

Databázové systémy

Entitno-relačný model

Ing. Ján Perháč, PhD.

Katedra počítačov a informatiky
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach



Košice, Slovensko

Projektovanie databáz

Dátové modelovanie

- Analýza a návrh informácií v systéme – oblasť modelovania a projektovania informačného systému.
- Dôraz na logické entity a logické závislosti medzi týmito entitami.
- Určitá úroveň dátovej abstrakcie – zakrytie detailov štruktúry, umiestnenia.

Dátový model

- množina pojmov a pravidiel opisujúca štruktúru databázy.
- Štruktúra databázy = dátové typy + vzťahy + integritné podmienky.
- Model – konceptuálny, implementačný, fyzický.

Projektovanie databáz

Proces projektovania databáz

je možné vo všeobecnosti rozdeliť do nasledujúcich etáp.

- **Systémová analýza** predmetnej oblasti a slovný opis objektov a vzťahov medzi nimi.
- **Konceptuálne projektovanie**- návrh konceptuálneho modelu predmetnej oblasti (Entitno-relačný model).
- **Voľba systému riadenia bázy dát.**
- **Implementačné projektovanie**- návrh logického modelu vzhľadom na dáta a zvolený SRBD (Relačný model v relačných databázach).
- **Fyzické projektovanie**- reálne vytvorenie objektov databázy použitím prostriedkov daného SRBD.

Entitno-relačný model

Entitno-relačný model

- konceptuálny údajový model,
- vnímanie sveta ako množiny objektov nazývaných entity a vzťahy (relácie) medzi týmito objektmi,
- používa sa na (grafické) modelovanie údajovej štruktúry v doméne – objektov a ich väzieb,
- pojmy: entita, atribút, vzťah.

Entitno-relačný model

Základné pojmy

- **Entita:**
 - existujúci objekt rozlíšiteľný od ostatných objektov - jednoznačne identifikovateľný,
 - napr. Janko Hraško, Jožko Mrkva.
- **Entitná množina:**
 - množina entít rovnakého charakteru,
 - napr. Študent, Kurz, Mesto, Kniha.
- **Atribút:**
 - vlastnosť typu entít; entity sú množinou atribútov,
 - napr. dátum narodenia, vek, meno.
- **Doména** - množina povolených hodnôt.

Entitno-relačný model

Základné pojmy

- **Vzťah (relácia, relationship):**
 - vzťah medzi entitami.
- **Vzťahová množina:**
 - matematická relácia nad množinami entít,
 - napr. učí (vzťah medzi entitami Učiteľ a Študent).
 - *Atribút* môže patriť aj vzťahovej množine (napr. rok, v ktorom začal učiteľ učiť daného študenta).

Entitno-relačný diagram (ERD)

Chen-ová notácia

- ERD je grafický nástroj pre zápis ER modelov (má viacero notácií).
- Základné stavebné bloky ERD.

- Entitná množina:



- Atribút:



- Vzťahová množina:

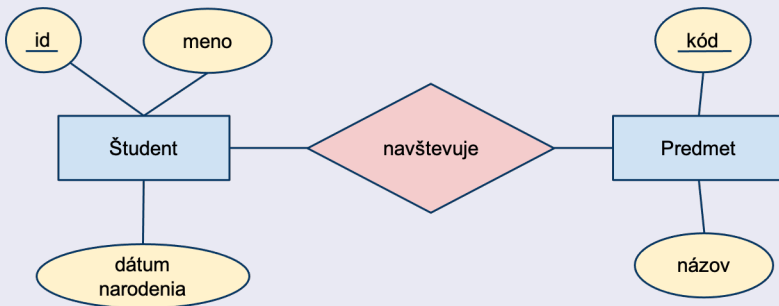


- Prepojenie:



Entitno-relačný diagram (ERD)

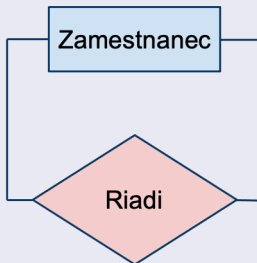
Príklad ERD



Vlastnosti vzťahov - Stupeň vzťahu I.

Stupeň vzťahu

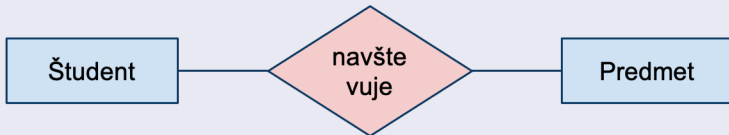
- počet zúčastnených entitných množín.
- rekurzívny vzťah (unárny):



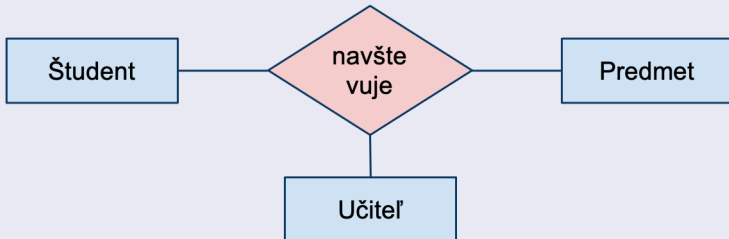
Vlastnosti vzťahov - Stupeň vzťahu II.

