

Baze podataka

*Katedra za računarstvo
Elektronski fakultet u Nišu*

Relacioni model podataka

Prof.dr Leonid Stoimenov

Pregled predavanja

- ▶ Istorijat relacionog modela podataka
- ▶ Strukturna komponenta
- ▶ Integritetna komponenta
- ▶ SQL - podjezik za definisanje podataka



Istorijat relacionog modela podataka

- ▶ **1970** E. Codd je predložio relacioni model

[E. Codd, "A Relational Model for Large Shared Data Banks", CACM, vol.13, No.6, June 1970]

- ▶ **1980** prve komercijalne implementacije
 - ▶ ORACLE RDBMS
 - ▶ SQL/DS system na IBM-ovom operativnom sistemu MVS
- ▶ Primeri komercijalnih RDBMS-ova
 - ▶ IBM-ova DB2 familija
 - ▶ Informix
 - ▶ Oracle
 - ▶ Sybase
 - ▶ Microsoft-ov Access i SQL Server
 - ▶ Foxbase
 - ▶ Paradox



Neke karakteristike relacionog modela podataka

- ▶ Jednostavan i elegantan
 - ▶ Baza podataka je **kolekcija više relacija**
 - ▶ Svaka relacija je **tabela** sa vrstama i kolonama
- ▶ Tabelarna reprezentacija
 - ▶ omogućava da se lako razume sadržaj baze podataka, i
 - ▶ da se za manipulaciju podacima koristi jednostavan jezik visokog nivoa SQL
- ▶ SQL obuhvata DDL, DML, SDL



Koncepti relacionog modela podataka

- ▶ Koncepti na nivou intenzije
 - ▶ Atribut
 - ▶ Šema relacije
 - ▶ Šema baze podataka
- ▶ Koncepti na nivou ekstenzije:
 - ▶ Domen atributa
 - ▶ Torka
 - ▶ Pojava relacije
 - ▶ Pojava baze podataka
- ▶ Svi ostali koncepti se izvode iz osnovnih primenom određenih formalnih – matematičkih pravila



Relacioni model – neformalni pogled

- ▶ U relacionom modelu podataka baza podataka se predstavlja kao kolekcija **relacija**
- ▶ **Relacija** je **tabela** vrednosti, koja sadrži **vrste** i **kolone**
- ▶ U terminologiji relacionog modela podataka:
 - ▶ **vrste** se nazivaju **torke**,
 - ▶ **kolone** su **atributi**,
 - ▶ **tabela** se naziva **relacija**



A diagram illustrating a table structure. The table has four columns: 'Ime', 'Prezime', 'Indeks', and 'MBR'. The rows are: 'Petar Petrović' with index '1111' and MBR '123456'; 'Milan Milanović' with index '2222' and MBR '654321'; and 'Jovan Jovanović' with index '3333' and MBR '345612'. A blue rounded rectangle highlights the 'Prezime' column and the 'Milan Milanović' row. A blue rectangle highlights the 'Milan Milanović' row. A red dashed line with an arrow points to the 'Jovan Jovanović' row.

Ime	Prezime	Indeks	MBR
Petar	Petrović	1111	123456
Milan	Milanović	2222	654321
Jovan	Jovanović	3333	345612

Relacioni model – neformalni pogled

- ▶ Svaka **vrsta** tabele predstavlja **kolekciju vrednosti podataka koje su u nekom odnosu**
 - ▶ ER model: vrsta tabele odgovara nekoj pojavi entiteta iz realnog sveta ili nekoj pojavi poveznika
- ▶ Svaka **kolona** tabele predstavlja kolekciju vrednosti podataka koja opisuje neku **odabranu osobinu** realnog sistema
 - ▶ ER model: kolona tabele odgovara nekom atributu tipa entiteta ili atributu tipa poveznika ili tipu poveznika
- ▶ Sve vrednosti u jednoj koloni pripadaju istom tipu podataka



