

- o u *Condition* dijelu koji se navodi u HAVING dijelu naredbe dopušteno je u izrazima izvan agregatnih funkcija koristiti samo one atribute koji su navedeni u GROUP BY dijelu naredbe

```
SELECT      nazPredmet AS naziv
            ,      AVG (ocjena) AS prosjek
FROM ispit
GROUP BY naziv
HAVING matbr > 104;
```

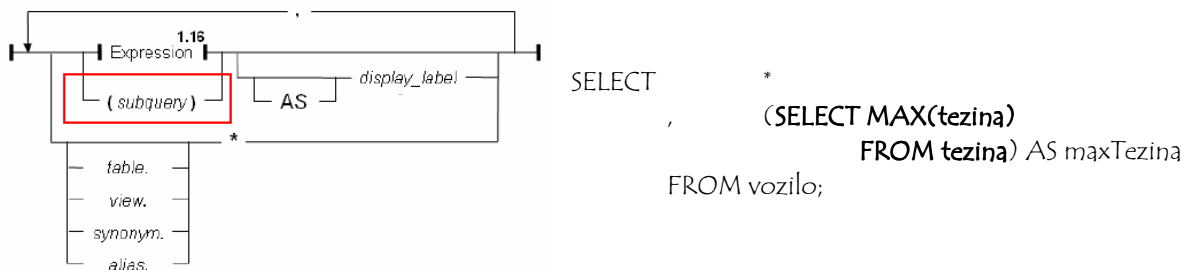
- **ORDER BY Clause** – koristi se za sortiranje rezultata upita te ukoliko se na navede smjer sortiranja podrazumjeva se uzlazni (ASC) smjer sortiranja
  - o mogu se koristiti i *izrazi koji nisu navedeni* u listi za selekciju
  - o jedino mjesto u SELECT naredbi u kojem je *dopušteno referencirati se* na zamjensko ime (display\_label)
  - o u jednoj SELECT naredbi može biti *najviše jedan ORDER BY* (ukoliko se koristi UNION ORDER BY ide iza zadnje SELECT naredbe)
  - o SQL standardi zahtjeva da se NULL vrijednost pri sortiranju uvijek smatra ili manjim ili većim od svih drugih vrijednosti (IBM Informix NULL tretira *kao uvijek manju vrijednost* od ostalih)
- **Redoslijed obavljanja dijelova SELECT naredbe** – FROM → WHERE → GROUP BY → HAVING → DISTINCT → UNION → ORDER BY

## 9. predavanje (1. ciklus; 24.03.2009.)

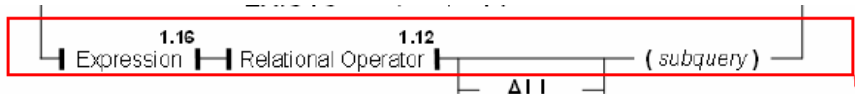
– [download slajdova](http://www.fer.hr/_download/repository/pred5.pdf) ([http://www.fer.hr/\\_download/repository/pred5.pdf](http://www.fer.hr/_download/repository/pred5.pdf))

- **Podupiti** – upit koji je ugrađen u neki drugi upit te se taj drugi upit u koji je ugrađen podupit naziva vanjski upit (*outer query*).
  - o može se ugraditi u listu selekcije (SELECT List) vanjskog upita, u uvjet (*Condition*) u WHERE dijelu vanjskog upita i u uvjet (*Condition*) u HAVING dijelu vanjskog upita.
  - o može sadržavati sve do sada spomenut SELECT naredbe osim ORDER BY
  - o u vanjski upit se može ugraditi više podupita u koje se opet može ugraditi više podupita, itd...
  - o Banek kaže da je to najteži dio gradiva definitivno ☺
- **Skalarni podupit** – najjednostavniji podupit čiji je rezultat jedna jednostavna vrijednost (skalarni) (npr. podatak tipa *cijeli broj*, *niz znakova*, *datum*, itd...), tj. relacija kardinalnosti 1 i stupnja 1 i vrijednost atributa dotične n-torke se u vanjskom upitu koristi kao skalarna vrijednost
- **Podupit u listi za selekciju (SELECT List)** – npr. *Ispisati podatke o svim vozilima ali uz njih ispisati podatak o najvećoj težini tereta*

