

Završni ispit iz Baza podataka

28. lipnja 2007.

Zadaci 1 – 4 se odnose na bazu podataka **provjera** koja sadrži podatke o studentima (relacija **stud**), predmetima (relacija **pred**), upisanim predmetima studenata (relacija **upisanPred**) i rezultatima koje su studenti postigli na provjerama znanja iz predmeta (relacija **rezProvjera**). Podvučeni atributi čine primarni ključ relacije.

stud

<u>JMBAG</u>	CHAR(10)	JMBAG studenta
ime	NCHAR(40)	ime studenta
prezime	NCHAR(40)	prezime studenta

pred

<u>sifPred</u>	SMALLINT	šifra predmeta
naziv	NCHAR(120)	naziv predmeta
ECTSBod	DECIMAL(3,1)	ECTS bodovi predmeta

upisanPred

<u>JMBAG</u>	CHAR(10)	
<u>sifPred</u>	SMALLINT	
ocjena	SMALLINT	ocjena koju je student dobio iz predmeta (NULL, 1, 2, 3, 4 ili 5).

rezProvjera

<u>JMBAG</u>	CHAR(10)	
<u>sifPred</u>	SMALLINT	
<u>rbrProvjera</u>	SMALLINT	
brBod	DECIMAL(3,1)	broj bodova koje je student osvojio na toj provjeri iz predmeta

1. Napisati SQL upite koji će:

- ispisati šifre, nazive predmeta i broj dobivenih ocjena izvrstan (5) za one predmete na kojima je više od 100 studenata dobilo ocjenu izvrstan. **Zadatak riješiti bez upotrebe podupita.** (2 boda)
- za svakog** studenta (uključujući i one koji nisu upisali niti jedan predmet) ispisati JMBAG, ime, prezime, broj upisanih predmeta i prosječnu ocjenu upisanih predmeta. (2 boda)
- ispisati JMBAG i naziv predmeta za svaki studentov **upisani** predmet iz kojeg **nije obavio niti jednu provjeru**. (2 boda)
- svaku ocjenu nedovoljan (1) prepraviti u ocjenu dovoljan (2) ukoliko je student na provjerama iz dotičnog predmeta skupio više od 30 bodova. (2 boda)

2. Napisati izraze relacijske algebre (ne SQL upit) kojim će se dobiti:

- relacija koja sadrži **različite** JMBAG-ove studenata koji su upisali barem jedan predmet s više od 6 ECTS bodova (1 bod)
- relacija s atributima jmbag, ime, prezime koja sadrži podatke o studentima koji su položili predmet sa šifrom 1001, ali nisu položili predmet sa šifrom 1002. (3 boda)

3. Napisati naredbe kojima će se osigurati sljedeće:

- atribut brBod u relaciji rezProvjera mora biti veći ili jednak 0.0 (1 bod)
- u relaciji **pred** ne smiju postojati dva predmeta istog naziva s istim brojem ECTS bodova. (1 bod)
- nije dozvoljena izmjena vrijednosti atributa ECTSBod za predmete koje je upisao jedan ili više studenata. Pokušaj izmjene spriječiti iznimkom koja signalizira pogrešku s tekstom 'Izmjena nije dozvoljena'. (3 boda)

4. Vlasnik baze podataka **provjera** i svih relacija u bazi podataka je korisnik *admin*. Niti jedan drugi korisnik nema nikakvu dozvolu. Napisati niz SQL naredbi koje treba obaviti korisnik *admin*, a koje će osigurati da korisnici *anic*, *babic* i *caric* imaju SVE potrebne dozvole za obavljanje sljedećih operacija:

korisnik:	operacije
<i>anic</i>	pregled svih podataka u relaciji stud s mogućnošću dodjeljivanja tih dozvola drugim korisnicima
<i>babic</i>	pregled i izmjenu podataka o predmetima (relacija pred) koje je upisao barem jedan student
<i>caric</i>	pregled JMBAG-a i prosjeka ocjena položenih predmeta za svakog studenta (korisnik <i>caric</i> smije vidjeti samo sumarne podatke, ali ne i podatke o pojedinim ocjenama iz predmeta)

(4 boda)