Definicija relacije

- ▶ Neka je

$$A = \{A_1, A_2, \dots, A_n\}$$

skup atributa realnog sistema,

i neka je svakom atributu A_i pridružen domen D_i

- ▶ Domeni ne moraju biti različiti

- ▶ **Definicija relacije:**

Relacija R na skupovima D_1, D_2, \dots, D_n je skup **n-torki (torki)** od kojih svaka sadrži prvi element iz D_1 , drugi iz D_2 , ..., n-ti iz D_n

- ▶ **Domen** D se odnosi na skup **atomičnih** vrednosti

- ▶ Domeni se specificiraju preko **tipa podataka**
 - ▶ Korisno je da svaki domen ima **naziv**
-



Domen (podsetnik)

- ▶ Logička definicija domena:
 - ▶ Lokalni_telefonski_broj: skup 6-cifrenih telefonskih brojeva koji su validni na području lokalne mrežne grupe
 - ▶ Ocena_studenta: celi broj između 6 i 10
 - ▶ Kod_obrazovnog_profila: skup oznaka profila Elektronskog fakulteta: E,EEN,EMK,MIM,RI,T,US
 - ▶ Tip podataka ili format domena:
 - ▶ Lokalni_telefonski_broj: CHARACTER string oblika ccc-ccc
 - ▶ Prelazna_ocena_studenta: INTEGER broj iz opsega 6-10
 - ▶ Kod_obrazovnog_profila: CHARACTER string iz skupa {E,EEN,EMK,MIM,RI,T,US}
-



Šema relacije

- ▶ **Šema relacije** je imenovana dvojka, u oznaci $N(R, C)$,
gde je:
 - ▶ N naziv šeme relacije,
 - ▶ $R \subseteq U$ skup atributa relacije, a
 - ▶ C skup ograničenja integriteta relacije
 - ▶ N je neformalna komponenta definicije koja opisuje njenu semantiku u prirodnom jeziku (*naziv šeme relacije*)
 - ▶ Skup ograničenja integriteta C opisuje odnose između elemenata domena atributa iz R
 - ▶ Ograničava koje će se torke pojaviti u instanci ove relacije
-



Primer šeme relacije

- ▶ Posmatračemo tip entiteta STUDENT sa atributima {IND, IME, PREZIME, BPI}
- ▶ Pri čemu je uočeno da važe ograničenja
($\gamma 1$) svaki student ima broj indeksa i ne postoje dva studenta sa istim brojem indeksa
($\gamma 2$) broj položenih ispita (BPI) je veći od nule i manji od 50
- ▶ **Šema relacije** koja predstavlja model ove klase entiteta bi imala oblik:

STUDENT({IND,IME,PREZIME,BPI}, { $\gamma 1$, $\gamma 2$ })



Obeležavanje šeme relacije

- ▶ Uobičajeno je da se šema relacije obeležava sa

$R(\underline{A}_1, A_2, \dots, A_n)$

$R(\underline{A}_1:D_1, A_2:D_2, \dots, A_n:D_n)$

- ▶ R naziv (ime) relacije
- ▶ A_1, A_2, \dots, A_n lista atributa relacije
- ▶ D_1, D_2, \dots, D_n lista domena atributa relacije
- ▶ \underline{A}_1 ograničenje tipa “primarni ključ relacije”

- ▶ **Primer :**

RADNIK2 (MBR, LIME, LD, SBR)

RADNIK1 (MBR:integer, LIME:char(15), LD:real, SBR:integer)