### 1.3 SELECT List



- **Podupit u WHERE dijelu naredbe** – ako se koristi ovako dolje navedeni oblik uvjeta (točka 1.18 u službenom podsjetniku) u WHERE ili HAVING dijelu naredbe tada je dopušteno koristiti isključivo skalarni podupite (npr. *Ispisati podatke o studentima koji su rođeni u mjestu Ludbreg, a stanuju u mjestu Varaždin*)



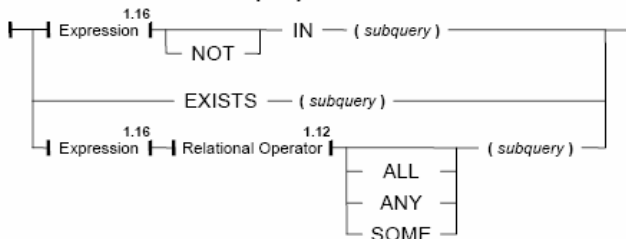
```
SELECT *
FROM vozilo
WHERE pbrRod = (SELECT pbr FROM mjesto WHERE nazMjesto = 'Ludbreg') AND
pbrSt = (SELECT pbr FROM mjesto WHERE nazMjesto = 'Varaždin');
```

- **Podupiti u HAVING dijelu naredbe** – koriste se na isti način kao i u WHERE dijelu naredbe
- **Korelirani upit** – podupit koji je koleriran s vanjskim upitom, tj. u podupitu se koriste atributi iz vanjskog upita. Najčešće se korelirani podupit mora izvršiti za svaku n-torku iz vanjskog upita.  
*Npr. Ispisati podatke o strojevima koji su ukupno korišteni više od dopuštenog broja radnih sati.*

```
SELECT oznStr, dopBrSati
FROM stroj
WHERE dopBrSati < (SELECT SUM(brSatiRada) FROM radStroja WHERE oznStr = stroj.oznStr);
```

- *rezultat upita ovisi o vrijednostima atributa vanjskog upita, tj. za svaku n-torku se dobiva drugačiji rezultat podupita*
- *upit se (logički promatrano) obavlja na sljedeći način:*
  - vanjski upit uzima jednu n-torku iz relacije *stroj* te na temelju sadržaja te n-torke i sadržaja relacija *RadStroja* u podupitu se izračunava sumu sati rada dotičnog stroja
  - ukoliko je uvjet usporedbe zadovoljen, testirana n-torka se pojavljuje u rezultatu
  - postupak se ponavlja za svaku n-torku relacije stroj
- *korištenje atributa vanjskog upita u podupitu:*
  - u vanjskom upitu se nemogu koristi atributi podupita
  - ukoliko se imena atributa vanjskog upita podudaraju s imenima atributa podupita onda se ime navedeno u podupitu odnosi na ime atributa navedeno u podupitu, a ime navedeno u vanjskom upitu na ime atributa u vanjskom upitu
  - ukoliko je potrebno razrješiti dvosmislenost dovoljno je preimenovati relaciju u vanjskom upitu ili u podupitu
- **Jednostupčani upit (Single-column subquery)** – rezultat je relacija stupnja 1 a kardinalnosti  $\geq 0$ , a koriste se u WHERE ili HAVING dijelu vanjskog upita (ne koriste se u listi za selekciju).  
*Npr. Ispisati podatke o studentima čije je prezime različito od svih prezimena nastavnika.*

#### 1.18. Condition with Subquery



```
SELECT *
FROM stud
WHERE prez <> ALL (
    SELECT prez
    FROM nastavnik);
```

*Npr. Ispisati podatke o dvoranama čiji je kapacitet veći od broja studenata u barem jednoj od grupa.*

```
SELECT *
FROM dvorana
WHERE kapacitet > SOME (SELECT brSt FROM grupa);
```

*Npr. Ispisati podatke o studentima koji su bilo koji predmet položili tijekom akademske godine 2005.*

```
SELECT *
FROM student
WHERE mbr IN (SELECT mbr FROM ispit WHERE akGod = 2005 AND ocjena > 1);
```

*Npr. Ispisati podatke o studentima koji u akademskoj godini u kojoj su upisali studij nisu položili niti jedan ispit.*

```
SELECT *
FROM student
WHERE NOT EXISTS (SELECT mbr FROM ispit WHERE ispit.mbr = student.mbr
                  AND akGod = akGodUpis
                  AND ocjena > 1;
```

