



VYSOKÉ UČENÍ FAKULTA ELEKTROTECHNIKY  
TECHNICKÉ A KOMUNIKAČNÍCH  
V BRNĚ TECHNOLOGIÍ

# PROJEKT DATABÁZOVÝ DESIGN - KNIHOVNA

**Autor:** Veronika Vojáčková

**Obor:** Informační bezpečnost

**Předmět:** Bezpečnost databázových systémů

## Obsah

Úvod.....	3
1. Odkaz na GitHub repository .....	3
2. Obrázek reprezentující databázový design.....	3
3. Popis aplikace .....	4
4. Popis každé tabulky .....	4
4.1. Tabulka user .....	4
4.2. Tabulka writer.....	4
4.3. Tabulka book.....	5
4.4. Tabulka type.....	5
4.5. Tabulka borrow .....	5
4.6. Tabulka reader .....	5
4.7. Tabulka address.....	5
4.8. Tabulka state.....	6
4.9. Tabulka contact_r.....	6
4.10. Tabulka library .....	6
4.11. Tabulka worker .....	6
4.12. Tabulka role.....	7
4.13. Tabulka contact_w.....	7
4.14. Tabulky bookwriter, booktype a workerhasaddress.....	7
5. Proč se jedná o databázi v 3 normální formě?.....	7
6. DDL skripty pro vytvoření databáze v SQLite .....	7
7. Důkaz o vytvoření databáze .....	8
Závěr .....	10
Seznam obrázků .....	10

