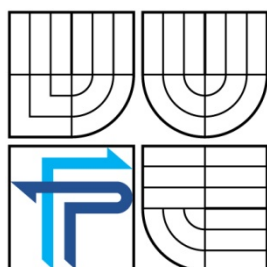


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH IS PRO MÍSTNÍ KNIHOVNU MĚSTA JEVIŠOVICE

THE CONCEPT OF IS FOR THE LOCAL LIBRARY OF JEVIŠOVICE
CITY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DANIEL VAŠÍČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAN LUHAN

BRNO 2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Vašíček Daniel

Manažerská informatika (6209R021)

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

Návrh IS pro Místní knihovnu města Jevišovice

v anglickém jazyce:

The Concept of IS for the Local Library in Jevišovice City

Pokyny pro vypracování:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení, přínos návrhů řešení
Závěr
Seznam použité literatury
Přílohy

Seznam odborné literatury:

- BASL, J. Podnikové informační systémy. Podnik v informační společnosti. 1. vyd. Praha : Grada, 2002. 142 s. ISBN 80-247-0214-2.
- HELD, B. Access VBA: Velká kniha řešení. 1. vyd. Brno : Computer Press, 2006. 639 s. ISBN 80-251-1112-1.
- HERNANDEZ, M. J. Návrh databází. 1. vyd. Praha : Grada, 2006. 408 s. ISBN 80-247-0900-7.
- KOCH, M. Datové a funkční modelování. 2. vyd. Brno : Akademické nakladatelství CERM, 2006. 108 s. ISBN 80-214-3252-7.
- ŘEPA, V. Analýza a návrh informačních systémů. 1.vyd. Praha : EKOPRESS, 1999. 403 s. ISBN 80-86119-13-0.

Vedoucí bakalářské práce: Ing. Jan Luhan

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/2010.

L.S.

Ing. Jiří Kříž, Ph.D.
Ředitel ústavu

doc. RNDr. Anna Putnová, Ph.D., MBA

V Brně, dne 21.04.2010

ABSTRAKT

Bakalářská práce pojednává o návrhu vytvoření databáze pro místní knihovnu města Jevišovice. Práce obsahuje vlastní návrh databáze a její vytvoření v programu Microsoft Access. Dále obsahuje vytvoření pracovního rozhraní pomocí VBA.

Klíčová slova

Návrh databáze, databáze, Microsoft Access, Informační systém, Visual Basic for Applications, VBA

ABSTRACT

The thesis deals with the concept of the Jevišovice city Local Library database creating. The thesis contains the concept of the database and its creation in the Microsoft Access software. Further, the work includes the creation of the VBA communication interface.

Keywords

Database design, database, Microsoft Access, Information system, Visual Basic for Applications, VBA

Bibliografická citace

VAŠÍČEK, D. *Návrh IS pro Místní knihovnu města Jevišovice*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2010. 74 s. Vedoucí bakalářské práce Ing. Jan Luhan

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve znění Zákona č. 121/2000 Sb. o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 17. května 2010

Daniel Vašíček

Poděkování

Děkuji tímto vedoucímu bakalářské práce Ing. Janu Luhanovi za cenné rady a připomínky při vypracování bakalářské práce.

Dále bych chtěl poděkovat paní Janě Staré za pomoc při získávání informací o místní knihovně města Jevišovice.

Obsah

Úvod.....	8
1 Vymezení problému a cíle práce	9
2 Teoretická východiska	10
2.1 Informační systém.....	10
2.2 Databázové systémy a modely.....	12
2.3 Relační databáze	15
2.4 Microsoft Access	20
2.5 VBA (Visual Basic for Application)	24
3 Analýza problému a současné situace	31
3.1 Rozdělení knihoven	31
3.2 Financování knihoven.....	32
3.3 Představení knihovny.....	32
3.4 Současný stav IS	34
3.5 Analýza SWOT	35
3.6 Současný stav informačního systému	37
3.7 Analýza procesů IS	37
4 Praktické řešení situace.....	39
4.1 Požadavky na databázi.....	39
4.2 Rozhraní databáze.....	39
4.3 Návrh databáze	40
4.4 Vstupní masky	46
4.5 Dotazy	47
4.6 Hlavní formulář.....	50
4.7 Umístění a zabezpečení databáze	57
5 Ekonomické zhodnocení a přínos řešení	59
Závěr	60
Seznam literatury	62
Seznam obrázků.....	64
Přílohy.....	66

Úvod

Ve své práci se zaměřím na vytvoření databáze pro místní knihovnu ve městě Jevišovice. Tato databáze by měla sloužit nejen k přehlednému zobrazení vypůjčených knih a evidenci čtenářů, ale také by mělo být vytvořeno rozhraní. Tedy formulář, který bude sloužit pro ovládání této databáze. Je důležité, aby databáze byla kvalitní a zároveň co jednoduchá a intuitivní pro obsluhu daným zaměstnancem. Abych splnil tyto požadavky bude databáze vytvořena v programu Microsoft Access a funkce formuláře pro ovládání bude naprogramována pomocí jazyka VBA (Visual Basic for Applications).

Nejprve se zaměřím na vysvětlení pojmů souvisejících s databázemi a jazykem VBA. V další části práce analyzuji současný stav a popis místní knihovny v Jevišovicích. V závěru se pokusím o navrhnutí databáze, tak aby byla vhodná pro použití v místní knihovně a splňovala její požadavky. Jednotlivé části popíšu a vysvětlím. Na úplný závěr provedu zhodnocení a pokusím se provést přínos své práce pro místní knihovnu v Jevišovicích.

1 Vymezení problému a cíle práce

V místní knihovně města Jevišovice, byl zveřejněn požadavek na zmodernizování současného IS na elektronický. Jedná se o vytvoření databáze a rozhraní pro práci s ní. Jde tedy o digitalizaci současných záznamů.

Cílem mé práce je vytvořit přehlednou a jednoduchou databázi čtenářů, knih a výpůjček naprogramovanými funkcemi. Takto vytvořený systém by měl umět registrovat nové čtenáře, evidovat platby, vytvářet výpůjčky a evidovat vrácení knihy. Zvolený systém by měl být schopný reagovat na případné další požadavky této knihovny.

Důležitou podmínkou firmy bylo, aby mohli databázi používat zaměstnanci s běžnou znalostí PC. Proto jsem se rozhodl databázi vytvořit v programu Microsoft Access 2003, který je součástí balíčku Microsoft Office 2003. Tento balíček je v knihovně na obou počítačích k dispozici a běžně ho využívají.

Předpokládám, že vytvoření nového informačního systému v knihovně povede ke zkvalitnění a zrychlení procesu mezi knihovníkem a čtenářem. Nový informační systém by měl pomoci zjednodušit a zefektivnit práci knihovníka.

2 Teoretická východiska

2.1 Informační systém

„Informační systém můžeme chápat jako množinu prvků, jejich vzájemných vazeb a určitého chování. Jeho cílem je shromažďovat, zpracovávat a poskytovat informace“ (6, str. 10) „V oblasti informačních technologií máme na mysli hardware a software.“ (6, str. 8)

Informační systém nemusí být vázán vyloženě na počítače, ale můžeme si pod tímto pojmem představit např. papírovou kartotéku či diář. V současné době jsou informační systémy neodmyslitelně propojeny s počítači. Tedy veškerou technikou jako hardware, počítačové sítě, tiskárny a faxy.

2.1.1 Informace a data

„Informace je zpráva (sdělení), která snižuje neurčitost (entropii) příjemce.“ (6, str. 10)

Pojmu informace se využívá u lidí. Pokud informaci převedeme do počítače, můžeme mluvit o datech. Opětovným dekódováním dostáváme informace. Data jsou tedy chápána ve smyslu technického záznamu schopného přenosu a dalšího zpracování.

2.1.2 Návrh

„Hlavním artefaktem jsou případy užití (nebo také *modely jednání*, angl. *use cases*). Základními prvky jsou: aktér, scénář a impuls-reakce (zpráva). Případy užití je možno, podobně jako v softwarovém inženýrství, rozšiřovat či generalizovat.

Model spolupráce je dalším artefaktem, který vzniká na základě případů užití. Hledáme zde první náznaky tříd, odpovědností a vztahů. To pak ústí v *objektový model*, který již přesně zachycuje celý systém, vztahy mezi objekty či hierarchii dědění.

Funkční model poskytuje kontrolní pohled na vytvářený systém. Standardem je zde DFD, jež poskytuje snadné grafické vyjádření propojitelné s datovým modelem. DFD obsahují:

- aktéry (obdélník) – například osoba, instituce, jiný systém a podobně,

- datové sklady (obdélník se zaoblenými rohy bez pravé strany) – uchovává data,
- procesy (obdélníky se zaoblenými rohy) – manipulují s daty, jsou algoritmy,
- datové toky (šipky) – předávání datových záznamů.

DFD model je **hierarchický**, to znamená, že procesy se dají postupně zjemňovat. Každý proces tedy obsahuje „vnořený“ diagram, až po takzvané *listové procesy*, které jsou atomické (nedělitelné). Každý proces v DFD obsahuje textový popis (například pseudokód, přirozený jazyk, různé podmínky a podobně), popis omezení (constraints) a také dodatečné informace (např. možnosti optimalizace).

Dynamický model přispívá k pochopení změn v systému. Možné popisy jsou například slovní scénáře, grafické scénáře (např. sekvenční diagramy), mapy událostí (jeden diagram na celý systém) nebo stavové diagramy a tabulky. Samostatnou kapitolou jsou pak ER-diagramy, které zachycují *datový model*.“(12)

2.1.3 Architektura informačního systému

„Jedná se o schématické zobrazení důležitých funkcí informačního systému a jejich vazeb. Obsahuje navíc dokumentaci k hlavním charakteristikám jednotlivých aplikací. „Velmi důležitým hlediskem je volba architektury. Téměř výhradně se používá 3. vrstevná architektura:

- presentační (interakce s uživatelem),
- funkční (vlastní aplikace, bezpečnost, propojení se světem, kontrola,...),
- datová (vlastní data).

Důležitá je i bezproblémová integrace IS, která má dvě hlediska. Vnitřní, kde jde o proškolení pracovníků, nastavení prostředí a podobně. A vnější, kde se jedná zejména o zákazníky a dodavatele. Je nutné si uvědomit, že zadavatel implementace IS bude hledět na tyto aspekty:

- základní údaje (nejen samotného IS, ale také dodavatele, cenu),
- architekturu (zda-li bude vyhovovat),
- reference (po ČR i ve světě),

- provozní prostředí (databázová platforma),
- vývojové prostředí (CASE nástroje),
- dokumentace, jazyková podpora,
- doplňující služby (podpora, školení),
- standardy, specifikace, certifikace (audity, ISO-9000),
- flexibilita (možnost přizpůsobení).“(12)

2.2 Databázové systémy a modely

„Databází myslíme pouze skladiště dat, ve kterých jsou data uložena a zpracovávána nezávisle na aplikačních programech.

Pro samotný přístup k datům uloženým v databázi používáme speciální software. Nazývá se anglicky *Database Management System* (DBMS) nebo česky *Systém řízení báze dat* (SŘBD). Fyzická struktura uložených údajů potom nemusí být aplikačnímu programu, a tím pádem ani uživateli, vůbec známa.

Databázový systém je pojmem, který zastřešuje samotné údaje spravované v databázi společně se softwarem pro přístup k těmto údajům.“ (11)

Při tvorbě databázového systému je možno využít 5 typů datových modelů. V případě využití této práce se bude jednat o relační model. Některé z následujících modelů nemají v současných databázových systémech využití.

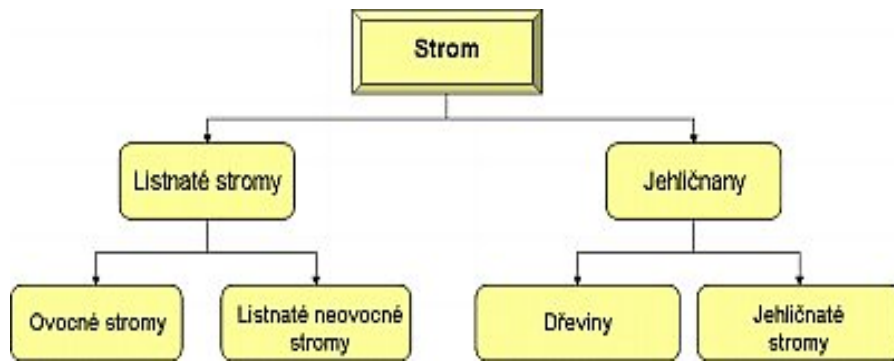
2.2.1 Lineární

„V lineárním modelu není žádná vazba mezi jednotlivými skupinami objektů – tabulkami. Je to jediný datový model, který můžeme implementovat na libovolném médiu.“ (5, str. 20)

Jedná se například o kartotéky v knihovně. V knihovně jsou evidovány knihy pomocí kartotéky, kde není žádná vazba mezi jiným objektem.

2.2.2 Hierarchický

V tomto typu databázových modelů jsou data strukturovaná hierarchicky do podoby obráceného stromu. Vztahy jsou reprezentovány termíny „rodič“ a potomek. Jeden potomek může mít jen jednoho rodiče, ale rodič může mít několik potomků.

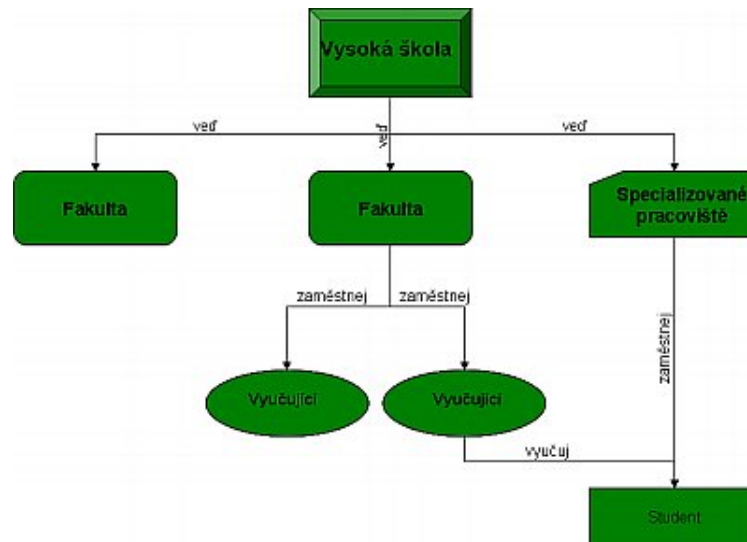


Obrázek 1: Hierarchická struktura (Zdroj: 16)

2.2.3 Síťový

Síťový model byl navrhnut jako pokus k vyřešení problému s hierarchickým modelem.

„Uzel reprezentuje soubor záznamů a množinová struktura reprezentuje a zřizuje vztah v síťové databázi. Je to snadno pochopitelná konstrukce, která vytváří vztah mezi dvěma uzly tak, že jeden uzel je definován jako vlastník a druhý jako prvek (tato metoda je značným vylepšením vztahu rodič/potomek). Množinová struktura podporuje vztah 1: N, neboli jeden záznam v uzlu vlastník může být v relaci k jednomu, nebo více záznamům v uzlu člen. Na druhé straně, jeden záznam v uzlu člen je ve vztahu pouze k jednomu záznamu typu vlastník. Záznam v uzlu typu člen navíc nemůže existovat, aniž by byl ve vztahu k nějakému záznamu v odpovídajícím uzlu typu vlastník.“ (17)



Obrázek 2: Síťový databázový model (Zdroj: 17)

2.2.4 Relační

„Vzniká z několika lineárních modelů spojených dohromady pomocí položky (položek), kterým říkáme **relační klíč**. Toto spojení není trvalé, jako u předchozích modelů, ale vzniká v okamžiku, kdy potřebujeme mít společně k dispozici data ze všech spojených tabulek a zaniká, když práci s modelem ukončíme.“ (5, str. 23)

2.2.5 Objektový

„Nejnovější databázový model. Objektové modely jsou vystavěny na základním prvku – objektu, kde tento objekt má krom svých atributů i definované metody, které určují chování objektu.“(5, str. 23)

„Každý objekt má svou identitu (tím je míněno, že je jednoznačně systémově identifikovatelný). Je charakterizován množinou vlastností objektu, množinou zpráv, na které objekt reaguje, množinou metod, z nichž každá je tvořena částí programového kódu, který je implementací reakce na zprávu. Objekty, které obsahují stejné typy hodnot a metody jsou sdružovány do tříd (typ objektu).

Velikou výhodou je, že definice dat se mohou dědit (a to i mnohonásobně) a objekty různě propojovat, zapouzdřenost a polymorfismus objektů.

