

# Интегритет у релационим базама података

Ненад Митић

Математички факултет  
`nenad.mitic@matf.bg.ac.rs`

# Основни појмови

- Појам *интегритет* се у контексту база података односи на прецизност, пуноважност и коректност података у бази
- Одржавање интегритета података је од највеће важности за РСУБП. Због тога се у систему дефинишу правила (тзв. ограничења интегритета) која се примењују на податке

# Основни појмови (наставак)

- Интуитивно, ограничење интегритета је логички израз придружен бази за кога се захтева да његово израчунавање увек даје вредност тачно
- Ограничења се проверавају при формирању објеката у бази или мењању њиховог садржаја

# Златно правило

Ни једној операцији ажурирања није дозвољено да остави било који релвар у стању које нарушава било које од ограничења тог релвар-а.

Верзија 1: Ни једној операцији ажурирања није дозвољено да остави било коју базу података у стању у коме се неки од атрибута базе израчунава као нетацно (последица: пре било каквог стварног ажурирања проверава се важење ограничења)

# Пример

Пример ограничења интегритета: Оцена добијена на испиту мора да буде у интервалу од 5 до 10.

```
CONSTRAINT OCENA1 IS EMPTY (ISPIT WHERE  
OCENA<5 OR OCENA>10)
```

# Класификација ограничења интегритета

## Класификација према DATE-AIDB

- Ограничења стања: дефинишу прихватљива стања у бази
- Ограничења прелаза: дефинишу прихватљива стања прелаза у бази

# Ограничења стања

## Ограничења стања

- Ограничења базе: ограничења која се односе на вредности које је дозвољено чувати у бази (тј. које се односе на две или више различитих релација)
- Ограничења релација (релвар-а): задаје се ограничење на вредност појединачне релације (релвар-а) које се проверава при ажурирању те релације

# Ограничења стања (наставак)

- Ограничења атрибута: ограничења на скуп дозвољених вредности датог атрибута
- Ограничења типа: дефиниција скупова вредности који чине дати тип



# Ограничења стања - примери

## Ограничења типа

```
TYPE POINT POSSREP
      CARTESIAN (X RATIONAL, Y RATIONAL)
CONSTRAINT ABS
      (THE_X (POINT)) <= 100.0 AND
      ABS(THE_Y (POINT)) <= 100.0 ;
```

# Ограничења стања - примери

## Ограничења атрибута

```
VAR PREDMET BASE RELATION
{
  ID_PREDMETA      INTEGER,
  SIFRA            SIFRA  ,
  NAZIV            NAZIV  ,
  BODOVI          SMALLINT
};
```

# Ограничења стања - примери

## Ограничења релација

```
CONSTRAINT REL1
IF NOT ( IS_EMPTY ( PREDMET ) ) THEN
    COUNT ( PREDMET
            WHERE SIFRA= SIFRA ('R270')) > 0
END IF;
```

Ако уопште постоји неки предмет тада бар једна од њих мора да има шифру Р270.

# Ограничења стања - примери

## Ограничења базе

```
CONSTRAINT BAZA1
FORALL DOSIJE D FORALL ISPIT I
IS_EMPTY (( D JOIN I )
    WHERE I.INDEKS > 20150000
    AND I.INDEKS = D.INDEKS
    AND GODINA_ROKA=GODINA_ROKA(2015);
```

Ни један студент уписан на студије 2015. године не може да полаже испит у 2015. години

# Ограничења прелаза

Пример: ако база садржи податке о особама тада су важећа следећа ограничења:

- Није дозвољено венчање већ венчаних особа
- Дозвољено је венчати се са разведеном особом
- Особе које више нису живе не могу да примају плату (пензију, ...)
- ....

Шта би била ограничења прелаза у случају базе података о студентима?

# Класификација ограничења интегритета

## - други поглед

Класификација према типу ограничења које мора да буде испоштовано у бази

- Референцијални интегритет
- Интегритет домена
- Интегритет редундатности
- Интегритет (пословних) ограничења

# Кандидат за кључ

- Кандидат за кључ релације  $R$  представља подскуп атрибута  $X$  те релације, ако важи:
  - Правило **јединствености**: не постоје две торке у релацији  $R$  које имају исте вредности за  $X$ , и
  - Правило **минималности**: не постоји прави подскуп скупа  $X$  који задовољава правило јединствености.
- Свака релација има бар једног кандидата за кључ (скуп свих атрибута или неки његов прави подскуп)

# Врсте кључева

- **Примарни кључ** - један од кандидата за кључ
- **Алтернативни кључеви** - остали кандидати
- **Спољашњи (страни) кључ** - скуп атрибута једног релвар-а  $R_2$  чије вредности треба да одговарају вредностима неког кандидата за кључ неког релвар-а  $R_1$
- **Суперкључ** - надскуп кандидата за кључ; поседује јединственост али не и минималност



# Кључеви - пример

## - Релвар *DOSIJE*

- примарни кључ је Индекс
- спољашњи кључеву су
  - IdPrograma (на табелу StudijskiProgram)
  - IdStatusa (на табелу StudentskiStatus)

## - Релвар *PREDMET* - примарни кључ је Id

## - Релвар *ISPITNI\_ROK*

- примарни кључ је пар атрибута (SkGodina, OznakaRoka)
- спољашњи кључ је SkGodina (на табелу SkolskaGodina)

# Кључеви - пример

## - Релвар *ISPIT*

- примарни кључ је четворка  
(SkGodina, OznakaRoka, Indeks, IdPredmeta)
- спољашњи кључеви су
  - пар (SkGodina, OznakaRoka) (на табелу IspitniRok)
  - тројка (Indeks, SkGodina, IdPredmeta) (на табелу UpisanKurs)

