

JEGYZŐKÖNYV

Adatbázis rendszerek II.

Féléves feladat

Készítette: **Kiss Gergő**

Neptunkód: **BKUPJ9**

Gyakorlatvezető: *Dr. Bednarik László*

1. A feladat leírása:

A következő feladatban egy autócentrum modellezését fogom bemutatni. Az adatbázisban 3 tábla szerepel, egy „Beszállító”, egy „Autó”, valamint egy „Vásárló”. A „Vásárló” tábla összeköttetésben áll az „Autó” táblával, a rendszámon keresztül.

2. A táblákat létrehozó SQL parancsok:

```
104
105 public static void StatikusTabelaLetrozas() {
106     String sqlp_beszallito="CREATE TABLE beszallito"
107         + "("
108         + " id int not null,"
109         + " cegnev char(30) NOT NULL,"
110         + " szekhely char(40) NOT NULL,"
111         + " raktarkeszlet integer,"
112         + " szallitas date NOT NULL,"
113         + " PRIMARY KEY (id)"
114         + ")";
115
116     String sqlp_auto="CREATE TABLE auto"
117         + "("
118         + " rsz char(6) NOT NULL,"
119         + " beszallito_id int NOT NULL,"
120         + " tipus char(20) NOT NULL,"
121         + " szin char(10) NOT NULL,"
122         + " motor char(10) NOT NULL,"
123         + " an integer NOT NULL,"
124         + " gyartasi_ido date NOT NULL,"
125         + " PRIMARY KEY (rsz)"
126         + ")";
127
128     String sqlp_vasarlo="CREATE TABLE vasarlo"
129         + "("
130         + " adoszam numeric(10) NOT NULL,"
131         + " nev char(25) NOT NULL,"
132         + " lakhely char(30) NOT NULL,"
133         + " szulido date NOT NULL,"
134         + " PRIMARY KEY (adoszam)"
135         + ")";
136
137     if(conn!=null) {
138         try {
139             s=conn.createStatement();
140             s.executeUpdate(sqlp_beszallito);
141             System.out.println("Beszállító tábla létrejött\n");
142             s.executeUpdate(sqlp_auto);
143             System.out.println("Autó tábla létrejött\n");
144             s.executeUpdate(sqlp_vasarlo);
145             System.out.println("Vásárló tábla létrejött\n");
146             s.close();
147         } catch (Exception ex) {
148             System.err.println(ex.getMessage());
149         }
150     }
151 }
152
```