Práci s databází lze rozdělit do dvou částí a těmi jsou tvorba databáze (definice objektů) a pokládání dotazů (pomocí příkazů SELECT). Pokládání dotazů v objektově orientované databázi je mnohem jednodušší než v relačních databázích, bohužel však na druhou stranu je mnohem složitější tvorba celé objektové struktury.

Samotné objektové databáze lze rozdělit na 2 typy:

- **Objektově orientované databáze (založené na perzistentních jazycích)**

Takové databáze umí manipulovat s daty, která jsou „trvalá“ (perzistentní). Nepracuje přímo s SQL, ale lze je vnořit do jazyka, mezi ostatní příkazy. S perzistentním jazykem pak sdílí datové typy. Výhodou je, že programátor nemusí znát jazyk SQL (pokud jej samozřejmě nevnoří) a vysoký výkon. Objektový přístup je zabudován přímo do jazyka. Avšak velkou nevýhodou je velmi malá možnost tvorby dotazů běžným uživatelem, perzistentní jazyky jsou totiž jazyky procedurální.

- **Objektově relační databáze**

Jedná se vlastně o nadstavbu relačních databází, kdy se vychází z relační struktury. Dotazovací jazyky (např. SQL) jsou doplněny o další příkazy a funkce, což umožňuje vznik tzv. složených typů (opakují se atributy, skupinové atributy,...). Vytvářejí se sice složitější datové typy, ale díky použití „silných“ dotazovacích jazyků je dosaženo vysoké úrovně ochrany dat. “ (10)

2.3 Relační databáze

Relační model je nejrozšířenějším způsobem uložení dat v databázi. Jedná se o uložení v logickém smyslu.

„Relační databázové modely jsou založeny na teorii relací. Relační databázový model spojuje data do takzvaných relací (tabulek), které obsahují n-tice (řádky tabulky). Relace tvoří základ relační databáze. Relace je struktura záznamů s pevně stanovenými položkami (sloupci / atributy). Skutečné uspořádání záznamů v databázi je úplně nepodstatné a každý záznam v tabulce je identifikovaný atributem, který obsahuje unikátní hodnotu.

Naproti hierarchickému či síťovému databázovému modelu uživatel nemusí znát fyzické umístění záznamu pokud z něho chce získat určitá data.

Když jsou v různých relacích (tabulkách) sloupce stejného typu, potom tyto sloupce mohou vytvářet vazby mezi jednotlivými tabulkami. Kolekce více tabulek, jejich vztahů a dalších součástí potom tvoří relační databázi.“(hernandez)

2.3.1 Relace

„Máme-li množiny, v terminologii teorie množin „domény“ například čísel studentů – D_1 , jmen studentů – D_2 a příjmení studentů D_3 , pak relace na doménách D_1, D_2, \dots, D_n je dvojice $R=(R, R^*)$,

kde $R = R (A_1:D_1, A_2:D_2, \dots, A_n:D_n)$ je **schéma relace**
a $R^* \supseteq D_1 \times D_2 \times \dots \times D_n$ je **tělo relace**.“(5, str. 26)

„Relační model zavádí jedinou datovou strukturu - n-ární relaci s pomocnou strukturou – schématem relace.

Pravidla pro tabulkovou prezentaci relace:

- každý řádek odpovídá jedné n-tici relace,
- pořadí řádků je nevýznamné,
- žádné dva řádky nejsou stejné (tabulka neobsahuje duplicitní řádky),
- pořadí sloupců je nevýznamné,
- význam každého sloupce je určen jménem atributu,
- žádné dva názvy sloupců (atributy) nejsou stejné,
- hodnoty ve sloupcích jsou atomické.“(5, str. 27)

2.3.2 Integrita relačního modelu

„Integrita modelu obecně je takový stav, při kterém uložená data odpovídají vlastnostem objektů reálného světa. Tohoto stavu dosáhneme pomocí integritních

omezení, která můžeme rozdělit na integritní omezení pro entity (relace) a integritní omezení pro vztahy entit (relační vazby).“ (5, str. 28)

2.3.2.1 Integritní omezení pro entity

a) Doménová integrita (integrita hodnot)

„Doména je pojmenovaná množina hodnot stejného typu, například doména křestních jmen. Doménová integrita znamená, že každá hodnota každého atributu relace musí být z množiny hodnot (domény) pro daný atribut přípustných. Z toho plyne, že doména musí být definovaná jako množina hodnot. Přípustné hodnoty pro daný atribut lze specifikovat pomocí datového typu pole, rozsahu hodnot, masky pro vkládání, neprázdné hodnoty a podobně.

b) Entitní integrita

Entitní integrita stanoví, že každá relace musí mít určený primární klíč, který se skládá z jednoho nebo více atributů a jednoznačně identifikuje každý řádek relace. Primární klíč je jednoznačný a minimální, to znamená, že v relaci nesmí existovat dvě n-tice se stejným primárním klíčem a že žádný atribut primárního klíče nelze vypustit. Každý atribut primárního klíče musí být vyplněn a každá n-tice relace musí být vždy identifikovatelná pomocí hodnoty primárního klíče.

c) Referenční integrita

Referenční integrita popisuje cizí klíč a propojení relací. Pro cizí klíč platí, že každá hodnota musí být plně zadaná nebo plně nezadaná, a že každá hodnota cizího klíče se shoduje s hodnotou primárního klíče některé n-tice jiné relace. Dále platí, že cizí klíč a odpovídající primární klíč musí být definovány na stejné doméně a že v databázi nesmí být žádná nesouhlasná hodnota cizího klíče.“ (5, str. 28-30)

Primární klíč a cizí klíč umožňují vytvářet spojení mezi relacemi, což je hlavním účelem relačního datového modelu.

2.3.2.2 Integritní omezení pro vztahy entit

„Integritní omezení pro vztahy omezuje kardinalitu vztahu na poměry 1:1, 1:N, N:1, N:M. Tento poměr uvádí, kolik n-tic relací sobě navzájem odpovídá.“ (5, str. 31)

a) Vztah 1:1

„Vztah 1:1 nám říká, že vždy jedné n-tici (větě) relace odpovídá jedna (nebo žádná) n-tice jiné relace.“ (5, str. 31)

Příkladem může být, že jeden čtenář vlastní jeden čtenářský průkaz (v případě, že uvažuje pouze jednu danou knihovnu).

b) Vztah 1:N a N:1

„Vztah 1:N nám říká, že vždy jedné n-tici (větě) relace odpovídá jedna nebo více n-tic jiné relace.“ (5, str. 32)

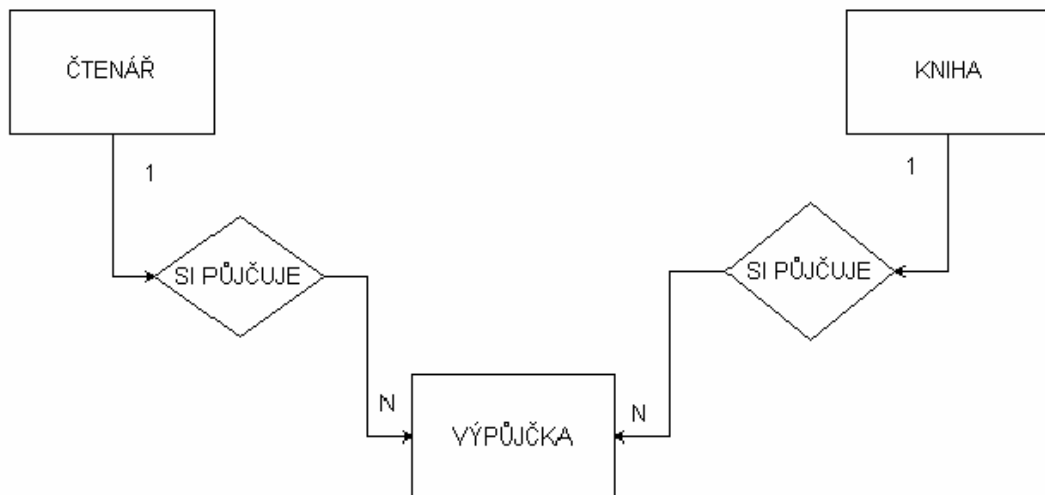
Př.: Jeden čtenář si může vypůjčit více knih. Zároveň z toho vyplývá, že tu stejnou knihu si současně nemůže vypůjčit více čtenářů. Tím je zajištěn vztah 1:N. Opakem je vztah N:1, který můžeme pochopit, že knihu si může půjčit více zájemců, v případě, že počítáme s delším obdobím. Tzn. v případě, že evidujeme např. kdo vše si knihu vypůjčil v posledním roce.

c) Vztah N:M

„Vztah N:M nám říká, že obecně několika n-ticím relace odpovídá jedna nebo více n-tic jiné relace.“ (5, str. 33)

Vzhledem k tomu, že tento vztah je pro nás nežádoucí, protože vazba N:M neumožňuje vést vazbu mezi dvěma entitami, musíme provést rozložení na vztahy 1:N, případně N:1.

Př.: Danou knihu si půjčí více čtenářů. Lze vidět, že knihu si více čtenářů půjčit může a současně čtenář si může půjčit více knih. Abychom tomuto problému předešli, vytvoříme si novou entitu. V tomto případě to může být vypůjčená kniha (vypůjčka, apod.). Výsledná dekompozice lze vidět na obrázku.



Obrázek 3: Dekompozice vztahu N:M (Zdroj: Vlastní tvorba)

2.3.3 Normalizace

„Normalizace je činnost, při které upravujeme návrhy datových struktur tak, aby splňovaly zvolené normalizační formy – úrovně. Tyto normalizační formy či pravidla vycházejí z požadavku na efektivní ukládání dat a minimalizující redundance při zachování integrity a konsistence dat. Datový model, který porušuje některou z normalizačních forem není navržený optimálně. Při normalizaci databáze na vyšší normalizační úrovni musí být normalizován na všech předcházejících.

Normalizace je postupná dekompozice relací (tabulek) do vhodnějšího tvaru, taky aby:

- Byla zachována bezztrátovost při zpětném spojení.
- Byly zachovány závislosti.
- Bylo odstraněno opakování informací (redundance)“ (5, str. 55)

Aby databáze byla normalizována, musí splňovat alespoň následující tři normální formy:

1.normální forma

„Relace je v první normální formě, pokud jsou všechny její atributy definovány nad skalárními obory hodnot (doménami).“ (5, str. 56)

Tzn. že všechny atributy musejí být jednoduché, nikoli vícehodnotové nebo složené.

Př.: Pokud evidujeme u čtenáře jeho adresu (Jevišovice, Znojemska 281, 67153) jedná se o složený atribut a musíme provést jeho dekompozici tak, že v databázi budeme evidovat Město, Ulici a PSČ.

2.normální forma

„Relace je ve druhé normální formě, pokud je v první normální formě a navíc všechny její atributy jsou závislé na celém kandidátním (primárním) klíči.“ (5, str. 58)

„Chceme-li uvést relaci do druhé normální formy, musíme prozkoumat, zda všechny atributy závisí na celém klíči či nikoli.“ (5, str. 58)

3.normální forma

„Relace je ve třetí normální formě, pokud je ve druhé normální formě a navíc všechny její neklíčové atributy jsou vzájemně nezávislé“ (5, str. 60)

2.4 Microsoft Access

Microsoft Access je program navržený pro tvorbu databázových systémů a práci s nimi. Program Microsoft Access umožňuje vytvářet relační databáze a práci v takto vytvořené databázi. Data jsou obsažena v datových tabulkách a ukládána v souborech s koncovkou .mdb, které charakterizují samostatné databázové soubory programu Microsoft Access. Microsoft Access umožňuje uživatelům tvorbu dotazů, filtrů, formulářů, sestav a datových schránek. Dále umožňuje tvorbu maker a programových modulů.

Access poskytuje zobrazení dat na jednom počítači, ale je i možno vytvoření multiuživatelského přístupu nebo sdílení dat pomocí intranetu nebo Internetu.

Access umožňuje i spojení s Microsoft SQL Serverem. Program Microsoft Access je součástí kancelářského balíku Microsoft Access a je tím pádem přístupný široké veřejnosti.

2.4.1 Databázové objekty MS Access

Obsahem této kapitoly je vymezení stručného přehledu databázových objektů nutných pro tvorbu a správu databáze.

2.4.1.1 Tabulky

„Tabulka představuje soubor polí. Tabulka představuje soubor polí (sloupců), která obsahují informace jednoho typu (např. příjmení zákazníka) a záznamů (řádek), kam se ukládají všechny potřebné údaje.“(7, str. 12)

K identifikaci v tabulkách je možné využití primárního a cizího klíče. V tabulce lze definovat datový typ pole (text, číslo, apod.), což zajišťuje správné zobrazení zadaných dat a následnou práci s nimi, dále je možné definovat název pole a jeho popis.

2.4.1.2 Dotazy

„Dotazy lze výhodně využít v případě potřeby analýzy dat, jiného způsobu zobrazení, vytvoření sestavy apod.

Pomocí dotazů lze např. shromáždit data z několika tabulek a seřadit je podle zadaného klíče, provádět výpočty ve skupinách záznamů, aktualizovat data nebo zakládat nové databázové objekty atd.“(7, str. 12)

Dotazy představují nezbytnou součást databázových programů. Užití dotazů a filtrů usnadňuje orientaci v databázi a analýzu dat.

2.4.1.3 Formuláře

„Formuláře usnadňují zadávání a prohlížení dat v databázi. Místo často komplikovaného zadávání dat přímo do tabulek lze využít přehledný formulář funkčně i graficky přizpůsobený vašim potřebám. Lze např. vytvořit vlastní dialog, pomocí kterého budete otevírat ostatní formuláře nebo sestavy, zadávat vstupní data do tabulky atd.

Formuláře slouží jako uživatelské rozhraní pro přístup k datům uloženým v tabulkách. Pomocí formulářů lze jednoduše zadávat a prohlížet data v databázi.“ (7, str. 13)

2.4.1.4 Sestavy

„Často je potřeba data uložená v databázi sumarizovat či prezentovat v tištěné formě. V takovém případě se vytvoří sestava z vybraných dat, a to v takové grafické úpravě, která nejlépe vyhovuje daným potřebám. Jednotlivé záznamy lze řadit a seskupovat do skupin podle zadaného klíče, přidat graf vytvořený na základě zobrazených dat, vytvořit adresní štítky, přidat do sestavy hlavičku firmy včetně grafického loga, přidat pole ze zdroje záznamů sestavy měnit velikost a typ písma, přesunout nebo měnit velikost ovládacích prvků atd.“ (7, str. 13)

2.4.1.5 Makra

„Makra je možno vytvořit v případě potřeby usnadnění nebo automatizace často se opakujících úkolů. Skládají se z jedné nebo více akcí. Makro lze připojit k příkazovému tlačítku formuláře nebo sestavy. Makro lze nadefinovat i tak, že provádí určitou akci pouze při splnění specifikovaných podmínek.“ (7, str. 13)

V případě maker se nejedná o žádné programování.

2.4.2 Datové typy

„Datový typ pole by měl odpovídat tomu, jaká data jsou v poli uložena. Pokud bude například pole obsahovat název firmy nebo jména zákazníka, bude vhodný datový typ text o maximální velikosti zhruba 50 znaků.

Každá tabulka by měla být co nejvíce strukturovaná. Informace je nutno ukládat v co nejmenších logických celcích. Usnadní a zvýší se tak i možnost práce s daty.“ (7, str. 40)

Velikost pole je vhodné zvolit takovou, aby nebylo definováno zbytečně pro velký počet znaků. Takto definované pole by zabíralo zbytečně moc místa na disku a utrpěl by tím i samotný výkon databáze.

2.4.2.1 Text

„Datový typ Text slouží pro ukládání kombinace libovolných znaků (číslic, písmen atd.) do pole tabulky. Pro tento datový typ, lze definovat upřesňující atributy.“ (7, str. 40)

Velikost pole je maximální počet znaků. Maximální velikost pole je 255 znaků. Access má přednastavenou Velikost pole na 50 znaků.

„Atribut Formát určuje, jakým způsobem bude text v poli zobrazen. Vstupní maska naopak určuje formát, ve kterém se tato pole budou zadávat.“ (7, str. 41)

Tento datový typ se využívá i pro určení různých číselných hodnot, které se nevyužívají pro matematické operace, např.: rodné číslo, PSČ, číslo popisné apod.

2.4.2.2 Memo

Datový typ Memo na rozdíl od datového typu Text může obsahovat až 65 535 znaků a nelze tyto pole indexovat.

