

**UNIVERZITA PARDUBICE**

**Fakulta elektrotechniky a informatiky**

**Návrh databáze pro web**

**Richard Severa**

**Bakalářská práce  
2014**

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2013/2014

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Richard Severa**  
Osobní číslo: **I09255**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Návrh databáze pro web**  
Zadávající katedra: **Katedra informačních technologií**

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

V teoretické části bude popsán obecný postup návrhu databáze se zaměřením na MySQL, tvorbu webového obsahu a možnosti importu dat do databáze. Dále bude vypracován přehled nástrojů pro správu obsahu webu (CMS) a existujících nástrojů typu framework. Praktická část bude návrh a realizace databáze v MySQL pro daný web, do které se importují existující data uložená v textové podobě ve formátu "flat-file" s prvky CSV. Dále bude pro tento web vytvořen grafický vzhled v CSS vycházející ze souvisejícího webu, vstupní stránka bude vhodně upravena tak, aby viditelně umožnila vstup oba dané weby. Vzhled a propojení webů budou konzultovány s majiteli (a správci) obou webů. Výsledný web musí splňovat příslušné normy W3C.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**Molinaro, Anthony: SQL: Kuchařka programátora. Computer Press, 2009. ISBN 978-80-251-2617-2.**

**Kofler, Michael: Mistrovství v MySQL 5: Kompletní průvodce webového vývojáře. Computer Press, 2007. ISBN 978-80-251-1502-2.**

**Kofler, Michael - Öggl, Bernd: PHP 5 a MySQL 5: Průvodce webového programátora. Computer Press, 2007. ISBN 978-80-251-1813-9.**

**Zeldman, Jeffrey: Tvorba webů podle standardů: XHTML, CSS, DOM, ECMAScript a dalších. Computer Press, 2011. ISBN 978-80-251-0347-0.**

**Lubbers, Peter - Albers, Brian - Salim, Frank: HTML5: Programujeme moderní webové aplikace. Computer Press, 2011. ISBN 978-80-251-3539-6.**

**Chaffer, Jonathan - Swedberg, Karl: Mistrovství v jQuery: Kompletní průvodce vývojáře. Computer Press, 2013. ISBN 978-80-251-4103-8.**

**Böhmer, Marian: Návrhové vzory v PHP. Computer Press, 2012. ISBN 978-80-251-3338-5.**

Vedoucí bakalářské práce:

**Mgr. Tomáš Hudec**

Katedra informačních technologií

Datum zadání bakalářské práce:

**20. prosince 2013**

Termín odevzdání bakalářské práce:

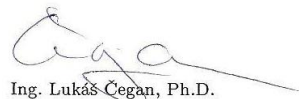
**9. května 2014**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2014

## Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 9. 5. 2014

Richard Severa

# Poděkování

Tímto bych chtěl poděkovat panu Mgr. Tomáši Hudcovi za podporu při tvorbě této bakalářské práce. A dále všem, kteří mě podporovali během studií.

# Anotace

Náplní této bakalářské práce je popsat postup návrhu databáze specializované na relační databázi MySQL. Tato bakalářská práce je rozdělena do dvou částí. V první, teoretické části, je zmíněn samotný vývoj MySQL a dále jsou vysvětlené základní pojmy používané v databázi. Dále je popsán postup navrhování schématu, importu dat a možnosti tvorby webové prezentace. V druhé, praktické části, je náplní tvorba samotného schématu návrhu databáze, realizace a tvorba návrhu webové prezentace.

## Klíčová slova

MySQL, import, relace, databáze, tabulka.

## Title

Database design for the web

## Annotation

The aim of this bachelor thesis is to explain the process of drafting a database specialized in relational database MySQL. This bachelor paper is divided into two parts. In the first part, theoretical one, there is the evolution of MySQL mentioned as well as the basic terms used in connection with this database. Then, there is a procedure of suggesting a scheme, data import, and the possibilities of designing a web presentation mentioned. The second part, practical one, is focused on the designing the draft of a database, its realization as well as a web presentation.

## Keywords

MySQL, import, relation, database, table.

# Obsah

Úvod	8
<b>1 Databáze MySQL</b>	<b>9</b>
1.1 Vývoj MySQL . . . . .	9
1.2 Základní pojmy . . . . .	9
1.3 Tabulky . . . . .	11
1.4 Datové typy . . . . .	13
<b>2 Návrh databáze</b>	<b>17</b>
2.1 Názvy . . . . .	17
2.2 Normální formy . . . . .	17
2.3 Relace . . . . .	19
2.4 Klíč . . . . .	19
2.5 Index . . . . .	19
2.6 Integrita . . . . .	20
<b>3 Import dat</b>	<b>21</b>
3.1 Přímé vložení . . . . .	21
3.2 Ze souboru . . . . .	21
3.3 Rozhraní phpMyAdmin . . . . .	22
<b>4 Tvorba webového obsahu</b>	<b>24</b>
<b>5 Redakční systém</b>	<b>26</b>
<b>6 Framework</b>	<b>28</b>
<b>7 Tvorba webu</b>	<b>30</b>
<b>Závěr</b>	<b>33</b>
<b>Seznam zkratk</b>	<b>34</b>

Seznam obrázků	35
Seznam tabulek	36
Seznam příloh	37
Reference	38

# Úvod

Tato bakalářská práce je zaměřena na tvorbu databázového schématu, obzvláště pro databázi typu MySQL. První část práce je rozdělena na více kapitol. V první kapitole je stručně popsán vývoj MySQL. Dále vypsány základní databázové pojmy, popis typů tabulek používaných v MySQL a datových typů.

Druhá kapitola je zaměřena na návrh schématu databáze. Nejprve jsou zmíněna doporučení k volbám pojmenování tabulek a atributů. Následně probrány normálové formy, sloužící jako pomocný nástroj k tvorbě kvalitních návrhů schémat. Zbytek této kapitoly se zabývá problematikou propojování tabulek. Tedy volbou správných klíčových dat, indexací a integritou databáze.

Třetí kapitola je zaměřena na samotné vkládání dat do databáze. Popsané způsoby jsou specializované na databázi MySQL a je zde taktéž zmíněna nastavba podobě aplikace phpMyAdmin.

V následujících třech kapitolách jsou popsány způsoby tvorby webového obsahu. V první z nich jsou popsány možnosti tvorby za pomoci přímého použití programovacích jazyků a jejich kombinací, kdy je znalost těchto jazyků nezbytná. Zbylé dvě kapitoly jsou zaměřeny na tvorbu obsahu za pomoci již předpřipravených systémů, u kterých uživatel nemusí znát jejich přesnou strukturu.

Druhá část této práce se stručně zabývá popisem vývoje návrhu databáze pro již existující web, problematikou vývoje schématu a následným importem dat do databáze typu MySQL. Dále se zaměřuje na návrh vzhledu webové prezentace, kde byly kladeny specifické požadavky.

# 1 Databáze MySQL

Pojmem databáze můžeme rozumět uspořádané uložení dat. Jednotlivé informace jsou rozdělené do tabulek, které mohou na sebe odkazovat. Kolonky v tabulkách obsahují specifické informace. Je potřeba ověřovat pravdivost a celistvost těchto informací, jinak by databáze nebylo možné použít.

