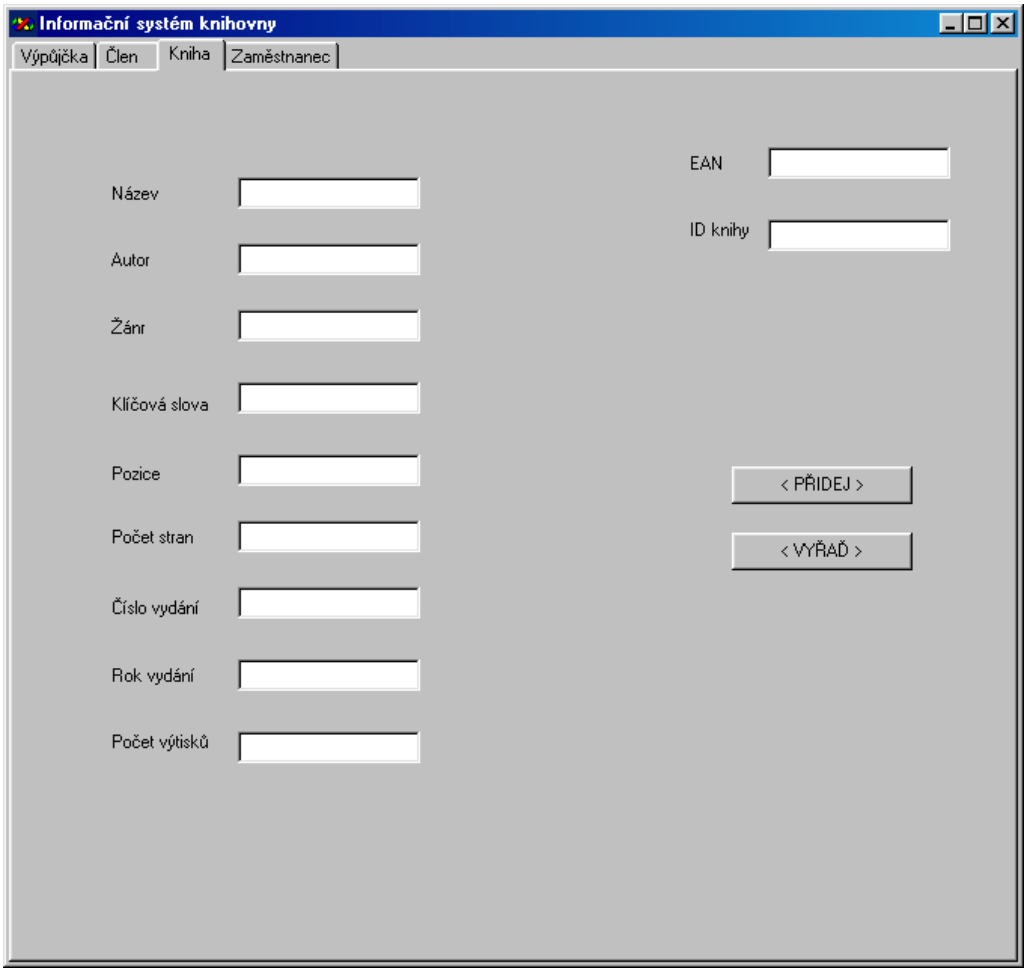


Obr. 33: Příklad grafického rozhraní aplikace pro knihovníka

7.3.4. Návrh aplikace pro správce systému

Parametry systému byly navrženy a nastaveny tak, že Správce systému má oprávnění ke všem akcím spojených s vyřizováním upomínek na knihy, vkládáním a odebíráním knih z fondu knihovny, úpravě dat v databázi zaměstnanců, půjčování knih. Má oprávnění plnit akce spojené s vkládáním nových knih do fondu knihovny, úpravě dat v databázi návštěvníků i zaměstnanců, či vyřizováním upomínek.

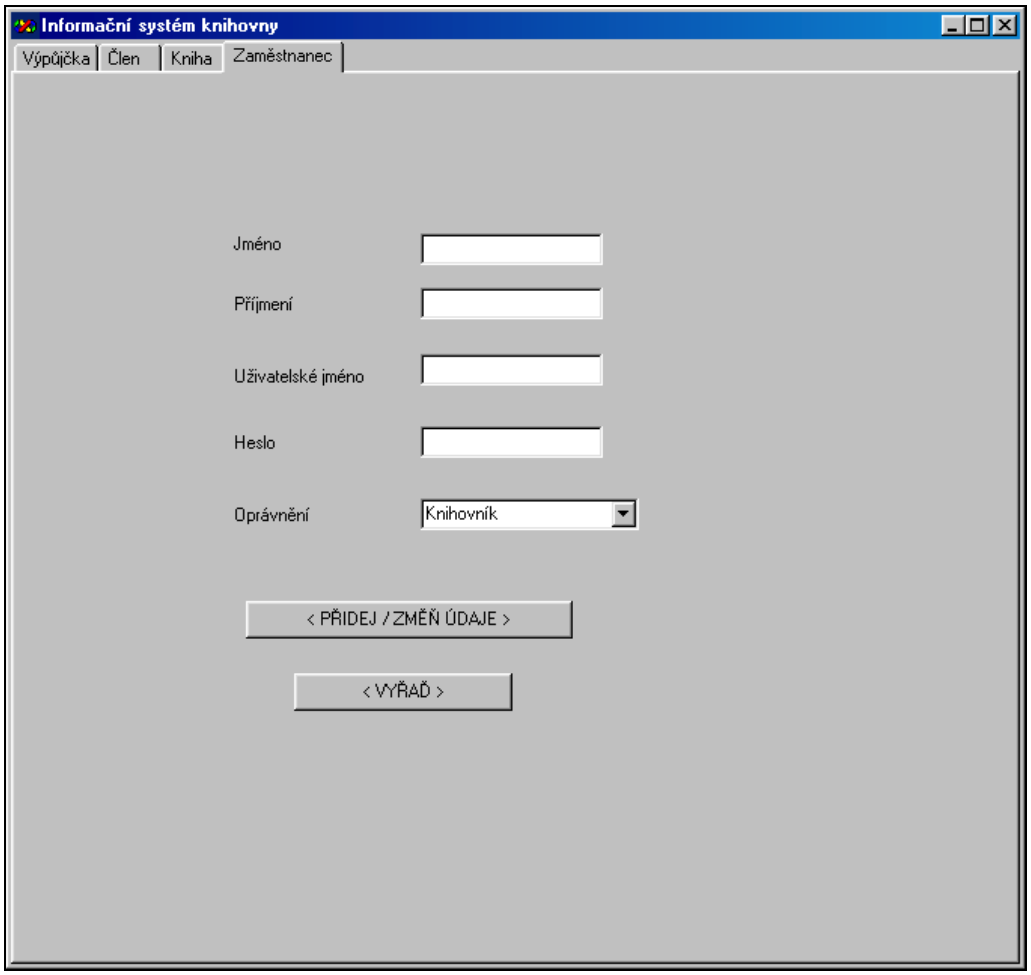
Na obrázku 34 je dialogový formulář pro manipulaci s knihou resp. s knihovním fondem. Správce systému, v našem případě osoba, která má na starosti vkládání nových knih a vyřazování knih z fondu, pracuje s poli, která jsou k základnímu identifikačnímu popisu díla nutná vyplnit. Pakliže se správce rozhodne knihu z fondu vyřadit, použije přímo pole EAN a veškeré dostupné údaje o titulu se automaticky vyplní do textových polí. K operaci vkládaní nebo vyřazení knihy z fondu postačí tlačítka PŘIDEJ a VYŘAĎ v pravé části dialogového formuláře.



Obr. 34: Příklad grafického rozhraní aplikace pro správce

Poslední záložka Zaměstnanec v aplikaci Informačního systému knihovny náleží opět do kompetence Správce systému. Dle obr. 35 má správce systému možnost manipulovat s databází zaměstnanců knihovny.

Tento dialogový formulář tedy reprezentuje nejvyšší autoritu knihovního informačního systému.



Obr. 35: Příklad grafického rozhraní aplikace pro správce

8. Automatizované knihovnické systémy

V současné době se používání databázových aplikací v knihovnách a informačních střediscích stalo běžným jevem. Nové informační technologie, zvláště integrace telekomunikační a výpočetní techniky, umožňují splnit základní cíle automatizace, kterými jsou kvalita a efektivnost služeb a zefektivnění činnosti knihovny [MÍKA, 2000].

Automatizace v knihovnách je tedy proces, jehož cílem je využívání prostředků telekomunikační a výpočetní techniky s cílem zvýšení produktivity práce, zlepšení operativnosti a kultury obsluhy čtenářů a uživatelů a umožnění využívání nových moderních typů služeb.

Nabídka automatizovaných knihovnických systémů v České republice je široká, existují však různá hlediska pro jejich hodnocení. Každá instituce, která se rozhodne pro automatizaci knihovnicko-informačních procesů, by se měla seznámit s nabídkou jednotlivých systémů a podle svých aktuálních potřeb a požadavků by mělo dojít k výběru vhodného integrovaného či samostatného automatizovaného knihovnického systému. Proto je také předmětem této práce předložit stručný přehled a popis knihovnických systémů, které jsou navrženy na relační databázový model.

8.1. Nejvýznamnější české automatizované knihovnické systémy

8.1.1. LANius

Tento knihovnický systém vyvíjí firma Lanius, s.r.o. v úzkém kontaktu s Městskou knihovnou v Táboře, kde je také testován a využíván. Vzniká speciálně pro používání ve veřejných knihovnách či knihovnách podobného typu, jako knihovna v Táboře, jejichž fond

obsahuje maximálně 150000 svazků. Má výhodu v tom, že neobsahuje nevyužitelné části, které často komplikují přehlednost a použitelnost programu v běžné praxi. Základní záznam je konstruován s ohledem na ty údaje, které jsou potřebné pro vyhledávání ve veřejných knihovnách [OTRUBOVÁ, 2006].

