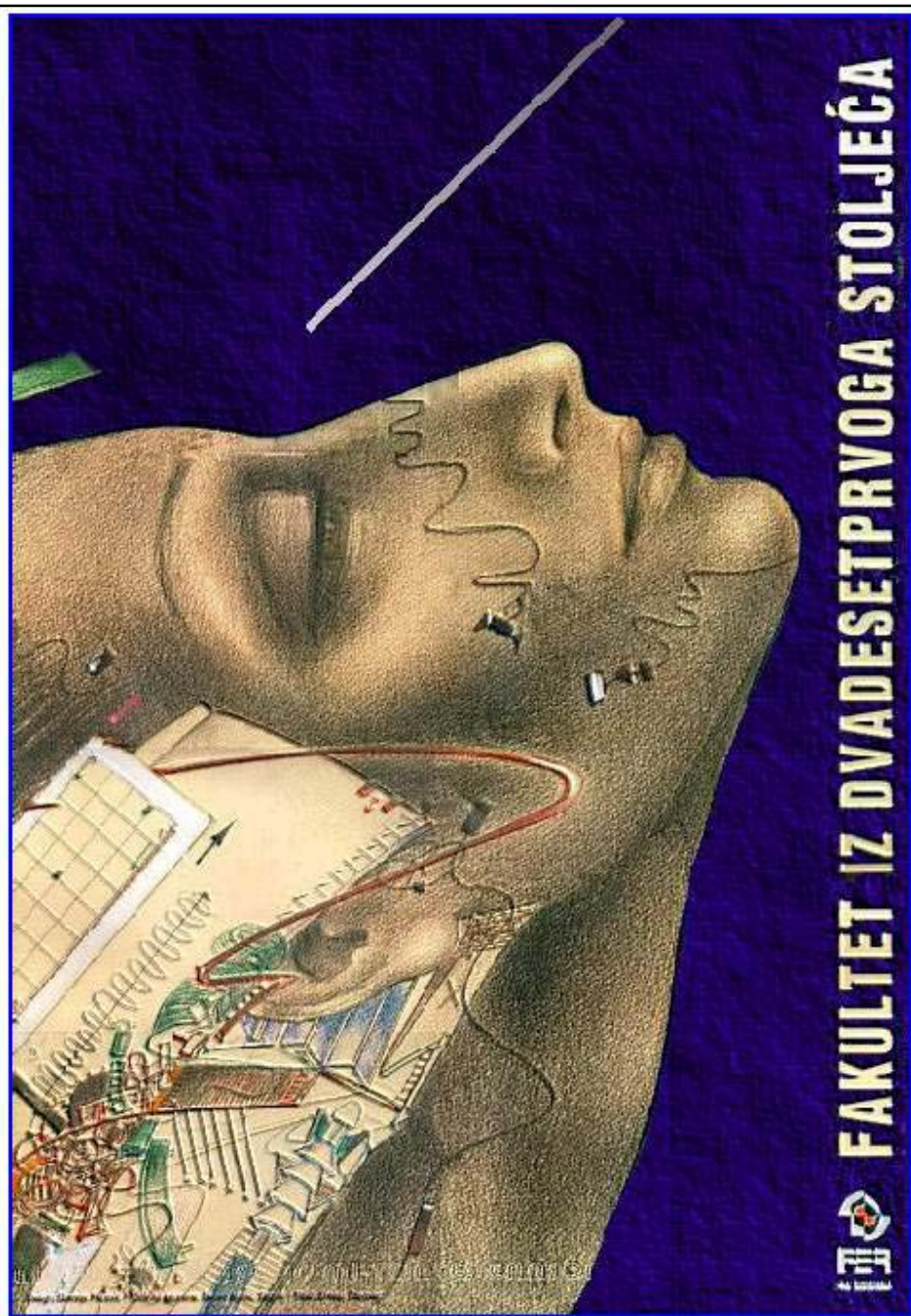


Baze podataka

Predavanja
travanj 2008.

