

## 33. Normalizace, normální formy, funkční závislost, aktualizací anomálie

### 33.1. Normalizace

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

### 33.2. Normální formy (NF)

Pravidla pro normalizaci databáze

Jazyk SQL je určen pro databáze, které používají tzv. relační databázový model. V něm jsou data uložena v tabulkách s jasně definovanou a relativně neměnnou strukturou. Datový záznam odpovídá řádku jedné nebo několika tabulek. Datový atribut pak průniku konkrétního sloupce a konkrétního řádku tabulky. Až na několik nepodstatných výjimek platí, že jedna buňka tabulky uchovává jednu hodnotu. Data se v tabulkách ukládají v předepsaném formátu a typu. Separací dat do samostatných tabulek se snažíme předejít redundanci dat.

- Relace jsou postupně testována 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
- 5NF -> 4NF -> BCNF -> 3NF -> 2NF -> 1NF

Normální formy:

- **1NF** Relace je v 1. normální formě, pokud neobsahuje vícehodnotové atributy a Relace je v 1. normální formě, pokud každý atribut obsahuje jen atomické hodnoty
  - 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 funkčně závislý jenom na celém primárním klíči.
  - Všechny neklíčové atributy popisují klíč, celý klíč a nic víc než klíč

- **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.
- **BCNF** Relace je v Boyce – Coddově normální formě, pokud je v 3. NF a zároveň pravidla pro 3. NF platí i pro kandidátní klíče. Relace je v BCNF tehdy a jenom tehdy, když je každý determinant FZ kandidátním klíčem.
  - Relevantní, když má relace alespoň dva složené kandidátní klíče, které mají společný atribut
- **4NF** Relace je v 4. NF pokud je ve 3. NF a neobsahuje 2 nebo více vícehodnotových údajů v řádku.

### 33.3. Funkční závislost

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
- 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  $\rightarrow$  jméno
- A a B mohou být složené z více atributů ( $A \rightarrow B$ )
- A se nazývá **determinant**, jednoznačně určuje B
- Vychází ze specifikace požadavků
- Musí platit pro všechny n-tice v relaci

Příklad: Studenti(rc A, jméno B, příjmení C, středni\_skola D, adresa\_s E, univerzita F, obor G):

- rc  $\rightarrow$  jméno příjmení ( $A \rightarrow BC$ )
  - Student se stejným rodným číslem bude mít vždy stejné jméno a příjmení.
- rc  $\rightarrow$  středni\_skola ( $A \rightarrow D$ )
  - Každý student může chodit pouze na jednu střední školu.
- středni\_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.

#### 33.3.1. Pravidla pro funkční závislosti

- Rozdělení (decomposition, splitting)
- Spojení (union, combining)
- Triviální FZ, reflexivnost
- Rozšíření (augmentation)
- Tranzitivnost

### 33.4. Aktualizační anomálie

- **Redundance dat**
  - uložené stejné data na více místech
- **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 Hálkověj 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