Nejběžněji používanou databází je relační databáze využívající logiky relační algebry. Přesná znalost této logiky uživatelem však není vyžadována.

## 1.1 Vývoj MySQL

Databáze MySQL je distribuována pod licencí open source, díky které si její tvůrci zajistili mnoho příznivců. To hlavně v době vzniku, kdy tato relační databáze neposkytovala podporu všech běžných funkcí jazyka SQL (např. cizí klíč, vnořené selecty). Výhodou této databáze je také možnost rozšíření o moduly a podporu jiných technologií (Java, Perl, Python, PHP atd.) a to hlavně díky kvalitně zpracované dokumentaci. Mluvit o databázi MySQL jako o relační databázi lze ve chvíli, kdy uživatel dá návrhu schématu určitou formu, strukturu a propojení.

Vznik této databáze se dá datovat až do roku 1979, kdy Monty Widenius pracující pro společnost TcX vytvořil UNIREG (terminálové rozhraní k přístupu k uložení dat). V roce 1994 však firma začala vyvíjet webové aplikace a tento systém již přestal vyhovovat. Monty Widenius s kolegy hledal řešení v databázi vyvinuté od Davida Hughese, mSQL. Ta však nezvládala indexaci, což pro použití velkých databází byl problém. Rozhodnutí padla na využití zkušeností z UNIREG a využít je při tvorbě vlastní databáze podobné mSQL. Ta ve verzi 1.0 dostala název MySQL. Od 4. série je v databázi zahrnuta podpora klíčů a transakcí atd., díky kterým se tato databáze může řadit mezi nejlepší databáze na trhu. [1]

## 1.2 Základní pojmy

Jak je vidět z úvodu kapitoly, takováto terminologie není přesná a nemusí být vždy srozumitelná, proto pro vytváření databází je nezbytné znát několik základních pojmů.

Znalost těchto pojmů ušetří čas při návrhu databáze samotné a mnohem více při konzultaci s další osobou.

1. Tabulka – je základní stavební částice databáze. Pro specifikaci dané tabulky se používá jednoznačný název. Tabulka samotná se skládá ze sloupců a řádků.

Tabulka 1: Studenti

<b>IdStudent</b>	<b>Jmeno</b>	<b>Mesto</b>	<b>Adresa</b>
1	Pavel Světlý	Klatovy	Sluneční 3
2	Iva Peterková	Brno	Komín 576
3	Hana Nová	Praha	Na magistrále 9
4	Petr Skořepa	Olomouc	Uhelná 298

Tabulka 2: Předměty

<b>Predmet</b>	<b>Predm_IdStudent</b>
Matematika	1
Programování	1
Matematika	2
Anglický jazyk	4

2. Sloupec – neboli atribut. Musí mít svůj jednoznačný a jedinečný název. Taktéž musí být určitého datového typu (řetězcový, číselný atd.). Znázorněno v tabulce číslo 1, kde sloupec „IdStudent“ obsahuje pouze číslice a další sloupce text.
3. Řádek – neboli záznam. Každý řádek obsahuje informace v určitém pořadí, které je dáno pořadím atributů tabulky.
4. Pole – neboli hodnota. Každé pole v tabulce v sobě ukládá konkrétní informaci, se kterou se může následně dále pracovat.
5. Klíč – v databázi se používají dva druhy klíčů. Primární a cizí klíč. Primární klíč v tabulce slouží k jednoznačnému označení záznamu. (Viz tabulka 1, sloupec

Predm\_IdStudenta). Cizí klíč v jedné tabulce ukazuje na primární klíč v tabulce druhé. Slouží tedy k propojování tabulek (Viz tabulka 2, sloupec IdStudenta).

6. Relace – jak je zmíněno již u cizího klíče, jen nepřesně řečeno. Propojení tabulek se říká relace. Z tohoto důvodu můžeme mluvit o relační databázi.
7. Index – u rozsáhlých databází pro zjednodušení hledání/výběru dat se používá indexace. Indexem se rozumí buď další soubor, nebo další oblast souboru (InnoDB) obsahující setříděné odkazy na záznamy v tabulce. [2]
8. Schéma databáze – je popis databáze samotné. Skládá se ze všech pojmů, které až do teď byly probrány.
9. Dotaz – slouží k získání informací z databáze. Pro zápis dotazů se používá jazyka SQL. Informace dotazem získané se zobrazí v tabulce, která se nastalo do databáze neukládá.

## 1.3 Tabulky

Při zakládání nové tabulky v MYSQL je nutné zadat její typ. Jednotlivé typy se liší svými vlastnostmi. Nejčastěji používané budou nyní popsány.

### MyISAM

MyISAM tabulka je standardním typem tabulek používaných v MySQL. Je taktéž nejpoužívanější a to díky své vyspělosti, spravovatelnosti a stabilitě.

### InnoDB

Tabulka typu InnoDB má stejné vlastnosti jako předchozí typ. Liší se však ve dvou výhodných bodech:

- Tento typ tabulek lze spouštět jako transakci.
- Využívají pravidel integrity.

Tato tabulka není však dokonalá, jsou i důvody proč ji nepoužívat:

- Stabilita není na takové úrovni jako u tabulky MyISAM.
- Nemožnost vytvoření fulltextového indexu.
- Složitější správa těchto tabulek.
- V neposlední řadě finance, kdy komerční licence MySQL s podporou těchto tabulek je přibližně dvakrát tak dražší než verze bez ní. U Open Source aplikací a jednodušších webových stránek postačí bezplatná verze MySQL.

## HEAP

Tento typ tabulek se vytváří pouze v operační paměti počítače. Nedochozí tedy k ukládání na pevný disk. Dále využívají hash index umožňující velmi rychlý přístup k záznamům. Vhodné použití těchto tabulek je v případě maximální rychlosti správy menších tabulek. Jelikož tabulky nejsou ukládány na pevný disk, tabulka se po ukončení databáze nenávratně odstraní.

Tento typ má svá omezení, nejdůležitější jsou:

- Nemožnost použití datových typů TEXT a BLOB.
- Prohlížení záznamů za pomoci znaků = či <=> (nelze >, <, <=, >=).
- Nemožnost využití funkce AUTO\_INCREMENT.
- Pouze v nenulových sloupcích lze použít indexy.

## Dočasné tabulky

Pro jednotlivě popsané typy je možné vytvořit jejich obdoby dočasné (pomocí příkazu CREATE TEMPORARY TABLE. . .). Většinou k jejich vytvoření dochází při připojení skriptu PHP k databázi. Výhodou je, že za paralelního běhu více skriptů si každý skript vytvoří svou vlastní dočasnou tabulku stejného názvu a nedochází ke konfliktům (dočasné tabulky nejsou viditelné pro ostatní skripty). Po ukočení skriptu se tabulka odstraní.

Tyto tabulky se neukládají do stejného adresáře jako ostatní tabulky v databázi. Obvyklé uložení je do dočasného adresáře (u Windows C:\Windows\Temp, Linux /tmp či /var/temp nebo /usr/temp). [2]

## 1.4 Datové typy

V této kapitole budou probrány datové typy používané v MySQL. Při zakládání tabulky se pro každý sloupec musí určit jakého datového typu bude. V databázi MySQL se rozlišují tři typy sloupců – číselné, datum a čas, textový řetězec. Další varianty typů jsou z nich většinou odvozené. [3]

### Číselné typy

Číselné typy se dělí do dalších kategorií, které budou následně probrány.