2.4.2.3 Číslo

„Datový typ Číslo se využívá především pro pole, ve kterých jsou uložena data, která se účastní matematických operací.“ (7, str. 44)

„Atribut Velikost pole je možné nastavit dle potřeby na:

1. bajt (velikost 1 bajt, tj. 0 až 255),
2. celé číslo (2 bajty, tj. -32 768 až 32 767),
3. dlouhé celé číslo (4 bajty, tj. -2 147 483 648 až 2 147 483 647),
4. jednoduchá přesnost (reálné číslo na 7 desetinných míst),
5. dvojitá přesnost (reálné číslo na 15 desetinných míst),
6. replikační identifikátor (16 bajtové pole použité pro vytvoření jednoznačného identifikátoru pro replikaci).“ (7, str. 45)

2.4.2.4 Datum a Čas

Datový typ Datum a Čas se používá pro ukládání dat a časových hodnot. Zde je nejdůležitějším atributem Formát. Možnosti jeho definování jsou zobrazeny v následující tabulce.

datum (obecné)	19.6.1994 17:34:23
datum (dlouhé)	19. června 1994
datum (střední)	19-VI-94
datum (krátké)	19.6.1994
čas (dlouhý)	17:34:23
čas (střední)	5:34 odp.
čas (krátký)	17:34

Tabulka 1: Možnosti formátu datového typu Datum a Čas

2.4.2.5 Ano/Ne

Datový typ Ano/Ne umožňuje ukládání logické hodnoty pravda nebo nepravda. V tabulce je implicitně zobrazen jako zaškrtačací políčko.

2.4.2.6 Měna

„Datový typ Měna je velice podobný číselnému datovému typu. Hodnoty měny mohou být využívány v matematických výpočtech zahrnující data s jedním až čtyřmi desetinnými místy. Přesnost výpočtu je na 15 číslic nalevo od desetinné čárky a 4 číslice napravo od desetinné čárky.“ (7, str. 52)

2.4.2.7 Automatické číslo

Automatické číslo se používá téměř výhradně jako jednoznačný identifikátor záznamu v tabulce. Jedná se o automaticky narůstající číslo buď o 1 nebo náhodně generované.

2.5 VBA (Visual Basic for Applications)

„Visual Basic a jeho odnože se v posledních letech staly hlavním programovacím prostředkem na platformě produktů firmy Microsoft. Různé mutace tohoto programovacího jazyka se používají nejen pro programování samostatných (i síťových) aplikací, ale také pro tvorbu maker v balíku programů Microsoft Office (Visual Basic for Applications) a při programování internetových aplikací, spouštěných na straně serveru (ASP) i klienta (Visual Basic Script).

Programování ve Visual Basicu je počítáno mezi objektově orientované a událostmi řízené techniky.

Programátor může používat velké množství předdefinovaných objektů jako jsou formuláře, textová pole pro zadávání a zobrazování dat, příkazová tlačítka, menu, popisky a velké množství dalších objektů. Souhrnně tyto objekty nazýváme ovládacími prvky.

Každý ovládací prvek má definovány své vlastnosti, metody a události. Tyto vlastnosti udávají vzhled a chování ovládacího prvku v aplikaci. U textového pole lze například definovat font, kterým bude v poli zobrazen text, velikost pole a jeho umístění na formuláři, název, pomocí něhož se na ovládací prvek bude odkazovat programový kód a mnoho dalších vlastností.“ (8, str. 3)

2.5.1 Základní prvky VBA

V této části budou popsány základní prvky, které budou využity v mém zpracování problému práce.

2.5.1.1 MsgBox

Užití této funkce je velmi časté. „Okno se zprávou se používá vždy, když program potřebuje uživatele na něco upozornit nebo o něčem informovat. Typickým příkladem může být zobrazení chybového hlášení.

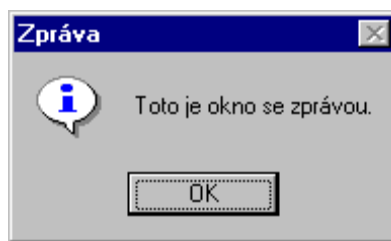
Funkce MsgBox zobrazí dialogové okno se zprávou a čeká dokud uživatel nestiskne tlačítko. Vrací hodnotu, která určuje, které tlačítko uživatel stiskl. Kromě textu zprávy uživatel také definuje, jaká tlačítka se mají zobrazit, které z nich bude předvoleno a která ikona bude zprávu graficky doplňovat.

Syntaxe funkce je následující:

```
MsgBox(prompt[, buttons][, title][, helpfile, context])“
```

(8, str. 15-16)

Argumenty uvedené v závorkách jsou nepovinné.



Obrázek 4: MsgBox

Popis argumentů:

prompt

Řetězcový výraz zobrazený jako dialogová zpráva. Maximální délka promptu je závislá na šířce znaků písma, které je použito (cca 1024 znaků).

title

Řetězcový výraz zobrazený v titulkovém pruhu dialogu. Pokud title vynecháte, je do titulkového pruhu vloženo jméno aplikace.

helpfile

Řetězcový výraz určující název souboru, který obsahuje text nápovědy k dialogu. Je-li argument helpfile určen, musí být také určen argument context.

context

Číselný výraz, který je přiřazen k tématu nápovědy.

2.5.1.2 TextBox

Textové pole se ve formuláři vyskytuje nejčastěji. Do textového pole uživatel doplňuje textové nebo číselné hodnoty. Jeho nejvýznamnější vlastností je Value (hodnota), která představuje obsah pole.

2.5.1.3 Label

Popisek umožňuje uživateli vložit do formuláře popisky a komentáře k prvkům. Většinou se přidává před každé textové pole, aby bylo jasné, co má textové pole obsahovat.

2.5.1.4 ComboBox

Pole se seznamem je ovládací prvek, ze kterého může uživatel vybrat jednu z předdefinovaných hodnot. Jakmile uživatel klikne myší na prvek, zobrazí se mu rozvírací seznam, ze kterého si vybere jednu hodnotu. Hodnoty mohou být předdefinované nebo přiřazené programovým kódem pomocí metody `AddItem`.

2.5.1.5 Image

Tento ovládací prvek se používá k zobrazení grafického obrázku. Používá se pro zobrazení grafického obrázku, ze souboru nebo ze schránky.

2.5.1.6 OptionButton

Prvek `OptionButton` (Přepínač) umožní uživateli vybírat z několika možností. Používají se vždy ve skupinách nejméně po dvou. “Grupování” lze zajistit u skupin vždy buď : shodným názvem ve vlastnosti objektu `GroupName` nebo přidáním ovládacího prvku `Frame` – rámeček.

2.5.1.7 CheckBox

Prvek `CheckBox` (Zaškrtačací políčko) se hodí , pokud má uživatel možnost ze dvou výběru (ANO/NE)

2.5.2 Proměnné a základní datové typy jazyka VBA

V této části bude vysvětlena deklarace proměnných, používání, jejich platnost a používané datové typy.

2.5.2.1 Proměnné

„Proceduře nebo funkci lze předávat jeden nebo více argumentů. Předává-li se více než jeden argument, jsou argumenty navzájem odděleny čárkou. Syntaxe každého argumentu je následující:

`[Optional] [ByVal] názevproměnné [()] [As typ]`

Implicitní způsob předávání argumentů je tzv. odkazem (někdy se uvádí jménem). To znamená, že procedura může měnit hodnotu této proměnné a nová

hodnota zůstane zachována i po skončení procedury při dalším běhu programu. Pokud chcete předávat argument hodnotou, musíte uvést před argument klíčové slovo **ByVal**. Případné závorky za názvem proměnné určují, že předávaným argumentem je indexové pole. Typ proměnné se uvádí za klíčové slovo **As**.“ (8, str. 21-22)

2.5.2.2 Datové typy

V následující tabulce jsou vyobrazeny používané datové typy, používané ve VBA.

„Visual Basic používá následující datové typy: **Byte, Boolean, Integer, Long, Single, Double, Date, Currency, String, Variant, Object** nebo uživatelem definovaný typ. Popis datových typů bude uveden později. Pokud datový typ argumentu není explicitně uveden, stává se datovým typem **Variant**, což znamená, že proměnná může obsahovat jakýkoli typ dat.“ (8, str. 21-22)

Název typu	Popis
Byte	Celočíselná hodnota zabírající jeden bajt, tj. 0 až 255.
Integer	Celočíselná hodnota se znaménkem zabírající nejvýše dva bajty, tj. -32 768 až 32 767.
Long	Celočíselná hodnota se znaménkem zabírající nejvýše čtyři bajty.
Single	Reálné kladné číslo s plovoucí čárkou zabírající nejvýše čtyři bajty.
Double	Reálné kladné číslo s plovoucí čárkou zabírající nejvýše osm bajtů.
Currency	Reálné číslo se čtyřmi desetinnými místy zabírající nejvýše osm bajtů.
String	Řetězec znaků, jehož délka je prakticky neomezená – činí asi dvě miliardy znaků.
Boolean	Dva bajty obsahující hodnotu True nebo False (pravda nebo nepravda).
Date	Osm bajtů obsahující hodnotu datum/čas, tj. 1. leden 100 až 31. prosinec 9999.
Object	Čtyři bajty obsahující odkaz na libovolný Object .
Variant	Tento datový typ může obsahovat kterýkoli z předešlých datových typů v rozsahu, který byl uveden. Tento datový typ je implicitně přiřazen každé proměnné, u které explicitní deklarace typu chybí. Nevýhodou, která je daní za flexibilitu proměnných tohoto typu, je jejich délka 16 bajtů a o něco pomalejší interpretace hodnot při běhu programu. Ve většině případů použití této proměnné musí Visual Basic provést odpovídající konverzi podle typu výrazu, ve kterém je proměnná typu Variant právě použita.

Tabulka 2: Datové typy jazyka VBA (zdroj: 8, str. 29)

2.5.2.3 Deklarace proměnných

„Proměnné ve Visual Basicu – stejně jako v jiných programovacích jazycích – slouží k dočasnému uchování hodnot během vykonávání programového kódu. Každá proměnná se musí deklarovat. Deklarace proměnné se provádí před jejím prvním použitím. Pojem deklarace proměnných v sobě zahrnuje nejen definici názvu proměnné a jejího datového typu, ale také obor (veřejné Public – či soukromé – Private – proměnné) a dobu platnosti (statické, které si uchovávají svou hodnotu i po skončení procedury, ve které jsou deklarovány, nebo takové proměnné, které „žijí“ pouze po dobu provádění procedury).“ (8, str. 25)

„Všechny proměnné jsou deklarovány v jedné z následujících částí programového kódu:

1. V deklarační sekci procedury. Proměnné, které se zde deklarují, musejí být vždy uvozeny klíčovým slovem **Dim**. Takové proměnné jsou ve všech případech soukromé vzhledem k proceduře a nejsou tudíž dostupné z jiných částí modulu ani aplikace.

2. V deklarační sekci modulu. Tj. v sekci označené jako **(General)** v levém rozbalovacím seznamu na obr. 8. Zde se dají deklarovat dva druhy proměnných:

- Soukromé vzhledem k modulu, ve kterém jsou deklarovány. Tyto proměnné mohou být uvozeny buď klíčovým slovem Dim nebo Private. V tomto případě není mezi Dim a Private žádný rozdíl. Takto deklarované proměnné mohou používat všechny procedury daného modulu, nejsou však dostupné v ostatních modulech aplikace.
- Veřejné. Veřejné proměnné se v deklarační části modulu definují pomocí klíčového slova **Public**. Tyto proměnné jsou dostupné ve všech modulech aplikace.“ (8, str. 26)

2.5.2.4 Doba platnosti proměnných

„Kromě oboru platnosti lze proměnným přisoudit také dobu platnosti. Hodnoty proměnných deklarovaných na úrovni modulu se zachovávají po celou dobu, kdy je databázová aplikace otevřená. Lokální proměnné deklarované v procedurách klíčovým slovem Dim existují pouze po dobu běhu kódu procedury. Při novém spuštění

procedury se tyto proměnné znovu inicializují. Existuje však způsob, jak zachovat hodnotu lokální proměnné i pro následující spuštění procedury, ve které je deklarovaná. Taková lokální proměnná musí být definována jako statická klíčovým slovem **Static**. Přístup k lokální proměnné deklarované v proceduře má pouze její procedura, avšak hodnota proměnné se zachovává i po ukončení běhu procedury a vstupuje do ní při jejím dalším volání.

[Dim] [Static] NázevProměnné [As typ]“ (8, str. 28)

3 Analýza problému a současné situace

3.1 Rozdělení knihoven

Knihovny poskytující veřejné knihovnické a informační služby se řídí podle zákona č. 257/2001 Sb. tzv. knihovního zákona.

3.1.1 Knihovny zřizované Ministerstvem kultury ČR

Národní knihovna České republiky

„Je knihovnou s univerzálním knihovním fondem doplněným specializovanými fondy. Národní knihovna trvale uchovává konzervační a historický fond. Je centrem systému knihoven, kde vykonává koordinační, odborné, informační, vzdělávací, analytické, výzkumné, standardizační, metodické a poradenské činnosti.

Knihovna a tiskárna pro nevidomé K. E. Macana

Knihovna a tiskárna pro nevidomé K. E. Macana je knihovnou s univerzálním knihovním fondem; garantuje zabezpečení a poskytování služeb.

Moravská zemská knihovna v Brně

Je knihovnou s univerzálním knihovním fondem, případně doplněným specializovanými fondy; trvale uchovává konzervační fond a historický fond. Plní funkci krajské knihovny v Jihomoravském kraji.“(14)

3.1.2 Krajské knihovny (zřízené příslušným orgánem kraje)

„Mezi krajské knihovny jsou zařazeny bývalé státní vědecké knihovny. Jejich význam se liší v závislosti na rozsahu a skladbě fondu, zpravidla však mají univerzální zaměření svého fondu.

Krajská knihovna má povinnost zajišťovat pro další knihovny v kraji (základní) tzv. regionální služby (vzdělávání knihovníků v regionech, funkce krajských vzdělávacích center; problematika výměnných fondů , organizace, evidence, doprava, financování; krajské systémy meziknihovních výpůjčních služeb; kontrola projektů

VISK, koordinace a kooperace...). Tyto knihovny mohou také plnit v místě svého sídla i funkci základní knihovny, podílí se tedy na zajištění veřejných knihovnických a informačních služeb obce.“(14)

3.1.3 Základní knihovny

„Základní knihovny jsou knihovny s univerzálním knihovním fondem nebo se specializovaným knihovním fondem a jsou součástí systému knihoven vykonávající informační, kulturní a vzdělávací činnosti.“(14)

3.1.4 Specializované knihovny

„Vedle univerzálních knihoven existují velké knihovny specializované, které se snaží doplňovat v relativní úplnosti svůj fond podle stanoveného profilu. Mohou být řízené státními organizacemi nebo působící v rámci jiných subjektů (např. vysoké školy). Spolupracují s knihovnami v oblasti své specializace při zavádění nových technologií v oblasti zajištění veřejných knihovnických a informačních služeb.“(14)

3.2 Financování knihoven

„Většina knihoven je financována z veřejných zdrojů, tj. z rozpočtu obcí, měst, škol, vědecký ústavů a dalších institucí, které provozují knihovny. Ze státního rozpočtu je financován jen omezený počet ústředních knihoven, jako jsou Národní knihovna ČR, Státní technická knihovna, Národní lékařská knihovna apod.

Knihovny mohou pro svou činnost využívat několik státních dotačních programů, podpůrné programy Evropské unie, nadační fondy a další zdroje:

- Projekt internetizace knihoven.
- VISK – Veřejné informační služby knihoven.
- Knihovna 21. století.
- Strukturální fondy Evropské unie.
- Další programy a projekty na podporu knihoven.“(15)

3.3 Představení knihovny

V této části bude představena knihovna, pro kterou bude navržen informační systém.

3.3.1 Profil knihovny

Místní knihovna města Jevišovic je veřejnou knihovnou, jejímž účelem je poskytování veřejných knihovnických a informačních služeb, při zachování rovných podmínek pro všechny občany. Tato knihovna umožňuje půjčování knih a časopisů pro děti i dospělé, případně další dodatečné služby. A to beletrii, naučnou literaturu, encyklopedie, jazykové slovníky, cizojazyčnou literaturu, časopisy, knihy z jiných knihoven (MVS), knihy z výměnného fondu (Městská knihovna Znojmo), internetové služby, poskytování naučných CD-ROM (prezenčně).

Pro veřejnost je knihovna otevřena dva dny v týdnu, v úterý a ve čtvrtek – v ostatní dny působí jako informační centrum pro turistické návštěvníky Jevišovic. Knihovní fond tvoří celkem 9 656 svazků, 32 exemplářů docházejících periodik. Knihovna využívá i výměnný fond Městské knihovny ve Znojmě. Spolupracuje s regionálním oddělením, v roce 2005 bylo půjčeno z VF 267 knih a 159 časopisů v celkové hodnotě 54 084,-Kč. Knihovnické služby využívá 179 registrovaných čtenářů.