*Npr. Ispisati naziv(e) predmeta s najvećim prosjekom.*

```
SELECT predmet
FROM ispit
GROUP BY predmet
HAVING AVG (ocjena) >= ALL (SELECT AVG(ocjena) FROM ispit GROUP BY predmet);
```

- **Presjek** – npr. Ispisati *studente* koji su položili i *Matematiku* i *Programiranje*

```
SELECT *
FROM položioMatem
WHERE EXISTS
(SELECT * FROM položioProgr WHERE položioProgr.mbr = položioMatem.mbr);
```

- **Razlika** – npr. *Studenti* koji su položili *Matematiku* ali nisu položili *Programiranje*

```
SELECT *
FROM položioMatem
WHERE NOT EXISTS
(SELECT * FROM položioProgr WHERE položioProgr.mbr = položioMatem.mbr);
```

## THE END 1. CIKLUSA 😊

### 10. predavanje (1. ciklus; 26.03.2009.)

– [download slajdova](http://www.fer.hr/_download/repository/pred5.pdf) ([http://www.fer.hr/\\_download/repository/pred5.pdf](http://www.fer.hr/_download/repository/pred5.pdf))

- na 1. satu smo vježbali zadatka sa prošlih MI, a na 2. satu smo obradili slajdove 45 – 66 ali pošto to gradivo ne ulazi u gradivo 1. MI onda sad neću pisati taj sažetak nego ću to napraviti u 1. tjednu 2. ciklusa kad će to biti potrebno
- INSERT – koristi se za unos jedne ili skupa n-torki u relaciju.

- Vrijednosti svih atributa unose se ili *redoslijedom kojim su atributi navedeni* (1) u CREATE TABLE naredbi ako se ne koristi lista atributa. U prvom (1) slučaju *postoje nedostaci* te se zbog toga preporuča korištenje slučaja (2):
  - a) kada se za veliki broj atributa treba vrijednost postaviti na NULL ili DEFAULT
  - b) u slučaju kada se relacijska shema promjeni, tj. promjeni se redoslijed atributa
- Ili se *u prvom dijelu naredbe navode imena atributa* (2) čije će se vrijednosti navesti u drugom dijelu naredbe istim redoslijedom. Atributi čije vrijednosti nisu navedene u drugom (2) slučaju postavljaju se na DEFAULT vrijednost ukoliko je ona definirana u CREATE TABLE naredbi ili u protivnom na NULL vrijednost.

```
CREATE TABLE mjesto (
    pbr          INTEGER
, nazMjesto    CHAR(30)
, sifZup       SMALL INT
);
INSERT INTO mjesto VALUES (42000, 'Varaždin',
INSERT INTO mjesto (pbr, sifZup, nazMjesto)
VALUES (52100, 4, 'Pula');
INSERT INTO mjesto (nazMjesto, pbr)
```

- Postoji još jedan način unosa novih n-torki, a to je *pomoću SELECT naredbe koja može sadržavati sve do sada navedene dijelove osim ORDER BY, FIRST n i UNION*. Treba paziti na redoslijed

atributa ako se ne koristi slučaj (2) iz gornjih natuknica.

*Npr. U relaciju položioFiz upisati podatke o studentima koji su položili predmet Fizika.*

```
INSERT INTO položioFiz
SELECT stud.mbr, ime, prez
FROM stud JOIN ispit ON stud.mbr = ispit.mbr
WHERE predmet = 'Fizika' AND ocjena > 1;
```

- DELETE – briše one n-torke iz relacije *mjesto* za koje se WHERE uvjet izračuna kao istinit (*true*). Naravno ako se WHERE uvjet ne navede brišu se sve n-torke iz relacije.
  - *dopušteno je koristiti sve oblike uvjeta (Condition) koji se koriste u WHERE dijelu SELECT naredbe ali u FROM dijelu nije dopušteno koristiti relaciju *mjesto**

```
DELETE FROM mjesto
WHERE pbr IN (SELECT pbr FROM mjesto WHERE nazMjesto LIKE 'V%');
```

*Npr. Iz relacije položioFiz obrisati n-torke onih studenata koji nisu položili predmet Fizika.*

```
DELETE FROM položioFiz WHERE mbr NOT IN
(SELECT mbr FROM ispit WHERE predmet = 'Fizika' AND ocjena > 1);
```

- UPDATE – mijenja vrijednost postojećih n-torki. U WHERE dijelu se opisuje kojim n-torkama će se mijenjati vrijednost (opet, ukoliko se uvjet ne navede vrijednosti se mijenjaju svim n-torkama i opet je dopušteno koristiti sve oblike uvjeta ali se u FROM dijelu ne smije koristiti relacija *ispit*, a u SET dijelu naredbe koji će atributi biti postavljeni na koje vrijednosti.