Naziv
relacije

Nazivi atributa
relacije

Domeni
atributa



Pojava (instanca) relacije

- ▶ Pojava relacije **r** nad šemom relacije **(R,C)**, u oznaci **r(R)**, je skup n-torki **t = {t₁,t₂,...,t_k}** koje zadovoljavaju ograničenja C
 - ▶ Svaka n-torka predstavlja uređenu listu od n-vrednosti, $t = \langle v_1, v_2, \dots, v_n \rangle$, gde $v_i \in D_i$ ili $v_i = \text{null}$
 - ▶ i-ta vrednost torke t, koja odgovara atributu A_i , se obeležava sa $t[A_i]$
- ▶ **Primer:** Jedna pojava relacije **radnik** nad šemom relacije **RADNIK (MBR, LIME, LD, SBR)** je:
 - <2203,MIRA,50000,10>
 - <2817,PERA,40000,20>
 - <2932,MIRA,35000,10>
 - ▶ <2995,GOCA,18000,30>

Tabelarno predstavljanje relacije

- ▶ Relacija **r** nad šemom relacije $R(A_1, A_2, \dots, A_n)$ se predstavlja **tabelom** čije kolone odgovaraju atributima, a vrste pojedinim n-torkama vrednosti ovih atributa
 - ▶ Sve vrste moraju biti različite

- ▶ **Primer:** Relacija radnik definisana nad šemom relacije RADNIK(MBR, LIME, LD, SBR) se predstavlja tabelom
radnik

<u>MBR</u>	LIME	LD	SBR
2203	MIRA	50000	10
2817	PERA	40000	20
2932	MIRA	35000	10
2995	GOCA	18000	30

Svojstva relacije

- ▶ Šema relacije **ne** sadrži dva jednaka naziva atributa
- ▶ Redosled naziva atributa u relaciji nije bitan
- ▶ Relacija **ne** sadrži dve jednake torke
- ▶ Redosled torki u relaciji nije bitan

- ▶ **Razlog:**
 - ▶ Prva dva svojstva proizilaze iz definicije šeme relacije (skup naziva atributa, po definiciji skup ne može sadržati dva međusobno jednaka elementa)
 - ▶ Druga dva svojstva proizilaze iz definicije relacije (skup torki). Posledica promene redosleda atributa i torki u relaciji neće biti gubitak informacije sadržane u relaciji.



Šema i pojava relacione baze podatka

- ▶ **Šema relacione baze podataka** predstavlja konačan skup šema relacija baze podataka R_i ($i=1, 2, \dots, m$) i skup međurelacionih ograničenja IC
 - ▶ Označava se sa $S(R_1, R_2, \dots, R_m; IC)$
- ▶ **Pojaва (instanca) baze podataka** nad šemom $S(R_1, R_2, \dots, R_m; IC)$, je skup pojava r_i šema relacija R_i ($i=1, 2, \dots, m$), pri čemu svako r_i zadovoljava ograničenja integriteta IC
 - ▶ Označava se sa s



Primer: Šema relacije baze podataka PREDUZEĆE

PREDUZEĆE ({RADNIK, SEKTOR, PROJEKAT}; IC)

RADNIK(MBR, LIME, LD, *SBR*)

SEKTOR(SBR, SNAZIV, *SLOK*)

PROJEKAT(PBR, PNAZIV, *SBR*, *PRUK*)

- ▶ Svaka šema relacije sadrži specifikaciju primarnog ključa (podvučeni atribut u šemi relacije) i specifikaciju stranih ključeva (*italic* u šemi relacije)



Pojava baze podataka PREDUZEĆE

radnik

<u>MBR</u>	LIME	LD	SBR
2203	MIRA	50 000	10
2817	PERA	40 000	20
2932	MIRA	35 000	10
2995	GOCA	18 000	30
3305	LAZA	60 000	40
3515	JOVAN	20 000	20
3819	VLADA	65 000	30

sektor

<u>SBR</u>	SNAZIV	SLOK
10	PROIZVODNJA	NIŠ
20	PROIZVODNJA	BEOGRAD
30	INŽINJERING	NIŠ
40	RAZVOJ	NIŠ

projekat

<u>PBR</u>	PNAZIV	SBR	PRUK
100	PC	10	2203
200	HOST	20	3305
300	LAN	30	3819
400	VIPX	40	2817



Ključ relacije (1)

Definicija ključa:

Skup atributa $X \subseteq R$ predstavlja **ključ** šeme relacije (R, C) , ako za svako r važe sledeća dva uslova:

$$\mathbf{U1: (\forall u, v \in r)(u[X]=v[X] \Rightarrow u=v)}$$

$$\mathbf{U2: (\forall Y \subset X)(\neg U1)}$$

U1: definiše **jedinstvenost** vrednosti ključa

U2: definiše **minimalnost** skupa atributa ključa

- ▶ Jedna šema relacije može imati više ključeva
 - ▶ To su **ekvivalentni ključevi** ili **ključevi kandidati**
 - ▶ Jedan od ekvivalentnih ključeva se bira za **primarni ključ**

Ključ relacije – dodatni pojmovi

- ▶ **Super ključ** (važi uslov U1)
 - ▶ Atribut ili skup atributa koji na jedinstven način identifikuje torku u relaciji
- ▶ **Kandidat ključ** (važe uslovi U1 i U2)
 - ▶ Super ključ čiji nijedan pravi podskup nema svojstvo super ključa u toj relaciji



Ključ relacije – dodatni pojmovi

- ▶ **Primarni ključ** (važe uslovi U1 i U2)
 - ▶ Kandidat ključ izabran od strane projektanta baze podataka da na jedinstven način identifikuje torke relacije
 - ▶ Preko ovog ključa se najčešće vrši traženje u relaciji
 - ▶ On se koristi kao strani ključ u drugim šemama relacija

 - ▶ **Strani ključ**
 - ▶ Atribut ili skup atributa jedne relacije koji se uparuje sa ključem kandidatom druge ili iste relacije
-

