

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA PEDAGOGICKÁ
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

NÁVRH, TVORBA A EDITACE SQLITE DATABÁZE
BAKALÁŘSKÁ PRÁCE

Pavel Ešner

Informatika se zaměřením na vzdělávání

Vedoucí práce: PhDr. Denis Mainz Ph.D.

Plzeň, 2017

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně
s použitím uvedené literatury a zdrojů informací.

V Plzni, 3. března 2017

.....
vlastnoruční podpis

PODĚKOVÁNÍ

Rád bych tímto poděkoval vedoucímu práce, PhDr. Denisovi Mainzovi Ph.D., za cenné rady, připomínky a odborné vedení při zpracování této práce. Dále bych rád poděkoval rodině a přátelům za podporu při studiu.

ZDE SE NACHÁZÍ ORIGINÁL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

OBSAH

| | |
|---|----|
| SEZNAM POJMŮ A ZKRATEK | 3 |
| ÚVOD | 5 |
| 1 SPECIFIKACE SQLITE V KONTEXTU INFORMAČNÍCH TECHNOLOGIÍ | 7 |
| 1.1 HISTORIE SQLITE | 7 |
| 1.2 ÚVOD DO SQLITE | 7 |
| 1.2.1 Jednoduchost..... | 9 |
| 1.2.2 Flexibilita..... | 9 |
| 1.2.3 Spolehlivost | 10 |
| 1.2.4 Datové typy v SQLite | 10 |
| 1.3 LICENCE | 11 |
| 1.3.1 Omezení v SQLite..... | 12 |
| 1.3.2 Vlastnosti SQLite..... | 13 |
| 1.3.3 Využití | 14 |
| 1.3.4 Výhody | 15 |
| 1.3.5 Nevýhody..... | 15 |
| 2 SROVNÁNÍ SQLITE S AKTUÁLNĚ POUŽÍVANÝMI DATABÁZOVÝMI SYSTÉMY | 16 |
| 2.1 ORACLE..... | 16 |
| 2.2 MICROSOFT SQL SERVER | 17 |
| 2.3 MYSQL | 19 |
| 2.4 POSTGRESQL..... | 20 |
| 2.5 POROVNÁNÍ DATABÁZOVÝCH SYSTÉMŮ..... | 21 |
| 3 VOLNĚ DOSTUPNÉ PROGRAMOVÉ NÁSTROJE | 26 |
| 3.1 SQLITE ADMINSTRATOR | 26 |
| 3.2 DB BROWSER FOR SQLITE | 27 |
| 3.3 SQLITE STUDIO | 28 |
| 3.4 SQLITE EXPERT PERSONAL | 29 |
| 3.5 SQLITE2009PRO | 30 |
| 3.6 OSTATNÍ SOFTWARE | 31 |
| 4 NÁVRH POSTUPU TVORBY SQLITE DATABÁZE..... | 35 |
| 4.1.1 Vysvětlení pojmů | 35 |
| 4.1.2 Relace | 35 |
| 4.1.3 Návrh databáze..... | 36 |
| 4.2 VYTVORENÍ DATABÁZE | 37 |
| 5 TVORBA APLIKACE NAD SQLITE DATABÁZÍ | 40 |
| 5.1 NÁVRH APLIKACE | 40 |
| 5.2 JAVAFX..... | 42 |
| 5.3 JDBC..... | 42 |
| 5.4 APACHE MAVEN..... | 42 |
| 5.5 SOUBORY APLIKACE..... | 43 |
| 5.6 APLIKACE NAD DATABÁZÍ..... | 44 |
| ZÁVĚR..... | 51 |
| RESUMÉ | 53 |
| KLÍČOVÁ SLOVA | 54 |
| SEZNAM LITERATURY | 55 |
| ELEKTRONICKÉ ZDROJE | 56 |
| SEZNAM OBRÁZKŮ | 60 |

| | |
|--|----|
| SEZNAM TABULEK | 61 |
| SEZNAM PŘÍLOH | 62 |
| PŘÍLOHY | 63 |
| PŘÍLOHA Č. 1 – POM.XML | 63 |
| PŘÍLOHA Č. 2 – BUILD.SH | 65 |
| PŘÍLOHA Č. 3 – DB-SCHEMA-INIT.SQL..... | 66 |
| PŘÍLOHA Č. 4 – OBSAH CD | 69 |

SEZNAM POJMŮ A ZKRATEK

SQL - Structured Query Language

HP – Hewlett Packard

IBM - International Business Machines Corporation

DBM - DataBase Manager

UTF - Unicode/UCS Transformation Format

RAM - Random Access Memory

ANSI - American National Standards Institute

BSD - Berkeley Software Distribution

MAC OS - Macintosh Operating System

AIX - Advanced Interactive eXecutive

PDA - Personal Digital Assistant

GB, TB, PB, EB – Zkratky jednotek datových velikostí

API - Application Programming Interface

PHP - Hypertext Preprocessor

TCP/IP - Transmission Control Protocol/Internet Protocol

XML - eXtensible Markup Language

GPL - General Public License

MS-DOS - Microsoft Disk Operating System

LAN - Local Area Network

IPV - Internet Protocol version

MAC address - Media Access Control address

JSON - JavaScript Object Notation

CSV - Comma-separated values

XLS – Microsoft Excel

HTML - HyperText Markup Language

PDF - Portable Document Format

AES - Advanced Encryption Standard

RSA - Rivest-Shamir-Adleman algorithm

CSS - Cascading Style Sheets

Úvod

V počítačovém světě vzniká denně velké množství dat, které je zapotřebí systematicky ukládat a přistupovat k nim v reálném čase. Řešení této problematiky usnadňuje existence mnoha databázových systémů, které se od sebe vzájemně liší velikostí, použitým jazykem, dostupností, přenositelností, licencováním nebo a možnostmi využití. Moje bakalářská práce je zaměřena na jeden z těchto systémů řízení báze dat - SQLite. Seznamuje s jejími základními vlastnostmi, možnostmi či omezeními a porovnává ji s aktuálně nejpoužívanějšími databázovými systémy. Dále srovnává freewarové aplikace umožňující vytvořit a editovat databázi. Databáze je následně vytvořena v nejlépe hodnocené freewarové aplikaci SQLite Studio a prakticky použita ve vlastní aplikaci s názvem Evidence.

Hlavními důvody pro volbu tématu „Návrh, tvorba a editace SQLite databáze“ byla vhodnost této knihovny pro edukační účely a množství zařízení, ve kterých je používána. Pro svoji univerzálnost se využívá v celé řadě zařízení, např. v mobilních telefonech, set top boxech, počítačových aplikacích jako je Skype či internetových prohlížečích jako je Google Chrome. Svou velikostí, přenositelností mezi platformami a hlavně podporou většiny příkazů standardu SQL-92 se z SQLite stává ideální nástroj pro tato zařízení. Vhodně zvolená aplikace pak dává možnost bez instalace pracovat s databází v grafickém prostředí s příkazy jazyka SQL. Výhody a nevýhody jednotlivých databázových systémů předurčují, pro jaký typ projektů jsou konkrétní systémy vhodné.

Cílem mé bakalářské práce je poskytnout potenciálnímu uživateli databázového systému SQLite komplexní pohled na tento systém a umožnit mu kvalifikovaně se rozhodnout o vhodnosti nasazení SQLite databáze na jeho projekt. Pokud se rozhodne použít SQLite, je seznámen s aplikacemi pracujícími s SQLite a následně s návodným postupem tvorby vlastní databáze. V praktické části pak bude seznámen se správným přístupem z grafického uživatelského rozhraní vlastní aplikace k datům v databázi.

Teoretická část bakalářské práce seznamuje s historií vzniku SQLite databáze, jejími datovými typy a vlastnostmi a shrnuje její výhody a nevýhody.

V praktické části bakalářské práce jsem využil poznatky o vlastnostech SQLite databáze a vytvořil aplikaci určenou pro základní a střední školy. Cílem této aplikace je evidence žáků, jejich rodičů, adres, předmětů, učitelů, známek a jejich vzájemných vazeb z pohledu

vyučujícího a jeho správy dat. Aplikace je naprogramována s třívrstvou architekturou rozdělenou na aplikační, logickou a databázovou vrstvu a umožňuje čtyři základní operace s databází: insert, create, update a delete. Jedná se o desktopovou aplikaci vytvořenou ve vývojovém prostředí Eclipse v programovacím jazyku Java a využívající knihovnu pro grafické rozhraní JavaFX. Aplikace přistupuje k datům přes programové rozhraní JDBC. Sestavení projektu do spustitelné podoby zajišťuje nástroj Apache Maven. Grafické rozhraní je koncipováno do přehledných panelů umožňujících uživateli snadnou orientaci a modální okna každého panelu řeší problematiku pouze konkrétní části editovaných dat. V aplikaci jsou smyšlená vzorová data ukazující funkčnost aplikace.

1 SPECIFIKACE SQLITE V KONTEXTU INFORMAČNÍCH TECHNOLOGIÍ

1.1 HISTORIE SQLITE

SQLite bylo vyvinuto ve společnosti General Dynamics¹, která se zabývá výzkumem a výrobou pro vojenské účely. Mezi výrobky firmy General Dynamics patří například ponorky třídy Ohio, Virginia či tank M1-Abrams, ale asi nejznámějším výrobkem je nadzvukový letoun F-16 Fighting Falcon. Pro vývoj softwaru řízených střel torpédoorců pro americké námořnictvo byl společností využíván program, který původně běžel na HP Unixu a využíval IBM Informix databázi. Tento systém ale nebyl pro účely výzkumu dostatečně flexibilní a byl velice náročný na administraci. Vývojář společnosti Dwayne Richard Hipp (*9. 4. 1961) s kolegou se proto začali v roce 2000 zabývat vývojem databáze fungující bez nutnosti instalace serveru či administrační podpory. Vyvinuli jednoduchou databázovou aplikaci, která je součástí programu a funguje bez ohledu na operační systém. Databáze pracuje s vnořeným jazykem SQL, GNU DBM hash knihovny².

První verzi SQLite vydal D. Richard Hipp v srpnu roku 2000, další významné upgrady následovaly v letech 2001 a 2004. Postupně rostla oblíbenost databáze u open source i komerčních projektů a psala se další rozšíření pro různé skriptovací jazyky a knihovny². Po implementaci textové podpory UTF-8 a UTF-16 následovala 25 % redukce velikosti, podpora binárních objektů (BLOB) a vacuum. Tyto změny však neměly vliv na velikost knihovny, která stále zůstávala menší než 500Kb, což je méně než kolik má většina dnešních internetových stránek. V současné době již SQLite poskytuje pokročilé funkce, jako jsou například rekurzivní *triggery*.

1.2 ÚVOD DO SQLITE

Databáze SQLite patří k druhu tzv. *embedded* databází, což znamená, že pracuje jako samostatný proces uvnitř spuštěné aplikace a není nutné používat řešení klient-server. Jednou z výhod, které mají *embedded* databáze je, že databáze funguje jako proces, který

¹ Our Businesses. *General Dynamics* [online]. Dalaware: General Dynamics, 2016 [cit. 2017-03-16]. Dostupné z: <http://www.generaldynamics.com/our-businesses>

² OWENS, Michael. *The definitive guide to SQLite*. New ed. Berkeley, Calif: Apress, 2006. ISBN 9781590596739.

se spouští pouze v případě potřeby a jinak nezatěžuje systémové prostředky zařízení, na kterém je spuštěn.

Vzhledem k tomu, že se jedná o knihovnu, která se připojí k programu, není nutné konfigurovat síťová nastavení, firewall či nastavovat složitá oprávnění. Jedná se tedy o databázový systém, který je pro svou univerzálnost velmi vhodný pro výuku a úvod do relačních databází.

SQLite může být použita také jako vyrovnávací paměť, ukládat konfigurační data nebo, pokud využijeme binární kompatibilitu napříč platformami, může dokonce pracovat jako formát souboru aplikace³.

SQLite může také být konfigurována tak, aby používala pevně zadanou velikost RAM paměti a pokud překročí stanovenou mez, tak uloží data na pevný disk. Navzdory své malé velikosti SQLite nabízí řadu funkcí díky podpoře většiny možností standardu ANSI SQL92⁴, mezi které patří transakce, složené dotazy a nalezneme zde i rysy relačních databází. Mezi tyto rysy můžeme zařadit trigger, cizí klíče, indexy nebo automatickou inkrementaci číselných hodnot sloupců.

Základní myšlenka návrhu SQLite byla vždy přenositelnost, a proto dnes můžeme SQLite používat na systémech Windows, Linux, BSD, MAC OS X, či na komerčních systémech Unixu jako jsou Solaris, HP-UX a AIX⁵. SQLite v současné době můžeme nalézt i v systémech mobilních telefonů a PDA jako Android, QNX, Palm OS, Windows CE, a to jak na 32bitové tak i na 64bitové platformě. Tímto však přenositelnost nekončí, neboť formát souboru databáze je binárně kompatibilní napříč všemi podporovanými operačními systémy. Jinými slovy: pokud vytvoříme databázi v operačním systému Linux, můžeme ji přenést a upravovat na operačním systému Windows, Mac a ostatních podporovaných systémech. Databáze může pojmout až 140 TB dat a soubor tedy může být omezen pouze souborovým

³ JAY A. KREIBICH. *Using SQLite*. Sebastopol, CA: O'Reilly, 2010. ISBN 9780596521189.

⁴ SQLite Overview. *Tutorials Point* [online]. Madhapur, 2017 [cit. 2017-03-16]. Dostupné z: https://www.tutorialspoint.com/sqlite/sqlite_overview.htm

⁵ Features Of SQLite. *SQLite* [online]. SQLite, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.sqlite.org/features.html>

systémem daného operačního systému⁶. Alokuje však minimum prostředků pro každé uložené pole, neboť používá variabilní délku záznamů.

1.2.1 JEDNODUCHOST

Jednoduchost patří mezi hlavní cíle vývojářů SQLite. Databáze proto disponuje snadnou údržbou, správou a není nutné při používání SQLite nic instalovat či konfigurovat. I programovací knihovny SQLite API⁷ patří mezi jedny z nejjednodušších, a to díky jejich kvalitní dokumentaci. Jsou navrženy tak, aby pomohly přizpůsobit jednoduše vlastní funkce v SQLite. Konstrukce SQLite je postavena tak, aby umožnila plnou rozšiřitelnost a zároveň si zachovala vysoký stupeň jednoduchosti. Každý modul je speciální, nezávislý systém, který provádí specifický úkol. Open source komunita již také vytvořila mnoho jazykových rozhraní a knihoven, které je možné využívat například pro Perl, Python, Java, PHP, Visual Basic, Delphi⁸ a mnoho dalších, jejichž kompletní seznam je uveden na oficiálních stránkách SQLite. Pokud se zamyslíme nad touto koncepcí jednoduchosti a vezmeme v úvahu velikost SQLite knihovny, bude se zákonitě jednat o velmi rychlý databázový systém.

1.2.2 FLEXIBILITA

Nejflexibilnější vlastnost databáze je bezesporu její přístup k datovým typům. Databáze je deklarována jako „typeless“, což znamená, že data je možné ukládat jako libovolný typ bez ohledu na deklaraci typu. Libovolnému textu je tedy možné přiřadit datový typ INTEGER. Nejedná se o chybu vývojářů, ale je to cílená vlastnost databáze, i když je potřeba zvážit, zda takto vytvořenou databázi bude možné použít v jiném databázovém programu, který nebude mít tuto vlastnost a zda bude vůbec funkční.

Jedinou výjimkou „typeless“ jsou data, jejichž typ je PRIMARY KEY, který musí být označen jako INTEGER a nelze použít ani INT, neboť INT je stejně jako ostatní typy „typeless“. Vložení jiného datového typu než INTEGER skončí chybou.

⁶ Limits In SQLite. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-16]. Dostupné z: <https://www.sqlite.org/limits.html>

⁷ An Introduction To The SQLite C/C++ Interface. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-16]. Dostupné z: <https://www.sqlite.org/cintro.html>

⁸ Wiki Index. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-16]. Dostupné z: www.sqlite.org/cvstrac/wiki?p=SqliteWrappers~~doobj

Jako embedded databáze nabízí SQLite výkon a flexibilitu relační databáze. Není nutné konfigurovat žádné velké databázové servery nebo nastavovat síťová připojení či firewall. Vezmeme-li ještě v úvahu přenositelnost databázového souboru napříč různými platformami a licence, pod kterou je SQLite uvolněna, dostáváme do ruky vskutku zajímavý nástroj.

1.2.3 SPOLEHLIVOST

Každý z nás očekává od programů a veškerého programového vybavení maximální spolehlivost, a to tím spíše, pokud se jedná o databázi, kde ztráta dat může způsobit velké škody. SQLite tuto spolehlivost může nabídnout díky nastavené politice vývoje nových verzí, neboť každá nová verze je velice pečlivě testována a podrobena více než deseti milionům automatických testů. Testuje se na chybu Out-of-memory, tedy zda nedojde k nedostatku paměti, Crash testem se zjišťuje, zda nedojde k poškození databáze při neočekávané chybě operačního systému nebo výpadku napájení, fuzz testování zajišťuje stabilitu programu při zadání chybných či neplatných dat. Samozřejmostí testů jsou i testy zpětné kompatibility, které jsou zaměřené na SQL syntaxi, formáty souborů a programování API. Tyto testy zajišťují opravdu vysokou stabilitu SQLite databáze a je vidět, že vývojáři SQLite berou stabilitu velmi vážně. Další testy jsou detailně popsány na oficiálních stránkách SQLite⁹.

1.2.4 DATOVÉ TYPY V SQLITE

Pomineme-li skutečnost, že SQLite je „typeless“, disponuje také několika skupinami datových typů.