UPDATE ispit	UPDATE ispit
SET ocjena = ocjena + 1	SET (ocjena, brBod) = (ocjena + 1, NULL)
, brBod = NULL	WHERE ocjena < 5;
WHERE ocjena < 5;	

- *nova vrijednost atributa može biti definirana kao:*
  - Expression: konstante, NULL, atributi iz relacije ispit, binarni i unarni operatori, funkcije, uvjetni izraz, zagrade, ...
  - singleton select: skalarni podupit (korelirani ili nekorelirani); relaciju ispit nije dopušteno koristiti u FROM dijelu naredbe (ponovo isto pravilo)

## 11. predavanje (2. ciklus; 14.04.2009.)

– [download slajdova](http://www.fer.hr/_download/repository/pred6.pdf) ([http://www.fer.hr/\\_download/repository/pred6.pdf](http://www.fer.hr/_download/repository/pred6.pdf))

- **Naslov: Oblikovanje sheme baze podataka**
- **Par napomena na početku ciklusa:**
  - *SQL će se i dalje tražiti*, tj. da pišemo kod, pogotovo na blicovima i domaćim zadaćama, a na 2. međuispitu će od 20 bodova SQL nositi 6-7 bodova
  - *ovu temu ćemo raditi ovaj tjedan i 1/2 drugog*, a sastoji se od uvoda što smo danas napravili, normalizacije što će se raditi u četvrtak i primjera što ćemo raditi idući utorak
  - *jedan od mogućih zadataka na 2. međuispitu* je da objasnimo što je npr. pseudotranzitivnost, nešto tipa zadatka da napišemo zašto ona tablica nije bila relacija (zadatak iz 1. međuispita)
  - *na blicu će se sigurno pojaviti 1 zadatak* tipa zadatka sa slajda 30, tj. korištenjem pravila akumulacije (algoritam) dokazati neku FZ
- **Karakteristike loše koncipirane sheme baze podataka:** vidi primjer na slajdu 3
  - *redundancija* čije su posljedice:
    - *neracionalno korištenje prostora* za pohranu (na više mjesta se pojavljuje isti podatak)
    - *anomalija unosa* (ne mogu se unjeti podaci o artiklima koje nitko nije naručio niti o prodavaonicama koje nisu ništa naručile te treba pri svakom novom unosu paziti na konzistentnost podataka, tj. da uvijek jednako unesu podaci za istu prodavaonicu ili artikl)
    - *anomalija izmjene* (ako neka prodavaonica promjeni adresu treba ju promijeniti na više mjesta kako bi se zadržala konzistentnost podataka)

- *anomalija brisanja* (brisanjem svih narudžbi jednog artikla gube se podaci o artiklu)
- *pojava lažnih n-torki* (ukoliko relaciju narudžbaArtikla podijelimo u dvije relacije te ih prirodno spojimo pojaviti će se lažne n-torke)
- **Kako odrediti zamjenu za loše koncipiranu relacijsku shemu?**
  - proučavanjem *značenja podataka* (semantike, logički)
  - proučavanjem *zavisnosti među podacima*
  - *uvođenjem ograničenja* koja su ovisna o semantici podataka
  - najvažnije od svih su *funkcijske zavisnosti (FZ)*
- **Funkcija** – svakom članu 1 skupa pridružuje se maksimalno 1 član drugog skupa
- **Funkcijska zavisnost (FZ)** – Neka je  $r$  relacija sa shemom  $R$  i neka su  $X$  i  $Y$  skupovi atributa takvi da je  $X \subseteq R$  i  $Y \subseteq R$ . *Funkcijska zavisnost  $X \rightarrow Y$  vrijedi na shemi  $R$  ukoliko u svim dopuštenim stanjima relacije  $r(R)$  svaki par  $n$ -torki  $t_1$  i  $t_2$  koje imaju jednake  $X$  vrijednosti također imaju i jednake  $Y$  vrijednosti, odnosno:*  

