

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Nástroj pro modelování relační databáze

Relational Database Modeling Tool

Zadání bakalářské práce

Student: **Radek Svoboda**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Nástroj pro modelování relační databáze**
Relational Database Modeling Tool

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je vytvořit aplikaci s grafickým uživatelským rozhraním pro návrh schématu relační databáze formou E-R diagramu. Aplikace bude podporovat jak aktualizaci schématu databáze na základě změn v E-R diagramu, tak aktualizaci E-R diagramu podle existujícího schématu. Aplikace bude pracovat se SŘBD Microsoft SQL Server a Oracle Database.

Práce bude splňovat následující body:

1. Popis notace E-R diagramu.
2. Popis a srovnání stávajících CASE nástrojů pro modelování relačních databází.
3. Návrh a implementace grafického editoru E-R diagramů.
4. Návrh a implementace synchronizace E-R diagramů s relačním schématem.

Seznam doporučené odborné literatury:

- [1] CHEN, Peter Pin-Shan. The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 1976, 1.1: 9-36.
- [2] ULLMAN, Jeffrey D.; WIDOM, J. Database Systems: The Complete Book. 2000.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Petr Lukáš**

Datum zadání: 01.09.2016

Datum odevzdání: 28.04.2017



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 28. dubna 2017

.....

Rád bych na tomto místě poděkoval vedoucímu mé bakalářské práce Ing. Petru Lukášovi za poskytnuté rady a čas strávený konzultacemi.

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací softwarového nástroje pro návrh schématu relační databáze formou ER diagramu. Nástroj umožní uživateli nejen tvorbu nových schémat databáze, ale také vizualizaci již existujících schémat a jejich následné modifikace prostřednictvím změn ER diagramu v grafickém editoru. V první části se práce zabývá notací ER diagramu a srovnáním již existujících nástrojů určených k tvorbě ER diagramů. Druhá část se zaměřuje na návrh a implementaci jak grafického editoru, tak na problematiku synchronizace ER diagramu s existujícím relačním schématem.

Klíčová slova: relační databáze, ER diagram, relační schéma, CASE nástroj

Abstract

This thesis describes design and implementation of software tool for relational database schema design in form of an ER diagram. Tool is not limited only for creation of new database schemas, but it also provides visualization of already existing schemas and their subsequent modifications through changing ER diagram in graphic editor. The first part deals with ER diagram notation and the comparison of existing tools for the creation of ER diagrams. The second part focuses on the design and implementation of both the graphical editor, and on the synchronization of ER diagram with existing relational schema.

Key Words: relational database, ER diagram, relational schema, CASE tool

Obsah

Seznam použitých zkratk a symbolů	7
Seznam obrázků	8
Seznam tabulek	9
1 Úvod	11
2 Popis notace ER diagramu	12
2.1 Historie	12
2.2 Využití ER diagramu	13
2.3 Prvky ER diagramu	14
2.4 Kategorie modelů	16
2.5 Vývoj notací	17
2.6 Rozdíly notací	17
2.7 Nejčastěji používané notace	17
2.8 Dědičnost	17
2.9 Omezení ER diagramu	17
2.10 Shrnutí	17
3 Popis a srovnání stávajících CASE nástrojů pro modelování relačních data- bází	18
4 Návrh a implementace grafického editoru ER diagramů	19
5 Návrh a implementace synchronizace ER diagramů s relačním schématem	20
6 Závěr	21
Literatura	22
Přílohy	22
A Použité knihovny třetích stran	23
B Struktura přiloženého optického média	24
C Instalace programu	25

Seznam použitých zkratk a symbolů

SQL	– Structured Query Language
DDL	– Data Definition Language
CASE	– Computer-Aided Software Engineering
ERD	– Entity-Relationship Diagram
DBMS	– Database Management System
DSD	– Data Structure Diagram
EER	– Enhanced Entity-Relationship Model

Seznam obrázků

1	Bachmanův diagram	12
2	Srovnání DSD a ERD (Chenova notace) a) Data structure diagram b) ER diagram	13
3	Znázornění entit podle Chenovy notace	14
4	Příklad ternárního vztahu	15
5	Znázornění vztahů podle Chenovy notace	15
6	Znázornění atributů podle Chenovy notace	16

