

UNIVERZITA PARDUBICE
FAKULTA EKONOMICKO-SPRÁVNÍ

BAKALÁŘSKÁ PRÁCE

2008

Barbora STARÁ

Univerzita Pardubice
Fakulta ekonomicko-správní

Návrh a realizace knihovny

Barbora Stará

Bakalářská práce

2008

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: 2007/2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Barbora STARÁ**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Regionální a informační management**

Název tématu: **Návrh a realizace knihovny**

Z á s a d y p r o v y p r a c o v á n í :

1. Základní analýza problému
2. Návrh R-schématu
3. Realizace

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. BRÁZDA, J.: PHP 4 - učebnice základů jazyka, Grada Publishing, Praha 2002, 224 s., ISBN 80-247-0442-0

2. KOSEK J.: PHP - tvorba interaktivních internetových aplikací, Grada Publishing, Praha 1999, 492 s., ISBN 80-7169-373-1

Vedoucí bakalářské práce:

doc. Ing. Pavel Petr, Ph.D.

Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce:


1. října 2007

Termín odevzdání bakalářské práce:

19. května 2008


prof. Ing. Jan Čapek, CSc.
děkan

L.S.


doc. Ing. Pavel Petr, Ph.D.
vedoucí ústavu

V Pardubicích dne 30. října 2007

SOUHRN

Bakalářská práce je věnována návrhu a realizaci vyhledávání půjčených knih v databázi. Hlavním úkolem je vytvořit program na vyhledávání v hlavní databázi knihovny. Dále se v práci nachází teoretická část, jež se týká základních pojmů a tvorby databází. Praktická část se zabývá úpravou dat a následným vytvořením programu.

Klíčová slova

databáze, ER – diagram, PHP, vyhledávání

TITLE

Concept and realization of library

ABSTRACT

This bachelor thesis is focused on design and implementation of software for borrowed books database searching. The main task is to make the searching tool for the main library database. The thesis includes the theoretical part, where the fundamental terms and database making are explained. The practical part is dealing with data modification and programming of software.

Keywords:

database, ER – diagram, PHP, searching

OBSAH

Úvod.....	4
1 Programovací jazyk PHP	6
1.1 Seznámení s PHP	6
1.2 Potřebné nástroje.....	8
2 Teoretické poznatky.....	9
2.1 Databáze.....	9
2.2 Relační databáze	10
2.3 Tvorba tabulek	12
2.4 Normalizace dat	12
2.5 Relační schéma	17
2.6 Zvolené prostředky	19
3 Realizace	21
3.1 Analýza problému	21
3.2 Návrh databáze a tvorba ER – diagramu	22
3.3 Vysvětlení ER – diagramu	26
3.4 Instalace zvolených prostředků.....	29
3.5 Naplnění databáze daty	32
3.6 Programování	36
Závěr	44
Použitá literatura	45
Seznam obrázků.....	47
Seznam tabulek	47
Přílohy.....	48

ÚVOD

V dnešní době je lidská potřeba evidence a uchovávání dat stále větší a větší (ať už se jedná o evidenci firemních informací, obchodů přes internet, platebních karet nebo snad volání z mobilních telefonů). Každý člověk se dnes a denně setkává s nějakou databází, ale doby, kdy data byla uchovávána v podobě papírových kartoték nebo na dřevných štítcích, jsou téměř nenávratně pryč. S rychlým vývojem informačních technologií souvisí i vývoj databází, neboť právě informační technologie je k ukládání dat hojně využívána. Lze říci, že bez databází mají informační technologie pouze zlomkový význam. Právě na databázových systémech je postavena celá dnešní moderní společnost.

Rozvoj programovacích jazyků a internetu umožnil mimo jiné i vzdálený přístup do databází. Databáze tak může poskytovat informace většímu počtu uživatelů. Vyhledávání v takovéto databázi z pohodlí domova není žádnou novinkou. Dnes existuje celá řada databází. Namátkou je možné jmenovat například Microsoft Access, Firebird, Oracle, MySQL databáze a jiné.

Cílem bakalářské práce je návrh a realizace programu na vyhledávání v databázi knihovny. Předmětem zmiňované knihovny budou především takové knihy, které byly zakoupeny pro určitý univerzitní ústav z finančních prostředků grantů. Knihy jsou majetkem ústavu a v univerzitní knihovně se tedy neobjeví. Další stěžejní téma bakalářské práce je navrhnout databázi, z níž pak bude možno pomocí vytvořeného programu jednotlivé vypůjčené knihy vyhledávat. Bakalářská práce je rozvržena do třech částí.

Celá první část se zabývá stručným a bližším seznámením s programovacím jazykem PHP. Tento programovací jazyk bude použit v praktické části k řešení nastíněného problému vyhledávání vypůjčených knih z databáze.

Druhá část je věnována teoretickým poznatkům spojeným s problematikou databází. Jsou zde uvedeny a rovněž vysvětleny základní pojmy týkající se databází. Důležité jsou především pojmy ER - diagram, tvorba relací a jejich normalizace, neboť právě ony budou nadále uplatněny v praktické části při návrhu databáze, která je pro tvorbu programu na vyhledávání nezbytná. Na konci druhé části jsou uvedeny zvolené prostředky, jež byly k řešení problému použity.

Do poslední – třetí části spadá vlastní realizace projektu. Nachází se zde postup získání zvolených prostředků i úryvky samotného programování. Nejprve je zde graficky vyobrazen ER – diagram a s ním bezprostředně související tabulka entit a atributů. Nechybí tu ani textový popis ER – diagramu, ani popis jeho samotné, postupné tvorby. V neposlední řadě je zde uvedena normalizace relací, vztahujících se k tomuto tématu. Každý úryvek části programového skriptu zde uvedený, je patřičně okomentován z pohledu funkcí a účelu.

1 PROGRAMOVACÍ JAZYK PHP

1.1 Seznámení s PHP

Zkratka PHP znamená Personal Hypertext Preprocessor. [7] Jedná se o jednu z nejrozšířenějších technologií pro zajištění dynamiky na straně serveru. Provádění kódu právě na straně serveru s sebou přináší několik výhod. Mezi zmiňované výhody můžeme zahrnout například nenáročné požadavky na software na straně klienta, data nemusíme přenášet po internetu apod. [7]

PHP je volně šiřitelný software, který lze stáhnout zdarma z internetu. Dnes je možné ho provozovat na různých platformách. Jako příklad lze uvést operační systémy Unix, Windows (95, 98 a NT) a operační systémy počítačů Macintosh. Nesporným kladem jazyka je jeho kompatibilita téměř se všemi servery. [7]

V roce 1994 si Rasmus Lerdorf vytvořil jednoduchý systém pro evidenci přístupu k jeho webovým stránkám. Pro velkou zátěž WWW-serveru je tento systém přepsán do jazyka C a postupně uvolněn pro použití několika dalším uživatelům. Noví uživatelé dali podklad pro jeho vylepšení. A tak jej autor doplnil o dokumentaci a uvolnil jej pod názvem Personal Home Page (později Personal Home Page Construction Kit). [7]

Rasmus Lerdorf vytvořil také nástroj, který umožnil začlenění SQL-dotazů do webových stránek, vytváření formulářů a zobrazení výsledků dotazů. [7] FI se často používá jako zkratka programu Form Interpreter. Pomocí tohoto programu jsou zpřístupněny databáze na Webu. [7]

Spojením těchto dvou programů vzniká systém PHP/FI 2.0. Jde o jednoduchý programovací jazyk zapisovaný přímo do HTML-stránek. Verze PHP/FI 2.0 se rozšířila po celém světě a pracovala i na mnoha českých serverech. [7]

Zrychlením celého systému a rozšířením o mnoho dalších funkcí vzniká verze PHP3.0. Na rozdíl od předchozí verze PHP/FI 2.0 běží verze PHP 3.0 i pod Windows. K dispozici ji máme od poloviny roku 1998. [7]

PHP verze 4 využívá zcela nové jádro Zend, vyznačuje se vyšším výkonem a mnoha novými vlastnostmi a funkcemi. Zároveň se však tato verze snaží zachovat co největší kompatibilitu s verzí PHP 3. [1]

PHP však stále není nějakou mrtvou a nevyvíjeno technologií, jeho vývoj není zdaleka ukončen. Do jazyka PHP jsou neustále přidávány nějaké nové možnosti pro efektivní vytváření internetových aplikací. [7]

Kód jazyka PHP tvoří v první řadě příkazy, které vykonávají určitou funkci a po jednotlivých krocích provádějí algoritmus daného programu až k požadovanému výsledku. Příkazy jsou různých tvarů. Může jít o volání funkce, početní operaci nebo třeba o přiřazení hodnoty. [1] Je ale nutné jednotlivé příkazy od sebe oddělit. Jako oddělovač se používá středník (;). [1] Existuje několik způsobů, jak lze programovací kód jazyka PHP začleňovat přímo do zdrojového kódu HTML-stránek. [7] Dokonce může být vložen na libovolné místo uvnitř HTML dokumentu. Musí však být oddělen pomocí znakových sekvencí. [1] Elementy definované v jazyce HTML jsou definovány pomocí jazyka SGML, který definuje uzavřít PHP kód do dvojice znaků `<?>` a `? >`. Druhá možnost je používána pro generování XML dokumentů, což je zjednodušenou verzí SGML. Jako počáteční sekvenci se využívá `<?php` a ukončovací sekvence je shodná s předchozím případem, čili `? >`. Poslední možnost se využívá tehdy, nedokáže-li si s tím nijak editor poradit. Počáteční sekvence je `<script language = „php“>` a ukončovací sekvence je `</script>`. [7] Po uvedení ukončovací sekvence se dále pokračuje opět ve zdrojovém kódu HTML stránek. Soubory je důležité uložit na webový server se zvláštní příponou. Nejčastěji používané přípony jsou `***.html`, `***.php` či `***.php3`. Při výběru vhodné přípony záleží na konfiguraci serveru. [7]

Často se stává, že je potřeba vložit do zdrojového kódu nějakou tu vysvětlující poznámku neboli komentář. V podstatě existují tři způsoby vkládání komentářů do PHP kódu. Jednoduché a kratší komentáře se do skriptu zapisují za dvě lomítka (`/**`). Delší poznámky nebo komentáře vnořené do sebe používají počáteční a ukončovací sekvenci. Pro počáteční sekvenci se používá `/*` a jako ukončovací sekvence poznámky se používá její zrcadlový tvar `*/`. [7] Třetí typ vkládání vysvětlujících poznámek je převzat z shellových skriptů pro operační systém Linux. Uvozuje se `#` a znamená to, že poznámkou je vše od křížku až do konce řádku. [1]

Pro uchovávání hodnot ve skriptech PHP slouží proměnné. Jednotlivá proměnná začíná vždy znakem '\$'. Hned po znaku - bez mezery - následuje název proměnné. Každá proměnná je deklarována předem. Jestliže nebude do proměnné nic uloženo, bude jejím obsahem prázdný řetězec. V okamžiku, kdy se proměnné přiřadí její hodnota, automaticky se také určí její typ. V jazyku PHP se rozlišuje pět typů proměnných [7]:

- integer jako celé číslo;
- double jako desetinné číslo s volitelným znaménkem a desetinnou tečkou;
- string jako znakový řetězec;
- array jako pole;
- object jako objekt.

1.2 Potřebné nástroje

Základním nástrojem pro tvorbu skriptů je editor. Editorem může být téměř jakýkoliv libovolný editor, včetně nejjednoduššího poznámkového bloku [7], který není nutno instalovat, neboť je součástí standardního vybavení každého počítače. Jedinou podmínkou na použitý editor je, aby výstupem byl čistý ASCII text. To znamená čistý text bez různých řídicích instrukcí editoru. [1]

Vytvořený skript v editoru se ukládá na WWW-server, který podporuje skripty v PHP. [7] Nejoblíbenějším a zároveň zdarma stažitelným webovým serverem na všech platformách je Apache. [1] Do serveru Apache může být PHP zkompileováno jako modul. Interpret jazyka PHP je pak společně s WWW-serverem zaveden v paměti. Tím odpadají veškeré režijní náklady spojené s opakovaným spouštěním Interpretu PHP. [7] Interpret překládá zdrojový kód do kódu strojového pro počítače, na nichž PHP běží. [1] Proto je důležité ukládat soubory s PHP kódem na webový server, neboť ten automaticky předá daný soubor interpretu PHP. Pokud by tak neučinil, informace zapsané kódem programovacího jazyka PHP by se na obrazovce neobjevily. [7]

Většina skriptů, napsaných v jazyce PHP, spolupracuje s databázemi. Proto se mezi potřebné nástroje uvádí i vhodný databázový server. Programovací jazyk PHP si rozumí se všemi běžně používanými databázemi. [7]

2 TEORETICKÉ POZNATKY

Následující část se zabývá teoretickými poznatky, které se k tématu databáze vztahují. Hned na počátku je osvětlen pojem databáze, ačkoliv v dnešní moderní době nebude zajisté nikomu cizí. Další pojmem je relační databáze, protože právě ona je v současnosti tou nejrozšířenější a nejpoužívanější databází. Dále je zde uveden teoretický postup při tvorbě tabulek (relací) a postup jejich normalizace. Nepochybně do této části spadá i ER – diagram a rovněž s ním související pojmy.

2.1 Databáze

Databáze jsou vhodným prostředkem pro uspokojování lidské potřeby evidovat a shromažďovat informace. [2] Jedná se o velmi propracovaný systém. Databáze je určitá uspořádaná množina dat, která je uložena na nějakém paměťovém médiu. V širším pojetí lze do pojmu zahrnout i softwarové prostředky, jež umožňují manipulaci s daty a přístup k nim. [12] Celé předešlé vysvětlení se dá velmi jednoduše shrnout do jedné jediné věty. Databáze vlastně představují jakýsi sklad, v němž jsou data uložena. [11]

Databáze je program, který umí informace nejen ukládat do tabulek, ale také je umí rychle třídit, filtrovat a poskytovat jiným uživatelským programům. Zajisté je známo, že většina internetových serverů pracuje nějakým způsobem s databázemi. Proto musí být databáze uložena na serveru. [10]

Existuje mnoho druhů databází. Jako nejznámější se uvádějí třeba MS Access, PostgreSQL, MySQL a jiné. Není vyloučeno využití tabulky Excel jako jednoho druhu databáze, i když tento způsob není zrovna praktický. Mezi nejoblíbenější a nejčastěji používané databáze spadá především MySQL. Důvody jsou zcela zřejmé. Tato databáze je rychlá, existuje pro všechny platformy a hlavně je zadarmo. [10]

Mezi základní prvky databáze patří nepochybně data a program určený pro práci s daty. Množina logicky uspořádaných dat tvoří datový obsah celé databáze. Veškerá data, uložená v paměti počítače nebo na jiném záznamovém médiu, jsou v určitém vzájemném vztahu a z hlediska obsažených informací tvoří také jeden určitý celek. [8]

Přístup k uloženým údajům má na starosti programové vybavení, jemuž se odborně říká SŘBD – Systém Řízení Báze Dat. Název SŘBD pochází z anglického termínu DBMS – Data

Base Management System. [7] Tímto se dostáváme k souvisejícímu pojmu datový model. Většina SRBD dnes vychází z relačního modelu dat. [7] Tento model byl navržen a zveřejněn pracovníkem firmy IBM, Američanem Dr. E. F. Coddem v roce 1970. [16] Název daného datového modelu je spojen s relační algebrou, což je vlastně matematický aparát, na němž relační model dat staví. Principem modelu je uspořádat uchovávaná data do tabulek, kdy každá z nich zpravidla shromažďuje údaje o jednom druhu objektu [7] (entitě). Databáze založené na tomto modelu dat se nazývají relační databáze.

2.2 Relační databáze

Relační databáze je nejčastěji používanou databází současné doby. Důvodem je především jejich snadná ovladatelnost, jednoduchost na pochopení a dobrá reprezentace běžných datových struktur[1].

Relační databází lze nazvat implementaci modelů reálného světa, vytvořeného podle relačního modelu dat. Je to typ datového systému, v němž jsou data uspořádána do dvourozměrné tabulky. Tato tabulka (neboli relace) je základem každé relační databáze. V jakékoliv relační databázi se může vyskytovat více tabulek najednou. [1] Z toho je patrné, že relační databázi tvoří soubor nějakým způsobem vzájemně provázaných tabulek. Je také možno říci, že tvoří relační schéma. Aby bylo možné jakoukoliv tabulku jednoznačně identifikovat, musí mít každá z tabulek své jméno. [7] Program může současně pracovat s více než jednou tabulkou najednou. [4] Související informace v různých tabulkách jsou totiž vyhledávány pomocí společného pole, které ve výsledku hledání prováže obě tabulky dohromady.