3. A táblát feltöltő SQL parancsok:

```
166 public static void StatikusAdatfelvetel() {
167     if(conn != null) {
168         String sqlp_besz[]={
169             "insert into beszallito values (1, 'Ford Hungaria Bt.', 'Budapest, Fő út 101.', 110, to_date('2022.05.12', 'yyyy.mm.dd'))",
170             "insert into beszallito values (2, 'Skoda Hungaria Bt.', 'Budapest, Mellék út 101.', 200, to_date('2022.06.11', 'yyyy.mm.dd'))",
171             "insert into beszallito values (3, 'Opel Hungaria Bt.', 'Budapest, Sáfrány út 3.', 300, to_date('2022.04.12', 'yyyy.mm.dd'))",
172             "insert into beszallito values (4, 'Mazda Hungaria Bt.', 'Budapest, Vágány út 101.', 400, to_date('2022.07.09', 'yyyy.mm.dd'))"
173         };
174
175         String sqlp_auto[]={
176             "insert into auto values ('jdc761',1, 'Ford Mondeo', 'piros', '1.8dt', 14000000, to_date('2022.01.12', 'yyyy.mm.dd'))",
177             "insert into auto values ('jbc767',2, 'Skoda Octavia', 'kék', '2.0 crtdi', 16000000, to_date('2021.03.12', 'yyyy.mm.dd'))",
178             "insert into auto values ('axk455',3, 'Opel Insignia', 'fehér', '1.7dti', 19000000, to_date('2022.09.21', 'yyyy.mm.dd'))",
179             "insert into auto values ('okl767',4, 'Mazda 6', 'piros', '1.6b', 11000000, to_date('2022.03.12', 'yyyy.mm.dd'))"
180         };
181
182         String sqlp_vas[]={
183             "insert into vasarlo values(8485825968, 'Próba László', 'Szerencs, Fő út 101.', to_date('1998.01.12', 'yyyy.mm.dd'),'jdc761')",
184             "insert into vasarlo values(8485825947, 'Próba Béla', 'Kazincbarcika, Fő út 101.', to_date('1994.01.30', 'yyyy.mm.dd'),'jbc767')",
185             "insert into vasarlo values(8485824724, 'Próba Péter', 'Eger, Fő út 101.', to_date('1991.01.18', 'yyyy.mm.dd'),'axk455')",
186             "insert into vasarlo values(8485827896, 'Próba Imre', 'Debrecen, Fő út 101.', to_date('1997.01.21', 'yyyy.mm.dd'),'okl767')"
187         };
188
189         for(int i = 0; i < sqlp_besz.length; i++) {
190             try {
191                 s=conn.createStatement();
192                 s.executeUpdate(sqlp_besz[i]);
193                 System.out.println("Beszállító felvéve\n");
194                 s.close();
195             } catch (Exception ex) {
196                 System.err.println(ex.getMessage());
197             }
198         }
199
200         for(int i = 0; i < sqlp_auto.length; i++) {
201             try {
202                 s=conn.createStatement();
203                 s.executeUpdate(sqlp_auto[i]);
204                 System.out.println("Auto felvéve\n");
205                 s.close();
206             } catch (Exception ex) {
207                 System.err.println(ex.getMessage());
208             }
209         }
210     }
211 }
```

```

208     }
209
210
211     for(int i = 0; i < sqlp_vas.length; i++) {
212         try {
213             s=conn.createStatement();
214             s.executeUpdate(sqlp_vas[i]);
215             System.out.println("Vásárló felvéve\n");
216             s.close();
217         }catch(Exception ex) {
218             System.err.println(ex.getMessage());
219         }
220     }
221 }
222
223

```

4. A program funkciói:

-Driver Regisztrálás, Kapcsolódás a szerverhez:-külön metódusokon keresztül történik, globális változókat használva.

```

67
68 public static void DriverReg() {
69     try {
70         Class.forName("oracle.jdbc.driver.OracleDriver");
71         System.out.println("Sikeres driver Regisztrálás!\n");
72     }catch(Exception e) {
73         System.err.println(e.getMessage());
74     }
75 }
76
77 public static void Connect() {
78     try {
79         conn = DriverManager.getConnection(url, user, pwd);
80         System.out.println("Sikeres kapcsolódás\n");
81     }catch (Exception e) {
82         System.err.println(e.getMessage());
83     }
84 }
85

```

```

Sikeres driver regisztrálás!
Sikeres kapcsolódás
Beállított tábla visszaev
Vasarlo tábla törlve!
Auto tábla törlve!
Beállított tábla létrejött
Autó tábla létrejött
Vásárló tábla létrejött
Autó tábla módosítva!
Beállított felvéve
Beállított felvéve
Beállított felvéve
Beállított felvéve
Auto felvéve
Auto felvéve
Auto felvéve
Auto felvéve