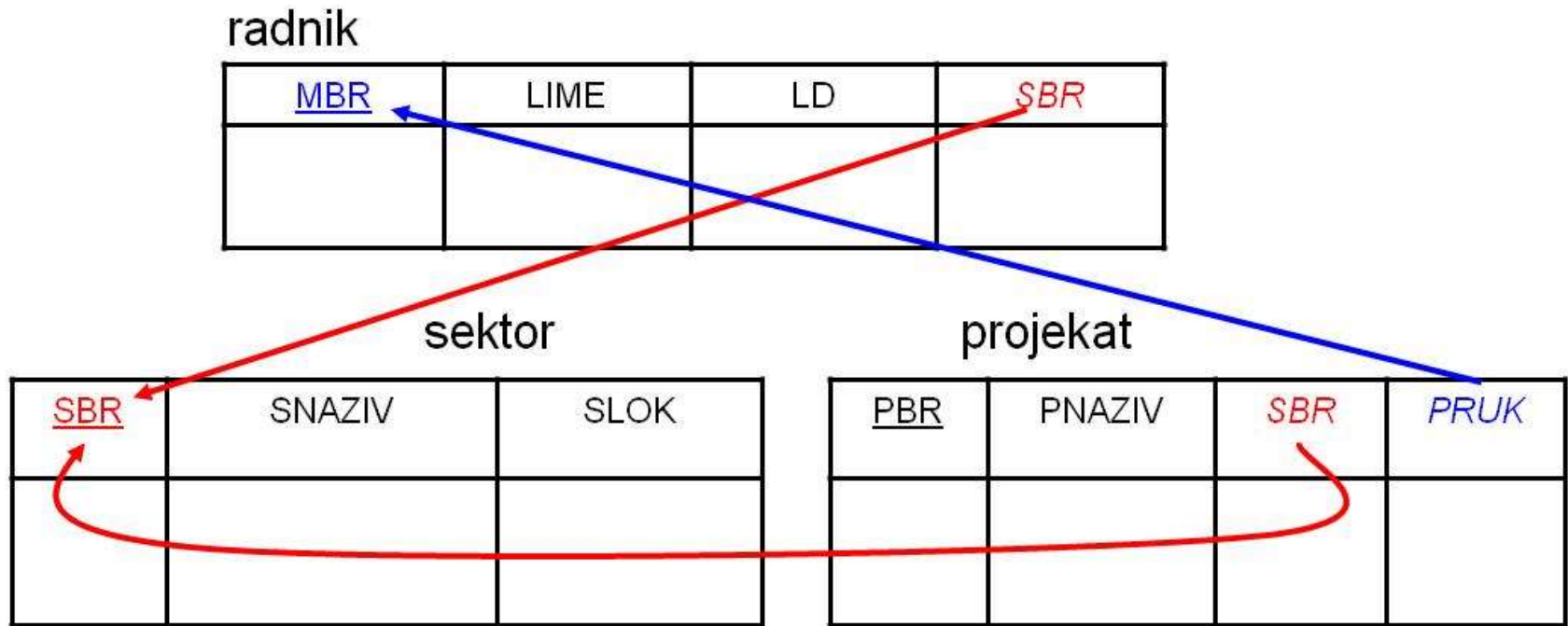


Strani (spoljni) ključ relacije

- ▶ Skup atributa SK relacije r je **strani ključ** relacije r ako zadovoljava sledeća dva uslova:
 1. Ako se relacija r referencira na relaciju q , tada atributi iz SK relacije r **moraju imati isti domen** kao atributi primarnog ključa PK relacije q
 2. Ako se torka t_1 relacije r referencira na torku t_2 relacije q , tada **vrednost stranog ključa SK u torki t_1 relacije r mora biti jednaka vrednosti primarnog ključa PK u torki t_2 relacije q ili može imati nedefinisanu (NULL) vrednost**



Strani (spoljnji) ključ relacije



Integritetna komponenta

- ▶ Koriste se pri formiranju i ažuriranju baze podataka u cilju održavanja sadržaja baze podataka u saglasnosti sa uočenim odnosima između atributa realnog sistema
 - ▶ **Ograničenja torki** – proveravaju se za svaku torku relacije
 - ▶ **Relaciona ograničenja** – proveravaju se međusobni odnosi više torki jedne relacije
 - ▶ Ovim se reguliše **lokalna konzistentnost** – usaglašenost pojave relacije sa definicijom šeme relacije
 - ▶ **Međurelaciona ograničenja**
 - ▶ Reguliše se **globalna konzistentnost** baze podataka - usaglašenost pojave baze podataka sa definicijom šeme baze podataka
-



Integritet ključeva

- ▶ Vrednosti ključeva kandidata moraju biti jedinstvene u pojavi šeme relacije
 - ▶ Ovim se definiše ograničenje jedinstvenosti
- ▶ **Razlog :**
 - ▶ Jedinstvenost vrednosti ključeva kandidata je posledica činjenice da relacija predstavlja skup torki, te stoga u relaciji ne mogu postojati dve iste torke



Integritet entiteta

- ▶ Niti vrednost primarnog ključa, niti bilo koje njegove komponente, ne sme imati nepoznatu (NULL) vrednost u pojavi relacije
- ▶ **Razlog:**
 - ▶ Pošto primarni ključ služi za identifikaciju pojedinih torki relacije, to nepoznata vrednost primarnog ključa bi onemogućila da se neke torke relacije identifikuju



Referencijalni integritet

- ▶ To je zahtev da referencirana torka mora postojati (kada semantika podataka to zahteva)
 - ▶ Važan tip referencijalnog integriteta je **ograničenje stranog ključa**
 - ▶ Ako u relaciji postoji strani ključ, tada vrednost stranog ključa mora biti jednaka vrednosti ključa kandidata neke torke referencirane relacije ili imati null vrednost
 - ▶ To je ograničenje koje se definiše između dve relacije u cilju održavanja konzistentnosti među torkama ovih relacija
 - ▶ **Razlog:**
 - ▶ Referencijalni integritet obezbeđuje da se svaka torka iz jedne relacije referencira samo na postojeću torku u drugoj ili istoj relaciji
-