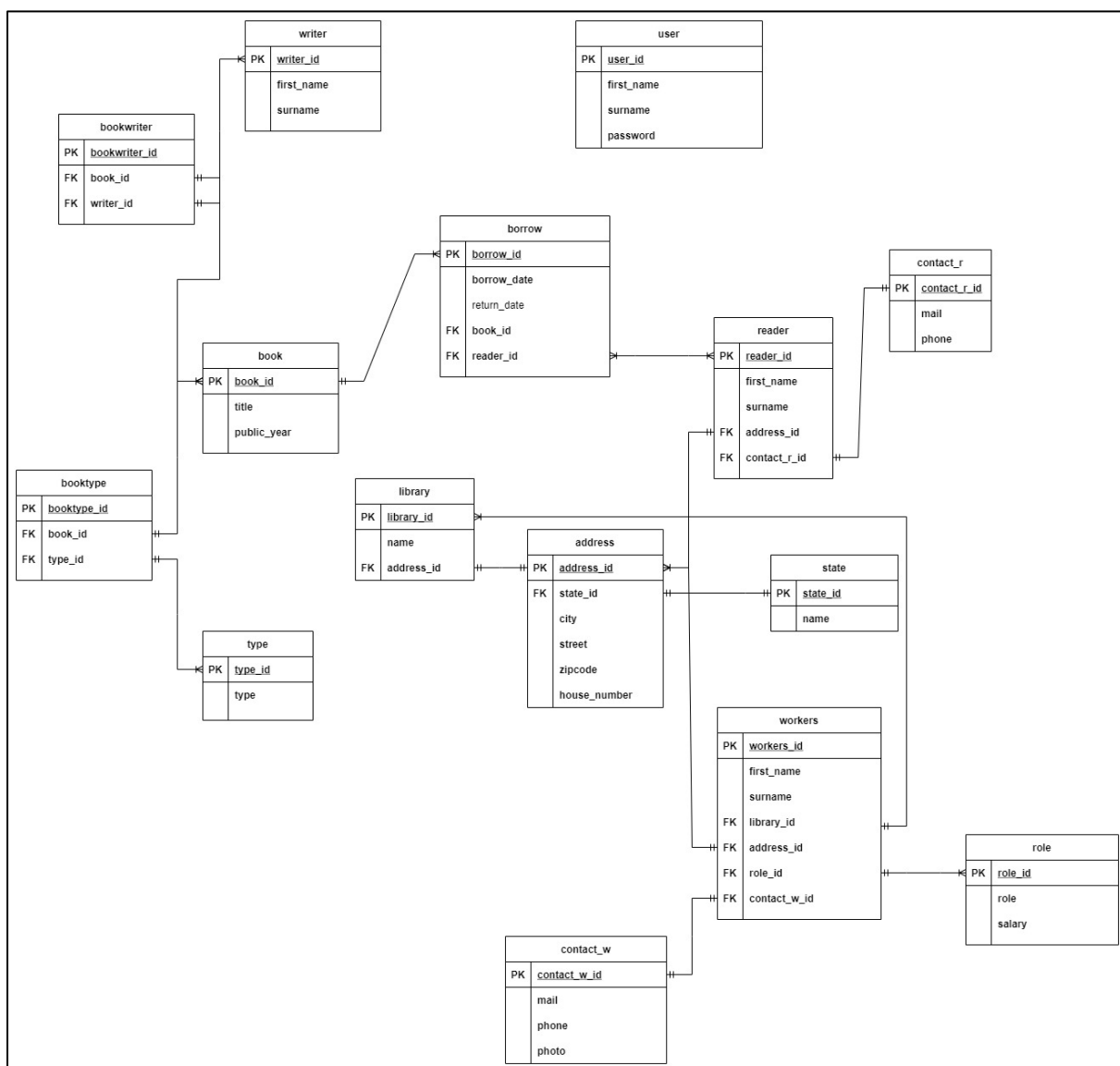
# Úvod

Tato práce shromažďuje teoretické informace o prvním projektu z předmětu Bezpečnost databázových systémů.

## 1. Odkaz na GitHub repository

<https://github.com/xvojac04/bds-db-design/tree/master>

## 2. Obrázek reprezentující databázový design



Obr. 1 Databázový design

Databázový design byl vytvořen v online softwaru draw.io.

### 3. Popis aplikace

Účel této aplikace je vytvoření databázového systému pro soubor knihoven. Jedná se o hromadný systém, kde ze samostatné tabulky user spravujeme systém několika knihoven. Každá knihovna má svou adresu a své zaměstnance. Každý ze zaměstnanců má pouze jednu funkci, není tedy možné, aby zastával více funkcí. Každý zaměstnanec má kontaktní údaje skládající se z telefonu, emailu a fotky. Dále také každý zaměstnanec může pracovat pouze v jedné knihovně.

Dále se zde nachází seznam čtenářů s jejich kontaktními údaji a jejich adresou. Každý čtenář má pouze jednu adresu, protože nám stačí adresa, kam posílat případné připomínky a nezajímá nás, zda se jedná o trvalou či přechodnou adresu. Kontaktní údaje se skládají z telefonu a emailu.

Další část databáze se skládá z tabulek o knihách. Každá kniha má svůj název a rok publikace, svého spisovatele a svůj žánr. Kniha může mít více spisovatelů, stejně tak může mít i více žánrů.

V neposlední řadě také můžeme vidět výpůjčky. Zde můžeme vidět který uživatel má půjčenou kterou knihu, ze které knihovny, kdy si ji půjčil a kdy ji má vrátit.

### 4. Popis každé tabulky

#### 4.1. Tabulka user

Tabulka user zastává roli správce. Máme zde unikátní ID pro identifikaci pod názvem user\_id s typem serial pro automatické číslování. Dále zde máme atribut name pro jeho jméno a surname pro příjmení, obě s typem varchar s délkou 64 z důvodu nutnosti uchovat text. Atribut password pak slouží k uchování jeho hesla. Tento atribut má také typ varchar s délkou 64 z důvodu nutnosti uchovat text.

#### 4.2. Tabulka writer

Tabulka writer slouží k uchování seznamu autorů. Atribut writer\_id slouží k jednoznačné identifikaci každého autora. Jedná se o serial, kvůli automatickému číslování. Atribut first\_name slouží k uložení jména každého autora. Jedná se o typ varchar s délkou 64, kvůli možnosti dlouhého jména. Typ varchar je zvolen na základě toho, že jméno je vždy znaková sada pole. Atribut surname slouží k uložení příjmení každého autora. Je zde opět ze stejných důvodů zvolen typ varchar s délkou 64.

