

Databázové systémy

Normalizácia

Vraciame sa k dátovému modelovaniu

- vieme, že existuje viacero riešení
- niektoré sú oveľa lepšie ako iné
- už poznáme databázy, takže to budeme vedieť zhodnotiť
- existuje k tomu pekná teória :)

Príklad

- Obedy v reštauráciách
 - meno a rodné číslo študenta
 - kde býval
 - reštaurácie v ktorých obedoval
 - lokalita reštaurácie
 - záľuby študenta
- Obedy
(r.č, š_meno, bydlisko, r_meno, lokalita, záľuba)

- Obedy
(r.č, š_meno, bydlisko, r_meno, lokalita, záľuba)
- Ferko s rodným číslom 123, ktorý býval v Ružomberku a Bratislave, jedol v hornej, dolnej a na prífuku, rád fotí a chodí na túry
- 123, Ferko, Ružomberok, horna, intraky, fotí
- 123, Ferko, Ružomberok, dolna, intraky, fotí
- 123, Ferko, Ružomberok, dolná, intraky, túry
- ..., ..., Bratislava, ...
- ...
-

Prečo to nie je dobrý návrh?

Prečo to nie je dobrý návrh?

- Redundancia
 - koľkokrát zachytíme rovnakú informáciu
 - že Ferko má r.č. 123
 - že rád fotí
 - že býval v Ružomberku
 - že horná je na intrákoch

Prečo to nie je dobrý návrh?

- Redundancia
 - koľkokrát zachytíme rovnakú informáciu
- Anomália aktualizácie
 - aktualizujem iba polovicu záznamov
 - aktualizujem ich rôzne
 - chodí na túry → chodí do baru

Prečo to nie je dobrý návrh?

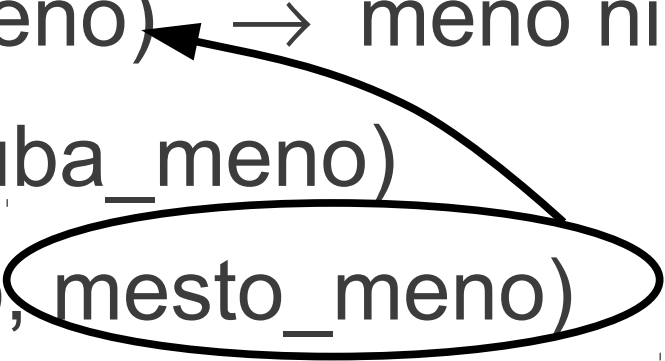
- Redundancia
 - koľkokrát zachytíme rovnakú informáciu
- Anomália aktualizácie
 - aktualizujem iba polovicu záznamov
 - aktualizujem ich rôzne
- Anomália pri vymazaní
 - neúmyselné vymazanie záznamu
 - ak nechcem záznamy o obedoch tých, ktorí radi chodia do baru, tak si vymažem komplet záznam o študentoch, ktorí majú chodenie do baru ako jediné hobby

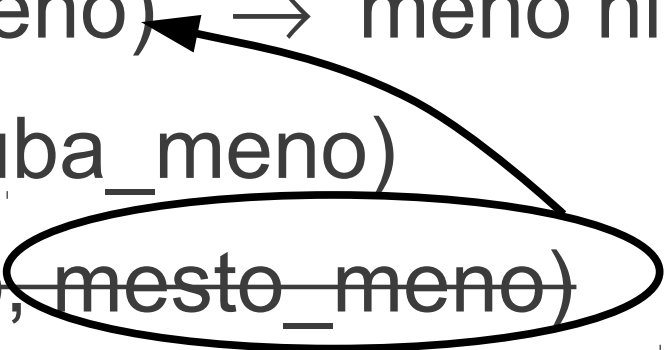
Prečo to nie je dobrý návrh?