LANius řeší řadu základních i speciálních činností, které na sebe navazují nebo jedna druhou ovlivňují. Společná pro ně je práce s určitou množinou dat (informací). Protože určité skupiny prací jsou úzce časově či obsahově blízké, jsou sdružovány a zpracovávány jedním programem, který se nazývá modul [ŠILHA, 2006].

Každý z modulů se chová jako samostatný program se všemi potřebnými volbami. Je důležité, že všechny moduly pracují nad společnou datovou základnou a jsou tedy úzce svázány. To znamená, že změníme-li data v jednom modulu, tak se změna projeví na příslušných místech i v ostatních modulech. Přitom je možné používat jeden či více modulů samostatně s možností postupné kompletace systému [ŠILHA, 2006].

8.1.1.1. Moduly knihovnického systému LANius

Modul Vstup údajů – základní část celého knihovnického systému. Zabezpečuje vkládání a údržbu o fondech knihovny. Jeho hlavní činností je možnost vkládání záznamů o dokumentech z mnoha pracovišť najednou. Efektivnost práce také zajišťují slovníky autorů, nakladatelství, edic či klíčových slov. Mimo hlavní knižní fond umožňuje režim zpracovávat regionální a výměnný fond. Vytváří a eviduje objednávky. S těmito základními informacemi pak pracují ostatní moduly a proto bez Modulu Vstupu údajů nemohou pracovat ostatní části systému. Základní modul usnadňuje namáhavou a časově

náročnou práci knihovníků v odděleních doplňování a zpracování knihovního fondu.

Modul On-Line katalog – řeší vlastní využívání uložených informací. Umožňuje vyhledávání libovolných údajů o fondech. Jeho předností je jednoduchá metoda zadávání dotazů. Výsledky vyhledávání jsou k dispozici na obrazovce nebo v několika variantách tištěných výstupů do sestav. Daný modul pracuje ve dvou základních režimech: [5]

- je určen pro knihovníky a umožňuje vyhledávání všech sledovaných údajů. Používá speciální dotazovací list, pomocí kterého lze zadat i nejsložitější kombinované dotazy přehlednou formou
- je určen pro vyhledávání podle požadavků uživatele, nahrazuje dosavadní katalogy. Je řešen tak, aby byl snadno použitelný.

Modul Výpůjční protokol – ulehčuje práci v půjčovnách. Umožňuje vkládání, opravy a vyhledávání údajů o čtenářích a výpůjčkách. Zároveň obsahuje režim tisku štítků čárového kódu určeného pro evidenci čtenářů. Základní funkcí tohoto modulu však je vlastní provádění a evidence výpůjček. Vstupní identifikace čtenáře, půjčených a vrácených dokumentů se provádí pomocí čárového kódu. [5] Při práci je možné ruční vstup evidenčních čísel nahradit rychlým a spolehlivým snímačem čárového kódu (např. scanner, světelné pero atd.). Modul také obsahuje navazující funkce, jako je tvorba statistických výkazů, tisk upomínek a rezervace dokumentů.

Modul Evidence periodik – slouží k automatizaci denní evidence došlých periodik pro vlastní knihovnu či kooperující knihovny v jiných obvodech. Dovoluje získat okamžitý přehled o počtu došlých čísel periodik, řeší informace o předplatném, o placení faktur a o dodavatelích. Podle běžné knihovnické praxe umožňuje kompletaci periodik pro

pozdější půjčování. Současně modul umožňuje uživatelům získávat konkrétní informace o určitém periodiku a pracovníkům usnadňuje práci při vytváření základních podkladů pro urgence. [5]

Modul Komunikace mezi knihovnami – tento modul je praktickým řešením dostupnosti o fondech knihoven v rámci České republiky. V současnosti je spojován s aktivitami praktického a univerzálnějšího řešení propojení v rámci sítě Internetu, vytváření souborného katalogu, sdílená katalogizace a emailovou komunikaci při vyřizování MVS. [5]

Modul Analytický popis - zpracovaný záznam obsahuje kompletní soubor údajů pro popis článků periodik, včetně anotace neomezeného rozsahu a počtu klíčových slov. Modul dovoluje vyhledávání podle všech ukládaných informací ve dvou rovinách. Tiskové výstupy jsou pak koncipovány tak, aby respektovaly novou normu a mezinárodní směnitelnost záznamů. V rámci modulu je zajištěn import a export dat z jiných knihoven. [5]

Modul Statistika - zpracovává dvě hlavní oblasti, tj. statistický deník konkrétní knihovny a celoroční statistické výkazy a jejich sumáře pro požadovaný region [OTRUBOVÁ, 2006].

Modul Regionální databáze – řeší zpracování regionálních databází, zejména báze osobností a evidenci regionálních událostí. Dovoluje zpracovávat libovolné agendy potřebné pro veřejnou knihovnu [OTRUBOVÁ, 2006].

Modul Výměnné soubory – umožňuje snadnou a přesnou evidenci výměnných souborů knih. Na zvolenou knihovnu je možné vytvářet výměnné soubory v minimálně měsíčním intervalu a vlastní evidence knih je velmi snadná a využívá technologie čárového kódu.

Modul také umožňuje mezi knihovnami evidenčně převádět celé výměnné soubory nebo jejich části a ke konkrétní knihovně je možné zobrazit kompletní přehled vypůjčených dokumentů. Z modulu je možné tisknout řadu sestav a zpracovávat statistické přehledy. [5]

8.1.2. CLAVIUS

Knihovnický systém Clavius začal vznikat na konci roku 1997, kdy firma ArrowSys započala s vývojem zcela nového knihovnického systému ve spolupráci se sdružením KAVKa se sídlem v Uherském Hradišti. Počátkem roku 1998 byl dokončen návrh struktury a ukládání dat a začal vlastní vývoj jednotlivých modulů. [5]

8.1.2.1. Moduly knihovnického systému CLAVIUS

Modul Akvizice – tvoří základní část tohoto systému. V režimu akvizice jsou definováni jednotliví odběratelé, včetně jejich rozpočtu, a dodavatelé. Požadované dokumenty jsou vkládány do hlavní báze a jsou volně dostupné v OPACu. Pro objednávky je možné vybrat určité tituly podle vlastních zadaných podmínek. V režimu objednávání dokumentu pak systém tiskne objednávky, nebo zasílá požadavky elektronickou poštou. [5] Systém zároveň sleduje lhůty plnění dodávek a generuje urgence. Realizované dodávky jsou sledovány také z pohledu čerpání rozpočtů a evidence dokladů.

Modul Katalogizace – zahrnuje zpracování všech druhů dokumentů, které probíhá do společné databáze. Standardně jsou k dispozici režimy pro katalogizaci monografií, AV médií, hudebnin, periodik, analytického popisu článků periodik. Pro jednotlivé dokumenty jsou připraveny vstupní formuláře, které jsou plně modifikovány a mezi jednotlivými vytvořenými záznamy je možné vytvářet vazby. Každé pole vstupního formuláře obsahuje odbornou nápovědu a příklady. Pro

usnadnění vkládaní údajů je systém vybaven osvědčenou technologií slovníků tvůrců kódů. Při ukládání jednotlivých polí systém kontroluje vyplnění povinných polí tak, jak jsou definována v konfigurační databázi. [5]

Modul Vyhledávání (OPAC) – obsahuje jednoduché vyhledávání podle jména autora, názvu dokumentu a věcného hesla, nebo je zadaný text vyhledáván ve všech polích současně. Pro usnadnění je zde také k dispozici pohled do slovníků a tezauru. Součástí modulu je i profesionální vyhledávání, ve kterém formou otázky a jejího spojení logickými operátory, lze zadat složené dotazy. Výsledky vyhledávání jsou standardně zobrazeny v přehledové tabulce, která dovoluje jejich seřazení či označení určitých záznamů pro další operace. Samozřejmostí modulu je možnost tisku, vyhledávání do přehledových sestav či export ve zvoleném formátu. [5]

Modul Výpůjční protokol – využívá technologii čárového kódu s možností doplnění informací ze skeneru. Součástí modulu je správa databáze uživatelů s možností zadání osobního finančního kreditu. Vlastní režim výpůjčního protokolu provádí evidenci absenčních i prezenčních výpůjček, prolongace, rezervace, tisk upomínek a oznámenek rezervací s možností využití elektronické pošty. Integrovány jsou i režimy na evidenci MVS. Součástí modulu je i komplexní statistický pohled do prováděné evidence a možnost vytváření propracovaných tiskových výstupních sestav. [5]

Modul Správa seriálů - Je součástí modulu katalogizace spolu s ostatními druhy dokumentů. Na úrovni titulů periodik je k dispozici popis údajů podle pravidel AACR2 včetně vazebních polí popisujících historii periodika. Vlastní denní evidence může probíhat pro denní objemy desítek kusů periodik nebo pomocí speciálního režimu hromadné

evidence periodik. Zde je možné provádět denní evidenci mnoha set exemplářů periodik (deníků, týdeníků) velice jednoduchou formou. Výsledkem evidence je přesný rozpis nákladů na každého odběratele za určité období. Rozpočty a evidence dokladů (objednávek, faktur a dobropisů) jsou součástí modulu akvizice. Samozřejmostí je automatická urgence nedodaných čísel. [5]

WWW Modul - pracuje na WWW serveru a primárním účelem je on-line vystavení databáze dokumentů knihovny v síti Internet. Díky tomu lze také v prostředí Intranetu realizovat možnost vyhledávání fondu knihovny pomocí běžného prohlížeče. Sekundárně modul slouží k odesílání písemných informací formou elektronické pošty na knihovní poštovní server. Další funkcí modulu je zajištění automatického zálohování dat systému popř. automatická aktualizace vzdálené databáze. [5]

Modul Dětský ONLINE katalog - Usnadňuje práci s počítačem dětem školního věku i těm, kteří ještě do školy nechodí. Formou symbolů - obrázků nabízí seznam nejžádanějších žánrů dětské beletrie a oborů lidské činnosti či vědy a kultury. Zároveň dovoluje plnohodnotné vyhledání záznamů pro starší děti, zkušené uživatele, podle jména autora, názvu a klíčových slov. Obrázková část pro nejmenší čtenáře je volně nastavitelná podle užívaných klíčových slov a zvyklostí knihovny [ŠILHA, 2006]. Celým katalogem děti provází animovaná postavička housenky.

