

Baze podataka

Predavanja

10. Integritet baze podataka

Svibanj, 2021.



UVOD - Integritet baze podataka

- Pojam integriteta baze podataka odnosi se na konzistentnost (ispravnost) podataka sadržanih u bazi podataka
- Integritet baze podataka može biti narušen zbog:
 - slučajne pogreške korisnika kod unosa ili izmjene podataka
 - pogreške aplikacijskog programa ili sustava
- **Integritetska ograničenja** osiguravaju konzistentnost podataka pri izmjenama podataka koje obavljaju autorizirani korisnici (odnosno, korisnici koji imaju pravo pristupa podacima i pravo izmjene podataka).
- O problemu djelovanja neautoriziranih korisnika, diverzije ili sabotaže brine poseban dio SUBP koji je zadužen za **sigurnost baze podataka**

UVOD - Shema i instanca baze podataka

- Shema baze podataka sastoji se od:

- skupa relacijskih shema

$$\mathcal{R} = \{ R_1, R_2, \dots, R_n \}$$

- i skupa integritetskih ograničenja (*integrity constraints*)

$$IC = \{ IC_1, IC_2, \dots, IC_m \}$$

- Instanca baze podataka (stanje baze podataka) definirana na shemi baze podataka $\mathcal{R} = \{ R_1, R_2, \dots, R_n \}$ je skup instanci relacija (stanja relacija)

$$r = \{ r_1(R_1), r_2(R_2), \dots, r_n(R_n) \}$$

- Ispravna instanca baze podataka je ona instanca koja zadovoljava **sva** definirana integritetska ograničenja

Primjer: definirati integritetska ograničenja za zadanu bazu podataka

stud

mbrStud	prez	ime	postBrPreb
1111	Novak	Ivan	10000
1234	Kolar	Petar	31000

pred

sifPred	nazPred	ECTS	sifOrgJed
1001	Mat-1	6	102
1002	Mat-2	6	102
1003	Fiz-1	5	101

ispit

mbrStud	sifPred	datIspr	ocj	sifNast
1111	1001	29.01.05	1	1111
1111	1001	05.02.05	1	2222
1111	1001	01.04.05	3	1111
1111	1003	03.02.05	2	3333
1111	1002	15.06.05	4	2222
1234	1001	29.01.05	3	2222

nast

sifNast	prezNast	postBr
1111	Pašić	10020
2222	Brnetić	10000
3333	Horvat	10430

mjesto

postBr	nazMjesto
10000	Zagreb
10020	Zagreb
10430	Samobor
21000	Split
31000	Osijek

orgJed

sifOrgJed	kratOrgJed	sifNadOrg
100	FER	NULL
101	ZPF	100
102	ZPM	100
103	ZOEEM	100
104	GOE	103
105	GEM	103

UVOD - Integritetska ograničenja

- **Primjer:** Shema baze podataka PODUZECE
- relacijske sheme:
 - RADNIK = { mbr, prez, ime, pbrStan, datRod, datZap }
 - $K_{\text{RADNIK}} = \{ \text{mbr} \}$
 - MJESTO = { pbr, nazMjesto }
 - $K_{\text{MJESTO}} = \{ \text{pbr} \}$
- integritetska ograničenja:
 - vrijednost atributa mbr je iz skupa cijelih brojeva iz intervala [1000, 9999]
 - vrijednost atributa pbr je iz skupa cijelih brojeva iz intervala [10000, 99999]
 - ista vrijednost atributa mbr ne smije se pojaviti u dvije ili više n-torki relacije radnik(RADNIK) - vrijednost atributa mbr je jedinstvena
 - vrijednost atributa mbr ne smije poprimiti NULL vrijednost
 - razlika između datZap (datum zaposlenja) i datRod(datum rođenja) ne smije biti manja od 16 godina niti veća od 65 godina
 - itd.

UVOD - Integritetska ograničenja

- definicije integritetskih ograničenja su sastavni dio sheme baze podataka
- definicije integritetskih ograničenja se pohranjuju u **rječnik podataka** baze podataka
 - na taj način pravila definirana integritetskim ograničenjima postaju nezaobilazna za svakog korisnika sustava
- SUBP provjerava integritetska ograničenja pri obavljanju svake operacije koja mijenja sadržaj baze podataka
 - u trenutku **završetka** operacije nad podacima, baza podataka mora biti u stanju u kojem su zadovoljena **sva** integritetska ograničenja
 - SUBP **odbija** obaviti operacije koje bi narušile neko integritetsko ograničenje, ili obavlja kompenzacijske akcije koje će u konačnici osigurati zadovoljenje svih integritetskih ograničenja

(Rječnik podataka)

- *Data dictionary, Catalogue, Repository*
- Opisi podataka (metapodaci) su pohranjeni u rječnik podataka. Prikazani su na isti način i može im se pristupiti na isti način kao i "običnim" podacima.
 - korisnici s pravom pristupa nad rječnikom podataka mogu primijeniti relacijski upitni jezik (npr. SQL)
- Rječnik podataka sadrži:
 - opis relacijskih shema
 - opis pravila integriteta
 - opis pravila pristupa - korisnik-objekt-dozvoljena akcija
 - opis pohranjenih procedura, poslovnih pravila
 - opis okidača (*triggers*)
 - ...