10. Integritet baze podataka



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
 - slučajne pogreške programera ili sustava
- **Integritetska ograničenja** osiguravaju da izmjene podataka koje obavljaju autorizirani korisnici ne rezultiraju narušavanjem konzistentnosti podataka
- Integritet baze podataka može biti narušen i kao posljedica djelovanja neautoriziranih korisnika, diverzije ili sabotaže, međutim o tome 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

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 nemaju to svojstvo ili obavlja kompenzacijske akcije koje osiguravaju da su u konačnici sva integritetska ograničenja zadovoljena

(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 IBM Informix SUBP sadrži nekoliko desetaka "sistemskih" relacija koje čine rječnik podataka. Te relacije se kreiraju automatski, prilikom kreiranja baze podataka
- npr, u relacijama *systables* i *syscolumns* pohranjeni su metapodaci o relacijama i atributima

```
SELECT *  
  FROM systables, syscolumns  
 WHERE systables.tabid = syscolumns.tabid  
       AND tabname = 'mjesto'  
 ORDER BY colno;
```

systables

syscolumns

tabname	owner	tabid	ncols	nrows	created	...	colname	tabid	colno	coltype	collength	...
mjesto	bpadmin	101	3	275	15.02.2007	...	pbr	101	1	258	4	...
mjesto	bpadmin	101	3	275	15.02.2007	...	nazmjesto	101	2	271	40	...
mjesto	bpadmin	101	3	275	15.02.2007	...	sifzupanija	101	3	1	2	...

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:

NASTAVNIK = { *sifNast*, *jmbgNast*, *prezNast* }

$PK_{NASTAVNIK} = \{ sifNast \}$

$K2_{NASTAVNIK} = \{ jmbgNast \}$

Primarni ključ označen je s PK

ISPIT = { *mbrStud*, *sifPred*, *datlsp*, *ocj*, *sifNas* }

$PK_{ISPIT} = \{ mbrStud, sifPred, datlsp \}$

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

⇒ atributi *mbrStud*, *sifPred*, *datlsp* 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:

$NASTAVNIK = \{ \text{sifNast, jmbgNast, prezNast} \}$

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

$K2_{NASTAVNIK} = \{ \text{jmbgNast} \}$

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

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

- \Rightarrow u relaciji *nastavnik*(*NASTAVNIK*) ne smiju postojati dvije n-torke koje imaju jednake vrijednosti atributa *sifNast*
- \Rightarrow u relaciji *nastavnik*(*NASTAVNIK*) ne smiju postojati dvije n-torke koje imaju jednake vrijednosti atributa *jmbgNast*
- \Rightarrow u relaciji *ispit*(*ISPIT*) ne smiju postojati dvije n-torke koje imaju jednake vrijednosti (istu kombinaciju vrijednosti) atributa *mbrStud*, *sifPred* i *datlsp*

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 attribute se može definirati ograničenje prema kojem vrijednost atributa ne smije poprimiti NULL vrijednost

▪ Primjer: $STUDENT = \{ mbrStud, imeStud, prezStud, adresa \}$
 $PK_{STUDENT} = \{ mbrStud \}$

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

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

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 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**, a relacija *mjesto* je **pozivana** 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 $t_2[PK_S] = t_1[FK]$
- ili
- barem jedna vrijednost atributa iz $t_1[FK]$ je NULL vrijednost
- 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 }
 $PK_{MJESTO} = \{ pbr \}$

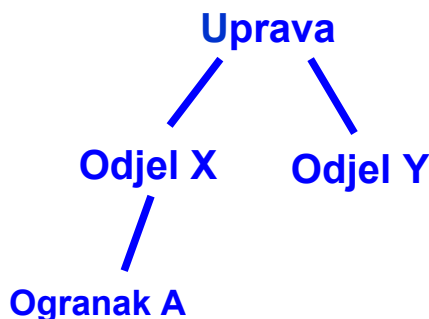
osoba(OSOBA)	mbr	prez	pbrStan
	100	Horvat	10000
	107	Kolar	NULL
	109	Novak	31000

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

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



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 jednak NULL vrijednosti. To vrijedi za slučaj kad se pravila referencijskog integriteta sukobe s pravilom entitetskog integriteta
 - 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, datlsp, ocj, sifNast }
PK_{ISPIT} = { mbrStud, sifPred, datlsp }

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*
⇒ 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

$RADNIK = \{ mbr, prez, ime, pbrStan, datRod, datZap \}$
 $PK_{RADNIK} = \{ mbr \}$

⇒ 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 [DEFAULT defaultExpr]  
  [columnConstraint] [, ...] | tableConstraint } [, ...] )
```

Column constraints:

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

action: NO ACTION, CASCADE, SET NULL, SET DEFAULT
condition - odnosi se samo na dotičnu kolonu

Integritetska ograničenja - za tablicu