### 4.3. Tabulka book

Tabulka book slouží k uchování seznamu všech dostupných knih. Atribut book\_id slouží jako jednoznačný identifikátor každé knihy. Jedná se o serial, kvůli automatickému číslování. Atribut title slouží pro uchování názvu knihy. Z důvodu možnosti dlouhého názvu knihy je zvolena znaková sada pole, tedy varchar, o délce 64. Atribut publicationyear slouží k uchování roku publikace knihy. Typ je zvolen jako smallint, protože tento typ svým rozsahem stačí.

### 4.4. Tabulka type

Tabulka type shromažďuje veškeré možné žánry knih. Atribut type\_id slouží jako jednoznačný identifikátor každého žánru. Jedná se o serial, kvůli automatickému číslování. Atribut type slouží pro uchování názvu samotných žánrů. Zvolen je typ varchar s délkou 32, kvůli nutnosti uchování textu.

### 4.5. Tabulka borrow

Tabulka borrow slouží k indikaci půjčení. Atribut borrow\_id slouží jako jednoznačný identifikátor každého výpůjčky. Jedná se o serial, kvůli automatickému číslování. Atribut borrow\_date slouží k určení dne, kdy k výpůjčce došlo. Atribut return\_date slouží k určení, kdy došlo k vrácení knihy. Oba atributy udržují datové informace, proto je zvolen typ timestamp. Atribut book\_id určuje id vypůjčené knihy. Atribut reader\_id určuje id čtenáře, který provedl výpůjčku. Oba tyto atributy jsou cizí klíče a zvoleny jako serial na základě toho, že všechny id v této databázi jsou serial.

### 4.6. Tabulka reader

Tabulka reader sdružuje data o čtenářích. Atribut reader\_id slouží jako jednoznačný identifikátor každého čtenáře. Jedná se o serial, kvůli automatickému číslování. Atribut first\_name slouží k uložení křestního jména a atribut surname k uložení příjmení čtenáře. Zvolen je u obou typ varchar s délkou 64. Tabulka dále obsahuje atributy address\_id a contact\_id. Tyto atributy jsou cizí klíče a jelikož databáze používá pouze serial jako id, je zvolen typ serial. Atribut address\_id slouží k přiřazení adresy. Atribut contact\_id slouží k přiřazení kontaktních údajů.

### 4.7. Tabulka address

Tabulka address nám vytváří seznam všech adres. Atribut address\_id slouží jako jednoznačný identifikátor každé adresy. Jedná se o serial, kvůli automatickému číslování. Atribut state\_id slouží k uchování id státu ke kterému adresa patří. Je zvolen typ serial, jelikož všechny id v této databázi tento typ mají. Atribut city slouží k uchování názvu měst. Je zvolen jako varchar s délkou 64, z důvodu uchování textu. Atribut street je užíván k uchování názvů ulic. Je zvolen jako varchar s délkou 64 znaků, kvůli uchování textu. Atribut zipcode slouží k uchování poštovního směrovacího čísla a atribut house\_number má účel uchování čísla domu. Oba tyto

atributy jsou zvoleny jako varchar s délkou 32, kvůli možnosti označení domu písmeny, případně písmen v poštovním směrovacím čísle.

#### 4.8. Tabulka state

Tabulka state obsahuje state\_id, jakožto jednoznačný identifikátor každé adresy. Je zvolen typ serial, kvůli automatickému číslování. Atribut name poté slouží pro zapsání názvu země. Jelikož se jedná o název je zvolen varchar s délkou 64 znaků.

#### 4.9. Tabulka contact\_r

Tabulka contactr sdružuje kontaktní údaje o čtenářích. Atribut contactr\_id slouží jako jednoznačný identifikátor každých kontaktních informací. Jedná se o serial, kvůli automatickému číslování. Atribut mail slouží pro uchování emailové adresy uživatele a je zvolen jako varchar s délkou 128 znaků pro uchování textového řetězce. V atributu phone uchováváme telefonní číslo čtenáře. Tento atribut má typ integer kvůli uchovávání čísla.