Malebné městečko Jevišovice leží v severní části znojemského okresu. Zhruba 20 km od Znojma. První zmínka o Jevišovicích pochází z roku 1289. V roce 1990 bylo historické jádro Jevišovic vyhlášeno městskou památkovou zónou. Jevišovice spadají do DSO (Dobrovolný svazek obcí) Jevišovicka. Do tohoto mikroregionu spadají další obce z okolí (Střelice, Rozkoš, Černín, Vevčice, Boskovštejn, Jiřice u Moravských Budějovic, Hluboké Mašůvky). DSO Jevišovicka díky příspěví SKUPINY ČEZ vybuodovala informační systém především s ohledem na cestovní ruch a propagaci mikroregionu jako celek s důrazem na památky a zajímavosti v jednotlivých obcích mikroregionu.

3.3.2 Historie knihovny

První zpráva o zřízení knihovny je z 23. dubna 1878. Starosta oznamoval, že se obecní knihovna ze školní budovy přenesla do obecní radnice. V roce 1920 byla zvolena knihovní rada a ustanoven první knihovník-obecní tajemník Vinc. Musil, v roce 1922 přebírá funkci knihovníka pan učitel Cyril Špalek, po té pan učitel Ed. Krechler. Knihovna je přemístěna do nové budovy Koménia. V roce 1923 má knihovna 1771

svazků a sice: Knihovna města Jevišovic 471 svazků, knihovna čtenářského spolku, Kroužku starožitnictví a Sokola 1300 svazků, půjčovalo si 164 čtenářů, půjčeno 3821 knih, z toho 3502 zábavných a 319 naučných.

V roce 1966 byla knihovna profesionalizována a přestěhována do budovy radnice, kde sídlí dosud.

V roce 2003 knihovna získala grant z Ministerstva kultury na rozšíření internetových stanic v knihovně. Kromě půjčování nabízí i veřejný internet, zprostředkovává meziknihovní výpůjční služby, bibliograficko-informační služby a turistické informace. V roce 2005 uskutečnilo 202 registrovaných čtenářů 9047 výpůjček. Knihovna spolupracuje se školní družinou, mateřskou školkou a základní školou. Žáci zde vystavují své kresby, jsou pořádány besedy a knihovnické lekce, soutěže o vzorného čtenáře. Pro staré občany v Domově důchodců v Jevišovicích je uskutečňováno půjčování knih z fondů jevišovické knihovny.

3.4 Současný stav IS

Stav informačních technologií v knihovně je velice zastaralý. V knihovně je možno využívat počítač k volnému přístupu k internetu. Současné využití výpočetní techniky slouží hlavně k administrativním účelům.

3.4.1 Hardware

Knihovna je vybavena dvěma počítači. První využívá zaměstnankyně knihovny a druhý je připraven pro použití návštěvníků knihovny k přístupu k internetu. Dále je knihovna vybavena multifunkčním zařízením HP LaserJet. Toto zařízení je využíváno pro tisk, kopírování a možno také jako skener pro čtenáře a zákazníky, kteří by měli zájem využít těchto služeb.

Obě dvě počítačové soustavy, které knihovna využívá jsou zastaralé, pomalé a postupně se začínají stávat nevyhovujícími. Stejně tak jako CRT monitory jsou v dnešní době již nevyhovující.

3.4.2 Software

Knihovna nevyužívá žádný speciální software. V obou počítačích je operační systém Microsoft Windows XP a kancelářský balík Microsoft Office 2003.

3.4.3 Počítačová síť

Knihovna je součástí počítačové sítě celého městského úřadu města Jevišovice. Tato síť je rozšířena do obou počítačů a pomocí této sítě připojeny na intranet a internet. Součástí této sítě je poštovní server. Žádná data v této síti nejsou sdílána ani nikde společně ukládána. Síť je vedena strukturovanou kabeláží, v současné době se pracuje na zavedení bezdrátového připojení WiFi, které prozatím funguje pouze pro potřeby starosty. K síti je trvale připojeno 9 počítačů.

Internet i poštovní server poskytuje pro městský úřad v Jevišovicích společnost SkyNet a.s.

3.5 Analýza SWOT

„SWOT analýza je metoda, pomocí které je možno identifikovat silné (anglicky: Strengths) a slabé (anglicky: Weaknesses) stránky, příležitosti (anglicky: Opportunities) a hrozby (anglicky: Threats), spojené s určitým projektem, typem podnikání, podnikatelským záměrem, politikou (ve smyslu opatření) apod. Jedná se o metodu analýzy užívanou především v marketingu, ale také např. při analýze a tvorbě politik (policy analysis). S její pomocí je možné komplexně vyhodnotit fungování firmy, nalézt problémy nebo nové možnosti růstu. Je součástí strategického (dlouhodobého) plánování společnosti.“ (13)

3.5.1 Silné stránky

Za silné stránky knihovny města Jevišovice považují možnost širokého výběru knih a časopisů. Knihovní fond tvoří 9523 svazků, z toho 3288 naučná literatura a 6235 beletrie.

Další silnou stránkou je spolupráce s jinými knihovnami. Knihovna si dlouhodobě půjčuje knihy z jiných knihoven a následně je může půjčovat vlastním čtenářům.

Mezi silné stránky taky patří možnost využít výpůjčního fondu městské knihovny města Znojma. Čtenář si vybere knihu z tohoto fondu a ta mu je následně poštou doručena do místní knihovny, kde si ji bude moci vypůjčit.

3.5.2 Slabé stránky

Za slabé stránky bych označil omezenost finančních zdrojů, špatnou vybavenost informačními technologiemi.

Knihovna města Jevišovice je závislá na finančních prostředcích, které jí poskytne město Jevišovice. Tyto finanční prostředky jsou velice omezené. Finance, které získá vybráním pravidelných ročních registračních poplatků jsou nízké a nepokryjí ani potřebné náklady.

Vybavení výpočetní technikou je ve velice špatném stavu. Vzhledem k tomu, že je zde možnost využití volného přístupu k internetu, považuji současnou vybavenost za nedostačující.

3.5.3 Příležitosti

Příležitostí pro knihovnu města Jevišovice považuji státní a evropské dotace. Jedná se hlavně o ministerstvo pro místní rozvoj anebo ministerstvo kultury. V současné době byla schválena dotace pro vybudování dětské knihovny. Město by mělo usilovat o získání více dotací od těchto ministerstev pro svůj rozvoj, aby mohlo nabídnout lepší služby a novější knihy pro své čtenáře.

Další příležitostí je pořádání akcí pro místní základní školu. Knihovna již dlouhá léta tuto spolupráci podporuje a škole nabízí různé tématické besedy podle ročníku, pro který je beseda určena. Rovněž pořádá akce, které nejsou spojeny se základní školou, ale i pro širokou veřejnost. Např. Březen měsíc knihoven, kdy jsou noví čtenáři osvobozeni od registračního poplatku nebo Týden knihoven, kdy jsou čtenářům odpuštěny sankce za pozdně vrácené knihy apod.

Příležitost je pro knihovnu i DSO (dobrovolný svazek obcí) Jevišovicka, který by se mohl podílet na rozvoji knihovny a zároveň tyto obce by mohly čerpat výhod z rozvoje knihovny v Jevišovicích.

3.5.4 Hrozby

V současné době se knihovna potýká s upadajícím zájmem čtenářů. Lidé nemají zájem o četbu, proto se snaží knihovna pořádat aktivity spojené s pořádáním akcí pro místní základní školu a tím i nalákat mladé čtenáře.

3.6 Současný stav informačního systému

Knihovna v současné době prochází modernizací informačního systému a to i vzhledem k tomu, že využívá výpočetní techniku zastaralou, její využití je dle mého názoru pod její možnosti. V dnešní době by měly i sebemenší firmy, společnosti a jiné subjekty využívat elektronických databází, které mohou usnadnit a zefektivnit práci. Pro zkvalitnění, usnadnění a zefektivnění práce by bylo vhodné využít elektronickou databázi.

3.7 Analýza procesů IS

V knihovně jsem vyznamenal tři důležité procesy, na které se bude třeba zaměřit při tvorbě návrhu IS. Jedná se o vypůjčení knihy, vrácení knihy a platbu poplatků. Rozhodování mezi jednotlivými procesy je následující. Čtenář po příchodu do knihovny vrátí půjčené knihy, v případě, že žádné vypůjčené knihy nemá může si rovnou jít vybrat knihu dle svého přání.

3.7.1 Vypůjčení knihy

Jakmile si čtenář vybere knihy, časopisy nebo CD-ROM, následuje provedení výpůjčky. Zde se запиší do jeho průkazu čísla titulů, které si vypůjčil. Zároveň musí proběhnout zápis do evidence knihovny. Dále se musí kontrolovat, zda má čtenář zaplacené registrační poplatky, v případě že nemá, musí ihned zaplatit registrační poplatek dle ceníku.

3.7.2 Vrácení knihy

Pokud jde čtenář vrátit knihu, je u každé knihy kontrolováno, zda byla vrácena včas. Pozdní vrácení se dělí do tří částí, kde se opět postupuje dle ceníku knihovny. První etapa pozdního vrácení je 31 dní po lhůtě vrácení, druhá etapa je 44 dní a třetí etapa je 75 dní po lhůtě vrácení. Čtenář si může knihu vypůjčit na jeden měsíc. Z tohoto

ceníku tedy lze vyvodit, že si může knihu ještě druhý měsíc bezplatně ponechat a nemusí ani hlásit její prodloužení.

3.7.3 Platba poplatků

Registrační poplatek

- | | |
|-------------------|-------|
| a) Dospělí | 30 Kč |
| b) Student a dítě | 10 Kč |
| c) Důchodce | 20 Kč |

Pozdní vrácení

- | | |
|----------------------------|-------|
| a) 31 dní po lhůtě vrácení | 10 Kč |
| b) 44 dní po lhůtě vrácení | 20 Kč |
| c) 75 dní po lhůtě vrácení | 40 Kč |

Další poplatky

- Ztráta nebo poškození čtenářského průkazu 20 Kč

4 Praktické řešení situace

V této kapitole se pokusím o navrhnutí a vytvoření informačního systému pro místní knihovnu města Jevišovice. Navrhnou databázi a následně se budu zabývat i tvorbou aplikací pro práci s touto databází.

4.1 Požadavky na databázi

Prvním krokem návrhu nové databáze bylo stanovení požadavků. Jedná se o následující požadavky: jednoduchost, databáze by měla být snadno obsluhovatelná, nemusí být sdílená na serveru a jedním z nejdůležitějších požadavků je její cena.

Jedním z požadavků na databázi je její jednoduchost a měla by být snadno obsluhovatelná, vzhledem k tomu, že zaměstnankyně která by měla s touto databází v budoucnu pracovat má jen základní znalosti práce s počítačem. Databáze by měla být jednoduchá, aby bylo možno bez problému přidávat nebo odebírat záznamy, či tabulky. Dalším faktem tohoto požadavku je její rychlost na starších počítačích, které jsou v knihovně využívány.

Dalším požadavkem je její cena. Vzhledem k tomu, že je knihovna financována z městského rozpočtu, ze kterého se jí samozřejmě mnoho financí nad potřebu nedostává.

Databáze nemusí využívat žádné síťové služby, bude umístěna pouze na jediném počítači, na kterém bude také využívána. Nemusí proto vyžadovat sdílení, umístění na serveru ani přístup ze vzdáleného počítače.

4.2 Rozhraní databáze

Po zvážení všech požadavků jsem se rozhodl pro vytvoření databáze v programu Microsoft Access.

Vhodné by bylo i použití například Microsoft SQL serveru, ovšem toto by se týkalo spíše složitějších databází a jejich propojení. V mém případě je však třeba vytvořit jednoduchou databázi a tuto databázi vybavit formulářem s naprogramovanými funkcemi. Pro tento případ je tedy použití SQL Serveru zbytečné.

Vzhledem k tomu, že je Microsoft Access součástí kancelářského balíku Microsoft Office a jedná se o jednoduchý databázový nástroj, zvolil jsem ho jako vhodnou variantu. V knihovně je již nainstalován na obou počítačích, proto i v případě výpadku jednoho počítače je možnost využít případně počítač druhý. Tím budou náklady na pořízení nulové a vzhledem k tomu, že se jedná i o jednoduchý a snadno obsluhovatelný nástroj budou splněny i další požadavky na tento projekt. V Accessu je také možné naprogramovat potřebné knihovnické funkce pomocí jazyka Visual Basic for Applications, což bude hlavní náplní této práce.

4.3 Návrh databáze

Navržená databáze je složena dohromady ze 12 tabulek. Za klíčové tabulky, nebo-li ty co považuji za nejdůležitější, bych určil následující: KNIHA, ARCHIV, VÝPŮJČKY, PLATBY, READER a NAPSAL.

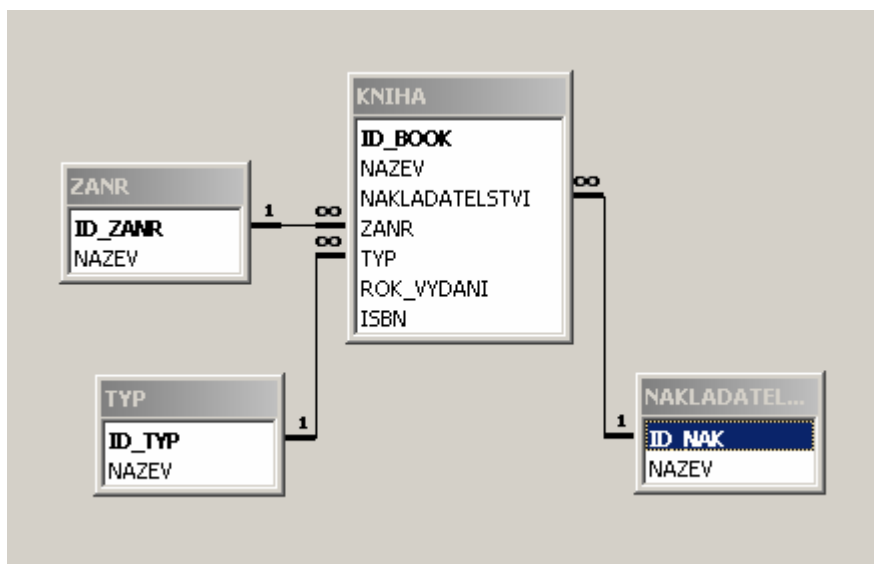
Na obrázku 5, lze vidět logický návrh databázové struktury. Celá databáze respektuje pravidla integritních omezení pro vztahy entit tzn., že jsou dodrženy pravidla vztahů 1:N, N:1 a v případě vztahu N:M je provedena dekompozice na potřebný vztah a pravidla normálních forem.

Veškerá identifikační čísla, která jsou v jednotlivých tabulkách použita jsou generována automaticky. A ve většině případů slouží jako primární klíče tabulky. Většinou je použit datový typ Text. Přihlížel jsem k tomu, že data budou zpracovávána spíše jako řetězec.

4.3.1 Tabulka KNIHA

Tabulka KNIHA obsahuje potřebné informace o knihách, časopisech, případně CD-ROM, které jsou v knihovně poskytovány k půjčení.

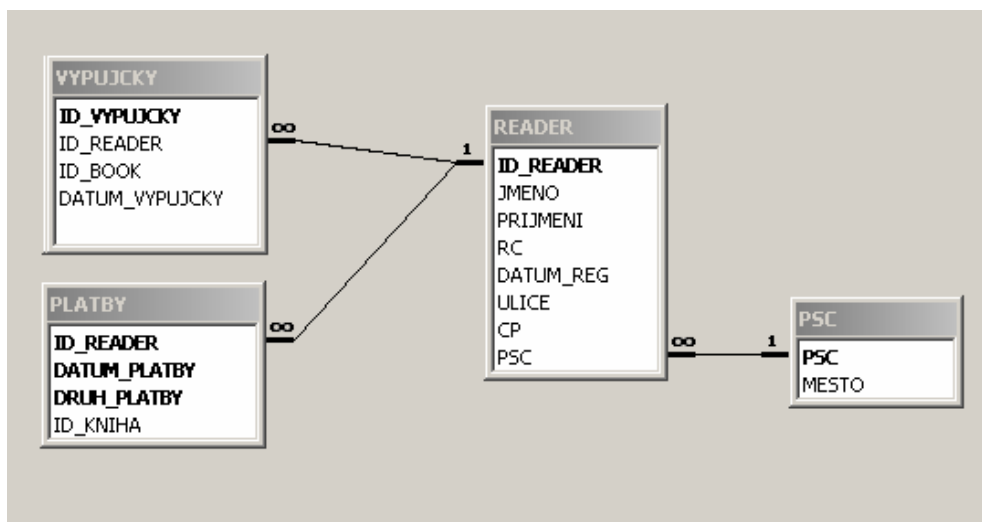
Každá kniha má své identifikační číslo (ID_BOOK), které je také primárním klíčem. Dále se v této tabulce eviduje také název knihy, rok vydání, ISBN (v případě časopisů se jedná o ISSN), žánr, typ a nakladatelství. Pole název, rok vydání a ISBN jsou datového typu Text. Žánr, typ a nakladatelství jsou datového typu číslo. Tyto vazby jsou provázány se souvisejícími tabulkami stejných názvů.