- Celá čísla – celá čísla je možné do databáze MySQL ukládat se znaménkem (+ nebo –). Pokud by bylo nutné ukládat jen čísla kladná, použije se atribut UNSIGNED.

Tabulka 3: Celá čísla

Typ	Význam
TINYINT(m)	8-Bit-Integer
SMALLINT(m)	16-Bit-Integer
MEDIUMINT(m)	24-Bit-Integer
INT(m), INTEGER(m)	32-Bit-Integer
BIGINT(m)	64-Bit-Integer

Parametr *m* je nepovinný. Vyjadřuje maximální počet cifer, při použití příkazu SELECT.

- Čísla s pevnou a pohyblivou desetinnou čárkou – rozdíl mezi těmito dvěma typy spočívá v podobě uložení dat. Čísla s pevnou desetinnou čárkou se ukládají ve formě řetězce, přičemž čísla s pohyblivou desetinnou čárkou se ukládají v binárním formátu.

Výhodou čísel s pevnou desetinnou čárkou je zamezení chyb při zaokrouhlování. Mají i své nevýhody, které je třeba zmínit. Tyto čísla mají vyšší paměťovou náročnost a jejich zpracování je také časově náročnější.

Tabulka 4: Čísla s desetinnou čárkou

Typ	Význam
FLOAT(m,d)	Přesnost 8 desetinných míst. Pro čísla s plovoucí desetinnou čárkou.
DOUBLE(m,d)	Přesnost 16 desetinných míst. Pro čísla s plovoucí desetinnou čárkou.
REAL(m,d)	Synonymum pro DOUBLE
DECIMAL(p,s)	Maximálně 64 cifer. Přesnost max. 30 desetinných míst. Pro čísla s pevnou desetinnou čárkou.
NUMERIC, DEC	Synonymum pro DECIMAL

Parametry  $m$  a  $d$  jsou nepovinné. Vyjadřují maximální počet cifer před a za desetinnou čárkou, při použití příkazu SELECT. Parametry  $p$  a  $s$  vyjadřují celkový počet cifer a počet cifer za desetinnou čárkou.

## Datum a čas

Nejčastěji používané časové formáty jsou DATETIME a TIMESTAMP. První z nich slouží k ukládání data a času a druhý jako záznam v tabulce. Přesněji, slouží jako zaznamenání času v tabulce, kde došlo k jakékoliv úpravě. Toto provádí MySQL automaticky bez nutnosti zásahu uživatele (vyžaduje se pouze nastavení atributu pro danou tabulku). Jak je zřejmé, záznam se po každé úpravě tabulky přepíše. Pro potřebu zaznamenávání jednotlivých časů úprav je vhodnější použít DATETIME.

## Řetězce znaků

Slouží k ukládání textu do tabulek. Pro kratší řetězce se používají typy CHAR a VARCHAR, u kterých uživatel může nastavit maximální délku řetězce. Typ CHAR má tu nevýhodu, že při vložení řetězce, který je kratší než maximální možná délka, jsou prázdná místa nahrazena nulou a uložena. Tuto nevýhodu řeší typ VARCHAR, u kterého se doplněné nuly neukládají.

Tabulka 5: Datum a čas

Typ	Význam
DATE	Formát zápisu: 2014-12-30.
TIME	Formát zápisu: 23:59:59.
DATETIME	Formát zápisu: 2014-12-30 23:59:59.
YEAR	Formát zápisu v rozsahu: 1900 – 2155.
TIMESTAMP	Shodné se zápisem DATETIME.

Tabulka 6: Řetězce znaků

Typ	Význam
CHAR(n)	Předem daná délka řetězce. Max.: 255 znaků.
VARCHAR(n)	Proměnná délka řetězce. Max.: 255 znaků (MySQL 4.1).
TINYTEXT	Proměnná délka řetězce. Max.: 255 znaků.
TEXT	Proměnná délka řetězce. Max.: $2^{16} - 1$ znaků.
MEDIUMTEXT	Proměnná délka řetězce. Max.: $2^{24} - 1$ znaků
LONGTEXT	Proměnná délka řetězce. Max.: $2^{32} - 1$ znaků.

Parametr  $n$  je povinný. Vyjadřuje maximální délku řetězce.

## Binární data

Umožňují do databáze ukládat binární soubory (obrázky, hudbu). Velikost souborů je omezena použitým typem. Ukládání souborů přímo do databáze není příliš efektivní a rychlé. Proto se spíše do databáze ukládají názvy souborů.

## Nastavení atributů

Při nastavování jednotlivých atributů je možné každému z nich nastavit určité vlastnosti. Některé z nich jsou vázány na určitý datový typ. V tabulce 8 jsou shrnuty nejčastěji používané.

Tabulka 7: Binární data

Typ	Význam
TINYBLOB	Proměnná délka dat. Max.: 255 bajtů.
BLOB	Proměnná délka dat. Max.: $2^{16} - 1$ bajtů.
MEDIUMBLOB	Proměnná délka dat. Max.: $2^{24} - 1$ bajtů.
LOB	Proměnná délka dat. Max.: $2^{32} - 1$ bajtů.

Tabulka 8: Nastavení atributů

Typ	Význam
NULL	Není nutné atribut vyplnit při vkládání záznamu.
NOT NULL	Je nutné atribut vyplnit při vkládání záznamu.
DEFAULT	Vložení nastavené hodnoty při nevyplnění atributu při vkládání.
PRIMARY KEY	Určení atributu v tabulce jako klíčový.
AUTO_INCREMENT	Použití pro celočíselné sloupce. Pojí se s Primary key a Not null.

## 2 Návrh databáze

Navržení dobře použitelné databáze není jednoduchá záležitost, jak by se na první pohled mohlo zdát. Před tím než dojde k vkládání tabulek do databáze, je dobré si návrh nakreslit a promyslet strukturovanost jejího použití. Například zabránit vkládání stejných dat do více tabulek, zamezit atributům typu sklad1, sklad2 atd. v jedné tabulce (Co kdyby bylo skladů více?), dělit rozsáhlé tabulky na menší. Tyto problémy při navrhování může řešit znalost normálních forem, které budou v této kapitole probrány.

### 2.1 Názvy

V MySQL by názvy databází, tabulek a atributů měly mít nejvíce 64 znaků. Při zadávání názvů databázím a tabulkám je potřebné dbát na rozlišování malých a velkých písmem. U atributů se malé/velké znaky již nerozlišují.

V názvech je dobré taktéž nepoužívat žádné speciální znaky a diakritiku. Ne snad proto, že by nebyla zaručena podpora znakových sad, ale závisí na operačním systému, na kterém MySQL je provozováno. Problém může nastat až tehdy, kdy dojde k přemístění databáze na jiný server (s jiným operačním systémem), kde původně zvolená znaková sada není plně podporována.

Při pojmenování je dobré volit krátké a výstižné názvy. Díky tomu se v budoucnu, při orientaci v databázi, ušetří mnoho času. Dále jednotnost pojmenování dělá databázi přehlednější. A hned od začátku si stanovit zda názvy budou v jednotném či množném čísle.