Stupeň vzťahu

- Binárny (najčastejšie):



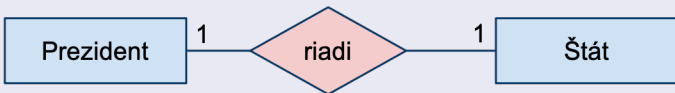
- Ternárny (prípadne n-árny, zriedkavé):



Kardinalita (násobnosť) vzťahu

3 typy kardinality

- 1:1 (One-to-One):



- 1:N (One-to-Many), N:1 (Many-to-One):

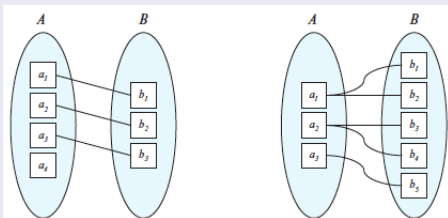


- M:N (Many-to-Many):

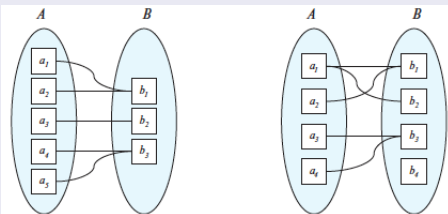


Kardinalita vzťahu - vizualizácia

1:1 (One-to-One) a 1:N (One-to-Many):



N:1 (Many-to-One) a M:N (Many-to-Many):



Povinnosť vzťahu

Povinnosť, resp. voliteľnosť vzťahu

- špecifikuje, či existencia entity závisí od existencie inej entity prostredníctvom vzťahu, ktorý je medzi týmito dvoma entitami.
- **totálna** - ku každému výskytu jednej entity musí existovať výskyt v druhej entite.
- **čiasťočná** - ku každému výskytu jednej entity môže alebo nemusí existovať inštancia druhej entity.

Povinnosť vzťahu - Príklad

V ERD dvojité prepojovacia čiara

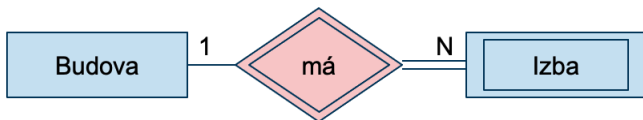


- Entitná množina s čiastočnou účasťou (partial participation) vo vzťahu - *Študent*.
 - Študent nemusí navštevovať žiaden kurz.
- Entitná množina s úplnou účasťou (total participation) - *Kurz*.
 - Ak žiadny študent nenavštevuje kurz, ten nemôže byť poskytnutý.
 - Každá entita kurzu musí byť vo vzťahu s aspoň jedným študentom.

Slabé entitné množiny

Slabá entitná množina

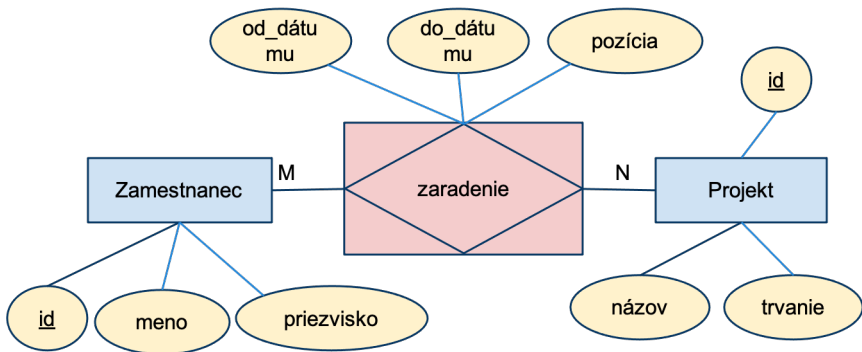
- nemá sama primárny kľúč - je identifikovaná silnou entitnou množinou vo vzťahu (identifikujúca entitná množina).
 - Diskriminátor (čiastočný kľúč) slabej entitnej množiny je množina atribútov umožňujúca odlíšiť slabé entity patriace tej istej silnej entite.
 - Primárnym kľúčom izby je kombinácia primárneho kľúča budovy (napr. adresa) a diskriminátora izby (napr. číslo izby v danej budove).



Asociatívna entitná množina

Asociatívna entitná množina

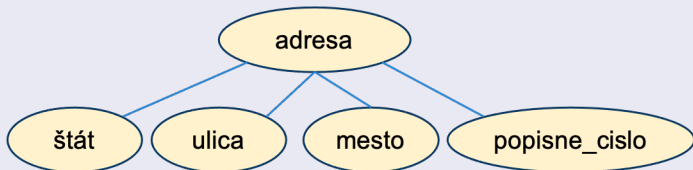
- Entitná množina vystupujúca vo význame vzťahovej množiny – asociácia (atribuovaná vzťahová množina).



Vlastnosti atribútov I.

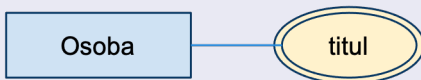
Jednoduché a zložené

- Napríklad: adresa (štát, mesto, ulica, číslo domu).



Jednohodnotové a viachodnotové

- Napríklad: titul (doc., Ing., PhD.).



Vlastnosti atribútov II.

Stále (pevné) a odvodené

- Napríklad: dátum narodenia vs. vek.



Hodnota NULL

- Špeciálna hodnota atribútu (neznáme (zatiaľ), resp. nepoužiteľné pre entitu).
- Napríklad: telefónne číslo (ak osoba nemá, resp. neuvedie, použije sa hodnota NULL) alebo ak je osoba slobodná, referencia na manžela/manželku je (zatiaľ) NULL.

Notácie ERD

Chen Style

Ordinality -
describes the
minimum
(optional vs
mandatory)



M:N



Cardinality -
describes the
maximum

1:N (n=0,1,2,3...)
one to zero or more

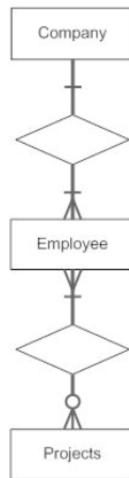
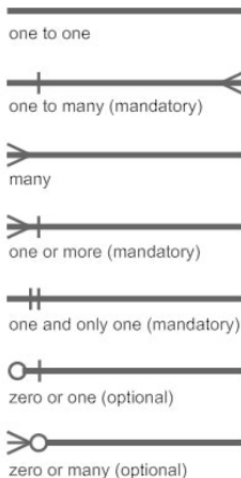
M:N (m and n=0,1,2,3...)
zero or more to zero or more
(many to many)

1:1
one to one



Notácie ERD

Information Engineering Style



Notácie ERD

Bachman Style



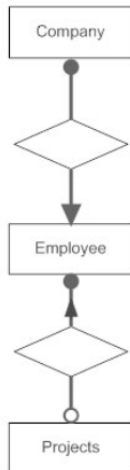
one to one



zero or more to one or more



one to one or more



Notácie ERD

Martin Style

1 - one, and only one (mandatory)

* - many (zero or more - optional)

1...* - one or more (mandatory)

0...1 - zero or one (optional)

(0,1) - zero or one (optional)

(1,n) - one or more (mandatory)

(0,n) - zero or more (optional)

(1,1) - one and only one (mandatory)



Výhody entitno-relačného modelovania

Zhrnutie

- Separácia údajov od aktivít.
- Jednoducho pochopiteľný grafický zápis.
 - Malá množina základných pojmov,
 - rýchlo naučiteľné.
- Možnosť reprezentovať štruktúry reálneho sveta.
- Priamočiara transformácia entitno-relačného modelu do relačného modelu.

