

## Kolokvijum iz baza podataka

Dozvoljeno vreme: 90 min

Ukupan broj poena: 100

Svaki student rešava zadatke individualno. Dozvoljeno je korišćenje knjiga, on-line dokumentacije i beleški. Nije dozvoljeno konsultovanje sa drugima, korišćenje email-a ili mobilnih telefona. Svako nepoštovanje pravila biće sankcionisano udaljavanjem sa kolokvijuma.

Ukupno ima 5 zadataka. Odgovori treba da budu kratki i jasni. Za sve zadatke, treba predati SQL naredbe i rezultati njihovog izvršenja (odgovor koji dobijete od Oracle DBMS-a) ili C# kod tamo gde je to neophodno. Rešenja treba pamtiti u tekstualnoj datoteci koju pre napuštanja laboratorije treba predati. Ukoliko datoteka nije predata smatra se je student odustao od polaganja kolokvijuma.

Student je položio kolokvijum ukoliko osvoji 50 poena.

Baza podataka o bibliotekama sadrži tri relacije čije su šeme zadate. Relacija BIBLIOTEKA čuva informacije o jedinstvenom identifikatoru i nazivu biblioteke. Relacija KNJIGA sadrži podatke o knjigama koje se čuvaju u bibliotekama: jedinstveni ISBN broj, naziv knjige i datum objavlјivanja. U relaciji KNJIGA\_U\_BIBLIOTECI se nalaze podaci o knjigama koje poseduju različite biblioteke: identifikator biblioteke u kojoj se knjiga nalazi, identifikator knjige i broj primeraka knjige koji se nalaze u toj biblioteci (pozitivan ceo broj, minimalna vrednost 1 a maksimalna vrednost 100).

```
BIBLIOTEKA(ID, NAZIV)  
KNJIGA(ISBN, NASLOV; DATUM_OBJAVLJIVANJA)  
KNJIGA_U_BIBLIOTECI(ID_BIBLIOTEKE, ISBN_KNJIGE, KOLICINA)
```

U pratećim tekstualnim datotekama su zadate SQL naredbe za kreiranje relacija BIBLIOTEKA, KNJIGA i KNJIGA\_U\_BIBLIOTECI.

### Zadaci:

- Napisati SQL upit kojim se kreira pogled IMA\_PROKLETU\_AVLIJU koji sadrži nazive biblioteka koje imaju bar jedan primerka knjige Prokleta avlja. Korišćenjem kreiranog pogleda odrediti naziv biblioteke koja ima najviše primeraka knjige Prokleta avlja. (25 poena)

```
CREATE VIEW IMA_PROKLETU_AVLIJU  
AS  
SELECT B.NAZIV, KB.KOLICINA  
FROM BIBLIOTEKA B, KNJIGA_U_BIBLIOTECI KB, KNJIGA K  
WHERE B.ID = KB.ID_BIBLIOTEKE  
    AND KB.ISBN_KNJIGE = K.ISBN  
    AND K.NASLOV = 'Prokleta avlja';  
  
SELECT *  
FROM IMA_PROKLETU_AVLIJU  
ORDER BY KOLICINA DESC;
```

```
CREATE VIEW succeeded.  
NAZIV                      KOLICINA  
-----  
Biblioteka Sv.Naum          10  
Narodna biblioteka          7  
Biblioteka Svetozar Markovic 5
```

### Alternativno resenje:

```
SELECT NAZIV, KOLICINA  
FROM IMA_PROKLETU_AVLIJU  
WHERE KOLICINA = (SELECT MAX(KOLICINA) FROM IMA_PROKLETU_AVLIJU);
```

2. U sistemu se pojavila nova knjiga Ljubav u doba kolere (ISBN: 100, datum izdavanja: 01/01/2010. godine). Napisati SQL naredbu kojom se ova knjiga dodaje u system. Prikazati SQL naredbu za ažuriranje i rezultujuću tabelu sa promenjenim podacima. (20 poena)

```
SELECT * FROM KNJIGA;
```

ISBN	NASLOV	DATUM_OBJAVLJIVANJA
1234567892	Beli Tigar	12-MAR-75
3	Znakovi	23-MAY-79
4	Prokleta avlija	05-OCT-69
5	Led	30-JUL-81
6	Deca	17-JAN-77
7	Vreme smrti	27-OCT-79
8	Gresnik	28-OCT-83
9	Izabrana dela	30-OCT-89
10	Mrtvo more	01-NOV-69
11	Dozivljaji Nikoletine Bursaca	03-NOV-99
12	Orlovi rano lete	04-NOV-79
13	Djecak prati zmaja	04-NOV-81
14	Ocevi i oci	04-NOV-83
15	Cekajuci Godoa	08-NOV-50
1234567891	Na drini cuprija	08-NOV-72