```
CREATE TABLE tableName  
( { columnName dataType [DEFAULT defaultExpr]  
    [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, datIsp, ocj, sifNast }
 $PK_{ISPIT} = \{ mbrStud, sifPred, datIsp \}$

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

- SUBP osigurava:

entitetski integritet i
integritet ključa

vidjeti sintaksne
dijagrame 5, 5.1 i 5.5

- 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:

NASTAVNIK (sifNast, jmbgNast, prezNast) nastavnik(NASTAVNIK)

$PK_{NASTAVNIK} = \text{sifNast}$ $K2_{NASTAVNIK} = \text{jmbgNast}$

```
CREATE TABLE nastavnik (  
    sifNast    INTEGER PRIMARY KEY  
    , jmbgNast CHAR(13)  
    , prezNast CHAR(40)  
);
```

vidjeti sintaksne
dijagrame 5, 5.1,
5.2 i 5.4

SQL: *UNIQUE*

- Primjer:

NASTAVNIK (sifNast, jmbgNast, prezNast) nastavnik(NASTAVNIK)

$PK_{NASTAVNIK} = \text{sifNast}$ $K2_{NASTAVNIK} = \text{jmbgNast}$

```
CREATE TABLE nastavnik (  
    sifNast    INTEGER  
    , jmbgNast CHAR(13)  
    , prezNast CHAR(40)  
    , PRIMARY KEY (sifNast)  
    , UNIQUE (jmbgNast)  
);
```

- SUBP osigurava:

integritet ključa

vidjeti sintaksne
dijagrame 5, 5.1 i 5.5

```
CREATE TABLE nastavnik (  
    sifNast    INTEGER    PRIMARY KEY  
    , jmbgNast CHAR(13)    UNIQUE  
    , prezNast CHAR(40)  
    ...);
```

vidjeti sintaksne dijagrame
5, 5.1, 5.2 i 5.4

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 -32767 do 32767
- Moguće je postići točnije određenje domene atributa:

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

- SUBP osigurava:
domenski integritet

vidjeti sintaksne dijagrame
5, 5.1, 5.5 i 5.7

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

vidjeti sintaksne dijagrame 5, 5.1, 5.2, 5.4 i 5.7

SQL: CHECK

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

Primjer: razlika između vrijednosti atributa *datZap* (datum zaposlenja) i *datRod* (datum rođenja) ne smije biti manja od 16 godina niti veća od 65 godina.

```
CREATE TABLE radnik (  
    mbr          INTEGER  
    , ime        CHAR(40)  
    , prez       CHAR(40)  
    , datRod     DATE  
    , datZap     DATE  
    , CHECK (datZap - datRod >= 16*365  
            AND datZap - datRod <= 65*365)  
);
```

U rješenju je zanemareno da prestupne godine broje po 366 dana

- 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    CHAR(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 *student* je { *mbrStud* }
 - primarni ključ u relaciji *predmet* je { *sifPred* }
 - primarni ključ u relaciji *nastavnik* je { *sifNast* }

```
CREATE TABLE ispit (  
    mbrStud    INTEGER  
    , sifPred   INTEGER  
    , datIsp   DATE  
    , ocj       SMALLINT  
    , sifNast   INTEGER  
    , PRIMARY KEY (mbrStud, sifPred, datIsp)  
    , FOREIGN KEY (mbrStud) REFERENCES student(mbrStud)  
    , FOREIGN KEY (sifPred) REFERENCES predmet(sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nastavnik(sifNast)  
);
```

vidjeti sintaksne dijagrame
5, 5.1, 5.5 i 5.6

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

Podrazumijeva se da su u relacijama *student*, *predmet* i *nastavnik* pomoću PRIMARY KEY definirana ograničenja (ent. integritet i integritet ključa)