(Rječnik podataka) - primjer

- baza podataka u PostgreSQL SUBP sadrži nekoliko desetaka "sistemskih" tablica koje čine rječnik podataka. Te tablice se kreiraju automatski, prilikom kreiranja baze podataka
- npr, u tablicama *tables* i *columns* pohranjeni su metapodaci o relacijama i atributima

```
SELECT *  
  FROM information_schema.tables  
NATURAL JOIN information_schema.columns  
WHERE table_name = 'student'  
ORDER BY ordinal_position
```

tables				columns					
table_catalog	table_schema	table_name	...	column_name	ordinal_position	...	is_nullable	data_type	...
studadmin	public	student	...	jmbag	1	...	NO	character varying	...
studadmin	public	student	...	imestudent	2	...	NO	character varying	...
studadmin	public	student	...	prezimestudent	3	...	NO	character varying	...
studadmin	public	student	...	oib	4	...	YES	character varying	...
...

Integritetska ograničenja

- Entitetski integritet (*Entity integrity*)
- Integritet ključa (*Key integrity*)
- Domenski integritet (*Domain integrity*)
- Ograničenja NULL vrijednosti (*Constraints on NULL*)
- Referencijski integritet (*Referential integrity*)
- Opća integritetska ograničenja (*General integrity constraints*)

Entitetski integritet

- Niti jedan atribut **primarnog** ključa ne smije poprimiti NULL vrijednost

- Primjer:

$NAST = \{ \text{sifNast}, \text{OIBNast}, \text{prezNast} \}$

$PK_{NAST} = \{ \text{sifNast} \}$

$K2_{NAST} = \{ \text{OIBNast} \}$

Primarni ključ označen je s PK

$ISPIT = \{ \text{mbrStud}, \text{sifPred}, \text{datIsplit}, \text{ocj}, \text{sifNas} \}$

$PK_{ISPIT} = \{ \text{mbrStud}, \text{sifPred}, \text{datIsplit} \}$

⇒ atribut *sifNast* ne smije poprimiti NULL vrijednost niti u jednoj n-torci relacije *nast* (*NAST*)

⇒ atributi *mbrStud*, *sifPred*, *datIsplit* ne smiju poprimiti NULL vrijednost niti u jednoj n-torci relacije *ispit* (*ISPIT*)

Integritet ključa

- U relaciji ne smiju postojati dvije n-torke s jednakim vrijednostima ključa (vrijedi za **sve moguće** ključeve)

- Primjer:

$NAST = \{ \text{sifNast, OIBNast, prezNast} \}$

$PK_{NAST} = \{ \text{sifNast} \}$

$K2_{NAST} = \{ \text{OIBNast} \}$

$ISPIT = \{ \text{mbrStud, sifPred, datIsplit, ocj, sifNas} \}$

$PK_{ISPIT} = \{ \text{mbrStud, sifPred, datIsplit} \}$

- \Rightarrow u relaciji *nast* (*NAST*) ne smiju postojati dvije n-torke koje imaju jednake vrijednosti atributa *sifNast*
- \Rightarrow u relaciji *nast* (*NAST*) ne smiju postojati dvije n-torke koje imaju jednake vrijednosti atributa *OIBNast*
- \Rightarrow u relaciji *ispit* (*ISPIT*) ne smiju postojati dvije n-torke koje imaju jednake vrijednosti (istu kombinaciju vrijednosti) atributa *mbrStud*, *sifPred* i *datIsplit*

Domenski integritet

- Atribut može poprimiti samo jednu vrijednost iz domene atributa
 - Primjer: $MJESTO = \{ pbr, nazMjesto \}$
 $PK_{MJESTO} = \{ pbr \}$
 - domena atributa *pbr* je skup cijelih brojeva iz intervala [10000, 99999]
- ⇒ vrijednost atributa *pbr* u svakoj n-torki relacije *mjesto*(*MJESTO*) mora biti cijeli broj iz intervala [10000, 99999]

Ograničenja NULL vrijednosti

- Za određene atribute se može definirati ograničenje prema kojem vrijednost atributa ne smije poprimiti NULL vrijednost

- Primjer: $STUD = \{ mbrStud, imeStud, prezStud, adresaStud \}$
 $PK_{STUD} = \{ mbrStud \}$

⇒ vrijednost atributa *imeStud* ne smiju poprimiti NULL vrijednosti niti u jednoj n-torci relacije *stud(STUD)*

⇒ vrijednost atributa *prezStud* ne smiju poprimiti NULL vrijednosti niti u jednoj n-torci relacije *stud(STUD)*

atribut *mbrStud* ?

Strani ključ (*Foreign key*) i referencijski integritet

- Referencijski integritet se odnosi na konzistentnost među n-torkama dviju relacija (ili n-torkama iste relacije). Neformalno: n-torka iz jedne relacije koja se poziva (referencira) na drugu relaciju se može pozivati (referencirati) samo na postojeće n-torke (primarne ključeve) u toj relaciji

- Primjer: OSOBA = { mbr, prez, pbrStan }

$$PK_{OSOBA} = \{ mbr \}$$

$$MJESTO = \{ pbr, nazMjesto \}$$

$$PK_{MJESTO} = \{ pbr \}$$

osoba(OSOBA)	mbr	prez	pbrStan
	100	Horvat	10000
	107	Kolar	10000
	109	Novak	51000

mjesto(MJESTO)	pbr	nazMjesto
	10000	Zagreb
	51000	Rijeka

- Skup atributa { pbrStan } je **strani ključ** u relaciji *osoba* koji se poziva (referencira) na relaciju *mjesto*. Relacija *osoba* je **pozivajuća** (*referencing*), a relacija *mjesto* je **pozivana** (*referenced*) relacija