Modul Správy a nastavení – nastavení uživatelů a jejich práv je součástí modulu Katalogizace. Nastavení parametrů výpůjčního protokolu se provádí speciálním samostatným modulem. [5]

Modul EMVS (emaily MVS a čtenářům) - Posílání MVS, upomínek a oznámení o rezervaci emailem. Tento modul spolupracuje s evidencí žádanek MVS a s výpůjčním protokolem. Pokud vaše organizace žádá MVS od knihovny, která má emailovou adresu, systém automaticky nabídne odeslání žádanky MVS prostřednictvím emailu. Kromě této základní funkce dovoluje modul EMVS velice snadno zjednodušit a zlevnit posílání oznámení o připravenosti rezervovaného dokumentu či o upomínce z prodlení čtenáři. Podmínkou je znalost adresy schránky elektronické pošty čtenáře popř. organizace. Pokud čtenář vlastní mobilní telefon s možností příjmu zpráv SMS, je možné, aby si zřídil emailovou adresu a informace o rezervacích či upomínkách dostával přímo na svůj mobilní telefon. Modul EMVS lze použít i pokud má vaše knihovna pouze telefonické připojení k Internetu. [5] [24]

Modul Z39.50 server - Tento modul zajišťuje vystavení fondu Vaší knihovny pomocí celosvětově používaného komunikačního protokolu. Ostatní knihovny mající klienta protokolu Z39.50 pak mohou hledat ve Vašem fondu a stahovat Vaše záznamy pro své vlastní použití. Z39.50 server také může zajistit snadné a plnohodnotné zapojení Vaší knihovny do Jednotné informační brány (JIB). [24] Protokol Z39.50 bude zevrubněji popsán v kapitole 9.

Modul Z39.50 client - Umožňuje vyhledávání ve vzdálených Z39.50 serverech (tuzemských i zahraničních knihoven) pomocí klienta, který je integrovaný do katalogizace. Obsahuje možnost přebírání bibliografických a autoritních záznamů do katalogizace z těchto knihoven. V případě rozšířené licence modulu o funkci „database update“ umožňuje klient i odesílání dat neboli ukládání záznamů do vzdáleného serveru. Používá se při kooperativní tvorbě souboru národních autorit v případě, že knihovna se chce aktivně zapojit

a přidávat vlastní návrhy nových autorit. Pro fungovaní klienta je nezbytné spolehlivé připojení k internetu tam, kde má být klient používán a ochota poskytovatele internetu či správce sítě, povolit komunikaci směrem ven z vaší organizace na určitém portu, pro určitý vzdálený server. Protokol Z39.50 bude zevrubněji popsán v kapitole 9.

8.1.3. KP-sys a KP-win

Firma KP-SYS spol.s.r.o dodává na náš trh integrovaný automatizovaný knihovnický systém KP-sys. Jedná se o systém, který pracuje pod operačním systémem MS-DOS. Export a import dat je zajištěn přes výměnný formát nebo UNIMARC. Na konci roku 1999 začala firma s distribucí nejnovější verze knihovnického systému KP-win. [6] Tato verze již plně pracuje v grafickém prostředí Windows.

S KP systémy mohou pracovat profesionální i neprofesionální knihovníci. Najdou uplatnění ve všech typech knihoven. Všechna data systémů jsou pravidelně zálohována a jednotliví pracovníci knihovny mají přístup přes hesla jen do předem definovaných funkcí. Data jsou kódována pro zamezení zneužití. Uživatelům knihovny je umožněn přístup do katalogu přes tzv. OPAC. Zpracovaný dokument je však uživatelům k dispozici v jakémkoli stupni záznamu. [6]

Při instalaci systémů knihovníci vybírají důležité prvky systémů. Definují se přístupová práva, velikost a tvar přírůstkových čísel, signatur, kategorií uživatelů i dokumentů pro výpůjčku. Jsou určeny vstupní formuláře pro jednotlivé druhy dokumentů a rovněž si knihovníci mohou vybrat z různých variant tiskových výstupů. Systémy umožňují knihovníkům provádět pomocí počítače všechny důležité agendy, bez zbytečné manuální a duplicitní práce. [6]

Všechny údaje je možné zapisovat pouze jedenkrát a důležitá pole v záznamu jsou kontrolována přes autority nebo validační slovníky. Systémy pracují s proměnnou délkou polí a záznamů. Pro rychlé pořízení dat v rámci retrospektivního zpracování fondu, je možné standardně využít externích katalogů např. ČNB nebo katalogů jiných knihoven. Retrospektivní zpracování je pak záležitostí nalezení záznamu v externí databázi a jednoduchého importu do vlastního formuláře. Pořízení dat se tím mnohonásobně urychlí. Oba systémy jsou modulární a všechny moduly lze zakoupit samostatně. [6]

8.1.3.1. Moduly knihovnického systému KP-sys a KP-win

Modul Akvizice – umožňuje evidovat požadavky na nákup dokumentů, jejich objednávku a následné dodání. Automaticky jsou generovány urgence na opožděné dodání dokumentů nebo na nevyřízení celé objednávky. V systémech jsou také automaticky vypočítávány náklady na více výtisků jednoho titulu, skutečné výdaje, ale i očekávané výdaje u zatím nedodaných knih. Automaticky se generují také přírůstková čísla pro celou knihovnu, či jednotlivé pobočky i podle typu dokumentu. [6] [24]

Modul Katalogizace – zde se provádí kompletní zpracování dokumentů. Jsou připraveny vstupní formuláře pro monografie, sborníky, výzkumné a cestovní zprávy, normy a ochranné dokumenty, diplomové práce, periodika, hudebniny, AV dokumenty, ale také články z časopisů a kapitoly ve sbornících. Uživatel má možnost definovat nebo upravovat vstupní formuláře. Struktura dat odpovídá formátu UNIMARC. Jednoduchost a jednotnost v editaci polí umožňují autority jmenné i věcné. Ostatní údaje lze vylistovat z předem připravených slovníků nebo ze slovníků již jednou uživatelem vložených. Uživatelům jsou k dispozici on-line specifikace jednotlivých údajů s příklady,které

jim umožní správné ukládání dat s přihlédnutím k AACR2. Součástí modulu je i kompletní správa seriálů, včetně akvizice. [6]

Modul Katalog – modul obsahuje OPAC pro nezaškolené uživatele a také profesionální vyhledávání pomocí Booleovy algebry a pravostranného a levostranného rozšíření. Vyhledávání je možné provádět i ve všech typech dokumentů současně. Vytvořené dotazy lze uchovávat pro příští použití a lze využít systém nadřazenosti a podřazenosti pro definování rešeršních dotazů. [6]

Výpůjční modul– umožňuje rychlou výpůjčku, vracení, prolongace výpůjček i jejich rezervace a automatické generování urgencí. Lze použít také v případě, že knihovna nemá zpracovaný celý fond v katalogu. Pracuje se s čárovým kódem, ale do modulu lze vstupovat i prostřednictvím klávesnice. Součástí modulu je i výpočet poplatků a pokut. [6]

Modul WWW-OPAC – umožňuje vyhledávání a zobrazení dat ve formě uživatelsky definovaného katalogizačního záznamu v síti Internet. [6]

Modul Nové knihy - tento modul umožňuje import dat z aktuální databáze Nové knihy do externího katalogu v KP-sys. [6]

Modul PSH a MeSH - moduly umožňují využívání známých tezaurů a heslářů pro on-line validaci deskriptorů a předmětových hesel při katalogizaci. [6]

8.1.4. SMARTLIB

Firma LUMARE spol.s.r.o. se již od svého založení věnovala vývoji programových prostředků pro obor knihovnictví. Výsledkem je komplexní knihovnický systém SMARTLIB, který je vhodný nejen pro

veřejné knihovny, ale také pro knihovny vědecké, odborné, univerzitní, školní a administrativní. Ve své komplexnosti je tento systém schopen zajistit automatizaci všech knihovnických procesů. Je možné využívat také moduly pro periodika, tisk čárových kódů, evidenci mezinárodní výměny časopisů o firmách a školách. [24]

SMARTLIB uživateli umožňuje nastavit celý systém tak, aby bylo možné pracovat s velkým objemem dat, protože ve většině knihoven nejsou stovky tisíc záznamů ničím neobvyklým. Náročnější uživatelé mohou využít program pod operačním systémem UNIX. [24]

Při aktivaci systému se v menu uživateli nabídne několik funkcí, z nichž si vybere alespoň ty nejpodstatnější. Jedná se především o nastavení definice dalšího modulu, kde lze vložit do menu nejen moduly základní, ale také doplňkové. Důležitým krokem je také nestavení definic tiskových výstupů, především pro tisk katalogizačních lístků, přírůstkových seznamů atd. [24]

8.1.4.1. Moduly knihovnického systému SMARTLIB

Modul Akvizice – zahrnuje všechny činnosti, které jsou v knihovnách důležité. Jedná se o práci s dezideráty, automatickou tvorbou objednávek apod. Jednotlivé údaje se vkládají pouze jednou a postupně se dále přenášejí do katalogizace. Dají se kdykoliv opravit. [24]

Modul Katalogizace – je základním modulem celého systému. Skládá se z jednotlivých konstantních databázi a z databází, které si může uživatel systému nastavit při jeho aktivaci. [24] Záznamy o dokumentech se skládají do polí a podpolí, která již odpovídají novým katalogizačním směrům, jež udává Národní knihovna v Praze.

