SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Bartul Brajković, 0036507098

23. ožujka 2021.

LABORATORIJ PROFILA 2

Odjeljak Sustavi baza podataka

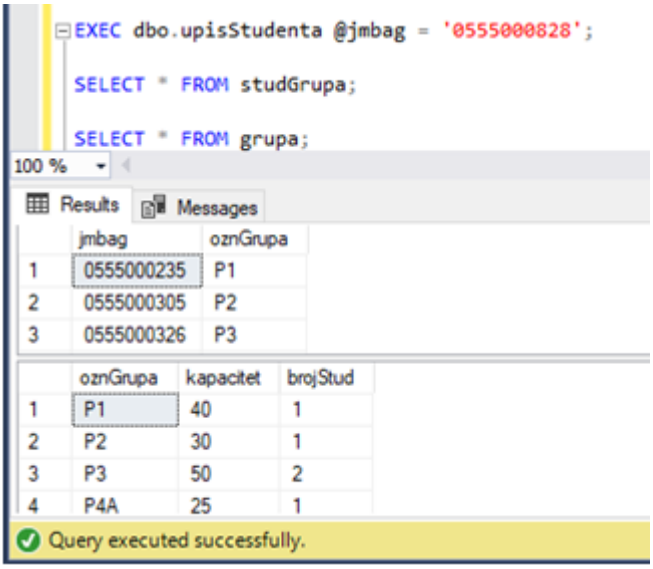
1. Vježba

4. Zadatak

4.1. Zadatak

```
IF OBJECT_ID('upisStudenta') IS NOT NULL DROP PROCEDURE upisStudenta

GO
CREATE PROCEDURE dbo.upisStudenta @jmbag CHAR(10)
AS
BEGIN
	DECLARE @min_grupa CHAR(10);

	SET @min_grupa =
	(SELECT TOP(1) oznGrupa
		FROM grupa
		ORDER BY CAST(brojStud AS DECIMAL)/kapacitet, oznGrupa
	);

	DECLARE @broj_slobodnih_grupa INT;

	SET @broj_slobodnih_grupa = (SELECT COUNT(*) FROM grupa WHERE
grupa.brojStud < grupa.kapacitet)

	IF @jmbag IN (SELECT jmbag FROM studGrupa)
		THROW 50501, 'Student je već upisan u nastavnu grupu', 1

	IF @jmbag NOT IN (SELECT jmbag FROM stud)
		THROW 50502, 'Student ne postoji', 1

	IF (@broj_slobodnih_grupa = 0)
		THROW 50503, 'Sve grupe su već popunjene', 1

	INSERT INTO studGrupa VALUES(@jmbag, @min_grupa);
	UPDATE grupa
		SET brojStud = brojStud + 1
		WHERE oznGrupa = @min_grupa;

END;
GO

EXEC dbo.upisStudenta @jmbag = '0555000828';

SELECT * FROM studGrupa;

SELECT * FROM grupa;
```

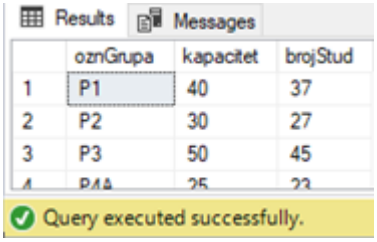
Pregled relacija studGrupa i grupa nakon izvršavanja procedure:



4.2. zadatak

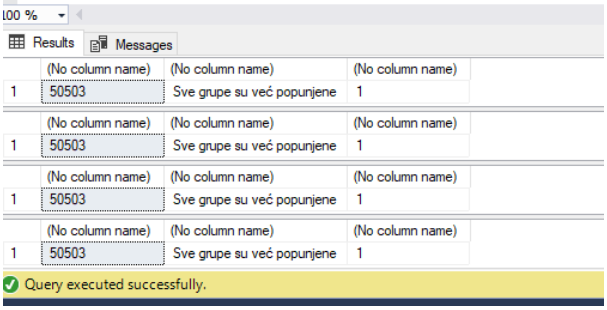
```
4   IF OBJECT_ID('rasporediPoGrupama') IS NOT NULL DROP PROCEDURE
    rasporediPoGrupama
5
6   GO
7   CREATE PROCEDURE dbo.rasporediPoGrupama
8   AS
9   BEGIN
10
11    DECLARE @jmbag CHAR(10);
12
13    DECLARE db_cursor CURSOR FOR
14           SELECT jmbag
15           FROM stud
16           ORDER BY prezStud ASC, imeStud ASC
17
18    OPEN db_cursor
19
20    FETCH NEXT FROM db_cursor INTO @jmbag
21
22    WHILE (@@FETCH_STATUS = 0) BEGIN
23
24           BEGIN TRY
25
26                  EXEC dbo.upisStudenta @jmbag = @jmbag;
27
28           END TRY
29           BEGIN CATCH
30
31                  IF ERROR_NUMBER() != 50501 AND ERROR_NUMBER() != 50502
32                  BEGIN
33
34                         DECLARE @error_number INT;
35                         DECLARE @error_message VARCHAR(200);
36                         DECLARE @error_state INT;
37
38                         SET @error_number = ERROR_NUMBER();
39                         SET @error_message = ERROR_MESSAGE();
40                         SET @error_state = ERROR_STATE();
41
42                         SELECT @error_number, @error_message, @error_state;
43
44                  END
45
46           END CATCH
47
48           FETCH NEXT FROM db_cursor INTO @jmbag
49
50    END
51
52    CLOSE db_cursor
53    DEALLOCATE db_cursor
54
55  END;
56  GO
```

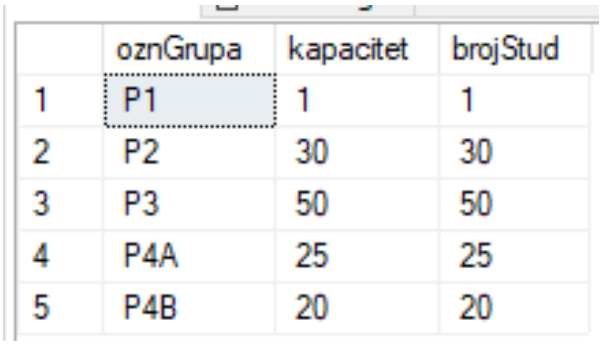
a) raspoređivanje studenata za početno stanje podataka u relacijama stud, grupa i studGrupa



b) raspoređivanje studenata za početno stanje podataka u relacijama stud, grupa i studGrupa, osim što je kapacitet grupe P1 postavljen na 1

```
UPDATE grupa SET kapacitet = 1 WHERE oznGrupa = 'P1';
```





	oznGrupa	kapacitet	brojStud
1	P1	1	1
2	P2	30	30
3	P3	50	50
4	P4A	25	25
5	P4B	20	20

c) raspoređivanje studenata za početno stanje podataka u relacijama stud, grupa i studGrupa, osim što je kapacitet grupe P1 postavljen na 1

```
UPDATE grupa SET kapacitet = 0 WHERE oznGrupa = 'P1';
```

100 %

Results | Messages

	(No column name)	(No column name)	(No column name)
1	8134	Divide by zero error encountered.	1

	(No column name)	(No column name)	(No column name)
1	8134	Divide by zero error encountered.	1

	(No column name)	(No column name)	(No column name)
1	8134	Divide by zero error encountered.	1

	(No column name)	(No column name)	(No column name)
1	8134	Divide by zero error encountered.	1

Query executed successfully.

4. Zadatak

5.1. zadatak

```
package sbp_lab1;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class UpisStudenta1 {

	public static void main (String argv[]) {

		Connection connection = otvoriKonekciju();

		try {
		   CallableStatement cstmt = connection.prepareCall("{call
dbo.upisStudenta}");

		   cstmt.execute();

		   System.out.println("Student je uspješno upisan.");

		   cstmt.close();

		   connection.close();
		}
		catch (SQLException exception) {

		   ispisiPogresku(exception);

		   System.exit(-1);
		}
	    }

	   private static void ispisiPogresku(SQLException exception) {

		System.out.println(exception.getErrorCode() + "; " +
exception.getMessage() + "; " + "State=" + exception.getSQLState());
	    }

	  private static Connection otvoriKonekciju () {

		String url =
		      "jdbc:sqlserver://localhost:1433;"
		    + "databaseName=labprof1;"
		    + "user=sa;"
		    + "password=xxxxxxx;" ;
```

```
            try {

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

            }
            catch (ClassNotFoundException exception) {

               System.out.println(exception.getMessage());

               System.exit(-1);
            }

            // uspostavljanje konekcije
            Connection conn = null;
            try {

               conn = DriverManager.getConnection(url);

            }
            catch (SQLException exception) {

               System.out.println(exception.getErrorCode() + " " +
exception.getMessage());

               System.exit(-1);
            }

            return conn;
         }

}
```