Obrázek 5: Provázání knihy s číselníky

4.3.2 Tabulka READER

Tabulka READER obsahuje informace o čtenáři. Každému čtenáři je opět přiřazeno identifikační číslo, jméno, příjmení, rodné číslo, datum registrace a adresa (ulice, číslo popisné, PSČ). Rodné číslo slouží pro určení jedinečnosti a zamezení duplicity záznamů v tabulce. Pole PSČ obsahuje vazbu na stejnojmennou tabulku PSC.

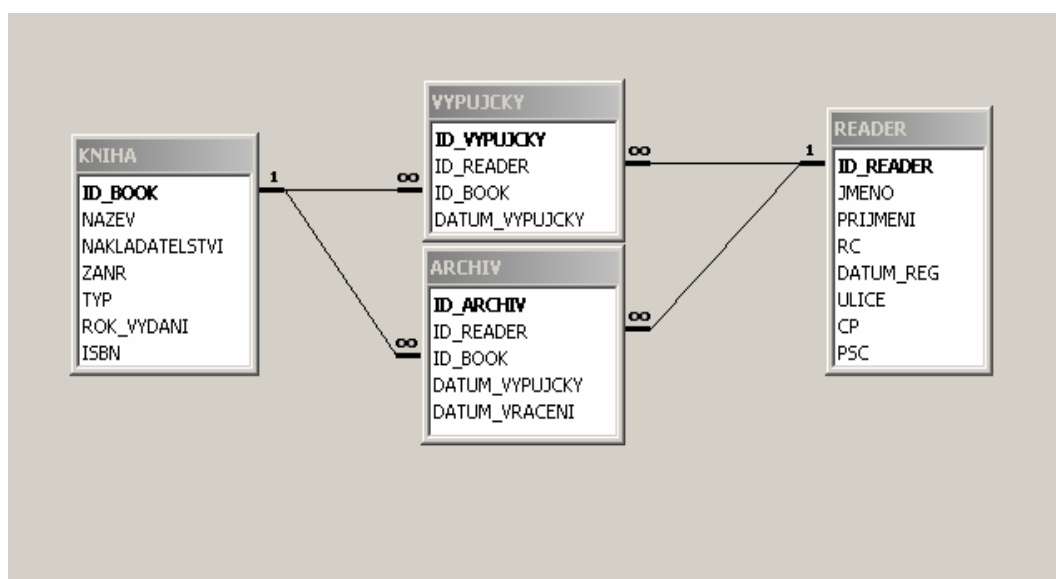


Obrázek 6: Vazby čtenáře na další tabulky

U čtenáře jsme se rozhodli neevidovat telefonický ani emailový kontakt. Vzhledem k tomu, že v případě nevrácení knihy se používá zasílání upomínek poštou, bylo by evidování těchto kontaktů v databázi zbytečné. Návrhem byla i změna tohoto systému na zasílání upomínek pomocí emailu. Bohužel tento systém by byl těžko realizovatelný, protože věkový průměr obyvatel města Jevišovic je 40,7 let, většina čtenářů je spíše důchodového věku a ne každý obyvatel města má pravidelný přístup k internetovému připojení. Proto by přechod na jiný systém byl nevhodný pro většinu čtenářů.

4.3.3 Tabulky VYPUJCKY, ARCHIV

V tabulce VYPUJCKY jsou evidovány veškeré knihy, které jsou momentálně vypůjčeny. V tabulce ARCHIV jsou evidovány knihy, které byly vypůjčeny, ale momentálně jsou vráceny a nachystány k vypůjčení. V těchto tabulkách se evidují hlavně identifikační čísla (ID čtenáře a knihy), dále pak datum výpůjčky, v případě tabulky ARCHIV se eviduje dále datum vrácení knihy. Primárním klíčem v těchto tabulkách je jejich vlastní identifikační číslo.

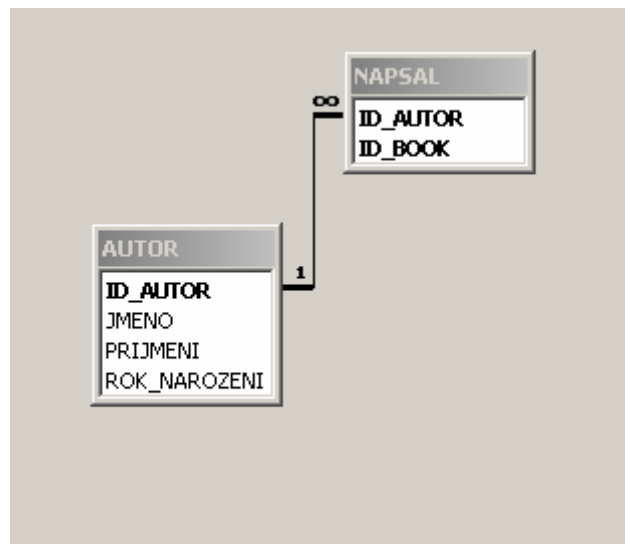


Obrázek 7: Vazby tabulek ARCHIV a VYPUJCKY

Po vrácení knihy jsou údaje přeneseny z tabulky výpůjček do tabulky archiv a přidáno datum vrácení. A z tabulky výpůjček je záznam vymazán.

4.3.4 Tabulka AUTOR

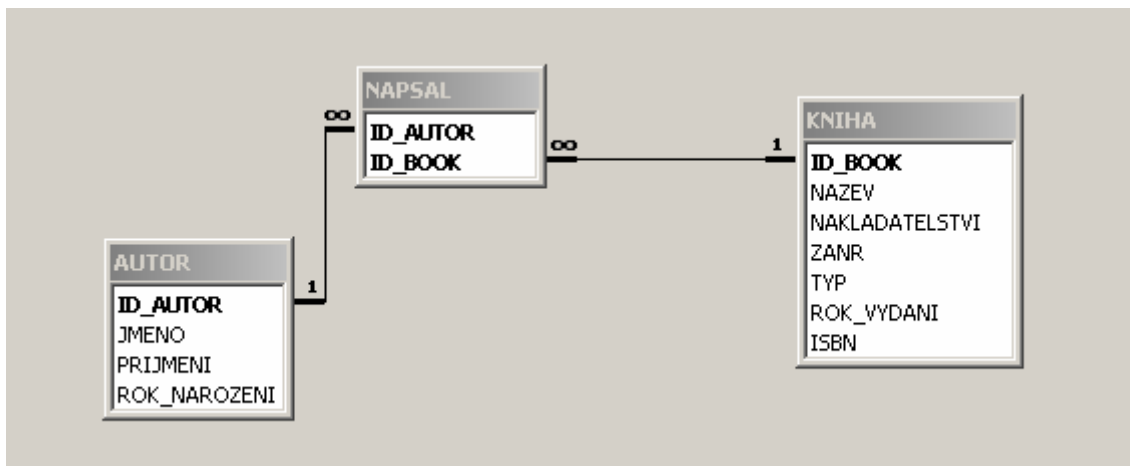
Tabulka AUTOR slouží pro evidenci autorů knih, které jsou v knihovním fondu. V této tabulce se eviduje pouze jméno, příjmení a rok narození autora. Primárním klíčem je opět identifikační číslo. Rok narození autora evidujeme hlavně proto, že v případě, že by se objevil druhý autor se stejným jménem (např. syn).



Obrázek 8: Vazba tabulky AUTOR

4.3.5 Tabulka NAPSAL

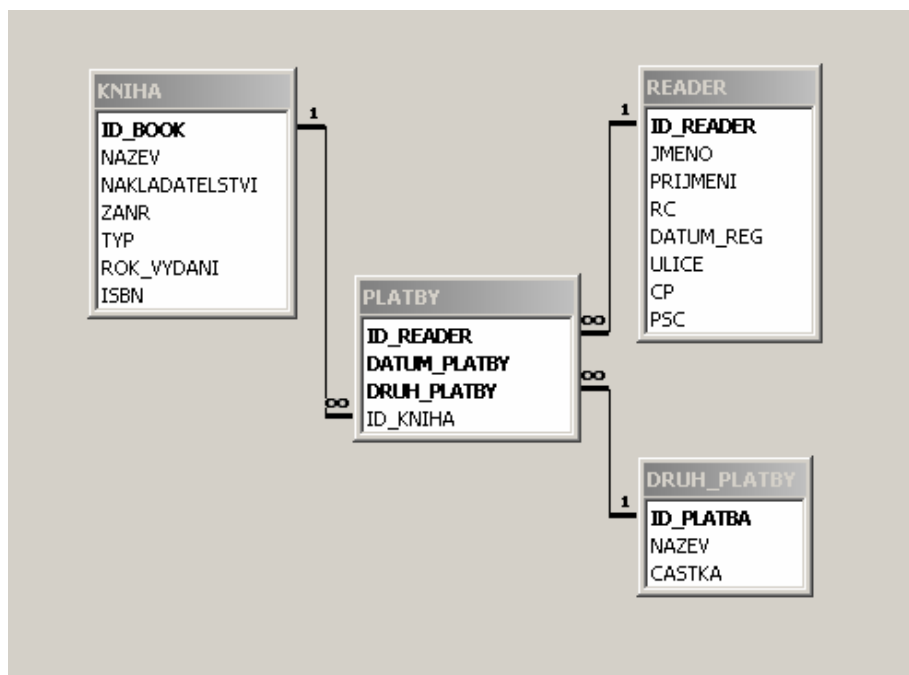
Tabulka NAPSAL je výsledkem dekompozice. Obsahuje identifikační číslo knihy a autora. Tato dekompozice zabraňuje, aby nastal vtaž N:M integritního omezení. Ten by mohl nastat v případě, že by knihu napsalo (nebo by na knize spolupracovali) více autorů. Obsahuje tedy vazbu na tabulky AUTOR a KNIHA. Primární klíč jsem v této tabulce použít složený z dvou použitých polí.



Obrázek 9: Vazby tabulky NAPSAL

4.3.6 Tabulka PLATBY

Tabulka PLATBY slouží k evidenci plateb od čtenářů. Jedná se o platby za pozdní vrácení, za registrace a další poplatky. V této tabulce se eviduje datum platby, druh platby a kniha. Kniha se zapisuje pouze v případě, že byla kniha vrácena. Druh platby je opět datový typ číslo. Jedná se o vazbu na tabulku DRUH_PLATBY. V této tabulce jsem opět použil složený primární klíč.



Obrázek 10: Vazby tabulky PLATBY

4.3.7 Tabulky ZANR, TYP, NAKLADATELSTVÍ

V případě tabulek ZANR, TYP a NAKLADATELSTVÍ se jedná o číselníky připojené k datové tabulce KNIHA. U každé z těchto tabulek je evidováno ID_*NAZEV TABULKY* (id_zanr, id_typ a id_nak) a dále je obsažen název, či typ dané kategorie. Identifikační číslo v každé tabulce je zároveň primárním klíčem.

Tabulky ZANR a TYP jsou předem již předvyplněné dle potřeb knihovny. Obsahy těchto tabulek je možno vidět v následujících tabulkách.

	ID typu	Název
▶	1	Kniha
	2	CD
	4	Časopis

Obrázek 11: Tabulka TYP

	ID žánru	Žánr
▶	1	Beletrie
	2	Biografie
	3	Cestování
	4	Chovatelství
	5	Detektivka
	6	Dobrodružná
	7	Encyklopedie
	8	Fauna
	9	Flora
	10	Historická
	11	Horror
	12	Hry
	13	Jazyk
	14	Kuchařka
	15	Mluvené slovo
	16	Román
	17	Sci-fi

Obrázek 12: Tabulka ZANR

Tabulku NAKLADATELSTVI si bude muset vyplnit pracovník knihovny podle potřeb.

4.3.8 Tabulka DRUH_PLATBY

Tabulka DRUH_PLATBY slouží jako číselník k poli DRUH_PLATBY v tabulce PLATBY. Tabulka obsahuje 3 pole. Jedná se o identifikační číslo, název (druh) platby a částku za daný druh platby.

	ID_PLATBA	Druh platby	Částka
+	1	Registrace dospělí	30
+	2	Registrace student	10
+	3	Registrace důchodce	20
+	4	Pozdní vrácení 1	10
+	5	Pozdní vrácení 2	20
+	6	Pozdní vrácení 3	40
+	7	Ztráta nebo poškození	20

Obrázek 13: Tabulka DRUH_PLATBY

4.3.9 Tabulka PSC

Tabulka PSC obsahuje veškerá poštovní směrovací čísla používaná pro Českou a Slovenskou republiku a k nim i přiřazené patřičné obce a města. Data jsou importována z textového souboru.

4.4 Vstupní masky

V této části přiblížím speciální vstupní masky, které jsem použil u některých polí datových tabulek. Jedná se o vzorová data, která budou vložena do tohoto pole.

4.4.1 Datum

Formát data jsem zvolil z nabídky jako Krátké (např. 17.5.2010). Vstupní maska k tomuto datu byla zvolena 90.90.0000. Takto zvolená maska při zadávání datumu do tabulky přednastaví pole na __.__.____. V případě, že číslo měsíce (dne) je pouze jednociferné je nutno zadávat dvojciferné číslo (01, 02, ...). Zároveň zabraňuje, případným nesrovnalostem v zadávání dat např. jednou bude měsíc zvolen pomocí římských číslic jednou pomocí arabských a do třetice slovně. Tato maska je nastavena u všech používaných datumů v databázi.

4.4.2 Rodné číslo

Vstupní maska pro rodné číslo je nastavena na 000000"/"0000. V tomto případě slouží hlavně proto, aby se zabránilo zbytečnému vyhledávání znaku / na klávesnici zaměstnanci knihovny. Číslice jsou opět nahrazeny znakem _ při zadávání záznamů do tabulky. Velikost pole je nastavena na 11, tzn. 10 číslic a 1 znaménko lomítka.

4.4.3 Jméno a příjmení

U jména a příjmení používám masku >L<?????????????. Nastavuje první písmeno zadávaného řetězce znaků na velké. Znak ? zde zastupuje jakékoliv písmeno abecedy nebo jiný libovolný znak. Počet znaků ? je odvozen podle velikosti datového typu. U jména jsem zvolil délku 15 znaků, u příjmení 20.

4.5 Dotazy

Pro efektivní využití databáze jsem vytvořil dotazy, které jsou využity v hlavním formuláři, kde v jednotlivých kartách jsou použity ve spodní části, která slouží pro zobrazení dat k danému tématu karty. Některé dotazy byly vytvořeny pomocí jazyka SQL, některé byly vytvářeny až voláním funkce pomocí jazyka VBA.

4.5.1 Dotazy pro hlavní formulář

Dotazy pro hlavní formulář se používají ve spodní části karet. Jedná se o dotazy pro karty Platby, Tituly, Čtenáři a Výpůjčky. Pro kartu archiv jsem dotaz nevytvářel, zde jsou data vypsána ve tvaru v jakém se nacházejí v tabulce Archiv. Ostatní dotazy slouží pro lepší prohlížení tabulek. V normálním zobrazení by zaměstnanec viděl pouze čísla, která by mu nic neříkala. Díky těmto dotazům jsou čísla nahrazena např. jménem a příjmením nebo názvem knihy apod.