- INTEGER obsahuje Int, Integer, TinyInt, SmallInt, MediumInt, BigInt, Unsigned BigInt, Int2 a Int8. Jedná se o celé číslo uložené v 1,2,3,4,6 nebo 8 bytech v závislosti na velikosti hodnoty.
- NULL- prázdná hodnota, jedná se o položku, u které není povinnost ji vyplnit.
- REAL obsahuje Real, Double, Double precision, Float představené 8 bytové číslem uložené s pohyblivou řádovou čárkou dle standardu IEEE754.

⁹ SQLite. *How SQLite is tested* [online]. The SQLite Consortium, 2017 [cit. 2017-04-01]. Dostupné z: <https://sqlite.org/testing.html>

- TEXTu náleží Character, Varchar, Varing Characte, Nchar, Native Character, Nvarchar, Text, Clob a jsou to textové řetězce uložené kódováním UTF-8 či UTF-16. Počet znaků zde není limitován, pokud není použit limit SQLITE_MAX_LENGTH.
- BLOB- uložení binárních dat (jako např. obrázků, zvukových či video souborů).
- NUMERICu náleží Numeric, Decimal, Boolean, Date, Datetime. Všechny číselné datové typy jsou převedeny na celé nebo desetinné číslo¹⁰.

V databázi nenajdeme datový typ BOOLEAN, místo kterého jsou zde uloženy logické hodnoty „1“ (True) a „0“ (False). Datum a čas nemá v SQLite definovaný datový typ a je ukládán jako TEXT, INTEGER nebo REAL. TEXT ukládá datum a čas dle ISO8601 jako String ve formátu („YYYY-MM-DD HH:MM:SS.sss“), REAL je definován jako počet dní, které uplynuly od poledne světového času podle proleptického Juliánského kalendáře 1. ledna 4713 př. n. l., případně 24. listopadu 4714 př. n. l. podle proleptického Gregoriánského kalendáře a INTEGER je ukládán stejně jako v UNIXu, tj. jako počet sekund od 1970-01-01 00:00:00 UTC. Kvůli maximální kompatibilitě mezi různými databázemi podporuje SQLite koncepci „Affinity type“, která funguje tak, že pokud připojíme jinou databázi k SQLite, přetypuje si databáze datové typy na své vlastní datové typy.

1.3 LICENCE

Všechny SQLite kódy jsou Public domain, což znamená, že autoři se vzdávají veškerých autorských práv na vlastnictví kódu. Autoři SQLite tedy neuplatňují autorská práva zdrojového kódu nebo knihovny. Každý vývojář musí podepsat prohlášení, že nebude vyžadovat autorské poplatky za provedené změny, které provedl ve zdrojovém kódu. V České republice však není možné licenci Public domain použít, protože podle zákona č. 121/2000 Sb. o právu autorském se nikdo nemůže vzdát autorských práv a majetková práva trvají po dobu autorova života a 70 let po jeho smrti. Skutečnost, že kódy jsou Public domain zde proto funguje tak, že autor pouze nezažaluje nikoho, kdo použije jím vytvořený kód.

¹⁰ SQLite. *Datatypes In SQLite Version 3* [online]. The SQLite Consortium, 2017 [cit. 2017-04-01]. Dostupné z: <https://sqlite.org/datatype3.html>

Public domain licence umožňuje každému používat zdrojový kód libovolně dle vlastních potřeb. Kód je možné upravovat, distribuovat nebo prodávat, a to i za komerčním účelem. Zdrojový kód SQLite nepodléhá žádným omezením ani poplatkům.

1.3.1 OMEZENÍ V SQLITE

SQLite byla původně navržena bez striktně nastavených limitů a jediným omezením byly pouze operační paměti a diskový prostor zařízení, na němž byla databáze spuštěna. To se později ukázalo jako chybné řešení, protože pokud maximální hranice nebyly správně definovány, nemohly být ani testovány, a to způsobovalo mnoho chyb včetně bezpečnostních problémů. V nových verzích jsou již tyto limity definovány a jsou součástí testovacích procesů.

V případě potřeby lze některé limity pomocí příkazů Pragma změnit a přizpůsobit si databázi podle konkrétních požadavků.¹¹ Pragma může do značné míry ovlivnit nejen výkon, ale i to jak se bude databáze chovat. Při použití těchto příkazů si však musíme dát pozor na správný zápis, protože špatně napsaný příkaz nás nijak neupozorní na chybu, jen bude v tichosti ignorován. Dále zde chybí záruka zpětné kompatibility⁹, takže vždy záleží i na aktuálně používané verzi.

Při tvorbě databáze nás bude v první řadě zajímat velikost databáze, kterou můžeme vytvořit. Každý databázový soubor je složen z tzv. stránek, kterých může být až 2147483646 o velikosti až 65536 bytů⁹. Tím dostaneme maximální možnou velikost databáze 140TB⁹. Jedná se spíše o teoretickou velikost a ani vývojáři nejsou schopni tuto maximální velikost otestovat. Pokud nepoužijeme Unixový souborový systém UFS2, který umí pracovat se souborem až 32PB¹², nemůžeme dosáhnout této velikosti na NTFS ani EXT4, které podporují soubor o maximální velikosti 16TB¹³.

Maximální počet sloupců je v SQLite databázi nastaven na 2000. I tuto hodnotu lze nastavením limitů zmenšit nebo zvětšit až na 32767⁹. Pokud však budeme mít dobře

¹¹ SQLite. *PRAGMA Statements* [online]. The SQLite Consortium, 2017 [cit. 2017-04-01]. Dostupné z: <https://www.sqlite.org/pragma.html>

¹² Filesystem Limits. *Purdue University* [online]. West Lafayette: Purdue University, 2017 [cit. 2017-03-20]. Dostupné z: <https://engineering.purdue.edu/ECN/Support/KB/Docs/FilesystemLimits>

¹³ Ext4: evoluční souborový systém. *ROOT.CZ* [online]. ROOT.CZ: root.cz, 2008 [cit. 2017-03-20]. Dostupné z: <https://www.root.cz/clanky/ext4-evolucni-souborovy-system/>

normalizovanou databází, měli bychom se vejít maximálně do 100 tabulek, a proto nám i toto omezení bude plně postačovat a vyhledávání nebude databázi zpomalovat.

Složený příkaz SELECT se skládá ze dvou nebo více příkazů SELECT spojených operátory UNION, UNION ALL, EXCEPT nebo INTERSECT¹⁴. V SQLite je výchozí hodnota nastavena na 500, ale v praxi se sloučené příkazy pohybují maximálně v jednotkách. Ve verzi 3.6.18 je již zavedena podpora rekurzivních Triggerů a od verze 3.7.0 jsou triggerly ve výchozím nastavení povoleny¹⁵. Výchozí nastavení hloubky rekurze je 1000, ale je možné je pomocí PRAGMA RECURSIVE TRIGGERS vypnout.

Maximální počet řádků databáze je 2^{64} , což je $1.8 \cdot 10^{19}$, ale dříve než bychom tohoto limitu dosáhli, narazíme na maximální velikost databáze 140TB, která je schopna pojmout pouze 10^{13} řádků, a to ještě za předpokladu, že v databázi bude velmi málo údajů a nebude obsahovat indexy¹⁶. Pokud se tedy zamyslíme nad tím, jaká omezení nám vývojáři implicitně nastavili v knihovně SQLite, dojdeme k názoru, že základní limity nás pravděpodobně omezovat nebudou, ale spíše to bude pouze základní vlastnost využití databáze s více uživateli, což by musela řešit aplikace nad databází.

1.3.2 VLASTNOSTI SQLITE

SQLite je relační databáze, což znamená, že v databázi jsou data uspořádána do tabulek obsahujících sloupce a řádky a je uložena na lokálním médiu jako jeden soubor, který obsahuje kromě vlastních dat také indexy, tabulky, triggerly a view.

SQLite je také transakční databázový stroj řešící zabezpečení dat v databázi. Tyto vlastnosti jsou nazývány zkratkou ACID¹⁷, což jsou počáteční písmena slov Atomic, Consistent, Isolated a Durable.

¹⁴ SQL As Understood By SQLite. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-20]. Dostupné z: https://www.sqlite.org/lang_select.html

¹⁵ Release History. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.sqlite.org/changes.html>

¹⁶ Limits In SQLite. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-16]. Dostupné z: <https://www.sqlite.org/limits.html>

¹⁷ SQLite Transaction. *SQLite* [online]. SQLite Tutorial, 2017 [cit. 2017-03-20]. Dostupné z: <http://www.sqlitetutorial.net/sqlite-transaction/>

Atomic čili nedělitelnost znamená, že při zápisu dat do databáze nebudou data zapsána, dokud nebude dokončen celý zápis. Databáze si při začátku zápisu nejprve uloží původní data do záložního souboru a teprve po potvrzení změn dojde ke smazání tohoto záložního souboru. Pokud by došlo například k výpadku elektrické sítě v polovině zápisu bloku dat, nedojde k zapsání žádné části dat do databáze a databáze bude při opětovné otevření obnovena ze záložního souboru před zápisem změn⁹.

Consistent čili konzistence znamená, že databáze musí zajistit konzistenci dat při každé transakci, která bude provedena. Data v databázi budou vždy konzistentní, nebude tedy porušena databázová integrita⁹.

Isolated čili izolování transakcí mezi klienty znamená, že pokud klient provádí třeba zápis do databáze, je databáze uzamčena a změny vidí pouze klient, který je provádí. Pro ostatní klienty budou změny dostupné až po ukončení zápisu. V případě přístupu více klientů najednou jsou klienti řazeni podle času přístupu a klient musí vždy počkat na dokončení operace klientem, který se přihlásil dříve⁹.

Durable čili trvanlivost nám zajišťuje, že data která zapíšeme, budou okamžitě zapsána na uložistiště a budou uložena trvale a nezávisle na případném pádu aplikace či výpadku napájení. Pokud dojde k jakémukoliv přerušení při zápisu dat, nebudou změny v databázi uloženy.

SQLite podporuje také triggerry, které použijeme v případě, kdy chceme zautomatizovat nějakou událost. V případě smazání uživatele mohu pomocí triggeru automaticky smazat z databáze či přesunout do archivu databáze také další data vedená daným uživatelem. Pokud například v školní databázi budeme z databáze mazat učitele, který přešel na jinou školu, bude určitě vhodné, aby se automaticky odstranily i záznamy adresa, seznam předmětů, které vyučoval či seznam tříd, kde působil jako třídní učitel.

1.3.3 VYUŽITÍ

Vzhledem k malé velikosti, rychlosti, přenositelnosti a spolehlivosti je možné SQLite používat téměř všude tam, kde není potřeba robustní databáze s nutností přístupu více klientů najednou. V dnešní době nalezneme SQLite v každém Androidu, iPhoneu a IOSu, Macu, Windows10, webovém prohlížeči Firefox, Chrome a Safari či Dropboxu. Je také

součástí PHP a Pythonu a dokonce ji nalezneme i jako součást televizorů, Set-top-boxů nebo v automobilních multimediálních systémech. Dle oficiálních stránek se celkově jedná o miliardy aplikací. Jednoduchost a nízká náročnost SQLite databáze přímo vybízí k jejímu používání k výukovým účelům a základnímu seznámení s jazykem SQL. Použít ji můžeme i v projektech, kde nebudeme potřebovat využívat složité funkce jazyka SQL a vystačíme si funkcemi SQLite, neboť jak již sám název napovídá, jedná se odlehčenou databázi.

1.3.4 VÝHODY

- Velmi rychlá a malá knihovna
- Snadná přenositelnost mezi platformami
- Licence Public Domain
- Bez nutnosti instalace
- Vhodná pro vývoj a testování

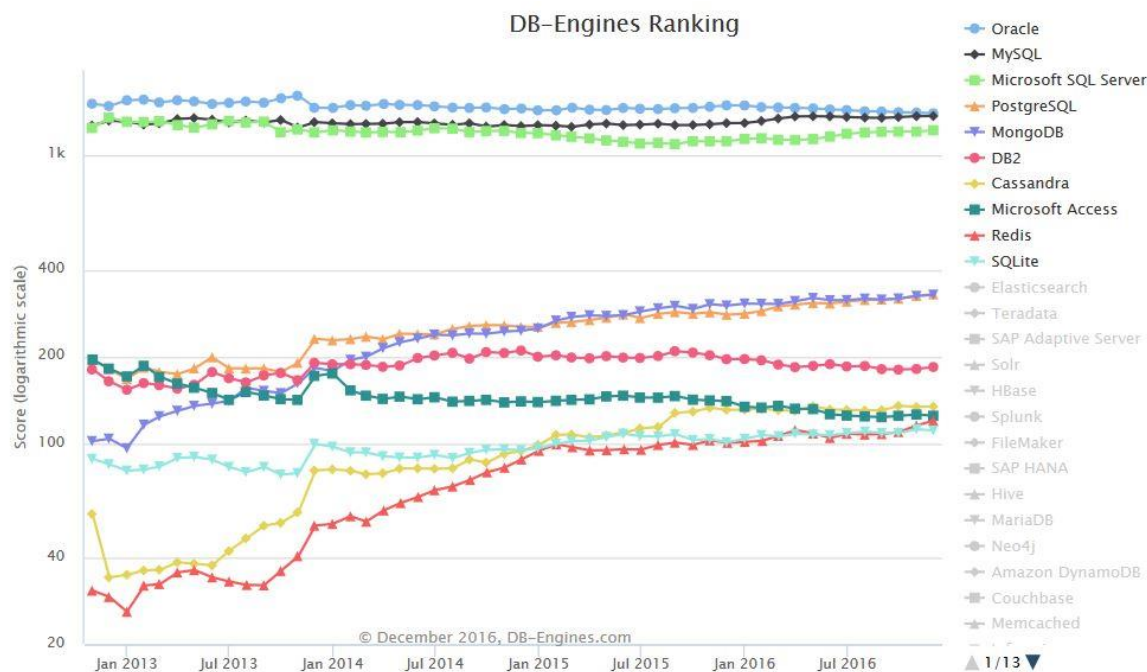
1.3.5 NEVÝHODY

- Při zápisu SQLite uzamkne celý databázový soubor a po celou dobu zápisu není možné k souboru přistupovat jinak než pro čtení. Při zápisu více procesů najednou je toto řešeno tak, že další přístupy jsou řazeny do fronty a teprve po ukončení předchozího zápisu je umožněn zápis následnému přístupu v pořadí. Toto omezení je nepraktické při použití databáze v aplikaci, ke které přistupuje více klientů, kteří budou vždy čekat na přístup k souboru.
- Nemožnost řízení uživatelských práv
- Neexistence serverové komunikace např. TCP/IP

Omezení uložení celé databáze pouze v jednom souboru nelze označit jednoznačně jako výhodu či nevýhodu, neboť záleží na tom, v jakém projektu budeme SQLite využívat.

2 SROVNÁNÍ SQLITE S AKTUÁLNĚ POUŽÍVANÝMI DATABÁZOVÝMI SYSTÉMY

Abychom mohli srovnat aktuálně používané databázové systémy, musíme nejdříve vědět, které to jsou. Na následujícím grafu je seřazeno 10 nejpopulárnějších databází a graf jejich popularity od listopadu 2012 do prosince 2016. Z grafu vyplývá, že mezi dlouhodobě nejpopulárnější databázové systémy patří Oracle, MySQL, Microsoft SQL Server a PostgreSQL. Proto bude vhodné se zaměřit na srovnání SQLite databáze právě s uvedenými systémy. Za zmínku by pravděpodobně stála i MariaDB, neboť její oblíbenost velice rychle roste a jedná se o databázi vyvíjenou přímo původním zakladatelem MySQL Michaelem Wideniusem, ale zatím nepatří mezi nejoblíbenější databázové systémy, a proto nebyla zařazena do hodnocení. Pravděpodobně to je způsobeno začátkem jejího vývoje až v roce 2009.



Obrázek 2.1 Graf oblíbenosti databázových systémů

Zdroj: http://db-engines.com/en/ranking_trend

2.1 ORACLE

Databázový systém Oracle Database patří dlouhodobě k nejsilnějším hráčům na trhu s databázovými systémy, kde působí již od roku 1977. Každá vydaná verze je označena číslem následovaným malým písmenem, které odkazuje na změny, na které se vývojáři

zaměřili. Například v aktuálně poslední vydané verzi nesoucí označení „12c“ je nejvíce řešena problematika Cloudu.

Software Oracle database můžeme získat v několika placených verzích, kde můžeme zakoupit licenci podle počtu použitých procesorů nebo počtu uživatelů. V bezplatné verzi Oracle XE je omezení možnosti využití pouze jednoho procesru, 1GB RAM a velikostí databáze 11GB.¹⁸ Maximální velikost databáze je omezena pouze použitým operačním systémem. Teoretická velikost lze spočítat vynásobením maximální velikosti datového souboru a maximálního počtu datových souborů, které databáze podporuje. Oracle ve verzi 9i a 10g podporuje maximálně 65 536 datových souborů a maximální velikost datového souboru je 128GB. Pokud vynásobíme počet datových souborů a velikost datového souboru, získáme teoretickou velikost databáze 8 exabytů.¹⁹

Pro ukládání řetězců používá Oracle Database datový typ Char, Varchar a Nchar či Nvarchar2 pro ukládání řetězců ve znakové sadě Unicode²⁰. Uložení data a času je možné v datovém typu Date a Timestamp, přičemž „Timestamp with local zone“ podporuje uložení času i s časovým pásmem. Samozřejmostí je podpora XML. Práci se soubory nám umožňují datové typy Blob, Clob, Nclob a Bfile, jejichž maximální velikost je 128 TB¹⁸.

Oracle Database je robustní, komplexní systém, který se dlouhodobě drží na prvních místech oblíbenosti (viz Obrázek 2.1), což může být způsobeno jeho kvalitou, ale zrovna tak i dlouhodobým vývojem, kdy vývojáři za dlouhá léta nasbírali mnoho zkušeností s databázovými systémy.