```

Adatbázis „letisztítása”: a program elindításakor, az előző futtatás után fennmaradt táblákat teljes egészében letörli.

```
86 public static void StatikusTablatörles() {
87     if(conn != null) {}
88     try {
89         String sqlp_beszallito="DROP TABLE BESZALLITO";
90         String sqlp_auto="DROP TABLE auto";
91         String sqlp_vasarlo="DROP TABLE vasarlo";
92         s=conn.createStatement();
93         s.executeUpdate(sqlp_beszallito);
94         System.out.println("Beszallito tábla törölve!\n");
95         s.executeUpdate(sqlp_vasarlo);
96         System.out.println("Vasarlo tábla törölve!\n");
97         s.executeUpdate(sqlp_auto);
98         System.out.println("Auto tábla törölve!\n");
99         s.close();
100     }catch(Exception ex) {
101         System.err.println(ex.getMessage());
102     }
103 }
104
```

```
Sikeres driver regisztrálás!
Kikeresés:
Beszallito tábla törölve!
Vasarlo tábla törölve!
Auto tábla törölve!
Beszallito vasaa létrejött
Auto tábla létrejött
Vasarlo tábla létrejött
Auto tábla módosítva!
Beszallito felvéve
Beszallito felvéve
Beszallito felvéve
Beszallito felvéve
Auto felvéve
Auto felvéve
Auto felvéve
Auto felvéve
```

-Statikus Adattörles: bekéri a beszállító id-ját és ez alapján a táblából kikeresve törli azt.

```
224 public static void StatikusAdattörles() {
225     System.out.println("Törlendő beszállito: ");
226     String id = sc.next();
227     String sqlp="delete from beszállito where raktarkeszlet >0 AND ID like '"+id+"'";
228     if(conn != null) {
229         try {
230             s=conn.createStatement();
231             s.executeUpdate(sqlp);
232             s.close();
233             System.out.println(id + " azonosítójú beszállító törölve!\n");
234         }catch(Exception ex) {
235             System.err.println(ex.getMessage());
236         }
237     }
238
239     System.out.println("Szeretnél még egy műveletet?(I/N)");
240     String igen_nem = sc.next().trim();
241     SzeretnelMegEgyet(igen_nem);
242 }
243
```

```
DINAMIKUSADATTÖRLÉS - DA
SAT
Törlendő beszállito:
2
2 azonosítójú beszállító törölve!
```

Statikus Tábla Módosítás: a „vásárló” táblához hozzáad egy auto id mezőt, mely az autó tábla rendszámával kapcsolódik a táblához.

```
153 public static void StatikusTabelaModositas() {
154     if(conn != null) {}
155     try {
156         String sqlp="alter table vasarlo add(auto_id references auto)";
157         s=conn.createStatement();
158         s.executeUpdate(sqlp);
159         System.out.println("Autó tábla módosítva!\n");
160         s.close();
161     }catch(Exception ex) {
162         System.err.println(ex.getMessage());
163     }
164 }
165
```

```

Sikeres driver regisztrálás!
Sikeres kapcsolódás
Beszálított tábla törölve!
Vasarlo tábla törölve!
Auto tábla törölve!
Beszálított tábla létrejött
Autó tábla létrejött
Vásárló tábla létrejött
Autó tábla módosítva!
Beszálított felvéve
Beszálított felvéve
Beszálított felvéve
Beszálított felvéve
Auto felvéve
Auto felvéve
Auto felvéve
Auto felvéve

```

-Műveletválasztó eljárás: a program fő „navigátora”, ezen belül tudjuk kiválasztani a műveleteket a program futtatása során

```
33 public static void MilyenMuvelet() {
34     System.out.println("Milyen műveletet szeretnél végrehajtani? Az alábbi lehetőségek közül válassz (a megadott formátumokat képes kiértékelni a program):");
35     System.out.println("StatikusAdattörles - SAT\n" + "StatikusLekerdezes - SL\n" + "MódosíthatóKurzor - MK\n"
36     + "DinamikusLekerdezes - DL\n" + "DinamikusAdattörles - DA");
37     String muvelet = sc.next();
38     if(conn != null) {
39         if(muvelet.equals("SAT")) {
40             StatikusAdattorles();
41         }else if(muvelet.equals("SL")) {
42             StatikusLekerdezes();
43         }else if(muvelet.equals("MK")) {
44             ModosithatoKurzor();
45         }else if(muvelet.equals("DL")) {
46             DinamikusLekerdezes();
47         }else if(muvelet.equals("DA")) {
48             DinamikusAdattorles();
49         }else {
50             System.out.println("Rossz választ adtál, kérlek a fentebb felsoroltakat add meg!");
51             MilyenMuvelet();
52         }
53     }
54 }
55
```

```

Milyen műveletet szeretnél végrehajtani? Az alábbi lehetőségek közül válassz (a megadott formátumokat képes kiértékelni a program):
StatikusAdattörles - SAT
StatikusLekerdezes - SL
MódosíthatóKurzor - MK
DinamikusLekerdezes - DL
DinamikusAdattörles - DA