- Redundancia
 - koľkokrát zachytíme rovnakú informáciu
- Anomália aktualizácie
 - aktualizujem iba polovicu záznamov
 - aktualizujem ich rôzne
- Anomália pri vymazaní
 - neúmyselné vymazanie záznamu
- Anomália pri vložení
 - ak chcem vložiť záznam o študentovi, musím vložiť aj atribúty o obede, ktoré nechcem/nepoznám

- Študenti(š_meno, r.č.)
- Obedy(r.č., r_meno)
- Záluby(r.č., záluba_meno)
- Lokality(r_meno, mesto_meno)
- Pobyty(r.č., mesto_meno)

- Študenti(š_meno, r.č.)
- Obedy(r.č., r_meno) → meno nie je UNIQUE
- Záluby(r.č., záluba_meno)
- Lokality(r_meno, mesto_meno)
- Pobyty(r.č., mesto_meno)

- Študenti(š_meno, r.č.)
 - Obedy(r.č., r_meno) → meno nie je UNIQUE
 - Záluby(r.č., záluba_meno)
 - Lokality(r_meno, mesto_meno)
 - Pobyty(r.č., mesto_meno)
- 
- A diagram consisting of a curved arrow pointing from the 'meno' attribute in the 'Obedy' table to the 'mesto_meno' attribute in the 'Lokality' table. The 'mesto_meno' attribute is circled, and the arrow indicates a foreign key relationship.

- Študenti(š_meno, r.č.)
 - Obedy(r.č., r_meno) → meno nie je UNIQUE
 - Záluby(r.č., záluba_meno)
 - Lokality(r_meno, mesto_meno)
 - Pobyty(r.č., mesto_meno)
- 
- A curved arrow points from the 'meno' attribute in the 'Obedy' table to the 'mesto_meno' attribute in the 'Lokality' table. The 'Lokality' table name and its attributes are circled.

Návrh pomocou dekompozície

- Začni jednou mega reláciou, ktorá obsahuje všetko, čo potrebujeme
- rozbi ju do menších relácií, ktoré nesú rovnakú informáciu
- dá sa to robiť automaticky
 - dekompozícia na základe *vlastností dát*
 - výsledok je v *normálnej forme*

Návrh pomocou dekompozície

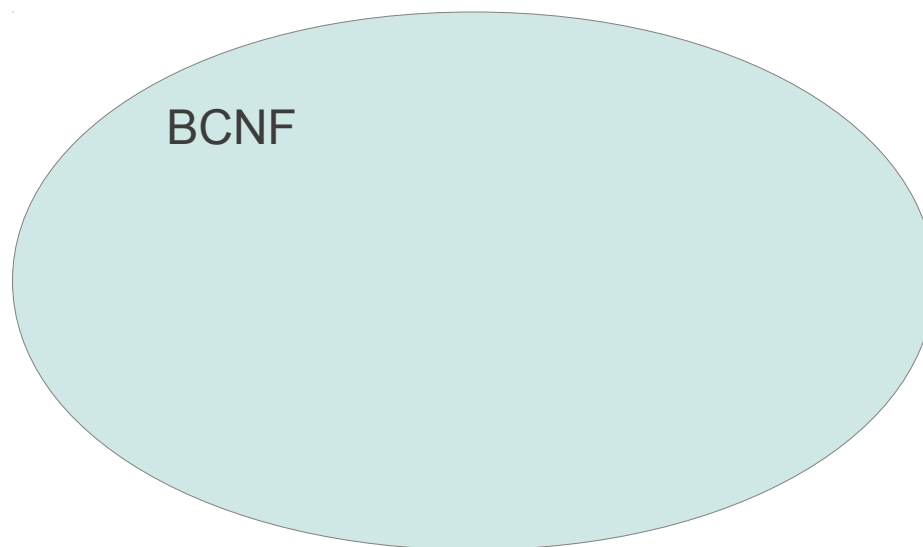
- Začni jednou mega reláciou, ktorá obsahuje všetko, čo potrebujeme
- rozbi ju do menších relácií, ktoré nesú rovnakú informáciu
- dá sa to robiť automaticky
 - dekompozícia na základe *vlastností dát*
 - výsledok je v *normálnej forme*

Vlastnosti a normálne formy

- Funkčné závislosti
 - Boyce-Codd normálna forma
- Viachodnotové závislosti
 - Štvrtá normálna forma