### 2.2 Normální formy

Pro navržení použitelné databáze je dobré, přinejmenším pro začátečníky, držet se normálních forem.

1. Nultá normální forma – tabulka je nulté normální formě právě tehdy, existuje-li alespoň jedno pole, které obsahuje více než jednu hodnotu. [4]

Tabulka takto navržená je neefektivní. Zisk informací je velice obtížný. Například atribut Adresa, ve kterém je uloženo město, ulice a číslo popisné. Následná formulace dotazu, pro zjištění kolikrát je v tabulce město XY, by byla zbytečně složitá.

2. První normální forma – tabulka je v první normální formě, jestliže lze do každého pole dosadit pouze jednoduchý datový typ (atributy jsou dále nedělitelné, tzv. atomické a tentýž atribut se neopakuje ve stejné tabulce). [4]

Mějme stejný příklad. Atribut adresa rozdělíme na atributy Mesto a Ulice a CisloPopisne. Již z tohoto je vidět, že kladení dotazů bude jednodušší.

3. Druhá normální forma – tabulka je ve druhé normální formě, jestliže je v první NF a navíc platí, že existuje klíč a všechna neklíčová pole jsou funkcí celého klíče (a tedy nejen jeho částí). [4]

Dalo by se říci, že již první normální forma je dostačující, ale na příkladu budou ukázány nedostatky. Představme si panelový dům. Lidé v něm bydlící mají stejnou adresu. Při vložení do vzorové tabulky dojde k opakování adres. To je velice neefektivní a paměťové náročné. Tabulku je nutné rozdělit na tři: Jmena, Adresy, Spoj\_Jmen\_Adr. Záznamy v tabulkách Jmena a Adresy mají přiřazený primární klíč. Tabulka Spoj\_Jmen\_Adr je spojovací (má dva atributy a to primární klíče z předchozích tabulek) a každý záznam v ní spojuje jednotlivé záznamy z předchozích tabulek.

4. Třetí normální forma – tabulka je ve třetí normální formě, jestliže každý neklíčový atribut není transitivně závislý na žádném klíči schématu neboli, je-li ve druhé normální formě a zároveň neexistuje jediná závislost neklíčových sloupců tabulky. [4]

Příklad, aby odpovídal definici, je nutné upravit. A to tak, že se odstraní tabulka s primárními klíči Spoj\_Jmen\_Adr a do tabulky Jmena se přidá atribut pro primární klíč z tabulky Adresy (zde se mu již říká cizí klíč).

## 2.3 Relace

Relace je spojení, vazba mezi tabulkami. Dělí se na tři typy, následně probrány.

1. Vazba 1:1 – toto označení říká, že k záznamu v jedné tabulce existuje záznam v tabulce druhé. Pro příklad: Osoba a číslo občanského průkazu.
2. Vazba 1:N – zde již k jednomu záznamu v první tabulce existuje více záznamů v tabulce druhé. Pro příklad: Osoba může vlastnit více aut, ale auto smí vždy vlastnit jen jedna osoba.
3. Vazba M:N – k záznamu v jedné tabulce existují záznamy v tabulce druhé a naopak. Pro příklad: Rodič může mít více potomků a jedno dítě může mít více rodičů.

## 2.4 Klíč

V databázi MySQL se používají dva druhy klíčů. Jsou jím primární a sekundární klíč, které jsou popsány v následujících dvou odstavcích.

1. Primární klíč – dále jen PK, v tabulce slouží k jednoznačnému a jedinečnému označení záznamu. Může se skládat z více než jednoho sloupce, ale stále platí, že tento PK musí být v tabulce jedinečný. Nesmí taktéž nabývat nulových hodnot. S funkcí PK se pojí nastavení `AUTO_INCREMENT`, která, v MySQL, nastavuje hodnotu číselného PK vzestupně (1, 2, 3 atd.).
2. Cizí klíč – tento klíč v jedné tabulce ukazuje na primární klíč v tabulce druhé. Slouží tedy k vytváření relací. Cizí klíč by měl být stejného datového typu jako primární klíč. Funkce `AUTO_INCREMENT` se v tomto případě nepoužívá.

## 2.5 Index

Indexem se rozumí buď další soubor, nebo další oblast souboru (InnoDB) obsahující setříděné odkazy na záznamy v tabulce. [2]

Indexování urychluje hledání dat v tabulce. Funguje v podstatě jako obsah v knize, není tedy nutné procházet list za listem a hledat požadovounou informaci. V MySQL se používají tři typy indexů.

1. Běžný index – definuje se slovem INDEX. Jediný účel tohoto indexu je zrychlení nalezení dat. Pro indexování je vhodné zvolit ty sloupce, které budou v budoucnu často používány při vyhledávání.
2. Unikátní index – definuje se slovem UNIQUE. Při použití běžného indexu v atributu není zajištěna nemožnost opakování se dat v atributu. Toto řeší unikátní index. V atributu s unikátním indexem mohou být hodnoty NULL.
3. Primární index – definuje se slovem PRIMARY. Je dalším krokem rozšíření a to tak, že v atributu kde je tento index použit není možné mít nulové hodnoty. Sloupec musí být typu NOT NULL.

## 2.6 Integrita

Nyní, když je známo jak by se databáze měla navrhovat, je nutné brát na zřetel integritu. Neboli důvěrnost a celistvost databáze. Pokud v databázi jsou využity primární a cizí klíče, je důležité při odstraňování záznamů zjistit zda neexistuje na daný záznam odkaz. Odstranění záznamu bez jakékoli uvážení by vedlo k tzv. porušení referenční integrity. Jsou dvě možnosti jak tento problém řešit. Buď odstranění neprovést či odstranit všechny záznamy ve všech závislých tabulkách. MySQL tyto typy zachování integrity podporuje, ale jen v tabulkách typu InnoDB. [2]

## 3 Import dat

Ukládání dat se v databázi MySQL převážně řeší pomocí příkazu INSERT a také příkazu INFILE. Možnosti jejich použití jsou popsány v této kapitole.

### 3.1 Přímé vložení

Pro přímé vložení záznamu se používá jazyka SQL, konkrétně příkaz INSERT. Syntaxe příkazu umožňuje několik druhů zápisu dat, v této podkapitole řešených.

Pokud se vkládá záznam se všemi hodnotami atributů tabulky, není nutné za názvem tabulky říkat do jakých sloupců se data ukládají.

```
INSERT INTO nazev_tabulky VALUES ('hodnota1', 'hodnota2', 'hodnota3');
```

Pokud je ale nezbytné vložit do tabulky jen část informací, musí se uvést do jakých sloupců se data budou ukládat. Poté jen dodržet pořadí vkládaných dat podle pořadí sloupců. Do nevyplněných sloupců MySQL automaticky uloží hodnotu NULL.

```
INSERT INTO nazev_tabulky (atribut1, atribut2) VALUES ('hodnota1',  
'hodnota2');
```

Příkaz INSERT umožňuje uložení více záznamů najednou. Stačí za modifikátor VALUES připsat do závorek další hodnoty záznamu.

```
INSERT INTO nazev_tabulky (atribut1, atribut2) VALUES ('hodnota1',  
'hodnota2'),('hodnota1A', 'hodnota2A');
```

### 3.2 Ze souboru

Pro vkládání dat ze souboru se používá příkazu INFILE. Jeho použití není triviální. Je nutné znát vnitřní strukturu souboru, aby bylo možné správně data do tabulky importovat.

```
LOAD DATA LOCAL INFILE 'nazev_souboru' INTO TABLE nazev_tabulky FIELDS  
TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n';
```

Tento zápis říká, že soubor je uložen v počítači, záznamy jsou v uvozovkách, oddělené čárkou a odentrováním jsou ukončené řádky. Níže je uveden popis identifikátorů.

