

28. Relační model, základní konstrukty, realizace vztahů v relačním modelu, integritní omezení. Normalizace, normální formy, funkční závislosti, aktualizací anomálie.

### Relační model

- založen na pevném matematickém základu
- koncept relací, relační algebra, relační kalkul
- uložení objektů (**entit**) a vztahů mezi nimi ve formě relací (**tabulek**)

### základní konstrukty

- Relace
  - schéma relace
  - n-tice
- Atribut
  - kandidátní klíč
- Integritní omezení
  - entitní
    - doménové
    - null
    - jedinečnost
    - pro primární klíč
  - referenční integrita
    - cizí klíč
  - enterprise

## Konstrukty relačního modelu

**Relace R** je pojmenovaná tabulka s řádky a sloupci (množina n-tic)

**Atribut A** je pojmenovaný sloupec

**Doména D** je množina všech přípustných hodnot pro atribut (nebo skupinu atributů)

**Stupeň n** (arita) je počet atributů relace

- n – tice je řádek tabulky
- počet n-tic je proměnlivý

### Relace

- **Schéma relace** (struktura tabulky)  
 $R(A_1:D_1, \dots, A_n:D_n)$
- Relace R je podmnožinou kartézského součinu  
 $D_1 \times \dots \times D_n$
- Relace je m x n-tice  
 $m \times (A_1:d_1, \dots, A_n:d_n)$ , kde  $d_1 \in D_1, \dots, d_n \in D_n$
- Konkrétní data tvoří řádky tabulky. Řádky tabulky reprezentují vztahy mezi daty.

### Vlastnosti relace:

- každá relace má jedinečný název
- každý prvek relace obsahuje právě jednu hodnotu
- každý atribut má jedinečné jméno v rámci relace a hodnotu z domény
- n-tice se v relaci neopakují
- každá n-tice musí být jednoznačně identifikovatelná

## Vlastnosti relace

---

- **Kandidátní klíč**
  - je atribut nebo kombinace atributů, které jednoznačně identifikují n-tice a žádný atribut v nich není nadbytečný.
- **Primární klíč**
  - nejvhodnější kandidátní klíč
  - může být uměle vytvořený
  - podtrháváme jej příkl.  
jednoduchý R1(klic, normalny\_atribut)  
složený R2(klic\_cast1, klic\_cast2, normalny\_atribut)
- **Cizí klíč**
  - je atribut, který reprezentuje primární klíč jiné relace
  - reprezentuje vztah

### Integritní omezení

- **Entitní integrita**
  - Doména atributu – množina přípustných hodnot
  - povinný/nepovinný atribut (NULL resp. absence hodnoty)
  - jedinečnost (unique)

**V relaci musí být všechny atributů primárního klíče povinné. Nesmí být NULL.**

- **Referenční integrita**
  - vztah mezi cizím klíčem jedné relace a primárním klíčem druhé relace

**Když relace obsahuje cizí klíč, hodnota cizího klíče se musí rovnat hodnotě primárního klíče některé n-tice v rodičovské relaci nebo musí být NULL.**

## Relační návrh databáze

---

Vychází z konceptuálního schématu a vytvořené dokumentace

1. Přizpůsobit konceptuální schéma
2. Transformovat na relační schéma
3. Validovat relační schéma normalizací
4. Validovat relační schéma vůči požadovaným transakcím
5. Nakreslit diagram
6. Definovat integritní omezení
7. Zkonzultovat navržený diagram se zákazníkem

### Přizpůsobení:

#### 1. Odstranit M:N vztahy

M:N vztah rozložíme na dva vztahy 1:M s pomocí nové vazebné entity

#### 2. Odstranit vztahy s atributy

Vztah s atributem rozložíme na dva vztahy s pomocí nové vazební entity, která bude mít daný atribut