Výpůjční modul – navazuje na modul Katalog a umožňuje pohyb jednotlivých dokumentů z fondu k uživateli a naopak. Automaticky se tiskne čtenářská legitimace, upomínky a rezervace, prolongace, doklady o výpůjčkách. S daným modulem je možné pracovat bez vazby na modul Katalogizace [MOTLOCHOVÁ, 1998].

8.1.4.2. Podpůrné a doplňkové moduly

On line katalog – pracuje jako nezávislý programový modul, samozřejmě využívající dat, které jsou vytvořeny ve SMARTLIBU.

On line katalog – Dětská beletrie – speciální verze katalogu pro dětské čtenáře, využívající grafických ikon a umožňující první setkání s počítačem v knihovně. [24]

SMARTLIB pro seriály – speciální modul pro evidenci jednotlivých čísel časopisů, doplněných samostatným výpůjčním procesem a řadou statistik a tiskových výstupů. Lze jim řešit i meziknihovní výpůjční služby časopisů. [24]

Mezinárodní výměna časopisů – speciální modul, který řeší evidenci vydaných a přijatých dokumentů v rámci mezinárodní výměny časopisů a příslušnou skladovou agendu.

8.1.5. DAWINCI

Knihovnický systém DAWINCI je společným produktem týmu elitních programátorů společnosti ASP a.s. a skupiny vysoce fundovaných a zkušených knihovníků Regionální knihovny Karviná. [24]

Systém byl vytvořen s použitím nejmodernějších objektových-komponentních programátorských technologií, díky kterým přestavuje DAWINCI robustní informační systém v „multi-tier“ architektuře. Tato architektura umožňuje univerzální šíření systému přes Internetové nebo

Intranetové technologie. Samozřejmostí systému je také plná podpora a kompatibilita Z39.50 protokolu, který zajišťuje univerzální celosvětovou dostupnost systému. [24]

Systém DAWINCI je prvním knihovnickým systémem, který pro správu dat a přístupu k nim, využívá kombinaci SQL databázových systémů a nejsilnějšího Full-text systému Topic. Tato kombinace s sebou přináší zcela nové a neomezené možnosti při vyhledávání a získávání informací ze systému. Standardní knihovní záznamy ve formátu UNIMARC je také možné vázat s informacemi jakékoliv struktury. [PEJŠOVÁ, 2004]

Systém DAWINCI obsahuje všechny standardní moduly knihovnicko-informačního systému. Součástí systému je také Administrátorský modul, pro vlastní univerzální nastavení systému a jeho následnou dokonalou správu [MOŠKOŘOVÁ, 1999].

8.1.5.1. Moduly knihovnického systému DAWINCI

Modul Administrátor – obsahuje všechny funkce pro univerzální nastavení a údržbu systému. Je přístupný jenom definovaným osobám (správce systému, odpovědní metodičtí pracovníci knihovny apod.). V rámci modulu je vloženo nastavení přístupu k datům systému, všeobecné provozní parametry, nastavení fondů, tématických skupin a rovněž nastavení generátorů identifikátorů. [24]

Modul Katalog – umožňuje vkládání fondů a tématických skupin pro libovolný počet záznamů. Volnou tvorbu vstupních formulářů, generování a úpravy přírůstků, tvorbu a kontrolu souboru autorit, využívání databází dodavatelů, tvorbu, sledování a tisk objednávek. Dále revizi stavu fondu, využití generátorů identifikátorů a modelového

Thezauru, import/export záznamů a dat, připojení mediálních informací k záznamům [PEJŠOVÁ, 2004].

Modul OPAC – je využíván ve full-textovém vyhledávání podle jednoduchého slova nebo složeného, volně tvořeného výrazu. V rámci tohoto modulu je možné vyhledávat podle autorit, hledat na základě výběru slova, tématu za stromu tématických hesel (Thezauru). Samozřejmě součásti OPAC je přehled o výpůjčkách titulů a zobrazování a tisk informací o nalezených titulech ve volitelném formátu. [24]

Modul Výpůjční systém – přináší prostředky pro tvorbu a manipulaci se záznamy čtenářů a uživatelů, zabezpečuje vlastní výpůjční proces, zobrazuje záznamy čtenářů podle zadaných kritérii a tiskne uživatelsky definované tiskové sestavy a statistické přehledy. Výpůjční modul zahrnuje také systém pro MVS a systém analýzy a zobrazování speciálních služeb, událostí a poplatků [PEJŠOVÁ, 2004].

8.2. Zahraniční automatizované knihovnické systémy

8.2.1. BIBIS

Knihovnický informační systém byl poprvé v České republice představen v prosinci roku 1994. Jedná se o komplexní modulární plně integrovaný knihovnický informační systém, který vyvinula holandská firma SQUARE B.V. Pro menší knihovny je výhodou, protože lze jednotlivé moduly zakoupit samostatně a postupně doplňovat kategorie Ibisu s přibývajícími tituly. V praxi se většinou instaluje na hardware, který již knihovna vlastní. Systém pracuje na veškerých platformách standardních operačních systémů a obrovskou výhodou je možnost přechodu mezi jednotlivými operačními systémy. V takovém případě se změní pouze softwarový toolkit a BIBIS i data zůstávají stejná [LHOTÁK, 2000].

Systém pracuje v reálném čase a umožňuje současný přístup více uživatelů do databáze. Umožňuje import a export dat a snadné napojení na databáze jiných knihoven. Systém podporuje multimediální aplikace a k jednotlivým záznamům lze připojit nascanovaný dokument, zvukový záznam nebo videosekci. Jedná se o multijazyčný systém, který existuje v anglické, německé, holandské, portugalské, české a francouzské jazykové verzi. I během práce se systémem, lze přepínat mezi jednotlivými jazyky [LHOTÁK, 2000].

Knihovna má možnost systém nakonfigurovat podle své potřeby a měnících se požadavků, které vznikají v souvislosti s automatizací, více než 150 konfiguračními variantami.

8.2.1.1. Moduly knihovnického systému BIBIS

Modul Katalog – je nejdůležitějším a největším modulem. Umožňuje pracovat nejen s velkým počtem standardních polí, která lze aktivovat pro jednotlivé pole dokumentů, navíc umožňuje velice snadno nadefinovat nová. Vyhledávat v katalogu lze podle základních polí, podle polí nadefinovaných uživatelem, jednotlivých slov ve všech záznamech, jednotlivých slov v určitých polích, klíčových slov, pomocí Booleovských operátorů. Výhodné je spojení modulu katalogu se systémem plnotextového vyhledávání EFS Excalibur. Značná pozornost je věnována také zpřístupnění katalogu veřejnosti a snadnost ovládání. Díky tomu mohou čtenáři v katalogu snadno vyhledávat a záleží pouze na knihovně, které informace z katalogu svým uživatelům zpřístupní. Z uživatelského katalogu je také přístup k multimediálním záznamům a lze použít i hlasovou navigaci, např. v dětských odděleních [LHOTÁK, 2000].

Modul Objednávky – zde jsou řešeny veškeré činnosti spojené s objednáváním nových přírůstků do knihovny. Aktuální objednávky

mohou být tištěny ve formě, kterou vyžaduje dodavatel. Systém po vystavení objednávky sám kontroluje, zda dodávka probíhá podle předpokladu. Pokud tomu není tak, postará se submodel urgencí o urgenční dopisy. Tento modul řeší také MVS a MMVS, včetně statistik [LHOTÁK, 2000].

Modul Evidence přírůstků – stará se nejen o zpracování nových akvizic, ale také nabízí možnost tisku seznamu přírůstků v požadovaném členění a formě. Třídění je volitelné a na tento modul navazují tiskové formuláře [LHOTÁK, 2000].

Modul Výpůjční protokol – zajišťuje jednoduchý systém při půjčování, vracení, rezervaci, prolongaci a pokutách. Systémy výpůjček jsou rozdílné pro jednotlivé typy knihoven a informačních institucí [LHOTÁK, 2000].

Modul Periodika – řeší komplexní evidenci a cirkulaci periodik. Automaticky hlídá přecházející periodika do knihovny. Cirkulace je řešena kombinací metod hvězdy a kruhu [LHOTÁK, 2000].

Modul Správa systému – tento modul se nezakupuje samostatně, ale je součástí každé instalace. Jsou v něm uloženy informace o vylisovaných tabulkách, konfiguraci systému a bezpečnostních pravidlech [LHOTÁK, 2000].

8.2.2. PC-LIB

PC-LIB je slovenský systém určený pro střední a větší knihovny, využitelný v knihovnické praxi pro výpůjční protokol, sestavování bibliografií, rešerší, nejrůznějších zajímavých přehledů, statistik, při objednávkách nových titulů, kontrole dodávek apod. [24] Jedná se tedy o systém, který je schopný zapamatovat si a později poskytnout logické

vazby mezi důležitými atributy. Systém byl vytvořen jako integrovaný, modulární a flexibilní systém, přičemž jeho jednotlivé moduly jsou přímo dostupné z hlavního uživatelského menu, bez nutnosti opustit základní prostředí. Systém umožňuje jednoduché postupné rozšiřování a propojování modulů, podle přání zákazníka.