2.2 MICROSOFT SQL SERVER

Microsoft SQL server je vyvíjen firmou Microsoft již od roku 1989, kdy byla vydána první 16bit verze 1.0 s kódovým jménem „Filipi“ až do současné verze 13.0 vydané v roce 2016, která jako první verze nenesé žádné jméno.

¹⁸ Oracle licensing. *Oracle FAQ's* [online]. Oracle FAQ's, 2017 [cit. 2017-03-20]. Dostupné z: http://www.orafaq.com/wiki/Oracle_licensing

¹⁹ Max size of datafiles. *Community Oracle* [online]. 2007 [cit. 2017-04-02]. Dostupné z: <https://community.oracle.com/thread/521373>

²⁰ Oracle Data Types. *Oracle Help Center* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm

Získat můžeme MS SQL server v několika verzích. Placené jsou verze Enterprise, Standart pro řešení potřeb business intelligence, Web licence, kde se cenu dozvíme od našeho poskytovatele či Developer licence za sníženou cenu určenou pro vývojáře k testování aplikací. Bezplatné jsou licence Express určené pro stolní počítače a databázové servery do 10GB²¹.

V Microsoft SQL Serveru najdeme funkce, jako jsou transparentní šifrování dat, zrcadlení databáze v placených verzích, podpora šifrování zálohy, podpora zálohování do cloudového systému Azure nebo vysoká dostupnost a obnova při havárii²².

Limity databáze jsou nastaveny podle použité verze. Vzhledem k tomu, že nás budou zajímat maximální možnosti MS SQL serveru, musíme se zaměřit na nejvýkonnější a nejdražší verzi Enterprise. Enterprise má pouze jediný limit, a to je velikost relační databáze o maximální velikosti 524PB. Možnost využití procesorů a počtu jejich jader je omezena pouze operačním systémem stejně jako využití pamětí RAM. MS SQL server disponuje širokou škálou datových typů rozdělených do kategorií Exact numerics, Approximate numerics, Date and Time, Character strings, Unicode character strings, Binary strings, kdy každá kategorie obsahuje svou skupinu datových typů řešících danou problematiku a ostatní datové typy, mezi které patří Cursor, Hierarchid, sql_variant, Timestamp, Uniqueidentifier a XML²³. Instalovat MS SQL server je možné na platformy 32bit i 64bit, ale pouze na produkty vytvořené firmou Microsoft a jejich virtuální systémy²⁴. V žebříčku oblíbenosti zaujímá třetí místo, mimo jiné je používán i velmi rozšířeným a oblíbeným účetním softwarem Pohoda²⁵.

²¹ Ceny systému SQL Server. *SQL Server 2016* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2016-pricing>

²² Porovnání verzí systému Microsoft SQL Server. *SQL Server 2016* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2016-comparison>

²³ Editions and Supported Features for SQL Server 2016. *SQL Server 2016* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: <https://msdn.microsoft.com/en-us/library/cc645993.aspx>

²⁴ Hardware and Software Requirements for Installing SQL Server 2012. *SQL Server 2016* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ms143506\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/ms143506(v=sql.110).aspx)

²⁵ Instalace MS SQL Server 2014 Express. *SQL Server 2016* [online]. Jihlava: STORMWARE Software Development, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.stormware.cz/podpora/faq/sql-server.aspx>

2.3 MySQL

MySQL je relační databázový systém napsaný v jazyce C/C++. Jedná se o multiplatformní, systémovou, víceuživatelskou databázi. Podporuje standardy SQL:2003²⁶. MySQL můžeme získat ve verzích Standart, Enterprise a Cluster Carrier Grade edicích, které jsou zpoplatněné pro komerční použití a samozřejmě s podporou více funkcí²⁷. Nás však bude zajímat MySQL Community edition, která je uvolněna pod GPL licenci a těší se velké oblibě mezi uživateli i open source vývojáři.

MySQL podporuje velmi širokou škálu datových typů a jen pro celá čísla máme k dispozici SmallInt, MediumInt, Int, Integer a BigInt a pro desetinná čísla používá Float a Double, která se liší pouze použitím 4Bytů pro jednoduchou přesnost a 8Bytů pro dvojitou přesnost.²⁸ Pro práci s datem a časem jsou k dispozici typy Date, Time, Datetime, Timestamp a Year.²⁹

MySQL patří do rodiny systémů klient/server, a proto nám dává některé výhody oproti SQLite, jako je například zabezpečené přihlášení uživatele k serveru a ověření veškerého přenosu dat uživatelským jménem.

Na oficiálních stránkách vývojáři také ubezpečují o velké rychlosti, spolehlivosti a snadné použitelnosti MySQL serveru, který lze přizpůsobit výpočetnímu výkonu stanice, na které je spuštěn. Pokud je spuštěn na stanici, která je vyhrazena pouze pro účely serverové části, je zde možné nastavit maximální využití hardwaru. Pokud se podíváme na limity MySQL, zjistíme, že je zcela jistě nevyužijeme a budou nám plně postačovat. MySQL používá pro ukládání dat různé formáty uložení dat, a proto je většina limitů podřízena pouze těmto formátům. MySQL například nemá žádná omezení pro počet databází, ale použitý souborový systém může mít určitá omezení pro počet vytvořených adresářů. Zrovna tak nejsou definována žádná omezení pro počet tabulek, ale souborový systém *InnoDB*

²⁶ What is MySQL? *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20].
Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

²⁷ MySQL Editions. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20].
Dostupné z: <https://www.mysql.com/products/>

²⁸ The Main Features of MySQL. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20].
Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/features.html>

²⁹ Data Types. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20].
Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

umožňuje použití „pouze“ 4 miliard tabulek³⁰. MySQL může použít pro práci s databází více tabulkových souborů, které obecně dělíme v případě velikosti větší než 1TB. Dalším limitem tak bude velikost celé databáze, ale při použití souborového systému *MyISAM* máme implicitně nastavenou velikost na 256TB, která může být při změně limitů zvětšena až na maximální hodnotu 65,536TB (256^{7-1} bytes).³¹ Počet sloupců v tabulce je striktně nastavený na 4096, ale i zde záleží na použitém souborovém systému. Například *InnoDB* má limit 1017 sloupců v tabulce.³²

MySQL patří mezi velmi oblíbené databázové systémy, o čemž jistě svědčí i její nasazení v mnoha hostingových serverech, kde se stala naprostou samozřejmostí a základním pilířem každého hostingu.

2.4 POSTGRESQL

Databázový systém PostgreSQL nebyl zvolen jen proto, že je na čtvrtém místě v oblíbenosti databázových systémů, ale hlavně proto, že je v současné době asi největším konkurentem MySQL. PostgreSQL je relační databázový systém, který jako open source projekt vznikl v roce 1996 ve verzi 6.0 z původní verze Postgre95³³. Nyní již PostgreSQL podporuje cizí klíče, trigger, view, JOIN operace a je plně ACID kompatibilní. Můžeme ho nasadit na Unix, Microsoft windows, Linux a navíc, pokud použijeme emulační program Cygwin, je možné ho spustit i na MS-DOS.

Zajímavá je podpora datových typů, mezi kterými nalezneme datový typ „money“ pro práci s financemi, geometrické datové typy či LAN datové typy pro práci s IP adresou ve formátu IPV4 a IPV6 či MAC adresou. Podporuje také moderní datové typy XML a JSON³⁴.

³⁰ Limits on Number of Databases and Tables. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/database-count-limit.html>

³¹ Limits on Table Size. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/table-size-limit.html>

³² Limits on Number of Databases and Tables. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/database-count-limit.html>

³³ PostgreSQL. *PostgreSQL* [online]. Global Development Group, 2017 [cit. 2017-03-20]. Dostupné z: <http://postgres.cz/wiki/PostgreSQL>

³⁴ PostgreSQL. *PostgreSQL* [online]. Global Development Group, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.postgresql.org/docs/9.1/static/datatype.html>

Limity má tento databázový systém nastaveny na neomezenou velikost databáze stejně jako počet řádků v tabulce a indexy, velikost tabulky 32TB a počet sloupců v tabulce 250 – 1600 dle typu.

Licencované je PostgreSQL vlastní liberální open source licenci „PostgreSQL Licence“, a to proto, že vývojáři chtěli vlastní licenci, která nebude s žádnou jinou zaměnitelná. Postgre je velmi oblíbená aplikace, což nám nedokazuje jen graf oblíbenosti databázových systémů, ale i její široké nasazení na internetu, v bankovníctví, průmyslu, vědě, výzkumu i vzdělání, kde je možné tyto reference doložit nasazením Postgre na Jihočeské univerzitě v Českých Budějovicích a katedře botaniky Univerzity Karlovy v Praze³⁵.

2.5 POROVNÁNÍ DATABÁZOVÝCH SYSTÉMŮ

Porovnání databázových systémů je obecně velmi těžké a nelze je srovnávat bez vazby na účel jejich použití. O tom, jaký systém pro naši databázi nasadíme, rozhoduje několik aspektů - jak velký projekt chceme realizovat, co budeme od systému požadovat pro naši databázi, jaké má databázový systém vlastnosti a zda je pro nás dostačující rychlost. Důležitý je samozřejmě i rozpočet plánovaného projektu. Pro realizaci malého či středního projektu si určitě vystačíme s open source softwarem, ale při realizaci většího projektu budeme muset pravděpodobně sáhnout po placeném softwaru, který nám nabídne více možností a větší výkon. V případě, kdy nám stačí souborový systém ukládání dat, můžeme využít SQLite, Microsoft Access nebo FoxPro a pro systémové databáze například MySQL, Postgre nebo Oracle. Rychlé srovnání základních vlastností máme k dispozici v následující tabulce.

³⁵ PostgreSQL. *PostgreSQL* [online]. Global Development Group, 2017 [cit. 2017-03-20]. Dostupné z: http://postgres.cz/wiki/Reference_pou%C5%BEit%C3%AD_PostgreSQL_v_%C4%8CR

Tabulka 1 Porovnání databázových systémů

Zdroj: Vlastní

| | Podpora více uživatelů | Systémová/Souborová databáze | Podpora platform | Licence | Podpora datových typů | Podpora serveru | SQL standard | Ověření uživatele |
|---------------|------------------------|------------------------------|--|----------------------------------|---|-----------------|--------------|-------------------|
| SQLite | Ne | souborová | Windows, Linux, BSD, Mac OS X, Unix, HP-UX, Solaris, Android ³⁶ | Public Domain | Typeless , (Int, Integer, TinyInt, SmallInt, MediumInt, BigInt, Unsigned Big Int, Int2, Int8, Date, Character, Varchar, Varing, Real, Characte, Nchar, Native, Blob, Character, Text, Nvarchar, Clob, Double, Double precision, Float, Numeric, Decimal, Boolean, Datetime | Ne | SQL-92 | Ne |
| MySQL | Ano | systémová | Linux, Solaris, Microsoft windows, IBM, Apple, HP-UX ³⁷ | Dle produktu (GNU GPL i placená) | Integer, Int, TynyInt, Set, BigInt, MediumInt, SlallInt, Decimal, Bit, Numeric, Float, Real, Double, Date, Boolean, Time, Datetime, Year, Timestamp, | Ano | SQL-2003 | Ano |

³⁶ Features Of SQLite. *SQLite* [online]. SQLite, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.sqlite.org/features.html>

³⁷ Supported Platforms: MySQL Database. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.mysql.com/support/supportedplatforms/database.html>

| | Podpora více uživatelů | Systémová/Souborová databáze | Podpora platform | Licence | Podpora datových typů | Podpora serveru | SQL standard | Ověření uživatele |
|-------------------|------------------------|------------------------------|--|---|---|-----------------|--------------|-------------------|
| | | | | | Char, Varchar, Binary, Varbinary, Text, Blob, Enum ³⁸ | | | |
| PostgreSQL | Ano | systémová | Linux, Unix, Solaris, HP-UX, MacOS, AIX, BSD, Microsoft windows, ³⁹ | PostgreSQL Licence - opensource licence podobná BSD | SmallInt, Integer, BigInt, Decimal, Numeric, Real, Double Precision, Serial, Bigserial, Character, Text, Bytea, Timestamp, Interval, Date, Time, XML, JSON, Boolean ⁴⁰ | Ano | SQL-2008 | Ano |
| Oracle | Ano | systémová | Microsoft win, Linux, Solaris, HP-UX, IBM, AIX, MACOS X ⁴¹ | Dle produktu (Free i placená) | Char, Varchar, Nchar, Nvarchar2, Long, Number, Binary_float, Binary_double, Bfile, Date Timestamp, Blob, Clob, Nclob | Ano | SQL-2011 | Ano |

³⁸ Data Types. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

³⁹ About. *PostgreSQL* [online]. The PostgreSQL Global Development Group, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.postgresql.org/about/>

⁴⁰ Data Types. *PostgreSQL* [online]. The PostgreSQL Global Development Group, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.postgresql.org/docs/9.1/static/datatype.html>

⁴¹ Oracle Database Online Documentation. *Oracle Helper center* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: https://docs.oracle.com/database/121/nav/portal_11.htm

| | Podpora více uživatelů | Systémová/Souborová databáze | Podpora platform | Licence | Podpora datových typů | Podpora serveru | SQL standard | Ověření uživatele |
|-----------------------------|------------------------|------------------------------|--|-------------------------------|--|-----------------|--------------|-------------------|
| Microsoft SQL Server | Ano | systémová | Podpora pouze pro produkty Microsoft ⁴² | Dle produktu (Free i placená) | BigInt, Bit, Decimal, Int, Money, Date, Numeric, SmallInt, TinyInt, SmallMoney, Float, Real, Time, Datetime, Datetime2, Datetimeoffset, Smalldatetime, Char, Text, Nvarchar, Varchar, Nchar, Ntext, Binary, Varbinary, Image, Cursor, Timestamp, Hierarchyid, XML, Table ⁴³ | Ano | SQL-2011 | Ano |

⁴² Hardware and Software Requirements for Installing SQL Server 2012. *Microsoft Developer Network* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ms143506\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/ms143506(v=sql.110).aspx)

⁴³ Data Types (Transact-SQL). *Microsoft Developer Network* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: <https://msdn.microsoft.com/library/ms187752.aspx>

Tabulka porovnání databázových systémů se zaměřuje na srovnání SQLite s MySQL, PostgreSQL, Oracle a Microsoft SQL Server. Porovnáme zde vůči sobě podporu uživatelů, platforem, datových typů, ověření uživatele, podporovaný SQL standard, licence, pod kterými jsou databáze uvolňovány a zda se jedná o souborovou nebo systémovou databázi. U SQLite je v datových typech uvedena také její vlastnost Typeless, neboť se jedná o záměrně naprogramovanou vlastnost této databáze. Z tabulky je také možné zjistit, že SQLite jednoznačně podporuje nejvíce platforem a určitou výhodou je i licence Public Domain.

3 VOLNĚ DOSTUPNÉ PROGRAMOVÉ NÁSTROJE

3.1 SQLITE ADMINISTRATOR

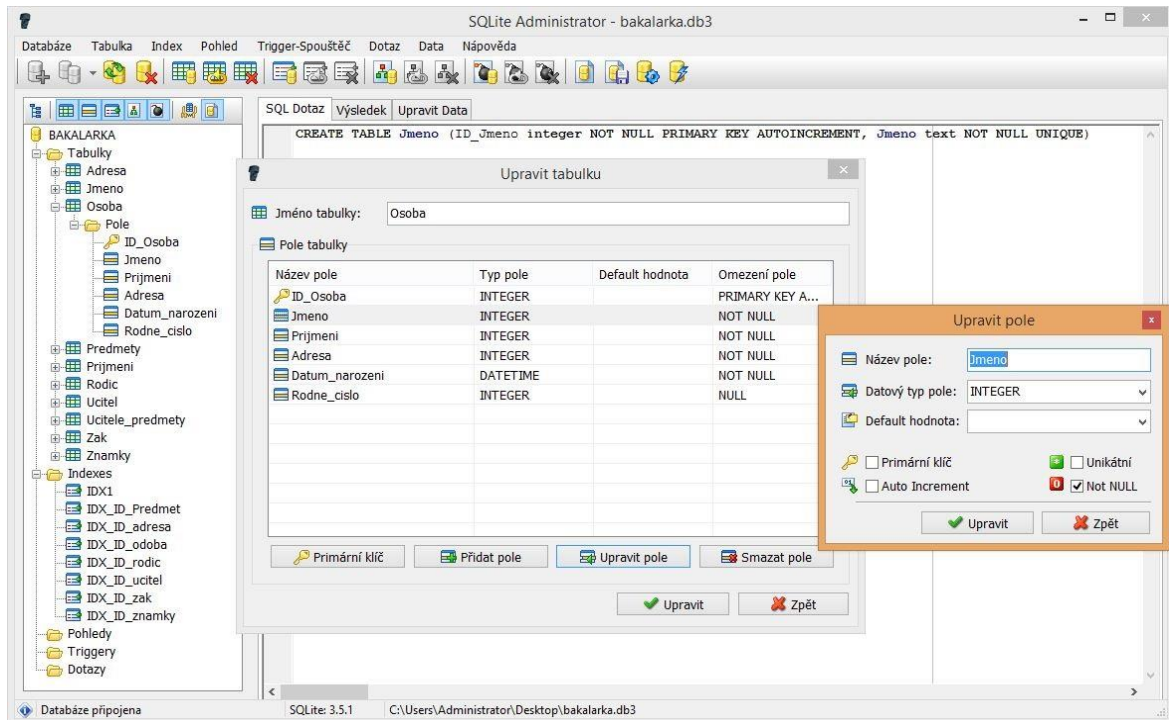
SQLite Administrator⁴⁴ je freeware dostupný pro osobní, nekomerční využití. Software je podporován pouze na platformách Windows 2000, Windows XP a Windows Vista, přesto v případě testování programu v rámci této práce nebyly zaznamenány problémy pod Windows 7 a Windows 8.1. SQLite Administrator nabízí podporu 17 jazyků včetně jazyka českého, i když je otázka, zda se tato podpora může považovat přímo za výhodu programu.