#### 3. Odstranit vícehodnotové atributy

Vícehodnotový atribut nahradíme novou entitou a vztahem 1:M

- pokud nevytvoříme umělý identifikační klíč, vazební entita bude identifikačně a existenčně závislá na entitách, které spojuje a vztahy mezi nimi budou identifikační. Identifikační klíč vazebné entity bude složený z primárních klíčů entit, které spojuje.
- pokud vytvoříme umělý identifikační klíč, vazební entita bude jenom existenčně závislá na entitách, které spojuje a vztahy mezi nimi budou neidentifikační

#### 4. Odstranit n-ární vztahy

Každý nebinární vztah může být reprezentován pomocí nové umělé entity a binárních vztahů mezi novou entitou a všemi původními entitami

#### 5. Překontrolovat vazby 1:1

Zkontrolovat jestli se nejedná o stejnou entitu, pokud ano vytvořit společnou entitu

#### 6. Odstranit redundantní vztahy

Vztah je **redundantní**, když je možné získat stejnou informaci přes jiný vztah.

- pozor na časové hledisko

### Převedení na Relační model:

1. **Silná entita** se transformuje na relaci, která obsahuje všechny jednoduché atributy entity. U složených atributů zahrneme jenom jejich jednoduché složky.

**STUDENTI**(rodne\_cislo, jmeno, prijmeni, mesto, ulice, psc, prumer)

**Primární klíč** rodne\_cislo

**SKOLY** (id\_skola, nazev, mesto, skolne)

**Primární klíč** id\_skola //umělý

2. **Slabá entita** se transformuje na relaci, která obsahuje všechny jednoduché atributy entity doplněné o cizí klíč. Cizím klíčem je primární klíč identifikačního vlastníka/ů.

**PRIHLASKY**(rodne\_cislo, id\_skola, uspesnost)

**Primární klíč** rodne\_cislo + id\_skola

**Cizí klíč** rodne\_cislo z relace STUDENTI, id\_skola z relace SKOLY

3. **Vztah 1:M** mezi entitami E1 (rodič) a E2 (dítě) – relaci, která reprezentuje E2 doplníme o cizí klíč. Cizím klíčem je primární klíč E1. Entita, která je na straně „1“ je rodič a entita, která je na straně „M“ je dítě.

**STUDENTI** (rodne\_cislo, jmeno, ..., id\_tymovaPrace)

**Primární klíč** rodne\_cislo

**Cizí klíč** id\_tymovaPrace z relace TYMOVE\_PRACE

**TYMOVE\_PRACE** (id\_tymovaPrace, nazev)

**Primární klíč** id\_tymovaPrace

Při nepovinném členství, aby jsme se vyhnuli NULL, vytvořit vazební relaci.

4. **Vztah 1:1** mezi entitami E1 (rodič) a E2 (dítě) – relaci, která reprezentuje E2 doplníme o cizí klíč. Cizím klíčem je primární klíč E1. Entita, která má nepovinné členství ve vztahu je rodič a entita, která má povinné členství je dítě.

- pokud je členství u obou entit povinné
  - můžeme si vybrat, která entita bude rodič a která dítě
  - nebo můžeme vytvořit jednu společnou relaci
- pokud je členství u obou entit nepovinné
  - můžeme si vybrat, která entita bude rodič a která dítě
  - nebo, aby jsme se vyhnuli NULL, vytvořit vazební relaci

## 5. **Rekurzivny vztah** na entitě E1

- relaci reprezentující E1 doplníme o cizí klíč. Cizím klíčem je primární klíč E1.

