

Adatbázisrendszerek BSc 2.

5.Gyak 2022.03.09

Készítette: Földesi Dominik

G616TL

2022.03.09

1.feladat

```
public void StatikusTablaLetrehozas() {  
  
    String sqlp_auto = "create table auto ( rsz char(6) primary key,"  
        + "tipus char(10) not null, szin char(10) default 'feher',"  
        + "evjarat number(4), ar number(8) check(ar>0) )";  
    String sqlp_tulaj = "create table tulaj( id number(3) primary key,"  
        + "nev char(20) not null, cim char(20), szuldatum date)";  
  
    if (conn != null) {  
  
        try {  
            s = conn.createStatement();  
            s.executeUpdate(sqlp_auto);  
            System.out.println("Autó tábla létrejött!");  
            s.executeUpdate(sqlp_tulaj);  
            System.out.println("Tulajdonos tábla létrejött!");  
            s.close();  
        } catch (Exception ex) {  
            System.err.println(ex.getMessage());  
        }  
  
    }  
  
}
```

2.feladat

```
public void StatikusTablaModositas() {  
  
    if (conn != null) {  
  
        try {  
  
            String sqlp = "alter table auto add(tulaj_id references tulaj)";  
            s = conn.createStatement();  
            s.executeUpdate(sqlp);  
            System.out.println("Autó tábla módosítva!");  
            s.close();  
  
        } catch (Exception ex) {  
            System.err.println(ex.getMessage());  
        }  
  
    }  
  
}
```

3.feladat

```
public void StatikusAdatfelvitel() {  
    if (conn != null) {  
        String sqlp_tulaj = "insert into tulaj values (1, 'Tóth Máté', "  
            + " 'Miskolc', to_date('1980.05.12', 'yyyy.mm.dd'))";  
        String[] sqlp = { "insert into auto values('aaa111', 'opel', 'piros', 2014, 1650000, 1)",  
            "insert into auto values('bbb222', 'mazda', null, 2016, 2800000, 1)",  
            "insert into auto(rsz, tipus, evjarat, ar) values('ccc333', 'ford', 2009, 15000000)" };  
  
        try {  
            s = conn.createStatement();  
            s.executeUpdate(sqlp_tulaj);  
            System.out.println("Tulaj felvéve!");  
            s.close();  
        } catch (Exception ex) {  
            System.err.println(ex.getMessage());  
        }  
  
        for (int i = 0; i < sqlp.length; i++) {  
            try {  
                s = conn.createStatement();  
                s.executeUpdate(sqlp[i]);  
                System.out.println("Autó felvéve");  
                s.close();  
            } catch (Exception ex) {  
                System.err.println(ex.getMessage());  
            }  
        }  
    }  
}
```

4.feladat

```
public void DinamikusAdatfelvitel() {  
    if (conn != null) {  
        String sqlp = "insert into auto (rsz, tipus, szin, evjarat, ar, tulaj_id)" + "values (?, ?, ?  
        System.out.println("Kérem a rendszámot: ");  
        String rsz = sc.next().trim();  
        System.out.println("Kérem a típust: ");  
        String tipus = sc.next().trim();  
        System.out.println("Kérem a színt: ");  
        String szin = sc.next().trim();  
        System.out.println("Kérem az évjáratot: ");  
        int ev = sc.nextInt();  
        System.out.println("Kérem az árat: ");  
        float ar = sc.nextFloat();  
        System.out.println("Kérem a tulajdonos azonosítóját: ");  
        int tulaj_id = sc.nextInt();  
        try {  
            ps = conn.prepareStatement(sqlp);  
            ps.setString(1, rsz);  
            ps.setString(2, tipus);  
            ps.setString(3, szin);  
            ps.setInt(4, ev);  
            ps.setFloat(5, ar);  
            ps.setInt(6, tulaj_id);  
            ps.executeUpdate();  
            ps.close();  
            System.out.println("Autó felvéve!");  
        } catch (Exception ex) {  
            System.err.println(ex.getMessage());  
        }  
    }  
}
```

5.feladat

```
public void DinamikusAdattorles() {  
  
    System.out.println("Törlendő autó: ");  
    String rsz = sc.next();  
    String sqlp = "delete from " + "user" + ".AUTO" + " where rsz=?";  
  
    if (conn != null) {  
  
        try {  
  
            ps = conn.prepareStatement(sqlp);  
            ps.setString(1, rsz);  
            ps.close();  
            System.out.println(rsz + " rendszámú autó törölve\n");  
  
        } catch (Exception ex) {  
  
            System.err.println(ex.getMessage());  
  
        }  
  
    }  
  
}
```

```
public void StatikusAdattorles() {
```

6.feladat

```
public void StatikusAdattorles() {  
  
    System.out.println("Törlendő autó: ");  
    String rsz = sc.next();  
    String sqlp = "delete from auto where rsz like '" + rsz + "'";  
  
    if (conn != null) {  
        try {  
            s = conn.createStatement();  
            s.executeUpdate(sqlp);  
            s.close();  
            System.out.println(rsz + " rendszámú autó törölve!\n");  
  
        } catch (Exception ex) {  
            System.err.println(ex.getMessage());  
        }  
    }  
}
```