Každá tabulka obsahuje údaje pouze o jednom druhu objektů [7], a proto v jedné tabulce existuje stejná struktura dat. Z předchozí věty vyplývá, že jedna tabulka popisuje jednu entitu. [2] Pojem entita je podrobněji vysvětlen níže, v části zabývající se ER – diagramem. Jednotlivé tabulky jsou určeny definovaným počtem sloupců a řádků [1]. Je výstižné říci, že údaje v jednom řádku tabulky vyobrazí aktuální stav modelovaného objektu. [2] Jakýkoliv konkrétní řádek odpovídá jednomu prvku relace (tabulky). Řádky tabulky se nazývají nejčastěji databázovým záznamem. Řádky je možné označovat i jako datové věty. Sloupce pak odpovídají jednotlivým atributům neboli prvkům datových vět. [1] Vysvětlení pojmu atribut je taktéž uvedeno v části zabývající se ER – diagramem. Jako atributy se volí takové vlastnosti, které jsou žádané pro evidenci a které jsou právě předmětem zájmu. [2] V místě

průniku sloupce a řádku vznikají pole, jinak řečeno položky, databáze. Všechna pole týkající se jednoho záznamu lze nazvat větou. [4]

U každé položky tabulky musí být přesně určený její datový typ. Přehled používaných datových typů v jazyce SQL uvádí následující tabulka (tabulka 1). [7]

Tabulka 1: Přehled datových typů SQL [zdroj: 7]

Typ	Popis
int	celé číslo v rozsahu od - 2 147 483 648 do 2 147 483 648
smallint	celé číslo v rozsahu od - 32 768 do 32 767
tinyint	celé číslo v rozsahu od 0 do 255
float	číslo s pohyblivou řádovou čárkou
char (n)	textový řetězec o délce n (maximálně však 255 znaků)
varchar (n)	textový řetězec o maximální délce n (maximálně však 255 znaků)
decimal (p)	desetinné číslo s p platnými číslicemi
decimal (p, d)	desetinné číslo s p platnými číslicemi a s d desetinnými místy
money	peněžní částka (tento typ nepodporují zdaleka všechny servery, můžeme ho však snadno nahradit například pomocí předchozího typu)
datetime	údaj o čase a datu ve formátu RRRR-MM-DD HH:MM:SS
time	údaj o čase ve formátu HH:MM:SS
date	údaj o datu ve formátu RRRR-MM-DD
blob, image	speciální typy pro uložení dlouhých binárních dat (každý server používá vlastní typ)

Tabulka relační databáze musí splňovat následující podmínky [4]:

- každá tabulka musí obsahovat pouze jeden typ záznamů, tzn., že pořadí jednotlivých polí a zároveň jejich délka musí být konstantní;
- každé pole v tabulce smí obsahovat pouze jednu hodnotu, zápis více hodnot v jednom poli není dovolen, hodnotou zde rozumíme uživatelská data;
- každý záznam se smí v tabulce vyskytovat pouze jednou;
- pořadí záznamů není definováno.

Jelikož řádky v relaci nemají specifické pořadí, nejsou dosažitelné číslem řádku. Přesto musí existovat nějaká konstrukce, aby uživateli umožnila jednotlivé řádky tabulky adresovat. Tato konstrukce je nazývána primárním klíčem. Jedná se o atribut, popřípadě soustavu atributů, jejichž hodnoty zajišťují jednoznačnou identifikaci jednotlivých řádků relace (tabulky). [16] Primární klíč musí být tedy jednoznačný a jeho hodnota se smí vyskytnout pouze jednou. V případě, že je primární klíč složen z více položek, mohou se některé z nich opakovat, ne

však všechny současně. [2] Každý atribut, jež je součástí primárního klíče, se nazývá klíčový atribut. Ostatní atributy jsou neklíčové. [16]

2.3 Tvorba tabulek

Tvorba tabulek je klíčovým a velmi důležitým úkolem při navrhování databází a to zejména z dlouhodobého hlediska. Není tedy divu, že je nutné tabulku navrhnout správně. Pokud by se totiž v nějakém databázovém systému, který je již nějakou dobu v provozu, vyskytla chyba, která spočívá ve špatně navržené databázi, byly by důsledky přímo katastrofální a to ne jen po finanční stránce. [2]

Tabulky, z nichž se tvoří databáze, musí vyhovovat předem definovaným požadavkům a omezením, ale přesto se v nich mohou vyskytovat redundantní (nadbytečná) data. Rovněž může docházet k anomáliím při manipulaci s nimi. Data v tabulce při základním návrhu zcela běžně obsahují opakující se informace. Jejich výskyt zabírá místo, komplikuje a zároveň zpomaluje vkládání, mazání i aktualizace dat. Pro eliminaci zmiňovaných problémů se provádí tzv. normalizace dat. [5]

2.4 Normalizace dat

Normalizace dat je proces postupné dekompozice datového modelu tím, že atributy budou rozděleny do většího počtu relací. Takto vzniklé relace již pak nevykazují dané nedostatky. Proces normalizace by měl vést ke vzniku takových tabulek, které lze snadno udržovat a na které se lze efektivně dotazovat. Ve výsledku by měla vzniknout přehlednější, rozšiřitelnější a výkonnější databáze. Výsledné normalizované schéma musí zachovávat všechny závislosti původního schématu a relace musí zachovat veškerá původní data. To znamená, že uživatel se musí pomocí přirozeného spojení dostat k původním datům. [15]

Pro dekompozici dat existuje standardizovaný algoritmus. Tento algoritmus se skládá z několika fází, nazývaných normální formy [5]:

- první normální forma (1NF);
- druhá normální forma (2NF);
- třetí normální forma (3NF);
- Boyce-Coddova normální forma (BCNF);

- čtvrtá normální forma (4NF);
- pátá normální forma (5NF).

V praxi si většina databází vystačí s normalizací do třetí normální formy. Na této úrovni procesu dekompozice (normalizace) dat dojde k odstranění většiny anomálií. Normalizace relací do nejvyššího stupně není vždy nutné a dokonce ani výhodné – například kvůli požadované výkonnosti. [5]

1. normální forma

První normální forma bývá stručně označována 1NF. Jedná se o první a zároveň nejjednodušší normální formu. Relace je v první normální formě, pokud mají všechny její atributy atomické hodnoty, tedy jsou dále nedělitelné. [2] První normální forma vyplývá již ze základních vlastností o nedělitelnosti atributů. Znamená to, že žádný atribut nesmí obsahovat více než jednu hodnotu. Obecně řečeno, je důležitá snaha o to, aby obsahem jedné databázové položky byla právě jedna hodnota. [2]

Nastane-li tedy případ, kdy jeden atribut obsahuje více hodnot, tudíž není atomický (dále nedělitelný), je neatomický atribut rozdělen do více atributů v příslušné relaci.

2. normální forma

Druhá normální forma užívá stručného označení 2 NF. Základním požadavkem 2. normální formy je, aby relace byla v 1 NF. Navíc musí splňovat podmínku, že každý neklíčový atribut, čili atribut, který není primárním klíčem, je úplně závislý na primárním klíči. [2] Při zmínění primárního klíče připadá v úvahu celý primární klíč a nikoliv jen nějakou jeho podmnožinu. Žádná položka, jež se v tabulce objevuje, nesmí být tedy závislá pouze na nějaké části primárního klíče. [15]

Z uvedené definice vyplývá, že problém 2 NF je nutno řešit pouze u tabulek, kde byl zvolen vícestupňový primární klíč. Na první pohled vypadá 2 NF složitě. Jinými slovy, pokud nemá tabulka složený primární klíč, je 2 NF splněna triviálně. [2]

Jestliže nastane taková situace, že 2 NF není splněna a závislost neklíčového atributu na primárním klíči je pouze částečná, dochází k další dekompozici. Tento problém bývá řešen rozkladem původní tabulky na dvě a více tabulek, přičemž každá z nich bude splňovat druhou

normální formu. Všechny tabulky jsou propojeny určitou, vhodně zvolenou, částí primárního klíče. Procesu dekompozice na úrovni druhé normální formy se říká dekompozice relačního schématu. [2]

Na této úrovni je patrné, že v případě nesplnění 2 NF dochází často k redundanci. Redundance je jev nežádoucí a obvykle doprovází nesplnění druhé normální formy. [2]

3. normální forma

Stručným označením třetí normální formy je 3 NF. Tuto normální formu splňuje taková relace, která se nachází v předchozí normální formě (čili v 2 NF) a každý její neklíčový atribut není tranzitivně závislý na primárním klíči. [2] Vypsaná definice třetí normální formy zní poněkud zmateně a složitě, avšak budou-li zvolena jiná slova, je její vysvětlení v podstatě docela jednoduché. Lze říci, že každý neklíčový atribut nesmí záviset na jiném atributu, závislém na primárním klíči. Tranzitivní závislost je vlastně závislost vyskytující se mezi minimálně dvěma atributy a klíčem. V takovém případě je jeden atribut funkčně závislý na klíči a druhý atribut je závislý na prvním atributu. [15]

Existuje ještě další způsob, jak vysvětlit třetí normální formu. 3 NF vyžaduje, aby veškeré atributy, jež nejsou součástí primárního klíče, byly na primárním klíči přímo závislé. A samozřejmě je nutné, aby relace splňovala druhou normální formu.

Řešením problému nesplnění 3 NF je obdobný postup jako v předchozím případě. Opět se jedná o řešení pomocí dekompozice. Původní tabulka se rozdělí do dvou a více tabulek, přičemž je zapotřebí brát ohled na to, aby nově vzniklé tabulky šlo spojit pomocí nějakého klíče opět dohromady. Ve druhé tabulce se stane primárním klíčem neklíčový atribut, na němž byl další neklíčový atribut tranzitivně závislý.

Boyce-Coddova normální forma

Boyce-Coddova normální forma používá označení BC NF. Ve většině případů normalizace dat už není tato normální forma zapotřebí. Již dříve bylo zmíněno, že většina procesů dekompozice končí právě předchozí třetí normální formou. Touto formou se zjednodušuje práce s tabulkami. BC NF je splněna ve většině případů, když je splněna 1 NF, 2 NF i 3 NF. [2]

Je pokládána za variaci třetí normální formy a dokonce je pravda, že BC NF je původní definicí 3 NF. Tato normální forma je vymezena stejnými pravidly jako 3 NF, a říká, že musí platit i mezi hodnotami uvnitř primárního klíče. [15]

Relace splňuje BC NF, jestliže pro dvě množiny atributů A a B platí: $A \rightarrow B$ a současně B není podmnožinou A, pak množina A obsahuje primární klíč tabulky. [2] Znamená to vlastně to, že mezi kandidátními klíči v tabulce nesmí být žádná funkční závislost. [15]

V takové situaci, kdy není splněna Boyce-Coddova normální forma, se nabízí opět klasické řešení. Z původní tabulky, jež vyhovuje třetí normální formě, budou dekompozicí vytvořeny dvě nové tabulky, které budou spojeny pomocí atributu B.

Shrnutě lze říci, že BC NF náleží taková relace, kde každý determinant funkční závislosti v jedné relaci je zároveň kandidátním klíčem.

4. normální forma

Ani čtvrtá normální forma nebude výjimkou ve svém stručném označení. Značí se 4 NF. Při znalosti splňování předchozích normálních forem, je snadné si i zde odvodit, že čtvrté normální formě náleží taková relace, která musí splňovat předchozí normální formu, tedy Boyce-Coddovu normální formu, a zároveň také popisuje pouze příčinnou souvislost (neboli jeden fakt). Takováto definice je sice nastíněna jednoduchým způsobem, ale bohužel málo pochopitelná. Proto je zmíněna ještě jiná definice. [15]

Čtvrtou normální formu splňuje taková relace, jež je v BC NF a navíc veškeré vícehodnotové závislosti jsou zároveň funkčními závislostmi z kandidátních klíčů. Z toho vyplývá, že všechny vícehodnotové závislosti jsou zároveň funkčními závislostmi. [15] Pro doplnění je následně vysvětlena i vícehodnotová závislost. Existuje relace R splňující BC NF. V relaci se objevují atributy A, B a C. Vícehodnotová závislost atributu B na atributu A nastává právě tehdy, když množina hodnot B přiřazená dvojici A a C závisí pouze na hodnotě atributu A a na hodnotě atributu C je nezávislá. [15]

Tato normální forma se zabývá vztahy uvnitř složeného primárního klíče. Objeví-li se v tabulce složený primární klíč, je možné, že některé hodnoty tohoto klíče budou na sobě nezávislé. Tím, že spolu tvoří klíč, vzniká mylná souvislost mezi těmito hodnotami a nemohou existovat nezávisle na sobě. To ale není v souladu s modelovanou realitou. Proto

čtvrtá normální forma vyžaduje, aby klíč tvořily pouze hodnoty se skutečnou vzájemnou souvislostí. [15]

Nastane-li problém že v tabulce existují atributy, které tvoří všechny dohromady primární klíč a mezi nimiž neexistuje žádná funkční závislost, řešení opět spočívá v rozkladu tabulek.

5. normální forma

I relace v páté normální formě, která se označuje 5 NF, musí být ve čtvrté normální formě. Navíc je zde podmínka, že nemůže být bezetrátově rozložena. Týká se primárních klíčů, které jsou tvořeny alespoň třemi atributy. Pakliže mezi těmito hodnotami existují párové cyklické závislosti, je zapotřebí tyto závislosti extrahovat do samostatných tabulek. Původní tabulku je však v některých případech nutno zachovat. [15]

K porušení této formy musí být splněno několik specifických podmínek. Relace ve 4 NF musí mít klíč složený ze třech či více atributů. Mezi těmito atributy musí být párové cyklické závislosti, tedy nikoliv funkční nebo multizávislosti. Typickým příkladem je vztah třech a více tabulek, kde platí M:N:O:M. Přičemž takovýto vztah je tvořen jednou relací. Pátá normální forma tedy řeší redundanci dat a možnou ztrátu závislostí. [15]

Je-li tato situace řešena dalším rozkladem, není nejvhodnější původní tabulku vymazat. Při pohledu na opětovné složení tabulek dohromady, není výsledkem původní tabulka. Pátá normálová forma tedy bývá o tom, uvědomit si skutečné vztahy. [15]

Ve výkladu normálních forem se objevily pojmy jako je kandidátní klíč nebo funkční závislost. Protože jejich význam není nikde výše uveden, dochází k vysvětlení těchto pojmů nyní. **Kandidátním klíčem** může být několik kandidátů na roli identifikačního neboli primárního klíče. Jen tak pro úplnost lze zmínit, že existují také cizí klíč (což je atribut nebo množina atributů, které jsou v jiné entitě primárním klíčem nebo alespoň jeho součástí), alternativní klíč (to jest minimální množina atributů, které zajišťují jednoznačnou identifikaci) a sekundární klíč (slouží pro vyhledávání a třídění dat).

Pojem **funkční závislost** se dá jinak pojmenovat jako speciální druh integritního omezení. Funkční závislost je definována mezi dvěma množinami atributů a to v rámci jednoho schématu relace. Jde tedy o vztah mezi daty. Bývá vztažena ke schématu relace a vymezuje množinu přípustných relací. [2]

2.5 Relační schéma

ER – model (entitně – relační model) se uvádí jako klasický způsob prezentace modelovaného světa. Základ ER – modelu spočívá v převedení komplexních struktur modelované skutečnosti do dvou rozměrných tabulek a v nalezení vztahů mezi nimi. ER – model je vhodný pro relační databáze zejména proto, že je snadno transformovatelný do tabulek. Slouží ke reprezentaci logických vztahů mezi entitami nebo objekty. Jeho úloha při návrhu, optimalizaci databází a odstraňování chyb je nezastupitelná. Pokud efektivní datový model obsahuje úplný a přesný popis zadání a pokud vyhovuje nárokům zadání, eliminuje redundanci dat a je jak hardwarově tak softwarově nezávislý. [5]

Jeho základní komponentou je ER – diagram, který se používá pro vizuální reprezentaci dat.[5] Každý ER – diagram je složen z několika prvků. Znázorníme ho pomocí entit a jejich atributů a vztahů mezi jednotlivými entitami.