- `FIELDS TERMINATED BY 'oddělovač záznamů'` – Do uvozovek se vkládá znak, který odděluje jednotlivé záznamy.
- `ENCLOSED BY 'uzavření záznamů'` – Do uvozovek se vkládá znak, který uzavírá a specifikuje hodnoty záznamu.
- `LINES TERMINATED BY 'ukončení řádku'` – Do uvozovek se vkládá znak, který ukončuje řádky.
- `LOW PRIORITY` – Tento modifikátor se píše za slovo `DATA` a znamená, že záznamy se do tabulky vloží až když nikdo s databází nebude pracovat.

Pokud by v souboru bylo jiné pořadí atributů než v tabulce, do které se bude importovat, za název tabulky se do závorek zapíše správné pořadí. [5]

### 3.3 Rozhraní phpMyAdmin

Jako třetí alternativa je možné pro databázi MySQL použít rozhraní phpMyAdmin. Oba předchozí příkazy jde přes toto rozhraní použít a to buď pomocí příkazové řádky, či pomocí formuláře (Viz obr. 3). Při použití formuláře není nutné znát přesnou syntaxi zápisu příkazů. Pouze zadat data do správných kolonek a rozhraní phpMyAdmin poté sestaví dotaz a dojde k jeho provedení. Pro získání informací jak pracovat s tímto rozhraním, je možné navštívit web výrobce a použít manuálovou dokumentaci. [6]

### Soubor pro importování:

---

Soubor může být komprimovaný (gzip, bzip2, zip) nebo nekomprimovaný.

Jméno komprimovaného souboru musí končit na **.[formát].[komprese]**. Například: **.sql.zip**

Procházet váš počítač:  Soubor nevybrán (Maximální velikost: 8 192KiB)

Znaková sada souboru:

### Částečný import:

---

Povolit přerušení importu v případě že skript pozná, že se blíží časový limit nastavený v PHP. *(Tímto způsobem můžete importovat i velké soubory, ale může to způsobit problémy s transakcemi.)*

Počet dotazů od začátku, které se mají přeskočit:

### Formát:

---

*Note: If the file contains multiple tables, they will be combined into one*

### Parametry pro výstupní formát:

---

Přepsat data tabulky souborem

Nepřerušit při chybě v příkazu INSERT

Pole oddělené:

Pole uzavřené do:

Pole escapována:

Řádky ukončené:

První řádka souboru obsahuje jména polí *(pokud toto nezaškrtnete, první řádka bude součástí dat)*

Obrázek 1: Formulář pro import ze souboru v rozhraní phpMyAdmin

## 4 Tvorba webového obsahu

Pro tvorbu webového obsahu je možné zvolit několik způsobů. Je nutné brát v potaz pro koho, či za jakým účelem jsou stránky tvořeny. Zda jen webová prezentace, fórum, internetový obchod atd. Podle toho je nutné zvolit postupy a určitý programovací jazyk, který může tvorbu usnadnit.

### HTML a CSS

Jazyk HTML je základ pro tvorbu webové stránky, který určuje její strukturu. Ta se tvoří takzvanými značkami. HTML stránky jsou statické. To znamená, že jak stránka byla napsána, tak bude i přesně v prohlížeči zobrazena. Pomocí tohoto jazyka je možné vytvářet jednoduché formuláře, publikovat texty, obrázky, audio a video záznamy, načítat informace pomocí hypertextových odkazů atd.

Již samotné HTML jde vzhledově upravovat, ale pro snadnější a udržitelnější formu správy vzhledu se používá jazyk CSS. Pro jednotlivé značky v HTML se nadefinují hodnoty (barva, pozice, velikost atd), které platí pro všechny stránky daného webu. Tedy v případě, že jsou nadefinované styly obsaženy v jednom souboru, který se do každé stránky musí připojit. [7]

Znalost a um použít tyto dva jazyky by měl být základ každého uživatele, který chce tvořit webové stránky.

### W3C

Pro standardizaci HTML, XHTML a dalších slouží normy od organizace W3C (World Wide Web Consortium). W3C vzniklo v polovině devadesátých let a to z důvodu nesourodosti v psaní HTML stránek. W3C dále zdokonaluje značkovací jazyky i jejich atributy. Při dodržování těchto standardů si programátor ušetří čas při validaci<sup>1</sup> a odladování, nejen vzhledu, stránek.[8]

---

<sup>1</sup>Ověření správnosti.

## PHP a MySQL

Jazyk PHP slouží k tvorbě dynamických webových stránek. Tedy obsah stránky nemusí být zcela předem znám. Stránka, která se má zobrazit, se vytvoří až ve chvíli, kdy o ni uživatel požádá (přesněji prohlížeč). Pokud tedy programátor bude chtít využívat jazyka PHP, musí ho server, na kterém budou stránky umístěny, podporovat. Jazyky PHP a HTML je možné v jedné stránce kombinovat. [9]

Jako další, velice funkční, rozšíření, je možné pomocí PHP získat/poslat informace z/do databáze. Pro zajištění spojení je nutné znát umístění, přístupové jméno a heslo do databáze. Poté do PHP kódu umístit požadovaný dotaz na databázi a odeslat ho. Databáze dotaz zpracuje, pošle zpět výsledek a pomocí PHP dojde k zobrazení. [10]

## Webový server Apache

Webový server je v podstatě program nainstalovaný na počítači, který poskytuje služby dalším programům/počítačům. Komunikace mezi klientem a serverem je zajištěna protokolem HTTP (HyperText Transfer Protocol).

Webový server Apache byl vyvinut v polovině devadesátých let. Používá se dodnes, stal se dokonce nejpoužívanějším web serverem vůbec. Instalace, základní konfigurace a použití je vcelku jednoduché. Nabízí i možnost rozšíření pomocí modulů (např.: ftp, pop3, smtpd atd.). Výhodou je též snadno dostupná dokumentace a přímá podpora, při řešení problému, od vývojářů či běžných uživatelů. [11]

## 5 Redakční systém

Slouží pro jednoduchou správu a publikování, nejen, webového obsahu. Často se označuje zkratkou CMS (Content Management System). Použití redakčního systému k uveřejnění webových stránek je pro uživatele určité usnadnění, není požadována přímá znalost programovacího jazyka. Zato je nutné se zorientovat v nastavení aplikace. Výběr redakčních systémů záleží na konkrétních požadavcích uživatele.