7.feladat

```
public void StatikusLekerdezes() {  
    if (conn != null) {  
        String sqlp = "select * from auto";  
        System.out.println("Rendszám Típus Szín Évjárat Ár Tulaj");  
        System.out.println("-----");  
  
        try {  
            s = conn.createStatement();  
            s.executeQuery(sqlp);  
            rs = s.getResultSet();  
            while (rs.next()) {  
  
                String rsz = rs.getString("rsz");  
                String tipus = rs.getString("tipus");  
                String szin = rs.getString("szin");  
                int evjarat = rs.getInt("evjarat");  
                int ar = rs.getInt("ar");  
                int tulaj_id = rs.getInt("tulaj_id");  
  
                System.out  
                    .println(rsz + "\t\t" + tipus + "\t" + szin + "\t" + evjarat + "\t" + ar + "\t"  
                );  
            }  
            rs.close();  
        } catch (Exception ex) {  
            System.err.println(ex.getMessage());  
        }  
    }  
}
```

8.feladat

```
public void ModosithatokKurzor() {  
  
    System.out.println("Szín: ");  
    String szin = sc.next().trim();  
    String sqlp = "select ar from auto where szin = '" + szin + "'";  
  
    if (conn != null) {  
        try {  
            s = conn.createStatement(ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_UPDATABLE);  
            rs = s.executeQuery(sqlp);  
            while (rs.next()) {  
                int regiar = rs.getInt("ar");  
                rs.updateInt("ar", (regiar * 2));  
                rs.updateRow();  
            }  
        } catch (Exception ex) {  
            System.err.println(ex.getMessage());  
        }  
    }  
}
```

9.feladat

```
public void InEljarasHivas() {
    if (conn != null) {
        try {
            String sqlp = "create or replace procedure arcsokkent " + "(kor IN number) is " + "begin "
                + "update auto set ar=ar*0.9 where " + "to_char(sysdate, 'yyyy')-evjarat > kor ; "
                + "end;";
            System.out.println("Kor: ");
            int kor = sc.nextInt();
            s = conn.createStatement();
            s.executeUpdate(sqlp);
            System.out.println("Függvény létrejött\n");
            cs = conn.prepareCall("{call arcsokkent(?)}");
            cs.setInt(1, kor);
            cs.execute();
        } catch (Exception ex) {
            System.err.println(ex.getMessage());
        }
    }
}
```

10.feladat

```
public void OutEljarasHivas() {
    if (conn != null) {
        try {
            String sqlp = "create or replace procedure atlagar " + "(sz IN char, atl OUT number) is "
                + "select (avg(ar) into atl from auto where szin=sz; " + "end;";
            System.out.println("Szín: ");
            String szin = sc.next();
            s = conn.createStatement();
            s.executeUpdate(sqlp);
            System.out.println("Eljárás létrejött\n");
            cs = conn.prepareCall("{call atlagar(?, ?)}");
            cs.setString(1, szin);
            cs.registerOutParameter(2, java.sql.Types.FLOAT);
            cs.execute();
            float atlag = cs.getFloat(2);
            System.out.println(szin + " autók átlagára: " + atlag + "\n");
        } catch (Exception ex) {
            System.err.println(ex.getMessage());
        }
    }
}
```


11.feladat

```
public void FuggvenyHivas() {
    if (conn != null) {
        try {
            String sqlp = "create or replace function atlagarfv " + "(sz IN char) return number is "
                + "atl number(10,2); " + "begin " + "select avg(ar) into atl from auto where szin="
                + "return atl; " + "end;";
            System.out.println("Szín: ");
            String szin = sc.next();
            s = conn.createStatement();
            s.executeUpdate(sqlp);
            System.out.println("Fgv létrejött\n");
            cs = conn.prepareCall("{? = call atlagarfv(?) }");
            cs.setString(2, szin);
            cs.registerOutParameter(1, java.sql.Types.FLOAT);
            cs.execute();
            float atlag = cs.getFloat(1);
            System.out.println(szin + " autók átlagára: " + atlag + "\n");
        } catch (Exception ex) {
            System.err.println(ex.getMessage());
        }
    }
}
```

12.feladat

```
public void DinamikusTablaTorles() {
    String sqlp = "create or replace procedure tablatorles(nev IN char) is " + "begin "
        + "execute immediate 'drop table' || nev; " + "end;";
    System.out.println("Törlendő tábla: ");
    String name = sc.next().trim();
    if (conn != null) {
        try {
            s = conn.createStatement();
            s.executeUpdate(sqlp);
            cs = conn.prepareCall("{call tablatorles(?) }");
            cs.setString(1, name);
            cs.execute();
            System.out.println(name + " tábla törölve\n");
        } catch (Exception ex) {
            System.err.println(ex.getMessage());
        }
    }
}
```