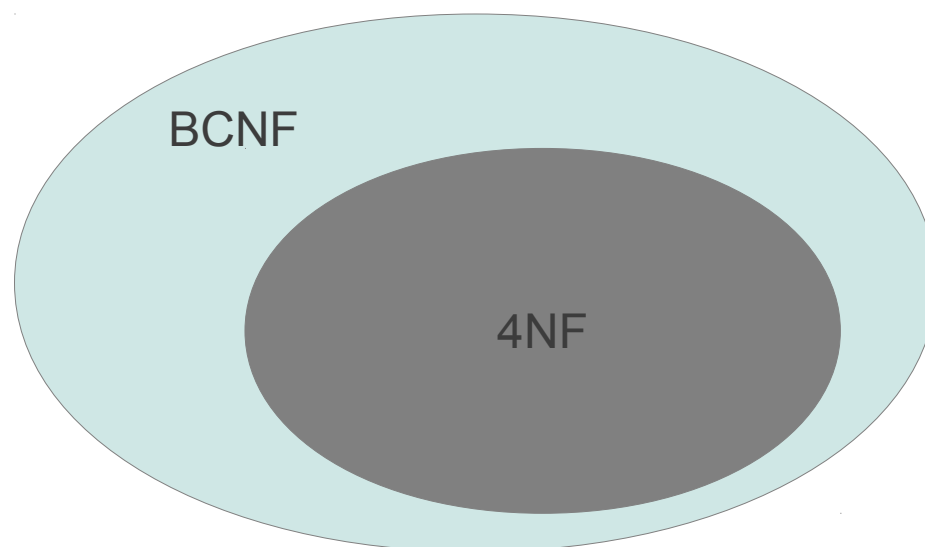
Vlastnosti a normálne formy

- Funkčné závislosti
 - Boyce-Codd normálna forma
- Viachodnotové závislosti
 - Štvrtá normálna forma



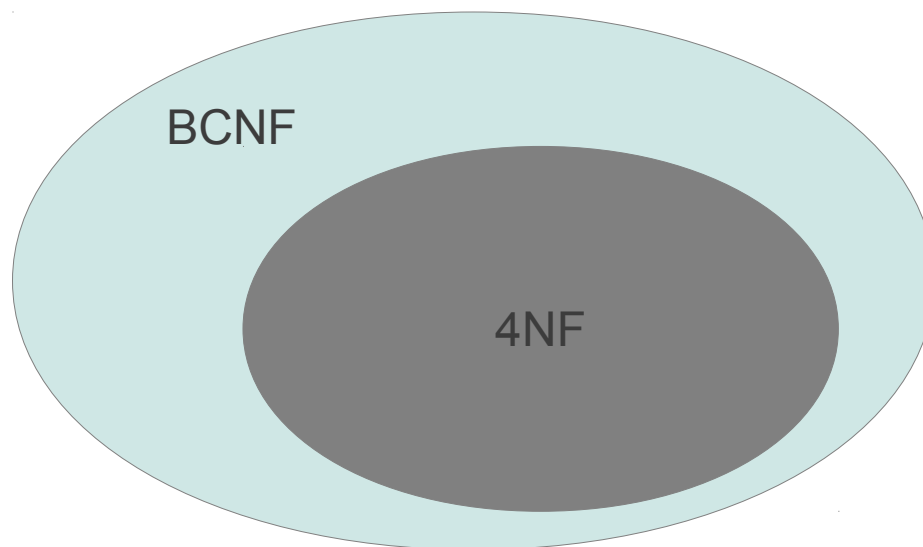
Vlastnosti a normálne formy

- Funkčné závislosti
 - Boyce-Codd normálna forma
- Viachodnotové závislosti
 - Štvrtá normálna forma