### Joomla

Tento redakční systém, založený na PHP, je možné použít pro webovou prezentaci, tvorbu blogu či jako menší eshop. Taktéž lze použít pro interní sítě k publikování dokumentů (firemní). Mezi plusové vlastnosti patří jednoduchá instalace, správa systému a modulů, mnoho rozšíření. Na druhou stranu je tento systém pro nastavení, a velkým počtem funkcí, složitý. I přes velkou oblibenost, není pro tento systém mnoho kvalitních šablon. [12]

### WordPress

Velice oblíbený redakční systém, založený na PHP, původně pro tvorbu blogů (vytvořen v roce 2001). Dnes již díky mnoha rozšířením (placených/bezplatných) lze použít například jako sociální síť, diskusní fórum. Díky velké komunitě uživatelů je k dispozici mnoho návodů a kvalitních šablon. Nevýhodou je žádná oficiální podpora. [12]

### Drupal

Velice kvalitní nástroj pro webovou prezentaci založený na PHP s možností instalace až 6000 doplňků dovoluje uživateli zaměřením se spíše na design. Stejně jako Joomla se často používá pro interní prezentaci. Díky velké pokročilosti a složitějšímu systému vytváření témat, může být pro jednodušší weby zbytečně náročný. [12]

## **TextPattern**

Ač je tento redakční systém méně známý, jedná se o kvalitní nástroj nejen pro tvorbu blogů. Je snadno použitelný a ovladatelný. Rozšíření o nástroj Textile slouží k převádění textu na validní XHTML kód. Určitou nevýhodou je jeho samotná nerozšířenost mezi uživateli, díky čemuž není k dispozici tolik nástrojů a kvalitních šablon jako u předchozích systémů. [12]

## 6 Framework

Slouží jako podpora při programování, kdy uživatel nemusí řešit všechny funkcionality, zabezpečení atd., ale může se zaměřit přímo na vývoj vlastní aplikace. Protože každý framework používá jinou syntaxi, je nutné nastudovat manual. Toto zabere dosti času oproti nevyužití framework. Tato nevýhoda se však ztrácí při opětovném použití, neboť další práce je již urychlena.

### Laravel

Je jeden z nejpoužívanějších PHP framework. Především díky snadno zapamatovatelné syntaxi a intuitivě použití. Základní verze není tak rozsáhlá, ale díky velkému množství přídatných balíčků je možné rozšířit pole působnosti. Nabízí též využití objektivě orientovaného programování a využití návrhového vzoru Model view controller. [13]

### Nette

Tento framework je velice oblíbený u české komunity PHP programátorů. Je napsaný v PHP a taktéž je objektivě orientovaný. Díky open source licenci je volně šiřitelný. Dá se použít jak na malé webové prezentace, tak i pro tvorbu CMS, eshopů atd. Dále se dá například použít společně s jiným framework. [14]

### Zend

Jako předešlé framework je i tento psán v PHP a je objektivě orientovaný. Díky své modularitě je možné použít jen potřebné balíčky a případně nové snadno přidat. Rozšířenost komunity uživatelů zajišťuje pomoc při nutnosti řešení vlastních problémů. Již v základu zajišťuje zabezpečení pro vytvářený web. [15]

### JQuery

Tento framework slouží pro vývoj javascriptových aplikací. Výhodou je podporovanost skoro všemi dnes používanými prohlížeči. Usnadňuje psaní a formátování HTML stránek

a to tak, že v prohlížečích se zobrazuje vždy stejně stylován (u formátování pomocí CSS tomu tak vždy není). Dále podporuje tvorbu animací, snadnější práci s poli a podpora AJAX<sup>2</sup>. [16]

---

<sup>2</sup>Umožňuje změnu části webové stránky bez nutnosti načtení stránky celé.

## 7 Tvorba webu

Hlavními úkoly praktické části bakalářské práce jsou návrh a realizace databáze, naplnění databáze daty a návrh grafického zpracování webové stránky.

### Design

Cílem při navrhování vzhledu byla změna stávající webové stránky a sjednocení grafického zpracování s již existující webovou stránkou. Před samotným začátkem tvorby vzhledu bylo nutné komunikovat se správcem a majiteli těchto webů. Z konzultace vyplulo, že na stávající stránce, ke které by se tvořený vzhled měl připodobňovat, dojde v budoucnu ke změnám samotného vzhledu, a tyto změny by se již mohly projevit v nově vznikajícím designu. Dále byly řečeny prvky a požadavky, které by návrh grafického zpracování měl nést.

Původní webová stránka, pro kterou byl tvořen návrh designu, byla generována ze souborů nesoucí prvky CSV. Zpracování probíhalo za pomoci shellových skriptů pro bash a programovacího jazyka AWK. Dále bylo využito unixových nástrojů např.: sed, grep, sort atd.

Pro programování a tvorbu vzhledu bylo vybráno prostředí programu NetBeans IDE 7.3. Samotné vytvoření webového návrhu bylo konstruováno programovacím jazykem HTML ve verzi 5. Strukturování a grafické zpracování vzhledu samotného bylo zajištěno použitím jazyka CSS.

### Databáze

Úkolem v této části práce bylo vyřešit problém návrhu a způsobu importování původních dat, které mají zajistit evidenci informací o hudební skupině (např.: písně, alba, informace o členech atd.).

Původní řešení zpracování dat nebylo zprostředkováno za pomoci relačního databázového systému. Data byla uložena v souborech a dále zpracována, jak již bylo zmíněno v předchozí podkapitole. Hlavními problémy tedy byly redundance, složitá správa a následné zpracování dat.

Realizace návrhu databáze (ER diagram) byla nejprve vytvořena v programu TOAD Data Modeler a to konkrétně pro databázi typu MySQL. Pomocí programu byl následně vygenerován SQL kód pro tvorbu databáze, který přes formulář aplikace phpMyAdmin vytvořil schéma databáze na serveru.

Dalším krokem bylo zpracování původních dat a importování do databáze. Jako účinný nástroj pro zpracování se ukázal tabulkový program Excel (podporuje ukládání dat do .csv souborů), do kterého byla data importována, předpřipravena a následně importována přímo do databáze. V tento moment, při kontrole importovaných dat, se ukázalo několik nedostatků, které při navrhování databáze nebyly vzaty v úvahu (např.: délka atributů). Bylo nutné zjednat nápravu a data následně znovu importovat.

Všechna data se však nepodařilo z původních souborů získat, bylo tedy nutné jejich ruční vkládání, které bylo velice časově náročné. Obdobně tomu bylo v případě propojování vazebních tabulek.

## Funkčnost webu

Propojení databáze a webové stránky bylo zajištěno programovacím jazykem PHP. Při jeho použití (tvorbě dotazů pro databázi), byly odhaleny další nedostatky při tvorbě schématu, které neumožňovaly získávat specifická data. Bylo tedy nutné dále některé tabulky upravit a to již do finálního provedení. Zároveň se také ukázaly mírné nedostatky v návrhu vzhledu některých stránek, kdy strukturu a přehlednost stránky bylo nutné opravit.

Následujícím krokem bylo provedení anglické mutace a validace stránek. Provedení úkonů v tomto pořadí nebylo správně zvolené. Při kontrole validace, provedené pomocí validačního nástroje dostupného na stránkách *validator.w3.org*, byly zjištěny chyby, které bylo nutné opravit na dvou místech zároveň.