Základním prvkem ER - diagramu je entitní typ neboli **entita**. [9] Entita reprezentuje nezávisle existující objekt reálného světa, který je předmětem našeho zájmu. Tento objekt je schopen nezávislé existence a je možné ho jednoznačně odlišit od objektů ostatních. [6] Mohou to být lidé, pojmy, objekty, hmotné i nehmotné věci či nějaké události. [3] Každá entita bývá popsána svým názvem a sadou atributů. [17] Názvem entity je podstatné jméno v jednotném čísle. V ER – diagramu je zakreslena klasickým obdélníkem. Entity mohou být obecného, slabého, silného či vazebního typu. [12]

Obecná entita je jakákoliv entita bez rozlišení druhu. Slabá entita, jinak řečeno popisná entita, je taková, která je buď existenčně závislá, nebo identifikačně závislá na jiných entitách. Existenční závislost je vyjádřena podle charakteristiky vztahu, kdežto identifikační závislost znamená, že nemají identifikační atributy, resp. identifikační klíč je vytvořený pouze z atributů entity slabého typu. Silná entita bývá často uváděna také jako kmenová, základní nebo regulární entita, která existuje nezávisle na ostatních entitách. Dalším typem je vazební entita. Ta realizuje určitou vazbu mezi entitami. Často se také označuje jako entita asociativní. [12]

Mezi typy entit se řadí i generalizace (nadtyp) a specializace (podtyp). Pojem generalizace představuje vytvoření entity vyšší úrovně ze dvou či více entit nižší úrovně. Specializace je jejím opakem. Znamená to vytvoření jedné nebo více entit nižší úrovně z jedné entity vyšší

úrovně, přičemž takto vytvořená entita přebírá veškeré atributy svého nadtypu. Se specializací souvisí pojmy, jako je exkluzivní podtyp a inkluzivní podtyp. O exkluzivním podtypu se hovoří tehdy, pokud entita má dva nebo více podtypů a nedochází přitom k jejich překrývání. To znamená, že jeden výskyt entity nemůže být zároveň také členem více podtypů. U inkluzivního podtypu k takovému překrývání podtypů dochází. [12]

Každá z těchto entit je popsána několika svými atributy. **Atributy** jsou datové prvky, které vyjadřují funkci přiřazující entitám či vztahům hodnotu. Přiřazená hodnota určuje některou podstatnou vlastnost entity, případně může blíže charakterizovat i vztah. [3] Ale pouze jeden z nich musí entitu jednoznačně identifikovat. Takový atribut se nazývá jedinečný identifikátor a odpovídá primárnímu klíči. Jedinečný identifikátor musí mít jedinečnou hodnotu pro každý výskyt entity. [9] Entita, jež nelze ničím identifikovat v podstatě neexistuje. [3] Pro znázornění klíčového atributu používáme podtržené písmo. Ostatní atributy jsou pouze popisné atributy. U každého atributu se upřesňuje – kromě jeho názvu – také množinu hodnot, kterých může nabývat.

Atributy se rozlišují podle volitelnosti či odvoditelnosti. Pokud jde o členění z hlediska volitelnosti, rozeznává se totální atribut a parciální atribut. Totální atribut má pro každý výskyt entity hodnotu [3], z následujícího ER - diagramu je to například kniha a její signatura. V opačném případě, kdy každý výskyt entity nemusí mít svou hodnotu, se jedná o parciální atribut [3]. Takový příklad atributu v následném ER - diagramu není. Příkladem může být entita vyučující a atribut například fax, protože každý vyučující nemusí mít svůj fax. Podle odvoditelnosti se atributy rozlišují na atributy základní a odvoditelné. Základní atributy nelze odvodit z jiných atributů [3] a jsou to například u knihy signatura nebo její název. Naopak odvoditelné atributy jsou takové atributy, jejichž hodnotu je možné získat odvozením ze základního atributu [3]. Každý atribut je stejně jako entita reprezentován podstatným jménem.

Mezi jednotlivými entitami jsou zaznamenány různé **vztahy** (vazby). Každý vztah vyjadřuje reálnou vazbu mezi dvěma a více entitami. Jeho obsahová informace nelze odvodit z atributů jednotlivých entit. Názvem vztahu je slovesem sloveso, vyjadřující podstatu vztahu z hlediska obou partnerských entit. [3] Existují dva typy vztahů. Binární vztahy jsou takové vztahy, které se vyskytují mezi dvěma entitami. Binární vztahy jsou také nejčastěji vyskytovaným typem vztahů. N-ární vztahy, jinak nazvané polynární či vícenásobné, jsou vztahy mezi více než dvěma entitami [3]. Vztahy mezi entitami (resp. mezi výskyty entit) vyjadřují asociaci (tj.

reálnou logickou vazbu) a podávají informace o entitách. [14] U vztahu se definuje integritní omezení: kardinalita, parcialita, exkluzivita a externí identifikace. [12]

Kardinalita vztahu vyjadřuje počty výskytů entit v určitém vztahu. Definuje se minimální nebo maximální kardinalita. Minimální kardinalita nabývá hodnot 0 nebo 1 a maximální kardinalita 1 nebo N. [d] Může nabývat následujících hodnot [18]:

- 1:1, kdy každé entitě se přiřadí maximálně jedna druhá (jiná) entita;
- 1:N, kdy první entitě se přiřadí více než jedna druhá entita, avšak té druhé entitě se přiřadí maximálně jedna první entita, druhá entita bývá označována jako determinant entity první;
- M:N, kdy první entitě se přiřadí více než jedna druhá entita a druhé entitě více než jedna první entita.

Parcialita, jinak označována jako volitelnost, definuje členství ve vztahu jako povinné či nepovinné členství jednotlivých entit, které se vztahu účastní. Každá entita může i nemusí vstupovat do vztahu, vztah tedy může nastat i nemusí. [3] Pokud má entita ve vztahu členství povinné, je tato entita existenčně závislá na druhé entitě vyskytující se v daném vztahu a toto členství bývá také označováno jako obligatorní. Často se pro takovou situaci, kdy entitní typ s obligatorním členstvím ve vztahu je existenčně závislý na jiném entitním typu, používá pojem existenční závislost. [14] Při nepovinném členství hovoříme pouze o částečné závislosti. Typickou hodnotou pro povinné členství entity ve vztahu je jednička a pro nepovinné členství ve vztahu je nula. [14]

Exkluzivita neboli výlučnost znamená, že pro jeden výskyt entity je realizován právě jeden ze vztahů. Slabý vztah čili **externí identifikace** vyjadřuje vnější identifikační závislost. [12]

2.6 Zvolené prostředky

Pro tvorbu praktické části bakalářské práce jsou nepochybně zapotřebí nějaké pomocné prostředky. Jako první prostředek je uveden programovací jazyk, v němž budou zapsány zdrojové kódy pro realizaci. Zvoleným programovacím jazykem je jazyk PHP. Tento jazyk byl vzpomenut v první části práce.

Dalším pomocníkem bude Apache2Triad. Apache, jak již bylo zmíněno, je nejoblíbenějším webovým serverem na všech platformách. Mezi jeho nesporné výhody patří to, že je zdarma dostupný a velmi dobře dokumentován, čímž je jeho zprovoznění snazší, než dříve [1]. Postup instalace tohoto webového serveru je popsán v následující realizační části.

Kódy programovacího jazyka je nutné zapisovat do nějakého textového editoru. Jako textový editor byl pro tuto práci zvolen PSPad editor. Databáze je důležitým pojmem v této bakalářské práci, neboť v ní budou uloženy veškeré údaje, které budou pomocí vytvořeného programu vyhledávány. Mezi zvolené prostředky je tedy potřeba zahrnout i nějakou databázi na databázovém serveru. K tomuto účelu byla vybrána MySQL databáze, jež je v současné době nejfrekventovanější databází. Důvody oblíbenosti MySQL databáze jsou jednoduché. Databázi lze pořídit zcela zdarma a rychlost připojení k databázi.

3 REALIZACE

V předchozí kapitole byla zmíněna teorie, která bude v této části bakalářské práce využita k návrhu databáze knihovny a tvorbě ER – diagramu. Nejprve je zde vysvětleno téma celé bakalářské práce a jsou zde zmíněny podmínky k jejímu vypracování. Dále budou využity poznatky o databázích a tvorbě ER – diagramu z předchozí teoretické části. Výsledkem postupu bude jeden celkový ER – diagram. Závěr této části bude věnován již samotnému programování. Programování bude řešeno pomocí programovacího jazyka PHP, kterým se zabývá celá první část bakalářské práce.

3.1 Analýza problému

Téma bakalářské práce zní „Návrh a realizace knihovny“. Jedná se o knihovnu pro školu a jejím předmětem jsou především knihy, které byly zapůjčeny na delší časové období. Tyto knihy se v univerzitní knihovně neobjeví, neboť byly zakoupeny z finančních prostředků grantů a jsou tedy majetkem ústavu. Každá kniha může být identifikována podle čárového kódu, signatury nebo přírůstkového čísla. K jednoznačné identifikaci knih se používá právě přírůstkové číslo.

Seznam knih je vyhotoven v souboru Excel. Ten bude následně importován do vytvořené MySQL databáze na databázovém serveru. V souboru se vyskytují informace jako je název knihy a její autor (resp. autoři), přírůstkové číslo, signatura, čárový kód a informace o datu vypůjčení a vrácení.

Vytvořený program by měl zajistit vyhledávání knih, které bude probíhat podle čtyř nabídnutých a pevně stanovených položek. Jedná se o název, autora, signaturu a přírůstkové číslo. Ve výsledku hledání by se měly objevit především informace o názvu knihy, jméně autora, datu vypůjčení a vrácení knihy a vyučujícím, jenž si knihu zapůjčil.

Realizace vyhledávacího programu v databázi by měla být přístupná přes webové stránky. Přístup bude zajištěn pomocí databázového a webového serveru prostřednictvím programovacího jazyka PHP a HTML kódu.

3.2 Návrh databáze a tvorba ER – diagramu

Vytvořený ER – diagram vyjadřuje relační schéma budoucí knihovny. Nejprve jsou identifikovány entity, jimž jsou následně přiřazeny jejich atributy.

Z grafického znázornění je patrné, že ER-diagram má čtyři entity. Entity jsou znázorněny pomocí klasického obdélníčku. Každá entita má svůj jedinečný identifikátor, který je psán podtrženým písmem. Mezi entitami existují různé vztahy nebo - li vazby. Vztahy se zobrazují kosočtvercem.

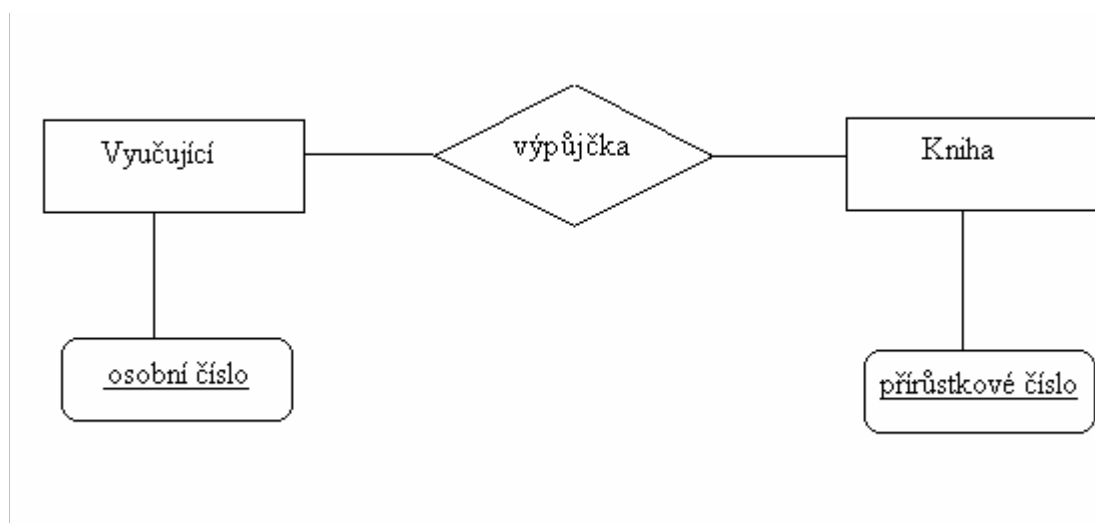
ER – diagram

Prvním krokem při tvorbě ER – diagramu je vytipování entit a jejich atributů. V tomto případě se jedná dvě entity. Vytipované entity a atributy uvádí následující tabulka (tabulka 2).

Tabulka 2: Vytipování entit a atributů [zdroj: vlastní]

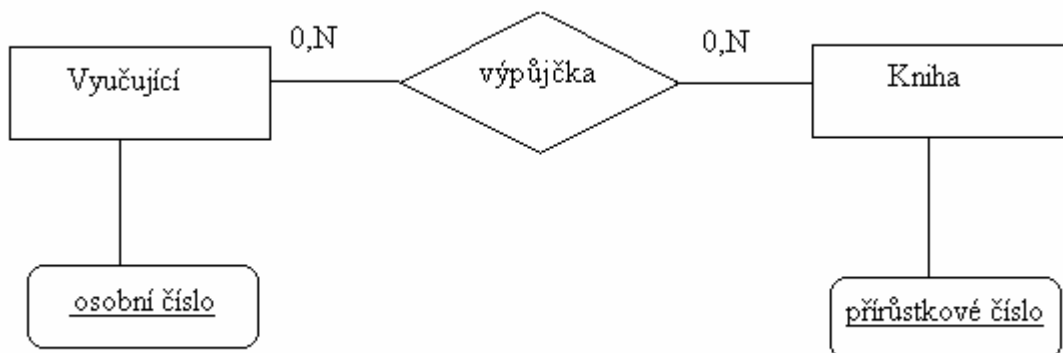
Entita	Atributy
Kniha	Přírůstkové číslo, signatura, název, autor
Vyučující	Osobní číslo, jméno, příjmení

Knihovna má svoji databázi knih, kde jsou jednotlivé knihy jednoznačně identifikovány jejich přírůstkovým číslem. Existuje také kartotéka vyučujících, kteří jsou jednoznačně identifikováni jejich osobním číslem. Entity a jejich klíčové atributy jsou zobrazeny na následujícím obrázku 1. Jednotlivá výpůjčka je pak dána klíčovými atributy obou entit, tedy osobním číslem vyučujícího a přírůstkovým číslem knihy.



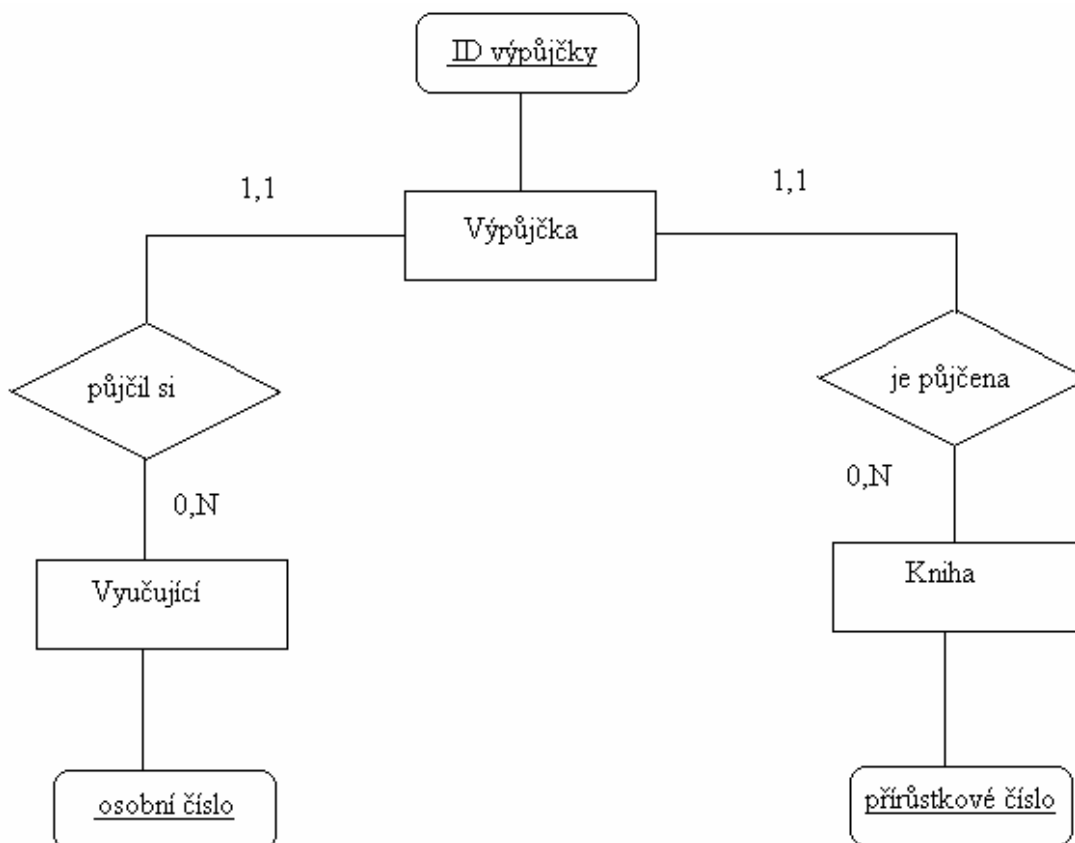
Obrázek 1: Diagram vytipovaných entit [zdroj: vlastní]

Dále se tvoří již první analytické schéma (obrázek 2), v němž je zakresleno, že každý vyučující si může půjčit více knih. Zároveň je ze schématu patrné, že každá kniha může být půjčena vícekrát.



Obrázek 2: První analytické schéma [zdroj: vlastní]

Dalším krokem je kontrola správnosti schématu. Ke každé výpůjčce bývá veden nějaký protokol. V protokolu jsou obsaženy informace o datu půjčení a o datu vrácení knihy. Jednotlivé protokoly jsou identifikovány jednoznačným číslem, ID výpůjčky, a jsou ukládány a archivovány. Na této úrovni dochází k převedení vztahu výpůjčka do entitního typu Výpůjčka. Na obrázku číslo 3 je zakreslena dekompozice vztahu výpůjčka do entitního typu Výpůjčka.