Pro příklad uvádím dotaz pro kartu Platby. Vznikl spojením tabulek Platby, Reader, Druh_platby. Dotaz byl vytvořen pomocí jazyka SQL.


```

select p.ID_READER, r.JMENO, r.PRIJMENI, p.DATUM_PLATBY,
d.NAZEV, d.CASTKA, p.ID_KNIHA
from READER r, PLATBY p, DRUH_PLATBY d
where p.ID_READER=r.ID_READER
and p.DRUH_PLATBY=d.ID_PLATBA;

```

Ostatní dotazy vložené do hlavního formuláře jsou na stejném principu. Vždy vznikají spojením tabulek propojených pomocí identifikačních čísel.

dot_platby : Výběrový dotaz							
ID_READER	Jméno	Příjmení	DATUM_PLATBY	Druh platby	Částka	KNiha	
1	Daniel	Vašíček	10.4 .2009	Registrace student	10		
3	Karel	Vomáčka	10.5 .2010	Registrace dospělí	30		
4	Josef	Novák	12.5 .2010	Registrace dospělí	30		
4	Josef	Novák	12.5 .2010	Pozdní vrácení 1	10		7

Obrázek 14: Náhled do tabulky dotazu Archiv

4.5.2 Dotazy vytvářené pomocí jazyka VBA

Tyto dotazy jsou zobrazovány po zavolání příslušné funkce pomocí příkazových tlačítek. Jedná se o funkce, které třídí data podle určitého identifikátoru, který zadává zaměstnanec knihovny. Těmito dotazy jsou přehled vypůjčených knih a přehled plateb čtenáře, kde jsou data tříděna podle identifikačního čísla čtenáře. Další dotazy vytvářené podobným způsobem je historie výpůjček knihy a čtenáře. Zde je zobrazeno kdo a kdy měl danou knihu vypůjčenou anebo kdy a jaké knihy měl čtenář vypůjčené.

Pro příklad uvedu vytvoření dotazu pro přehled výpůjček čtenáře. Po zadání identifikačního čísla čtenáře se toto id načte do proměnné.

```
prom_id_reader = Text177.Value
```

Dále probíhá načtení SQL příkazu do proměnné typu řetězec. Proměnná pro zápis příkazu se jmenuje strSQL. Řetězec příkazu zapisuji postupně. Pro vytvoření dotazu jsem zvolil příkaz SELECT INTO. Pomocí tohoto příkazu vyberu data a vložím je do nově vytvořené tabulky, kterou pojmenuji dot_vypujcky_reader. V posledním řádku příkládám do řetězce proměnnou s identifikačním číslem čtenáře, díky které budou následně vybrána pouze data čtenáře, kterého si vybral zaměstnanec knihovny.

```

strSQL = "SELECT AUTOR.JMENO, AUTOR.PRIJMENI,
VYPUJCKY.ID_BOOK, KNIHA.NAZEV, VYPUJCKY.DATUM_VYPUJCKY "

strSQL = strSQL & "INTO dot_vypujcky_reader "
strSQL = strSQL & " FROM VYPUJCKY, KNIHA, READER, AUTOR,
NAPSAL "

strSQL = strSQL & "WHERE VYPUJCKY.ID_BOOK = KNIHA.ID_BOOK "
strSQL = strSQL & "AND VYPUJCKY.ID_READER =
READER.ID_READER "

strSQL = strSQL & "AND NAPSAL.ID_BOOK=VYPUJCKY.ID_BOOK "
strSQL = strSQL & "AND NAPSAL.ID_AUTOR=AUTOR.ID_AUTOR "
strSQL = strSQL & "AND VYPUJCKY.ID_READER = "
strSQL = strSQL & prom_id_reader & ";"

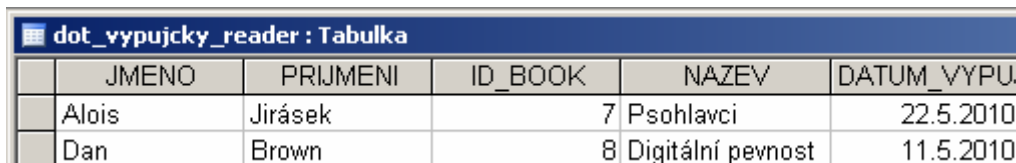
```

Po vytvoření SQL příkazu se spouští tento příkaz pomocí metody RunSQL. Před tím je nutno nastavit na hodnotu False metodu SetWarnings, která spouští upozornění na spuštění SQL dotazu a pro pracovníka by byla akorát matoucí.

```

DoCmd.SetWarnings False
DoCmd.RunSQL strSQL
DoCmd.SetWarnings True

```



	JMENO	PRIJMENI	ID_BOOK	NAZEV	DATUM_VYPUJCKY
	Alois	Jirásek	7	Psohlavci	22.5.2010
	Dan	Brown	8	Digitální pevnost	11.5.2010

Obrázek 15: Dotaz Přehled výpůjček vybraného čtenáře

4.5.3 Pomocné dotazy

Tyto dotazy slouží pouze jako pomocná část při provádění funkcí vytvořených pomocí jazyka VBA. V této databázi je pouze jediný pomocný dotaz a to na volné knihy, nebo-li knihy, které jsou k dispozici v knihovně k vypůjčení. Dotaz je využíván při vytváření výpůjčky ke kontrole, zda je možno knihu vypůjčit nebo ne. Mohlo by se stát, že by zaměstnanec knihovny omylem zapomněl vrátit knihu přes IS a vznikaly by duplicitní záznamy v tabulce výpůjček. Tento dotaz vznikl podle postupu, jak je

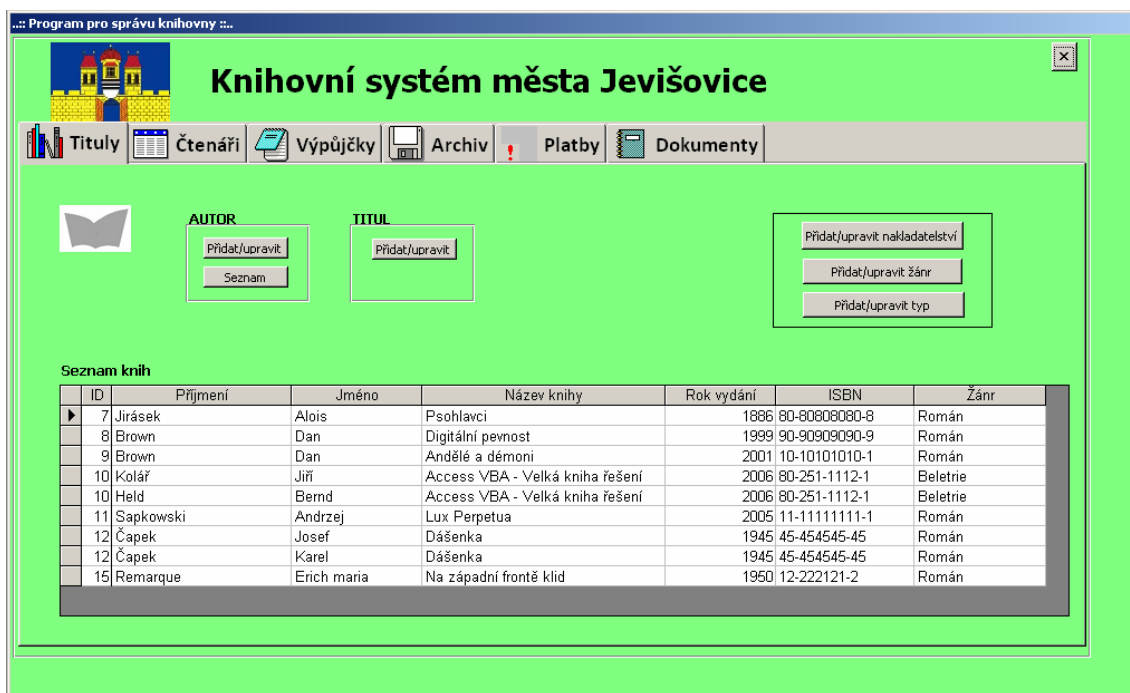
popsáno v kapitole 4.5.1, navíc se zde ještě navíc kontroluje, zda existuje záznam v tabulce Výpůjček.

4.6 Hlavní formulář

Nejdůležitějším formulářem databáze je formulář Hlavní. Tento formulář je spouštěn automaticky po spuštění aplikace MS Access. Jedná se o hlavní ovládací prostředek pro komunikaci s databází knihovny. Přes tento formulář má zaměstnanec knihovny možnost pracovat s databází automaticky bez jakékoliv znalosti MS Access. Tento formulář zajišťuje veškeré funkce potřebné pro vedení knihovny (tvorbu výpůjček, vrácení knih a vedení archivu vypůjčených knih, výpočet a evidenci plateb čtenářů). Zaměstnanec knihovny tedy ovládá celou databázi díky naprogramovaným tlačítkům.

4.6.1 Vzhled

Formulář je složen z 5-ti karet, kde každá karta zastupuje jednu z potřebných funkcí tohoto systému. Karta Tituly slouží pro správu knih, časopisů a CD-ROM, které je si možno vypůjčit. Karta Čtenáři slouží ke správě čtenářů knihovny. Karta Výpůjčky a Archiv k evidenci, tvorbě a archivaci vypůjčených knih. Přes kartu Platby má zaměstnanec možnost provádět a prohlížet platby čtenářů za pozdní vrácení knih nebo za registraci čtenářů. Vytvořil jsem také kartu Dokumenty, kde má zaměstnanec knihovny možnost prohlížet, či vytvářet potřebné tiskové sestavy a je zde také možnost nahlížení do různých statistik. Dále je součástí tohoto formuláře tlačítko pro ukončení aplikace.



Obrázek 16: Hlavní formulář

Všechny karty jsou složeny ze dvou částí. Ve spodní části je tabulka (dotaz) s příslušnými hodnotami z dané sekce a v horní části jsou umístěny příkazová tlačítka pro ovládání příslušných tabulek nebo plnění potřebnou funkci v dané sekci.

4.6.2 Funkce karty Tituly

Na kartě Tituly jsou umístěny příkazová tlačítka pro přidávání a úpravu číselníků nakladatelství, žánr a typ. Dále jsou zde obsaženy tlačítka pro přidání a úpravu nových titulů a autorů, je zde také tlačítko pro vypsání seznamu autorů. Rozmístění těchto tlačítek lze vidět na obrázku.



Obrázek 17: Ovládací část karty Tituly

Tlačítko Přidat/upravit v části ohraničené jako Autor slouží pro otevření formuláře sloužící pro editaci současných autorů nebo přidání nového autora. Tlačítko Seznam otevře dotaz s výpisem z tabulky Autor.



The image shows a Windows-style dialog box with a blue title bar containing the text "...: Přidání nového autora :...". The main area has a green background and is titled "Prohlížení, úprava a přidání autorů". Inside, there is a white-bordered form with four input fields: "ID autora" with the value "1", "Jméno" with "Alois", "Příjmení" with "Jirásek", and "Rok narození" with "1851". Below the form are two arrow buttons (left and right) and two buttons labeled "NOVÝ" and "KONEC".

Obrázek 18: Formulář pro přidání nového autora

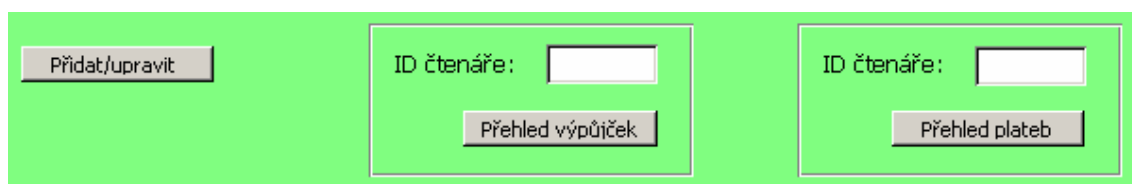
Formulář pro přidání nového autora je totožný s formuláři, které otvírají další tlačítka v této kartě označená *Přidat/upravit navez_tabulky*. Jsou zde tlačítka pro postupné přecházení mezi záznamy po jednom, dále tlačítko pro přejítí na přidání nového záznamu a tlačítko pro uzavření formuláře. Ukládání nových záznamů se provádí automaticky.

V případě formuláře pro přidání nového titulu, je zde navíc umístěno tlačítko pro uložení. Zde je nutné provádět uložení ještě do tabulky Napsal, kde je evidováno identifikační číslo titulu a autora(ů).

Propojení číselníků se ve formulářích projevuje tak, že zaměstnanec knihovny bude moci pracovat jako ComboBox (pole se seznamem). Tím se ulehčí práce s hledáním potřebného identifikačního čísla a na druhou stranu pokud zaměstnanec si bude číslo potřebného záznamu pamatovat stačí zapsat číslo.

4.6.3 Funkce karty Čtenáři

Karta Čtenáři slouží pro evidenci nových čtenářů. Proto je v ovládací části obsaženo tlačítko Přidat/upravit. Dále jsou zde dvě pole pro vložení identifikačního čísla čtenáře a po stisknutí tlačítka pod tímto polem, se vytvoří pomocí SQL příkazu tabulka buď s veškerými výpůjčkami tohoto čtenáře nebo přehled plateb čtenáře.

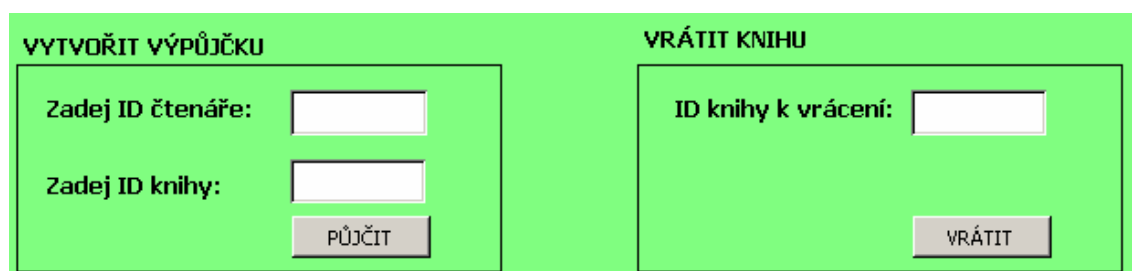


Obrázek 19: Ovládací část karty Čtenáři

Důležitou funkcí formuláře pro přidání nového čtenáře je, že se po registraci čtenáře musí zaplatit ještě registrační poplatek. Po uložení nového čtenáře se tedy předvyplní potřebné údaje v kartě Platby. Tím je opět ulehčena práce zaměstnanci knihovny.

4.6.4 Funkce karty Výpůjčky

Nejdůležitější ovládací prvky navržené aplikace je obsažena v kartě Výpůjčky. Horní ovládací část je složena ze dvou textových polí a příkazového tlačítka sloužícího pro tvorbu výpůjček a jednoho textového pole a příkazového tlačítka pro vrácení knihy.



Obrázek 20: Ovládací část karty Výpůjčky

Pokud probíhá vrácení knihy, zadá zaměstnanec knihovny v pravé části identifikační číslo knihy a stiskne tlačítko Vrátit. Po stisknutí tlačítka proběhne

kontrola, zda byla kniha vrácena včas. Dále proběhne vymazání záznamu z tabulky Výpůjčky a provede se zápis do tabulky Archiv. Kontrola včasného vrácení probíhá tak, že se nejdříve načte datum, kdy byla kniha půjčena a datum kdy je vrácena do proměnných. Následně je rozdíl mezi těmito daty přepočítán na počet dní a probíhá kontrola tohoto čísla s hodnotami pro pozdní vrácení. Pokud byla kniha vrácena pozdě vybere se ta varianta, do které spadá pozdní vrácení a jsou předvyplněny údaje v kartě Platby a na obrazovce MsgBox s upozorněním o pozdním vrácení.

Pro vytvoření výpůjčky je nutné zadat identifikační číslo čtenáře a knihy. Následuje kontrola existence těchto hodnot v příslušných tabulkách. Následuje kontrola, zda na knihu již neexistuje výpůjčka. Vzhledem k tomu, že si knihu vybírá sám čtenář, tak na knihu fyzicky výpůjčka neexistuje, problém by mohl nastat v případě, že by se např. knihovník spletl v zadávání hodnot nebo by mohl zapomenout knihu vrátit přes IS. Dále se vytvoří SQL dotaz pro vytvoření záznamu v tabulce Výpůjček. Před spuštěním vytvořeného SQL dotazu se provede ještě kontrola, zda má čtenář zaplacený registrační poplatek pro současný kalendářní rok. V případě, že ne, vyskočí opět zaměstnanci upozornění a předvyplní se údaje v kartě Platby. Nakonec proběhne zápis do tabulky Výpůjčky.

4.6.5 Funkce karty Platby

Ovládací část funkce platby je složena z textového pole pro zadání identifikačního čísla čtenáře, který má platit poplatek. Dále jsou zde dvě textová pole, která jsou uzamčená pro zápis, která se automaticky vyplní jménem a příjmením čtenáře po zadání identifikačního čísla. Textové pole ID knihy je také zamčeno proti zápisu a je automaticky vyplňováno pokud má čtenář pozdě vrácenou knihu. ID čtenáře bývá také automaticky vyplňováno, pokud má platit poplatek, ať za pozdní vrácení nebo registrační poplatek.