Zdroje a použitá literatúra

- Jaroslav Porubän, Miroslav Biňas, Milan Nosál', *Databázové systémy*- prednášky, FEI, TUKE, 2011 - 2016.
- Henrieta Telepovská, *Databázové systémy*- prednášky, FEI, TUKE, 2008.
- Ramez Elmasri, Shamkant B. Navathe: *Fundamentals of Database Systems*, Addison Wesley, 5 edition, 2006, 1168 p. ISBN 0321369572.
- Abraham Silberschatz, Henry F. Korth, S. Sudarshan: *Database System Concepts*, The McGraw-Hill Companies, 2011, 6th edition, ISBN 978-0-07-352332-3.
- Ярцев В.П. *Організація баз даних та знань*: навчальний посібник, Державний університет телекомунікацій, 214с, 2018.
- <https://www.smartdraw.com/entity-relationship-diagram/>

Otázky?

Databázové systémy

Relačný model

Ing. Ján Perháč, PhD.

Katedra počítačov a informatiky
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach



Košice, Slovensko

Relačný model

Relačný databázový model

- Údaje sú reprezentované vo forme matematickej **relácie** (podmnožina karteziánskeho súčinu) domén atribútov.
- **Tabuľka** (relácia) reprezentuje určitú entitnú množinu (napr. Student).
- **Riadok** v tejto tabuľke (záznam) predstavuje konkrétnu entitu (napr. Student Janko Hraško).
- **Stĺpce** predstavujú jednotlivé modelované vlastnosti (atribúty) danej entitnej množiny (napr. Meno Janko).

Relačný model

Príklad

tabuľka (entitná množina)

študent		
<u>id</u>	meno	priezvisko
1	Janko	Hraško
2	Jožko	Mrkvička
3	Hanka	Veselá

riadok (entita)

stĺpec (atribút)

Relačný model

Relačný model

- Každá tabuľka má jednoznačný názov.
- Každý záznam tabuľky má rovnakú štruktúru.
- Každý stĺpec tabuľky má svoj názov.
- Na poradí stĺpcov v tabuľke nezáleží.
- Každý stĺpec obsahuje hodnoty toho istého atribútu.
- Každý záznam v tabuľke zodpovedá jednému výskytu entity daného typu.
- Každý záznam je jednoznačne identifikovateľný primárnym kľúčom.
- Na poradí záznamov nezáleží.
- Všetky hodnoty v danom zázname sú jednoznačne a plne závislé na primárnom kľúči.

Relačný model

Základné pojmy - zhrnutie:

- 1971 - princíp relačného dátového modelu bol prvýkrát navrhnutý E.F. Coddom.
- **RM** - databáza ako množina relácií.
- Každá **relácia** je podobná tabuľke príp. súboru.
- Riadok v tabuľke - množina súvisiacich **dátových hodnôt**.
- Riadok - **záznam** (tuple) resp. **inštancia** (výskyt entity).
- Hlavička (názov) stĺpca – **atribút**.
- **Tabuľka** sa nazýva relácia.
- **Doména** - dátový typ popisujúci typ hodnôt v každom stĺpci.

Relačná Schéma

relačná schéma popisuje štruktúru relácie:

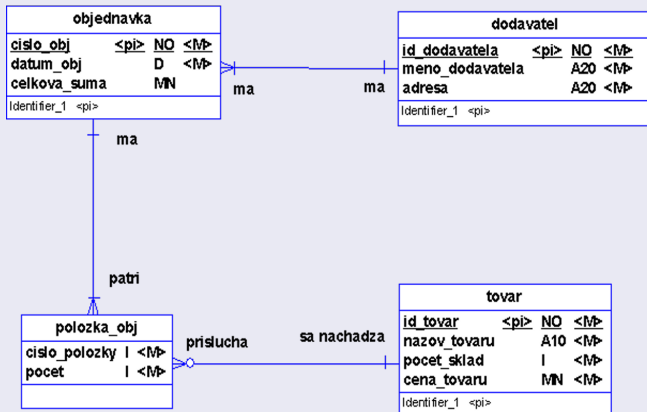
- Definícia podľa Elmasri, Navathe. Relačná schéma je:

$$R(A_1, A_2, \dots, A_n), \quad (1)$$

- kde:
 - R - meno relácie,
 - A_1, A_2, \dots, A_n - zoznam atribútov ,
 - $D = \text{dom}(A_i)$ - doména atribútu A_i ,
 - $r(R)$ - inštancia relácie r relačnej schémy $R(A_1, A_2, \dots, A_n)$, množina entít $r = t_1, t_2, \dots, t_m$,
 - Entica** - zoznam n hodnôt $t = \langle v_1, v_2, \dots, v_n \rangle$, kde každá hodnota $v_i, 1 \leq i \leq n$, je prvkom domény $\text{dom}(A_i)$ alebo špeciálna hodnota *null*.