Seznam tabulek

1	Rozdíly mezi modely	17
---	-------------------------------	----

Seznam výpisů zdrojového kódu

1 Úvod

Strukturované uložení dat je dnes jednou ze základních potřeb většiny vyvíjených aplikací. S rostoucím množstvím uchovávaných dat a potřebou jejich analýzy už dávno není dostačující prosté uložení do souboru. Na tyto potřeby reagují, dnes ve velké míře používané, relační databáze. Před samotným uložením dat je nutné definovat jejich strukturu, konkrétně relační model. Dnešní svět se velmi rychle vyvíjí, tedy mění, a tyto změny často vyvolají potřebu upravit datový model tak, aby co nejlépe reflektoval modelovanou realitu. Právě pro tuto činnost jsou dnes stále častěji používané CASE nástroje, které umožňují, nejen, snadnou změnu datového modelu prostřednictvím grafického uživatelského rozhraní bez nutnosti psaní vlastních DDL skriptů.

Vyvíjená aplikace slouží pro návrh a aktualizace schématu relační databáze, konkrétně podporuje DBMS Microsoft SQL Server a Oracle Database, z toho důvodu je první kapitola věnována popisu notace ER diagramu, který se používá pro vizualizaci vztahů mezi entitami v relačním schématu nejčastěji. Další kapitola se zaměřuje na srovnání možností již existujících nástrojů pro tvorbu ER diagramu. V následující kapitole je popsán návrh a implementace grafického editoru ER diagramu, včetně algoritmů pro vyhledávání cest při vizualizaci vztahů a porovnání jejich výkonosti. Poslední kapitola se zabývá synchronizací mezi ER diagramem a relačním schématem, zejména spoluprací použitých návrhových vzorů pro dosažení požadované funkcionality pro oba, dříve zmíněné, systémy řízení báze dat.

2 Popis notace ER diagramu

V kontextu relačních databází jsou ER diagramy nejběžnějším způsobem vizualizace entit vyskytujících se v databázi a jejich vzájemných vztahů. V průběhu let vznikla celá řada notací, které se liší nejen použitými grafickými symboly pro znázornění jak entit, tak kardinality a povinnosti vztahů, ale i dalšími vlastnostmi.

Dodnes nepředstavuje žádná z notací standard v oblasti modelování databází, což je nejvíce patrné na celé řadě CASE nástrojů, které používají mnohdy svou vlastní notaci či dokonce kombinaci symbolů z několika různých notací.

2.1 Historie

Za předchůdce ER diagramu jsou považovány diagramy datových struktur (DSD), které rovněž slouží k zachycení entit, vztahů mezi nimi včetně vztažených omezení. Cílem DSD je graficky znázornit dekompozici složitých entit na jednotlivé elementy, k zachycení struktury dat se předpokládalo použití tzv. datových slovníků, které se dají zjednodušeně popsat jako množiny tabulek obsahující metadata, z tohoto principu vychází i systémové katalogy moderních DBMS.



Obrázek 1: Bachmanův diagram

Speciálním typem DSD je Bachmanův diagram, který slouží k vytvoření relačního datového modelu bez ohledu na způsob fyzického uložení dat v systému. Tento diagram měl patrně největší vliv na pozdější vznik ER diagramu, který publikoval Peter Chen roku 1976 [1].

Největším rozdílem mezi DSD a ERD, opomineme-li grafickou podobu, je fakt, že DSD se zaměřuje na vztahy mezi jednotlivými elementy entit, zatímco ERD řeší vztahy mezi samotnými entitami.



Obrázek 2: Srovnání DSD a ERD (Chenova notace)
a) Data structure diagram b) ER diagram

2.2 Využití ER diagramu

Nejčastější použití ER diagramu představuje tvorba nových relačních schémat při návrhu databází, ale rozhodně se nejedná o jedinou oblast využití. Tvorba ER diagramu velmi často stojí na počátku vývoje mnoha informačních systémů, protože relační databáze jsou obvykle primárním datovým zdrojem těchto systémů pro uložení dat transakčního charakteru.

ER diagramy umožňují zachytit vztahy a omezení entit v problémové doméně, proto jsou používány nejen při návrhu datové vrstvy, ale i v dalších fázích softwarového procesu. Doménový model aplikace je často ovlivněn právě ER diagramem relační databáze, protože na základě jejich odlišností jsme schopni odhadnout trivialitu objektově-relačního mapování. Trivialita tohoto mapování je často zásadní pro volbu vhodného návrhového vzoru pro implementaci vrstvy přístupu k datovému zdroji. Časté je i využití ER diagramu ve spojitosti s diagramy datových toků, pro znázornění uložení dat business procesu.