$$t_1(x) = t_2(x) \Rightarrow t_1(y) = t_2(y)$$
  - *FZ proizlaze iz značenja podataka (semantike), a ne iz trenutnog stanja relacije*
  - *postojanje FZ nemože se dokazati pomoću postojećih podataka* u relaciji već je analizom postojećih podataka tek moguće pretpostaviti da bi FZ mogla vrijediti te *dokaz da ona vrijedi treba tražiti u značenju pojedinih atributa*
  - ako u relacijskoj shemi  $R$  vrijedi  $X \rightarrow Y$ , *relacija  $r(R)$  ne može sadržavati dvije  $n$ -torke koje imaju jednake  $X$ , a različite  $Y$  vrijednosti*
  - proučavanjem sadržaja relacije možemo samo zaključiti da *FZ ne vrijedi*
- **SQL naredba za ispitivanje bi li u relaciji eventualno mogla postojati FZ**

ispit				
mbr	sifPred	datlspit	sifNast	ocjena
101	10	30.1.2006	1003	1
101	10	15.1.2007	1002	4
102	10	30.1.2006	1001	3
103	11	15.1.2006	1002	5

SELECT \*

FROM ispit AS t1 JOIN ispit AS t2 ON t1.mbr = t2.mbr  
 WHERE t1.sifNast = t2.sifNast AND (t1.ocjena <> t2.ocjena OR t1.datlspit <> t2.datlspit);

- ako takve  $n$ -torke ne postoje onda *FZ možda vrijedi*
- **Armstrongovi aksiomi (3+3)** – projektant sheme baze podataka specifikira FZ koje su mu semantički očite, a ostale (nove) FZ se izводе koristeći Armstrongove aksiome. Neka je  $R$  relacijska shema,  $X$ ,  $Y$ ,  $Z$  i  $V$  skupovi atributa takvi da  $X \subseteq R$ ,  $Y \subseteq R$ ,  $Z \subseteq R$  i  $V \subseteq R$ 
  - *A1 – refleksivnost*
    - ako je  $Y \subseteq X$  onda  $X \rightarrow Y$
    - uvijek vrijedi  $X \rightarrow X$
  - *A2 – uvećanje*
    - ako u shemi  $R$  vrijedi  $X \rightarrow Y$  onda vrijedi i  $XZ \rightarrow Y$
    - možemo uvećati lijevu stranu funkcijske zavisnosti
  - *A3 – tranzitivnost*
    - ako u shemi  $R$  vrijedi  $X \rightarrow Y$  i  $Y \rightarrow Z$  onda vrijedi  $X \rightarrow Z$
    - $X \rightarrow Z$  je tranzitivna zavisnost
  - *P1 – pravilo unije* (pravilo o aditivnosti)
    - ako u shemi  $R$  vrijedi  $X \rightarrow Y$  i  $X \rightarrow Z$  onda vrijedi  $X \rightarrow YZ$
  - *P2 – pravilo dekompozicije* (pravilo o projektivnosti)
    - ako u shemi  $R$  vrijedi  $X \rightarrow YZ$  onda vrijedi  $X \rightarrow Y$  i  $X \rightarrow Z$
  - *P3 – pravilo o pseudotranzitivnosti*
    - ako u shemi  $R$  vrijedi  $X \rightarrow Y$  i  $VY \rightarrow Z$  onda vrijedi  $XV \rightarrow Z$
- **Pravilo o akumulaciji** – dodatno pravilo izvedeno iz Armstrongovih aksioma koje „omogućava“ *algoritamski pristup rješavanju zadataka* tipa kao na slajdu 29 gdje treba dokazati da neka tvrdnja vrijedi
  - ako u shemi  $R$  vrijedi  $X \rightarrow VZ$  i  $Z \rightarrow W$  onda vrijedi i  $X \rightarrow VZW$