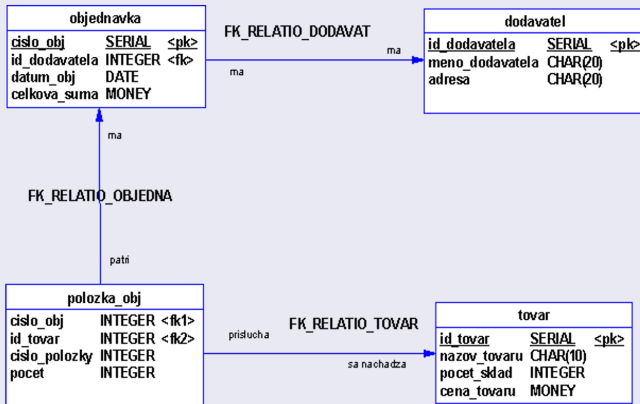
Logická schéma

Príklad



Fyzická schéma

Príklad



Relačná schéma

Príklad

- *dodavatel(id_dodavately, meno_dodavately, adresa)*
- *tovar(id_tovar, nazov_tovaru, pocet_sklad, cena_tovaru)*
- *objednavka(cislo_obj, id_dodavately, datum_obj, celkova_suma)*
- *polozka_obj(cislo_obj, id_tovar, cislo_polozky, pocet)*

Kľúče

Primárne a cudzie kľúče

- **Kľúč** je atribút (jednoduchý kľúč) alebo množina atribútov (zložený kľúč), ktoré **jednoznačne** identifikujú záznam tabuľky.
- **Primárny kľúč**:
 - nesmie byť NULL,
 - musí byť jednoznačný a minimálny (jednoznačne identifikuje záznam v tabuľke),
 - môže byť umelý, vytvorený špecificky pre identifikáciu entít (atribút id).
- **Cudzí kľúč** - jednoznačne identifikuje (odkazuje sa) záznam v inej tabuľke.

Kľúče

Super kľúč entitnej množiny

množina atribútov jednoznačne identifikujúca každú entitu (napr. id, meno, a priezvisko študenta).

Kandidátsky kľúč

minimálny super kľúč (napr. id študenta).

Primárny kľúč

jeden vybraný kandidátsky kľúč.

Kľúče

Príklad

predmet

<u>id</u>	názov	počet_kreditov
...
2	Fyzika	4
...

primárny kľúč
predmetu
Fyzika

predmet_zápis

<u>id_predmet</u>	<u>id_student</u>
...	...
2	3
...	...

cudzí kľúč
ukazujúci na
fyziku

študent

<u>id</u>	meno	priezvisko	ročník
...
3	Hanka	Veselá	5
...

Kľúče

Podmienky pre kľúče

- žiadne dve entice nemôžu mať tú istú kombináciu hodnôt atribútov:

$$t_1[SK] \neq t_2[SK],$$

- kde:
 - SK - podmnožina atribútov, superkľúč relačnej schémy R ,
 - t_1, t_2 - dve jedinečné inštancie relácie.

Integrita údajov

Integrita údajov v databáze

- **Integrita údajov** - databáza sa musí starať o to, aby boli uložené údaje celý čas konzistentné a neporušené.
- DBS poskytuje mechanizmy (tzv. integritné obmedzenia) na zabezpečenie integrity údajov pri operáciách na vloženie (*INSERT*), aktualizovanie (*UPDATE*) a odstránenie (*DELETE*) záznamov,
- Rozličné typy integritných obmedzení:
 - doménová integrita,
 - entitná integrita,
 - referenčná integrita.

Integrita údajov

Entitná integrita

- **Primárny kľúč** nemôže mať hodnotu *NULL*.
- V rámci relácie (tabuľky) sú riadky jednoznačne odlišené primárnym kľúčom.
 - 2 rôzne riadky majú rôznu hodnotu primárneho kľúča.
- Zabezpečuje jednoznačnú identifikovateľnosť entity.
 - Napríklad dvaja rôzni študenti s rovnakým menom - potrebujeme správne určiť, ktorý dostane A a ktorý E.
 - Slabá entita je však identifikovaná aj jej identifikujúcim vzťahom.

Integrita údajov

Doménová integrita

- Doména definuje množinu povolených hodnôt v stĺpci relácie (obor hodnôt).
- Hodnota každého stĺpca musí byť:
 - atomická,
 - z definovanej domény.
- Príklady:
 - Vek je kladné celé číslo.
 - Pohlavie je muž alebo žena.
- Zabezpečuje rovnakú štruktúru záznamov a v dôsledku toho ich spracovateľnosť.

Integrita údajov

Referenčná integrita

- Každý cudzí kľúč musí
 - jednoznačne identifikovať záznam z inej tabuľky na základe jej primárneho kľúča,
 - alebo má hodnotu NULL, ak vzťah neexistuje.
- Zabezpečuje správnosť vzťahov (napr. študent neštuduje neexistujúci predmet).

Normalizácia

Normalizovaná schéma

- Ciel':
 - ukladať údaje bez duplicít,
 - zabrániť anomáliám pri manipulácii s údajmi.
- **Normálne formy:**
 - Sada pravidiel opisujúcich "*ideálnu databázu*" (v niekoľkých úrovniach) z pohľadu redundancie.
 - Každá úroveň vyžaduje splnenie svojho pravidla a všetkých pravidiel nižších úrovní.

Normalizácia

Normalizácia

- Založená na závislostiach medzi stĺpcami v tabuľkách.
- Dobrý a premyslený návrh ER modelu zvyčajne vedie k vhodnému naplneniu normalizačných podmienok.
 - Napríklad používanie jednoduchých a jednohodnotových atribútov.
- Z dôvodu výkonu nemusí byť normalizácia vždy žiadaná.

Normalizácia

Normálne formy

- V praxi používané tri normálne formy:
 - 1 NF - každý atribút obsahuje iba atomické hodnoty.
 - Napríklad meno, priezvisko.
 - 2 NF - každý neklúčový atribút funkčne závisí na celom primárnom kľúči.
 - Kľúč tabuľky má jediný účel (problém pri zložených kľúčoch).
 - 3 NF - atribúty sú navzájom nezávislé.
 - Atribút závisí iba na primárnom kľúči.
- Praktický dôsledok použitia NF - údaje sa nachádzajú vo viacerých tabuľkách.