Rozšíření stránek o možnost přepínání mezi jazyky bylo řešeno v návrhu webové prezentace tak, že v hlavičce stránky byly umístěny dvě ikony umožňující kliknutím přepínat mezi jazykovými mutacemi. V programové části bylo toto vyřešeno vytvořením nové složky, do které byly překopírovány soubory obsahující český jazyk. Nebylo tedy nutné překopírovat úplně všechny soubory (např.: složky obsahující obrázky, texty písni

atd.). Následným přeložením a změnami dotazů na databázi, k získávání anglických popisů a názvů, bylo dosaženo požadované funkčnosti návrhu webové prezentace.

## Závěr

Tato bakalářská práce popisuje pravidla tvorby databázových schémat se zameřením na relační databázi MySQL. Tato pravidla a pojmy jsou přehledně rozepsány a některé i vysvětleny na příkladech pro snadnější pochopení. Při tvorbě návrhu schématu v praktické části mi znalost a dodržování těchto pravidel velice pomohlo. Hlavně při hledání informací v odborné literatuře.

Dále v teoretické části byly popsány možnosti vkládání dat do databáze. MySQL nabízí nadstavbu v podobě phpMyAdmin. Toto rozhraní je uživatelsky velice přívětivé. Umožňuje snadnou orientaci v již existujícím schématu databáze či jednoduché vytváření schémat nových. Nabízí více funkcí, které mají kvalitně zpracovanou dokumentaci. Toto nejvíce ocení začínající uživatelé. Práce při tvorbě schématu a následné importování dat probíhalo právě přes toto rozhraní, kdy bylo využito obou možností vkládání (příkazový řádek, formulář).

Náplní praktické části byly návrhy vzhledu webové prezentace a schématu databáze, s příkladem zameření na hudební kapelu. Bylo tedy nutné řešit evidenci alb, singlů, písní atd. Samotná tvorba návrhu webové prezentace byla vytvořena za pomoci jazyků HTML, CSS a propojení s databází jazyky PHP a SQL. Nebylo tedy využito žádného nástroje typu framework či redakčního systému. Přehled těchto nejčastěji používaných nástrojů je sepsán v kapitolách Redakční systém a Framework.

## Seznam zkratek

SQL	Structured Query Language
MySQL	My Structured Query Language
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
HTTP	HyperText Transfer Protocol
CMS	Content Management System
AJAX	Asynchronous JavaScript and XML
CSV	Comma-separated values

## Seznam obrázků

1	Formulář pro import ze souboru v rozhraní phpMyAdmin . . . . .	23
2	Příloha č.1 – ER diagram . . . . .	40
3	Příloha č.2 – Návrh vzhledu stránky . . . . .	41

## Seznam tabulek

1	Studenti . . . . .	10
2	Předměty . . . . .	10
3	Celá čísla . . . . .	13
4	Čísla s desetinnou čárkou . . . . .	14
5	Datum a čas . . . . .	15
6	Řetězce znaků . . . . .	15
7	Binární data . . . . .	16
8	Nastavení atributů . . . . .	16