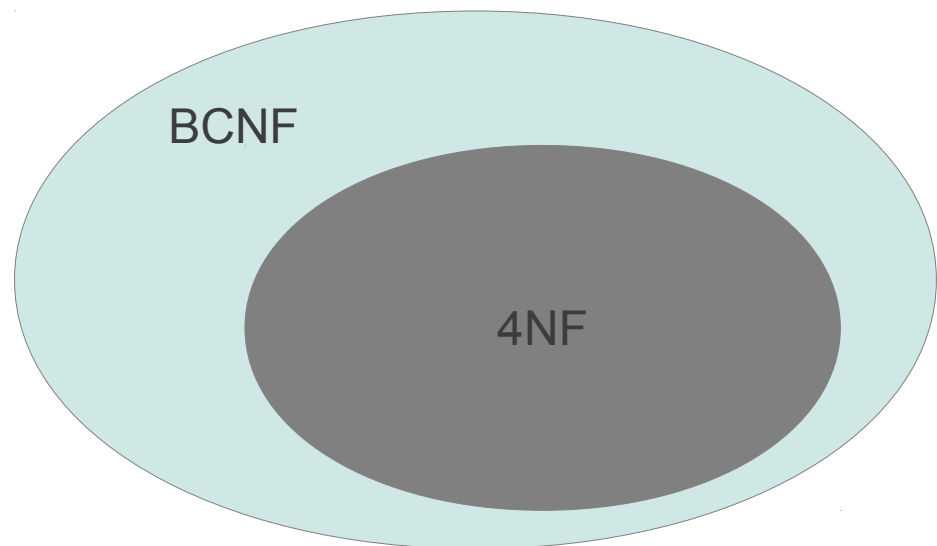
Ostatné NF

- 1. NF – dáta v reláciach (tabuľkách) s atomickými hodnotami v každej bunke



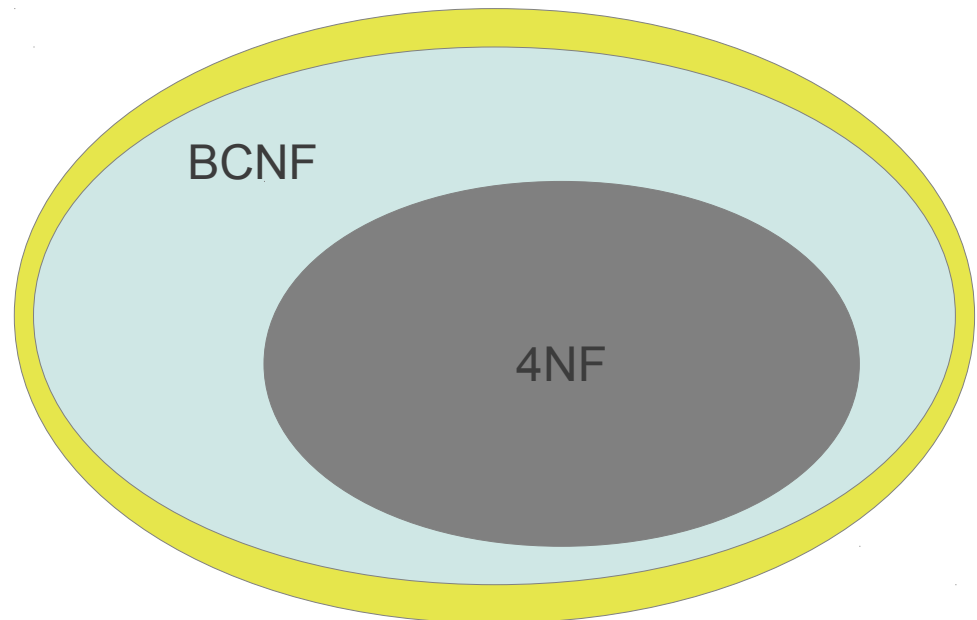
Ostatné NF

- 1. NF – dáta v reláciach (tabuľkách) s atomickými hodnotami v každej bunke
- 2. NF – každý údaj, ktorý nie je v kandidátoch PK je závislý na celom PK alebo na inom údaji, ktorý nie je kandidátom PK



Ostatné NF

- 1. NF – dáta v reláciach (tabuľkách) s atomickými hodnotami v každej bunke
- 2. NF – každý údaj, ktorý nie je v kandidátoch PK je závislý na celom PK alebo na inom údaji, ktorý nie je kandidátom PK
- 3. NF – každý údaj je závislý na PK



3. NF (wikipedia)

<u>Tournament</u>	<u>Year</u>	Winner	Winner BDAY
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Funkčné závislosti

- Obedy(r.č., š_meno, r_meno)
 - máme redundanciu
 - ukladáme si r.č. - š_meno pre každú reštauráciu, kde jedol
- Funkčná závislosť r.č. \rightarrow r_meno
 - rovnaké r.č. má vždy rovnaké š_meno
 - r.č. funkčne podmieňuje š_meno
 - naopak to nemusí platiť
 - mali by byť teda uložené samostatne
- BCNF hovorí, že ak
 - $A \rightarrow B$, tak A je kľúč