Normalizácia

Porušenie 1. NF

- Ak dodatočne spracúvame údaje z nejakého atribútu, jedná sa porušenie 1. NF
 - Napríklad uloženie mena a priezviska do jedného atribútu.
 - Z neužívanie atribútu poznámka.
- Porušenie 1. NF si vyžaduje dodatočné parsovanie údajov v aplikačnom kóde (to by mala byť úloha DB).

User	
<u>username</u>	user
nezbednik	Feri Bandaska 01.04.87 (M)
juzek	Jozef Rapavý 04.04.88 (M)

Normalizácia

Porušenie 2. NF

- Ak je niektorý atribút závislý iba na časti kľúča, jedná sa o porušenie 2. NF.
- V skutočnosti sú to viaceré entitné množiny.

<u>username</u>	name	surname	birthday	sex	<u>date</u>	type	text
nezbednik	Feri	Bandaska	01.04.87	M	16/02/2016 10:09:00	friends	Ako si...
nezbednik	Feri	Bandaska	01.04.87	M	21/02/2016 06:00:55	public	Lenivec,...
juzek	Jozef	Rapavý	04.04.88	M	15/02/2016 13:33:00	friends	Dota je...
anezka	Anežka	Pekná	NULL	F	NULL	NULL	NULL

Normalizácia

Porušenie 3. NF

- Tranzitívna závislosť.
 - Ďalšia redundancia údajov.
- Opäť je vhodné vytvoriť viacero tabuliek (autor je prirodzená entitná množina so vzťahom ku knihe).

Kniha				
<u>id_kniha</u>	názov	meno autora	priezvisko autora	rok narodenia autora
1	Starec a more	Ernest	Hemingway	1899
2	Zbohom zbraniam	Ernest	Hemingway	1899

Normalizácia

Porušenie 3. NF

- Napr. dopočítavané stĺpce.
 - Aktuálnu cenu je možné vždy vypočítať.
 - Niekedy je porušenie potrebné kvôli výkonu (predvypočítanie si častých/výpočtovo náročných dopytov).

Nakup				
<u>id zakaznik</u>	<u>id tovar</u>	kusov	cena	celková cena
1	2	10	20	200
1	13	10	30	300

Normalizácia

Tipy

- Používať atomické hodnoty.
 - Nekódovať viacero atribútov do jedného stĺpca.
- Viachodnotové a zložené atribúty nahradiť novou entitnou množinou.
 - Napr. pri adrese osoby nevznikne viacero atribútov osoby, ale nová tabuľka adresa so vzťahom s osobou.
- Nedopočítavať stĺpce z iných.

Normalizácia

Tipy

- Jasne oddelovať entitné množiny (tabuľky).
 - Každá tabuľka má mať jeden účel.
 - Napr. tabuľka zaznamenávajúca atribúty používateľa by nemala zaznamenávať aj jeho posty (príspevky) - to už je ďalšia tabuľka, tabuľka Post.
- Pozor na opakujúce sa hodnoty v súvisiacich stĺpcoch.
 - Napr. v prípade, že spojíme posty a používateľov, ak bude používateľ publikovať viacero postov, pri každom sa zopakujú všetky atribúty jeho autora.

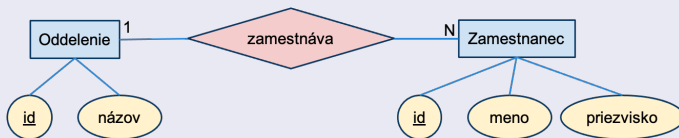
Transformácia ERM na RM

Transformácia ERM na RM

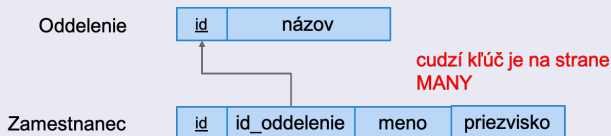
- Pre implementáciu databázy v relačnom SRBD (RDBMS) je nutné transformovať ERM na RM.
- Vzhľadom na podobnosť modelov je prechod jednoducho realizovateľný:
 - Meno entity = meno tabuľky.
 - Atribút entity = stĺpec tabuľky (pokiaľ nie je zložený/viachodnotový).
 - Vzťah = kombinácia primárny a cudzí kľúč (referenčná integrita).
 - Je potrebné doplniť pravidlá pre doménovú a entitnú integritu.

Transformácia relácie 1:N (1:1)

Entitno-relačný diagram

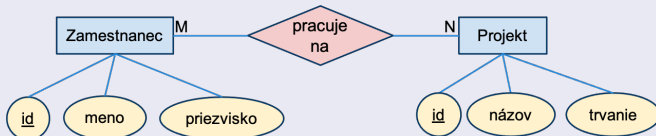


Relačný model

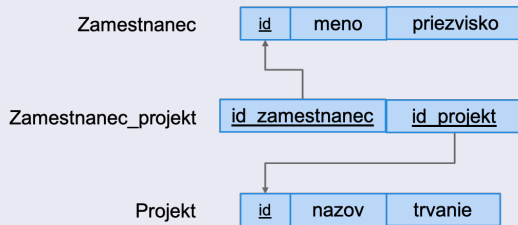


Transformácia relácie M:N

Entitno-relačný diagram

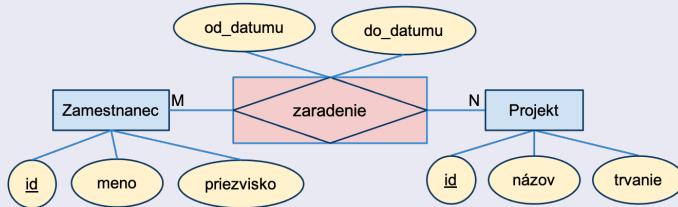


Relačný model



Transformácia relácie s asociatívnou entitou

Entitno-relačný diagram



Relačný model

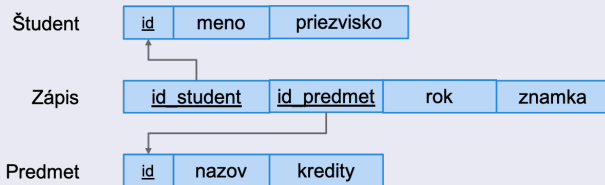


Kľúče asociatívnych entít

Podmienky

- Minimálnym primárnym kľúčom vzťahu zvykne byť kombinácia cudzích kľúčov.
- Správny kľúč vyplýva zo sémantiky vzťahu:
 - id_student, id_predmet
 - id_student, id_predmet, znamka
 - id_student, id_predmet, rok
 - ...