#### 4.10. Tabulka library

Tabulka library slouží pro uchování informací o jednotlivých pobočkách knihoven. Atribut library\_id slouží jako jednoznačný identifikátor každé knihovny. Jedná se o serial, kvůli automatickému číslování. Atribut name slouží pro uchování názvu knihovny. Jedná se o varchar s délkou 64, pro nutnost uchování textového pole. Tabulka také uchovává cizí klíč address\_id jako odkaz na tabulku address kde se nachází adresa dané knihovny. Tento atribut využívá typ serial, na základě toho, že všechny id v této databázi jsou serial.

#### 4.11. Tabulka worker

Tabulka worker ztělesňuje informace o jednotlivých zaměstnancích. Atribut worker\_id slouží jako jednoznačný identifikátor každého pracovníka. Jedná se o serial, kvůli automatickému číslování. Atribut first\_name slouží pro uchování křestního jména zaměstnance. Jedná se o varchar s délkou 64, pro nutnost uchování textového pole o různé délce jména. Kvůli možnosti dvou a více jmen, dlouhého jména atd. je zvolena větší délka. Atribut surname slouží pro uchování příjmení zaměstnance. Jedná se o varchar s délkou 64, pro nutnost uchování textového pole o různé délce příjmení. Kvůli možnosti dvou a více příjmení, dlouhého příjmení atd. je zvolena větší délka. Atribut role\_id je cizí klíč k určení pozice zaměstnance. Atribut library\_id je cizí klíč k identifikaci knihovny, ve které daný zaměstnanec pracuje. Atribut contactw\_id je cizí klíč k určení kontaktních údajů zaměstnance. Všechny cizí klíče jsou serial z důvodu, že všechny id v databázi jsou serial.

## 4.12. Tabulka role

Tabulka role slouží jako seznam pracovních pozic s jejich platy. Atribut `role_id` slouží jako jednoznačný identifikátor každé pozice. Jedná se o serial, kvůli automatickému číslování. Atribut `role` slouží jako textové pole pro zápis jednotlivých pozic. Jelikož se jedná o textové pole je zvolen typ `varchar` s délkou 64. Atribut `salary` slouží k uchování platů jednotlivých pozic. Toto si můžeme dovolit, protože každý zaměstnanec může mít pouze jednu pozici. Tento atribut uchováváme jako integer, z důvodu potřeby zapsání čísla.

## 4.13. Tabulka contact\_w

Tabulka `contactr` sdružuje kontaktní údaje zaměstnanců. Atribut `contactr_id` slouží jako jednoznačný identifikátor každých kontaktních údajů. Jedná se serial, kvůli automatickému číslování. Atribut `mail` slouží pro uchování emailové adresy zaměstnance a je zvolen jako `varchar` s délkou 128 znaků pro uchování textového řetězce. V atributu `phone` uchováváme soukromé telefonní číslo zaměstnance. Tento atribut má typ integer kvůli uchovávání čísla. Atribut `photo` slouží k uchování fotografie zaměstnance. Jedná se o typ `varchar` s délkou 128, která je dostatečná pro uchování cesty k fotografii.

## 4.14. Tabulky bookwriter a booktype

Tyto tabulky jsou sdruženy do jedné kapitoly z důvodu, že se jedná o spojové tabulky, které využíváme při M:M vztazích. Tyto tabulky obsahují pouze své id ve formě serial a dvakrát typ serial pro reprezentaci cizích klíčů z tabulek které spojují. U tabulky `bookwriter` je to tedy id spisovatele a id knihy a u tabulky `booktype` se jedná o id knihy a id žánru knihy.