**ZAMESTNANCI** (id\_zamestnanec, ..., id\_nadrizeny)

**Primární klíč** id\_zamestnanec

**Cizí klíč** id\_nadrizeny odkazuje na atribut id\_zamestnanec z relace ZAMESTNANCI

- nebo aby jsme se vyhnuli NULL, vytvoříme novou relace se dvěma cizími klíči

**ZAMESTNANCI** (id\_zamestnanec, ...)

**Primární klíč** id\_zamestnanec

**PRIRAZENI** (id\_zamestnanec, id\_nadrizeny)

**Cizí klíč** id\_zamestnanec z relace ZAMESTNANCI,  
id\_nadrizeny odkazuje na atribut id\_zamestnanec z relace ZAMESTNANCI

## integritní omezení

Když data v databáze splňují integritní omezení, které požaduje specifikace požadavků a relační model je databáze v **konzistentním** stavu.

- IO vycházející z konceptuálního modelu
- entitní integrita (primární klíč nesmí být NULL)
- referenční integrita (existence cizího klíče)
- podniková (enterprise) omezení

## Zabezpečení referenční integrity

---

Může být cizí klíč NULL?

- jestli je členství dítěte povinné Not NULL

Primární klíč

- vložení nového – nemá vliv
- změna – různé strategie
- vymazání – různé strategie

Cizí klíč

- vložení – kontrola referenční integrity
- změna – kontrola referenční integrity
- vymazání – nemá vliv, pokud povolen NULL

Strategie když existuje dítě, kterého hodnota cizího klíče odkazuje na primární klíč, který má být změněn, nebo vymazán.

1. NO ACTION – nemůžeš smazat, existují na něj odkazy
2. CASCADE – vymaž i záznamy, které na něj odkazují
3. SET NULL – vymaž a odkaz nastav na NULL
4. SET DEFAULT – vymaž a odkaz nastav na defaultní hodnotu
5. NO CHECK – bez kontroly

## Dekompozice

Studenti(rc, jmeno, prijmeni, univerzita, mesto, adresa\_u, stredni\_skola, oceneni)

- **Dekompozice**  
rozdělení velkých relací na lepší (z pohledu anomálií)  
menší relace, přičemž zůstanou zachyceny stejné  
informace.

Studenti(rc, jmeno, prijmeni, stredni\_skola)

Prihlasky(rc, univerzita)

Univerzity(univerzita, mesto, adresa\_u)

Oceneni(rc, oceneni)

Normalizace je technika:

- dekompozice na základě vlastností dat (vztahů, závislostí mezi atributy),
  - funkční závislosti (2NF, 3NF)
  - vícehodnotové závislosti (4NF, 5NF)
- kdy získám menší relace, které splňují normální formy
  - omezují redundanci a nekonzistenci dat
  - zachycují stejné informace

Normalizované schéma je

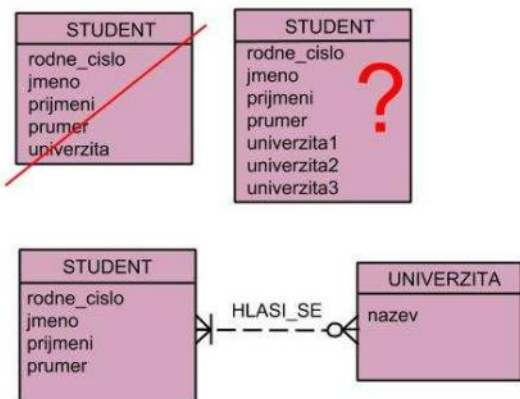
- snadno udržovatelné
- zabírá méně místa - možnost komprese
- vhodné zejména, když se data často mění (hodně UPDATE)
- ! má větší počet tabulek než nenormalizované => bude v něm náročnější vyhledávání

### Normální formy

- Relace jsou postupně testovány na pravidla jednotlivých normálních forem. Když pravidlo nesplňují, relace se rozdělí na menší relace, které už pravidlo splňují.
- Pro relační model je kritická 1NF
- Všeobecně se doporučuje pokračovat alespoň po 3NF