V tomto programu se velice snadno vytvářejí, upravují nebo odstraňují tabulky použitím záložky Tabulka v hlavní nabídce. Stejně snadná je i editace polí v tabulce, kde je možné jednoduše zvolit datový typ, defaultní hodnotu, primární klíč, automatické přidání čísla v pořadí (Auto Increment), nastavení nenulové hodnoty (Not Null) a jedinečnosti. Se stejnou jednoduchostí umí program vytvořit Index, Trigger a SQL dotaz, ve kterém můžeme pomocí příkazů SQL libovolně editovat databázi. Data je možné do programu importovat z formátu CSV a exportovat do formátů XLS, CSV, XML a HTML.

Nevýhodou tohoto programu je, že ukládá textová data jen v kódování ISO-8859-1, ale SQLite3 podporuje i kódování UTF-8 a UTF-16. Pokud tedy textová data budeme vkládat v tomto softwaru, nemusí být vždy v pořádku.

Na závěr je třeba upozornit, že tento nástroj na editaci SQLite databáze byl naposled aktualizován v roce 2008 ve verzi 0.8.3.2 Beta a nejedná se tudíž o aktuální software, i když je zcela použitelný.

⁴⁴ <http://sqliteadmin.orbmu2k.de>



Obrázek 3.1 Grafické prostředí SQLite Administrator

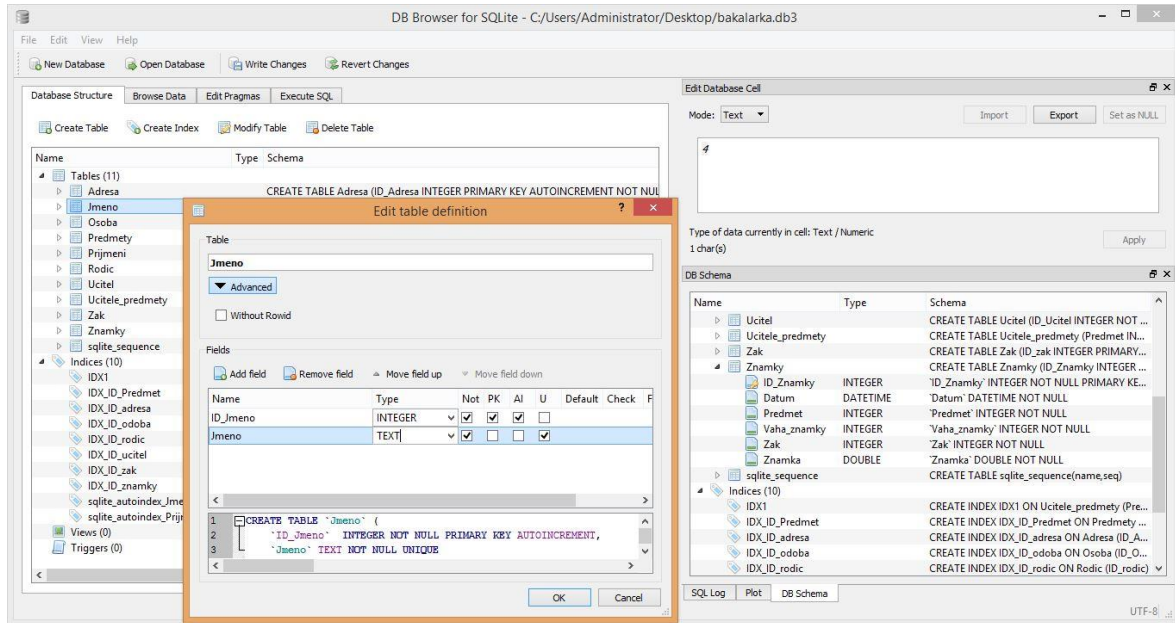
Zdroj: Vlastní

3.2 DB BROWSER FOR SQLITE

DB Browser for SQLite je vázán licenci Mozilla Public License Version 2 a zároveň druhou licenci GNU General Public License Version 3 nebo vyšší. Vzhledem k těmto licencím je možné DB Browser instalovat na platformách Windows 32bit i 64bit s možností instalace či portable verzí, MAC OS X. Samozřejmě jsou zkompileované instalační balíčky na různé 32bit i 64bit distribuce Linuxu. DB Browser disponuje podporou 12-ti jazyků bez možnosti volby jazyka českého.

Vytváření a editace tabulek v tomto jednoduchém nástroji je velice snadné a rychlé, neboť jsou u základního menu již předdefinována tlačítka pro tyto operace. Příjemná pro práci s databází jsou i tlačítka pro zápis změny Write Changes a zrušení změny Revert Changes v případě nechtěného zápisu do databáze. Program podporuje import ze souborů TXT, SQL, CSV a export do souborů CSV a SQL. V nastavení je možné zvolit kódování databáze UTF-8 či UTF-16, zapnutí cizích klíčů, defaultní nastavení datového typu, vlastnosti fontu nebo nastavení barev při psaní příkazů pro přehlednější orientaci. V grafickém rozhraní je možné jednoduše vytvořit či smazat indexy, ale triggery je nutné editovat přes záložku Execute

SQL. Záložka Execute SQL má možnost otevřít několik oken pro vykonání příkazů SQL, které se mohou hodit při testování databáze v případě, že nechceme stále přepisovat již jednou napsané příkazy. Tento software je plně dostačující k tvorbě a editaci SQLite3 databáze a značnou výhodou je častá aktualizace a dobrá podpora.



Obrázek 3.2 Uživatelské prostředí DB Browseru for SQLite

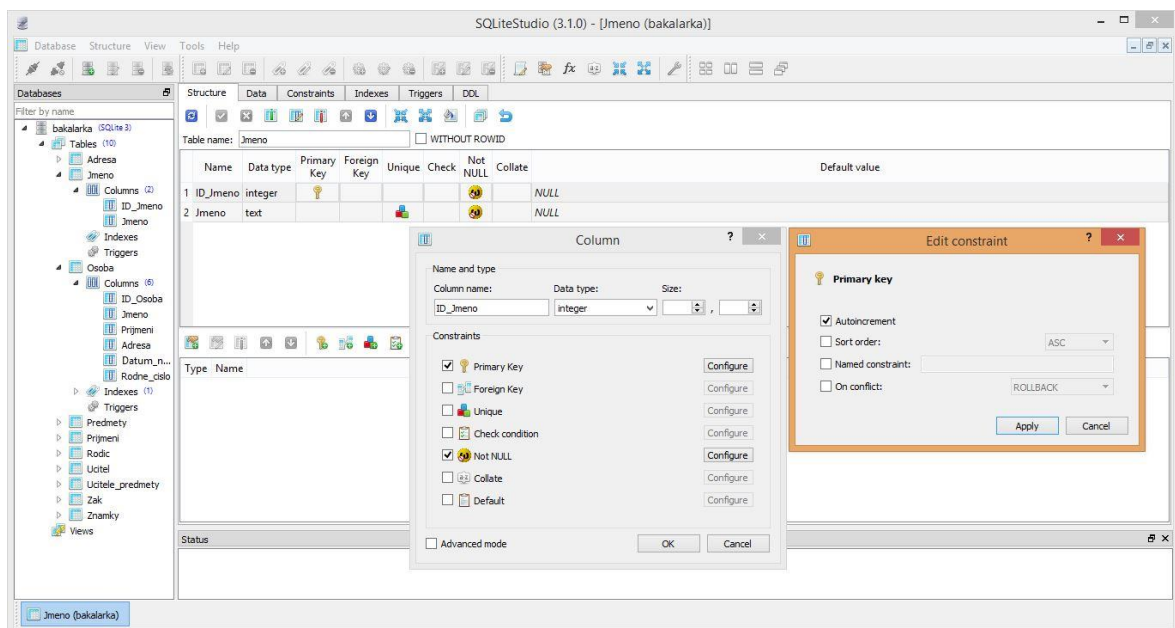
Zdroj: Vlastní

3.3 SQLITE STUDIO

SQLite Studio je uživatelsky velice přívětivý program, který vyvíjí polský programátor Pawel Salawa a je vázaný licencí GNU General Public License Version 3 a vyšší.

Program umožňuje nastavení v šesti různých jazycích, ale z nám příbuzných jazyků je možná pouze volba slovenštiny. Pro editaci databáze můžeme využít volby v hlavním menu nebo databázovou lištu pro rychlou volbu. V hlavním menu nabízí záložka Database hlavní funkce pro práci s databází a jejími daty. První položka Connect to the database slouží k připojení databázového souboru a následující položka disconnect to the database k odpojení od souboru. Zajímavá je položka Add a database, která umožňuje otevřít najednou více databázových souborů SQLite2 nebo SQLite3 a pracovat tak s více databázovými soubory najednou. Záložka Structure nám nabídne možnost vytvořit, editovat či smazat tabulku, index, trigger a view. Záložka View zpříjemní nastavení pracovní plochy pro lepší přehled v databázi. Jsou zde možnosti rozložení více otevřených oken, uspořádání a práci s nimi.

Můžeme zavřít aktuální okno, všechna okna nebo okna, která nejsou aktuální, nebo je můžeme přejmenovat pro snadnější přehled. V záložce Tools nalezneme nástroje na editaci SQL, editor a historii změn ze všech databázových souborů, se kterými jsme do té doby pracovali. Data můžeme importovat ze souborů CSV nebo textových souborů s využitím regulárních výrazů a možností libovolného kódování. Pro export databáze máme možnost využít několik formátů jako HTML, PDF, JSON, SQL a XML s volbou formátu kódování. Dialogové okno nastavení programu můžeme otevřít ze záložky Tools anebo klávesou F2. Program si zdarma můžeme stáhnout na stránkách autora <http://sqlitestudio.pl>.



Obrázek 3.3 Uživatelské prostředí SQLiteStudio

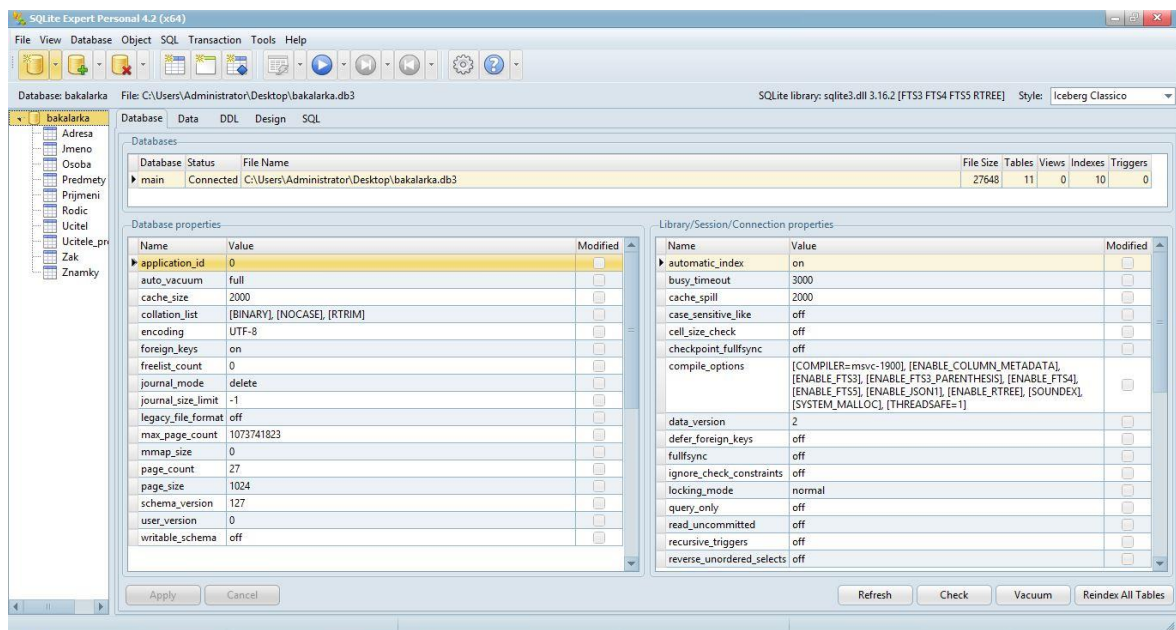
Zdroj: Vlastní

3.4 SQLITE EXPERT PERSONAL

SQLite Expert Personal⁴⁵ je freewarový software pro správu SQLite databází, který vychází z placené verze SQLite Expert Professional a jsou zde tedy některá programová omezení. Z těchto programových omezení považuji za nejzávažnější omezení importu a exportu dat z různých datových souborů. Jedinou možností, jak nainportovat data, je připojení dalšího databázového souboru a zkopírování dat mezi databázemi. Program pracuje se soubory db, db3 a sqlite. S tímto programovým nástrojem je možné databázi vytvořit a editovat

⁴⁵ <http://www.sqliteexpert.com/download.html>

včetně view, trigger, cizích klíčů, indexů, vacuum, tabulek a uživatelských dat. Samozřejmostí je možnost přímého vkládání SQL příkazů a jejich úprava pomocí databáze. Příjemné je zobrazení všech informací o připojené databázi na úvodní stránce v záložce „Database“. Program samotný disponuje velkým množstvím grafických vzhledů. Výhodou programu jsou také velmi časté aktualizace, za které program vděčí placené verzi „Professional“. Program je k dispozici pouze na platformách Windows 2000, XP, Vista, 7, 8, 10. Stahovat program můžeme z oficiálních stránek výrobce, a to jak ve free verzi „Personal“, tak i v trial verzi „Professional“ s možností vyzkoušení na 30 dní zdarma.



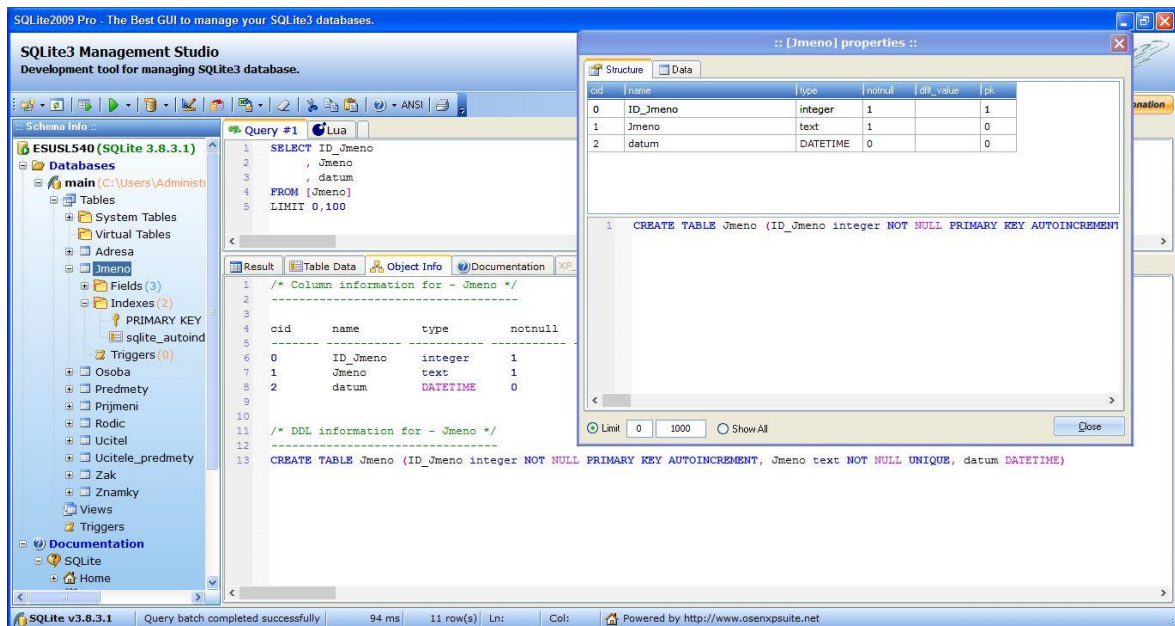
Obrázek 3.4 Uživatelské prostředí SQLite Expert Personal

Zdroj: Vlastní

3.5 SQLITE2009PRO

SQLite2009Pro je program, který napsal programátor Osen Kusnadi a jeho poslední aktualizaci provedl v únoru 2014. Program je uživatelsky velice přívětivý a veškeré důležité funkce jsou k dispozici k rychlému použití v hlavním panelu. Program zvládá v grafické podobě veškeré základní úkony s databází. Databázi je zde možné vytvořit, připojit další databázový soubor, uložit s šifrováním 128bit AES či RSA, vacuum a zkontrolovat integritu databáze. Importovat data do programu lze z databázových souborů MS Access, MS SQL Server, MySQL, PostgreSQL. Export lze provést přímo do sešitu MS Excel, nebo můžeme

data uložit jako CSV, XML a HTML. Velice jednoduše a přehledně můžeme v programu přímo z hlavní nabídky spravovat indexy, vytvořit view, trigger a tabulku. SQLite2009Pro také poskytuje jako jeden z mála freewarových programů Visual Query Builder pro grafické zobrazení a vytvoření SQL dotazů. Zajímavým způsobem vyřešil programátor nápovědu, kdy různé problematiky SQLite jsou řešeny online odkazy na oficiální stránky SQLite knihovny, a proto máme přístup ke stále aktualizované nápovědě tak, jak probíhá její vývoj.