## 5. Proč se jedná o databázi v 3 normální formě?

Věřím, že se jedná o databázi ve 3 normální formě, díky nevyskytujícím se duplicitním sloupcům. Dále databáze obsahuje samostatnou tabulku pro každou skupinu dat. Obsah žádného sloupce se mi nedaří již dále dělit na další sloupce. Z tabulky jsou odstraněny všechny data, která nezávisí přímo na primárním klíči. Veškeré duplicitní informace mám nahrazeny pouze cizím klíčem a je pro ně vytvořena další samostatná tabulka.

## 6. DDL skripty pro vytvoření databáze v SQLite

```
CREATE TABLE user (user_id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, surname TEXT, password TEXT);
```

```
CREATE TABLE state (state_id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT);
```

```
CREATE TABLE contact_r (contactr_id INTEGER PRIMARY KEY AUTOINCREMENT, mail TEXT, phone INTEGER);
```

*CREATE TABLE contact\_w (contactw\_id INTEGER PRIMARY KEY AUTOINCREMENT, mail TEXT, phone INTEGER, photo TEXT);*

*CREATE TABLE address (address\_id INTEGER PRIMARY KEY AUTOINCREMENT, state\_id INTEGER, city TEXT, street TEXT, zipcode TEXT, house\_number TEXT, FOREIGN KEY(state\_id) REFERENCES state(state\_id));*

*CREATE TABLE reader (reader\_id INTEGER PRIMARY KEY AUTOINCREMENT, first\_name TEXT, surname TEXT, address\_id INTEGER, contact\_r\_id INTEGER, FOREIGN KEY (address\_id) REFERENCES address(address\_id), FOREIGN KEY (contact\_r\_id) REFERENCES contact\_r(contact\_r\_id));*

*CREATE TABLE library (library\_id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, address\_id INTEGER, FOREIGN KEY (address\_id) REFERENCES address(address\_id));*

*CREATE TABLE role (role\_id INTEGER PRIMARY KEY AUTOINCREMENT, role TEXT, salary INTEGER);*

*CREATE TABLE worker (worker\_id INTEGER PRIMARY KEY AUTOINCREMENT, first\_name TEXT, surname TEXT, role\_id INTEGER, library\_id INTEGER, address\_id INTEGER, contactw\_id INTEGER, FOREIGN KEY (address\_id) REFERENCES address(address\_id), FOREIGN KEY (contact\_w\_id) REFERENCES contact\_w(contact\_w\_id), FOREIGN KEY (library\_id) REFERENCES library(library\_id), FOREIGN KEY (role\_id) REFERENCES role(role\_id));*

*CREATE TABLE type (type\_id INTEGER PRIMARY KEY AUTOINCREMENT, type TEXT);*

*CREATE TABLE writer (writer\_id INTEGER PRIMARY KEY AUTOINCREMENT, first\_name TEXT, surname TEXT);*

*CREATE TABLE book (book\_id INTEGER PRIMARY KEY AUTOINCREMENT, title TEXT, public\_year INTEGER);*