Relačný model



Kľúčové slová

Kľúčové slová

- Entitno-relačný model, Relačný model.

Key words

- Entity–relationship model, Relational model.

Ключевые слова

- Модель «сущность – связь», Реляционная модель данных.

Ключові слова

- Модель «сутність - зв'язок», Реляційна модель даних.

Zdroje a použitá literatúra

- Jaroslav Porubän, Miroslav Biňas, Milan Nosál', *Databázové systémy- prednášky*, FEI, TUKE, 2011 - 2016.
- Henrieta Telepovská, *Databázové systémy- prednášky*, FEI, TUKE, 2008.
- Ramez Elmasri, Shamkant B. Navathe: *Fundamentals of Database Systems*, Addison Wesley, 5 edition, 2006, 1168 p. ISBN 0321369572.
- Abraham Silberschatz, Henry F. Korth, S. Sudarshan: *Database System Concepts*, The McGraw-Hill Companies, 2011, 6th edition, ISBN 978-0-07-352332-3.
- Ярцев В.П. *Організація баз даних та знань: навчальний посібник*, Державний університет телекомунікацій, 214с, 2018.

Otázky?

Databázové systémy

Jazyk SQL - DDL časť

Ing. Ján Perháč, PhD.

Katedra počítačov a informatiky
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach



Košice, Slovensko

Jazyk SQL - Structured Query Language

Jazyk SQL - Štrukturovaný dopytovací jazyk

- Je štandardizovaný jazyk určený na komunikáciu s relačným SRBD pre ukladanie, manipuláciu a získavanie dát v relačných databázach.
- Neprocedurálny (deklaratívny) jazyk.
 - Čo chceme získať, nie ako chceme údaje získať (nie je to postupnosť krokov).
 - SRBD získa údaje z databázy (fyzické úložisko).
- Založený na relačnom kalkule n-tíc.

SQL - Structured Query Language

História SQL

- Začiatok 1970 rokov – IBM vyvinula pôvodnú verziu SQL (nazývaná Sequel).
- Začiatok 1980 rokov – existuje množstvo SRBD. Každý používal svoj jazyk dopytov (Query Language).
- Rok 1986 – prvý štandard SQL86 (ANSI, ISO: SQL-87).
- Aktualizácie štandardu v rokoch 1989, 1992, 1996, 1999, 2003, 2006, 2008, 2011, 2016 [1].

[1] STANDARDS BY ISO/IEC JTC 1/SC 32 Data management and interchange, dostupné online

<https://www.iso.org/committee/45342/x/catalogue/p/1/u/0/w/0/d/0>

SQL - Structured Query Language

Jazyk SQL - Štrukturovaný dopytovací jazyk

- Je štandardizovaný jazyk pre ukladanie, manipuláciu a získavanie dát v relačných databázach..
- Typy príkazov jazyka SQL:
 - **DDL** (Data Definition Language) – jazyk na definíciu štruktúry databázy vrátane zmien v štruktúre – CREATE, DROP, ALTER, TRUNCATE.
 - **DML** (Data Manipulation Language) – jazyk na manipuláciu s údajmi vrátane dopytov – SELECT, INSERT, UPDATE, DELETE.
 - **DCL** (Data Control Language) – jazyk pre riadenie prístupu k údajom – GRANT, REVOKE.
 - **TCL** (Transaction Control Language) - jazyk pre riadenie správy transakcií - COMMIT, ROLLBACK, BEGIN WORK.

Údajové typy

Údajové typy

- Údajový typ definuje množinu hodnôt, ktoré môže nadobúdať stĺpec (doména atribútu).
- Umožňuje definovať doménovú integritu.
- Kategórie niektorých zabudovaných údajových typov pre prácu s:
 - reťazcami znakov (bitov),
 - číselnými hodnotami,
 - dátumom a časom,
 - multimediálnymi údajmi (BYTEA).

Údajové typy

Základné údajové typy

- *INTEGER* - celé čísla,
- *FLOAT* - desatinné čísla,
- *NUMERIC*(*n,s*) - číslo s deklarovanou presnosťou,
- *CHAR*(*n*) - reťazec s fixnou dĺžkou,
- *VARCHAR*(*n*) - reťazec s premenlivou dĺžkou,
- *DATE* - rok, mesiac, dátum, čas, minúty, sekundy,
- *TIMESTAMP* - rozširuje typ *DATE* o zlomky sekúnd.

Jazyk SQL - CREATE TABLE

Príkaz CREATE TABLE

- Vytvára tabuľku (schémy jednotlivých vzťahov).
- Syntax:

```
CREATE TABLE meno_tabulky  
( nazov1 typ1 [obmedzenia],  
  nazov2 typ2 [obmedzenia],  
  ... ,  
  nazovN typN [obmedzenia]  
);
```

Jazyk SQL - CREATE TABLE

Príklad použitia CREATE TABLE

```
CREATE TABLE DBSUser (  
    username VARCHAR(30),  
    name VARCHAR(30),  
    surname VARCHAR(40),  
    birthday DATE,  
    sex CHAR  
);
```

Jazyk SQL - DROP TABLE

Príklad použitia DROP TABLE

- Odstráni tabuľku.
- Syntax:

DROP TABLE nazov_tabulky ;

- Príklad:

DROP TABLE DBSUser ;

Jazyk SQL

Preventívne odstránenie

- Odstránenie predchádzajúcej verzie tabuľky (alebo pohľadu) pred vytvorením novej.
- Umiestňuje sa pred samotné vytvorenie tabuľky (alebo pohľadu).
- Zabezpečí dostupnosť názvu.
- Syntax:

```
DROP TABLE IF EXISTS nazov;  
CREATE TABLE nazov ...;
```

Jazyk SQL

Odporúčané kroky pri vytváraní tabuľky

- Identifikovať údajové typy jednotlivých atribútov (stĺpcov),
- identifikovať stĺpce, ktoré musia a ktoré nesmú byť nulitné,
- identifikovať stĺpce, ktoré musia byť unikátne,
- identifikovať páry primárnych a cudzích kľúčov,
- identifikovať predvolené hodnoty,
- identifikovať obmedzenia pre stĺpce (vzhľadom na použitú doménu),
- nepoužívať rezervované slová jazyka SQL ako názvy atribútov, tabuliek atď.,
- vytvoriť tabuľku.