Strani ključ (*Foreign key*) i referencijski integritet

- Zadane su relacije $r(R)$ s primarnim ključem PK_R i $s(S)$ s primarnim ključem PK_S . Skup atributa FK , $FK \subseteq R$, je **strani ključ** u relaciji $r(R)$ koji se poziva na relaciju $s(S)$ ukoliko vrijedi:
 - atributi u skupu FK imaju domene jednake domenama korespondentnih atributa u skupu PK_S
 - za svaku n -torku $t_1 \in r(R)$
 - postoji n -torka $t_2 \in s(S)$ takva da je $t_2[PK_S] = t_1[FK]$
- III
- vrijednost barem jednog atributa iz $t_1[FK]$ je NULL
- Relacija $r(R)$ se naziva pozivajuća, a relacija $s(S)$ se naziva pozivana relacija
 - (relacije $r(R)$ i $s(S)$ ne moraju nužno biti različite relacije)
- Referencijski integritet se odnosi na ograničenje koje proizlazi iz definicije stranog ključa

Strani ključ (*Foreign key*) i referencijski integritet

- Primjer: OSOBA = { mbr, prez, pbrStan }
PK_{OSOBA} = { mbr }
MJESTO = { pbr, nazMjesto }
PK_{MJESTO} = { pbr }

osoba(OSOBA)	mbr	prez	pbrStan
	100	Horvat	10000
	107	Kolar	NULL
	109	Novak	31000

mjesto(MJESTO)	pbr	nazMjesto
	10000	Zagreb
	51000	Rijeka

- Skup atributa { pbrStan } je strani ključ u relaciji *osoba* koji se poziva na relaciju *mjesto*
- Relacije *osoba* i *mjesto* **ne zadovoljavaju** pravilo referencijskog integriteta jer u relaciji *osoba* postoji n-torka $t_1 = \langle 109, \text{Novak}, 31000 \rangle$ za koju u relaciji *mjesto* ne postoji odgovarajuća n-torka t_2 (s vrijednošću atributa pbr jednako 31000)
 - n-torka $\langle 107, \text{Kolar}, \text{NULL} \rangle$ **ne** narušava referencijski integritet!

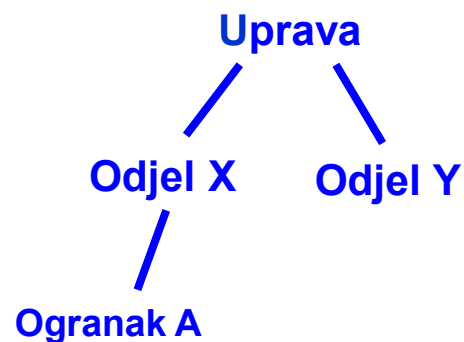
Strani ključ (*Foreign key*) i referencijski integritet

	osoba(OSOBA)				mjesto(MJESTO)			drzava(DRZAVA)	
	<u>mbr</u>	prez	oznDrz	pbr	<u>oznDrz</u>	<u>pbr</u>	nazMj	<u>oznDrz</u>	nazDrz
t ₁	101	Horvat	HR	10000	HR	10000	Zagreb	HR	Hrvatska
t ₂	102	Jones	GB	51000	GB	51000	Leeds	GB	V. Britanija
t ₃	103	Kolar	NULL	NULL	HR	51000	Rijeka		
t ₄	104	Smith	GB	NULL	GB	10000	Bristol		
t ₅	105	Novak	NULL	47000					
t ₆	106	Clark	USA	NULL					
t ₇	107	Adams	GB	47000					
t ₈	108	Wilson	USA	10000					

- $FK_1 = \{ \text{oznDrz, pbr} \}$ je strani ključ u relaciji *osoba* koji se poziva na relaciju *mjesto*
- $FK_2 = \{ \text{oznDrz} \}$ je strani ključ u relaciji *osoba* koji se poziva na relaciju *drzava*
- $FK_3 = \{ \text{oznDrz} \}$ je strani ključ u relaciji *mjesto* koji se poziva na relaciju *drzava*
- n-torke t₁, t₂, t₃, t₄, t₅ **zadovoljavaju** pravila referencijskog integriteta za FK₁ i FK₂
- n-torka t₆ ne zadovoljava pravilo referencijskog integriteta za FK₂
- n-torka t₇ ne zadovoljava pravilo referencijskog integriteta za FK₁
- n-torka t₈ ne zadovoljava pravila referencijskog integriteta za FK₁ i FK₂

Strani ključ (*Foreign key*) i referencijski integritet

- Primjer: $ORGJED = \{ \text{sifOrgjed}, \text{nazOrgjed}, \text{sifNadOrgjed} \}$
 $PK_{ORGJED} = \{ \text{sifOrgjed} \}$



orgjed(ORGJED)	sifOrgjed	nazOrgjed	sifNadOrgjed
	1	Uprava	NULL
	2	Odjel X	1
	3	Odjel Y	1
	4	Odjel Z	6
	5	Ogranak A	2

- Skup atributa $\{ \text{sifNadOrgjed} \}$ je strani ključ u relaciji *orgjed* koji se poziva na relaciju *orgjed*
- Relacija *orgjed(ORGJED)* **ne zadovoljava** pravilo referencijskog integriteta jer u relaciji postoji n-torka $t_1 = \langle 4, \text{Odjel Z}, 6 \rangle$ za koju u istoj relaciji ne postoji odgovarajuća n-torka t_2 (s vrijednošću atributa *sifOrgjed* jednako 6)
 - n-torka $\langle 1, \text{Uprava}, \text{NULL} \rangle$ **ne** narušava referencijski integritet!