Obrázek 3.5 Uživatelské prostředí SQLite2009Pro

Zdroj: Vlastní

3.6 OSTATNÍ SOFTWARE

Díky oblíbenosti SQLite databáze můžeme nalézt velké množství freewarových programů pro její editaci. Vybral jsem z nich programy, které začátečníkovi poskytnou nejjednodušší možnost seznámení s vytvořením vlastní databáze. Mezi ně můžeme dále ještě zařadit SQLiteSpy, který je volně dostupný pro nekomerční použití a nedisponuje sofistikovaným grafickým rozhraním nebo SQLite manager volně dostupný jako doplněk internetového prohlížeče Mozilla Firefox. Pokud někdo preferuje platformu Linux a nevybere si žádný z výše představených programů, které jsou pro Linux k dispozici, může si nainstalovat kancelářský balík Calligra, jehož součástí je open source databázový nástroj Kexi. Jednoduchý, ale velmi mocný nástroj je Adminer, který je napsaný v PHP a je k dispozici zdarma pod Apache nebo GPL2 licencí. Tento nástroj plně postačuje na administraci nejen

SQLite, ale i MySQL a mnoho dalších databází, jejichž seznam nalezneme na oficiálních stránkách www.adminer.org. Pokud chceme databázi mít stále u sebe a upravovat ji, můžeme využít některou z aplikací pro zařízení, která podporují operační systém Android. Na Google play je nesčetné množství placených i freewarových aplikací, z nichž je v současné době nejlépe hodnocený SQLiteManager.

Tabulka 2 Grafické uživatelské rozhraní aplikací

Zdroj: Vlastní

| GUI | SQLite Administrator | DB Browser for SQLite | SQLite Studio | SQLite Expert Personal | SQLite2009 Pro | SQLiteSpy |
|-------------------|---|-----------------------|---------------------------|------------------------|---|-----------|
| Vytvoření DB | *** | *** | **** | **** | ** | ** |
| Vytvoření Tabulky | *** | *** | **** | **** | *** | *** |
| Vkládání dat | ** | *** | *** | ** | *** | ** |
| Editace dat | ** | *** | *** | *** | *** | * |
| Index | Ano | Ano | Ano | Ano | Ano | Ne |
| Trigger | Ano | Ne | Ano | Ne | Ano | Ne |
| Query designer | Ne | Ne | Ano | Ne | Ano | Ne |
| Import data | Soubory obsahující znaky oddělené hodnoty | SQL, CSV | CSV, RegExp | Ne | MS Access, MySQL, PostgreSQL, MS SQL Server | Ne |
| Export data | XLS, CSV, XML, HTML | SQL, CSV | HTML, JSON, PDF, SQL, XML | Ne | Excel, CSV, XML, HTML | Ne |

| GUI | SQLite Administrator | DB Browser for SQLite | SQLite Studio | SQLite Expert Personal | SQLite2009 Pro | SQLiteSpy |
|----------------------|----------------------|-----------------------|---------------|------------------------|----------------|-----------|
| View | Ano | Ano | Ano | Ano | Ano | Ano |
| změna kódování | Ne | Ano | Ne | Ne | Ano | Ano |
| podpora cizích klíčů | Ne | Ano | Ano | Ano | Ano | Ne |
| Vacuum | Ne | Ano | Ano | Ano | Ano | Ano |
| Execute SQL | Ano | Ano | Ano | Ano | Ano | Ano |
| SQL Log | Ne | Ano | Ne | Ano | Ne | Ne |
| Poslední Aktualizace | 2008 | 2016 | 2016 | 2017 | 2014 | 2005 |

Legenda:

***** - Příjemná a uživatelsky snadná editace

*** - Rychle dostupná, ale složitější editace

** - Editace je hůře dostupná

* - Editace je možná, ale velmi špatně se obsluhuje

Ano - Funkce je součástí aplikace

Ne - Funkce není součástí aplikace

Tabulka přehledně zobrazuje možnosti grafických uživatelských rozhraní vybraných programových nástrojů pro editaci SQLite databáze. Všechny zvolené programové nástroje jsou primárně určeny pro vytváření a editaci SQLite databáze, ale každý nástroj může být jinak uživatelsky přívětivý, a proto jsem zvolil čtyři možnosti označené počtem hvězdiček, které jsou vysvětleny v legendě tabulky pro rozlišení přívětivosti aplikace. Tabulka se zabývá srovnáním funkcí dostupných v grafickém rozhraní. To ale neznamená, že pokud je

aplikace v tabulce uvedená jako nepodporující cizí klíče, není vůbec možné s cizími klíči v této aplikaci pracovat. Může zde být například možnost přímého vložení SQL příkazu. Vložena k porovnání byla také položka „Poslední aktualizace“. Nejedná se o funkci k editaci SQLite, ale je důležité vědět, jak je daný nástroj aktuální a případně jak často autoři aplikací reagují na rychle se vyvíjející databázi SQLite.

4 NÁVRH POSTUPU TVORBY SQLITE DATABÁZE

Správně navrhnout databázi znamená dodržet určitá pravidla, aby databáze poskytovala rychle správné a přesné výsledky. Je také nutné si uvědomit, že práce na databázi nekončí jejím vytvořením a uvedením do ostrého provozu, neboť databáze se dá přirovnat k dítěti. Čím více databáze roste, tím více se o databázi musíme starat, aby byla stále rychlá a dobře použitelná. Také je nutné si vysvětlit základní pojmy pro práci s databází.

4.1.1 VYSVĚTLENÍ POJMŮ

Entita – Zbyněk Bureš popisuje entitu jako „*objekt reálného světa, který je schopen nezávislé existence a je jednoznačně identifikovatelný*⁴⁶“. Jedná se tedy o objekt, který nám v databázi představuje tabulka. V naší vzorové databázi se jedná o tabulky class, student, course a course_type.

Atribut – popisuje vlastnost entity. Představuje ji sloupec tabulky.

Záznam – jedná se o řádku v tabulce, která představuje jeden objekt.

Primární klíč – je jednoznačný identifikátor záznamu v tabulce. Nikdy nesmí nabývat hodnoty NULL a musí být jedinečný. Většinou tento atribut označujeme ID.

Cizí klíč – je atribut, kterému relací přidělíme primární klíč z jiné tabulky.

Autoincrement je vlastnost, která nám zajistí automaticky jedinečnou hodnotu ke každému nově přidanému záznamu. Zpravidla ji přiřazujeme k primárnímu klíči.

Index – vytvoří si pomocný soubor, který pomáhá rychleji vyhledávat data v tabulce. Jedna tabulka může mít více indexů, a pokud je v tabulce primární klíč, jedná se vždy o index.

4.1.2 RELACE

Relace nám popisují vztahy mezi entitami. Kardinalitu vztahů dělíme do tří skupin. První relace 1:1 je používána v případě, že jeden záznam v tabulce odpovídá pouze jednomu záznamu v druhé tabulce. V naší vzorové databázi je relace 1:1 reprezentována entitami Person-parent, Person-teacher a Person-student neboť Osoba může být vždy jen učitel, rodič nebo žák. Do této relace patří také vztahy 0:1 a 1:0.

⁴⁶ BUREŠ, Zbyněk. *Databázové systémy I: studijní opora*. Jihlava: Vysoká škola polytechnická Jihlava, 2014. ISBN 978-80-87035-88-7.

Relaci 1:N používáme, když jeden záznam v první tabulce odpovídá více záznamům v druhé tabulce. Například naše databáze má tuto relaci vyjádřenou vztahem Student a Grade, kdy student může mít více známek z různých předmětů, ale konkrétní známka z předmětu je přiřazena pouze jednomu určitému studentovi. Relace 1:N obsahuje také vztahy 0:1, 1:0 a 1:1.

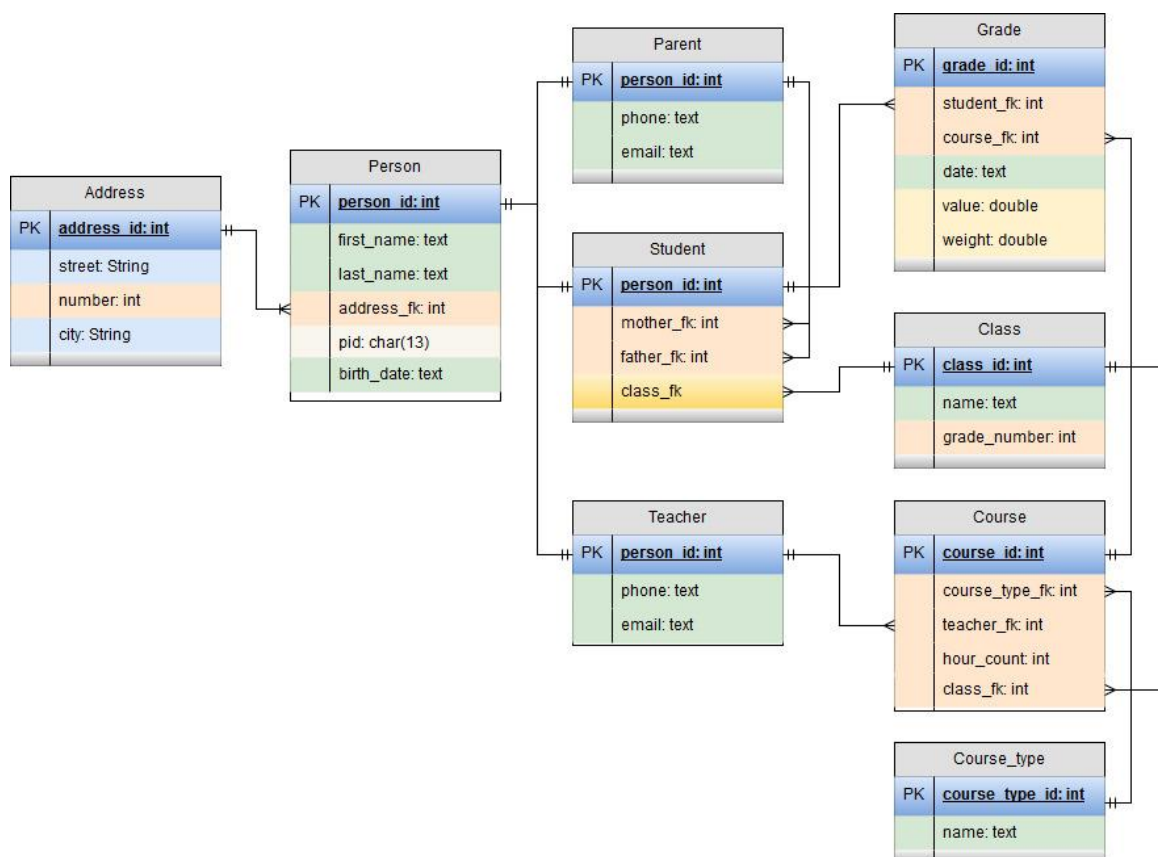
Relace M:N používáme, pokud záznamům v první tabulce odpovídá více záznamů v druhé tabulce a obráceně - záznamy z druhé tabulky odpovídají více záznamům z první tabulky. Pro použití této relace ji však musíme rozdělit na dva vztahy 1:N (dekompozice)⁴⁷. Relační databáze relaci M:N nepodporuje, a pokud tuto relaci chceme použít, je nutné vytvořit pomocnou tabulku. V naší vzorové databázi tato relace není použita.

4.1.3 NÁVRH DATABÁZE

Nejdříve si vytvoříme logický model databáze, kdy se zatím budeme pohybovat na čistě teoretické rovině, ale z tohoto modelu bude zřejmé, jaké entity, atributy a relace budeme fyzicky realizovat v naší databázi. Následující obrázek nám představí ER Diagram (Entity Relationship Diagram) naší databáze vytvořený aplikací draw.io⁴⁸, podle kterého vytvoříme fyzický model databáze ve vybrané freewarové aplikaci

⁴⁷ BUREŠ, Zbyněk. *Databázové systémy I: studijní opora*. Jihlava: Vysoká škola polytechnická Jihlava, 2014. ISBN 978-80-87035-88-7.

⁴⁸ <https://www.draw.io>



Obrázek 4.1 ER Diagram vzorové databáze

Zdroj: Vlastní

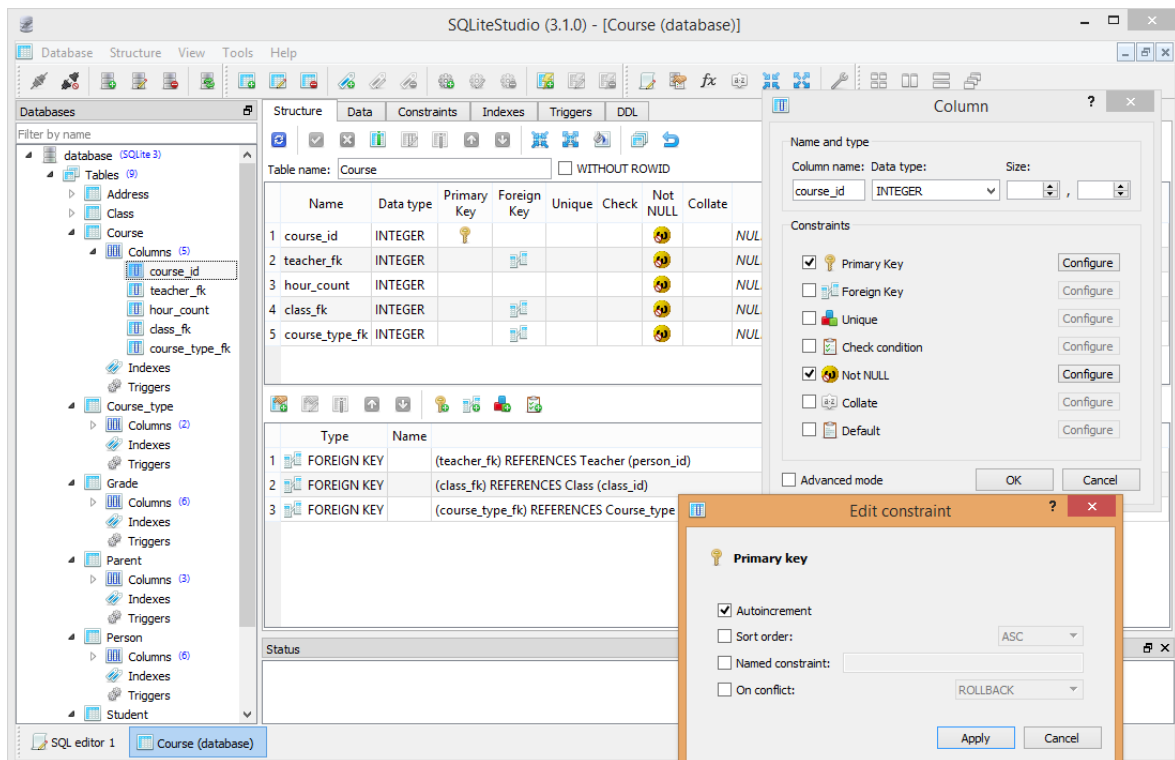
Pro vytvoření databáze využijeme nám všem známé školní prostředí. Databáze nebude příliš složitá, ale měla by nám poskytnout ucelený pohled na práci s databází a ukázat možnosti využití knihovny SQLite. Databáze bude obsahovat devět tabulek a jejím cílem bude vést údaje k žákům ze strany vyučujícího. Budeme tedy potřebovat tabulky učitel (Teacher), žák (Student), u kterého musíme znát jeho rodiče (Parent). Tabulka osoba (Person) bude obsahovat všechny zadané osoby pro tabulky Parent, Student a Teacher a bude k nim vedena tabulka adresa bydliště (Address). Dále bude nutné vytvořit tabulky obsluhující předměty, třídy a známky a naplnit tabulky atributy s relacemi a záznamy.

4.2 VYTVOŘENÍ DATABÁZE

Velice dobře a jednoduše se používá programový prostředek SQLite Studio, a proto právě v něm ukážeme tvorbu databáze. Po spuštění aplikace a stisknutí ikony „Add a database“ budeme vyzváni k výběru datového souboru. Na výběr máme k dispozici „SQLite 2“, „SQLite 3“ a „SQL Cipher“ pro možnost šifrování databáze. Po zvolení názvu a následném potvrzení vytvoříme databázi, ke které je ještě nutné se přes ikonu „Connect to the

database“ připojit. Entitu vytvoříme ikonou „Create a table“, ikonou v rychlém panelu nebo záložce „Structure“ a následně jí přiřadíme atributy, bez kterých entitu nelze uložit. První vytvoříme entitu „address“ a přiřadíme jí atributy. Po stisku ikony „Add column“ zvolíme v nově zobrazeném okně název atributu, v našem případě „address_id“ a zvolíme datový typ „Integer“. Zkratka „id“ za podtržítkem znamená jednoznačný identifikátor pro každý řádek, a proto je nutné zde zvolit volbu „Not Null“, aby například nedošlo k nezadání důležité položky pro funkci databáze, dále „Primary Key“ a následně v jeho konfiguračním nastavení zaškrtnout volbu „Autoincrement“ pro automatické přiřazování čísel k řádkům. Stejným způsobem vytvoříme atributy „street“, „city“, „number“ a uložíme přes ikonu „Commit structure changes“. Příjemné v této aplikaci je zobrazení kontrolního okna s SQL příkazy před každým uložením entity pro kontrolu zadaných údajů s možností volby zobrazení při každé operaci s entitou.

Jako druhou entitu vytvoříme „Person“. Zde si ukážeme vytvoření cizího klíče. Pro vytvoření entity a atributu postupujeme stejně jako v předešlém kroku, pouze místo volby „Primary Key“ zvolíme „Foreign Key“ a v jeho konfiguraci zvolíme v „Foreign Table“ entitu „person“ a v „Foreign column“ zvolíme „person_id“. Takto postupně doplníme všechny entity a atributy v naší databázi a můžeme ji naplnit daty. Vlastní data přidáme dvojitým kliknutím na atribut a v záložce data, kde máme možnost přidávat jeden záznam nebo přes volbu „Insert multiple rows“ zvolit počet vložení více záznamů. Kompletní vytvořenou databázi se zobrazením všech náležitostí databáze podle ER Diagramu můžeme vidět na následujícím obrázku.