Jazyk SQL

Ohraničenia pre tabuľky

- Vynútenie ohraničenia nad tabuľkou pri vkladaní, zmene alebo mazaní:
 - povinnosť uvedenia hodnoty – *NOT NULL*,
 - unikátne hodnoty – *UNIQUE*,
 - primárne kľúče – *PRIMARY KEY*,
 - cudzie kľúče – *FOREIGN KEY*,
 - kontrola – *CHECK*,
 - predvolená hodnota – *DEFAULT*.

Jazyk SQL

Povinnosť uvedenia hodnoty, unikátne a predvolené hodnoty

```
CREATE TABLE osoba (  
    username VARCHAR(30) PRIMARY KEY,  
    name VARCHAR(30) NOT NULL,  
    surname VARCHAR(40) NOT NULL,  
    birthday DATE,  
    sex CHAR(1) DEFAULT 'M'  
);
```


Jazyk SQL

NOT NULL, UNIQUE a PRIMARY KEY

- **NOT NULL** stĺpec:
 - **nie je možné** vložiť hodnotu **NULL**
 - hodnoty sa môžu opakovať.
- **UNIQUE** stĺpec:
 - **je možné** vložiť hodnotu **NULL**
 - hodnoty sa nemôžu opakovať.
- **PRIMARY KEY** stĺpec:
 - **nie je možné** vložiť hodnotu **NULL**
 - hodnoty sa nemôžu opakovať.

Jazyk SQL

Primárne kľúče - jednoduchý kľúč (jeden stĺpec)

```
CREATE TABLE predmet (  
    id INT PRIMARY KEY,  
    nazov VARCHAR(32) NOT NULL);
```

```
CREATE TABLE predmet (  
    id INT,  
    nazov VARCHAR(32) NOT NULL,  
    CONSTRAINT predmet_pk PRIMARY KEY (id)  
);
```

Jazyk SQL

Primárne kľúče - zložený kľúč (viacero stĺpcov)

```
CREATE TABLE predmet_zapis (  
    id_predmet INT,  
    id_student INT,  
    PRIMARY KEY (id_predmet , id_student )  
);
```

Jazyk SQL

Cudzie kľúče a referenčná integrita I.

- Cudzí kľúč predstavuje vzťah (linku) medzi stĺpcami dvoch tabuliek:
 - tabuľka s cudzím kľúčom: *child table*,
 - tabuľka obsahujúca primárny kľúč: *referenced (parent) table*.
- Kľúčové slovo: *REFERENCES*.
- Syntax:

`REFERENCES tabulka (stlpec)`

Jazyk SQL

Cudzie kľúče a referenčná integrita II.

```
CREATE TABLE predmet_zapis (  
    id_predmet INT REFERENCES predmet(id),  
    id_student INT REFERENCES student(id),  
    PRIMARY KEY (id_predmet, id_student)  
);
```

Jazyk SQL

Cudzie kľúče a referenčná integrita III.

```
CREATE TABLE predmet_zapis (  
  id_predmet INT,  
  id_student INT,  
  PRIMARY KEY (id_predmet , id_student),  
  FOREIGN KEY(id_predmet) REFERENCES predmet(id),  
  FOREIGN KEY(id_student) REFERENCES student(id)  
);
```

Jazyk SQL

Kontrola CHECK

- Kontrola podmienky pri vkladaní a aktualizácii; môže ich byť viacero:

```
CREATE TABLE student (  
    ...  
    rocnik INT NOT NULL  
    CHECK(rocnik >=1 AND rocnik <=5)  
);
```

Jazyk SQL

Príkaz ALTER TABLE

- Zmení štruktúru tabuľky:
 - pridaním (ADD) stĺpca,
 - odstránením (DROP COLUMN) stĺpca, alebo
 - zmenením (ALTER COLUMN) stĺpca.
- Syntax:

```
ALTER TABLE tabulka  
operacia stlpec [typ]  
[ohranicenie];
```


Jazyk SQL

Príklad použitia príkazu ALTER TABLE

- Pridanie stĺpca:

```
ALTER TABLE student ADD sex CHAR;
```

- Odstránenie stĺpca:

```
ALTER TABLE student DROP COLUMN sex;
```

- Zmena stĺpca:

```
ALTER TABLE tabulka  
ALTER COLUMN name TYPE VARCHAR(20);
```

Zdroje a použitá literatúra

- Jaroslav Porubän, Miroslav Biňas, Milan Nosál', *Databázové systémy*- prednášky, FEI, TUKE, 2011 - 2016.
- Henrieta Telepovská, *Databázové systémy*- prednášky, FEI, TUKE, 2008.
- Ramez Elmasri, Shamkant B. Navathe: *Fundamentals of Database Systems*, Addison Wesley, 5 edition, 2006, 1168 p. ISBN 0321369572.
- Abraham Silberschatz, Henry F. Korth, S. Sudarshan: *Database System Concepts*, The McGraw-Hill Companies, 2011, 6th edition, ISBN 978-0-07-352332-3.
- Ярцев В.П. *Організація баз даних та знань*: навчальний посібник, Державний університет телекомунікацій, 214с, 2018.

Otázky?

Databázové systémy

Jazyk SQL - DML časť

Ing. Ján Perháč, PhD.

Katedra počítačov a informatiky
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach



Košice, Slovensko

SQL - Structured Query Language

Jazyk SQL - Data Manipulation Language

- **DML** (Data Manipulation Language) – jazyk na manipuláciu s údajmi vrátane dopytov:
 - SELECT,
 - INSERT,
 - UPDATE,
 - DELETE.