Obrázek 3: Dekompozice vztahu výpůjčka [zdroj: vlastní]

V této fázi dochází k opakování prvního kroku a opětovnému definování entit a atributů. Definované entity a jejich atributy jsou uvedeny v tabulce 3.

Tabulka 3: Definované entity a atributy [zdroj: vlastní]

Entita	Atributy
Kniha	Přírůstkové číslo, signatura, název, autor
Vyučující	Osobní číslo, jméno, příjmení
Výpůjčka	ID výpůjčky, přírůstkové číslo, osobní číslo, datum půjčení, datum vrácení

Postupným vývojem vznikly relace:

Relace Kniha (přírůstkové číslo, signatura, název, autor)

Relace Vyučující (osobní číslo, jméno, příjmení)

Relace Výpůjčka (ID výpůjčky, přírůstkové číslo, osobní číslo, datum půjčení, datum vrácení)

Normalizace

1NF

Relace Vyučující a relace Výpůjčka zůstanou zachovány, protože splňují první normální formu.

Relace Vyučující (osobní číslo, jméno, příjmení)

Relace Výpůjčka (ID výpůjčky, přírůstkové číslo, osobní číslo, datum půjčení, datum vrácení)

Relace Kniha první normální formu nesplňuje. Atribut autor není atomický (je vícehodnotový). Proveďte dekompozici, čímž vzniknou z jednoho původního atributu autor dva nové – jméno autora a příjmení autora.

Relace Kniha (přírůstkové číslo, signatura, název, jméno autora, příjmení autora)

2NF

Druhá normální forma je všemi relacemi splněna.

Relace Vyučující (osobní číslo, jméno, příjmení)

Relace Výpůjčka (ID výpůjčky, přírůstkové číslo, osobní číslo, datum půjčení, datum vrácení)

Relace Kniha (přírůstkové číslo, signatura, název, jméno autora, příjmení autora)

3NF

Relace Vyučující a relace Výpůjčka opět splňují i třetí normální formu. Jejich tvar tedy zůstává zachován.

Relace Vyučující (osobní číslo, jméno, příjmení)

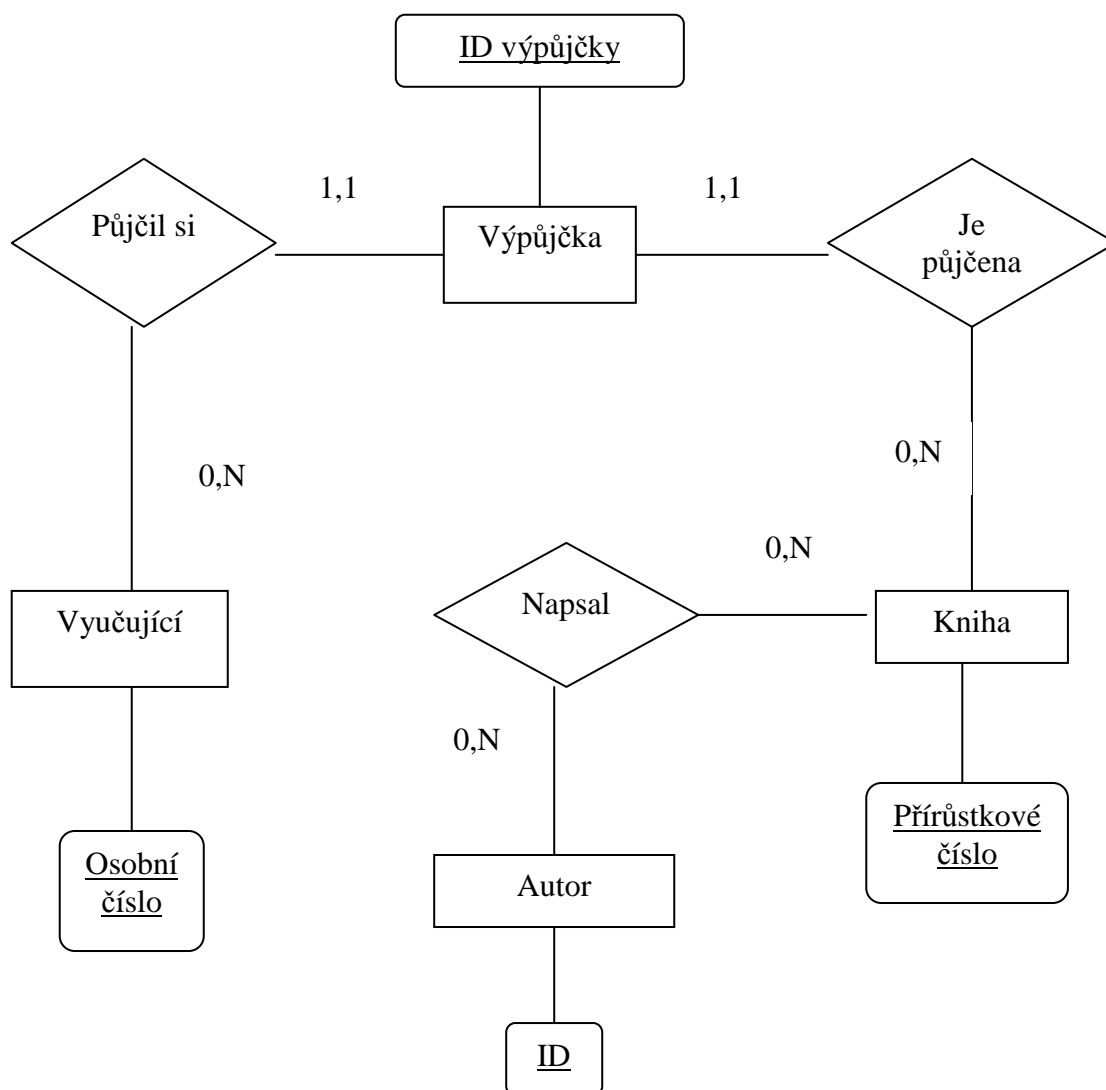
Relace Výpůjčka (ID výpůjčky, přírůstkové číslo, osobní číslo, datum půjčení, datum vrácení)

Relace Kniha třetí normální formu nesplňuje. Atributy jméno autora a příjmení autora nejsou přímo závislé na primárním klíči relace Kniha. Proveďte dekompozici, čímž vzniknou z jedné původní relace dvě nové.

Relace Kniha (přírůstkové číslo, signatura, název, ID autora)

Relace Autor (ID autora, jméno autora, příjmení autora)

Po dokončení procesu normalizace vzniká již výsledný, celkový ER – diagram, který je zobrazen na obrázku 4.



Obrázek 4: ER - diagram [zdroj: vlastní]

3.3 Vysvětlení ER – diagramu

Každý vyučující si může půjčit jednu i více knih. Entitu Vyučující jednoznačně identifikuje jeho osobní číslo. Každá kniha, jejímž jedinečným identifikátorem je přírůstkové číslo, může být půjčena jednomu i více vyučujícím. Jednotlivá výpůjčka musí být přiřazena pouze

jednomu vyučujícímu a může se týkat pouze jedné knihy. Výpůjčku charakterizuje jedinečný identifikátor ID výpůjčky. Autor je identifikován svým jedinečným číslem ID autora a může napsat jednu i více knih. Zároveň také platí, že každá kniha může být napsána jedním ale i více autory.

Z níže uvedené tabulky (tabulka 4) lze vyčíst údaje o jednotlivých entitách. Ke každé entitě jsou zaznamenány její jednotlivé atributy. U každého atributu je dále uveden typ, který vyjadřuje (například text, číslo atd). V posledním sloupci tabulky je uvedena velikost atributového pole. Velikost je udána počtem znaků.

Tabulka 4: Tabulka entit a atributů [zdroj: vlastní]

Entita	Atributy	Typ	Velikost
Vyučující	Osobní číslo	Text	5
	Jméno	Text	50
	Příjmení	Text	50
Výpůjčka	Id výpůjčky	Číslo	3
	Datum vypůjčení	Číslo	10
	Datum vrácení	Číslo	10
	Osobní číslo	Text	5
	Přírůstkové číslo	Číslo	6
Kniha	Přírůstkové číslo	Číslo	6
	Signatura	Text	10
	Název	Text	90
	Autor	Text	50
Autor	ID autora	Číslo	3
	Jméno	Text	50
	Příjmení	Text	50

Popis entit

Vyučující – entita nazvaná Vyučující reprezentuje osobu, která je zaměstnaná jako pedagog na zdejší univerzitě. Jako atributy k dané entitě jsou zvoleny jméno, příjmení a osobní číslo, které je také jedinečným identifikátorem entity Vyučující. Mezi atributy lze zahrnout i mnoho dalších informací jako například adresa, datum narození atd. Ale pro řešení tohoto návrhu knihovny nejsou tyto atributy podstatné, a proto nejsou v tabulce ani uvedeny.

Výpůjčka – tato entita je vlastně záznam jednotlivých provedených výpůjček knih. Jejími atributy jsou ID výpůjčky, jež je právě jedinečným identifikátorem, dále pak datum vypůjčení a datum vrácení, osobní číslo a přírůstkové číslo.

Kniha – jedná se o entitu reprezentující jednotlivé knihy, které jsou předmětem tvorby návrhu a realizace knihovny. Pro identifikaci této entity jsou vybrány atributy ID autora, signatura, název a jako jedinečný identifikátor se používá přírůstkové číslo každé knihy.

Autor – nepochybně v uvedeném výčtu entit nesmí chybět ani entita Autor, která reprezentuje osobu, jež je autorem knihy. Osoba autora se u jednotlivých knih liší a někdy je možné zaznamenat u jedné knihy také více autorů. Atributy jsou jméno, příjmení a ID autora. Jedinečným identifikátorem zmiňované entity je právě ID autora.

Popis vztahů

Půjčil si – tento vztah je mezi entitami Vyučující a Výpůjčka. Znamená to, že vyučující si vypůjčí nějakou knihu a tato skutečnost je zaznamenána pod příslušnou výpůjčkou.

Je půjčena – vztah, který je nazván Je půjčena, se nachází mezi entitami Výpůjčka a Kniha. Z daného vztahu plyne, že každá výpůjčka je záznamem o knize, která je půjčena nějakému vyučujícímu.

Napsal – jedná se o poslední vztah v předchozím ER – diagramu. Tento vztah existuje mezi entitami Autor a Kniha. Vyplývá z toho, že každá kniha je napsána jedním (popřípadě více autory) a každý z uvedených autorů napsal jednu (popřípadě více knih).

Integrovní omezení vztahů

V předchozím ER – diagramu existují pouze dva typy vztahů. Jeden vztah je 1:1 a druhý 0:N. Co se prvního typu (1:1) týče, kardinalita je zde označena jedničkou. To znamená, že počet výskytů entity v určitém vztahu je maximálně jedna. Parcialita je rovněž označena jedničkou. Z výše uvedené teorie je známo, že jednička definuje povinné členství ve vztahu. Tento typ vztahu se v diagramu vyskytuje pouze dvakrát.

Druhým typem je vztah 0:N, který se vyskytuje čtyřikrát. Nulou je označena parcialita, z níž je možno vyčíst nepovinné členství entity ve vztahu. Kardinalita je reprezentovaná písmenkem N.

3.4 Instalace zvolených prostředků

Na závěr předešlé kapitoly bylo věnováno několik slov k prostředkům, jež jsou k řešení tohoto problému potřebné. V této části je uveden postup, pomocí něhož je možné zmiňované prostředky získat.

Instalace Apache2Triad

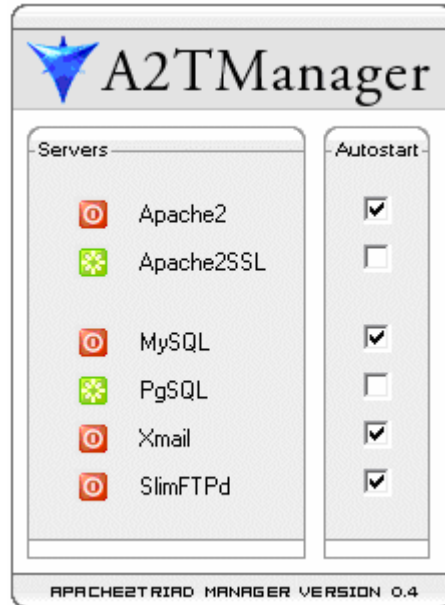
Nejprve je dobré si vyhledat a stáhnout vhodnou verzi (např. verzi Apache2Triad 1.5.4). Apache2Triad se spustí a nechá se uložit na místo, které je samo nabídnuto. Instalace požaduje vložit osmi místné heslo. Dále se postupuje v průvodci tlačítkem Next a souhlasem s licenčními podmínkami. Po dokončení instalace se objeví dialogové okno. Stiskem tlačítka OK je potvrzena konfigurace Apache2Triad. Při procesu konfigurace je opět požadováno osmi místné heslo - shodné s tím, jež bylo zadáno na začátku. Po dokončení konfigurace opět naběhne dialogové okno. Po kliknutí na tlačítko OK proběhne restart počítače. To je známka toho, že Apache2Triad byl úspěšně nainstalován. Apache2Triad je dobrý kvůli podpoře W3C konsorcia a zároveň také obsahuje dobrou nápovědu i v českém jazyce.

Instalace domácího lokálního serveru

Prvním krokem při instalaci je otevření webového prohlížeče, v němž se na nový panel vypíše slovo LOCALHOST. Stiskem klávesy Enter, se otevře stránka místního serveru, kde je uveden seznam položek. Z daného seznamu je asi nejvíce zajímavé PHPmyadmin, jež je potřeba při zřizování nové databáze. Další postup vede přes tlačítko Start na hlavním panelu, tam bude vybrána položka Všechny programy a z té se spustí Apache2Triad MANAGER a Apache2Triad MONITOR. Po otevření Apache2Triad Manager je důležité zkontrolovat autostart (obrázek 5), v němž mají být zaškrtnuta následující políčka:

- apache 2
- mysql
- xmail
- slim FTPd.

Po kliknutí na ikonu Apache v dolním panelu se Apache2Triad manager automaticky zavře.



Obrázek 5: Autostrt [zdroj: vlastní]

Další postup je přes ikonu Tento počítač na Místní disk C. Otevřeme Apache2Triad. Důležitým upozorněním je, že adresáře, nacházející se zde, se nikdy nepřejmenovávají! V těchto místech se nachází adresář nazvaný „htdocs“, do něho se vytvoří nový adresář (barunka) pro pozdější ukládání kódů. Název adresáře se, pro lepší orientaci, píše malými písmenky a bez diakritických znamének.

Adresář se po návratu do webového prohlížeče na localhost a po stisku klávesy F5 objeví i zde. Tím je nainstalován domácí lokální server.

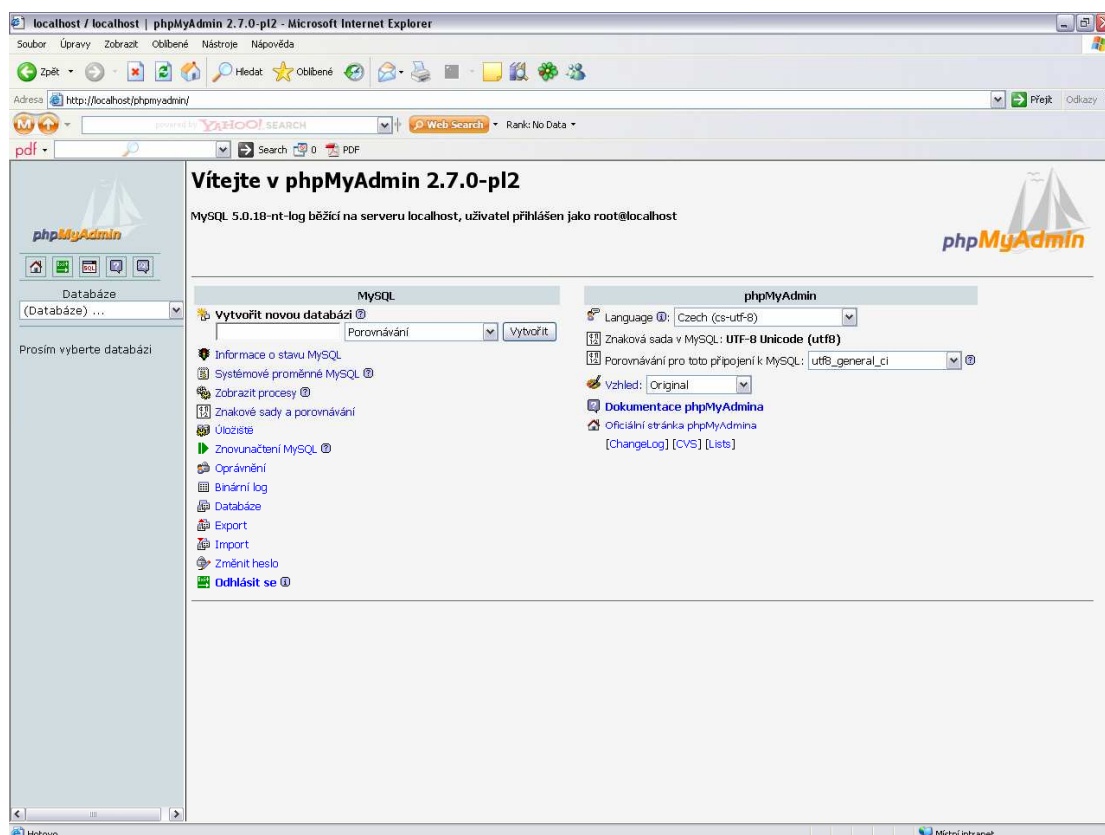
Tvorba MySQL databáze

Na domácím lokálním serveru se objevuje odkaz PHPmyadmin. Po kliknutí na tento odkaz se zobrazí autentizace (obrázek 6), kde je potřeba vložit uživatelské jméno (root) a heslo. Heslo bude opět shodné s tím, které bylo již použito při instalaci Apache. Uživatelské jméno je psáno malými písmenky.