Strani ključ (*Foreign key*) i referencijski integritet

- Postoje slučajevi u kojima strani ključ iz $r(R)$ ne smije biti NULL. To vrijedi za slučaj kad se pravila referencijskog integriteta primjenjuju na attribute za koje su definirana neka druga pravila integriteta (npr. pravilo entitetskog integriteta, ograničenje NULL vrijednosti)
 - strani ključ relacije $r(R)$ koji je ujedno **dio primarnog ključa** relacije $r(R)$ ne smije poprimiti NULL vrijednost!
- Primjer:
 $STUD = \{ mbrStud, imeStud, prezStud \}$
 $PK_{STUD} = \{ mbrStud \}$
 $ISPIT = \{ mbrStud, sifPred, datIsplit, ocj, sifNast \}$
 $PK_{ISPIT} = \{ mbrStud, sifPred, datIsplit \}$

Skup atributa $\{ mbrStud \}$ je strani ključ u relaciji *ispit*(*ISPIT*) koji se poziva na relaciju *stud*(*STUD*), ali je ujedno dio prim. ključa u relaciji *ispit*
 \Rightarrow **atribut *mbrStud* u relaciji *ispit* ne smije poprimiti NULL vrijednost!**

Opća integritetska ograničenja

- Opća integritetska ograničenja su ograničenja općeg (generalnog) oblika
 - npr. poslovna pravila
- Primjer: ograničenje odnosa među vrijednostima atributa

$\text{RADNIK} = \{ \text{mbr}, \text{prez}, \text{ime}, \text{pbrStan}, \text{datRod}, \text{datZap} \}$
 $\text{PK}_{\text{RADNIK}} = \{ \text{mbr} \}$

⇒ razlika između vrijednosti atributa *datZap* (datum zaposlenja) i *datRod* (datum rođenja) ne smije biti manja od 16 godina niti veća od 65 godina. To integritetsko ograničenje proizlazi iz (za ovaj primjer izmišljenog) zakonskog ograničenja da se osobe mlađe od 16 godina ili starije od 65 godina ne smiju zapošljavati.

Implementacija integritetskih ograničenja u SQL-u

- integritetska ograničenja se mogu definirati
 - u okviru naredbe za kreiranje relacije ili
 - naknadnim definiranjem pomoću naredbe `ALTER TABLE`

Integritetska ograničenja - za atribut

```
CREATE TABLE tableName  
( {columnName dataType      [columnConstraint] [...] } |  
                                     tableConstraint [...] )
```

Column constraints:

```
[CONSTRAINT constraintName]  
{ NOT NULL | UNIQUE | PRIMARY KEY |  
CHECK (condition) |  
DEFAULT defaultExpr |  
REFERENCES reftable [(refcolumn)] [ ON DELETE action]  
  [ON UPDATE action] }
```

action: NO ACTION, CASCADE, SET NULL, SET DEFAULT

condition odnosi se samo na dotičnu kolonu

defaultExpr ne smije uključivati varijable, podupite
niti attribute

Integritetska ograničenja - za tablicu

```
CREATE TABLE tableName  
( {columnName dataType    [columnConstraint] [...] } |  
    tableConstraint [...] )
```

Table constraints:

```
[CONSTRAINT constraintName]  
{ UNIQUE (columnName [, ...]) |  
  PRIMARY KEY (columnName [, ...]) |  
  CHECK (condition) |  
  FOREIGN KEY (columnName [, ...])  
    REFERENCES reftable [(refcolumn [, ...])]  
    [ON DELETE action]  
    [ON UPDATE action] }
```

SQL: PRIMARY KEY

- Primjer: ISPIT = { mbrStud, sifPred, datIspit, ocj, sifNast }
 $PK_{ISPIT} = \{ mbrStud, sifPred, datIspit \}$

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred  INTEGER  
    , datIspit DATE  
    , ocj      SMALLINT  
    , sifNast  INTEGER  
    , PRIMARY KEY (  
        mbrStud, sifPred, datIspit)  
    );
```

- SUBP osigurava:
entitetski integritet i
integritet ključa

vidjeti sintaksne
dijagrame 5, 5.1 i 5.4

- entitetski integritet i integritet ključa za primarni ključ se uvijek osigurava pomoću PRIMARY KEY

Definiranje int. ograničenja pri definiciji atributa

- U slučajevima kad se integritetsko ograničenje odnosi na samo jedan atribut, može se definirati neposredno uz definiciju atributa (to vrijedi za sve vrste ograničenja)

- Primjer:

$NAST = \{ \text{sifNast}, \text{OIBNast}, \text{prezNast} \}$ $\text{nast} (NAST)$

$PK_{NAST} = \{ \text{sifNast} \}$ $K2_{NAST} = \{ \text{OIBNast} \}$