SQL: *FOREIGN KEY*

```
CREATE TABLE ispit (  
    mbrStud  INTEGER REFERENCES student(mbrStud)  
    , sifPred INTEGER REFERENCES predmet(sifPred)  
    , datIsp  DATE  
    , ocj      SMALLINT  
    , sifNast  INTEGER REFERENCES nastavnik(sifNast)  
    , PRIMARY KEY (mbrStud, sifPred, datIsp)  
);
```

vidjeti sintaksne dijagrame
5, 5.1, 5.2, 5.4 i 5.6

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 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 izmijenjenu n-torku postaviti na NULL vrijednosti
 - ON UPDATE SET NULL
 - vrijednosti stranog ključa u n-torkama koje se pozivaju na obrisanu 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*

student	mbr	prez	ime
	1111	Novak	Ivan
	1234	Kolar	Petar

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

nastavnik	sifNast	prezNast
	1111	Pašić
	2222	Brnetić
	3333	Horvat

ispit	mbr	sifPred	datlsp	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

Operacije koje bi narušile referencijski integritet:

- unos ispita za nepostojećeg studenta
- unos ispita iz nepostojećeg predmeta
- unos ispita kod nepostojećeg nastavnika
- izmjene u tablici ispit:
 - mbr se mijenja na neku vrijednost koja ne postoji u tablici student
 - sifPred se mijenja na neku vrijednost koja ne postoji u tablici predmet
 - sifNast se mijenja na neku vrijednost koja ne postoji u tablici nastavnik
- **SUBP ODBIJA OBAVITI OVE AKCIJE !!!**

Odgovara li nam to? **DA**

SQL: FOREIGN KEY

student	mbr	prez	ime
	1111	Novak	Ivan
	1234	Kolar	Petar

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

nastavnik	sifNast	prezNast
	1111	Pašić
	2222	Brnetić
	3333	Horvat

ispit	mbr	sifPred	datlsp	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

Operacije koje bi također narušile referencijski integritet:

- brisanje podataka o studentu koji se ispisao s fakulteta (npr. 1111 Novak Ivan) iz tablice student
- 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*

- Različite reakcije na pokušaj narušavanja referencijskog integriteta brisanjem pozivanih n-torki:
 - odbijanje operacije (za strani ključ `sifPred`)
 - obavljanje kompenzacijskih akcija
 - uz kaskadno brisanje (za strani ključ `mbr`)
 - uz postavljanje na NULL vrijednosti (za strani ključ `sifNast`)

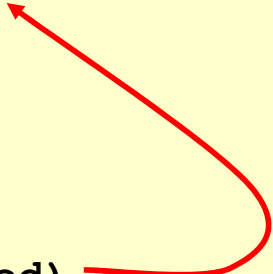
```
CREATE TABLE ispit (  
    mbr          INTEGER  
    , sifPred    INTEGER  
    , datIsp     DATE  
    , ocj        SMALLINT  
    , sifNast    INTEGER  
    , PRIMARY KEY (mbr, sifPred, datIsp)  
    , FOREIGN KEY (mbr) REFERENCES student (mbr)  
        ON DELETE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES predmet (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nastavnik (sifNast)  
        ON DELETE SET NULL  
);
```

Ref. integritet definiran uz odbijanje operacije

Ukoliko se pokušaju obrisati n-torke iz tablice predmet na čije se šifre predmeta pozivaju n-torke iz tablice ispit

```
CREATE TABLE ispit (  
    mbr          INTEGER  
    , sifPred    INTEGER  
    , datIsp     DATE  
    , ocj        SMALLINT  
    , sifNast     INTEGER  
    , PRIMARY KEY (mbr, sifPred, datIsp)  
    , FOREIGN KEY (mbr) REFERENCES student (mbr)  
        ON DELETE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES predmet (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nastavnik (sifNast)  
        ON DELETE SET NULL  
);
```

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

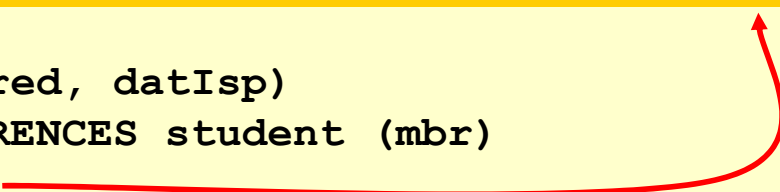


Ref. integritet definiran uz kaskadno brisanje

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

```
CREATE TABLE ispit (  
    mbr            INTEGER  
    , sifPred      INTEGER  
    , datIsp       DATE  
    , ocj          SMALLINT  
    , sifNast       INTEGER  
    , PRIMARY KEY (mbr, sifPred, datIsp)  
    , FOREIGN KEY (mbr) REFERENCES student (mbr)  
        ON DELETE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES predmet (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nastavnik (sifNast)  
        ON DELETE SET NULL  
);
```