Dalším velmi častým použitím ER diagramu je vizualizace již existujících schémat databáze, protože právě grafické znázornění schématu je ideální pro snadné odhalení logických chyb, které mají za následek omezené možnosti aplikace, v některých případech i propady výkonu při náročnějších operacích s daty. Vizualizace existujících schémat nemusí být použita pouze pro odhalení chyb, ale také při situaci, kdy došlo ke změně některých business procesu. Tyto změny často vyvolají nutnost přizpůsobit relační model změnám, ER diagram je tedy vhodný prostředek nejen k analýze stávajícího schématu pro návrh změn, ale také pro provádění těchto změn použitím CASE nástrojů.

2.3 Prvky ER diagramu

V této části jsou popsány prvky používané při tvorbě ER diagramu, včetně speciálních případů těchto prvků a příkladů. Ilustrační grafické znázornění odpovídá notaci podle P. Chena, nicméně v další části budou ostatní používané notace dále rozvedeny.

2.3.1 Entitní typ

Entitní typy (často označovány v souvislosti s ERD jen jako entity) jsou jednoznačně identifikovatelné objekty vyskytující se v problémové doméně, které jsou pro nás zajímavé svými atributy, daty, které chceme uchovávat např. *zákazník*, *oddělení*, *automobil*. Obvykle se označují podstatným jménem v jednotném čísle.

Rozlišujeme následující typy entit:

- **Silný entitní typ** - může existovat nezávisle na ostatních entitních typech, protože obsahuje alespoň jeden atribut, který entitu jednoznačně odlišuje od ostatních, značí se obdelníkem
- **Slabý entitní typ** - jeho existence je závislá na jiném entitním typu, protože neobsahuje atribut umožňující jednoznačnou identifikaci, značí se obdelníkem s dvojitým okrajem
- **Asociativní entitní typ** - slouží k vyznačení vztahu mezi entitami, obsahuje atributy identifikující tento vztah, značí se kosočtvercem v obdelníku



Obrázek 3: Znázornění entit podle Chenovy notace

2.3.2 Vztah

Vztah slouží k vyjádření asociace mezi entitními typy, obecně vyjadřuje informace, které nelze vyjádřit pouhými entitními typy, např. student *studuje* předmět, novinář *napsal* článek. Označuje se slovesem. Znázorňuje se kosočtvercem.

Identifikující vztah je speciální typ vztahu, který spojuje slabou entitu s entitou, na které je závislá, ta se označuje jako vlastník. Značí se kosočtvercem s dvojitým okrajem.

Vztahy dělíme podle počtu entit, které se v nich vyskytují na vztahy:

- **Unární** - ve vztahu vystupuje pouze jedna entita, ta je ve vztahu sama se sebou, např. zaměstnanec *je nadřazeným* jiného zaměstnance

- **Binární** - vztah mezi dvěma entitami, např. zaměstnanec *je vedoucím* oddělení
- **Ternární** - vztah mezi třemi entitami najednou, např. učitel *doporučuje* knihu třídě
- **N-ární** - vztah mezi n entitami zároveň



Obrázek 4: Příklad ternárního vztahu

Obecně se ternární vztahy vyskytují v ER diagramech velmi málo, protože vedou k problémům s dekompozicí na několik binárních vztahů při přechodu z konceptuálního modelu na relační (logický), kdy může dojít ke ztrátě některých informací.



Obrázek 5: Znázornění vztahů podle Chenovy notace

Vztahy dále dělíme podle kardinality, která udává kolikrát může instance entitního typu být ve vztahu s instancí jiného entitního typu, pro binární typ vztahu rozlišujeme následující typy kardinality:

- **1:1**
- **1:N**
- **M:N**

Pro ternární typ vztahu rozlišujeme kardinalitu 1:1:1, 1:1:N, 1:N:M, a M:N:P, pro n-ární typy vztahu se typy kardinality odvozují analogicky.

Poslední důležitou vlastností vztahu je povinnost, určující zda může instance entitního typu existovat bez vztahu k jiné instanci entitního typu. Rozlišujeme povinné a nepovinné vztahy.

2.3.3 Atribut

Atributy jsou vlastnosti charakterizující entitní typ. Označují se podstatným jménem, znázorňují se elipsou.