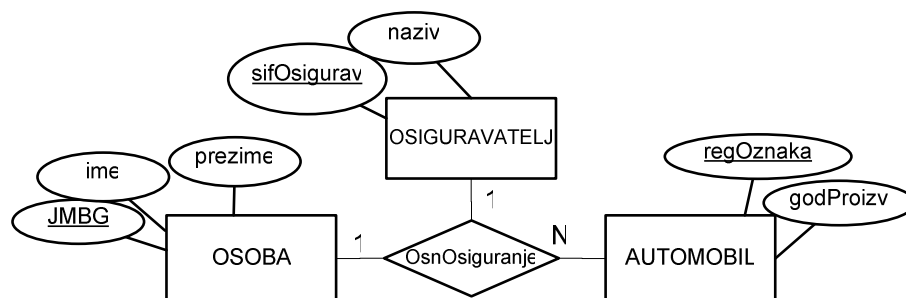
5. Tijekom ljeta na Jadranu se održavaju jednodnevne jedriličarske regate. Istog dana se može održati nekoliko različitih regata. Svaka regata ima samo jednog sponzora za kojeg se evidentira šifra i naziv sponzora. Jedan sponzor može sponzorirati nekoliko regata, ali regate istog sponzora sigurno počinju različitim datumima. Za svaku se regatu, pored sponzora, evidentira datum održavanja regate, trajanje regate izraženo u satima, grad iz kojeg regata isplovjava i grad u kojeg regata uplovjava. Za gradove se evidentiraju poštanski brojevi i nazivi gradova.

Za svaku pojedinu jedrilicu koja sudjeluje u regati evidentira se registarski broj koji jedinstveno identificira jedrilicu, ime jedrilice i najveći dopušteni broj članova posade. Za jedriličara se evidentira jmbg, prezime i ime, te u kojoj regati plovi na kojoj jedrilici. Jedriličar ne mora na svakoj regati ploviti na istoj jedrilici, a očito je da tijekom iste regate ne može ploviti na dvije jedrilice.

Nacrtati ER model i opisati entitete i veze. Entitete, osim slabih entiteta, opisati isključivo vlastitim atributima. Sve sheme moraju zadovoljavati 3NF.

(6 bodova)

6. Napišite SQL naredbe za kreiranje relacija s ugrađenim integritetskim ograničenjima (integritet ključa, entitetski integritet, referencijski integritet) kojima će se prikazani ER model preslikati u ekvivalentan **relacijski model**. Ključevi entiteta su podvučeni. Tipove podataka pojedinih atributa odaberite po nahođenju.



(3 boda)

RJEŠENJA:

1.

a)

```
SELECT pred.sifPred, naziv, COUNT(*) as br5
FROM pred, upisanPred
WHERE pred.sifPred = upisanPred.sifPred
AND ocjena = 5
GROUP BY pred.sifPred, naziv
HAVING COUNT(*) > 100;
```

b)

```
SELECT stud.jmbag, ime, prezime, COUNT(sifPred) AS brUpPred, AVG(ocjena) AS
prOcj
FROM stud LEFT JOIN upisanPred
ON stud.jmbag = upisanPred.jmbag

GROUP BY stud.jmbag, ime, prezime;
```

ILI

```
SELECT jmbag, ime, prezime,
(SELECT COUNT(sifPred) FROM upisanPred
WHERE stud.jmbag = upisanPred.jmbag),
(SELECT AVG(ocjena) FROM upisanPred
WHERE stud.jmbag = upisanPred.jmbag
AND upisanPred.ocjena > 1)
FROM stud
```

c)

```
SELECT jmbag, naziv
FROM upisanPred, pred
WHERE upisanPred.sifPred = pred.sifPred
AND NOT EXISTS (
SELECT *
FROM rezProvjera
WHERE rezProvjera.sifPred = upisanPred.sifPred
AND rezProvjera.jmbag = upisanPred.jmbag);
```

d)

```
UPDATE upisanPred
SET ocjena = 2
WHERE ocjena = 1
AND (SELECT SUM(brBod)
FROM rezProvjera
WHERE rezProvjera.sifPred = upisanPred.sifPred
AND rezProvjera.jmbag = upisanPred.jmbag) > 30;
```

2.

a)

$\pi_{jmbag}(\sigma_{ECTSbod > 6.0}(pred) \bowtie upisanPred)$

b)

$stud \bowtie (\pi_{jmbag}(\sigma_{sifPred=1001 \wedge ocjena>1}(upisanPred))) \setminus \pi_{jmbag}(\sigma_{sifPred=1002 \wedge ocjena>1}(upisanPred)))$

3.

a)

```
ALTER TABLE rezProvjera ADD CONSTRAINT CHECK (brBod >= 0.0))  
CONSTRAINT cBrBod;
```

b)

```
ALTER TABLE pred ADD CONSTRAINT UNIQUE (naziv, ECTSBod) CONSTRAINT uiPred;
```

ILI

```
CREATE UNIQUE INDEX uiPred ON pred (naziv, ECTSBod);
```

c)

```
CREATE PROCEDURE dojavipogresku (p_tekstPogreske CHAR(74))  
RAISE EXCEPTION -746, 0, p_tekstPogreske;  
END PROCEDURE;
```

```
CREATE TRIGGER predUpd  
UPDATE OF ECTSBod ON pred  
REFERENCING OLD AS oldPred NEW AS newPred  
FOR EACH ROW  
WHEN (oldPred.ECTSBod <> newPred.ECTSBod  
AND EXISTS (SELECT *  
FROM upisanPred  
WHERE upisanPred.sifPred = oldPred.sifPred))  
(EXECUTE PROCEDURE dojavipogresku ('Izmjena nije dozvoljena'));
```