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

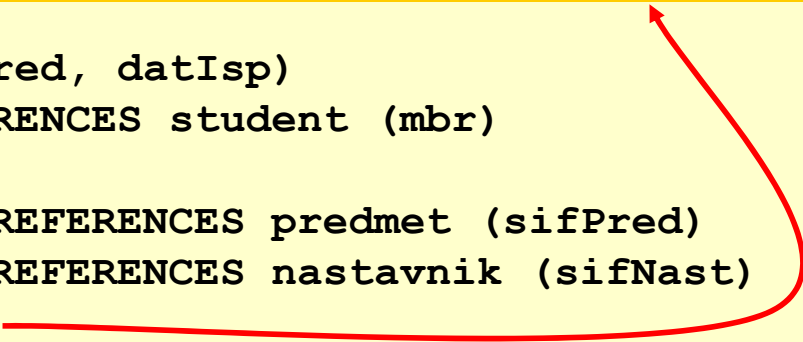


Ref. int. definiran uz postavljanje na NULL vrijednosti

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

```
CREATE TABLE ispit (  
    mbr          INTEGER  
    , sifPred    INTEGER  
    , datIsp     DATE  
    , ocj        SMALLINT  
    , sifNast    INTEGER  
    , PRIMARY KEY (mbr, sifPred, datIsp)  
    , FOREIGN KEY (mbr) REFERENCES student (mbr)  
      ON DELETE CASCADE  
    , FOREIGN KEY (sifPred) REFERENCES predmet (sifPred)  
    , FOREIGN KEY (sifNast) REFERENCES nastavnik (sifNast)  
      ON DELETE SET NULL  
);
```

Obrisat će se n-torke iz tablice nastavnik, 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  
    , datIsp    DATE  
    , ocj       SMALLINT  
                NOT NULL CONSTRAINT ocjNotNull  
                CHECK (ocj BETWEEN 1 AND 5) CONSTRAINT chkOcj  
    , sifNast    INTEGER NOT NULL  
    , PRIMARY KEY(mbrStud, sifPred, datIsp) CONSTRAINT pkIspit  
    , FOREIGN KEY(mbrStud) REFERENCES stud(mbrStud) CONSTRAINT fkIspitStud  
    , FOREIGN KEY(sifPred) REFERENCES pred(sifPred) CONSTRAINT fkIspitPred  
    , FOREIGN KEY(sifNast) REFERENCES nast(sifNast) CONSTRAINT fkIspitNast  
);
```

- Primjer uklanjanja definiranog integritetskog ograničenja:

```
ALTER TABLE ispit DROP CONSTRAINT ocjNotNull;
```

SQL: Napomene

- **Isključivo** u onim slučajevima kada SUBP ne podržava mogućnost definiranja ograničenja tipa *PRIMARY KEY* i *UNIQUE*
 - entitetski integritet se može osigurati specificiranjem ograničenja *NOT NULL* uz attribute primarnog ključa
 - integritet ključa se može osigurati kreiranjem indeksa (*UNIQUE INDEX*) nad ključem
 - po jedan takav indeks se kreira za svaki mogući ključ (ne svaki atribut ključa)
- Većina današnjih sustava za upravljanje bazama podataka podržava mogućnost definiranja tih tipova ograničenja

SQL: Napomene

- Većina sustava za upravljanje bazama podataka 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)
- Neki sustavi (npr. IBM Informix) također automatski kreiraju indekse pri definiranju ograničenja referencijskog integriteta:
 - FOREIGN KEY (a, b, c) REFERENCES *reftable* (e, f, g)
 - SUBP automatski kreira INDEX za (a, b, c)

SQL: Napomene

- IBM Informix: definicija referencijskog integriteta
 - nisu podržane opcije:
 - ON UPDATE SET NULL
 - ON UPDATE SET DEFAULT
 - ON UPDATE CASCADE
 - ON DELETE SET NULL
 - ON DELETE SET DEFAULT
 - podržane su opcije:
 - ON DELETE CASCADE
 - ON DELETE NO ACTION (ali se ne navodi jer se podrazumijeva u slučaju kad nije navedena opcija ON DELETE CASCADE)
 - ON UPDATE NO ACTION (ali se ne navodi jer se podrazumijeva)