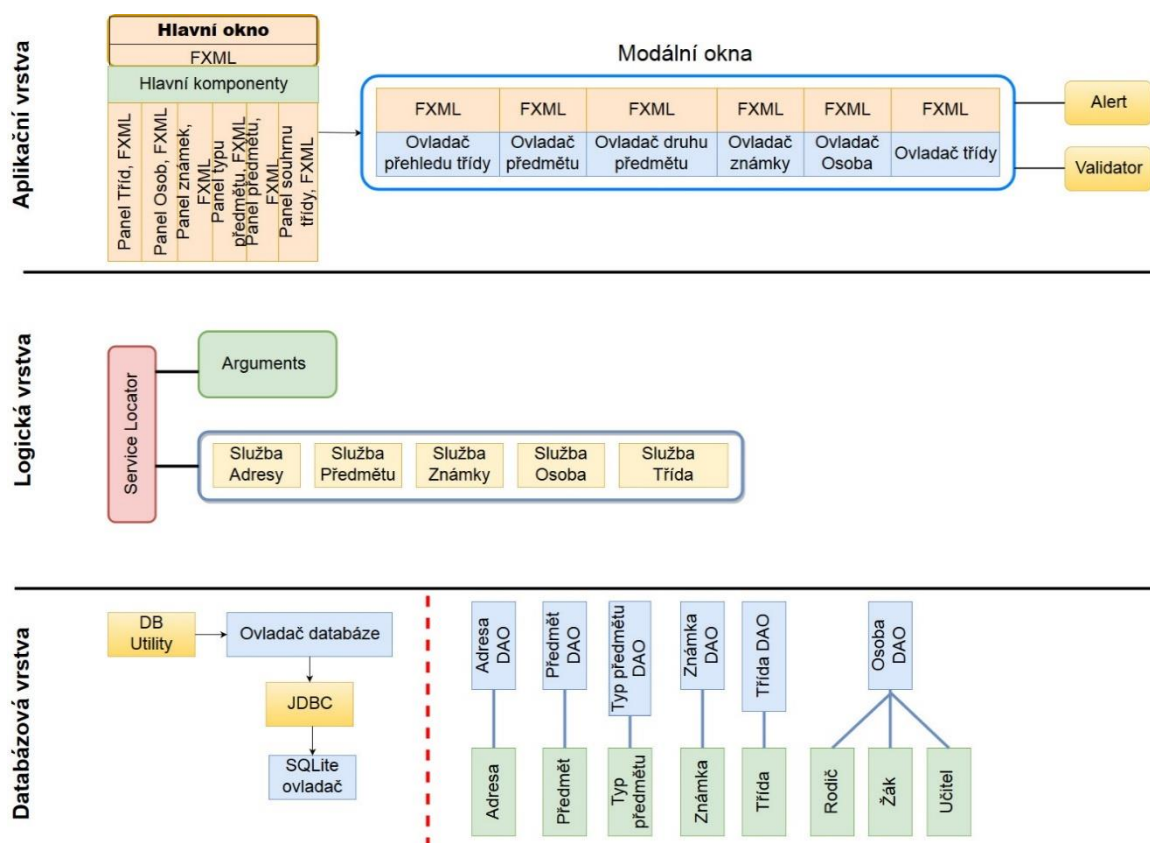
Obrázek 4.2 Fyzický model databáze

Zdroj: Vlastní

5 TVORBA APLIKACE NAD SQLITE DATABÁZÍ

5.1 NÁVRH APLIKACE

Aplikace je naprogramována jako desktopová s třívrstvou architekturou. Vrstvy jsou rozdělené na databázovou, logickou a aplikační. Databázová vrstva se připojuje k souboru s databází a zajišťuje CRUD nad jejími daty. Tyto funkce dále poskytuje jako služby logické vrstvě. Logická vrstva s využitím databázové vrstvy obstarává logické operace a poskytuje služby vrstvě aplikační. Aplikační vrstva zobrazuje data v desktopovém rozhraní pomocí technologie JavaFX a směřuje akce vyvolané uživatelem do logické vrstvy. Grafické rozdělení na jednotlivé vrstvy nám ukazuje následující obrázek.



Obrázek 5.1 Třívrstvá architektura aplikace Evidence

Zdroj: Vlastní

Tato třívrstvá dekompozice umožňuje vytvářet jednotlivé funkční bloky (v našem případě třídy), které mají tyto vlastnosti:

Jednoduchost - bloky jsou jednoduché a jejich funkce je přímočará. Usnadňuje to údržbu a rozšiřitelnost kódu.

Zapouzdřenost - bloky fungují jako blackboxy. Ten, kdo využívá jejich služeb, zná rozhraní čili funkcionalitu, ale nemusí znát její vnitřní logiku.

Nahraditelnost - pokud jsou dobře zvolena rozhraní služeb, lze jejich implementace měnit, například změnou databázového stroje z SQLite na MySQL, přičemž zbytek programu zůstane zachován.

Aplikace je vytvořena open source vývojovým prostředím *Eclipse*⁴⁹ v programovacím jazyku *Java*. Dále je použit *JavaFX Scene Builder* pro vytvoření grafického rozhraní a *Apache Maven* ke kompilaci celého projektu. Vzhledem k tomu, že je aplikace kompilována do souboru *aplikace.jar*, můžeme ji spouštět i na platformách, na kterých je možné instalovat JRE 1.8 a vyšší.

Vytvořená aplikace musí být uživatelsky přívětivá a systémem výjimek a upozornění informovat uživatele o případných chybách při vkládání vstupních dat u telefonu, emailu či rodného čísla. Z programátorského hlediska musíme rozlišovat závislost entit mezi sebou. Pokud například smažu jednu osobu a adresa bude vedena ještě i u druhé osoby, nesmí být adresa smazána. Z ER diagramu je vidět, že tedy můžeme upravovat všechny tabulky kromě adresy. Stejná logika musí fungovat i u učitele, neboť pokud učitel odejde ze školy a budu ho mazat, musí aplikace vyzvat k jeho odstranění ze všech předmětů, ke kterým je veden. Stejně tak pokud budeme chtít smazat třídu, ve které jsou žáci, nebo předmět, ke kterému jsou přiřazeny kurzy a známky, musí nás aplikace upozornit, zda chceme vymazat i údaje spojené s těmito tabulkami. Nesmíme ani zapomenout, že aplikace musí splňovat základní operace označované akronymem *CRUD*, což jsou počáteční písmena *create, read, update a delete*. Pro databáze platí používání příkazů *Insert, Select, Update a Delete*.

⁴⁹ www.eclipse.org

5.2 JAVA FX

JavaFX je aplikační programové rozhraní napsané v jazyku Java. Jedná se o moderní prostředí pro tvorbu okenního grafického rozhraní, které postupně nahrazuje svého předchůdce *SWING*. Hlavní výhodou je přizpůsobení vzhledu pomocí CSS nebo - jako v našem případě - oddělení logické vrstvy od aplikační, použitím souborů návrhu formulářů *FXML*. Soubory *FXML* vycházejí ze struktury jazyka *XML* a slouží pouze k informacím o rozložení komponent a o jejich vlastnostech, ale jsou odděleny od logiky aplikace.

5.3 JDBC

Vzhledem k tomu, že aplikace bude přistupovat k datům přes JDBC, je nutné si říci několik vět o tomto rozhraní. Java Database Connectivity častěji označováno jako JDBC, je programové rozhraní pro přístup k databázím. Pro každý databázový systém vývojáři vytváří vlastní JDBC, které pracuje jako mezičlánek mezi aplikací a databází, jako překladač mezi instancemi tříd a SQL dotazy. Jeho největší výhodou je přenositelnost aplikace, neboť pokud změníme JDBC-SQLite v aplikaci na JDBC-MySQL, můžeme spouštět naši aplikaci na jiném databázovém systému bez nutnosti úpravy aplikace a zásahu do SQL příkazů. Mezi hlavní třídy pro práci s databází patří třída *Statement*, jejíž instance zasílají příkazy SQL přímo databázi, kde se provedou a vrátí výsledek. Využívat budeme také třídu *Prepared Statement*. Jedná se o předpřipravené příkazy, kdy místo proměnné dosadíme otazníky, které jsou uloženy odděleně a poté vloženy do databáze. Pokud je *Prepared Statement* použit správně, jedná se o jednu z ochranných proti *SQL Injection*.

5.4 APACHE MAVEN

*Apache Maven*⁵⁰ je nástroj napsaný v Javě a slouží k sestavení projektu ze zdrojového kódu do spustitelné podoby. Pro zkompileování musí být vytvořen soubor *Project Object Model* s názvem *pom.xml*, který musíme také vytvořit. V souboru jsou některé údaje povinné jako například sekce *project*, verze Mavenu, GroupID, ArtifactID a použité závislosti. Sekce *properties* obsahuje použité kódování a použitou verzi Javy, v *dependency* je uvedené JDBC rozhraní a knihovny JCommander, včetně jejich verzí. Důležitá je sekce *Include*, kde říkáme, aby při kompilaci byly zahrnuty i všechny soubory *xfml*, protože právě v těchto souborech je grafické rozhraní aplikace. Poslední sekce *DescriptorRef* nám příkazem *jar-with-*

⁵⁰ <https://maven.apache.org/>

dependencies přibalí do zkompilevaného souboru všechny využívané knihovny a není potřeba zjišťovat, jaké knihovny aplikace využívá. Výsledný soubor `pom.xml` nalezneme v příloze č. 1. Když máme vytvořený `pom.xml` a celý projekt, musíme vytvořit skript, který aplikaci zkompileje. Skripty je obecně jednodušší vytvářet v linuxovém prostředí. Pokud někdo používá pouze systémy Windows, může použít aplikaci *Cygwin*⁵¹. *Cygwin* je linuxové prostředí spuštěné pod systémem Windows, které umožňuje spustit linux `.sh` i windows `.bat` skripty. Vytvoříme skript `buil.sh`, ve kterém nejdříve smažeme a následně vytvoříme nový adresář `comlpet`, spustíme instalaci Mavenu a výsledný soubor přejmenujeme na `bp-esner.jar`, zkopírujeme inicializační soubory ze složky `static` a nastavíme práva pro spouštěcí skripty. Skript vidíme v příloze č. 2.

5.5 SOUBORY APLIKACE

Základní soubory aplikace jsou *App.java*, *Arguments.java* a *ServiceLocator.java*. *App.java* obsahuje hlavní metodu `main` a spuštění grafického rozhraní. *Arguments.java* obsahuje argumenty pro spuštění aplikace z příkazového řádku pro vyčištění, vytvoření a načtení inicializačních dat databáze. Pro tuto funkci je použita knihovna *Jcommander*⁵², která umí namapovat pole stringových argumentů z příkazové řádky na anotovanou třídu. *ServiceLocator.java* poskytuje jednotlivé služby jiným objektům, například načítá již zmíněné argumenty z příkazové řádky. Pro kontrolu vstupních dat od uživatele vytvoříme soubor *ValidatorUtils.java*, ve kterém pomocí regulárních výrazů kontrolujeme správně zadané rodné číslo, email, telefonní číslo a vyplnění povinných údajů data narození s rodným číslem. *AlertUtil.java* je pouze úprava vyskakovacího okna, které vyskočí při zachycení výjimky způsobené uživatelem, neboť metoda `validate` sama zachytává výjimky a nedovolí uživateli zadávat dále data, dokud nejsou zadána správně. *DatabaseService.java*, která vytváří připojení k databázi, inicializuje tabulky, načtení souboru a vzhledem k tomu, že se připojujeme k jednomu databázovému souboru, jedná se o jedinou instanci. Další součástí databázové vrstvy jsou doménové objekty, na které se pomocí *PreparedStatement* mapují data z databáze pro vložení, update, vyhledání a vkládání dat. Mapování těchto objektů nám poskytují soubory *Data Access Object*, reprezentované soubory s názvem entity a zkratkou DAO, pro adresu například soubor `AddressDAO.java`, které zajistí přímý

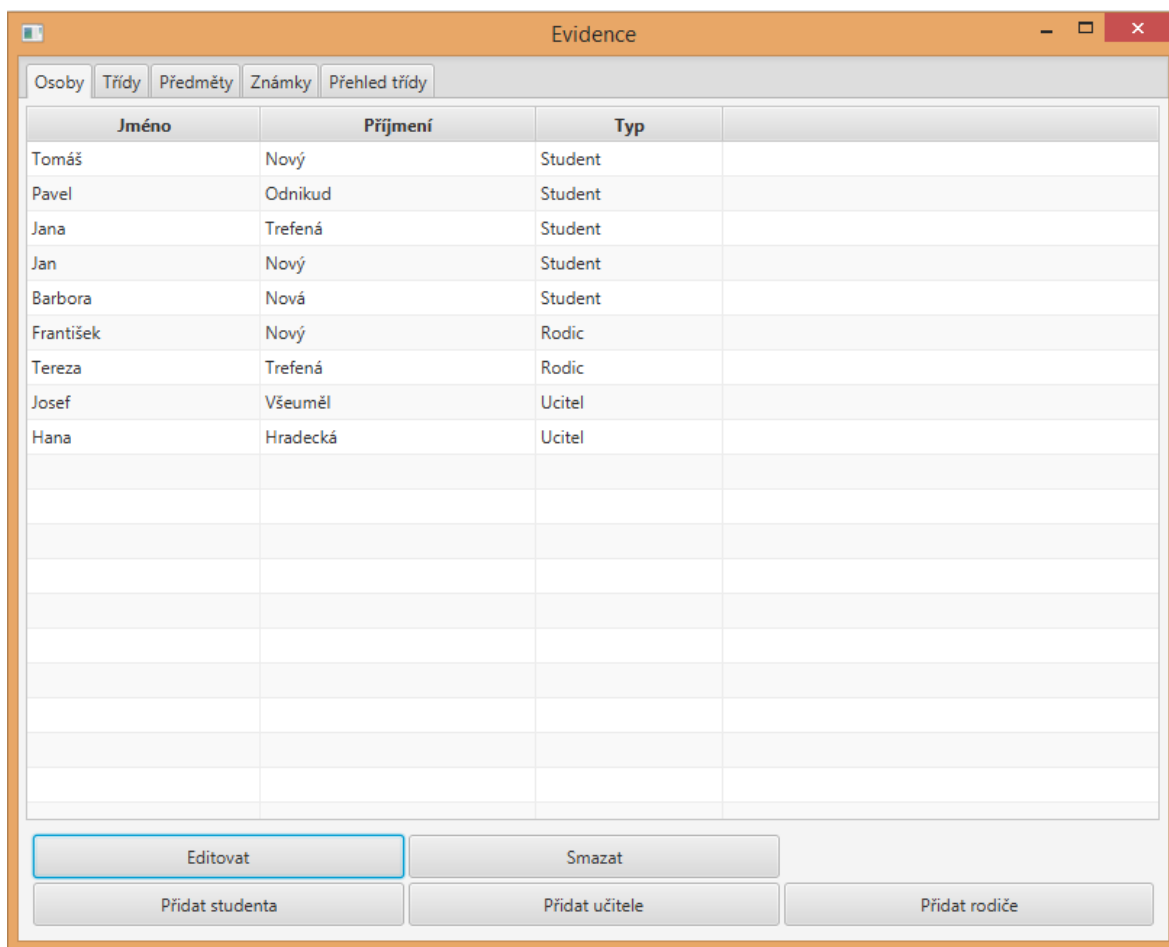
⁵¹ <http://www.cygwin.com/>

⁵² <http://jcommander.org/>

kontakt s databází pro daný objekt jednoduchým příkazem. Nalezneme zde také jeden důležitý soubor s názvem `DBUtils.java`. Vzhledem k tomu, že SQLite knihovna nepodporuje datový typ `Date`, je nutné zajistit správné uložení data do databáze. O to se nám stará právě soubor `DBUtils.java` uložením data do datového typu `string`. V logické vrstvě nalezneme servisní soubory, které se požívají mezi sebou a poskytují služby aplikační vrstvě. Tyto soubory již nejsou rozdělené podle entit, ale podle logických celků, a proto například `CourseService.java` již obsluhuje `Course` i `CourseType`. Pokud se vytváří nová databáze nebo není databázový soubor nalezen při spuštění aplikace, využívá se soubor `db-schema-init.sql`, který zajišťuje vytvoření potřebných tabulek - viz příloha č.3.

5.6 APLIKACE NAD DATABÁZÍ

Hlavní okno aplikace se skládá z šesti hlavních panelů. Panel `Osoby` nám přehledně zobrazuje jména všech osob zavedených v databázi a rozdělených podle typu na studenty, rodiče a učitele. V panelu `třídy` jsou zobrazené učebny vedené na dané škole s přiřazením ročníku, kterému náleží. Panel `Předměty` je jediný panel v aplikaci, který se skládá ze dvou panelů. První panel zobrazuje všechny předměty a druhý panel zobrazuje typ předmětu, jméno vyučujícího předmětu a hodinovou dotaci předmětu. Panel `Známky` slouží k zobrazení přidělených známek studentům. Je zde uvedené jméno studenta, známkový předmět, datum kdy byla známka udělena a hodnota s váhou známky. Poslední panel `Přehled třídy` soustřeďuje informace o jednotlivých učebnách, kdy po výběru učebny získáme počet žáků vedených v učebně a u každého jména žáka vážený průměr známek z jednotlivých předmětů. Následující obrázek nám ukazuje grafické rozhraní hlavního okna aplikace `Evidence`.