Uživatelské prostředí je velmi příjemné a přístup k požadovaným informacím je jednoduchý a rychlý, realizovaný formou nabídek s možností vyhledávání. Prostředí je vybaveno také kontextovou nápovědou. [24]

PC-LIB je navržený a realizovaný formou, která umožňuje evidovat všechny typy dokumentů, včetně trojrozměrných dokumentů, speciálních tiskovin, výzkumných zpráv, firemní literatury atd. podle zásad a pravidel pro katalogizaci až po úroveň analytického popisu. Systém také nabízí využití čárového kódu pro evidenci uživatelů a knihovních jednotek, což výrazně zpřesňuje a zrychluje evidenci, zjednodušuje práci knihovníků při katalogizaci, výpůjčním procesu a revizích knihovního fondu. [24]

Kromě udržování elektronických katalogů umožňuje systém tisknout katalogizační lístky s možností jejich editace. Systém tak disponuje různými přehledy a revizemi fondu a následným výstupem na tiskárnu. U každého modulu uživatel také volí formy výstupů na obrazovku, tiskárnu nebo do souboru. [24]

8.2.2.1. Moduly knihovnického systému PC-LIB

Modul Katalogizace – umožňuje vkládat do databáze nové údaje, opravovat již existující záznamy a postupně doplňovat záznamy k jinému titulu. Poskytuje všechny možnosti vyhledávání a při zadávání jiného údaje poskytuje jeho výběr podle jiného údaje, jehož hodnotu zná z již

existující databáze. Výsledky katalogizace se přenášejí do modulu definování. [24]

Modul Akvizice – obsahuje tvorbu objednávek, dodávek, pomáhá vytvářet vlastní databázi kontaktů na vydavatele a distributory atd. Modul umožňuje aktualizaci knihovního fondu, vyřizování či přeřazování, export a import údajů ve formátu ISO 2709. Pomocí uvedeného modulu je možné komunikovat s více pobočkami knihovny, pokud nejsou soustředěny v jedné budově. [24]

Modul Výpůjčky-rezervace-upomínky-urgence – zahrnuje veškeré možnosti výpůjčního protokolu. Od registrace čtenáře či uživatele, přes výpůjční protokol, rezervace, vracení, upomínky, prolongace atd. V rámci uvedeného modulu je také agenda spravující oblast MVS s využitím jejího propojení na Internet. [24]

Modul Rešerše a bibliografie – evidence vytvořených rešerší podle čísla a názvu. Modul také eviduje, člení a umožňuje třídit vytvořené bibliografie. [24]

Modul Výročí osobnosti – velký pomocný modul, který buduje a vyhodnocuje výročí autorů a osobností. Vychází z údajů na definování. V této části systému jsou běžně sestavována různá kalendária, slovníky, personální bibliografie a rešerše. [24]

Modul Metodika – zahrnuje evidenci jednotlivých organizačních jednotek, zaměstnanců, evidenci obecných knihovních záležitostí, s možností využití v rámci akvizice a výpůjčního protokolu. Součástí jsou statistické přehledy a výkazy za určité období.[24]

Modul Přehledy – evidence s možností statistických výkazů a přehledů, které jsou využitelné při práci v knihovnách. [24]

Modul Revize – pomáhá při revizích knihovního fondu a s evidencí a fyzickým zpracováním. Umožňuje následné vyhodnocení a sestavení výkazu o provedené revizi. [24]

8.2.3. T-series (Tinlib)

Automatizovaný knihovnický systém T-series je vyzrálým knihovnickým systémem třetí generace, jehož základem je objektově orientovaný relační databázový řídící systém. Tento systém byl vytvořen speciálně pro potřeby knihovnických a informačních aplikací. Tato struktura vytvořená hypertextovými technikami, umožňuje uživateli navigaci mezi jednotlivými entitami [TUŽOVÁ, 2004].

Celá aplikace je psaná v 4GL (programovací jazyk 4.generace) a představuje jednu z nejperspektivnějších komerčních aplikací hypertextu. T-series je určen pro systémy Open Systems Architecture, procuje pod operačním systémem MS-DOS a pod UNIX. Přechod od prostředí MS-DOS k prostředí UNIX je snadný a bezproblémový. [24]

V systému je velká péče věnována bezpečnosti uložených dat, úroveň přístupu a práva přístupu k bázi dat jsou hierarchicky seřazena a zabezpečena přidělenými identifikátory a hesly. Všechna pole záznamů jsou opakovatelná a dovolují proměnnou délku pole [TUŽOVÁ, 2004].

T-series je charakterizován modulární architekturou, integrovaným prostředím a mimořádně vlídným uživatelským rozhraním. Modularita systému umožňuje knihovnám doplnit zakoupené sestavy dalšími moduly, které byly nově vyvinuty, popř. doplnit ty moduly, které knihovny na počátku nezakoupily [TUŽOVÁ, 2004].

T-series je považován za světově nejkomplexnější knihovnický a rešeršní systém a skládá se z níže popsaných modulů.

8.2.3.1. Moduly knihovnického systému T-series

Modul Katalogizace a vyhledávání – data mohou být vkládána do libovolného katalogizačního formátu systému T-series. Důmyslné řešení editace v T-series umožňuje vytváření pružné struktury záznamu s poli libovolně dlouhými a libovolně opakovatelnými. Jediné omezení ve skutečnosti vyplývá z typu použitého hardwaru. Pomocí inteligentních oken, mohou knihovníci položky vybrat v otevřených oknech a překopírovat. Je možné v oknech plně využít všech vyhledávacích možností a dokonce je možné uvnitř oken editovat. Tezaurus systému, začleněný v modulu katalog, poskytuje možnost ukládat nadřazené výrazy, podřazené výrazy, příbuzné výrazy apod. Tedy všechno, co knihovníci potřebují k tomu, aby mohli vytvořit plně hierarchický tezaurus [TUŽOVÁ, 2004].

Modul Výpůjční protokol – je dokonale propojen s ostatními moduly T-series. Zahrnuje všechny funkce a kroky potřebné při sledování jednotlivých dokumentů a při jejich půjčování. Modul výpůjční protokol umožňuje knihovnám nastavit vlastní parametry, pomocí nichž mohou knihovníci lépe klasifikovat své dokumenty a čtenáře. V tomto modulu je možné nastavit texty pro sdělení potřebná při rozesílání upomínek a při vyřizování rezervací. Vypůjčení a vracení dokumentů je možné provádět z jakékoli pozice v systému, několika různými způsoby. Upomínky a různá vyrozumění mohou být automaticky generovány a tištěny jednotlivě nebo v dávkách. Tento modul také obsahuje možnost vytvářet zprávy a statistiky týkající se chodu knihovny [TUŽOVÁ, 2004].

Modul Akvizice – zahrnuje všechny kroky potřebné při objednávání, urgencích a přijímání monografií [TUŽOVÁ, 2004]. Administrativní podrobnosti, např. nákladová střediska, osobní záznamy,

kódy a adresy dodavatelů je třeba vložit jenom jednou. Bibliografické údaje se okamžitě zobrazují v katalogu s jasně označeným statutem – objednáno. Modul Akvizice umožňuje kompletní vedení účetnictví.

Modul Správa seriálů – základní funkcí správy seriálů je jejich denní vstupní kontrola seriálů. V systému T-series se provádí pomocí jediné funkční klávesy, a to i v případě více kopií. Při vstupní kontrole se automaticky tisknou kopie příslušných cirkulářů. Tento modul dává možnost také objednávat, fakturovat, rušit, urgovat a vracet seriály a tvořit různé zprávy a statistiky [TUŽKOVÁ, 2004].

Modul Formátování a přenosu dat – tento modul, plně propojený s modulem Katalog, umožňuje konverzi formátu záznamů. Používá se při importu záznamů do databáze T-series a při exportu záznamů ze systému do jiného formátu. Je vysoce adaptabilní a dovoluje efektivní a rychlé začlenění požadavků pro jednotlivá lokální data. Dokáže pracovat se strukturou formátů s pevnou délkou, proměnou délkou, s tagy i se speciálními strukturami formátů, včetně výměnného formátu ISO 2709. [24] [TUŽKOVÁ, 2004]

Modul TINGEN, Generátor zpráv – umožňuje třídit data T-series, formátovat je a řadit do tabulky. Typickým využitím tohoto modulu je tvorba přírůstkových seznamů nebo tvorba katalogizačních lístků, přičemž lze použít všechny relevantní informace uložené v systému. [24]

8.2.4. ALEPH

ALEPH (Automated Library Expandable Program) u nás distribuuje firma Ex-Libris. Systém byl vyvinut programátory, analytiky a knihovníky Hebrejské univerzity v Jeruzalémě. [24]

ALEPH je navržen tak, aby jej mohly používat kromě knihoven také archívy, muzea a informační centra. Jedná se o modulární systém, což znamená, že se každá z institucí může rozhodnout, který modul si zakoupí a který pro svůj provoz nepotřebuje. Systém ALEPH je vícejazyčný [PŘÍBRAMSKÁ, 2001].

8.2.4.1. Moduly knihovnického systému ALEPH

Modul Katalog - Klíčovými prvky katalogizace v systému Aleph 500 jsou upravovatelné šablony, formuláře, validační pravidla a zobrazení spolu s účinnou správou autorit. Aleph současně podporuje MARCovské a neMARCovské záznamy v jediné databázi pro vyhledávání, katalogizaci a zpracování. Je kompatibilní s více formáty MARC - podporuje USMARC, UNIMARC, UKMARC, MARC21 a DANMARC a německý katalogizační formát MAB. NeMARCovské záznamy mohou být v jakémkoli formátu definovaném knihovnou pro pokrytí potřeb různých typů dokumentů a sbírek. [24] [PŘÍBRAMSKÁ, 2001]

Modul OPAC - OPAC systému Aleph je všem typům uživatelů přístupný přes standardní webový prohlížeč s využitím všech výhod, které WWW nabízí. Knihovny využívající Aleph tak mohou těžit z výhod otevřené architektury, která umožňuje úplnou kontrolu nad vzhledem a formátem obrazovek, vyhledávacích nabídek a hlášek, přístup k dalším zdrojům, kontextovou nápovědu apod. [PŘÍBRAMSKÁ, 2001].