```
CREATE TABLE nast (  
    sifNast INTEGER PRIMARY KEY  
    , OIBNast VARCHAR(13)  
    , prezNast VARCHAR(40)  
);
```

vidjeti sintaksne
dijagrame 5, 5.1,
5.2 i 5.4

SQL: *UNIQUE*

- Primjer - relacijska shema: $NAST = \{ \text{sifNast, OIBNast, prezNast} \}$
- Relacija: nast (NAST) $PK_{NAST} = \{ \text{sifNast} \}$ $K2_{NAST} = \{ \text{OIBNast} \}$

```
CREATE TABLE nast (  
    sifNast INTEGER  
    , OIBNast VARCHAR(13)  
    , prezNast VARCHAR(40)  
    , PRIMARY KEY (sifNast)  
    , UNIQUE (OIBNast)  
);
```

vidjeti sintaksne dijagrame 5, 5.1 i 5.4

```
CREATE TABLE nast (  
    sifNast INTEGER PRIMARY KEY  
    , OIBNast VARCHAR(13) UNIQUE  
    , prezNast VARCHAR(40)  
    ... );
```

vidjeti sintaksne dijagrame 5, 5.1, 5.2 i 5.3

SUBP osigurava: **integritet ključa**

VAŽNO: svaki alternativni ključ mora se zaštititi UNIQUE ograničenjem.

PostgreSQL :

UNIQUE ograničenje
dozvoljava pojavu više NULL vrijednosti

- uzima u obzir **pravu prirodu NULL vrijednosti**, a ne definiciju kopije unutar skupa, odnosno, definiciju kopije n-torki (za složene ključeve)

SQL: CHECK

- Domenski integritet je djelomično osiguran samom definicijom tipa podatka za atribut
 - npr. definiranjem podatka tipa SMALLINT određena je njegova domena kao skup cijelih brojeva u intervalu -32768 do 32767
- Moguće je postići točnije određenje domene atributa:

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred   INTEGER  
    , datIspit  DATE  
    , ocj       SMALLINT  
    , sifNast   INTEGER  
    , PRIMARY KEY(mbrStud, sifPred, datIspit)  
    , CHECK (ocj BETWEEN 1 AND 5)  
);
```

- SUBP osigurava:
domenski integritet

vidjeti sintaksne
dijagrame 5, 5.1, 5.4 i
5.6

```
CREATE TABLE ispit (  
    ...  
    , ocj       SMALLINT CHECK (ocj BETWEEN 1 AND 5)  
    , sifNast   INTEGER  
);
```

vidjeti sintaksne dijagrame 5, 5.1, 5.2, 5.3 i 5.6

SQL: CHECK

- Također se može koristiti za definiranje ograničenja odnosa među vrijednostima atributa u istoj n-torci (vidjeti primjer za opće pravilo integriteta)

Primjer: na dan zaposlenja osoba mora imati barem 16 i ne više od 65 godina.

```
CREATE TABLE radnik (  
    mbr          INTEGER  
    , ime        VARCHAR(40)  
    , prez       VARCHAR(40)  
    , datRod     DATE  
    , datZap     DATE  
    , CHECK ((datRod + '16 YEARS'::INTERVAL)::DATE <= datZap  
            AND (datRod + '65 YEARS'::INTERVAL)::DATE >= datZap)  
);
```

- Ograničenje koje se tiče odnosa među vrijednostima atributa se ne može napisati neposredno uz definiciju atributa

SQL: *NOT NULL*

- Ograničenje NULL vrijednosti se postiže navođenjem rezerviranih riječi *NOT NULL* iza tipa podatka pri definiciji atributa
- Primjer: ORGJED = { sifOrgjed, nazOrgjed, sifNadOrgjed }
 PK_{ORGJED} = { sifOrgjed }

```
CREATE TABLE orgjed (  
    sifOrgjed      INTEGER  
    , nazOrgjed    VARCHAR(40) NOT NULL  
    , sifNadOrgjed INTEGER  
    , PRIMARY KEY (sifOrgjed)  
    , FOREIGN KEY (sifNadOrgjed)  
        REFERENCES orgjed (sifOrgjed)  
);
```

- Hoće li SUBP dopustiti da vrijednost atributa sifOrgjed bude NULL?
- Hoće li SUBP dopustiti da vrijednost atributa sifNadOrgjed bude NULL?

SQL: FOREIGN KEY

- Primjer:
 - primarni ključ u relaciji *stud* je { *mbrStud* }
 - primarni ključ u relaciji *pred* je { *sifPred* }
 - primarni ključ u relaciji *nast* je { *sifNast* }

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred   INTEGER  
    , datIspit  DATE  
    , ocj       SMALLINT  
    , sifNast   INTEGER  
    , PRIMARY KEY (mbrStud, sifPred, datIspit)  
    , FOREIGN KEY (mbrStud) REFERENCES stud(mbrStud)  
    , FOREIGN KEY (sifPred) REFERENCES pred(sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nast(sifNast)  
);
```

vidjeti sintaksne dijagrame
5, 5.1, 5.4 i 5.5

SUBP osigurava ref. integritet: strani ključ u relaciji *ispit* (skup atributa { *sifNast* }) poziva se na primarni ključ u relaciji *nast* (skup atributa { *sifNast* })

Podrazumijeva se da su u relacijama *stud*, *pred* i *nast* pomoću PRIMARY KEY definirana ograničenja (ent. integritet i integritet ključa)

SQL: *FOREIGN KEY*