Obrázek 6: Autentizace [zdroj: vlastní]

Tvorba nové databáze následuje po provedení autentizace. V místech, kde se nachází MySQL, se bude vytvářet novou databázi (bara) jako je tomu na obrázku 7. Pro lepší orientaci jsou zadány v názvu opět malá písmenka a případně podtržítka pro dělení slov. Po té se pouze dá pokyn k vytvoření. V této chvíli už je založena nová databáze na domácím lokálním serveru.



Obrázek 7: Tvorba databáze [zdroj: vlastní]

PSPad editor

Jako textový editor pro práci byl vybrán PSPad editor. Pokud není v počítači k dispozici, je potřeba jej také nainstalovat. PSPad editor je možné stáhnout z adresy [g].

Bude použita možnost instalátor 2.92 MB a uloží se. Jakmile je ukládání dokončeno, spustí se průvodce instalací.

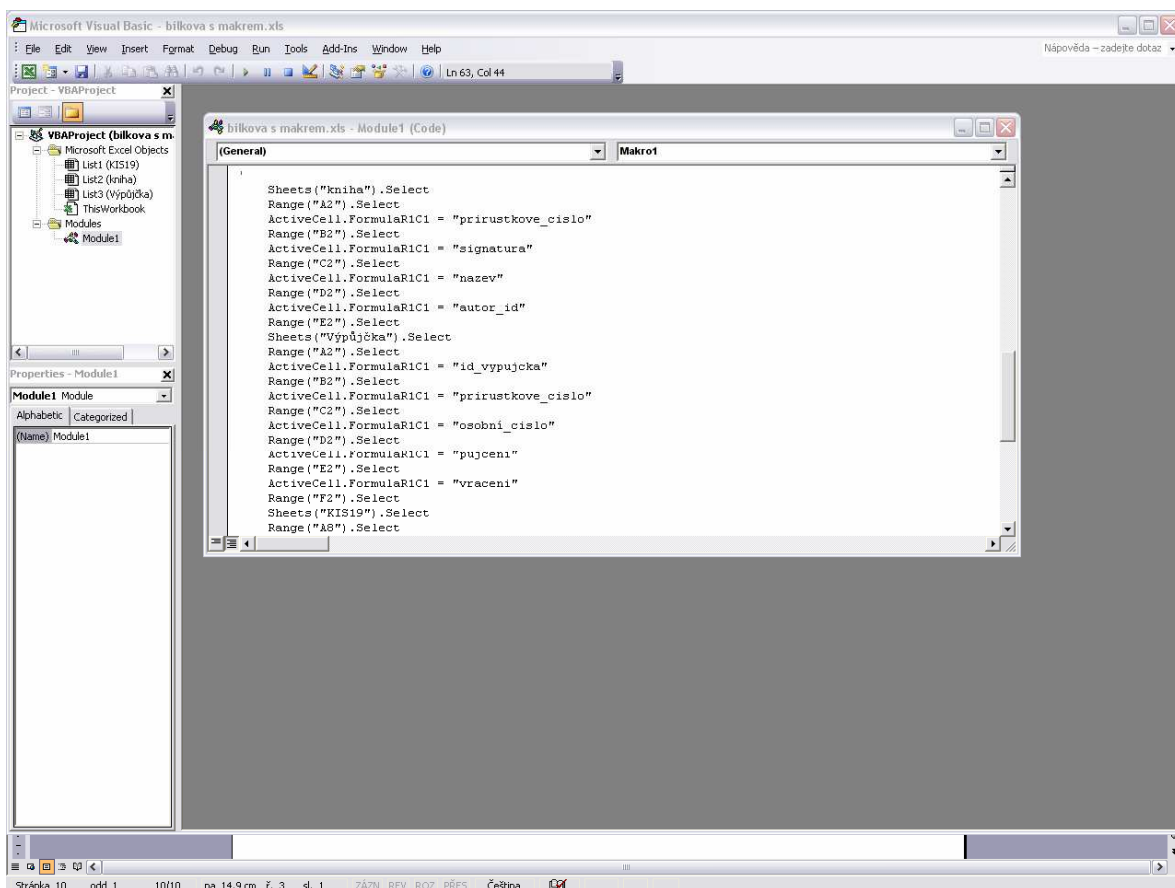
V počítači na místním disku existuje konfigurační soubor PHP.INI. Ikona souboru má podobu poznámkového bloku s ozubeným kolečkem. Po kliknutí pravým tlačítkem myši na tuto ikonu se volí možnost PSPad.

Jedná se o konfigurační program, který přesně udává to, co a jak bude php dělat a jak se bude ukládat. Program patří mezi konfigurační soubory apache2tírad. Nastavují se zde všechny globální vlastnosti apache a podobných „pomocníků“ při programování. Zelená část textu jsou poznámky a černá část textu jsou příkazy, které lze v konfiguračním souboru upravit. Po daných úpravách budou změny v dokumentu uloženy a soubor zavřen.

3.5 Naplnění databáze daty

Vstupní datové soubory byly vyhotoveny v Excelu. Pro každého vyučujícího byl samostatný datový soubor s údaji o vypůjčení knihy. Nejprve se ze vstupních dat vytvoří tabulky v souboru Excel, které budou shodné s tabulkami navržené databáze. To znamená, že se bude dbát na to, aby v každé vytvořené tabulce byly shodně uspořádané a pojmenované sloupce.

Úpravu vstupních dat v souboru Excel lze provést pomocí makra. Makro se vytváří několika způsoby. V tomto případě byl k vytvoření využit Microsoft Visual Basic. V nabídce Nástroje se zvolí příkaz Makro. Z nabídnutých možností je vybrán Editor jazyka Visual Basic. Při klepnutí na tuto možnost se Visula Basic otevře (obrázek 8). Z nabídky Vložit (Insert) se vybere příkaz Modul (Module). Do něho se vkládá kód (celý kód pro vytvoření makra je uveden v příloze). Jakmile je zápis makra dokončen, v nabídce Soubor (File) se vybere příkaz Konec a návrat do aplikace Microsoft Excel (Close and Return to Microsoft Excel).



Obrázek 8: Makra - Microsoft Visual Basic [zdroj: vlastní]

Pomocí makra jsou vytvořeny i tlačítka „Vygeneruj knihy a výpůjčky“ a tlačítka „Smazat vygenerovaná data“ (obrázek 9).

	A	B	C	D	E	F	G	H	I	J	K
1	Signatura	Přirůstkové	Čárový kód	Půjčeno	Vrátit do	Titul					
2	61305e	227265	32100087924	5.9.2005	5.10.2005	Databázové systémy					
3	65660	242406	32100105628	26.1.2006	17.9.2030	E-business and e-commerce management					
4	68340	249096	32100120061	17.1.2007	8.9.2031	Informační systémy v podnikové praxi					
5	68341	249098	32100120057	17.1.2007	8.9.2031	Velký průvodce infrastrukturou PKI a technologií elektronickéh					
6	66108	248467	32100103177	11.4.2007	11.5.2007	Podnikové procesy					
7	61280b	227401	32100087916	11.4.2007	11.5.2007	Ochrana dat					
8											
9											
10		Vygeneruj knihy a výpůjčky									
11			Smazat vygenerovaná data								
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											

Obrázek 9: Vytvořené makro [zdroj: vlastní]

Při stisku tlačítka „Vygeneruj knihy a výpůjčky“ se spustí naprogramované makro. Do předem připraveného listu „knihy“ bude vygenerováno přírůstkové číslo, signatura a název knihy. Tuto skutečnost je možné si prohlédnout na obrázku 10.

	A	B	C	D	E	F	G	H	I
1	prirustkove_cislo	signatura	nazev	autor_id					
2	227265	61305e	Databázové systémy						
3	242406	65660	E-business and e-commerce management						
4	249096	68340	Informační systémy v podnikové praxi						
5	249098	68341	Velký průvodce infrastrukturou PKI a technologií elektronického podpisu						
6	248467	66108	Podnikové procesy						
7	227401	61280b	Ochrana dat						
8									
9									
10									
11									
12									
13									
14									
15									

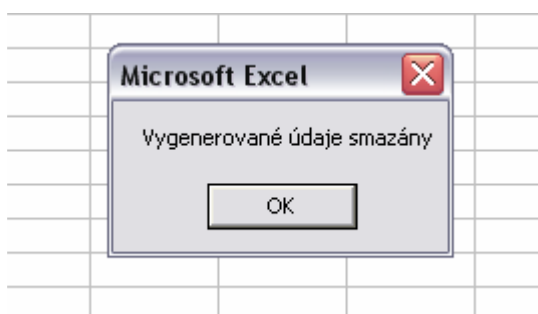
Obrázek 10: Vygenerované knihy [zdroj: vlastní]

Následující obrázek (obrázek 11) ukazuje stisk toho samého tlačítka. Je zde však patrné, že tentokrát se generovaly údaje o datu vypůjčení, datu vrácení, přírůstkovém čísle jednotlivých knih. Vygenerování proběhlo opět na předem připravený list „výpůjčka“.

	A	B	C	D	E	F	G	H	I
1	id_vypujck	prirustkove_cislo	osobni_cislo	pujceni	vraceni				
2		227265	KIS19	5.9.2005	5.10.2005				
3		242406	KIS19	26.1.2006	17.9.2030				
4		249096	KIS19	17.1.2007	8.9.2031				
5		249098	KIS19	17.1.2007	8.9.2031				
6		248467	KIS19	11.4.2007	11.5.2007				
7		227401	KIS19	11.4.2007	11.5.2007				
8			KIS19						
9									
10									
11									
12									
13									
14									
15									

Obrázek 11: Vygenerované výpůjčky [zdroj: vlastní]

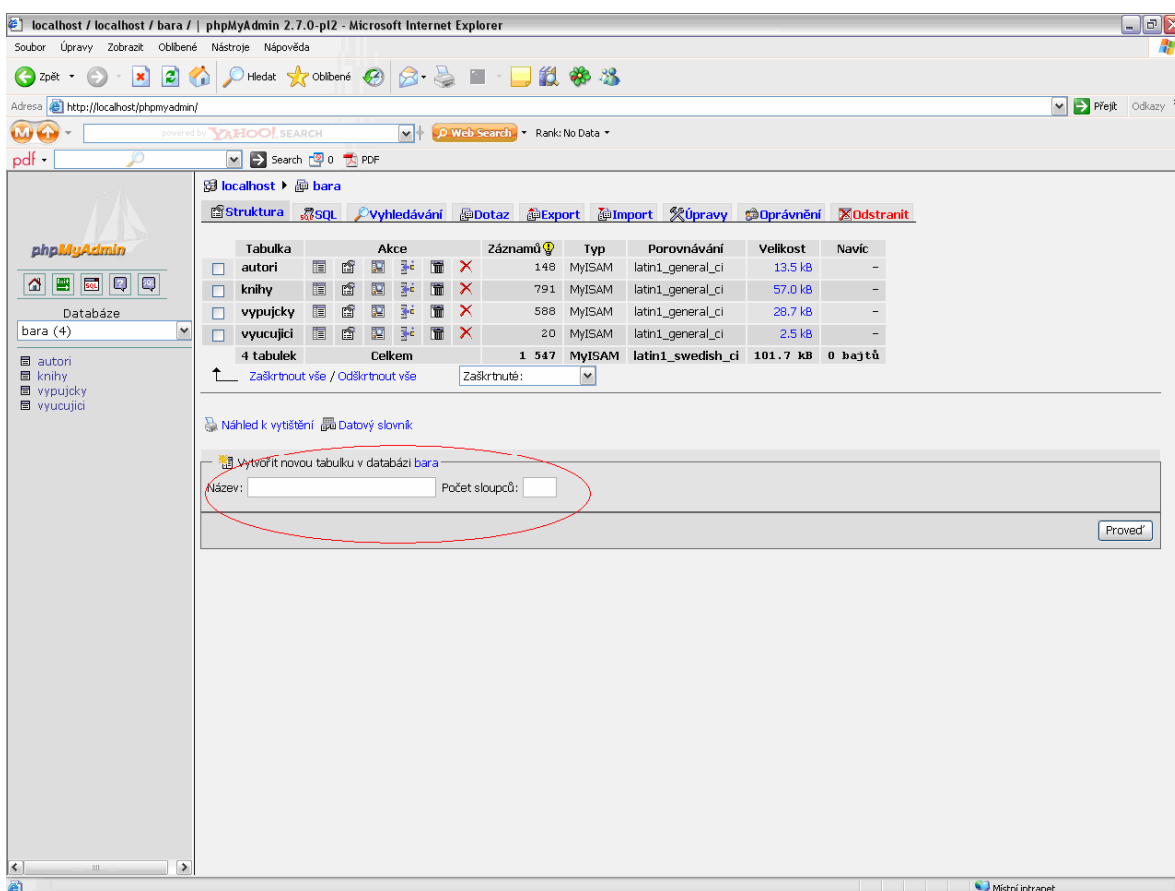
Při stisku tlačítka s názvem „Smazat vygenerovaná data“ budou všechna vygenerovaná data smazána a na monitoru se objeví informativní hláška jako na obrázku 12.



Obrázek 12: Smazání vygenerovaných údajů [zdroj: vlastní]

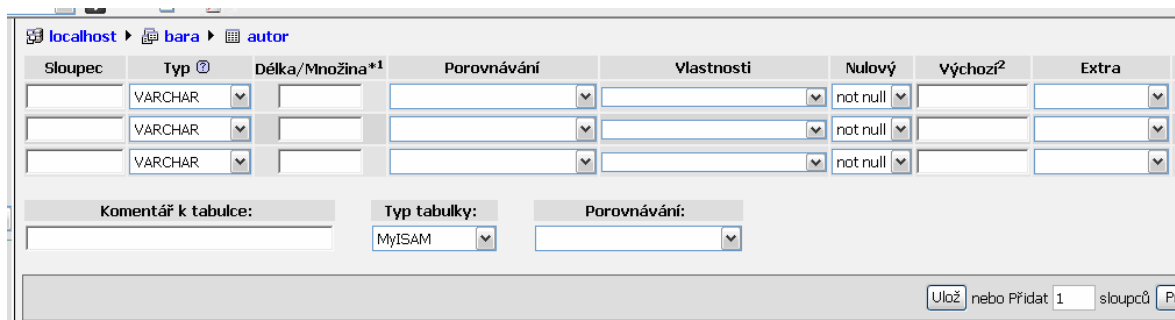
Dále se soubor uloží jako CSV dokument (oddělený středníkem). Soubor má tedy koncovku *.csv. Každá nově vytvořená tabulka má svůj samostatný soubor. Takto jsou data připravena k nahrání do databáze.

Ve vytvořené databázi se připraví tabulky pro nahrání dat. Na obrázku 13 je červeně zakroužkována část, do níž se doplní název tabulky a počet sloupců. Pak se stiskne tlačítko „Proved“ a tabulka je vytvořena. Tímto způsobem se vytvoří všechny potřebné tabulky.