# Референцијални интегритет

Основна идеја очувања интегритета је да **све вредности у табелама треба да буду усаглашене**

Пример нарушавања интегритета: ако табела *Ispit* садржи податак да је неки студент полагао испит у некој школској години, али не постоји информације у табели *УписанКурс* да је тај студент уписао курс у школској години у којој је евидентирано да га је полагао, тада је дошло до нарушавања интегритета базе

# Референцијални интегритет (наставка)

Спољашњи кључ представља референцу на торку који садржи кандидат за кључ. Одатле је проблем осигуравања да база података не садржи погрешне спољашње кључеве познат као проблем **референцијалног интегритета**, а ограничења која то омогућују се називају **референцијална ограничења**.

# Референцијални интегритет (наставак)

Релација која садржи примарне кључеве (кандидате за кључ) се назива *родитељ релација*, а релација која садржи спољашње кључеве који се реферишу на родитељ релацију се назива *дете релација*.

Референцијални интегритет:

**База не сме да садржи неупарене вредности спољашњих кључева**

# Референцијални интегритет (наставак)

- Релвар-и који немају кандидате за кључ (тј. садрже дупле слоге) се понашају непредвидиво у појединим ситуацијама (видети primer1a.sql из 1.primeri.sql)
- Систем који не поседује знање о кандидатима за кључ понекад показује карактеристике које нису "чисто релационе".

# Референцијални интегритет (наставак)

Дефиниција спољашњих кључева:

```
FOREIGN KEY (lista atributa)
REFERENCES ime_relvar-a} [(lista atributa)]
[MATCH {FULL | PARTIAL}]
[ON DELETE {pravilo brisanja}]
[ON UPDATE {pravilo ažuriranja}]
```

# Референцијална акција

- Правило брисања
  - CASCADE
  - RESTRICT
  - NO ACTION
  - SET NULL
  - SET DEFAULT
- Правило ажурирања
  - CASCADE
  - RESTRICT
  - NO ACTION
  - SET NULL
  - SET DEFAULT



# Референцијални циклус

$$T_n \rightarrow T_{n-1} \rightarrow T_{n-2} \rightarrow \dots \rightarrow T_1 \rightarrow T_n$$

Родитељ табела и дете табела не морају да буду различите табеле

```
VAR STUDENTI BASE RELATION
```

```
{INDEKS INDEKS, ..., PARLAMENT INDEKS, ...}
```

```
PRIMARY KEY {INDEKS}
```

```
FOREIGN KEY {RENAME PARLAMENT AS INDEKS}
```

```
REFERENCES STUDENTI;
```

# Ограничења домена

```
CREATE DOMAIN NAZIV_PREDMETA
            CHAR(30) DEFAULT '???'
CONSTRAINT POSTOJECI_PREDMETI
CHECK ( VALUE IN
        ( 'Relacione baze podataka',
          'Programiranje 1',
          'Programiranje 2',
          'Analiza 1',
          '???'
        ));

CREATE TABLE PREDMETI
        (... , NAZIV NAZIV_PREDMETA, ...);
```

# Ограничења основних табела

- Дефиниција кандидата за кључеве
  - UNIQUE (листа назива атрибута)
  - PRIMARY KEY (листа назива атрибута)
  - NOT NULL
- Дефиниција спољашњих кључева
  - FOREIGN KEY (листа назива атрибута)
  - REFERENCES основна табела [(листа назива атрибута)]
  - [ON DELETE референцијална акција]
  - [ON UPDATE референцијална акција]

# Ограничења основних табела (наставак)

## Дефиниција ограничења: CHECK (условни израз)

```
create table Ispit (  
  SkGodina      smallint not null,  
  OznakaRoka    varchar(20) not null,  
  Indeks        integer not null,  
  IdPredmeta    integer not null,  
  Status        char not null with default 'p',  
               constraint chk_Status check(Status in ('p','n','o','d','x','s')),  
  DatPolaganja date,  
  Poeni         smallint,  
  Ocena         smallint,  
               constraint chk_ocena check(  
                 (Status in ('p','n') and Poeni is null and Ocena is null)  
                 or (Status in ('d','s') and Poeni<50 and Ocena=5)  
                 or (Status in ('o','x') and Poeni between 0 and 100  
                   and Ocena between 5 and 10)),  
  primary key (SkGodina, OznakaRoka, Indeks, IdPredmeta),  
  foreign key fk_Ispit_Rok(SkGodina, OznakaRoka) references IspitniRok,  
  foreign key fk_Ispit_UpisanKurs(Indexs, SkGodina, IdPredmeta) references UpisanKurs);
```

# Ограничења у општем смислу

- Тврдња (енг. *assertion*) је логичке вредност која мора увек да буду испуњене
- Окидач (енг. *trigger*) је низ акција које су придружене одређеним догађајима, и који се извршавају сваки пут када се такав догађај догоди

Имплементације РСУБП не подржавају тврдње али подржавају окидаче.

# Ограничења у општем смислу (наставак)

```
CREATE ASSERTION <назив ограничења>  
    CHECK ( <логички израз>);
```

```
DROP ASSERTIONS <назив ограничења>
```

# Ограничења у општем смислу (наставак)

```
CREATE TRIGGER <назив окидача>  
    <пре или после догађаја>  
    <догађај> ON <основна табела>  
[REFERENCING <листа имена>]  
[FOR EACH <слог или наредба>]  
[WHEN (<логички израз>) ]  
<акција>;
```