```
CREATE TABLE ispit (  
    mbrStud  INTEGER REFERENCES stud(mbrStud)  
    , sifPred INTEGER REFERENCES pred(sifPred)  
    , datIspit  DATE  
    , ocj       SMALLINT  
    , sifNast  INTEGER REFERENCES nast(sifNast)  
    , PRIMARY KEY (mbrStud, sifPred, datIspit)  
);
```

vidjeti sintaksne dijagrame
5, 5.1, 5.2, 5.3 i 5.5

SQL: FOREIGN KEY

stud	mbrStud	prez	ime
	1111	Novak	Ivan
	1234	Kolar	Petar

pred	sifPred	nazPred
	1001	Mat-1
	1002	Mat-2
	1003	Fiz-1

nast	sifNast	prezNast
	1111	Pašić
	2222	Brnetić
	3333	Horvat

ispit	mbrStud	sifPred	datIspit	ocj	sifNast
	1111	1001	29.01.12	1	1111
	1111	1001	05.02.11	1	2222
	1111	1001	01.04.11	3	1111
	1111	1003	03.02.12	2	3333
	1111	1002	15.06.11	4	2222
	1234	1001	29.01.12	3	2222

Operacije koje bi narušile referencijski integritet:

- **unos** ispita:
 - za nepostojećeg studenta
 - iz nepostojećeg predmeta
 - kod nepostojećeg nastavnika
- **izmjene** u tablici ispit:
 - mbrStud se mijenja na neku vrijednost koja ne postoji u tablici stud
 - sifPred se mijenja na neku vrijednost koja ne postoji u tablici pred
 - sifNast se mijenja na neku vrijednost koja ne postoji u tablici nast

SUBP ODBIJA OBAVITI OVE AKCIJE !

SQL: FOREIGN KEY

stud	mbrStud	prez	ime
	1111	Novak	Ivan
	1234	Kolar	Petar

pred	sifPred	nazPred
	1001	Mat-1
	1002	Mat-2
	1003	Fiz-1

nast	sifNast	prezNast
	1111	Pašić
	2222	Brnetić
	3333	Horvat

ispit	mbrStud	sifPred	datIspit	ocj	sifNast
	1111	1001	29.01.12	1	1111
	1111	1001	05.02.11	1	2222
	1111	1001	01.04.11	3	1111
	1111	1003	03.02.12	2	3333
	1111	1002	15.06.11	4	2222
	1234	1001	29.01.12	3	2222

Operacije koje bi također narušile referencijski integritet:

- brisanje podataka o studentu koji se ispisao s fakulteta (npr. 1111 Novak Ivan) iz tablice stud
- brisanje podataka o predmetu koji više ne postoji u novom nastavnom programu
- brisanje nastavnika koji je otišao u mirovinu
- **SUBP ODBIJA OBAVITI I OVE AKCIJE !**

Primjedbe?

Željeli bismo arhivirati podatke o studentima/nastavnicima koji su napustili fakultet i izbrisati ih iz aktualne baze podataka !!!

SQL: *FOREIGN KEY*

- pri definiciji ograničenja referencijskog integriteta moguće je specificirati da li će SUBP pri pokušaju narušavanja ograničenja **brisanjem pozivane n-torke**:
 - odbiti operaciju brisanja pozivane n-torke
 - ON DELETE NO ACTION
 - obaviti operaciju brisanja pozivane n-torke, ali pri tome obaviti i **kompenzacijske akcije** koje će rezultirati time da integritetsko ograničenje u konačnici bude zadovoljeno. Moguće akcije su:
 - vrijednosti stranog ključa u n-torkama koje se pozivaju na obrisanu n-torku postaviti na NULL vrijednosti
 - ON DELETE SET NULL
 - vrijednosti stranog ključa u n-torkama koje se pozivaju na obrisanu n-torku postaviti na *default* vrijednosti
 - ON DELETE SET DEFAULT
 - obrisati pozivajuće n-torke
 - ON DELETE CASCADE

SQL: *FOREIGN KEY*

- pri definiciji ograničenja referencijskog integriteta također je moguće specificirati da li će SUBP pri pokušaju narušavanja ograničenja **izmjenom primarnog ključa u pozivanoj n-torci**:
 - odbiti operaciju izmjene pozivane n-torke
 - ON UPDATE NO ACTION
 - obaviti operaciju izmjene pozivane n-torke, ali pri tome obaviti i **kompensacijske akcije** koje će rezultirati time da integritetsko ograničenje u konačnici bude zadovoljeno. Moguće akcije su:
 - vrijednosti stranog ključa u n-torkama koje se pozivaju na izmijenjenu n-torku postaviti na NULL vrijednosti
 - ON UPDATE SET NULL
 - vrijednosti stranog ključa u n-torkama koje se pozivaju na izmijenjenu n-torku postaviti na *default* vrijednosti
 - ON UPDATE SET DEFAULT
 - vrijednosti stranog ključa u n-torkama koje se pozivaju na izmijenjenu n-torku postaviti na novu vrijednost primarnog ključa pozivane n-torke
 - ON UPDATE CASCADE

SQL: *FOREIGN KEY*

- Različite reakcije na pokušaj narušavanja referencijskog integriteta brisanjem/izmjenom pozivanih n-torki:
 - odbijanje operacije (za strani ključ sifPred)
 - obavljanje kompenzacijskih akcija
 - uz kaskadno brisanje (za strani ključ mbrStud)
 - uz kaskadnu izmjenu(za strani ključ mbrStud)
 - uz postavljanje na NULL vrijednosti (za strani ključ sifNast) pri brisanju