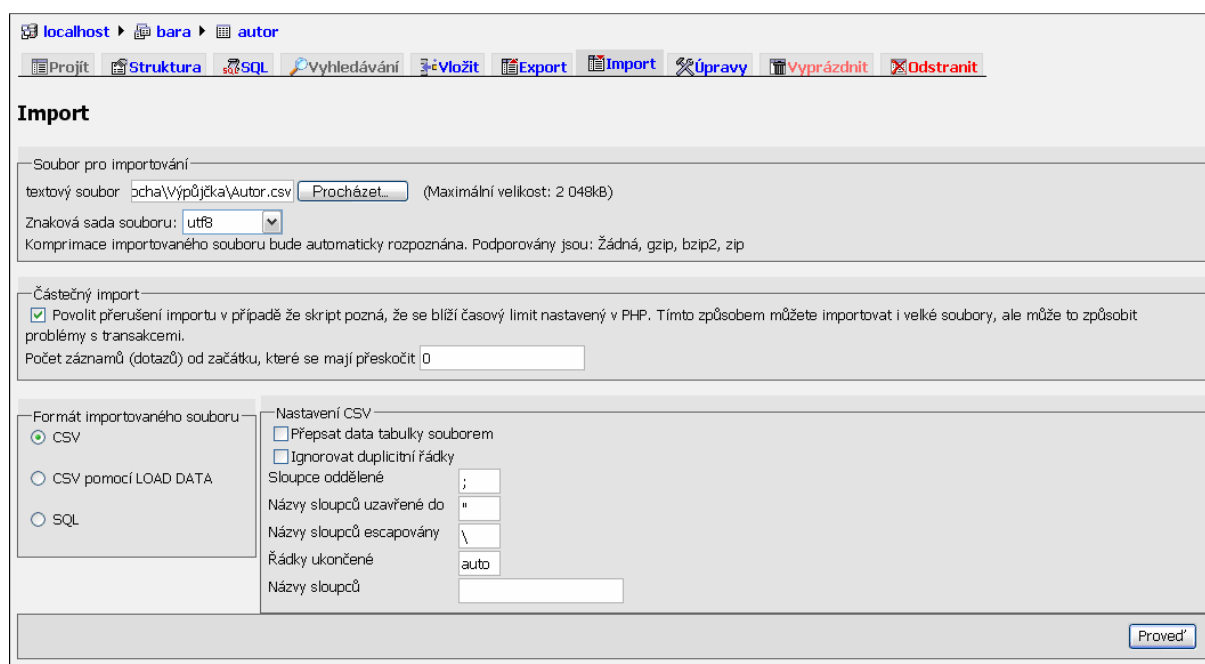
Obrázek 13: Tvorba tabulky [zdroj: vlastní]

V této fázi jsou sice tabulky vytvořeny, ale nejsou zde žádné sloupce pro nahrání dat. Po otevření tabulky se objeví pokyny pro vytvoření sloupců, jako je tomu na obrázku 14. Tady se určí název sloupce, datový typ, délka a primární klíč. Stiskem tlačítka „Ulož“ budou již tabulky kompletní a připraveny pro nahrání dat.



Obrázek 14: Vytvoření sloupců [zdroj: vlastní]

Data budou nahrány pomocí záložky „Import“. Zde se zvolí textový soubor pro importování, formát importovaného souboru nastaví jako CSV, překontroluje se nastavení CSV (obrázek 15). A stiskem tlačítka „Proved“ budou data nahrány do příslušné tabulky databáze.

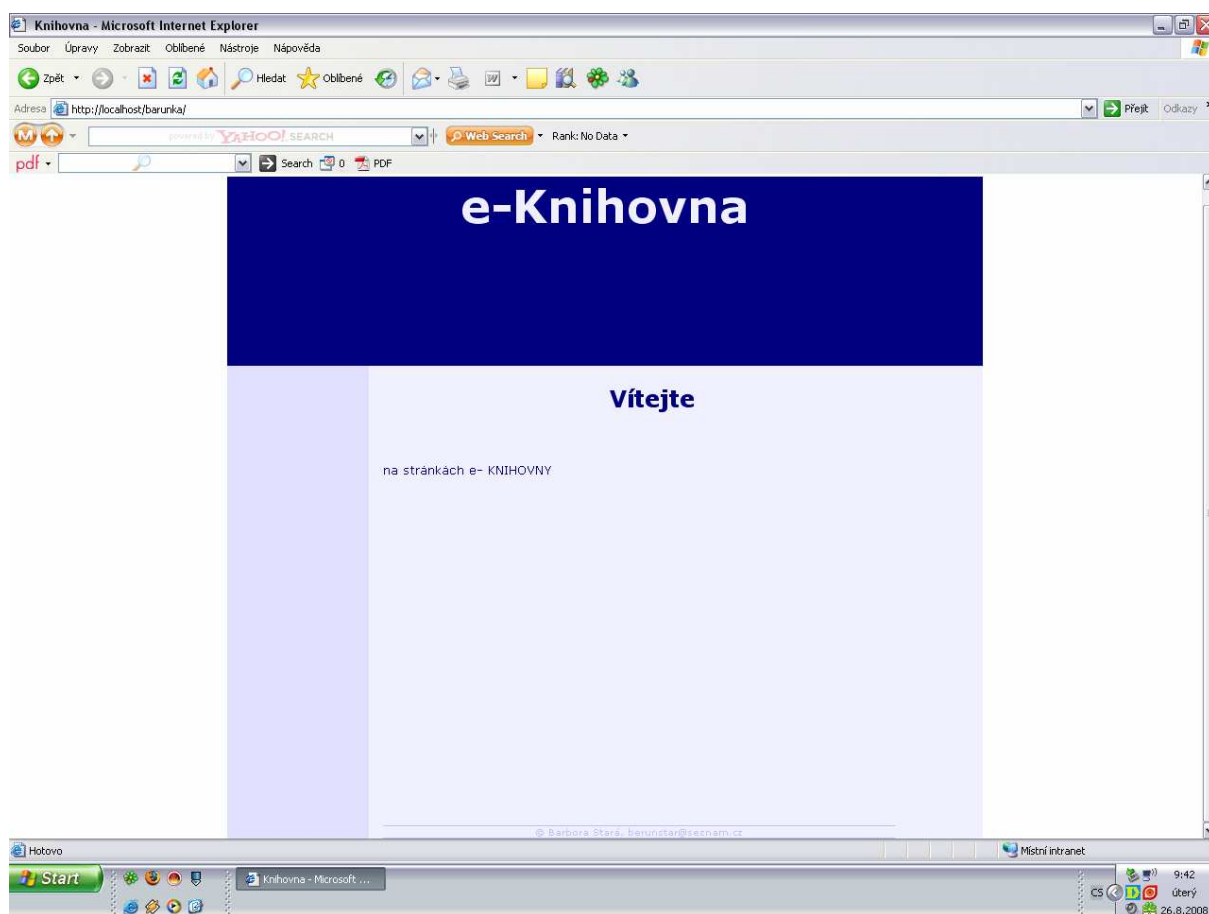


Obrázek 15: Nahrání dat [zdroj: vlastní]

3.6 Programování

Tato část se zabývá uvedením částí skriptu a vysvětlením, k čemu byla právě tato část použita. Výsledný program se skládá z několika programovacích jazyků. Byly použity HTML kódy, PHP kódy, kaskádové styly.

Pomocí HTML kódu a kaskádových stylů je vytvořena první stránka, která se načte po spuštění prohlížeče na příslušné adrese. Na obrázku 16 je možno vidět, jak stránka vypadá.



Obrázek 16: Hlavní stránka [zdroj: vlastní]

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN">  
<html>  
  <head>  
    <meta http-equiv="content-type" content="text/html;  
charset=windows-1250">  
    <title>Knihovna</title>  
    <meta name="author" content="Barbora Stará">  
    <meta name="description" content="stránka e-Knihovny">  
    <meta name="keywords" content="autor, název">  
    <link rel="stylesheet" type="text/css" href="style.css">  
  </head>  
<body>  
</body>  
</html>
```

Tento úryvek kódu je nejznámější částí. Jedná se o základní strukturu html dokumentu, který má po uložení příponu `***.html`. Vyskytují se zde tagy pro definování HTML dokumentu,

hlavičky a těla dokumentu. V hlavičce se nachází jednotlivé meta tagy. Důležitým tagem, jenž se v hlavičce také objevuje je tak link. Ten slouží pro odkaz na dokument definující kaskádové styly. Do těla dokumentu, které obsahuje veškerý zobrazovaný obsah stránky, jsou pak vloženy další příkazy.

```
<!--Hlavní rám na středu okna prohlížeče-->
<div id="hlavni">
<!--Horní rám-->
  <div id="horni">
<h1>e-Knihovna</h1>
  </div>
<!--Průběžně se obměňuje na všech stránkách, stránky jsou
statické a nejsou zde použity iframe-->
  <div id="levy">
  </div>
<!--obsah-->
  <div id="obsah">
    <center><h2>Vítejte</h2></center>
    <p>&nbsp;</p>
    <p>na stránkách e- <span id="verzalky">knihovny</span></p>
    <!--autor a e-mail-->
    <div id="paticka">
      &copy; Barbora Stará, <a
href="mailto:berunstar@seznam.cz">berunstar@seznam.cz</a>
    </div>
  </div>
</div>
```

Obsah těla dokumentu je složen převážně z poznámek a tagů <div></div>. Div je jakýsi blokový element, který před sebou i za sebou udělá konec řádku. Velmi často bývá používán ve spojení s kaskádovými styly, jako tomu je i v tomto případě. Jak je možno vidět, každý počáteční tag <div> obsahuje id="". Pomocí id se odkazujeme do nadefinovaných kaskádových stylů. Podobným tagem pro definování stylů je tag . Span je řádkový element a smí být použit pouze na řádku.

```
/* CSS Document */
body {font-family: Verdana, Arial, sans-serif;
      font-size: 80%;
      margin: 0px;
      padding: 0px;
      border: 0px none;
      background: #ffffff;
      color: blue;
      text-align: center;
      left: auto;
      right: auto;
```

}

Tímto kódem je nadefinováno tělo HTML dokumentu. Je zde určeno písmo i jeho velikost. Margin určuje šířku vnějšího okraje prvku a padding šířku vnitřního okraje prvku. Border je definovaný jako žádný a znamená určení rámečku. V poslední řadě je zde definováno pozadí, barva, zarovnání textu a x-ové souřadnice umístění prvku.

Podobným způsobem jsou definovány rámy (#hlavni, #horni, #levy) i obsah vytvořený na stránce. Při definování kaskádových stylů u těchto částí se objevují navíc i vlastnosti pro určení výšky, šířky a centrování.

Na obrázku 12 je vyobrazena patička stránky, v níž jsou obsaženy informace o autorovi. Zde se v kódu kromě definování barev, pozadí, rámečku a zarovnání také objevuje link, hover a visited. Link slouží jako odkaz pro e-mailovou adresu, hover reaguje na přejetí kurzoru. V tomto případě dochází při najetí kurzoru na e-mailovou adresu k jejímu znázornění jako odkaz, což je právě vidět na obrázku 17. Visited znamená už navštívený odkaz.



Obrázek 17: Patička [zdroj: vlastní]

Na konci dokumentu kaskádových stylů jsou ještě nadefinované verzálky a nadpisy první, druhé a třetí úrovně.

```
h1{color: #f0f0ff;
    font-size: xx-large;
        font-weight: bold;
            font-style: normal;
    }
h2, h3 {color: #000080;
        font-weight: bold;
            font-style: normal;
    }
#verzalky {text-transform: uppercase;
    }
```

SQL dotazy

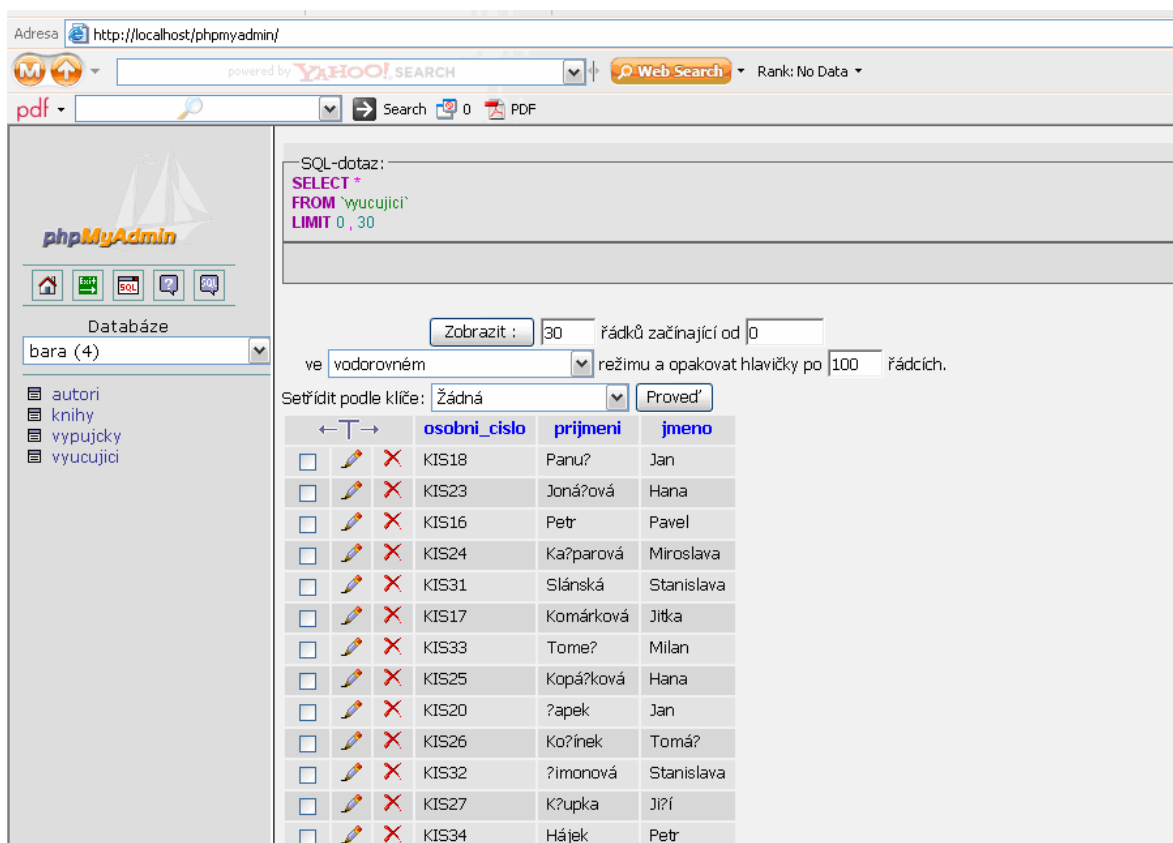
```
--  
-- Struktura tabulky `autori`  
--  
  
CREATE TABLE IF NOT EXISTS `autori` (  
  `id` int(11) NOT NULL,  
  `prijmeni` varchar(255) collate latin1_general_ci NOT NULL,  
  `jmeno` varchar(255) collate latin1_general_ci NOT NULL,  
  FULLTEXT KEY `ft` (`jmeno`,`prijmeni`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1  
COLLATE=latin1_general_ci;
```

Data do databáze mohou být vložena i pomocí SQL dotazu. Tímto kódem se vytvoří v databázi tabulka pro autory. Postup bude shodný i při tvorbě ostatních tabulek pro knihy, vyučující i výpůjčky. Následující kód udává způsob vypisování dat pro určitou vytvořenou tabulku.

```
--  
-- Vypisuji data pro tabulku `autori`  
--  
  
INSERT INTO `autori` (`id`, `prijmeni`, `jmeno`) VALUES  
(1, 'Abe', 'Shigeo'),  
(2, 'Adamczyk', 'Gregor'),  
(3, 'Adamec', 'František'),  
(4, 'Albright', 'S. Christian'),
```

V této chvíli je připravena databáze, tabulky v databázi a jsou naplněny daty (obrázek 18). Je připraven i vzhled stránky. A tedy je čas na samotné programování v PHP. Jako první a důležitá věc je, aby byly nadefinovány potřebné údaje k připojení k databázi. Definuje se server, uživatel, heslo a název databáze.

```
define('HOST','localhost');  
define('USER','root');  
define('PASS','argo8585');  
define('DB','bara');
```



Obrázek 18: Naplněná databáze [zdroj: vlastní]

Jelikož vyhledávání v databázi bude zřízeno pomocí jednoduchého formuláře, je součástí kódu i jednoduchý HTML kód, který vytvoří podobu formuláře. Následující úryvek kódu vytváří formulář pro vyhledávání v databázi. Je zde použito tagů pro vytvoření tabulky, aby byl zachován nějaký pravidelný a hezcí vzhled. Pro vypisování požadované hledané hodnoty je jako vstup použito jednoduché textové pole. Formulář zároveň obsahuje tlačítko pro odeslání údajů, které chceme vyhledat. V kódu se také objevuje skryté pole. To se ve formuláři vůbec nezobrazí. Toto pole slouží pro uchování stavové informace, jež je odeslána s vyplněným formulář skriptu. Formulář je možno vidět v manuálu na obrázku 13.