Rozlišujeme následující kategorie atributů:

- **Jednoduchý** - atomický atribut, nelze dále dělit, např. *značka automobilu*
- **Složený** - dělí se na více atomických atributů, např. *adresa* se dělí na *město*, *ulici* a *PSČ*
- **Odvozený** - hodnota je vypočtena na základě hodnot ostatních atributů, fyzicky nemusí v relační databázi existovat, znázorněn je elipsou s čárkovanou hranou, např. *průměrná mzda*

Další rozdělení je podle počtu hodnot atributu na:

- **Jednohodnotové** - atribut obsahuje pouze jednu hodnotu, např. *rodné číslo*
- **Vícehodnotové** - atribut může obsahovat více hodnot, značí se elipsou s dvojitým okrajem, např. *telefonní číslo na pevnou linku i mobilní telefon*



Obrázek 6: Znázornění atributů podle Chenovy notace

Označení atributů lze kombinovat, např. *jednohodnotový odvozený atribut*

2.4 Kategorie modelů

Podle úrovně abstrakce rozlišujeme tři typy datových modelů, které využíváme při návrhu databázového schématu. Nejvyšší úroveň abstrakce nabízí konceptuální model, který zachycuje pouze entity a vztahy mezi nimi, je nejméně detailní. Více detailů nabízí logický model, poskytuje nižší úroveň abstrakce, ale stále je nezávislý na použitém DMBS. Posledním typem je fyzický model, ten vychází z logického modelu, nicméně se může v mnoha ohledech lišit, protože jeho cílem je

obsáhnout dostatečný počet technických detailů pro implementaci databáze. Rozdíly mezi logickým a fyzickým modelem způsobuje fakt, že fyzický model je vytvářen s ohledem na konkrétní použitou technologii, proto i v případě, že zvolíme relační databázi, se mohou pro různé DMBS fyzické modely více či méně lišit. Přehledné shrnutí rozdílů mezi modely nabízí tabulka 1.

Tabulka 1: Rozdíly mezi modely

Modelované vlastnosti	Konceptuální	Logický	Fyzický
Názvy entit	✓	✓	✗
Vztahy	✓	✓	✗
Atributy	✓	✓	✗
Primární klíče	✗	✓	✓
Cizí klíče	✗	✓	✓
Názvy tabulek	✗	✗	✓
Názvy sloupců	✗	✗	✓
Datové typy sloupců	✗	✗	✓

2.5 Vývoj notací

Stejně jako se vyvíjely potřeby návrhu datových modelů, vznikaly i nové notace ER diagramu. Cílem nově vznikající notací, zejména v průběhu 80-tých let, bylo odstranění některých problémů již existujících notací.

Dalším zásadním mezníkem pro vývoj notací byl objektově orientovaný přístup k vývoji softwaru, na který se mnohé notace snažily reagovat. Tento trend způsobil rozšíření notací ER modelu o koncept generalizace/specializace a dědičnosti, tento typ modelu se označuje jako Enhanced entity–relationship model (EER). Vlivem popularity objektově-relačních databází vznikly i notace umožňující v modelu zahrnout metody a operace entit.

2.6 Rozdíly notací

Na první pohled mohou různé notace vyvolávat dojem, že nejvýraznějším rozdílem mezi nimi je grafické znázornění entit a vztahů, nicméně rozdílů je mnohem více a některé mohou mít vliv na výslednou podobu fyzického modelu databáze.

2.7 Nejčastěji používané notace

2.8 Dědičnost

2.9 Omezení ER diagramu

2.10 Shrnutí

3 Popis a srovnání stávajících CASE nástrojů pro modelování relačních databází

4 Návrh a implementace grafického editoru ER diagramů

5 Návrh a implementace synchronizace ER diagramů s relačním schématem

6 Závěr

Literatura

- [1] CHEN, Peter Pin-Shan. *The entity-relationship model—toward a unified view of data.* *ACM Transactions on Database Systems* [online]. 1(1), 9-36 [cit. 2017-03-10]. Dostupné z: <http://portal.acm.org/citation.cfm?doid=320434.320440>
- [2] Lucidchart. *What is an Entity Relationship Diagram* [online]. [cit. 2017-03-10]. Dostupné z: <https://www.lucidchart.com/pages/er-diagrams>

A Použité knihovny třetích stran

- MetroLib
- AvalonDock

B Struktura přiloženého optického média

Složka **Popis**

C Instalace programu

Popis instalace.