5.2. zadatak

```
package sbp_lab1;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class UpisStudenta2 {

        public static void main(String argv[]) throws SQLException {

              Connection connection = otvoriKonekciju();

              Scanner scan = new Scanner(System.in);

            String jmbag = scan.next();

              Statement stmt = null;

            stmt = connection.createStatement();

            ResultSet rs = stmt.executeQuery("SELECT oznGrupa FROM grupa
ORDER BY CAST(brojStud AS DECIMAL)/kapacitet, oznGrupa LIMIT 1");

            String oznGrupa = rs.getString("oznGrupa").trim();

            ResultSet rs2 = stmt.executeQuery("SELECT COUNT(*) FROM grupa
WHERE grupa.brojStud < grupa.kapacitet");

            int brojSlobodnihGrupa = rs2.getInt(0);

            if (stmt.executeQuery("SELECT jmbag FROM
studGrupa").getString(jmbag) != null) {

              throw new SQLException("Student je već upisan u grupu");
            }

            if (stmt.executeQuery("SELECT jmbag FROM
stud").getString(jmbag) == null) {

              throw new SQLException("Student ne postoji");
            }

            if (brojSlobodnihGrupa == 0) {

              throw new SQLException("Sve grupe su već popunjene");
            }

            stmt.executeUpdate("INSERT INTO studGrupa " + "VALUES (" +
jmbag + "," + oznGrupa +")");
```

```
            String sql = "UPDATE grupa " + "SET brojStud = brojStud + 1
WHERE oznGrupa = "+ oznGrupa +" ";

            stmt.executeUpdate(sql);


            rs.close();
            rs2.close();
            stmt.close();
            connection.close();

        }


         private static Connection otvoriKonekciju () {

             String url =
                     "jdbc:sqlserver://localhost:1433;"
                   + "databaseName=labprof1;"
                   + "user=sa;"
                   + "password=xxxxxxx;" ;

             try {


Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

             } catch (ClassNotFoundException exception) {

                System.out.println(exception.getMessage());

                System.exit(-1);
             }

             Connection conn = null;

             try {

                conn = DriverManager.getConnection(url);

             } catch (SQLException exception) {

                System.out.println(exception.getErrorCode() + " " +
exception.getMessage());

                System.exit(-1);
             }
             return conn;
          }
}
```

5.3. zadatak

```
package sbp_lab1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class ListaPoGrupama {

	public static void main(String argv[]) throws SQLException {

		Connection connection = otvoriKonekciju();

		Statement stmt = null;

	    stmt = connection.createStatement();

//	    ResultSet grupe = stmt.executeQuery("SELECT oznGrupa FROM grupa
ORDER BY oznGrupa");

//	    ResultSet studenti = stmt.executeQuery("SELECT jmbag, imeStud,
prezStud FROM stud ORDER BY prezStud, imeStud");

//	    ResultSet studGrupa = stmt.executeQuery("SELECT jmbag, oznGrupa
FROM studGrupa");

	    ResultSet studentiUGrupi = stmt.executeQuery("SELECT oznGrupa,
jmbag, imeStud, prezStud FROM studGrupa NATURAL JOIN stud ORDER BY
oznGrupa, prezStud, imeStud");

	    while (studentiUGrupi.next()) {

		System.out.println(studentiUGrupi.toString());
	    }

	}

	private static Connection otvoriKonekciju () {

	     String url =
	             "jdbc:sqlserver://localhost:1433;"
	           + "databaseName=labprof1;"
	           + "user=sa;"
	           + "password=xxxxxxx;" ;

	     try {


Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");

	     } catch (ClassNotFoundException exception) {

		System.out.println(exception.getMessage());
```

```
                System.exit(-1);
            }

            Connection conn = null;

            try {

                conn = DriverManager.getConnection(url);

            } catch (SQLException exception) {

                System.out.println(exception.getErrorCode() + " " +
exception.getMessage());

                System.exit(-1);
            }
            return conn;
        }


/*     private boolean isThere(ResultSet rs, String column){

          try{

              rs.findColumn(column);

              return true;

          }
          catch (SQLException sqlex){

              System.out.println(sqlex.getMessage());
          }

          return false;
      }
*/
}
```