```
CREATE TABLE ispit (  
    mbrStu      INTEGER  
    , sifPred   INTEGER  
    , datIspit  DATE  
    , ocj       SMALLINT  
    , sifNast   INTEGER  
    , PRIMARY KEY (mbrStud, sifPred, datIspit)  
    , FOREIGN KEY (mbrStud) REFERENCES stud (mbrStud)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES pred (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nast (sifNast)  
        ON DELETE SET NULL  
);
```

Ref. integritet definiran uz odbijanje operacije

Ako se pokušaju obrisati n-torke iz tablice pred na čije se šifre predmeta pozivaju n-torke iz tablice ispit

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred   INTEGER  
    , datIspit  DATE  
    , ocj       SMALLINT  
    , sifNast   INTEGER  
    , PRIMARY KEY (mbrStud, sifPred, datIspit)  
    , FOREIGN KEY (mbrStud) REFERENCES stud (mbrStud)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES pred (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nast (sifNast)  
        ON DELETE SET NULL  
);
```

Operacija brisanja n-torki iz tablice pred će biti odbijena - korisnik ili aplikacija će dobiti poruku o pogrešci



Ref. integritet definiran uz kaskadno brisanje

Ukoliko se pokušaju obrisati n-torke iz tablice stud na čije se matične brojeve pozivaju n-torke iz tablice ispit

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred   INTEGER  
    , datIspit  DATE  
    , ocj       SMALLINT  
    , sifNast   INTEGER  
    , PRIMARY KEY (mbrStud, sifPred, datIspit)  
    , FOREIGN KEY (mbrStud) REFERENCES stud (mbrStud)  
      ON DELETE CASCADE  
      ON UPDATE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES pred (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nast (sifNast)  
      ON DELETE SET NULL  
);
```

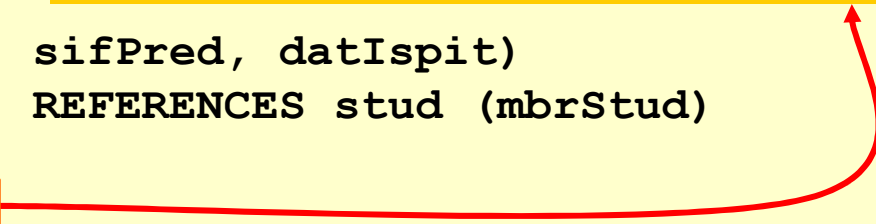
Obrisat će se n-torke iz tablice stud i sve n-torke iz tablice ispit koje se pozivaju na obrisane n-torke iz tablice stud

Ref. integritet definiran uz kaskadnu izmjenu

Ukoliko se pokušaju primijeniti vrijednosti atributa mbrStud n-torki iz tablice stud na čije se matične brojeve pozivaju n-torke iz tablice ispit

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred   INTEGER  
    , datIspit  DATE  
    , ocj       SMALLINT  
    , sifNast   INTEGER  
    , PRIMARY KEY (mbrStud, sifPred, datIspit)  
    , FOREIGN KEY (mbrStud) REFERENCES stud (mbrStud)  
      ON DELETE CASCADE  
      ON UPDATE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES pred (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nast (sifNast)  
      ON DELETE SET NULL  
);
```

Pri promjeni vrijednost atributa mbrStud u relaciji stud, promijenit će se i vrijednost atributa mbrStud svih n-torki iz tablice ispit koje se pozivaju na promijenjenu n-torku iz tablice stud

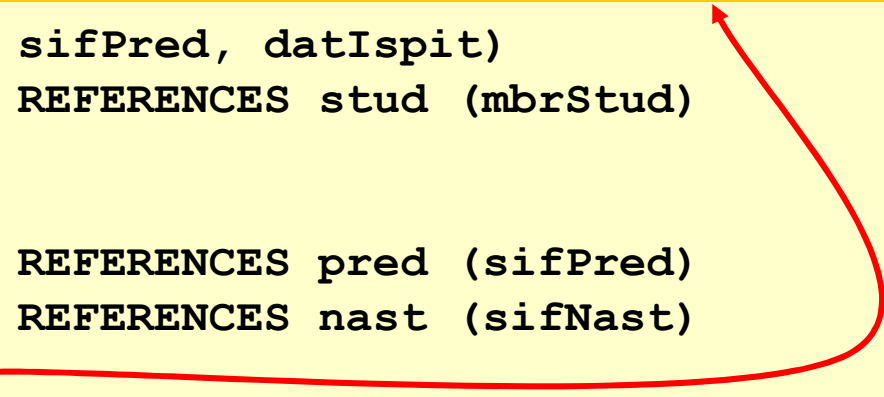


Ref. int. definiran uz postavljanje na NULL vrijednosti

Ukoliko se pokušaju obrisati n-torke iz tablice nast na čije se šifre nastavnika pozivaju n-torke iz tablice ispit