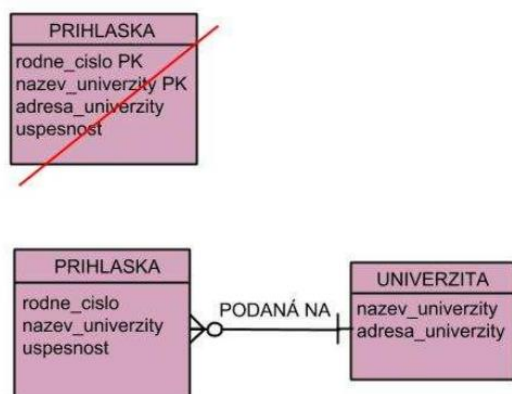
#### 1NF

- Relace je v 1. normální formě, pokud každý atribut obsahuje jen atomické hodnoty
- Relace je v 1. normální formě, pokud neobsahuje vícehodnotové atributy
- Každý student má jedno rodné číslo, jedno jméno, jedno příjmení, jeden průměr ale hlásí se na víc škol



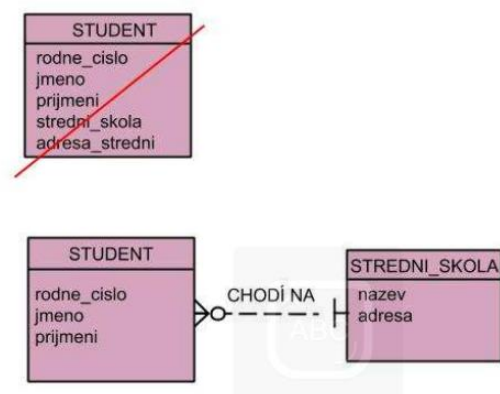
## 2NF

- Relace je v 2. normální formě, pokud je v 1. NF a každý neklíčový atribut je plně funkčně závislý na primárním klíči.



## 3NF

- Relace je v 3. normální formě, pokud je v 2. NF a všechny neklíčové atributy jsou navzájem nezávislé resp. žádný NEKLÍČOVÝ atribut není tranzitivně závislý na klíči





## Funkční závislosti (1)

### Funkční závislost (FZ)

- popisuje vztah mezi atributy v relaci
- nechť  $A$  a  $B$  jsou atributy relace  $R$ .  $B$  je funkčně závislé na  $A$  ( $A \rightarrow B$ ), když ke každé hodnotě  $A$  existuje právě jedna hodnota  $B$

$$\forall u, t \in R: t.A = u.A \Rightarrow t.B = u.B$$

- v každém řádku, kde  $A$  má stejnou hodnotu, musí být i  $B$  stejné (vztah 1:1). V každém řádku, kde  $B$  má stejnou hodnotu, nemusí být i  $A$  vždy stejné (vztah 1:M).  
přík.  $rc \rightarrow jmeno$
- $A$  a  $B$  mohou být složené z více atributů ( $\overline{A} \rightarrow \overline{B}$ ).

$$\forall u, t \in R: t[A_1, \dots, A_n] = u[A_1, \dots, A_n] \Rightarrow t[B_1, \dots, B_n] = u[B_1, \dots, B_n]$$

přík. univerzita mesto  $\rightarrow$  adresa\_u

- $A$  se nazývá **determinant**, jednoznačně určuje  $B$

## Funkční závislosti (2)

- Vychází ze specifikace požadavků
- Musí platit pro všechny  $n$ -tice v relaci

Studenti( $rc$   $A$ ,  $jmeno$   $B$ ,  $prijmeni$   $C$ ,  $stredni_skola$   $D$ ,  $adresa_s$   $E$ ,  $univerzita$   $F$ ,  $obor$   $G$ )  
přík.