- *algoritam*: vidi primjer na slajdu 31
  - *prvi korak*: A1 – refleksivnost ( $X \rightarrow X$ )
  - *u sljedećim koracima*: pomoću pravila akumulacije „uvećavati desnu stranu FZ“ sve dok lijeva ne sadrži Y
  - *zadnji korak*: pomoću P2 – dekompenzacije izdvojiti s desne strane samo Y
- **Ključ entiteta (relacije)** – skup atributa koji nedvosmisleno određuje n-torku relacije te ima svojstvo da funkcijski određuje attribute u preostalom dijelu relacije. *Definicija: Ključ relacijske sheme R je skup atributa  $K \subseteq R$  koji ima svojstvo  $K \rightarrow (R \setminus K)$ , tj. da ključ funkcijski određuje attribute u preostalom dijelu relacijske sheme i ne postoji  $K' \subseteq K$  za kojeg vrijedi  $K' \rightarrow R$ , tj. ključ je minimalan skup atributa koji funkcijski određuje attribute u preostalom dijelu relacijske sheme*
  - *entitet* – bilo što što ima suštinu ili bit i posjeduje značajke s pomoću kojih se može razlučiti od svoje okoline
  - *moгуći ključevi* (candidate key)
  - *primarni ključ* (primary key) – odabire se jedan od mogućih ključeva
  - *alternativni ključ* (alternate key) – ostali mogući ključevi
  - najbolji su ključevi kada je samo 1 atribut ali može se sastojati i od skupa atributa
- **Struktura relacije** – relacijska shema se sastoji od:
  - atributa koji su *dio ključa* (ključni atributi, ključni dio relacije)
  - atributa iz *zavisnog dijela relacije* (neključni atributi, neključni dio relacije)

## 12. predavanje (2. ciklus; 16.04.2009.)

– [download slajdova](http://www.fer.hr/_download/repository/pred7.pdf) ([http://www.fer.hr/\\_download/repository/pred7.pdf](http://www.fer.hr/_download/repository/pred7.pdf))

- **Naslov: Normalizacija**
- **Cilj normalizacije** – koristeći znanja o međusobnim funkcijskim zavisnostima atributa relacije ukloniti redundanciju (anomalije unosa, izmjene i brisanja te neracionalno korištenje prostora za pohranu) i spriječiti pojavu lažnih n-torki
- **Postupci normalizacije** – omogućavaju da se postupno, točno definiranom metodom, odredi zamjena za loše koncipiranu relacijsku shemu
- **Normalne forme temeljene na funkcijskim zavisnostima:**
  - Prva normalna forma – 1 NF
  - Druga normalna forma – 2 NF
  - Treća normalna forma – 3 NF
  - Boyce-Coddova normalna forma – BCNF
- **Postupci normalizacije:**
  - dekompozicija početne relacije (relacijske sheme) na temelju uočenih funkcijskih zavisnosti
  - sinteza skupa zadanih atributa i funkcijskih zavisnosti zadanih nad njima u relacijske sheme koje zadovoljavaju 3 NF
- **Dekompozicija relacije  $r(R)$  na relacije  $r_1(R_1), r_2(R_2), \dots, r_n(R_n)$  bez gubitaka informacije (lossless decomposition)** se događa ako vrijedi da je  $r_1(R_1) \triangleright \triangleleft r_2(R_2) \triangleright \triangleleft \dots \triangleright \triangleleft r_n(R_n) = r(R)$  odnosno  $\pi_{R_1}(r) \triangleright \triangleleft \pi_{R_2}(r) \triangleright \triangleleft \dots \triangleright \triangleleft \pi_{R_n}(r) = r(R)$
- **Relacija se razlaže na svoje dvije projekcije bez gubitaka ako:**
  - projekcije imaju zajedničke attribute
  - zajednički atributi su ključ u barem jednoj od projekcija
- **Prva normalna forma 1 NF**
  - *relacijska shema je u 1 NF ako:*
    - domene atributa sadrže samo jednostavne (nedjeljive) vrijednosti
    - vrijednost svakog atributa je samo jedna vrijednost iz domene tog atributa
    - neključni atributi relacije funkcijski ovise o ključu relacije
  - *normalizacija na 1 NF*
    - *izdvajanjem atributa u posebnu relaciju*: izdvaja se skup atributa koji se ponavlja s jednakom kratnošću zajednos ključem originalne relacije