Obrázek 21: Ovládací část karty Platby

Dále tato část obsahuje tři zaškrťovací políčka (Registrace, Pozdní vrácení, Ztráta/poškození čtenářského průkazu). Jedná se o tři druhy plateb, které mohou nastat. Po zaškrtnutí tlačítek Registrace a Pozdní vrácení se zobrazí další možnosti. Zde zaměstnanec vybere možnost pomocí tlačítek `OptionButton`. V případě, že se jedná o pozdní platbu, je předvyplněná i možnost i lhůta, ve které byla kniha vrácena.

Tlačítko Spočítat vyplní částkami všechny tři textová pole v levé části, která jsou uzamčena pro zápis uživatele a do textového pole Celkem vypíše celkovou částku, kterou by měl čtenář zaplatit.

Tlačítko Zapsat vytvoří záznamy v tabulce Platby. Pokud by měl čtenář platit poplatek za více položek, provede se zápisů do tabulky více. Zápis se opět provádí pomocí jazyka SQL.

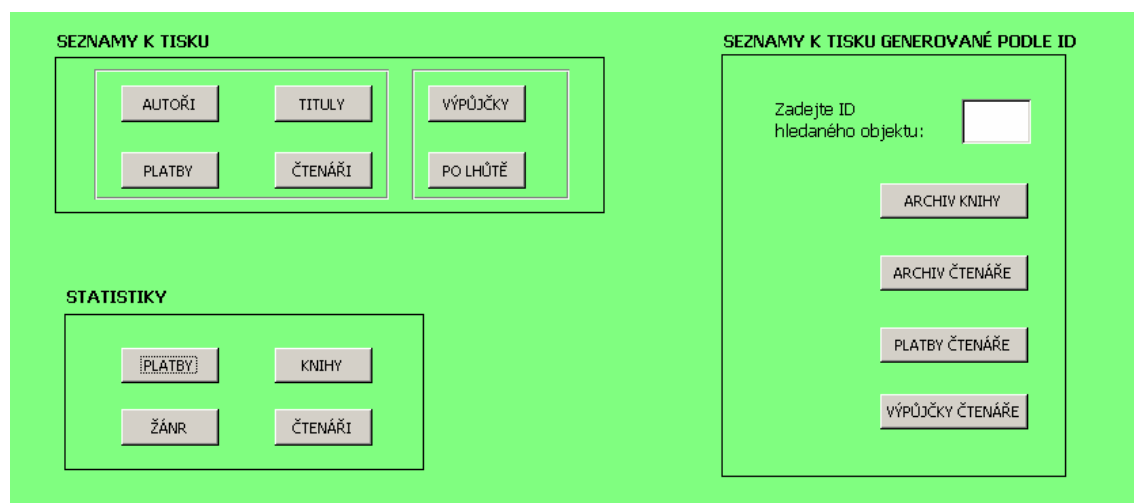
4.6.6 Karta Archiv

Karta Archiv slouží pro kontrolu vypůjčených knih, které již byly vráceny. Tato karta nemá žádné funkce, které by byly důležité pro provoz této aplikace. V horní ovládací části jsou pouze textová pole pro vložení identifikačního čísla čtenáře nebo knihy a po stisknutí příslušného příkazového tlačítka je otevřena tabulka s historií výpůjček vybraných dle čtenáře nebo knihy.

4.6.7 Karta Dokumenty

Tato karta na rozdíl od ostatních není složena ze dvou částí, ale ze tří. Na rozdíl od předchozích karet zde není ve spodní části vypsán dotazový seznam. Ve dvou částech je možno otevřít sestavy k tisku a ve třetí je možno otevřít grafy ukazující statistiky knihovny. Tato karta usnadňuje práci knihovníkovi, při sčítání poplatků,

pokud potřebuje vytisknout seznam knih, čtenářů. Pokud by čtenář chtěl vytisknout seznam knih, které má půjčené je to zde také možno provést.



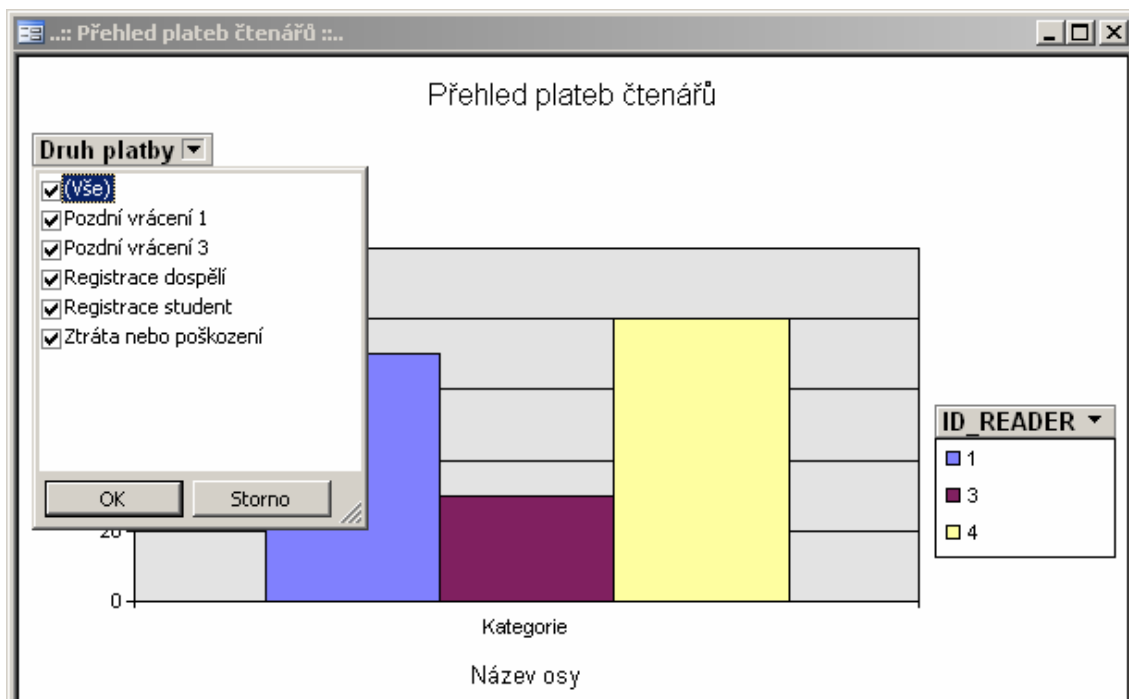
Obrázek 22: Funkce karty Dokumenty

V první části je možno otevření předdefinovaných sestav dle třídění podle hlavních tabulek. Otevření sestavy je lehce proveditelné pomocí příslušných příkazových tlačítek. V pravé části první tabulky jsou samostatně ohraničená tlačítka Výpůjčky a Po lhůtě. Tlačítko Po lhůtě generuje seznam čtenářů, kteří mají vypůjčené knihy a jsou již po lhůtě vrácení. Tato sestava se vytváří tak, že po stisku tlačítka se spustí funkce, která vytvoří tabulku, ze které je vytvořena automaticky sestava. V příloze 9 lze vidět zdrojový kód. Je nutné vždy vymazat data, která v této tabulce jsou z dřívějších sestav, dále se prochází každý záznam tabulky Výpůjčky a porovnává datum výpůjčky s aktuálním datem datem prohlížení. V případě nalezení pozdě vrácené knihy se vytvoří SQL dotaz a záznam se zapíše do tabulky.

V další části si může knihovník nechat vygenerovat seznam podle ID knihy nebo čtenáře. Do textového pole zadá požadované ID a stiskne příslušné tlačítko. Opět se provede vytvoření SQL příkazu, ten se následně spustí a vytvoří se automaticky sestava.

Dále je zde možnost statistik. V této části je možnost otevření kontingenčního grafu. Grafy pro knihy a čtenáře jsou přednastaveny na 5 nejčtenějších knih nebo nejčastěji si půjčujících čtenářů. Graf platby zobrazuje přehled plateb jednotlivých

čtenářů. Zde je možno nechat si graf předělat dle představ zaměstnance, např. pokud chce poplatky všechny, může nechat zatrhnutou možnost Vše, jak lze vidět na obrázku 23 nebo si vybrat jednu z nabízených možností a filtrovat si tuto možnost.



Obrázek 23: Graf plateb čtenářů

4.7 Umístění a zabezpečení databáze

Databáze bude umístěna na lokálním disku na počítači zaměstnance knihovny. Vzhledem k tomu, že není vyžadováno síťové sdílení dat, není nutno nastavovat tuto databázi pro práci více uživatelů.

Vzhledem k tomu, že se v databázi uchovávají některé citlivé údaje čtenářů je nastaveno před spuštěním databáze heslo. Bylo by žádoucí, aby nadále zaměstnanec dbal na bezpečnost tohoto hesla, prováděl jeho změnu v pravidelných intervalech, volil správnou délku a složitost.

Zároveň s nastavením hesla byl vytvořen účet pro zaměstnance knihovny, kterému je zamezeno, aby se mohl dostat k nastavení databáze. Databázi bude moci

otevírat pouze přes formuláře. Tím se zamezí tomu, aby neodborný personál prováděl zásahy do databáze a tím by mohlo dojít k jejímu porušení. U tohoto účtu jsou nastavena pouze oprávnění, která jsou potřebná pro chod informačního systému z uživatelského hlediska. Z hlediska úplného přístupu je vytvořen administrátorský účet, který je také zaheslován.

Dále by měl zaměstnanec knihovny provádět pravidelné zálohování. Vzhledem k tomu, že je knihovna otevřena pouze dva dny v týdnu, doporučil bych její zálohování každý pracovní den. Za vhodné považuji mít uloženou zálohu např. na CD-ROM nebo USB flash paměti, a toto médium uzamknout do bezpečné schránky.

5 Ekonomické zhodnocení a přínos řešení

Přínos návrhu nového IS pro místní knihovnu města Jevišovice spočívá ve zrychlení, zjednodušení a zkvalitnění služeb. Tyto přínosy jsou realizovány díky formuláři, který pracuje s databází.

Naprogramované funkce ve formuláři umožňují zaměstnanci knihovny lehce provádět výpůjčky, evidovat vrácení knihy a zaměstnanec má možnost lehce spočítat platby za pozdě vrácené knihy nebo registrační poplatky. Díky tomu je dosaženo i zkvalitnění služeb, zaměstnanec bude moci rychleji obsluhovat další čtenáře, případně zákazníky informačního centra.

Práce se díky novému IS stane pro zaměstnance přehlednější a rychlejší. Bude moci rychleji dohledat, kdo má knihu vypůjčenou. Zda je kniha vypůjčena po lhůtě vrácení. Pracovník má možnost prohlížet seznam výpůjček, seznamy čtenářů a knih přehledně a rychle přímo v hlavním formuláři.

Finanční přínos pro knihovnu je nulový. Vzhledem k tomu, že je knihovna neziskovou organizací není nutné, aby ze systému plynuly zisky, ale je nutné, aby plnil svou základní úlohu.

Náklady na vytvoření IS jsou pro místní knihovnu města Jevišovice nulové. Jediným nákladem na pořízení bylo zakoupení balíčku MS Office. Ten byl, ale zakoupen již v minulosti, takže v této chvíli se s tímto výdajem nepočítá. Jediný výdaj, který by se zde mohl objevit, by bylo zakoupení novější verze MS Office 2007. Zde bych, ale doporučoval raději investici Města Jevišovice do licence pro více počítačů pro celý městský úřad včetně místní knihovny.

Závěr

Tato bakalářská práce pojednává o navrhnutí IS pro místní knihovnu města Jevišovice. Součástí návrhu je vytvoření elektronické databáze a následně vytvoření rozhraní pro práci s touto databází. Nově navržený informační systém umožňuje vytvářet výpůjčky, po vrácení knihy ji zaevidovat do archivu vypůjčených knih.

V kapitole Teoretická východiska jsem se zaměřil na vysvětlení problematiky související s touto prací. Jednalo se o vysvětlení pojmu informační systém, dále osvětlení problematiky ohledně databází, nutno bylo také přiblížit informace o programu MS Access a jazyku VBA. V další kapitole Analýza problému a současného stavu, kde jsem představil místní knihovnu města Jevišovice, dělení knihoven v rámci celé České Republiky a financování knihovny. Dále jsem v této kapitole představil problém související s další kapitolou a to Praktické řešení situace.

V praktické části této práce byla vytvořena databáze v aplikaci MS Access, která je součástí kancelářského balíčku MS Office 2003. MS Access byl zvolen z důvodu nízkých finančních prostředků, minimálních znalostí pracovníka práce s informačními technologiemi. Tím, že jsem zvolil MS Access a vypracoval rozhraní pomocí jazyka VBA, se staly náklady na pořízení tohoto informačního systému nulové. Dále zde považuji za velké plus, že v případě, kdy pochybí nějaká z funkcí, která se v současné době v systému nenachází, lze snadno doprogramovat, či vytvořit.

Databáze je složena z malého počtu tabulek, ovšem dle průzkumů z předchozích let by se mohlo stát, že by mohla databáze nabýt většího počtu záznamů, proto se u čtenářů evidují pouze nejnutnější údaje, u knih se neevidují různé textové popisy apod. Návrh nové databáze se tedy přizpůsobil současnému stavu, s tím rozdílem, že dochází k digitalizaci záznamů.

Pro snadné a intuitivní ovládání databáze bylo vytvořeno ovládací rozhraní pomocí formulářů a jazyka VBA. Díky tomu rozhraní je možno provádět všechny důležité procesy pomocí ovládacích tlačítek bez nutnosti znalosti samotného programu MS Access. Zaměstnanci knihovny je zamezeno, aby zasahoval do návrhu a rozložení databáze. Tím se také zamezí případné kolizi.

V případě tohoto informačního systému není podmínkou, že může být provozován pouze v místní knihovně města Jevišovice. Město Jevišovice může bez problému tento IS poskytnout i dále dalším obcím např. spadajícím do DSO Jevišovicka.

Seznam literatury

Knihy:

- 1) BASL, J. *Podnikové informační systémy. Podnik v informační společnosti*. Praha: Grada, 2002. 142 s. ISBN 80-247-0214-2.
- 2) HELD, B. *Access VBA: Velká kniha řešení*. Brno: Computer Press, 2006. Vyd. 1. 639 s. ISBN 80-251-1112-1.
- 3) HERNANDEZ, M. J. *Návrh databází*. Praha: Grada, 2006. 1. vyd. 408 s. ISBN 80-247-0900-7.
- 4) ŘEPA, V. *Analýza a návrh informačních systémů*. Praha: Ekopress, 1999. 403 s. ISBN 80-86119-13-0.
- 5) KOCH, M. *Datové a funkční modelování*. Brno: VUT FP, 2008. 121 s. ISBN 978-80-214-3731-9.
- 6) KOCH, M. *Informační systémy a technologie*. Brno: VUT FP, 2001. 103 s. ISBN 80-214-1970-9.
- 7) MORKEŠ, D. *Microsoft Access 2003: Podrobná uživatelská příručka*. Brno: Computer Press, 2004. 350 s. ISBN 80-251-0179-7.
- 8) MORKEŠ, D. *Učebnice Visual Basic 6.0*. Praha: Computer Press, 2006. 165 s. ISBN 80-7226-312-9.

Elektronické zdroje:

- 9) ZELENKA, P. *Vícejazyčné webové aplikace a relační databáze* [online]. 2005 , 23.8.2005 [cit. 2009-12-15]. Dostupný z WWW: <<http://interval.cz/clanky/vicejazycne-webove-aplikace-a-relacni-databaze/>>.
- 10) Objektivně orientované databázové systémy [online]. 2009 [cit. 2010-01-19]. Dostupný z WWW: <<http://www.managed-dedicated-serverny.net/objektive-orientovane-databazove-systemy.html>>.
- 11) Vyvíjíme databázový a informační systém IV. [online]. 2004 , 26.5.2004 [cit. 2010-01-19]. Dostupný z WWW: <<http://www.dbsvet.cz/view.php?cisloclanku=2004052601>>.