```

<div id="formular">
<table>
  <form action="index.php" method=get>
    <tr><td><font color=#000080>Autor:</font></td></tr>
    <tr><td><input type="text" name="autor" id="autor"
value="" /></td></tr>
    <tr><td><font color=#000080>Název
knihy:</font></td></tr>
    <tr><td><input type="text" name="nazev_knihy"
id="nazev_knihy" value="" /></td></tr>

```

```

                <tr><td><font
color=#000080>Signatura:</font></td></tr>
                <tr><td><input type="text" name="signatura" id="signatura"
value="" /></td></tr>
                <tr><td><font color=#000080>Přírůstkové
číslo:</font></td></tr>
                <tr><td><input type="text" name="prirustkove_cislo"
id="prirustkove_cislo" value="" /></td></tr>
                <input type="hidden" name="odeslano" value="true" />
                <tr><td colspan="2"><input type="submit"
value="Vyhledej" /></td></tr>
        </form>
    </table>
</div>

```

Často se ve skriptu objevuje příkaz echo. Například echo \$form;. Tento příkaz slouží k vypsaní jednoho či více výrazů. Vypíše hodnotu všech výrazů, které jsou za příkazem uvedeny. V tomto případě je uveden pouze jeden a to formulář.

V následující části je používán příkaz isset, který slouží ke zjištění, zda je proměnná inicializována. Pokud vlastně proměnná existuje a obsahuje nějakou hodnotu, vrací tento příkaz true. Isset je použit za podmínkou if, což znamená jestliže. Dalším příkazem je příkaz trim. Používá se pro odstranění mezer a tabulátorů ze začátku a konce řetězce. Jinak řečeno jsou odstraněny všechny netisknutelné znaky.

```

if(isset($_REQUEST['odeslano']) AND
$_REQUEST['odeslano']=='true')
{
    if(isset($_REQUEST['autor']) AND trim($_REQUEST['autor']
!= ''))
    {
        $autor = $_REQUEST['autor'];
        $kontrola++;
    }
}

```

V této části kódu dochází ke spojení se serverem a databází. A následnému vyhledávání podle toho, co bylo ve formuláři zadáno. Spojení je realizováno pomocí příkazu mysql_connect a příkaz mysql_select_db slouží pro výběr databáze, ve které se bude hledat.

```

{
    $con=mysql_connect(HOST, USER, PASS) or
die(mysql_error());
    mysql_select_db(DB,$con) or die(mysql_error());
}

```

```

        if($autor != '')
        {
            $query = 'SELECT v.osobni_cislo, v.prijmeni,
v.jmeno, vp.pujceni, vp.vraceni, k.signatura,
k.prirustkove_cislo, k.nazev, a.prijmeni, a.jmeno FROM
vyucujici v, vypujcky vp, knihy k, autori a WHERE (a.jmeno
like \''.$autor.'\' or a.prijmeni like \''.$autor.'\' ) AND
k.autor_id=a.id AND vp.prirustkove_cislo=k.prirustkove_cislo
AND vp.osobni_cislo=v.osobni_cislo;';
        }

```

V kódu je začleněn také příkaz pro zjištění počtu záznamů, shodujících se s hledaným výrazem. K tomu je využit příkaz `mysql_rows`. Pokud je hledání rovno nule, vypíše se informativní hláška, že nebyly nalezeny žádné půjčené knihy. V opačném případě se vypíše do podoby tabulky údaje o vyučujícím, datu vypůjčení i vrácení a signatura, název i autor knihy. K tomu je ve skriptu použit příkaz `mysql_fetch_array`. Před ním je zapsán ještě příkaz `while`, který slouží k opakování cyklu do té doby, dokud bude platit příslušná podmínka.

```

        $res = mysql_query($query,$con) or
die(mysql_error());
        if(mysql_num_rows($res) ==0)
        {
            echo '<div id="nalezene"><h1>Nenalezeny žádné
půjčené knihy.</h1></div>';
        }
        else
        {
            echo '<div id="nalezene">'.
            '<table>'.
            '<tr><th>osobní číslo
</th><th>jméno</th><th>příjmeni</th><th>vypůjčení</th><th>vrác
ení</th><th>signatura knihy</th><th>přírůstkové
číslo</th><th>název knihy</th><th>autor knihy</th></tr>';
            while($row = mysql_fetch_array($res))
            {
                echo '<tr><td>' . $row['vyucujici.osobni_cislo'] . '</td><td>' .
                $row['vyucujici.jmeno'] . '</td><td>' . $row['vyucujici.prijmeno']
                . '</td><td>' . $row['vypujcky.pujceni'] . '</td><td>' . $row['vypujc
                ky.vraceni'] . '</td><td>' . $row['knihy.signatura'] . '</td><td>' . $
                row['knihy.prirustkove_cislo'] . '</td><td>' . $row['knihy.nazev']
                . '</td><td>' . $row['autori.prijmeni'] . '
                ' . $row['autori.jmeno'] . '</td></tr>';
            }
            echo '</table>'.
            '</div>';
        }

```

ZÁVĚR

Bakalářská práce byla věnována problematice vyhledávání v databázi knihovny. Cílem bylo navrhnout a zrealizovat příslušnou databázi a vytvořit vhodný program pro vyhledávání v ní. Na úvod bylo uvedeno lehké seznámení s touto problematikou v dnešní době a pokračovalo se bližším obeznámením s programovacím jazykem PHP, který byl pro realizaci této práce zvolen.

Dále v práci nechybí ani teoretické seznámení s databázemi a postupem jejich tvorby. Tomuto je věnován začátek druhé kapitoly. Nechybí zde ani seznámení s ER – diagramem, jenž je prostředkem návrhu databáze. Ve druhé kapitole jsou zmíněny základní pojmy, které je při tvorbě návrhu databáze nutné znát a které by nikomu, kdo se tímto problémem bude někdy zabývat, neměly být cizí.

Poslední kapitolou bakalářské práce je popis praktické části. Hned na úvod jsou vloženy podmínky, jež téma „Návrh a realizace knihovny“ doprovázejí. Dále se pokračovalo využitím teoretických poznatků ze druhé kapitoly pro tvorbu návrhu databáze. V realizaci návrhu nesmí chybět postupný návrh ER – diagramu, tabulky entit a atributů ani jejich normalizace. Závěr třetí části je věnován již instalaci potřebných nástrojů pro programování a samotnému programování. Do textu jsou vloženy části zdrojového kódu vytvořeného za účelem realizace vyhledávání v databázi knihovny přes internet.

V příloze bakalářské práce je uveden návod na obsluhu vytvořené knihovny. Dále příloha obsahuje také celé skripty kódů, jak jsou zapsány v jednotlivých souborech.

POUŽITÁ LITERATURA

- [1] BRÁZDA, Jiří. *PHP 4 - učebnice základů jazyka*. Praha : Grada Publishing, 2002. 224 s. ISBN 80-247-0442-0.
- [2] *Databáze a jazyk SQL : Relační databáze* [online]. [2000] [cit. 2008-08-13]. Dostupný z WWW: <<http://interval.cz/clanky/databaze-a-jazyk-sql/>>. ISSN 1212-8651
- [3] *Databázové systémy* [online]. [2005] , 1.11.2005 [cit. 2008-08-13]. Dostupný z WWW: <www.sweb.cz/Martin.Matuna/Download/DBS.doc>.
- [4] HARTMANN, Dr. Norbert. *MS-FoxPro 2.5 pro Windows*. Brno : Proxima, 1994. 637 s. ISBN 3-89362-256-X.
- [5] *Jak objektové databáze podporují škálovatelnost poprvé : Relační databáze a ER modely* [online]. 2004 [cit. 2008-08-13]. Dostupný z WWW: <<http://www.dbsvet.cz/view.php?cisloclanku=2006051501>>.
- [6] *Konceptuální datový model* [online]. [2004] [cit. 2008-08-13]. Dostupný z WWW: <<http://www.owebu.cz/databaze/vypis.php?clanek=298>>. ISSN 1214-7958.
- [7] KOSEK, Jiří. *PHP - tvorba interaktivních internetových aplikací*. Praha : Grada Publishing, 1999. 492 s. ISBN 80-7169-373-1.
- [8] KRIST, Pavel. *Databáze* [online]. 2007 [cit. 2008-08-13]. Dostupný z WWW: <<http://www.krist.euweb.cz/page32.html>>.
- [9] *Metodiky a Notace-Select : Select Perspective* [online]. [2008] , 03.02.2008 [cit. 2008-08-13]. Dostupný z WWW: <<http://objekty.vse.cz/Objekty/MetodikyANotace-Select>>.
- [10] *Možnosti PHP : Spolupráce s databázemi* [online]. [2008] , 1. července 2008 [cit. 2008-08-13]. Dostupný z WWW: <<http://www.jakpsatweb.cz/php/moznosti-php.html>>.
- [11] PÍSEK, Slavoj. *Databáze v Accessu v rekordním čase*. Praha : Grada Publishing, 2003. 88 s. ISBN 80-247-0572-9.

- [12] PANUŠ, J. *Přednášky z předmětu Databázové systémy*. Univerzita Pardubice, zimní semestr 2006/2007
- [13] *PSPad : Download PSPadu* [online]. 2001-2008 [cit. 2008-08-13]. Dostupný z WWW: <<http://www.pspad.com/cz/download.php>>.
- [14] *Softwarové inženýrství : Entity Relationship Diagram* [online]. 2008 [cit. 2008-08-13]. Dostupný z WWW: <<http://dudka.cz/studyIUS>>.
- [15] *Teorie relačních databází : Normalizace* [online]. 2005-2006 [cit. 2008-08-13]. Dostupný z WWW: <<http://www.manualy.net/article.php?articleID=13>>.
- [16] *Teorie relačních databází : Relační model dat* [online]. 2005-2006 [cit. 2008-08-13]. Dostupný z WWW: <<http://www.manualy.net/article.php?articleID=9>>.
- [17] *WebML -datové modelování : Úvod do datového modelování* [online]. [2004] [cit. 2008-08-13]. Dostupný z WWW: <<http://interval.cz/clanky/webml-datove-modelovani/>>. ISSN 1212-8651.
- [18] *Základy relačních databází* [online]. 2006 [cit. 2008-08-13]. Dostupný z WWW: <<http://iss.unas.cz/ids/04.pdf>>.

SEZNAM OBRÁZKŮ

Obrázek 1: Diagram vytipovaných entit [zdroj: vlastní]	22
Obrázek 2: První analytické schéma [zdroj: vlastní]	23
Obrázek 3: Dekompozice vztahu výpůjčka [zdroj: vlastní].....	24
Obrázek 4: ER - diagram [zdroj: vlastní].....	26
Obrázek 5: Autostrt [zdroj: vlastní]	30
Obrázek 6: Autentizace [zdroj: vlastní]	31
Obrázek 7: Tvorba databáze [zdroj: vlastní].....	31
Obrázek 8: Makra - Microsoft Visual Basic [zdroj: vlastní]	33
Obrázek 9: Vytvořené makro [zdroj: vlastní]	33
Obrázek 10: Vygenerované knihy [zdroj: vlastní].....	34
Obrázek 11: Vygenerované výpůjčky [zdroj: vlastní]	34
Obrázek 12: Smazání vygenerovaných údajů [zdroj: vlastní]	35
Obrázek 13: Tvorba tabulky [zdroj: vlastní].....	35
Obrázek 14: Vytvoření sloupců [zdroj: vlastní]	36
Obrázek 15: Nahrání dat [zdroj: vlastní]	36
Obrázek 16: Hlavní stránka [zdroj: vlastní].....	37
Obrázek 17: Patička [zdroj: vlastní]	39
Obrázek 18: Naplněná databáze [zdroj: vlastní].....	41
Obrázek 19: Formulář pro vyhledávání [zdroj: vlastní]	48
Obrázek 20: Vkládání údaje [zdroj: vlastní].....	49
Obrázek 21: Úspěšné hledání [zdroj: vlastní].....	49
Obrázek 22: Neúspěšné hledání [zdroj: vlastní]	49
Obrázek 23: Zpráva e-mailem [zdroj: vlastní].....	50

SEZNAM TABULEK

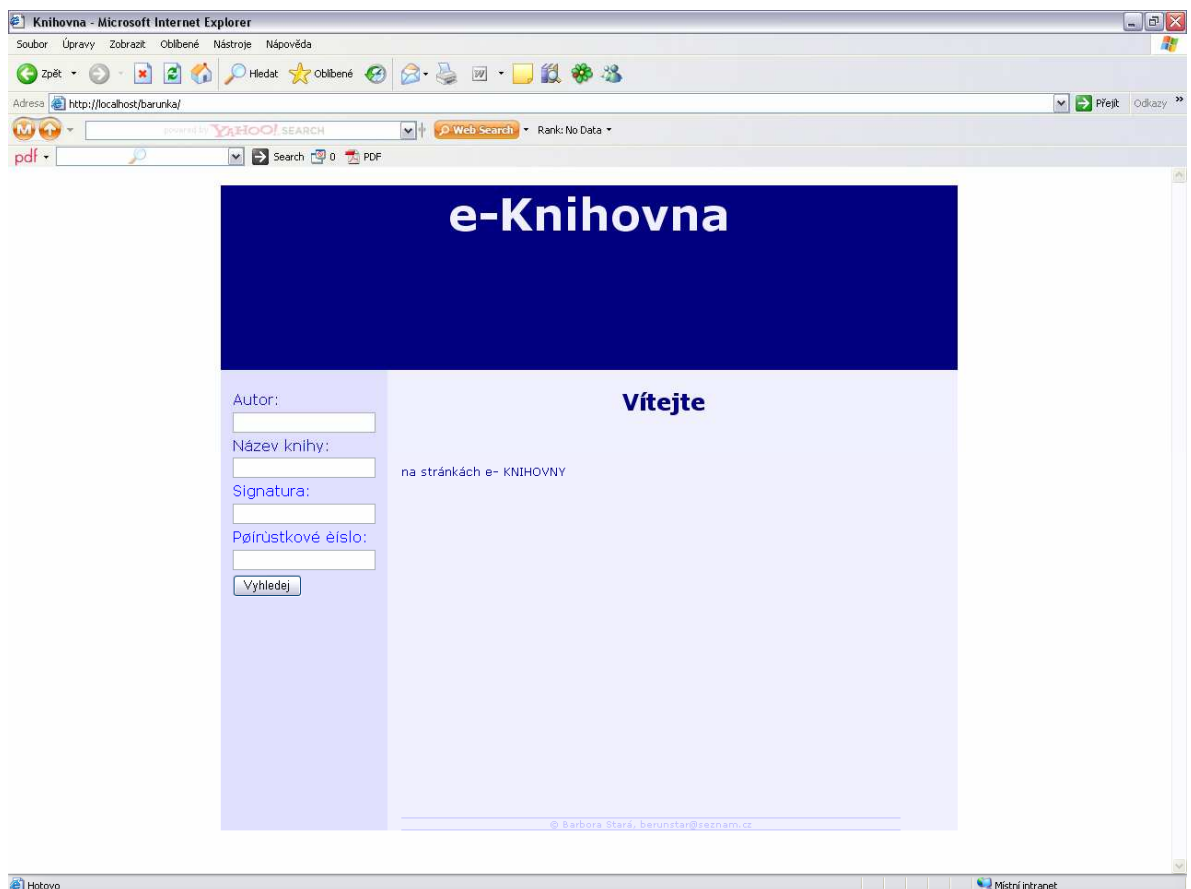
Tabulka 1: Přehled datových typů SQL [zdroj: 7].....	11
Tabulka 2: Vytipování entit a atributů [zdroj: vlastní]	22
Tabulka 3: Definované entity a atributy [zdroj: vlastní].....	24
Tabulka 4: Tabulka entit a atributů [zdroj: vlastní]	27

PŘÍLOHY

Manuál

Na stránce e-Knihovny je v levém sloupci zobrazen jednoduchý formulář, který obsahuje čtyři textová políčka a jedno tlačítko. Do každého textového pole lze vpisovat slovo popřípadě číselný údaj, který chceme v databázi vyhledat.

První pole má předepsáno „Autor“. Zde se napíše požadavek na hledání autora podle jeho příjmení. Druhé pole je pro vyhledávání knihy podle jejího názvu. Další pole slouží k vyhledávání podle signatury, jež má každá kniha jinou. A poslední políčko je určené k vepsání přírůstkového čísla patřičné knihy, kterou chceme vyhledat. Podoba formuláře je patrná na obrázku 19. Do formuláře zapíšeme vždy jen jeden klíčový údaj, podle něhož chceme hledat. Po té stačí pouze stisknout tlačítko „Vyhledej“ a program začne procházet databázi.



Obrázek 19: Formulář pro vyhledávání [zdroj: vlastní]

Do formuláře se zadává vždy pouze jeden hledaný výraz. Jakmile je jeden výraz zadán, ostatní políčka se stanou nepřístupná a nelze již do nich vkládat žádné znaky (obrázek). Po té dochází ke spojení s databází a požadovaný výraz bude vyhledán.

Autor:

 Název knihy:

 Signatura:

na stránkách e- KNIHOVNY

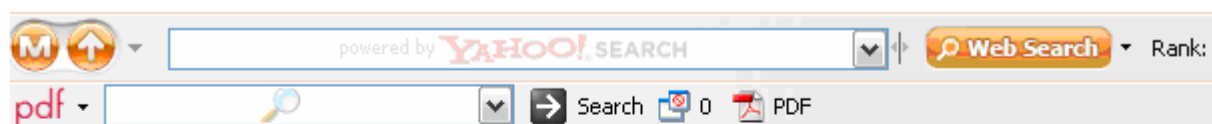
Obrázek 20: Vkládání údaje [zdroj: vlastní]

Existuje-li jedna nebo více knih, určených zadaným hledaným výrazem, objeví se na stránce tabulka s patřičnými údaji (obrázek 20). V tabulce bude osobní číslo, jméno a příjmení vyučujícího, který by měl mít v daném okamžiku požadovanou knihu půjčenou. Dále se v tabulce objeví údaje o datu vypůjčení a datu vrácení knihy, její signatura, název a autor knihy.

osobní číslo	jméno	příjmení	vypůjčení	vrácení	signatura knihy	pořizovací číslo	název knihy	autor knihy
KIS20	Jan	Ěapek	2003-12-01	2028-07-22	62943	231940	Algoritmy	Novák Vilém

Obrázek 21: Úspěšné hledání [zdroj: vlastní]

Může se však také stát, že hledanému výrazu nebude odpovídat žádný z řádků v databázi. Potom nebudou nalezeny žádné údaje a na stránce se vypíše chybová hláška: „Nenalezeny žádné půjčené knihy“ (obrázek 21).

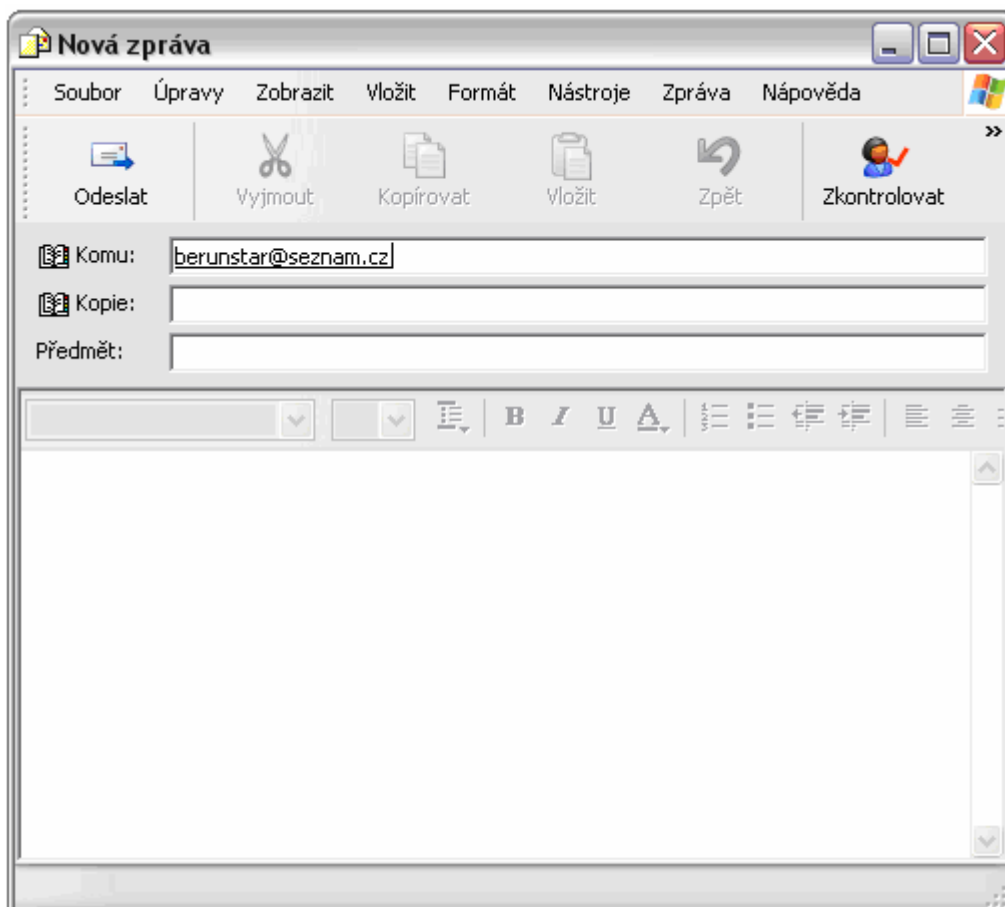


Nebyly nalezeny žádné půjčené knihy.

Obrázek 22: Neúspěšné hledání [zdroj: vlastní]

Ještě jedna malá poznámka na závěr. Dole, v patičce stránky, je vypsáno jméno a email. V případě nějakých dotazů stačí kliknout myší na email a otevře se vám dialogové okno,

v němž už bude předepsaná emailová adresa. To je zachyceno níže na obrázku 22. Napíšete svůj dotaz a odešlete.



Obrázek 23: Zpráva e-mailem [zdroj: vlastní]

Index.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html;
charset=windows-1250">
    <title>Knihovna</title>
    <meta name="author" content="Barbora Stará">
    <meta name="description" content="stránka e-Knihovny">
    <meta name="keywords" content="autor, název">
    <script type="text/javascript" src="jquery.js"></script>
    <script type="text/javascript" src="script.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <!--Hlavní rám na středu okna prohlížeče-->
    <div id="hlavni">
```

```

<!--Horní rám-->
  <div id="horni">
<h1>e-Knihovna</h1>
  </div>
<!--Průběžně se obměňuje na všech stránkách. Stránky jsou
statické, nejsou použity iframe-->
  <div id="levy">
    <div id="formular">
<table>
  <form action="index.php" method=get>
    <tr><td><font color=#000080>Autor:</font></td></tr>
    <tr><td><input type="text" name="autor" id="autor"
value="" /></td></tr>
    <tr><td><font color=#000080>Název
knihy:</font></td></tr>
    <tr><td><input type="text" name="nazev_knihy"
id="nazev_knihy" value="" /></td></tr>
    <tr><td><font
color=#000080>Signatura:</font></td></tr>
    <tr><td><input type="text" name="signatura" id="signatura"
value="" /></td></tr>
    <tr><td><font color=#000080>Přírůstkové
číslo:</font></td></tr>
    <tr><td><input type="text" name="prirustkove_cislo"
id="prirustkove_cislo" value="" /></td></tr>
    <input type="hidden" name="odeslano" value="true" />
    <tr><td colspan="2"><input type="submit"
value="Vyhledej" /></td></tr>
  </form>
</table>
    </div>
  </div>
<!--obsah-->
  <div id="obsah">
    <center><h2>Vítejte</h2></center>
    <p>&nbsp;</p>
    <p>na stránkách e- <span id="verzalky">knihovny</span></p>

    <!--autor a e-mail-->
    <div id="paticka">
      &copy; Barbora Stará, <a
href="mailto:berunstar@seznam.cz">berunstar@seznam.cz</a>
    </div>
  </div>

</div>
</body>
</html>