ILI

```
CREATE PROCEDURE chkUpdPred (p_sifPred LIKE pred.sifPred)  
DEFINE cnt SMALLINT;  
  
SELECT COUNT(*)  
INTO cnt  
FROM upisanPred  
WHERE upisanPred.sifPred = p_sifPred;  
IF cnt > 0 THEN  
RAISE EXCEPTION -746, 0, 'Izmjena nije dozvoljena';  
END IF  
END PROCEDURE;
```

```
CREATE TRIGGER predUpd  
UPDATE OF ECTSBod ON pred  
REFERENCING OLD AS oldPred NEW AS newPred  
FOR EACH ROW  
WHEN (oldPred.ECTSBod <> newPred.ECTSBod)  
(EXECUTE PROCEDURE chkUpdPred (oldPred.sifPred));
```

4.

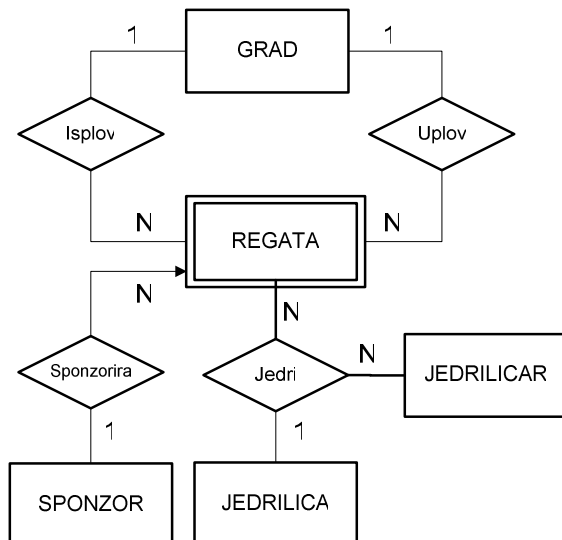
```
GRANT CONNECT TO anic;  
GRANT CONNECT TO babic;  
GRANT CONNECT TO caric;
```

```
GRANT SELECT ON stud TO anic WITH GRANT OPTION;
```

```
CREATE VIEW pred1 AS  
SELECT * FROM pred  
  WHERE EXISTS (SELECT * FROM upisanPred WHERE pred.sifPred =  
upisanPred.sifPred)  
  WITH CHECK OPTION;  
GRANT SELECT, UPDATE ON pred1 TO babic;
```

```
CREATE VIEW pred2 (JMBAG, prosjek) AS  
SELECT JMBAG, AVG(ocjena)  
  FROM upisanPred  
  WHERE ocjena > 1  
GROUP BY JMBAG;  
GRANT SELECT ON pred2 TO caric;
```

5.



GRAD pbr nazGrad K={pbr}	SPONZOR sifSponzor nazSponzor K={sifSponzor}	JEDRILICA regBrJedr imeJedr maxClanova K={regBrJedr}	JEDRILICAR JMBG prezime ime K={JMBG}
REGATA sifSponzor datRegata trajanje K={sifSponzor, datRegata}			
ISPLOV sifSponzor datRegata pbrIsplv K={sifSponzor, datRegata}		UPLOV sifSponzor datRegata pbrUplov K={sifSponzor, datRegata}	
SPONZORIRA sifSponzor datRegata K={sifSponzor, datRegata}		JEDRI sifSponzor datRegata JMBG regBrJedr K={ sifSponzor, datRegata, JMBG}	

```

6. CREATE TABLE osoba (
    JMBG          CHAR(13)      PRIMARY KEY CONSTRAINT pkOsoba
    , ime         NCHAR(40)
    , prezime     NCHAR(40)
);
  
```

```

CREATE TABLE osiguravatelj(
    sifOsigurav  SMALLINT  PRIMARY KEY CONSTRAINT pkOsiguravatelj
    , naziv      NCHAR(120));
  
```

```

CREATE TABLE automobil(
    regOznaka    CHAR(20)  PRIMARY KEY CONSTRAINT  pkAutomobil
    , godProizv  SMALLINT
);
  
```

```

CREATE TABLE osnOsiguranje(
    JMBG          CHAR(13)
    , sifOsigurav SMALLINT
    , regOznaka    CHAR(20)

    , PRIMARY KEY (regOznaka, JMBG) CONSTRAINT pkOsnOsiguranje
    , UNIQUE      (regOznaka, sifOsigurav) CONSTRAINT uiOsnOsiguranje
    , FOREIGN KEY (JMBG) REFERENCES osoba (JMBG) CONSTRAINT fk1
    , FOREIGN KEY (sifOsigurav) REFERENCES osiguravatelj(sifOsigurav) CONSTRAINT fk2
    , FOREIGN KEY (regOznaka) REFERENCES automobil(regOznaka) CONSTRAINT fk3
);
  
```