„Semantički“ integritet - Opšta ograničenja

- ▶ Dodatna pravila koja specificira korisnik ili DBA, a koja definišu ili ograničavaju neke aspekte realnog sistema
- ▶ Primer
 - ▶ Broj zaposlenih u svakom sektoru preduzeća ne sme biti veći od 20
 - ▶ Plata rukovodioca mora biti veća od plata njegovih radnika
 - ▶ Ako korisnik postavi takvo ograničenje, onda on očekuje da ga DBMS proverava i da neće dozvoliti da se u relaciju sektor unese 21-vi radnik, odnosno u relaciju radnik LD radnika koji je veći od LD-a njegovog rukovodioca



Specifikacija ograničenja

- ▶ U relacijama baze podataka i između relacija baze podataka postoji veliki broj ograničenja integriteta
 - ▶ Ova ograničenja treba eksplicitno uneti u šemu baze podataka
 - ▶ U šemi relacije se eksplicitno specificiraju
 - ▶ Domeni atributa
 - ▶ Primarni ključ
 - ▶ Strani ključevi i referenca na primarni ključ
 - ▶ Da li atribut može imati NULL vrednost
 - ▶ Da li atribut mora imati jedinstvenu vrednost
 - ▶ Da li atribut ima podrazumevanu vrednost
 - ▶ Ostala semantička ograničenja (koristi se mehanizam trigera)
-



Ograničenja integriteta i DBMS

- ▶ Ograničenja integriteta se specificiraju u šemi baze podataka i eksplicitno specificiraju korišćenjem DDL-a, tako da ih DBMS može automatski nadzirati
 - ▶ Semantička ograničenja se mogu specificirati i kontrolisati unutar aplikacionih programa za ažuriranje baze podataka ili se mogu specificirati u šemi baze podataka korišćenjem jezika za specificiranje ograničenja
 - ▶ Većina komercijalnih DBMS-a podržava integritet ključa i entiteta, dok manji broj podržava referencijalni i semantički integritet
-



SQL - definisanje podataka

2

Specificiranje tipa relacije

STUDENT (Ind:INTEGER, Ime:STRING,
Adresa:STRING, Status:STRING), **Ključ: {Ind}**

CREATE TABLE STUDENT (

Ind	INTEGER,
Ime	CHAR(20),
Adresa	CHAR(50),
Status	CHAR(10))



SQL- definisanje podataka

Specificiranje ograničenja ključa (1)

STUDENT (Ind:INTEGER, Ime:STRING,
Adresa:STRING, Status:STRING), **Ključ: {Ind}**

CREATE TABLE STUDENT (
 Ind INTEGER,
 Ime CHAR(20),
 Adresa CHAR(50),
 Status CHAR(10),
 PRIMARY KEY (Ind))



SQL - definisanje podataka

Specificiranje ograničenja ključa (2)

PREDMET(KatId:STRING,
PrKod:STRING,PrNaziv:STRING, Opis:STRING),
Ključevi: {PrKod}, {KatId,PrNaziv}

CREATE TABLE PREDMET (
 KatId CHAR(6),
 PrKod CHAR(4),
 PrNaziv CHAR(20),
 Opis CHAR(100),
 PRIMARY KEY (PrKod),
 UNIQUE (KatId,PrNaziv))



SQL - definisanje podataka Specificiranje NULL i podrazumevanih vrednosti

STUDENT(Ind:INTEGER, Ime:STRING, Adresa:STRING, Status:STRING),
Ključ: {Ind}

CREATE TABLE STUDENT (
 Ind INTEGER,
 Ime CHAR(20) **NOT NULL**,
 Adresa CHAR(50),
 Status CHAR(10) **DEFAULT** 'Brucoš',

 PRIMARY KEY (Ind))

ILI

CREATE TABLE STUDENT (
 Ind INTEGER **PRIMARY KEY**,
 Ime CHAR(20) **NOT NULL**,
 Adresa CHAR(50),
 Status CHAR(10) **DEFAULT** 'Brucoš')