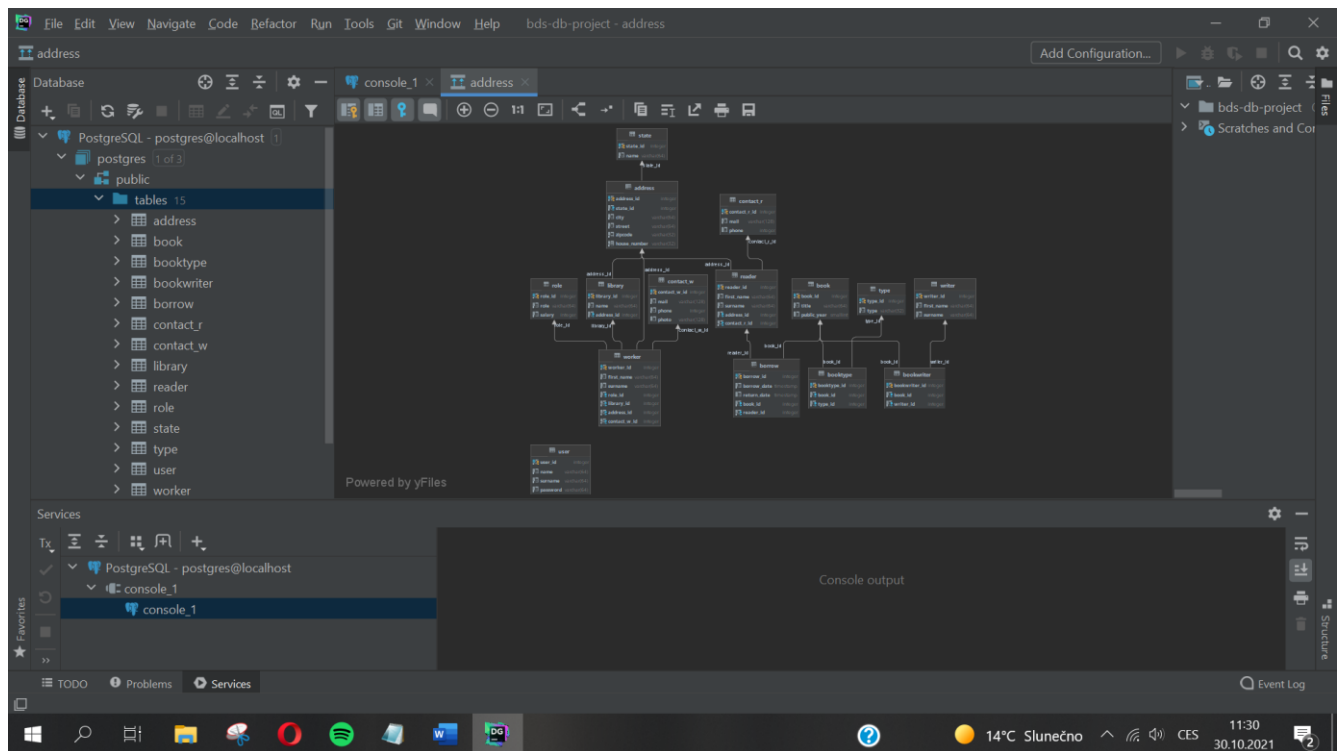
*CREATE TABLE bookwriter (bookwriter\_id INTEGER PRIMARY KEY AUTOINCREMENT, book\_id INTEGER, writer\_id INTEGER, FOREIGN KEY (book\_id) REFERENCES book(book\_id), FOREIGN KEY (writer\_id) REFERENCES writer(writer\_id));*

*CREATE TABLE booktype (booktype\_id INTEGER PRIMARY KEY AUTOINCREMENT, book\_id INTEGER, type\_id INTEGER, FOREIGN KEY (book\_id) REFERENCES book(book\_id), FOREIGN KEY (type\_id) REFERENCES type(type\_id));*

*CREATE TABLE borrow (borrow\_id INTEGER PRIMARY KEY AUTOINCREMENT, borrow\_date TEXT, return\_date TEXT, book\_id INTEGER, reader\_id INTEGER, FOREIGN KEY (book\_id) REFERENCES book(book\_id), FOREIGN KEY (reader\_id) REFERENCES reader(reader\_id));*

## 7. Důkaz o vytvoření databáze





Obr. 2 Screenshot z aplikace DataGrip(JetBrains) o skutečném vytvoření databáze

Snímek obrazovky z aplikace DataGrip od společnosti JetBrains s.r.o. spolu s vygenerovaným EM diagramem.

## Závěr

Tento projekt se projevil jako velice časově náročný, díky své obsáhlosti. Téma knihovna jsem si zvolila kvůli velkému množství informací, co se zde uchovává, což se projevilo jako krok dobrým směrem. S realizací jsem z hlediska časové tísně přijatelně spokojena a věřím, že jsem hlavní body zadání splnila.

## Seznam obrázků

Obr. 1 Databázový design .....	3
Obr. 2 Screenshot z aplikace DataGrip(JetBrains) o skutečném vytvoření databáze.....	9