- *promjenom ključa*: vidi primjer na slajdu 16
- **Druga normalna forma 2 NF**
  - *relacijska shema je u 2 NF ako*:
    - relacijska shema je u 1 NF
    - svaki atribut iz zavisnog dijela potpuno funkcijski ovisan o svakom ključu relacije
  - *skup atributa Y potpuno je funkcijski ovisan o skupu atributa X relacijske sheme R ako*:
    - Y funkcijski ovisi o X
    - ne postoji pravi podskup od X koji funkcijski određuje Y
  - *nepotpuna funkcijska zavisnost* – zadana je relacijska shema R i skupovi atributa X i Y iz R, tj.  $X \subseteq R$ ,  $Y \subseteq R$  i neka u R vrijedi  $FZ X \rightarrow Y$ 
    - $FZ X \rightarrow Y$  je nepotpuna ako postoji skup atributa Z koji je podskup od X za koje vrijedi  $Z \rightarrow Y$  odnosno  $FZ X \rightarrow Y$  je nepotpuna ako  $(\exists Z)(Z \subset X) : Z \rightarrow Y$
    - Primjer: Zadan je skup  $FZ F = \{ABC \rightarrow D, BC \rightarrow E, E \rightarrow D\}$ . Je li {D} potpuno funkcijski ovisan o {A, B, C}? Nije jer postoji skup  $\{B, C\} \subset \{A, B, C\}$  takav da  $\{B, C\} \rightarrow \{D\}$
  - *normalizacijom na 2 NF nastaju*:
    - relacijska shema koja sadrži skup atributa koji su nepotpuno funkcijski ovisni o ključu i dio ključa o kojem su potpuno funkcijski ovisni
    - relacijska shema koja sadrži ključ originalne relacije i skup atributa koji su potpuno funkcijski ovisni o ključu
- **Treća normalna forma 3 NF**
  - *relacijska shema je u 3 NF ako*:
    - relacijska shema je u 1 NF
    - niti jedan atribut iz zavisnog dijela nije tranzitivno funkcijski ovisan o bilo kojem ključu relacije
  - *tranzitivna funkcijska zavisnost* – skup atributa Z je tranzitivno ovisan o X ako vrijedi  $X \rightarrow Y$ ,  $Y \rightarrow X$  i  $Y \rightarrow Z$ .
  - *normalizacijom na 3 NF nastaju*: vidi slajd 27
    - relacijska shema koja sadrži skup atributa relacijske sheme OSOBA koji su tranzitivno ovisni o ključu (nazMjesto) te srednji skup atributa uočene tranzitivne zavisnosti (postBr)
    - relacijska shema koja sadrži ključ relacijske sheme OSOBA (matBr) i neključne attribute relacijske sheme OSOBA koji nisu tranzitivno ovisni o ključu
  - *komentar na 3 NF* – normalizacija na 2 NF nije nužni preduvjet za provođenje normalizacije na 3 NF jer se nepotpune FZ mogu promatrati kao tranzitivne FZ
- rješavanje primjera sa normalizacijom – vidi slajdove od 33. do 44.
- **Boyce-Coddova normalna forma BCNF**
  - *relacijska shema je u BCNF ako*:
    - relacijska shema je u 1 NF
    - niti jedan atribut nije tranzitivno funkcijski ovisan o bilo kojem ključu relacije
  - *iako je BCNF stroža od 3 NF* rijetki su slučajevi da je relacijska shema u 3 NF a nije u BCNF
  - *normalizaciju na BCNF nije potrebno provoditi* jer se smatra da je shema baze podataka ima dobra svojstva ako zadovoljava 3 NF

## 13. predavanje (2. ciklus; 21.04.2009.)

– [download slajdova](http://www.fer.hr/_download/repository/pred8.pdf) ([http://www.fer.hr/\\_download/repository/pred8.pdf](http://www.fer.hr/_download/repository/pred8.pdf))

- rješavanje primjera sa normalizacijom cijelo predavanje

## 14. predavanje (2. ciklus; 23.04.2009.)

– [download slajdova](http://www.fer.hr/_download/repository/pred9.pdf) ([http://www.fer.hr/\\_download/repository/pred9.pdf](http://www.fer.hr/_download/repository/pred9.pdf))

- **!!! POŠTO JOŠ NIJE UPLODANO 9. PREDAVANJE NA MATERIJALE KORISTIO SAM PREDAVNJE IZ A.G. 2006/2007 ALI SAM ODUSTAO JER IMAM OSJEĆAJ DA SE PREVIŠE TOGA PROMJENILO I NAPRAVIT ČU SAŽETAK KAD SE 9. PREDAVNJE IZ OVE A.G. UPLODA NA FER WEB MATERIJALE !!!**