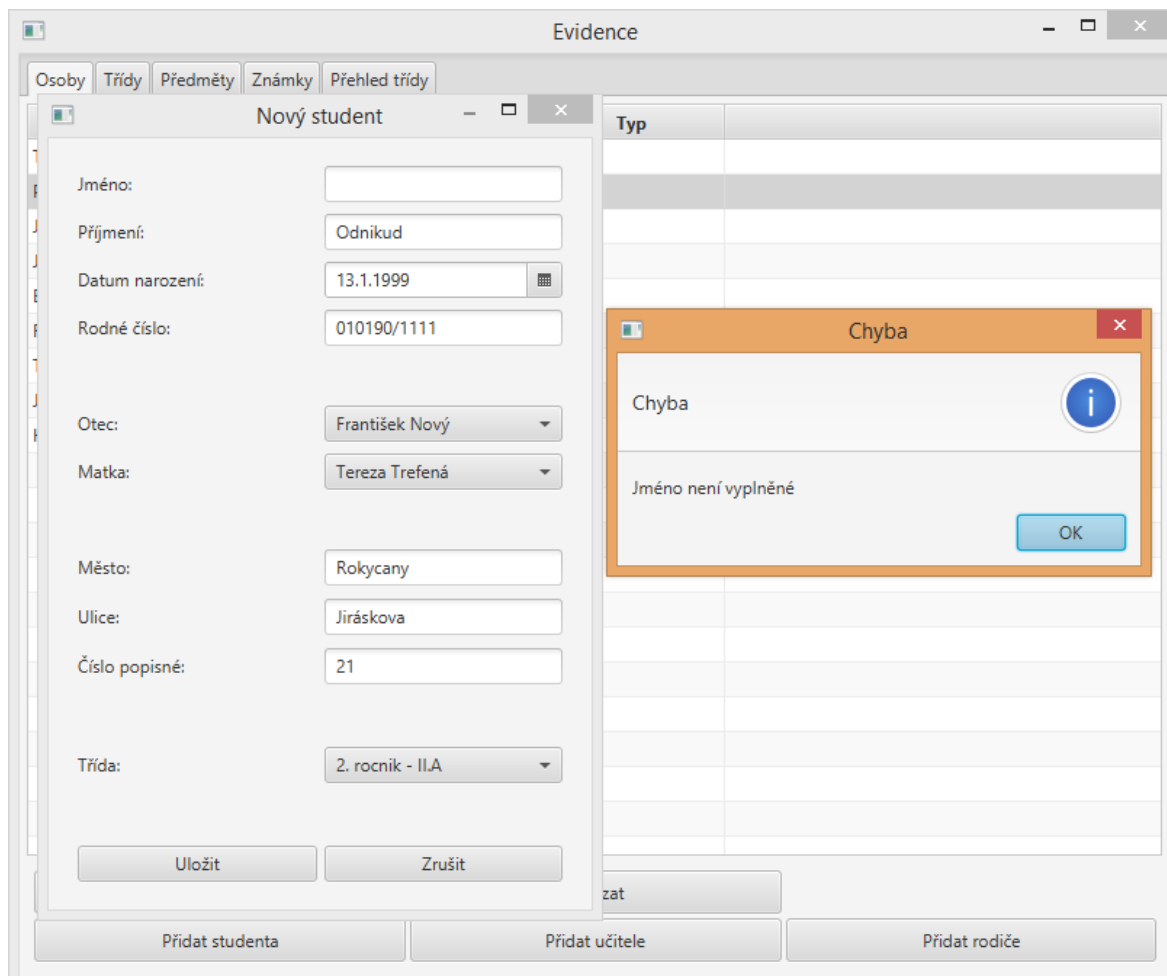


Obrázek 5.2 GUI aplikace Evidence

Zdroj: Vlastní

Všechny panely vyjma panelu *Přehled třídy* obsahují editační tlačítka pro editaci dat v databázi, která při stisknutí vyvolají modální okna. Panel *osoby* obsahuje editační tlačítka pro přidání studenta, učitele, rodiče a úpravu či smazání libovolné osoby z databáze. Modální okna pro vložení nového žáka obsahují pole *Jméno*, *Příjmení*, *Rodné číslo*, *Město*, *Ulice* a *Číslo popisné* pro vložení textu, volby pro výběr rodičů *matky*, *otce* a volby *třída*, do které jsou žáci zařazeni. Datum narození je možné zvolit výběrem z kalendáře nebo zapsáním textu. Modální okno je možné ukončit bez editace nebo uložit. Při každém uložení do databáze aplikace zkontroluje vyplnění povinných údajů a pomocí vhodných regulárních výrazů správnost zapsaných položek emailu a telefonního a rodného čísla. V případě nalezení nevyplněného pole povinných informací nebo chybně zadaného formátu je uživatel informován dialogovým oknem

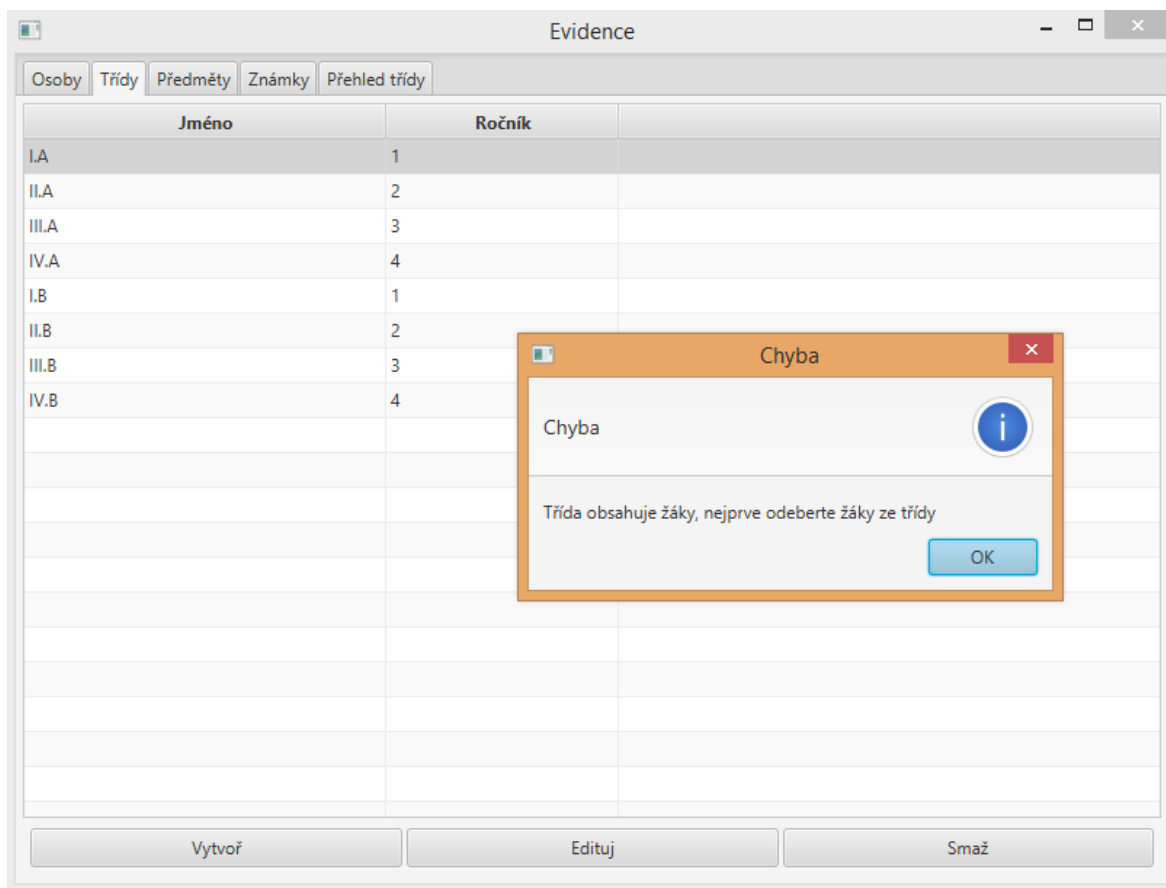
s popisem zachycené výjimky. Zobrazená chyba při pokusu o uložení údajů o studentovi bez zadaného jména je ukázána na následujícím obrázku.



Obrázek 5.3 Zobrazení chyby aplikace

Zdroj: Vlastní

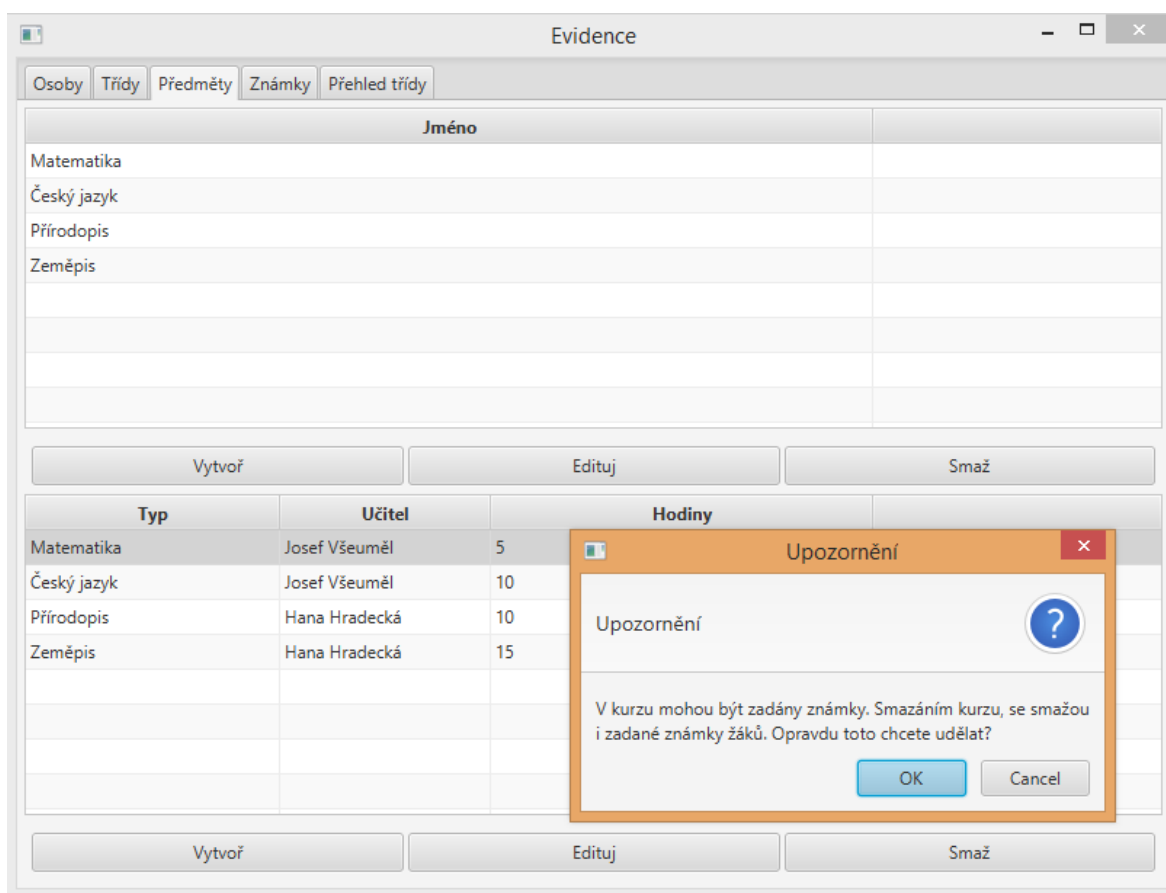
Modální okno pro vložení učitele a rodiče jsou shodná a obsahují mimo vložení Jména, Příjmení, Data narození, Rodného čísla a Adresy navíc Email a Telefon. Kontrola správnosti údajů je provedena shodně jako v modálním okně pro přidání žáka, neboť pro validaci dat je používán v celé aplikaci shodný validátor. Modální okno v panelu *Třídy* umožňuje vytvořit učebnu, přidělit jí ročník, provádět změny v již vytvořené učebně a mazat učebny. Logika aplikace nám však nedovolí smazat učebnu, ve které jsou přiřazeni žáci, a o této chybě jsme opět informováni dialogovým oknem - viz následující obrázek.



Obrázek 5.4 Chyba odstranění třídy obsahující žáky

Zdroj: Vlastní

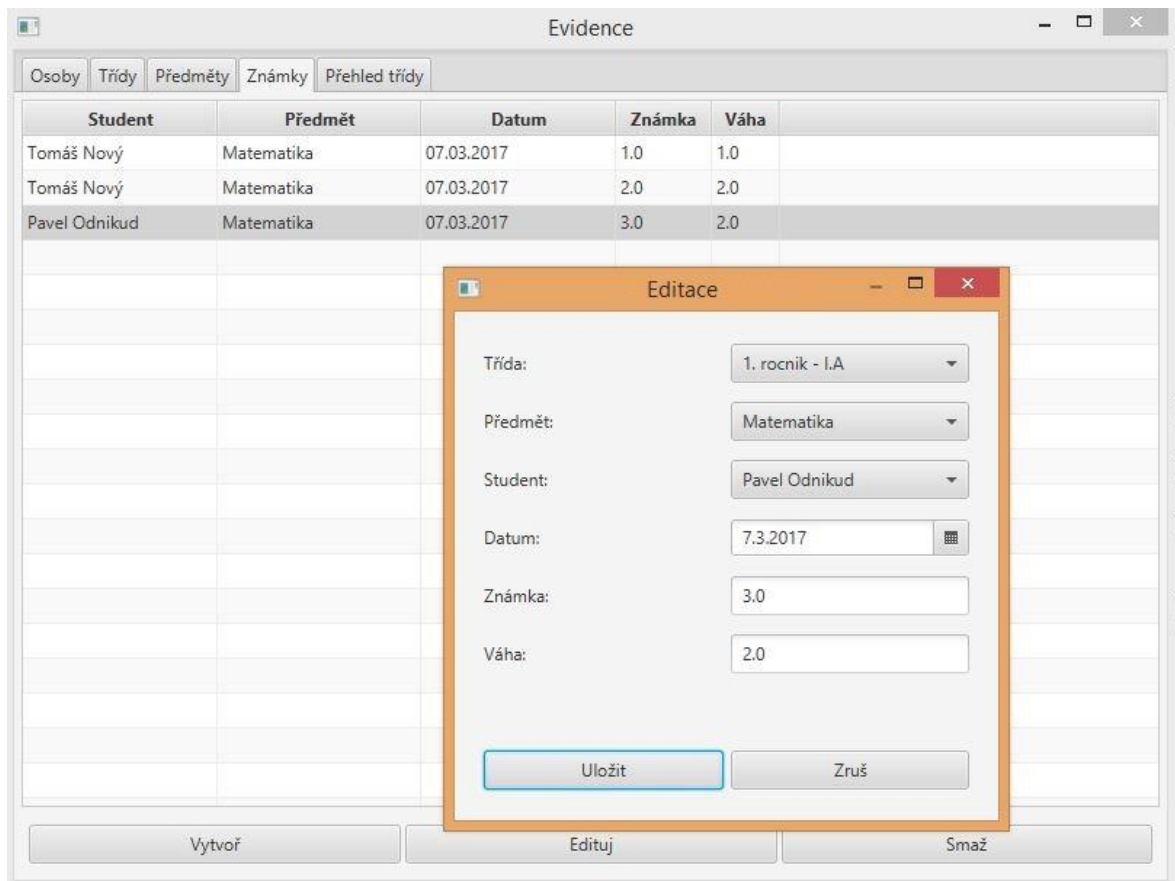
Panel **Předměty** obsahuje dva vlastní panely pro správu předmětů. Horní část je přidělena vytvoření, editaci a mazání jednotlivých předmětů s kontrolou proti smazání předmětu, jenž obsahuje navázané kurzy k předmětu. Spodní část je přidělena vytvoření, editaci a mazání kurzů. Rozdělení na předměty a kurzy je záměrné, neboť pokud máme například předmět Matematika, je jisté, že pod stejným názvem, ale s jinou hodinovou dotací, jiným vyučujícím a hlavně náplní předmětu bude předmět vyučován v různých ročnících. Při vytvoření kurzu jsme proto požádání o vložení typu předmětu, učitele, třídy, ve které je vyučován a hodinovou dotací pro daný kurz. Pokud chceme kurz odstranit a kurz stále obsahuje známky přidělené žákům, jsme požádáni dialogovým oknem o potvrzení smazání kurzu i přesto, že dojde k odstranění zadaných známek žákům u daného kurzu - viz následující obrázek.



Obrázek 5.5 Chyba odstranění předmětu

Zdroj: Vlastní

Poslední panel, který obsahuje modální okno, jsou *Známky*. Známky lze mazat pouze výběrem studenta a předmětu, ke kterému je známka přidělena bez nutnosti další kontroly napojení na jinou tabulku. Pro přidělení známky studentovi je nutné po stisknutí tlačítka *Vytvoř* v modálním okně zadat třídu, ve které se student nachází, předmět, ze kterého byla známka udělena, jméno studenta, datum udělení známky, hodnotou a váhu známky. Editaci známky vidíme na následujícím obrázku.



Obrázek 5.6 Editace známky

Zdroj: Vlastní

Panel Přehled třídy slouží pouze jako informativní okno o jednotlivých studentech, počtu studentů ve třídě a přehledu jejich průměrných známek z jednotlivých předmětů. Průměrná známka je vypočítána z hodnoty a váhy známky – viz následující obrázek.

| Jméno žáka | Matematika | Český jazyk |
|--------------|------------|-------------|
| Tomáš Nový | 2.2 | 0.0 |
| Jana Trefená | 0.0 | 1.0 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Obrázek 5.7 Přehled známek třídy

Zdroj: Vlastní

Aplikace může po rozšíření sloužit jako evidence základním a středním školám, ale zcela jistě nebude nahrazovat aplikaci Bakaláři, která je vyvíjena již řadu let firmou BAKLÁŘI software s.r.o. Vzorová data v aplikaci jsou smyšlená, a slouží pouze pro ukázkou funkčnosti této aplikace. Veškerá podobnost s osobou v reálném světě je čistě náhodná.

ZÁVĚR

Tato práce byla zaměřena na návrh, tvorbu a editaci SQLite databáze. Hlavním cílem bakalářské práce byla specifikace vlastností knihovny SQLite a poskytnutí dostatek informací pro volbu vhodného databázového systému a vytvoření vlastního projektu.