SQL - definisanje podataka

Specificiranje opštihograničenja

ZAPISNIK(StudId:INTEGER, PrKod:STRING, Rok:STRING,
Ocena:INTEGER), **Ključ: {StudId,PrKod,Rok}**

```
CREATE TABLE ZAPISNIK (  
  StudId      INTEGER,  
  PrKod       CHAR(6),  
  Rok         CHAR(6),  
  Ocena       INTEGER NOT NULL,  
  PRIMARY KEY (StudId,PrKod,Rok),  
  CHECK (Ocena IN ( 5,6,7,8,9,10) AND VALUE IS NOT NULL),  
  CHECK (StudId >0 AND StudId <7000) )
```



SQL - definisanje podataka

Kreiranje korisnički definisanog domena

```
CREATE DOMAIN OCENA INTEGER
```

```
CHECK (VALUE IN (5,6,7,8,9,10) AND VALUE IS NOT NULL )
```

```
CREATE DOMAIN OCENA INTEGER
```

```
CHECK (Ocena >4 AND Ocena <11 AND VALUE IS NOT NULL)
```

```
CREATE TABLE ZAPISNIK (
```

```
StudId      INTEGER,
```

```
PrKod       CHAR(6),
```

```
Rok         CHAR(6),
```

```
Ocena       OCENA,
```

```
PRIMARY KEY (StudId,PrKod,Rok),
```

```
CHECK      (StudId >0 AND StudId <7000) )
```



SQL - definisanje podataka

2

Specificiranje ograničenja stranog ključa

STUDENT(Ind, Ime, Adresa, Status)

PROFESOR(Id, Ime, KatId)

PREDMET(KatId, PrKod, PrNaziv, Opis)

ZAPISNIK(StudId, PrKod, Rok, Ocena)

IZBOR(StudId, PrKod, Semestar)

NASTAVA(ProfId, PrKod, Semestar)

CREATE TABLE ZAPISNIK (

StudId INTEGER,

PrKod CHAR(6),

Rok CHAR(6),

Ocena OCENA,

PRIMARY KEY (StudId, PrKod, Rok),

FOREIGN KEY (StudId) **REFERENCES** STUDENT(Ind),

FOREIGN KEY (PrKod) **REFERENCES** PREDMET)



Unos, brisanje i ažuriranje torki

INSERT

INTO **STUDENT**(Ind, Ime, Adresa, Status)

VALUES (11708, 'Ana', 'Nišavska 11 Pirot', 'apsolvent')

- ▶ **Mogu se izostaviti imena atributa (kolona) u klauzuli INTO,**
- ▶ **Vrednosti atributa moraju navesti u redosledu koji je definisan**
 - ▶ Šemom relacije
 - ▶ Redosledom atributa
- ▶ **Dobar stil je da se imena atributa eksplicitno navedu**



SQL - manipulacija podacima

2

Unos, **brisanje** i **ažuriranje** torki

```
DELETE  
FROM STUDENT S  
WHERE S.ime= 'Ana'
```

- ▶ Iz tabele **STUDENT** briše torku koja u koloni Ime ima vrednost Ana

```
UPDATE STUDENT S  
SET S.Status= 'poslediplomac'  
WHERE S.ind= 11708
```

- ▶ U tabeli **STUDENT** ažurira torku koja u koloni Ind ima vrednost 11708 tako što postavlja novu vrednost poslediplomac u koloni Status
-



Operacijska komponenta

- ▶ Nad relacijama je definisan skup operacija
 - ▶ **Operacije** omogućavaju **specificiranje upita** nad bazom podataka
 - ▶ **Rezultat** pretraživanja je **nova relacija** koja može biti formirana od jedne ili više relacija
- ▶ Postoje dva tipa operacija nad relacionim modelom podataka:
 - ▶ Operacije relacione algebre
 - ▶ Operacije relacionog računa
 - ▶ Nad torkama
 - ▶ Nad domenima



Relacioni model podataka

Pitanja ???