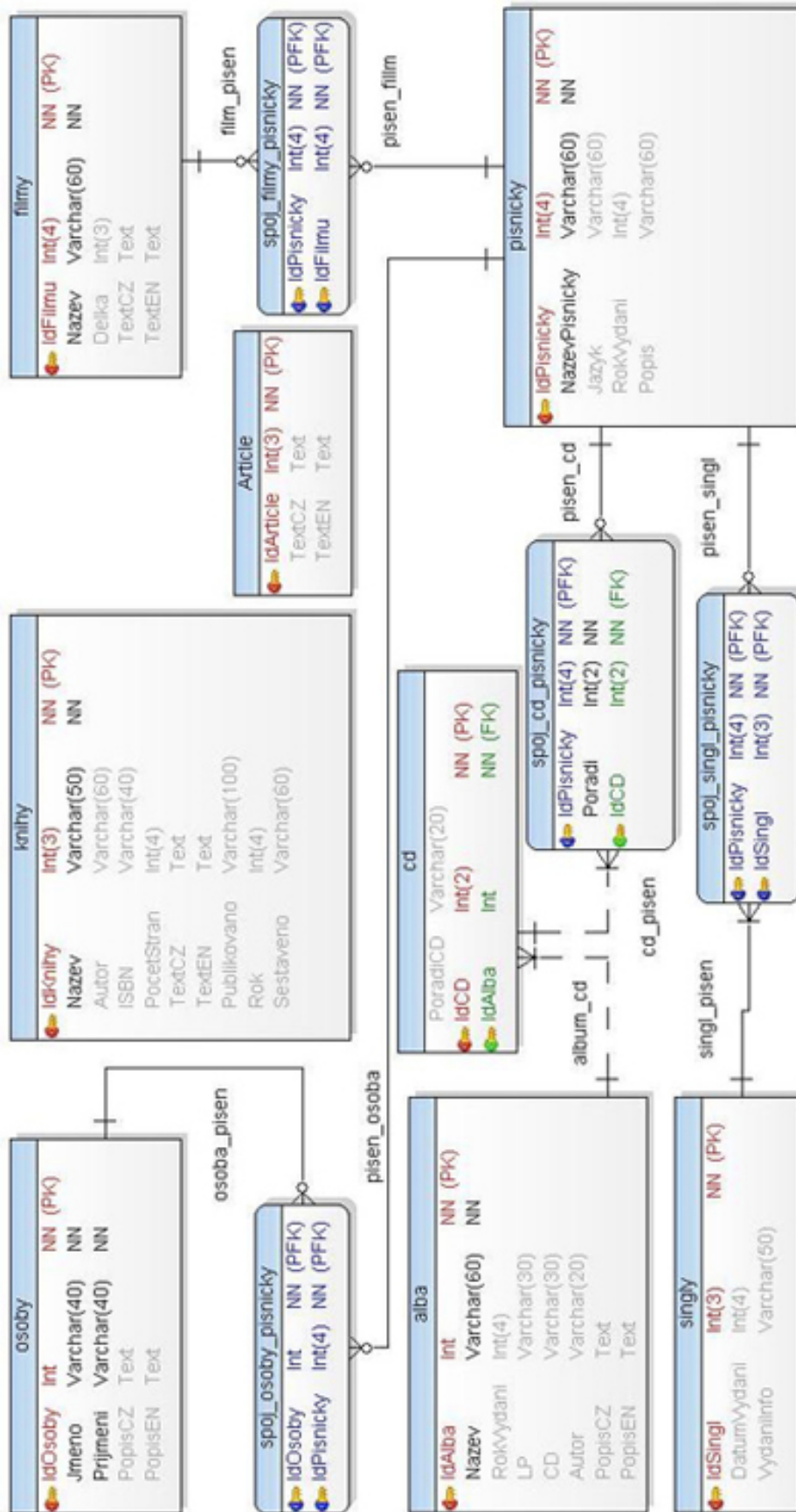
# Seznam příloh

Příloha č. 1	ER diagram
Příloha č. 2	Návrh vzhledu stránky
Příloha č. 3	CD

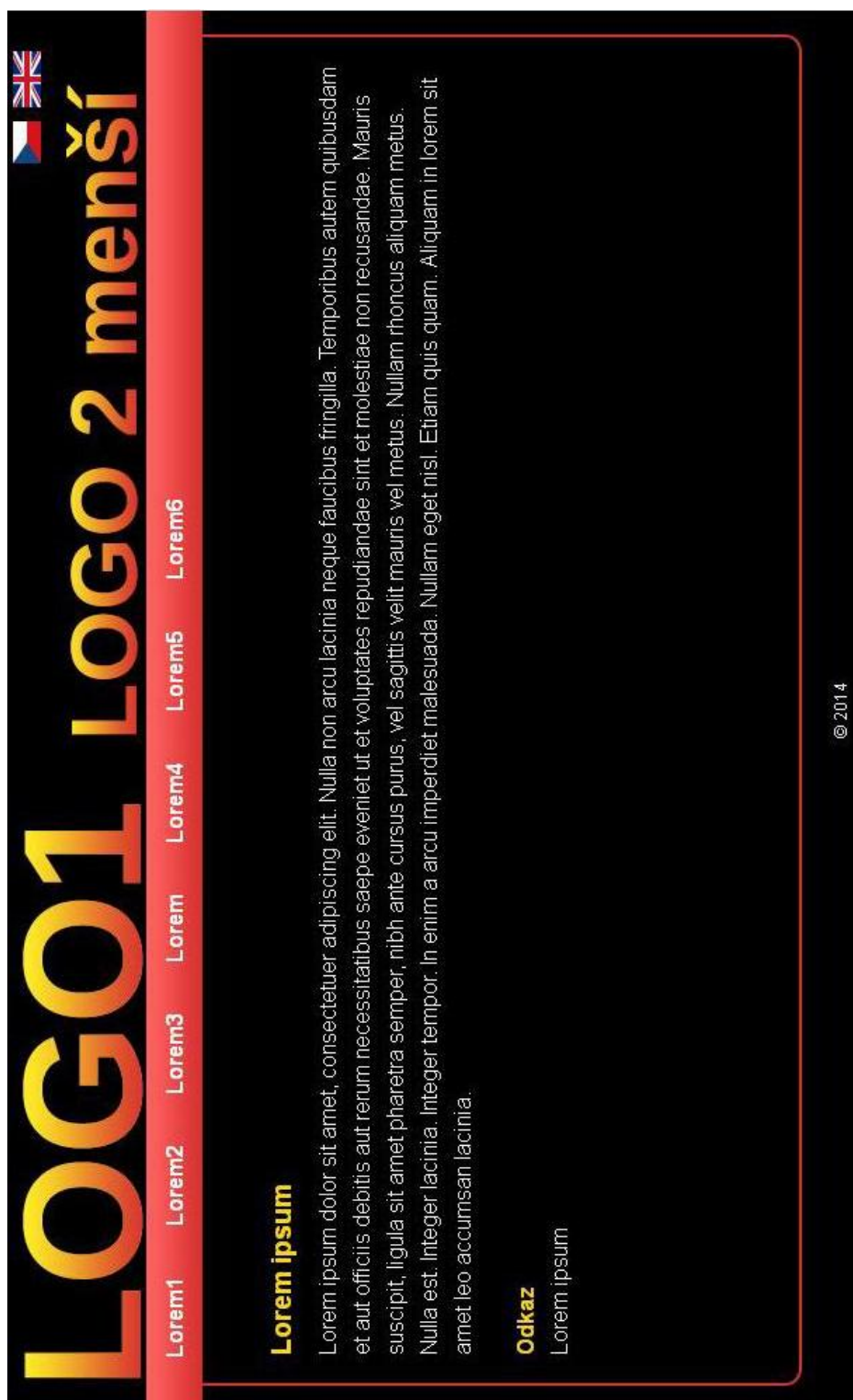
## Reference

- [1] CONVERSE, Tim a Joyce PARK. *PHP5 and MySQL bible*. Vyd. 3. Indianapolis, IN: Wiley, © 2004, xxxvii, 1042 p. ISBN 07-645-5746-7.
- [2] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2007, 607 s. ISBN 978-80-251-1813-9.
- [3] WELLING, Luke a Laura Thomson. *PHP a MySQL: rozvoj webových aplikací*. Vyd. 3. Praha: SoftPress, 2005, 830 s. ISBN 80-864-9783-6.
- [4] RYDVAL, Slávek. *Normální formy* [online]. 07. 08. 2005 [cit. 2014-04-23]. Dostupné z: <http://www.rydval.cz/phprs/view.php?cislocclanku=2005123127>
- [5] JUN, Adam. *MySQL databáze - český manuál: 5. PRÁCE S DATY*. JUNEXT [online]. 2014 [cit. 2014-04-25]. Dostupné z: <http://www.junext.net/mysql/>.
- [6] PHPMYADMIN. *Documentation* [online]. 2014 [cit. 2014-04-26]. Dostupné z: [http://www.phpmyadmin.net/home\\_page/docs.php](http://www.phpmyadmin.net/home_page/docs.php).
- [7] WORLD WIDE WEB CONSORCIUM (W3C). *HTML & CSS* [online]. © 2013 [cit. 2014-04-26]. Dostupné z: <http://www.w3.org/standards/webdesign/htmlcss>
- [8] WORLD WIDE WEB CONSORCIUM (W3C). *About W3C* [online]. © 2012 [cit. 2014-04-26]. Dostupné z: <http://www.w3.org/Consortium/>
- [9] ZAJÍC, Petr. Linuxsoft.cz. *PHP (2) – Jak to funguje [online]. 28.5.2004* [cit. 2014-04-26]. Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=172](http://www.linuxsoft.cz/article.php?id_article=172)
- [10] ZAJÍC, Petr. Linuxsoft.cz. *PHP (38) – Dolujeme data z MySQL* [online]. 16.8.2004 [cit. 2014-04-26]. Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=344](http://www.linuxsoft.cz/article.php?id_article=344)
- [11] THE APACHE HTTP SERVER PROJECT. *About the Apache HTTP Server Project* [online]. © 1997 – 2014 [cit. 2014-05-02]. Dostupné z: <http://www.w3.org/Consortium/>

- [12] CHAPMAN, Cameron. *10 nejlepších redakčních systémů (CMS)* [online]. 09. 11. 2011 [cit. 2014-04-30]. Dostupné z:  
<http://interval.cz/clanky/10-nejlepsich-redakcnich-systemu-cms/>
- [13] JAKOUBĚ, Jaroslav. *ORM test PHP frameworků – Kohana, Laravel* [online]. 26.7.2013 [cit. 2014-05-01]. Dostupné z:  
<http://www.zdrojak.cz/clanky/orm-test-php-frameworku-kohana-laravel/>
- [14] GRUDL, David. *Nette Framework: zvýšte svoji produktivitu* [online]. 10.3.2009 [cit. 2014-05-01]. Dostupné z:  
<http://www.zdrojak.cz/clanky/nette-framework-zvyste-svoji-produktivitu/>
- [15] ZEND TECHNOLOGIES LTD. *Zend Framework* [online]. © 2006–2014 [cit. 2014-05-01]. Dostupné z: <http://framework.zend.com>
- [16] THE JQUERY FOUNDATION. *JQuery* [online]. © 2014 [cit. 2014-05-01]. Dostupné z: <http://jquery.com>



Obrázek 2: Příloha č.1 – ER diagram



Obrázek 3: Příloha č.2 – Návrh vzhledu stránky