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred   INTEGER  
    , datIspit  DATE  
    , ocj       SMALLINT  
    , sifNast   INTEGER  
    , PRIMARY KEY (mbrStud, sifPred, datIspit)  
    , FOREIGN KEY (mbrStud) REFERENCES stud (mbrStud)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES pred (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nast (sifNast)  
        ON DELETE SET NULL  
);
```

Obrisat će se n-torke iz tablice nast, a vrijednosti stranog ključa (sifNast) u tablici ispit koje se pozivaju na obrisane n-torke će se postaviti na NULL



SQL: Imenovanje integritetskih ograničenja

- naziv integritetskog ograničenja (**CONSTRAINT** *constraint*) se navodi opcionalno: ako se navede, korisnik (ili aplikacija) će pri pokušaju obavljanja naredbe koja narušava integritetsko ograničenje dobiti informaciju o kojem se točno integritetskom ograničenju radi

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred   INTEGER  
    , datIspit DATE  
    , ocj       SMALLINT  
                CONSTRAINT ocjNotNull NOT NULL  
                CONSTRAINT chkOcj CHECK (ocj BETWEEN 1 AND 5)  
    , sifNast   INTEGER NOT NULL  
    , CONSTRAINT pkIspit PRIMARY KEY(mbrStud, sifPred, datIspit)  
    , CONSTRAINT fkIspitStud FOREIGN KEY(mbrStud) REFERENCES stud(mbrStud)  
    , CONSTRAINT fkIspitPred FOREIGN KEY(sifPred) REFERENCES pred(sifPred)  
    , CONSTRAINT fkIspitNast FOREIGN KEY(sifNast) REFERENCES nast(sifNast)  
);
```

- Primjer uklanjanja definiranog integritetskog ograničenja:

```
ALTER TABLE ispit DROP CONSTRAINT fkIspitStud;
```

SQL: integritet ključa

Integritet ključa može se osigurati:

- kreiranjem indeksa (*UNIQUE INDEX*) nad ključem, ili
- *UNIQUE* ograničenjem nad ključem
 - kreira se takav indeks ili ograničenje za svaki mogući ključ (ne svaki atribut ključa)

PostgreSQL :

Ako su **dozvoljene NULL vrijednosti** za attribute nad kojima postoji:

- *UNIQUE* ograničenje ili *UNIQUE INDEX*,

tada je dozvoljena višestruka pojavnost *NULL* vrijednosti - uzima se u obzir prava priroda *NULL* vrijednosti, a ne definicija kopije unutar skupa, odnosno, definicija kopije n-torki (za složene ključeve)

Primjer:

za ***UNIQUE (a)*** dozvoljene vrijednosti za **a** su npr. {1, 2, *NULL*, 5, *NULL*, ...}

za ***UNIQUE (a,b)*** dozvoljene vrijednosti za **a, b** su npr.

{< *NULL*, *NULL*>, < *NULL*, *NULL*>, <1, *NULL*>, <1, *NULL*>, <1, 1>, ...}

SQL: Napomene

- PostgreSQL automatski kreira *UNIQUE* indekse pri definiranju sljedećih ograničenja:
 - PRIMARY KEY (a, b, c)
 - SUBP automatski kreira UNIQUE INDEX za (a, b, c)
 - UNIQUE (a, b, c)
 - SUBP automatski kreira UNIQUE INDEX za (a, b, c)
- Kod definicije stranog ključa:
 - FOREIGN KEY (a, b, c) REFERENCES *reftable* (e, f, g)
 - za **reftable (e, f, g)**, koji je primarni ili alternativni ključ u tablici **reftable**, već je kreiran UNIQUE INDEX
 - Neki sustavi automatski kreiraju i indeks na strani ključ - INDEX(a, b, c),
 - PostgreSQL ne generira indeks na strani ključ – može ga se kreirati naredbom CREATE INDEX za skup atributa (a, b, c)

SQL: Napomene

- PostgreSQL: definicija referencijskog integriteta
 - **podržane su** opcije:
 - ON UPDATE NO ACTION se podrazumijeva kad nije navedena neka druga opcija
 - ON UPDATE SET NULL
 - ON UPDATE SET DEFAULT
 - ON UPDATE CASCADE
 - ON DELETE NO ACTION se podrazumijeva kad nije navedena neka druga opcija
 - ON DELETE SET NULL
 - ON DELETE SET DEFAULT
 - ON DELETE CASCADE

Kreirati integritetska ograničenja u bazi *studadmin*

```
ALTER TABLE mjesto
    ADD CONSTRAINT mjestoPK PRIMARY KEY (pbr);

ALTER TABLE student
    ADD CONSTRAINT studentPK PRIMARY KEY (jmbag),
    ADD CONSTRAINT studentCHKspol CHECK (spol IN ('M', 'Ž')) ,
    ADD CONSTRAINT studentFKpbrrod FOREIGN KEY (pbrrodstudent)
        REFERENCES mjesto(pbr),
    ADD CONSTRAINT studentFKpbrstan FOREIGN KEY (pbrstanstudent)
        REFERENCES mjesto(pbr) ;
```

Hoće li se, na postojećoj bazi podataka *studadmin*, sljedeća naredba moći uspješno izvršiti?

```
ALTER TABLE student ADD CONSTRAINT studentUQ UNIQUE (oib);
```

Objašnjenje?

Ukidanje integritetskog ograničenja – primjer:

```
ALTER TABLE student DROP CONSTRAINT studentPK
```

Za vježbu napisati i izvesti sve preostale naredbe potrebne za očuvanje integriteta baze *studadmin*.