Modul Výpůjční protokol - Prostředí výpůjčního procesu v Alephu umožňuje flexibilně nastavovat výpůjční politiku pomocí velkého množství upravovatelných parametrů. Aleph umožňuje definovat délky výpůjček podle statusu čtenáře, statusu položky, lokace a kalendáře. Provádí okamžité kontroly rezervací a výpůjček čtenářů k zajištění správné cirkulace dokumentů. Rychlé zpracování je rozšířeno

o integrovaný platební systém, který umožňuje zavádění pokut, poplatků a dalších knihovnou definovaných transakcí. Dovoluje také pracovníkům přijímat poplatky a pokuty nebo je naopak zamítat. Systém také generuje bohaté statistické přehledy a zprávy, včetně podrobných auditů mapujících všechny platební transakce [PŘÍBRAMSKÁ, 2001]. Čtenáři ocení sofistikovanou MVS a funkce pro doručování dokumentů, které nabízejí plynulou výměnu informací mezi knihovnami a dodavateli dokumentů po celém světě. Aleph je schopen vyřizovat velké množství transakcí současně a aktualizovat databází on-line v reálném čase.

Modul Akvizice - V Alephu ocení akviziční pracovníci značnou pružnost v řízení všech aspektů objednávacích a dodacích procesů, zahrnujících objednávky všech typů dokumentů, vstupní kontrolu a fakturování dokumentů a aktivity týkající se urgencí, rozpočtů a dodavatelů. Rozvržení akvizičního prostředí Alephu pomůže spravovat složité finanční hierarchie, údaje o dodavatelích, objednávkách, fakturách a urgencích s jednoduchým napojením na daný účetní systém [PŘÍBRAMSKÁ, 2001].

Modul seriály - je zcela integrován s akvizicí, katalogizací a OPACem. Seriály v Alephu nabízí knihovnímu personálu bezproblémový přístup k bibliografickým a holdingovým údajům a informacím o objednávkách, finančním údajům, informacím o dodavatelích atd. Knihovníci se mohou k údajům o seriálech dostat pomocí různých knihovnou definovaných rejstříků, včetně SICI, ISBN a klíčových slov. Aleph plně podporuje standard MARC pro harmonogram očekávané periodicity seriálů a poskytuje okamžité kontroly pro všechny typy publikací včetně těch nejméně obvyklých. Sdílená databáze harmonogramů periodicity usnadňuje import, export a výměnu dat. Hned po nastavení může být očekávaná periodicita výtisků

jednoduše upravena pro zřízení zvláštních harmonogramů pro číslo časopisu, které se liší od běžných, ale přesto je třeba vytvořit jeho záznam nebo se ujistit, že systém nebude urgovat číslo, které vydavatel nebude publikovat. Urgence mohou být generovány automaticky podle parametrů definovaných knihovnou nebo mohou být vytvořeny ručně. Prostředí seriálů v Alephu pracuje se zabudovaným systémem sledování titulů, které jsou v oběhu. Zpracování vazeb a kapacit oběhu je plně integrováno s řízením výpůjček, seriálů a účetnictvím. [24]

Modul Konsorcia - Aleph nabízí knihovnám účinné a elegantní konsorciální řešení. Protože se stále více knihoven stává členy místních, regionálních a národních konsorcií, společnost Ex-Libris vyhověla požadavku a nabízí několik konsorciálních modelů podporujících složité konsorciální kombinace a umožňuje flexibilně konfigurovat databáze. Přitom klade značný důraz na samostatnost jednotlivých začleněných knihoven. Flexibilní architektura Alephu může být konfigurována tak, aby vyhověla politice v zpřístupňování zdrojů a katalogizační politice daného konsorcia, nezávisle na složitosti. V průběhu minulých dvaceti let Ex-Libris spolupracoval s různými velkými světovými konsorcii - jedno však bylo pro konsorcia vždy stejné - každá knihovna je odlišná! To podnítilo vznik širokého spektra konsorciálních řešení zahrnujících souborné katalogy „just in case“ a „just in time“, klasické centralizované katalogy, virtuální katalogy atd. Všechny tyto modely jsou současně implementovány v instalacích Alephu po celém světě. [24]

9. Protokol Z39.50

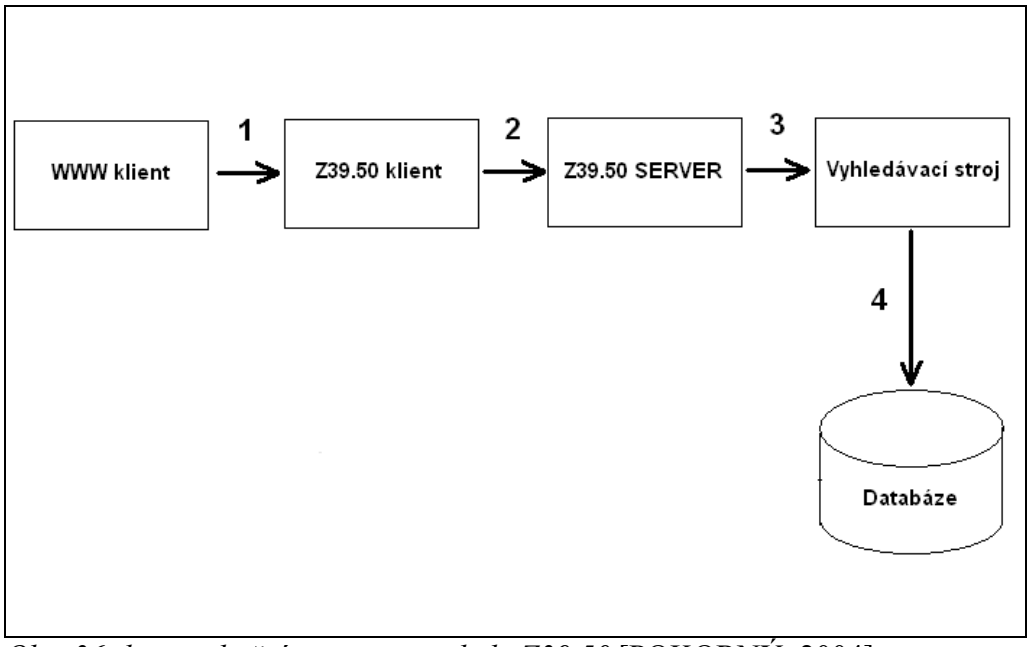
V předešlé kapitole byl prezentován seznam hlavních knihovnických systémů, které jsou založeny na relačním databázovém modelu. V této kapitole bude zmíněn komunikační protokol Z39.50, který představuje sjednocující prvkem pro sdílení informací mezi knihovnickými systémy.

Tento protokol je standardizovaný pro vyhledávání a přejímání dat. Moduly, které zpracovávají požadavky přijaté na bázi protokolu Z39.50 jsou v moderní době souputníky relačních databázových systémů, resp. knihovních systémů. Z39.50 specifikuje abstraktní informační systém s velkou množinou funkcí pro vyhledávání a přejímání záznamů, procházení uspořádaných seznamů, atd. Na straně serveru je tento abstraktní systém mapován na konkrétní systém správy bází dat. Komunikace mezi serverem a klientem je přesně definována protokolem. Implementační detaily serveru jsou zakryté síťovým rozhraním, takže klient může přistupovat k libovolnému typu databáze prostřednictvím stále stejného protokolu. Na straně klienta je abstraktní informační systém zpětně namapován na uživatelské rozhraní, které může být šité na míru konkrétním požadavkům uživatele [RUBRINGER, 1999]. Výhodou Z39.50 tedy je, že umožňuje odlišným informačním zdrojům vystupovat pod jednotným uživatelským rozhraním a současně dává každému informačnímu systému možnost vytvořit několik rozhraní pro odlišné skupiny uživatelů.

Obsah dalších kapitol nebude popisovat vznik a vývoj tohoto standardizovaného aplikačního protokolu, ale spíše se zaměříme na konkrétní procesní postup mezi požadavkem klienta a reakcí databázového systému.

9.1. Jednotlivé komunikační kroky

Na obr. 36 je znázorněno schéma, jenž odpovídá procesu kladení dotazu směrem od uživatele na databázi. Jednotlivé datové procesy jsou označeny od 1. do 4.



Obr. 36: komunikační proces protokolu Z39.50 [POKORNÝ, 2004]

9.1.1. Klient naváže spojení

Spojení navazuje klient zasláním tzv. **Init** požadavku, kterým se představí serveru a ve kterém navrhuje hodnoty parametrů (např. verze protokolu, ověření identity, podporované operace, maximální délka zprávy) nezbytných pro vytvoření Z39.50 relace, tj. navázání interaktivní komunikace mezi konkrétním Z39.50 klientem a jiným konkrétním Z39.50 serverem.

9.1.2. Server Z39.50 odpoví

Server zareaguje zasláním **Init** odpovědi, ve které oznamuje klientovi své hodnoty parametrů a informuje ho, zda souhlasí s vytvořením Z39.50 relace.

9.1.3. Klient formuluje a odešle dotaz

Následně klient zformuluje dotaz a odešle jej jako tzv. **Search** požadavek.