- $rc \rightarrow jmeno$   $prijmeni$  ( $A \rightarrow BC$ )  
Student se stejným rodným číslem bude mít vždy stejné jméno a příjmení.
- $rc \rightarrow stredni\_skola$  ( $A \rightarrow D$ )  
Každý student může chodit pouze na jednu střední školu.
- $stredni\_skola \rightarrow adresa\_s$  ( $D \rightarrow E$ )  
Každá střední škola má vždy jednu a tu samou adresu.
- $rc$   $univerzita \rightarrow obor$  ( $AF \rightarrow G$ )  
Každý student se může hlásit na každé univerzitě pouze na jeden obor.
- Není závislost  $rc \rightarrow univerzita$   
Student se může hlásit na víc univerzit.

## Pravidla pro funkční závislosti (1)

### 1. Rozdělení (decomposition, splitting)

$$\overline{A} \rightarrow [B_1, \dots, B_n] \Rightarrow \overline{A} \rightarrow B_1, \dots, \overline{A} \rightarrow B_n$$

přík.  $rc \rightarrow jmeno$   $prijmeni \Rightarrow rc \rightarrow jmeno, rc \rightarrow prijmeni$

$$A \rightarrow BC \Rightarrow A \rightarrow B, A \rightarrow C$$

### 2. Spojení (union, combining)

$$\overline{A} \rightarrow B_1, \dots, \overline{A} \rightarrow B_n \Rightarrow \overline{A} \rightarrow [B_1, \dots, B_n]$$

přík.  $rc \rightarrow jmeno, rc \rightarrow prijmeni \Rightarrow rc \rightarrow jmeno$   $prijmeni$

$$A \rightarrow B, A \rightarrow C \Rightarrow A \rightarrow BC$$

### 3. Triviální FZ, reflexivnost

$$\overline{B} \text{ je podmnožina } \overline{A} \Rightarrow \overline{A} \rightarrow \overline{B}$$

přík.  $rc$   $univerzita \rightarrow univerzita$

$$AF \rightarrow F$$



#### 4. Rozšíření (augmentation)

$$\overline{A} \rightarrow \overline{B} \Rightarrow \overline{AC} \rightarrow \overline{BC}$$

přík.  $rc \rightarrow \text{stredni\_skola} \Rightarrow rc \text{ univerzita} \rightarrow \text{stredni\_skola univerzita}$

a podle pravidla rozdělení  $\Rightarrow rc \text{ univerzita} \rightarrow \text{stredni\_skola}$

$$A \rightarrow D \Rightarrow AF \rightarrow DF \Rightarrow AF \rightarrow D$$

když  $A \rightarrow B$ , pak  $B$  je závislé i na všech kombinacích atributů, které obsahují  $A$ .

#### 5. Tranzitivnost

$$\overline{A} \rightarrow \overline{B}, \overline{B} \rightarrow \overline{C} \Rightarrow \overline{A} \rightarrow \overline{C}$$

přík.  $rc \rightarrow \text{stredni\_skola}, \text{stredni\_skola} \rightarrow \text{adresa\_s} \Rightarrow rc \rightarrow \text{adresa\_s}$

$$A \rightarrow D, D \rightarrow E \Rightarrow A \rightarrow E$$

- Pravidlá 3., 4. a 5. – Armstrongove axiomy
- Když

$\overline{A} \rightarrow$  všechny atributy v relaci  $R$ , pak platí, že  $\overline{A}$  je kandidátní klíč relace  $R$

## Uzávěr atributu (1)

- Mějme relaci  $R$ , funkční závislosti  $FZ$  a množinu atributů  $\overline{A}$ . Najdi  $\overline{A}^+$  **uzávěr atributů**  $\overline{A}$ , t.j. všechny atributy  $B$ , pro které platí, že jsou funkčně závislé na  $\overline{A}$ . ( $\overline{A} \rightarrow B$ )
- Algoritmus určení uzávěru
  - Do uzávěru  $\overline{A}^+$  přidej všechny atributy  $\overline{A}$ .
  - Opakuj pokud je v  $\overline{A}^+$  změna {
    - Když je v  $FZ$  takové  $\overline{U} \rightarrow \overline{V}$ , že  $\overline{U}$  je v uzávěru, přidej do uzávěru i  $\overline{V}$ .}
- Když jsou v uzávěru všechny atributy relace  $R$ ,  $\overline{A}$  je kandidátní klíč  $R$ .