```

Style.css

```
/* CSS Document */
body {font-family: Verdana, Arial, sans-serif;
      font-size: 80%;
      margin: 0px;
      padding:0px;
      border: 0px none;
      background: #ffffff;
      color:blue;
      text-align:center;
      left:auto;
      right:auto;
    }
#hlavni {font-size: 13px;
        font-family: Verdana, Arial, sans-serif;
        margin: 20px 0px 20px -400px;
        padding:0px;
        border: 0px;
        min-height: 700px; /*pro standardní prohlížeče*/
        _height: 700px;    /*pro IE, jako min-height*/
        height: auto;
        width: 800px;
        margin-left:auto;/*centrování firefox*/
        margin-right:auto;/*centrování firefox*/
        color: #0000ff;
        background: #F0F0FF;
    }
#horni {padding: 0px 27px;
        color: #f0f0ff;
        background: #000080;    /*zmenit na obrazok*/
        height: 200px;
    }
#levy {margin: 0px;
        padding-top: 20px;
        padding-left: 10px;
        padding-right: 10px;
        border: 0px;
        text-align: left;
        float: left;
        left: 0px;
        color: #ffffff;
        background-color: #E0E0FF;
        width: 150px;
        /* _width: 150px; */
        height: auto;
        min-height: 500px;
        _height: 500px;/* v IE, min-height*/
    }
#obsah {position: relative;
        left: 0px;
```

```

right:auto;
float: left;
text-align:left;
margin: 0px 0px 0px 0px;
padding-top: 20px;
padding-right: 15px;
padding-left: 15px;
border: 0px;
color: #000080;
background-color: #F0F0FF;
width: 600px;
    /* _width: 565px; */
height: auto;
min-height: 500px;
_height: 500px;
}
#paticka {margin: 0px;
padding: 0px;
clear: both;
color: #C0C0ff;
background-color: #F0F0FF;
border-top: solid 1px #C0C0ff;
border-bottom: solid 1px #C0C0ff;
text-align: center;
font-size:80%;
position: absolute;
bottom: 0px;
width: 95%;
}
#paticka a:link {border: 0px;
padding: 0px;
margin: 0px;
color: #C0C0ff;
text-decoration: none;
}
#paticka a:hover {border: 0px;
padding: 0px;
margin: 0px;
color: #000080;
text-decoration: none;
}
#paticka a:visited {border: 0px;
padding: 0px;
margin: 0px;
color: #C0C0ff;
text-decoration: none;
}
h1{color: #f0f0ff;
font-size: xx-large;
font-weight: bold;
font-style: normal;

```



```

    }
h2, h3 {color: #000080;
        font-weight: bold;
        font-style: normal;
    }
#verzalky {text-transform: uppercase;
           }

```

Index.php

```

<?php

define('HOST','localhost');
define('USER','root');
define('PASS','argo8585');
define('DB','bara');

$head='<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">'.
'<html>'.
'<head>'.
'<title>Knihovna</title>'.
'<meta http-equiv="Content-Type" content="text/html;
charset=windows-1250">'.
'<script type="text/javascript" src="jquery.js"></script>'.
'<script type="text/javascript" src="script.js"></script>'.
'</head>'.
'<body>';

$foot='</body>'.
'</html>';

echo $head;

$autor='';
$nazev_knihy='';
$signatura='';
$prirustkove_cislo='';
$kontrola=0;

if(isset($_REQUEST['odeslano']) AND
$_REQUEST['odeslano']=='true')
{
    if(isset($_REQUEST['autor']) AND trim($_REQUEST['autor']
!= ''))
    {
        $autor = $_REQUEST['autor'];
        $kontrola++;
    }
}

```

```

        if(isset($_REQUEST['nazev_knihy']) AND
trim($_REQUEST['nazev_knihy'] != ''))
        {
            $nazev_knihy = $_REQUEST['nazev_knihy'];
            $kontrola++;
        }
        if(isset($_REQUEST['signatura']) AND
trim($_REQUEST['signatura'] != ''))
        {
            $signatura = $_REQUEST['signatura'];
            $kontrola++;
        }
        if(isset($_REQUEST['prirustkove_cislo']) AND
trim($_REQUEST['prirustkove_cislo']))
        {
            $prirustkove_cislo = $_REQUEST['prirustkove_cislo'];
            $kontrola++;
        }

        if($kontrola > 1)
        {
            echo '<div id="error"><h1>Zadejte prosím jen jeden
výraz na vyhledávání!</h1></div>';
        }
        else
        {
            $con=mysql_connect(HOST, USER, PASS) or
die(mysql_error());
            mysql_select_db(DB,$con) or die(mysql_error());
            if($autor != '')
            {
                $query = 'SELECT v.osobni_cislo, v.prijmeni,
v.jmeno, vp.pujceni, vp.vraceni, k.signatura,
k.prirustkove_cislo, k.nazev, a.prijmeni, a.jmeno FROM
vyucujici v, vypujcky vp, knihy k, autori a WHERE (a.jmeno
like \''.$autor.'\' or a.prijmeni like \''.$autor.'\' ) AND
k.autor_id=a.id AND vp.prirustkove_cislo=k.prirustkove_cislo
AND vp.osobni_cislo=v.osobni_cislo;';
            }

            if($nazev_knihy != '')
            {
                $query = 'SELECT v.osobni_cislo, v.prijmeni,
v.jmeno, vp.pujceni, vp.vraceni, k.signatura,
k.prirustkove_cislo, k.nazev, a.prijmeni, a.jmeno FROM
vyucujici v, vypujcky vp, knihy k, autori a WHERE k.nazev like
 \''.$nazev_knihy.'\' AND k.autor_id=a.id AND
vp.prirustkove_cislo=k.prirustkove_cislo AND
vp.osobni_cislo=v.osobni_cislo;';
            }
        }
    }
}

```

```

        if($signatura != '')
        {
            $query = 'SELECT v.osobni_cislo, v.prijmeni,
v.jmeno, vp.pujceni, vp.vraceni, k.signatura,
k.prirustkove_cislo, k.nazev, a.prijmeni, a.jmeno FROM
vyucujici v, vypujcky vp, knihy k, autori a WHERE k.signatura
like \''.$signatura.'\' AND k.autor_id=a.id AND
vp.prirustkove_cislo=k.prirustkove_cislo AND
vp.osobni_cislo=v.osobni_cislo;';
        }

        if($prirustkove_cislo != '')
        {
            $query = 'SELECT v.osobni_cislo, v.prijmeni,
v.jmeno, vp.pujceni, vp.vraceni, k.signatura,
k.prirustkove_cislo, k.nazev, a.prijmeni, a.jmeno FROM
vyucujici v, vypujcky vp, knihy k, autori a WHERE
k.prirustkove_cislo like \''.$prirustkove_cislo.'\' AND
k.autor_id=a.id AND vp.prirustkove_cislo=k.prirustkove_cislo
AND vp.osobni_cislo=v.osobni_cislo;';
        }

        $res = mysql_query($query,$con) or
die(mysql_error());
        if(mysql_num_rows($res) ==0)
        {
            echo '<div id="nalezene"><h1>Nebyly nalezeny
žádné půjčené knihy.</h1></div>';
        }
        else
        {
            echo '<div id="nalezene">'.
            '<table>'.
            '<tr><th>osobní číslo
</th><th>jméno</th><th>příjmeni</th><th>vypůjčení</th><th>vrác
ení</th><th>signatura knihy</th><th>přírůstkové
číslo</th><th>název knihy</th><th>autor knihy</th></tr>';
            while($row = mysql_fetch_array($res))
            {
                echo '<tr><td>' . $row['vyucujici.osobni_cislo'] . '</td><td>' .
                $row['vyucujici.jmeno'] . '</td><td>' . $row['vyucujici.prijmeno']
                . '</td><td>' . $row['vypujcky.pujceni'] . '</td><td>' . $row['vypujc
ky.vraceni'] . '</td><td>' . $row['knihy.signatura'] . '</td><td>' . $
row['knihy.prirustkove_cislo'] . '</td><td>' . $row['knihy.nazev']
                . '</td><td>' . $row['autori.prijmeni'] . '
                ' . $row['autori.jmeno'] . '</td></tr>';
            }
            echo '</table>'.
            '</div>';
        }
    }

```

```

    }
}

echo $foot;
?>
Makro

Option Explicit
Sub vygenerujKnihyAVypujcky()
    Dim row As Integer, val As String

    row = 1
    Dim cark, prirc, sign, nazev, osob, pujc, vrac, titul As
String
    sign = Cells(2, 1).Value
    While Len(sign) > 0
        row = row + 1

        Sheets(1).Select
        sign = Cells(row, 1).Value
        prirc = Cells(row, 2).Value
        cark = Cells(row, 3).Value
        pujc = Cells(row, 4).Value
        vrac = Cells(row, 5).Value
        titul = Cells(row, 6).Value

        If Len(sign) > 0 Then
            Sheets("kniha").Select
            'prirustkove_cislo    signatura    nazev autor_id

            Cells(row, 1).Value = prirc
            Cells(row, 2).Value = sign
            Cells(row, 3).Value = titul

            'id_vypujcka prirustkove_cislo    osobní_cislo
            pujceni vraceni
            Sheets("Výpůjčka").Select
            Cells(row, 2).Value = prirc
            Cells(row, 3).Value = Sheets(1).Name
            Cells(row, 4).Value = pujc
            Cells(row, 5).Value = vrac
        End If
    Wend
End Sub

Sub Makrol()
    '
    ' Makrol Makro
    '

```

```

    Sheets("kniha").Select
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "prirustkove_cislo"
    Range("B2").Select
    ActiveCell.FormulaR1C1 = "signatura"
    Range("C2").Select
    ActiveCell.FormulaR1C1 = "nazev"
    Range("D2").Select
    ActiveCell.FormulaR1C1 = "autor_id"
    Range("E2").Select
    Sheets("Výpůjčka").Select
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "id_vypujcka"
    Range("B2").Select
    ActiveCell.FormulaR1C1 = "prirustkove_cislo"
    Range("C2").Select
    ActiveCell.FormulaR1C1 = "osobní_cislo"
    Range("D2").Select
    ActiveCell.FormulaR1C1 = "pujzeni"
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "vraceni"
    Range("F2").Select
    Sheets("KIS19").Select
    Range("A8").Select
    ActiveCell.FormulaR1C1 = "xxx"
    Range("A9").Select
End Sub

Sub smazVygenerovaneUdaje()
'
' smazVygenerovaneUdaje Makro
'

    Sheets("kniha").Select
    Range("A2:E263").Select
    Selection.ClearContents

    Sheets("Výpůjčka").Select
    Range("A2:E263").Select
    Selection.ClearContents

    MsgBox "Vygenerované údaje smazány"

End Sub

```