V teoretické části byly představeny hlavní výhody systému SQLite - jednoduchost, flexibilita, spolehlivost, množství datových typů, ale také omezení, kterými SQLite disponuje. Byly zde vymezeny pojmy atomic, consistent, isolated a durable. Na závěr první kapitoly byly představeny možnosti jejího využití a shrnuty hlavní výhody a nevýhody.

Druhá kapitola se zabývala srovnáním SQLite s aktuálně nejpoužívanějšími databázovými systémy. Výsledkem bylo zobrazení jejich hlavních vlastností v přehledné tabulce. Třetí kapitola představuje pět desktopových freewareových aplikací pro tvorbu a editaci SQLite databáze včetně seznámení s dalšími možnostmi editace na jiných platformách.

Čtvrtá kapitola se nejprve zabývala vysvětlením pojmů entita, atribut, záznam, primární a cizí klíč, autoincrement, index a relační vztah, neboť jejich znalost je nutná pro tvorbu databáze. Na základě těchto znalostí popsala vytvoření fyzického modelu databáze programovým prostředkem SQLite Studio podle navrhnutého ER diagramu.

Cílem praktické části popsané v páté kapitole bylo vytvořit aplikaci editující vytvořenou databázi a ukázat optimální přístup k datům z grafického uživatelského rozhraní. Proto byla aplikace naprogramována s třívrstvou architekturou, která optimální přístup k datům zajišťuje. Aplikace je rozdělena na aplikační, logickou a databázovou vrstvu a složení jednotlivých částí vrstev je graficky zobrazeno v návrhu aplikace. Byly zde popsány použité nástroje JavaFX, JDBC, Apache Maven a soubory, které aplikace ke správné funkci využívá. Závěr praktické části se zabýval popisem a funkcemi vytvořené aplikace.

Všechny možnosti SQLite v této práci zcela jistě nejsou zahrnuty, ale byly zde představeny ty vlastnosti, které pomohou získat ucelený přehled při rozhodování o použití SQLite v praxi. Osobně si myslím, že vývojáři SQLite vytvořili specifický a unikátní databázový systém, kterému jen těžko může v této kategorii konkurovat jiný databázový systém.

Z provedeného detailního popisu vyplývá, že databázi SQLite je vhodné používat spíše v zařízeních či aplikacích s jedním přístupem, kde nám dostačuje malá databáze kvůli omezení hardwarových prostředků, ale požadujeme rychlé odezvy a podporu jazyka SQL.

Není však vhodná do projektů, kde by byl požadován přístup k datům v databázi přes síťové rozhraní, a to kvůli zamykání databázového souboru při editaci. V tomto případě bude rozhodně lepší vybrat jiný databázový systém, například ze systémů porovnávaných v této práci.

V modelovém příkladu popsaném v praktické části je nemožnost přístupu k databázi přes síťové rozhraní zásadní – SQLite databázi je vhodné použít pouze za předpokladu, že s ní v jeden okamžik bude pracovat pouze jedna osoba. Nasazení SQLite bych tedy doporučil pouze v menších školních zařízeních nebo např. jako soukromou databázi s hodnocením žáků pro jednoho učitele. Výhodou by pak byla jednoduchost aplikace, rychlost odezvy, minimální pořizovací náklady, snadná údržba a umístění na lokálním počítači bez nutnosti provozu serverové části.

RESUMÉ

Tato bakalářská práce se zabývá databázovým systémem SQLite. V teoretické části jsou definovány vlastnosti SQLite a termíny související s tvorbou databáze, provedeno srovnání s aktuálně používanými databázovými systémy a freewarovými programovými nástroji pro editaci SQLite. Praktická část popisuje použité programové nástroje k vytvoření aplikace s třívrstvou architekturou a postup sestavení vlastní aplikace. Cílem této práce je poskytnout uživateli databázového systému SQLite celkový pohled na tento systém a umožnit mu rozhodnout se o vhodnosti nasazení SQLite databáze na jeho projekt.

SUMMARY

This bachelor's thesis is focused on database's system SQLite. In the theoretical part of my thesis SQLite's characters and terms related to database's creation are defined, a comparison of SQLite with actual used database's system and freeware program tools for SQLite's editing is drawn. The practical part describes using program tools to create an application with three-tier architecture and methods of the application's creation. The aim of this thesis is to provide a complex view of this system to potential users and enable them to decide about a suitability of using SQLite's database for their projects.

Klíčová slova

SQLite, databáze, databázový systém, databázová aplikace, Oracle, Microsoft SQL Server, MySQL, PostgreSQL.

KEY WORDS

SQLite, database, database system, database application, Oracle, Microsoft SQL Server, MySQL, PostgreSQL.

SEZNAM LITERATURY

GRANT ALLEN AND MIKE OWENS. *The definitive guide to SQLite: [take control of this compact but powerful tool to embed sophisticated SQL databases within your applications!]*. 2nd. ed. New York: Apress, 2010. ISBN 9781430232254.

KREIBICH, Jay a. *Using SQLite*. Beijing: O'Reilly, c2010. ISBN 978-0-596-52118-9.

BUREŠ, Zbyněk. *Databázové systémy 1: studijní opora*. Jihlava: Vysoká škola polytechnická Jihlava, 2014. ISBN 978-80-87035-88-7

ELEKTRONICKÉ ZDROJE

Our Businesses. *General Dynamics* [online]. Dalaware: General Dynamics, 2016 [cit. 2017-03-16]. Dostupné z: <http://www.generaldynamics.com/our-businesses>

SQLite Overview. *Tutorials Point* [online]. Madhapur, 2017 [cit. 2017-03-16]. Dostupné z: https://www.tutorialspoint.com/sqlite/sqlite_overview.htm

Features Of SQLite. *SQLite* [online]. SQLite, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.sqlite.org/features.html>

Limits In SQLite. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-16]. Dostupné z: <https://www.sqlite.org/limits.html>

An Introduction To The SQLite C/C++ Interface. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-16]. Dostupné z: <https://www.sqlite.org/cintro.html>

Wiki Index. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-16]. Dostupné z: www.sqlite.org/cvstrac/wiki?p=SqliteWrappers~~dobj

SQLite. *Datatypes In SQLite Version 3* [online]. The SQLite Consortium, 2017 [cit. 2017-04-01]. Dostupné z: <https://sqlite.org/datatype3.html>

SQLite. *PRAGMA Statements* [online]. The SQLite Consortium, 2017 [cit. 2017-04-01].

Dostupné z: <https://www.sqlite.org/pragma.html>

SQLite. *How SQLite is tested* [online]. The SQLite Consortium, 2017 [cit. 2017-04-01].

Dostupné z: <https://sqlite.org/testing.html>

Filesystem Limits. *Purdue University* [online]. West Lafayette: Purdue University, 2017 [cit. 2017-03-20]. Dostupné z:

<https://engineering.purdue.edu/ECN/Support/KB/Docs/FilesystemLimits>

Ext4: evoluční souborový systém. *ROOT.CZ* [online]. ROOT.CZ: root.cz, 2008 [cit. 2017-03-20]. Dostupné z: <https://www.root.cz/clanky/ext4-evolucni-souborovy-system/>

SQL As Understood By SQLite. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-20]. Dostupné z: https://www.sqlite.org/lang_select.html

Datatypes In SQLite Version3. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-20]. Dostupné z: <https://sqlite.org/datatype3.html>

Release History. *SQLite* [online]. The SQLite Consortium, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.sqlite.org/changes.html>

SQLite Transaction. *SQLite* [online]. SQLite Tutorial, 2017 [cit. 2017-03-20]. Dostupné z: <http://www.sqlitetutorial.net/sqlite-transaction/>

Oracle licensing. *Oracle FAQ's* [online]. Oracle FAQ's, 2017 [cit. 2017-03-20]. Dostupné z: http://www.oraFAQ.com/wiki/Oracle_licensing

Physical Database Limits. *Oracle Help Center* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <http://docs.oracle.com/database/121/REFRN/GUID-939CB455-783E-458A-A2E8-81172B990FE9.htm#REFRN0042>

Max size of datafiles. *Community Oracle* [online]. 2007 [cit. 2017-04-02]. Dostupné z: <https://community.oracle.com/thread/521373>

Oracle Data Types. *Oracle Help Center* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm

Ceny systému SQL Server. *SQL Server 2016* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2016-pricing>

Porovnání verzí systému Microsoft SQL Server. *SQL Server 2016* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.microsoft.com/cs-cz/sql-server/sql-server-2016-comparison>

Editions and Supported Features for SQL Server 2016. *SQL Server 2016* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: <https://msdn.microsoft.com/en-us/library/cc645993.aspx>

Hardware and Software Requirements for Installing SQL Server 2012. *SQL Server 2016* [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ms143506\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/ms143506(v=sql.110).aspx)

Instalace MS SQL Server 2014 Express. *SQL Server 2016* [online]. Jihlava: STORMWARE Software Development, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.stormware.cz/podpora/faq/sql-server.aspx>

What is MySQL? *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

MySQL Editions. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.mysql.com/products/>

The Main Features of MySQL. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/features.html>

Data Types. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

Limits on Number of Databases and Tables. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/database-count-limit.html>

Limits on Table Size. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/table-size-limit.html>

Limits on Table Column Count and Row Size. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20]. Dostupné z: <https://dev.mysql.com/doc/refman/8.0/en/column-count-limit.html>

PostgreSQL. *PostgreSQL* [online]. Global Development Group, 2017 [cit. 2017-03-20]. Dostupné z: <http://postgres.cz/wiki/PostgreSQL>

PostgreSQL. *PostgreSQL* [online]. Global Development Group, 2017 [cit. 2017-03-20]. Dostupné z: <https://www.postgresql.org/docs/9.1/static/datatype.html>

PostgreSQL. *PostgreSQL* [online]. Global Development Group, 2017 [cit. 2017-03-20].

Dostupné z:

http://postgres.cz/wiki/Reference_pou%C5%BEit%C3%AD_PostgreSQL_v_%C4%8CR

Features Of SQLite. *SQLite* [online]. SQLite, 2017 [cit. 2017-03-20]. Dostupné z:

<https://www.sqlite.org/features.html>

Supported Platforms: MySQL Database. *MySQL* [online]. ORACLE, 2017 [cit. 2017-03-20].

Dostupné z: <https://www.mysql.com/support/supportedplatforms/database.html>

About. *PostgreSQL* [online]. The PostgreSQL Global Development Group, 2017 [cit. 2017-

03-20]. Dostupné z: <https://www.postgresql.org/about/>

Data Types. *PostgreSQL* [online]. The PostgreSQL Global Development Group, 2017 [cit.

2017-03-20]. Dostupné z: <https://www.postgresql.org/docs/9.1/static/datatype.html>

Oracle Database Online Documentation. *Oracle Helper center* [online]. ORACLE, 2017 [cit.

2017-03-20]. Dostupné z: https://docs.oracle.com/database/121/nav/portal_11.htm

Hardware and Software Requirements for Installing SQL Server 2012. *Microsoft Developer*

Network [online]. Microsoft, 2017 [cit. 2017-03-20]. Dostupné z:

[https://msdn.microsoft.com/en-us/library/ms143506\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/library/ms143506(v=sql.110).aspx)

Data Types (Transact-SQL). *Microsoft Developer Network* [online]. Microsoft, 2017 [cit.

2017-03-20]. Dostupné z: <https://msdn.microsoft.com/library/ms187752.aspx>

SEZNAM OBRÁZKŮ

| | |
|--|----|
| Obrázek 2.1 Graf oblíbenosti databázových systémů | 16 |
| Obrázek 3.1 Grafické prostředí SQLite Administrator | 27 |
| Obrázek 3.2 Uživatelské prostředí DB Browseru for SQLite | 28 |
| Obrázek 3.3 Uživatelské prostředí SQLiteStudio | 29 |
| Obrázek 3.4 Uživatelské prostředí SQLite Expert Personal..... | 30 |
| Obrázek 3.5 Uživatelské prostředí SQLite2009Pro | 31 |
| Obrázek 4.1 ER Diagram vzorové databáze..... | 37 |
| Obrázek 4.2 Fyzický model databáze..... | 39 |
| Obrázek 5.1 Třívrstvá architektura aplikace Evidence..... | 40 |
| Obrázek 5.2 GUI aplikace Evidence | 45 |
| Obrázek 5.3 Zobrazení chyby aplikace | 46 |
| Obrázek 5.4 Chyba odstranění třídy obsahující žáky | 47 |
| Obrázek 5.5 Chyba odstranění předmětu | 48 |
| Obrázek 5.6 Editace známky | 49 |
| Obrázek 5.7 Přehled známek třídy | 50 |

SEZNAM TABULEK

| | |
|--|----|
| Tabulka 1 Porovnání databázových systémů..... | 22 |
| Tabulka 2 Grafické uživatelské rozhraní aplikací | 32 |

SEZNAM PŘÍLOH

Příloha č. 1 – pom.xml

Příloha č. 2 – build.sh

Příloha č. 3 – db-schema-init.sql

Příloha č. 4 – Obsah CD

PŘÍLOHY

PŘÍLOHA Č. 1 – POM.XML

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>cz.esner</groupId>
  <artifactId>bp</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>BP</name>

  <properties>
    <project.build.sourceEncoding>UTF-
8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies><!--
https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc -->
    <dependency>
      <groupId>org.xerial</groupId>
      <artifactId>sqlite-jdbc</artifactId>
      <version>3.16.1</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/com.beust/jcommander-
->
    <dependency>
      <groupId>com.beust</groupId>
      <artifactId>jcommander</artifactId>
      <version>1.60</version>
    </dependency>
  </dependencies>

  <build>
    <resources>
      <resource>
        <filtering>>false</filtering>
        <directory>src/main/java</directory>
        <includes>
          <include>/**/*.fxml</include>
        </includes>
      </resource>
    </resources>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-assembly-plugin</artifactId>
        <version>2.6</version>
        <configuration>
          <descriptorRefs>
            <descriptorRef>jar-with-
dependencies</descriptorRef>
          </descriptorRefs>
          <finalName>bp-esner</finalName>
          <archive>

```

```
        <manifest>
<mainClass>cz.esner.bp.application.App</mainClass>
        </manifest>
    </archive>
</configuration>
<executions>
    <execution>
        <phase>package</phase>
        <goals>
            <goal>single</goal>
        </goals>
    </execution>
</executions>
</plugin>
</plugins>
</build>
</project>
```

PŘÍLOHA Č. 2 – BUILD.SH

```
rm -fr complet/  
mkdir complet  
mvn clean install  
mv target/bp-esner-jar-with-dependencies.jar complet/bp-esner.jar  
cp static/* complet/  
chmod +x complet/start.sh  
chmod +x complet/start-purge.sh
```

PŘÍLOHA Č. 3 – DB-SCHEMA-INIT.SQL

```
PRAGMA foreign_keys = ON;
```

```
CREATE TABLE "Address" (  
    `address_id`      INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    `street`         TEXT NOT NULL,  
    `number`         INTEGER NOT NULL,  
    `city`           TEXT NOT NULL  
);
```

```
CREATE TABLE `Class` (  
    `class_id`      INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    `name`          TEXT NOT NULL,  
    `grade_number`  INTEGER NOT NULL  
);
```

```
CREATE TABLE "Person" (  
    `person_id`     integer NOT NULL PRIMARY KEY AUTOINCREMENT,  
    `first_name`    TEXT NOT NULL,  
    `last_name`     TEXT NOT NULL,  
    `address_fk`    INTEGER,  
    `pid`           char(13) NOT NULL,  
    `birth_date`    TEXT NOT NULL,  
    FOREIGN KEY(address_fk) REFERENCES Address(address_id)  
);
```

```
CREATE TABLE "Student" (  
    `person_id`     INTEGER NOT NULL,  
    `mother_fk`     INTEGER,  
    `father_fk`     INTEGER,
```

```
`class_fk` INTEGER NOT NULL,  
PRIMARY KEY(`person_id`),  
FOREIGN KEY(mother_fk) REFERENCES Person(person_id),  
FOREIGN KEY(father_fk) REFERENCES Person(person_id),  
FOREIGN KEY(class_fk) REFERENCES Class(class_id)  
);  
  
CREATE TABLE `Teacher` (  
    `person_id` INTEGER NOT NULL,  
    `phone` TEXT NOT NULL,  
    `email` TEXT NOT NULL,  
    PRIMARY KEY(`person_id`)  
);  
  
CREATE TABLE "Parent" (  
    `person_id` INTEGER NOT NULL,  
    `phone` TEXT,  
    `email` TEXT,  
    PRIMARY KEY(`person_id`)  
);  
  
CREATE TABLE "Course_type" (  
    `course_type_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    `name` TEXT  
);  
  
CREATE TABLE "Course" (  
    `course_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
    `teacher_fk` INTEGER NOT NULL,  
    `hour_count` INTEGER NOT NULL,  
    `class_fk` INTEGER NOT NULL,  
    `course_type_fk` INTEGER NOT NULL,  
    FOREIGN KEY(teacher_fk) REFERENCES Teacher(person_id),
```

```
FOREIGN KEY(class_fk) REFERENCES Class(class_id),
FOREIGN KEY(course_type_fk) REFERENCES Course_type(course_type_id)
);
CREATE TABLE "Grade" (
    `grade_id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    `course_fk` INTEGER NOT NULL,
    `student_fk` INTEGER NOT NULL,
    `date` TEXT NOT NULL,
    `value` INTEGER NOT NULL,
    `weight` REAL NOT NULL,
    FOREIGN KEY(course_fk) REFERENCES Course(course_id),
    FOREIGN KEY(student_fk) REFERENCES Student(person_id)
)
```


PŘÍLOHA Č. 4 – OBSAH CD

- Složka `COMPLETE` – obsahující spustitelnou verzi aplikace Evidence
- Složka `SOURCE` – obsahující zdrojové kódy aplikace Evidence
- Soubor `BP_Esner.docx` – obsahující bakalářskou práci
- Soubor `BP_Esner.pdf` – obsahující bakalářskou práci