9.1.4. Server hledá v databázích a zašle odpověď

Server prohledá databáze, vytvoří si množinu výsledných záznamů a odpovídá zasláním počtu prvků této množiny v **Search** odpovědi. V závislosti na počtu nalezených záznamů může tato odpověď obsahovat také některé, či všechny nalezené databázové záznamy. Ostatní nalezené záznamy jsou pro klienta k dispozici prostřednictvím dodatečných dotazů, tzv. **Present** požadavků, po kterých následují **Present** odpovědi serveru, obsahující požadované záznamy [POKORNÝ, 2004].

9.1.5. Ukončení spojení

Proces ukončení Z39.50 relace může zahájit jak klient, tak server, a to zasláním **Close** požadavku a obdržením **Close** odpovědi.

9.1.6. Jiné operace

Kromě výše uvedených služeb Init, Search a Present nabízí protokol Z39.50 mnoho dalších operací, jako např. vyhledávání v uspořádaném seznamu (služba **Scan**), setřídění či zrušení množiny výsledků (služba **Sort**, resp.**Delete**), ověřování totožnosti klienta (služba **Access-control**), zasílání služebních informací (služba **Resource-control**), atd. Databázové záznamy musí vyhovovat některému z registrovaných formátů, přičemž 2. verze protokolu Z39.50 z roku 1995 podporuje patnáct především MARC-formátů (UNIMARC, USMARC, UKMARC, NORMARC, apod.) [POKORNÝ, 2004].

9.2. Popis a nastavení atributů protokolu Z39.50

Nastavení atributů Z39.50 v tomto profilu vychází ze standardu ANSI/NISO Z39.50-1995 a platí pro podmnožinu Bib-1, tedy pro bibliografické záznamy. Definuje pouze atributy, které se prakticky využívají při vzdáleném vyhledávání. Doplněno je základní mapování na pole UNIMARC a MARC21 – využijí se ta pole, která jsou v dané databázi k dispozici [POKORNÝ, 2004]. Pro příklad použijme uživatelův dotaz na autora Stephena Hawkinga. Předvedeme si transformaci dotazu v jednotlivých datových procesech, které jsou na obr. 36 označeny 1 až 4.

Profil určuje nastavení následujících atributů:

USE	1
RELATION	2
POSITION	3
STRUCTURE	4
TRUNCATION	5
COMPLETENESS	6

Pro bibliografické údaje nechť uživatel použije zadaní alespoň do jednoho z následujících polí:

AUTOR
NÁZEV
PŘEDMĚT
ROK
ISBN
ISSN

Ukažme si algebraické transformace, jenž jsou součástí převodového můstku klienta Z39.50.

Nastavení hodnot pro pole **AUTOR**:

1=1003	autor
2=3	rovno
3=1,3	první v poli, kdekoli v poli
4=1,2	fráze, slovo
5=1,100	pravé, žádné
6=1	nekompletní

[POKORNÝ, 2004]

Pozn.: Hodnota 1=1003 představuje autora obecně, obsahuje v sobě tedy souhrn hodnot 1=1004, 1005 a 1006.

Základní mapování na UNIMARC: pole 70X$* a 71X$*.

Základní mapování na MARC21: pole 100$*, 110$* a 111$*, 700$*, 710$* a 711$*.

Nastavení hodnot pro pole **NÁZEV**:

1=4	název
2=3	rovno
3=1,3	první v poli, kdekoli v poli
4=1,2	fráze, slovo
5=1,100	pravé, žádné
6=1	nekompletní

[POKORNÝ, 2004]

Základní mapování na UNIMARC: pole 200$aei, 500$*, 510-520$*, 530$*, 532$*, 540$*, 541$*, 545$*.

Základní mapování na MARC21: pole 245$abp, 130$*, 730$*, 222$*, 240$*, 242$*, 246$*, 247$*.

Nastavení hodnot pro pole **ROK**:

1=30,31 rok, rok vydání
2=3 rovno
3=1,3 první v poli, kdekoli v poli
4=2 slovo
5=1,100 pravé, žádné
6=1 nekompletní
[POKORNÝ, 2004]

Základní mapování na UNIMARC: pole 210$d (číselný údaj).

Základní mapování na MARC21: pole 260$c (číselný údaj).

Nastavení hodnot pro pole **PŘEDMĚT**:
1=21 předmět
2=3 rovno
3=1,3 první v poli, kdekoli v poli
4=1,2 fráze, slovo
5=1,100 pravé, žádné
6=1 nekompletní
[POKORNÝ, 2004]

Základní mapování na UNIMARC: pole 600$*, 601$*, 602$*, 604$*, 605$*, 606$*,607$*, 608$*,610$*,615$*,626$*, předmětová pole z 9XX$*.

Základní mapování na MARC21: pole 072$*, 600$*, 610$*, 611$*, 630$*, 650$*, 651$*,653$* a 655$*, předmětová pole z 69X$*.

Nastavení hodnot pro pole **ISBN**:
1=7 ISBN
2=3 rovno
3=1,3 první v poli, kdekoli v poli

4=2	slovo
5=1,100	pravé, žádné
6=1	nekompletní

[POKORNÝ, 2004]

Základní mapování na UNIMARC: pole 010$az.

Základní mapování na MARC21: pole 020$az.

Nastavení hodnot pro pole **ISSN**:

1=8	ISSN
2=3	rovno
3=1,3	první v poli, kdekoli v poli
4=2	slovo
5=1,100	pravé, žádné
6=1	nekompletní

[POKORNÝ, 2004]

Základní mapování na UNIMARC: pole 011$ayz.

Základní mapování na MARC21: pole 022$ayz.

Nyní se tedy dostáváme zpět k našemu požadavku na vyhledání díla. Uživatel zadá do pole AUTOR Stephen, Hawking a v datovém procesu č. 1 dojde k první transformaci dotazu na **FIND 100$a = "Hawking, Stephen"**.

Tento dotaz (proces č. 1) převezme klient Z39.50 a podle nastavených hodnot pole **AUTOR** přetransformuje dotaz pro Z39.50 server do podoby **FIND 1=1003 2=3 3=1 4=1 5=100 6=1 "Hawking, Stephen"**. To v našem případě reprezentuje datový tok č. 2

SERVER Z39.50 převezme tuto syntaxi a přetransformuje dotaz do podoby:

```
SELECT COUNT(id)
FROM knihy
WHERE autor =“Hawking, Stephen”
```

Tady vstupuje do procesu vyhledávací stroj, který tento požadavek resp. datový tok č. 3 převezme a podle jazyka SQL dojde k vyhledání dotazu v repozitáři databáze.

9.2.1. Formáty

Zdroje musí vracet záznamy v některém z následujících formátů:

- UNIMARC
- MARC21
- MARC mapovaný na XML (DTD podle LC, http://www.loc.gov /marc/marcxml.html#marcdtd).

9.2.2. Znaková sada

Zdroje musí komunikovat v některé z následujících znakových sad, pakliže se pohybujeme v infrastruktuře knihovních systémů v České republice:

- ISO-8859-2
- WINDOWS 1250
- UTF-8

9.2.3. Kombinování podmínek dotazu

Zdroj musí podporovat kombinace minimálně 2 výrazů s použitím operátorů AND, OR, NOT.

9.2.4. Řazení

Zdroj musí být schopen řadit záznamy jednotlivě podle autora, názvu a roku. Kombinace řazení nejsou vyžadovány.

9.2.5. Vyhledávání podle ISBN a ISSN

Z důvodů rozdílných zápisů ISBN a ISSN v různých zdrojích je nutné sjednotit možnost vyhledávání pomocí úpravy indexů [POKORNÝ, 2004].

Pole ISBN: indexovat bez mezer a pomlček, ponechat pouze alfanumerické znaky, např. 8071854840 (nikoli jako 80-7185-484-0 nebo 80 7185 484 0).

Pole ISSN: indexovat bez mezer a pomlček, ponechat pouze alfanumerické znaky, např. 12118214 (nikoli jako 1211-8214 nebo 1211 8214).

9.3. Shrnutí popisu protokolu Z39.50

Ačkoli je představa o využití protokolu Z39.50 jako sjednocujícího prvku při sdílení informací mezi různými knihovnickými systémy velmi revoluční, v praxi je třeba zůstat na zemi. Aplikace, využívající protokol Z39.50, mohou poskytovat jednotné rozhraní jen tehdy, budou-li tento protokol beze zbytku podporovat. Z39.50 je však protokolem velice rozsáhlým, takže ve skutečnosti každý systém založený na protokolu Z39.50 využívá spíše jen větší či menší podmnožinu jeho vlastností. Rozhraní těchto systémů jsou tedy sice založena na standardu protokolu Z39.50, avšak není zaručena jejich vzájemná kompatibilita. Při vytváření distribuovaných či virtuálních souborných katalogů je proto třeba, aby se zúčastněné strany dohodly na konkrétní množině podporovaných vlastností Z39.50, čili na tzv. profilu,

jehož součástí jsou např. typy atributů a jejich hodnoty, syntaxe záznamů, velikosti zpráv, apod. [KRČMÁŘOVÁ, 2002].

Bohužel se zatím protokol Z39.50 pohybuje ještě stále v neunifikovaném informativním prostředí a uživatel či správce databázového systému může brzy zjistit, že různé servery mají nejenom odlišné vlastnosti, ale jejich chování je dokonce nezřídka v rozporu s protokolem Z39.50. Takže je téměř nemožné vytvořit univerzálního Z39.50 klienta, který by dokázal komunikovat s libovolným Z39.50 serverem bez předchozí studie jeho chování. Nemluvě o dlouhých prodlevách spojených s nekvalitním internetovým připojením. Obecně se deklaruje, že vyřízení dotazu přes protokol Z39.50 by nemělo být delší než 60 sekund.