$$A^+ = \{A\}$$

Žádná  $FZ$  nemá na levé straně jenom  $A$ .

$$D^+ = \{D\}$$

$D \rightarrow B$  má na levé straně  $D$ , které je v  $D^+$ , proto přidám  $B$ .

$$D^+ = \{D, B\}$$

Žádná další  $FZ$  nemá na levé straně  $D$ ,  $B$  nebo jejich kombinaci.

$$AB^+ = \{A, B\}$$

$AB \rightarrow C$  má na levé straně kombinaci  $AB$ , která je v  $AB^+$ , proto přidám  $C$ .

$$AB^+ = \{A, B, C\}$$

Žádná další  $FZ$  nemá na levé straně  $A$ ,  $B$ ,  $C$  nebo jejich kombinaci.

## Klíč relace

Mějme relaci  $R(A,B,C,D,E)$  a funkční závislosti FZ  $AB \rightarrow C$ ,  $AE \rightarrow D$ ,  $D \rightarrow B$

- Zjisti, který atribut nebo skupina atributů je kandidátním klíčem relace R.

Jenom uzávěr  $AE^+$  obsahuje všechny atributy relace R, proto je kandidátním klíčem této relace.

aktualizační anomálie

## Návrhové anomálie (1)

Studenti(rc, jmeno, prijmeni, univerzita, mesto, adresa\_u, stredni\_skola, oceneni)

- Alice Malá rodné číslo 7608318951 chodila na gymnázium FXŠ, hlásí se na ČVUT v Praze, VŠE v Praze a UK v Praze. ČVUT má sídlo v Dejvicích 2, UK na Náměstí míru a VŠE na Náměstí Winstona Churchilla 4. Alice získala ocenění za matematickou a fyzikální olympiádu.
- Bob Velký rodné číslo 7810085478 se hlásí na TUL v Liberci ČVUT v Praze. TUL má sídlo na Hálkově 6. Získal ocenění ve fyzikální olympiádě na SPŠ Masarykova.

rc	jmeno	prijmeni	univerzita	mesto	adresa_u	stredni_skola	oceneni
7608318951	Alice	Malá	ČVUT	Praha	Dejvická 2	GymFXŠ	MO
7608318951	Alice	Malá	VŠE	Praha	Náměstí W. Ch. 4	GymFXŠ	FO
7608318951	Alice	Malá	UK	Praha	Náměstí míru	GymFXŠ	MO
7810085478	Bob	Velký	ČVUT	Praha	Dejvická 2	SPŠMasarykova	FO
7810085478	Bob	Velký	TUL	Liberec	Hálkova 6	SPŠMasarykova	FO

- **Aktualizační anomálie** (pro operace, které mění obsah)
  - **INSERT anomálie** - při vložení nového záznamu (n-tice) musíme upravovat všechny výskyty a znát i všechny další atributy
    - u každého nového studenta musím vložit i adresu univerzity, kde se hlásí. Když to popletu, v DB budou dvě různé adresy pro jednu univerzitu. => **nekonzistence dat**
    - když přidáme novou univerzitu, museli by jsme ostatní údaje vyplnit NULL.

- **DELETE anomálie** - při vymazání může dojít ke ztrátě dat
  - když chci vymazat záznamy o Bobovi, vymažu i informaci, že TUL sídlí na Hálkovej 6.
- **UPDATE anomálie** - při změně musíme upravovat všechny výskyty
  - když chceme změnit příjmení studenta, musíme to změnit ve všech řádcích, kde se daný student vyskytuje. Když to popletu, v DB budou dva různí studenti se stejným rodným číslem. =>  
**nekonzistence dat**