15 rows selected

```
INSERT INTO KNJIGA (ISBN, NASLOV, DATUM_OBJAVLJIVANJA)
VALUES(100, 'Ljubav u doba kolere', '01-JAN-10');
```

```
SELECT * FROM KNJIGA;
```

ISBN	NASLOV	DATUM_OBJAVLJIVANJA
1234567892	Beli Tigar	12-MAR-75
3	Znakovi	23-MAY-79
4	Prokleta avlija	05-OCT-69
5	Led	30-JUL-81
6	Deca	17-JAN-77
7	Vreme smrti	27-OCT-79
8	Gresnik	28-OCT-83
9	Izabrana dela	30-OCT-89
10	Mrtvo more	01-NOV-69
11	Dozivljaji Nikoletine Bursaca	03-NOV-99
12	Orlovi rano lete	04-NOV-79
13	Djecak prati zmaja	04-NOV-81
14	Ocevi i oci	04-NOV-83
15	Cekajuci Godoa	08-NOV-50
1234567891	Na drini cuprija	08-NOV-72
100	Ljubav u doba kolere	01-JAN-10

16 rows selected

3. Napisati SQL upite kojima se tri primerka knjige Prokleta avlja prebacuju iz biblioteke Biblioteka Sv.Naum u biblioteku Biblioteka Krusevac. (20 poena)

Najpre se knjige uzimaju iz biblioteke Sv.Naum:

```
UPDATE KNJIGA_U_BIBLIOTECI
SET KOLICINA = KOLICINA - 3
WHERE ID_BIBLIOTEKE = (SELECT ID FROM BIBLIOTEKA WHERE NAZIV = 'Biblioteka Sv.Naum')
AND ISBN_KNJIGE = (SELECT ISBN FROM KNJIGA WHERE NASLOV = 'Prokleta avlja');
```

Knjige se nakon toga prebacuju u biblioteku Krusevac:

```
INSERT INTO KNJIGA_U_BIBLIOTECI
VALUES((SELECT ID FROM BIBLIOTEKA WHERE NAZIV = 'Biblioteka Krusevac'),
(SELECT ISBN FROM KNJIGA WHERE NASLOV = 'Prokleta avlja'),
3);
```

4 . Korišćenjem programskog jezika C# i ADO.NET biblioteke napisati program (funkciju Main) koji za uneti naslov knjige prikazuje nazive biblioteka koje imaju primerke te knjige. (35 poena)

```
static void Main(string[] args)
{
    try
    {

        //definise se connection string za ORACLE
        string strConnection = "Data Source=160.99.12.154/GISLAB;" +
                               + "User Id=aca;Password=aca;";

        Console.Write("Uneti naslov knjige:");
        string naslov = Console.ReadLine();

        //kreira se novi connection objekat
        OracleConnection con = new OracleConnection(strConnection);

        //uspopstavlja se konekcija sa bazom podataka
        con.Open();

        //definise se SQL komanda koju treba izvrsiti
        string strSQL = "SELECT B.NAZIV FROM BIBLIOTEKA B INNER JOIN "
                       + "KNJIGA_U_BIBLIOTECI ON B.ID = ID_BIBLIOTEKE "
                       + "INNER JOIN KNJIGA K ON ISBN_KNJIGE = K.ISBN "
                       + "WHERE NASLOV = '" + naslov + "'";

        OracleCommand cmd = new OracleCommand(strSQL, con);

        //izvsava se SQL komanda a njene rezultate ce prihvatiti DataReader objekat
        OracleDataReader r = cmd.ExecuteReader();

        //stampaju se podaci o bibliotekama
        while (r.Read())
        {
            string strBiblioteka;

            strBiblioteka = r.GetString(0);

            Console.WriteLine(strBiblioteka);
        }

        //kraj rada sa bazom podataka
        r.Close();

        con.Close();
    }
}
```

```
        }
        catch (Exception ex)
        {
            Console.WriteLine("Doslo je do greske prilikom pristupanja bazi
podataka: " + ex.Message);

        }
    }
```