Funkčné závislosti

- Obedy(r.č., š_meno, r_meno)
 - máme redundanciu
 - ukladáme si r.č. - š_meno pre každú reštauráciu, kde jedol
- Funkčná závislosť $r.č. \rightarrow r_meno$
 - rovnaké r.č. má vždy rovnaké š_meno
 - r.č. funkčne podmieňuje š_meno
 - naopak to nemusí platiť
 - mali by byť teda uložené samostatne
- BCNF hovorí, že ak
 - $A \rightarrow B$, tak A je kľúč

Funkčné závislosti

- Obedy(r.č., š_meno, r_meno)
 - máme redundanciu
 - ukladáme si r.č. - š_meno pre každú reštauráciu, kde jedol
- Funkčná závislosť $r.č. \rightarrow r_meno$
 - rovnaké r.č. má vždy rovnaké š_meno
 - r.č. funkčne podmieňuje š_meno
 - naopak to nemusí platiť
 - mali by byť teda uložené samostatne
- BCNF hovorí, že ak
 - $A \rightarrow B$, tak A je kľúč

Funkčné závislosti

- Obedy(r.č., š_meno, r_meno) Nie je BCNF
 - máme redundanciu
 - ukladáme si r.č. - š_meno pre každú reštauráciu, kde jedol
- Funkčná závislosť $r.č. \rightarrow r_meno$
 - rovnaké r.č. má vždy rovnaké š_meno
 - r.č. funkčne podmieňuje š_meno
 - naopak to nemusí platiť
 - mali by byť teda uložené samostatne
- BCNF hovorí, že ak
 - $A \rightarrow B$, tak A je kľúč

Funkčné závislosti

- Obedy(r.č., š_meno, r_meno) Nie je BCNF
 - máme redundanciu
 - ukladáme si r.č. - š_meno pre každú reštauráciu, kde jedol
- Funkčná závislosť r.č. \rightarrow r_meno
 - rovnaké r.č. má vždy rovnaké š_meno
 - r.č. funkčne podmieňuje š_meno
 - naopak to nemusí platiť
 - mali by byť teda uložené samostatne
- BCNF hovorí, že ak
 - $A \rightarrow B$, tak A je kľúč
- Dekompozícia: Študent(r.č., š_meno), Obedy(r.č., r_meno)

Viachodnotové závislosti

- Obedy(r.č, r_name, š_mesto_meno)
 - študent, ktorý jedol v hornej to bude mať zaznamenané pre každé mesto v ktorom býval
 - ak jedol v N jedálňach a býval v M mestách, tak máme $N * M$ záznamov namiesto $N + M$
 - napriek tomu táto relácia je v BCNF
 - nemá žiadne funkčné závislosti
 - žiadne r.č. nie je asociované práve s jedným š_mesto_meno

Viachodnotové závislosti

- Obedy(r.č, r_name, š_mesto_meno)
- Viachodnotová závislosť
 - r.č. -->> š_mesto_meno
 - r.č. -->> r_name
 - r.č. má všetky kombinácie š_mesto_meno a r_name
 - napriek tomu, že tie atribúty sú nezávislé
 - každé r_name a š_mesto_meno by mali byť uložené iba raz
- 4 NF: Ak $A \twoheadrightarrow B$, tak A je kľúč

Nevýhody BCNF a 4NF

- Prihláška(r.č., škola, dátum, odbor)
 - môžem sa prihlásiť na každú školu raz na jeden odbor
 - školy majú neprekrývajúce sa dátumy prihlášok
 - FD: r.č., škola ==> dátum, odbor; dátum ==> škola
 - Key: r.č., škola
 - Nie je BCNF
 - A1(dátum, škola)
 - A2(r.č, dátum, odbor)
 - ale teraz potrebujem JOIN

Zhrnutie

- Existujú formalizované kritéria databázového návrhu
 - Normálne formy
- Existuje algoritmus ako sa z jednej mega relácie dostať do BCNF, resp. 4NF
- ...ale lepšie je rozmýšľať, aby ste nedostali príliš dekomponovaný model