Jazyk SQL - INSERT

Príkaz INSERT

- Vloží nový riadok do tabuľky.
- Syntax 1. variant (záleží na poradí a počte atribútov):

```
INSERT INTO tabulka  
VALUES (hod1, ..., hodN);
```

- Syntax 2. variant (nezáleží na poradí):

```
INSERT INTO tabulka  
(naz_poloz1, ..., naz_polozN)  
VALUES (hod1, ..., hodN);
```

Jazyk SQL - INSERT

Príkaz INSERT

```
INSERT INTO student  
(id, meno, priezvisko)  
VALUES (1, 'Janko', 'Hrasko');
```

```
INSERT INTO student (meno, priezvisko, id)  
VALUES ('Jozko', 'Mrkvicka', 2);
```

```
INSERT INTO student  
VALUES (3, 'Hanka', 'Vesela');
```

Jazyk SQL - UPDATE

Príkaz UPDATE

- Aktualizuje údaje v tabuľke.
- Syntax:

```
UPDATE tabulka  
SET atr1=nova_hod1 [ ,... atrN=nova_hodN ]  
[WHERE podmienka];
```

- Príklad:

```
UPDATE student  
SET name='Fferko' WHERE id=3;
```


Jazyk SQL - DELETE

Príkaz DELETE

- Odstráni riadok/riadky z tabuľky:
- Syntax:

```
DELETE FROM tabulka  
[WHERE podmienka];
```

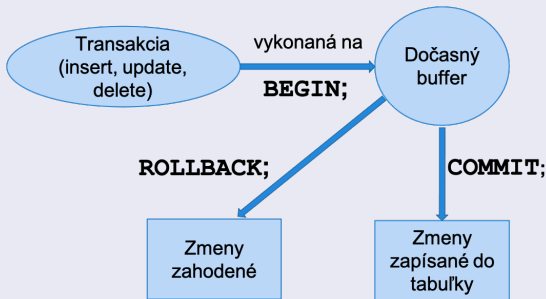
- Príklad:

```
DELETE FROM student  
WHERE id=3;
```

Jazyk SQL - transakcie

Transakčné spracovanie

- Sekvencia modifikačných príkazov vykonaná ako jeden celok - **transakcia**.
 - Buď sa zapíšu úspešne všetky zmeny,
 - alebo sa nezapíše žiadna.



Jazyk SQL - TRUNCATE TABLE

Príkaz TRUNCATE TABLE

- Odstráni všetky riadky z tabuľky:
- Syntax:

```
TRUNCATE TABLE tabulka ;
```

- Príklad:

```
TRUNCATE TABLE student ;
```

- Rovnaký výsledok je možné dosiahnuť použitím (TRUNCATE však nie je transakčná operácia):

```
DELETE FROM tabulka ;
```

Jazyk SQL

Dodržanie referenčnej integrity I.

- Porušenie referenčnej integrity:
 - Odstránenie záznamu, na ktorý sa odkazuje iný záznam.
 - Vloženie/aktualizácia referencie na neexistujúci záznam.
- Typy prístupov:
 - **kaskádny,**
 - **reštrikčný,**
 - **nulitný.**
- Predvolené správanie (PostgreSQL):

ON DELETE NO ACTION

Jazyk SQL

Dodržanie referenčnej integrity II.

- **Kaskádny prístup:** *CASCADE*
 - Ak je zmazaný riadok v odkazujúcej tabuľke, sú zmazané aj riadky v príslušných odkazovaných tabuľkách.
- **Reštrikčný prístup:** *RESTRICT (NO ACTION)*
 - Záznam nemôže byť zmazaný, ak naň existuje odkaz pomocou cudzieho kľúča.
- **Nulitný prístup:** *SET NULL*
 - Ak je príslušný záznam v odkazovanej tabuľke odstránený alebo aktualizovaný, cudzí kľúč sa nastaví na *NULL*.

Jazyk SQL

Dodržanie referenčnej integrity - Príklad

```
CREATE TABLE predmet_zapis (  
    id_predmet INT,  
    id_student INT,  
    PRIMARY KEY (id_predmet, id_student),  
    FOREIGN KEY(id_predmet)  
    REFERENCES predmet(id) ON DELETE CASCADE,  
    FOREIGN KEY(id_student)  
    REFERENCES student(id) ON DELETE CASCADE  
);
```

Jazyk SQL - SELECT

Príkaz SELECT

- Najdôležitejší príkaz v SQL.
- Získa (vyfiltruje) údaje z tabuľky.
 - Vybrané riadky (WHERE).
 - Vybrané stĺpce (SELECT).
- Syntax:

```
SELECT   co
FROM    odkial
WHERE    co_nas_zaujima
ORDER BY podla_coho_triedit
DESC;
```

Jazyk SQL - SELECT

Príkaz SELECT - Príklad

```
SELECT * FROM student;
```

```
SELECT meno, priezvisko  
FROM student;
```

```
SELECT meno, priezvisko  
FROM student  
ORDER BY priezvisko, meno;
```

```
SELECT meno, priezvisko  
FROM student  
WHERE meno = 'Janko';
```


Zdroje a použitá literatúra

- Jaroslav Porubän, Miroslav Biňas, Milan Nosál', *Databázové systémy*- prednášky, FEI, TUKE, 2011 - 2016.
- Henrieta Telepovská, *Databázové systémy*- prednášky, FEI, TUKE, 2008.
- Ramez Elmasri, Shamkant B. Navathe: *Fundamentals of Database Systems*, Addison Wesley, 5 edition, 2006, 1168 p. ISBN 0321369572.
- Abraham Silberschatz, Henry F. Korth, S. Sudarshan: *Database System Concepts*, The McGraw-Hill Companies, 2011, 6th edition, ISBN 978-0-07-352332-3.
- Ярцев В.П. *Організація баз даних та знань*: навчальний посібник, Державний університет телекомунікацій, 214с, 2018.

Otázky?