- 12) *Informační systém*. Wikipedia, otevřená encyklopedie [online]. 2001, 16.12.2009 v 11:29 [cit. 2010-01-19]. Dostupný z WWW:
<http://cs.wikipedia.org/wiki/Informa%C4%8Dn%C3%AD_syst%C3%A9m>.
- 13) *SWOT*. Wikipedie, otevřená encyklopedie [online]. 3.4.2010 [cit. 2010-04-10]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/SWOT>>.
- 14) Národní knihovna ČR. *Co to je knihovna – Systém knihoven v ČR* [online]. 2008 [cit. 2010-04-10]. Knihovny.cz. Dostupné z WWW:
<<http://www.knihovny.cz/co-to-je-knihovna/system-knihoven-v-cr>>.
- 15) Národní knihovna ČR. *Co to je knihovna – Financování*. [online]. 2008 [cit. 2010-04-10]. Knihovny.cz Dostupné z WWW:
<<http://www.knihovny.cz/co-to-je-knihovna/financovani>>.
- 16) Hierarchický databázový model [online]. 2009 [cit. 2010-01-19]. Dostupný z WWW:<<http://www.managed-dedicated-serverny.net/hierarchicky-databazovy-model.html>>
- 17) Síťový databázový model [online]. 2009 [cit. 2010-01-19]. Dostupný z WWW:
<<http://www.managed-dedicated-serverny.net/sitovy-databazovy-model.html>>

Seznam obrázků

Obrázek 1: Hierarchická struktura (Zdroj: 16)	13
Obrázek 2: Síťový databázový model (Zdroj: 17)	14
Obrázek 3: Dekompozice vztahu N:M (Zdroj: Vlastní tvorba)	19
Obrázek 4: MsgBox	26
Obrázek 5: Provázání knihy s číselníky	41
Obrázek 6: Vazby čtenáře na další tabulky	41
Obrázek 7: Vazby tabulek ARCHIV a VYPUJCKY	42
Obrázek 8: Vazba tabulky AUTOR	43
Obrázek 9: Vazby tabulky NAPSAL	44
Obrázek 10: Vazby tabulky PLATBY	44
Obrázek 11: Tabulka TYP	45
Obrázek 12: Tabulka ZANR	45
Obrázek 13: Tabulka DRUH_PLATBY	46
Obrázek 14: Náhled do tabulky dotazu Archiv	48
Obrázek 15: Dotaz Přehled výpůjček vybraného čtenáře	49
Obrázek 16: Hlavní formulář	51
Obrázek 17: Ovládací část karty Tituly	51
Obrázek 18: Formulář pro přidání nového autora	52
Obrázek 19: Ovládací část karty Čtenáři	53
Obrázek 20: Ovládací část karty Výpůjčky	53
Obrázek 21: Ovládací část karty Platby	55
Obrázek 22: Funkce karty Dokumenty	56
Obrázek 23: Graf plateb čtenářů	57

Seznam zkratk

DSO – Dobrovolný svazek obcí

IS – Informační systém

MS – Microsoft

SQL – Structured Query Language

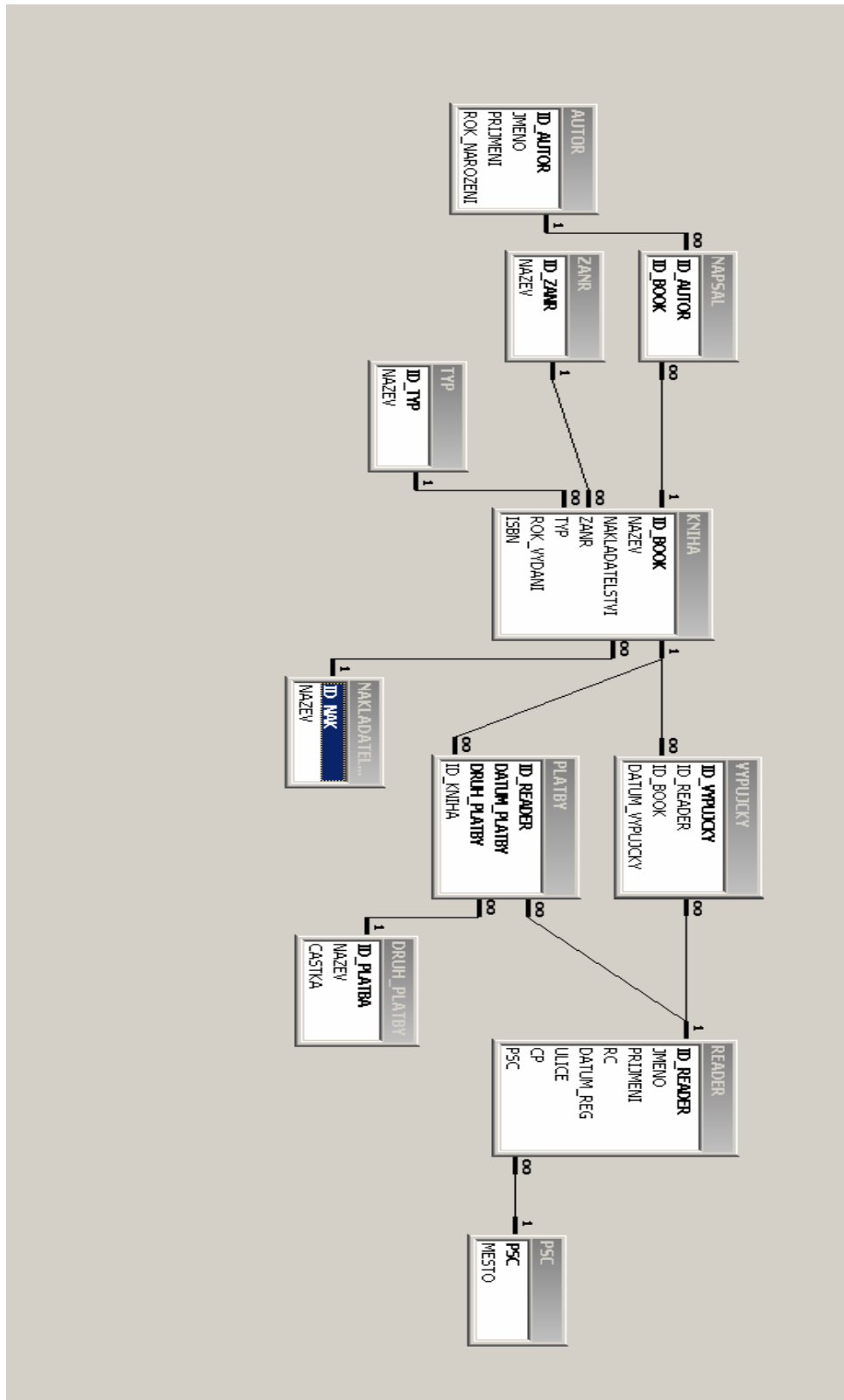
VBA – Visual Basic for Applications

Přílohy

Seznam příloh

Příloha 1: Logická struktura databáze	67
Příloha 2: Nastavení přepínání tlačítka OptionButton	68
Příloha 3: Aktualizace hodnot v textových polích po zadání ID čtenáře.....	68
Příloha 4: Funkce tlačítka SPOČÍTAT	69
Příloha 5: Otevření formuláře pro přidání nového čtenáře	70
Příloha 6: Kontrola pozdního vrácení knihy	71
Příloha 7: Kontrola zda byl zaplacen registrační poplatek	72
Příloha 8: Tvorba SQL příkazu pro vložení záznamu do tabulky Platby	73
Příloha 9: Funkce tlačítka PO LHŮTĚ	74

Příloha 1: Logická struktura databáze



Příloha 2: Nastavení přepínání tlačítka OptionButton

```
Private Sub Volba103_Click()  
    Volba103 = True  
    Volba105 = False  
    Volba107 = False  
End Sub
```

```
Private Sub Volba105_Click()  
    Volba103 = False  
    Volba105 = True  
    Volba107 = False  
End Sub
```

```
Private Sub Volba107_Click()  
    Volba103 = False  
    Volba105 = False  
    Volba107 = True  
End Sub
```

Příloha 3: Aktualizace hodnot v textových polích po zadání ID čtenáře

```
Private Sub Text87_AfterUpdate()  
    Set dbs = CurrentDb  
    Set radek = dbs.OpenRecordset("select * from READER")  
  
    prom_id_reader = Text87.Value  
    'nalezení údajů čtenáře v DB  
    Do While Not radek.EOF  
        If radek.Fields!ID_READER.Value = prom_id_reader Then  
            exist_reader = 1  
            JMENO = radek.Fields!JMENO.Value  
            PRIJMENI = radek.Fields!PRIJMENI.Value  
        End If  
        radek.MoveNext  
    Loop  
  
    Text94.Value = JMENO  
    Text96.Value = PRIJMENI  
End Sub
```

Příloha 4: Funkce tlačítka SPOČÍTAT

```
Private Sub Příkaz85_Click()  
Set dbs = CurrentDb  
Set radek = dbs.OpenRecordset("select * from DRUH_PLATBY")  
'přiřazení do proměnných hodnoty plateb  
Do While Not radek.EOF 'procházení pr. radek až nakonec  
If radek.Fields!ID_platba.Value = 1 Then  
    castka1 = radek.Fields!castka.Value  
ElseIf radek.Fields!ID_platba.Value = 2 Then  
    castka2 = radek.Fields!castka.Value  
ElseIf radek.Fields!ID_platba.Value = 3 Then  
    castka3 = radek.Fields!castka.Value  
ElseIf radek.Fields!ID_platba.Value = 4 Then  
    castka4 = radek.Fields!castka.Value  
ElseIf radek.Fields!ID_platba.Value = 5 Then  
    castka5 = radek.Fields!castka.Value  
ElseIf radek.Fields!ID_platba.Value = 6 Then  
    castka6 = radek.Fields!castka.Value  
ElseIf radek.Fields!ID_platba.Value = 7 Then  
    castka7 = radek.Fields!castka.Value  
End If  
radek.MoveNext  
Loop  
' přiřazení hodnoty penále podle OptionButton  
If Zaškrtnutí98 = True Then  
    Select Case Zaškrtnutí98 = True  
        Case Volba103.Value = True  
            Text125.Value = castka1  
        Case Volba105.Value = True  
            Text125.Value = castka2  
        Case Volba107.Value = True  
            Text125.Value = castka3  
    End Select  
Else  
    Text125.Value = 0  
End If  
' přiřazení hodnoty penále podle OptionButton  
If Zaškrtnutí100 = True Then  
    Select Case Zaškrtnutí100 = True  
        Case Volba114.Value = True  
            Text127.Value = castka4  
        Case Volba116.Value = True  
            Text127.Value = castka5  
        Case Volba118.Value = True  
            Text127.Value = castka6  
    End Select  
Else
```

```
Text127.Value = 0
End If

If Zaškrtnutí120 = True Then
    Text129.Value = castka7
Else
    Text129.Value = 0
End If

Text133.Value = Text125.Value + Text127.Value +
Text129.Value
End Sub
```

Příloha 5: Otevření formuláře pro přidání nového čtenáře

```
Private Sub Příkaz49_Click()
Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "READER_NEW"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
End Sub
```

Příloha 6: Kontrola pozdního vrácení knihy

```
'výpočet počtu dnů po vrácení
datum_vrac = Date
pocet_dnu = DateDiff("d", prom_datum_v, datum_vrac)
'kontrola pozdního vrácení knihy
  If pocet_dnu > 61 Then
    Text87.Value = prom_id_reader
    Zaškrtnutí100 = True
    Zaškrtnutí98 = False
    Pole111.Visible = True
    Volba114.Visible = True
    Volba116.Visible = True
    Volba118.Visible = True
    Popisek115.Visible = True
    Popisek117.Visible = True
    Popisek119.Visible = True
    Text94.Value = JMENO
    Text96.Value = PRIJMENI
    Text139.Value = prom_id_book
  If pocet_dnu > 74 Then
    If pocet_dnu > 105 Then
      Volba114 = False
      Volba116 = False
      Volba118 = True
      MsgBox ("Knihy byla pozdě vrácena")
    Else
      Volba114 = False
      Volba116 = True
      Volba118 = False
      MsgBox ("Knihy byla pozdě vrácena.")
    End If
  Else
    Volba114 = True
    Volba116 = False
    Volba118 = False
    MsgBox ("Knihy byla pozdě vrácena.")
  End If
  Else
    Zaškrtnutí100 = False
    Pole111.Visible = False
    Volba114.Visible = False
    Volba116.Visible = False
    Volba118.Visible = False
    Popisek115.Visible = False
    Popisek117.Visible = False
    Popisek119.Visible = False
  End If
```


Příloha 7: Kontrola zda byl zaplacen registrační poplatek

```
Set radek4 = dbs.OpenRecordset("select * from PLATBY")

' kontrola zda je zaplacen registrační poplatek
Do While Not radek4.EOF
    If radek4.Fields!ID_READER.Value = prom_id_reader Then
        exist_platba = 1
    ' načtení dat z řádku do proměnných
        prom_dat_pl = radek4.Fields!DATUM_PLATBY.Value
        prom_druh = radek4.Fields!DRUH_PLATBY.Value
    End If
    radek4.MoveNext
Loop

' oddělení roku od data
letos = Year(Date)
rok_pl = Year(prom_dat_pl)

If rok_pl < letos Then
    Text87.Value = prom_id_reader
    'zviditelnění OptionButtonu v kartě Plateb
        Zaškrtnutí98 = True
        Pole102.Visible = True
        Volba103.Visible = True
        Volba105.Visible = True
        Volba107.Visible = True
        Popisek104.Visible = True
        Popisek106.Visible = True
        Popisek108.Visible = True
        Text94.Value = JMENO
        Text96.Value = PRIJMENI

    MsgBox ("Nutno zaplatit registrační poplatek za rok " &
        letos)
End If
```

Příloha 8: Tvorba SQL příkazu pro vložení záznamu do tabulky Platby

```
If Zaškrtnutí98 = True Then
    Select Case Zaškrtnutí98
        Case Volba103.Value = True
            strSQL1 = "INSERT INTO PLATBY"
            strSQL1 = strSQL1 &
" (ID_READER,DATUM_PLATBY,DRUH_PLATBY,ID_KNIHA)VALUES ("
            strSQL1 = strSQL1 & prom_id_reader & ", "
            strSQL1 = strSQL1 & "'" & Date & "' , "
            strSQL1 = strSQL1 & "1, "
            strSQL1 = strSQL1 & "Null);"
            DoCmd.SetWarnings False
            DoCmd.RunSQL strSQL1
            DoCmd.SetWarnings True
        Case Volba105.Value = True
            strSQL1 = "INSERT INTO PLATBY"
            strSQL1 = strSQL1 &
" (ID_READER,DATUM_PLATBY,DRUH_PLATBY,ID_KNIHA)VALUES ("
            strSQL1 = strSQL1 & prom_id_reader & ", "
            strSQL1 = strSQL1 & "'" & Date & "' , "
            strSQL1 = strSQL1 & "2, "
            strSQL1 = strSQL1 & "Null);"
            DoCmd.SetWarnings False
            DoCmd.RunSQL strSQL1
            DoCmd.SetWarnings True
        Case Volba107.Value = True
            strSQL1 = "INSERT INTO PLATBY"
            strSQL1 = strSQL1 &
" (ID_READER,DATUM_PLATBY,DRUH_PLATBY,ID_KNIHA)VALUES ("
            strSQL1 = strSQL1 & prom_id_reader & ", "
            strSQL1 = strSQL1 & "'" & Date & "' , "
            strSQL1 = strSQL1 & "3, "
            strSQL1 = strSQL1 & "Null);"
            DoCmd.SetWarnings False
            DoCmd.RunSQL strSQL1
            DoCmd.SetWarnings True
    End Select
End If
```

Příloha 9: Funkce tlačítka PO LHŮTĚ

```
'vymázání záznamů z tabulky dot_vyp_polhute
strSQL1 = "delete * from dot_vyp_polhute"
DoCmd.SetWarnings False
DoCmd.RunSQL strSQL1
DoCmd.SetWarnings True

Do While Not radek.EOF
'nastavení datumů do proměnných
datum_vypujcky = radek.Fields!datum_vypujcky.Value
datum_vraceni = datum_vypujcky + 31
datum_ted = Date

'porovnání datumů
If datum_vraceni < datum_ted Then
    prom_id_reader = radek.Fields!ID_READER.Value
    prom_id_book = radek.Fields!ID_BOOK.Value
'nalezení potřebných dat
Do While Not radek2.EOF
    If radek3.Field!ID_BOOK.Value = prom_id_book Then
        prom_nazev = radek3.Fields!NAZEV.Value
    End If
    radek3.MoveNext
Loop

'zápis záznamů do tabulky
strSQL = "INSERT INTO dot_vyp_polhute"
strSQL = strSQL & "(JMENO,PRIJMENI,DATUM_VYPUJCKY,
ID_BOOK, NAZEV)VALUES ("
strSQL = strSQL & "'" & prom_name & "',"
strSQL = strSQL & "'" & prom_prijmeni & "',"
strSQL = strSQL & "'" & datum_vypujcky & "',"
strSQL = strSQL & prom_id_book & ","
strSQL = strSQL & "'" & prom_nazev & "'"
DoCmd.SetWarnings False
DoCmd.RunSQL strSQL
DoCmd.SetWarnings True
End If
radek.MoveNext
Loop

'otevření sestavy
Dim stDocName As String

    stDocName = "POLHUTE"
    DoCmd.OpenReport stDocName, acPreview
Exit Sub
```