I přes tyto skutečnosti přináší protokol Z39.50 mnohá zjednodušení. Usnadňuje například přejímání autorit, protože záznam s sebou nese informaci o použitém formátu [KRČMÁŘOVÁ, 2002]. Velký přínos Z39.50 je však zejména v tom, že výrazně zjednodušuje implementaci souborných katalogů, protože rozdíly mezi chováním jednotlivých serverů nejsou zdaleka tak veliké, jaké jsou mezi systémy, které Z39.50 nepodporují.

10. Závěr

Cílem této práce bylo překlenout propast mezi akademickou teorií databází a skutečným vytvářením databází v reálném knihovnickém světě. Práce je zaměřena na výklad informací, které je třeba vzít v úvahu, pokud se rozhodneme pro návrh databází pro knihovní aplikace. Nevynecháno nebylo ani nezbytné vypracování analytické dokumentace k databázovým projektům. Stejně tak ani návrh případných grafických uživatelských prostředí. Případný programátor by rozhodně měl mít předlohu, podle které dokáže přenést pouhý návrh tvůrce do funkční podoby.

Při své práci jsem vycházel z případů, které se vyskytují v reálném provozu knihovny, resp. v knihovnickém systému, který v dnešní moderní době je stěžněm každého provozu knihovny. Proto je v práci zakomponována řada reálných příkladů, postavených na běžných implementacích jazyka SQL.

Když jsem se rozhodl pro napsání diplomové práce, mé znalosti v oboru relačních databází byly povrchní. Spojení příjemného s užitečným dalo vzniknout této diplomové práci. Když jsem poprvé usedal k psaní této práce, říkal jsem si, jaké že to vděčné téma jsem si vybral. Postupem času jsem začal vnímat, že dané téma obsahuje nejen množství informací a technik, které vyplývají z relační teorie, ale řadu informací, s jakými bych se nesetkal ani během několikaleté praktické zkušenosti s navrhováním databází. Za takové považuji např. zabezpečení toho, že hotový návrh systému skutečné řeší zadaný problém z reálného světa. Toto je rozhodně mnohem obtížnější a také důležitější než samotná specifika dané implementace. Abych se dokázal vypořádat s konkrétním návrhem relační databázové struktury pro konkrétní

aplikaci a nemusí se jednat jen o aplikaci knihovního systému, musel bych vycházet z bohatých zkušeností, které jsem načerpal z výstaveb kriticky důležitých databázových aplikací pro různé obory či zákazníky.

I přes mé nedokonalé znalosti v návrhu aplikací, které jsou založeny na relačním modelu databáze však mohu s klidným svědomím napsat, že jsem se snažil dodržet klíčové zásady, jenž by uspokojily potřeby uživatele knihovnického systému.

V práci se nehovoří o administraci, auditu a bezpečnosti systému. To však neznamená, že by měly být tyto aspekty návrhu opomíjeny. Ba právě naopak, v praxi se tyto požadavky lehce podceňují a odkládají na dobu, kdy už bývá příliš pozdě.

Seznam použité literatury

1. FARANA, Radim. *Databázové systémy* [online]. 2003 [cit. 2006-03-21]. Dostupný z WWW: <http://www.fs.vsb.cz/books/dbacc20/dbacc02.htm>.
2. *Fórum Databázového světa* [online]. 2007, 2.12.2007 [cit. 2006-05-05]. Dostupný z WWW: <http://forum.dbsvet.cz/>.
3. HERNANDEZ, Michael J. *Návrh databází*. Jan Bouda. 1. vyd. Praha: GRADA, 2006. 408 s. ISBN 80-247-0900-7.
4. KALUŽA, Radovan. *Teoretická východiska dotazovacích jazyků* [online]. 2004 [cit. 2006-04-12]. Dostupný z WWW: <http://www.dbsvet.cz/view.php?cisloclanku=2004093001>.
5. *Knihovní systémy Clavius a LANius* [online]. 2007 [cit. 2007-10-05]. Dostupný z WWW: <http://www.lanius.cz/>.
6. *KP-SYS - Úvodní stránka* [online]. 2006, 6.6.2006 [cit. 2007-11-11]. Dostupný z WWW: <http://www.kpsys.cz>.
7. KRČMAŘOVÁ, Gabriela. *Z39.50 - protokol z minulého století*. Ikaros [online]. 2002, roč. 6, č. 3 [cit. 2007-12-03]. Dostupný z WWW: <http://www.ikaros.cz/node/919>. ISSN 1212-5075.
8. LACKO, Luboslav. *Oracle* : *Správa programování a použití databázového systému*. 1. vyd. Brno : Computer Press, 2003. 464 s. ISBN 80-7226-699-3.
9. LACKO, Luboslav. *Oracle : Správa, programování a použití databázového systému*. 2. doplněné vyd. Brno : Computer Press, 2007. 573 s. ISBN 978-80-251-1490-2.
10. LHOTÁK, Martin. *BIBIS* [online]. 2000 [cit. 2007-11-11]. Dostupný z WWW: <http://www.lib.cas.cz/knav/bibis/bibis.htm>.
11. MÍKA, Jiří. *Vliv zavádění automatizovaných knihovnických systémů na organizaci a provoz knihovny* [online]. 1 Národní knihovna ČR, 2000 [cit.

2007-10-17]. Dostupný z WWW: <http://full.nkp.cz/nkkr/Nkkr0001/0001006.html>. ISSN 1214-0678.

12. MOŠKOŘOVÁ, Magda; SLANINOVÁ, Kateřina. *Automatizovaný knihovnický systém DAWINCI - systém knihoven informační epochy*. Ikaros [online]. 1999, roč. 3, č. 9 [cit. 2007-11-11]. Dostupný na World Wide Web: <http://www.ikaros.cz/node/1041>. URN-NBN:cz-ik1041. ISSN 1212-5075.
13. MOTLOCHOVÁ, Věra. *PROGRAM SMARTLIB V REGIONÁLNÍ KNIHOVNĚ V KARVINÉ* [online]. 1998 [cit. 2007-10-05]. Dostupný z WWW: <http://www.vkol.cz/obzory/961_05.htm>. ISSN 1214-6498.
14. OTRUBOVÁ, Alena. Osobní komunikace. Únor 2006.
15. PEJŠOVÁ, Petra. *Konverze dat do výměnného formátu UNIMARC*. Praha, 2004. 108 s. Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví. Vedoucí rigorózní práce PhDr. Anna Stöcklová.
16. POKORNÝ, Jan. *Profil pro jednoduché vyhledávání a stahování záznamů* [online]. Praha : UK, Ústav výpočetní techniky, 2004 [cit. 2007-11-10]. PDF. Dostupný z WWW: <http://www.inforum.cz/inforum2004/pdf/Pokorny_Jan1.pdf>.
17. POKORNÝ, Martin. *Vyvíjíme databázový a informační systém II* [online]. 2004, 12.5.2004 [cit. 2006-12-4]. Dostupný z WWW: <http://www.dbsvet.cz/view.php?cisloclanku=2004051201>.
18. POKORNÝ, Martin. *Vyvíjíme databázový a informační systém V*. [online]. 2004, 2.6.2004 [cit. 2006-07-12]. Dostupný z WWW: <http://www.dbsvet.cz/view.php?cisloclanku=2004060201>.

19. PŘÍBRAMSKÁ, Iva, VOJNAR, Martin. *SUAleph - návod na zprovoznění Z39.50* [online]. 2001, 15.11.2001 [cit. 2007-11-11]. Dostupný z WWW: <http://www.sualeph.cz/z39.htm#kapa>.

20. RIORDAN, Rebecca M. *Vytváříme relační databázové aplikace*. 1. vyd. Praha : Computer Press, 2000. 280 s. ISBN 80-7226-360-9.

21. RUBRINGER, Tomáš. *Z39.50*. Ikaros [online]. 1999, roč. 3, č. 8 [cit. 2007-11-23]. Dostupný z WWW: <http://www.ikaros.cz/node/1034>. ISSN 1212-5075.

22. STEJSKAL, Milan. Osobní komunikace. Leden 2006.

23. SVOBODOVÁ, Eva. *Porovnání vyhledávače Google a báze (Aleph) NK ČR z hlediska věcného vyhledávání* [online]. 2004, 30.3.2004 [cit. 2007-10-22]. Dostupný z WWW: <http://www.phil.muni.cz/kivi/clanky.php?cl=26&rubrika=clanky>.

24. *Systémy automatizace knihoven* [online]. [cit. 2007-10-05]. Dostupný z WWW: <http://mujweb.cz/kultura/automatizace/?X>.

25. ŠEŠERA, Lubor, MIČOVSKÝ, Aleš, ČERVEŇ, Juraj. *Datové modelování v příkladech*. 1. vyd. Praha : GRADA, 2001. 151 s. ISBN 80-247-0049-2.

26. ŠILHA, Jiří. Osobní komunikace. Únor 2006.

27. TUŽOVÁ, Eva. *T SERIES - Úvodem* [online]. Univerzita Karlova v Praze, Ústav výpočetní techniky, 2004, 19.5.2004 [cit. 2007-11-11]. Dostupný z WWW: <http://tseries.cuni.cz/TSERIES-49.html>.

28. Vysoká škola báňská - Technická univerzita Ostrava. *Tabulkové vyjádření relace a její vlastnosti* [online]. 2003 [cit. 2006-05-25]. Dostupný z WWW: <http://www.fs.vsb.cz/books/dbacc20/dbacc02.htm#dbacc020302>.

Seznam obrázků

Seznam tabulek

Evidence výpůjček

Prohlášení:
Dávám svolení k půjčování této diplomové práce. Uživatel potvrzuje svým podpisem, že bude tuto práci řádně citovat v seznamu použité literatury.

V Praze, 14.12. 2007.

Lukáš Marek

Jméno	Katedra / Pracoviště	Datum	Podpis