```

Statikus Lekérdezés: ki lehet választani, hogy melyik tábla adatait szeretnénk, ezután pedig a táblák minden adatát kilistázza.

```

243
244 public static void StatikusLekerdzes() {
245     System.out.println("Add meg a táblát, amelynek szeretnéd az adatait:");
246     String tabla = sc.next().trim();
247     if(conn != null) {
248         if(tabla.equals("auto")) {
249             if(conn != null) {
250                 String sqlp="select * from auto";
251                 System.out.println("Rendszám BeszállítóID Típus Szín Motor Ár Évjárat");
252                 System.out.println("-----");
253                 try {
254                     s=conn.createStatement();
255                     s.executeQuery(sqlp);
256                     rs=s.getResultSet();
257                     while(rs.next()) {
258                         String rsz = rs.getString("rsz");
259                         int beszallito_id=rs.getInt("beszallito_id");
260                         String tipus = rs.getString("tipus");
261                         String szin = rs.getString("szin");
262                         String motor = rs.getString("motor");
263                         int ar = rs.getInt("ar");
264                         String evjarat = rs.getString("gyartasi_ido");
265
266                         System.out.println(rsz+" "+beszallito_id+" "+tipus+" "+szin+" "+motor+" "+ar+" "+evjarat);
267                     }
268                     Eldontendo();
269                     rs.close();
270                 }catch(Exception ex) {
271                     System.out.println(ex.getMessage());
272                 }
273             }
274         }
275     }else if(tabla.equals("beszallito")) {
276         if(conn != null) {
277             String sqlp="select * from beszallito";
278             System.out.println("ID Cegnev Székhely Raktarkeszlet Szállítás");
279             System.out.println("-----");
280             try {
281                 s=conn.createStatement();
282                 s.executeQuery(sqlp);
283                 rs=s.getResultSet();
284                 while(rs.next()) {
285                     int id=rs.getInt("ID");
286                     String cegnev = rs.getString("cegnev");
287                     String szekhely = rs.getString("szekhely");
288                     int raktarkeszlet = rs.getInt("raktarkeszlet");
289                     String szallitas = rs.getString("szallitas");
290
291                     System.out.println(id+" "+cegnev+" "+szekhely+" "+raktarkeszlet+" "+szallitas);
292                 }
293                 Eldontendo();
294                 rs.close();
295             }catch(Exception ex) {
296                 System.out.println(ex.getMessage());
297             }
298         }
299     }else if(tabla.equals("vasarlo")){
300         if(conn != null){
301             String sqlp="select * from vasarlo";
302             System.out.println("Adoszam Nev Lakhely Születési idő");
303             System.out.println("-----");
304             try {
305                 s=conn.createStatement();
306                 s.executeQuery(sqlp);
307                 rs=s.getResultSet();
308                 while(rs.next()) {
309                     long adoszam=rs.getLong("adoszam");
310                     String nev = rs.getString("nev");
311                     String lakhely = rs.getString("lakhely");
312                     String szulido = rs.getString("szulido");
313
314                     System.out.println(adoszam+" "+nev+" "+lakhely+" "+szulido);
315                 }
316                 Eldontendo();
317                 rs.close();
318             }catch(Exception ex) {
319                 System.out.println(ex.getMessage());
320             }
321         }
322     }else {
323         System.out.println("Ilyen tábla nincs!");
324         StatikusLekerdzes();
325     }
326     System.out.println("Szeretnél még egy műveletet?(I/N)");
327     String igen_nem = sc.next().trim();
328     SzeretneLMegEgyet(igen_nem);
329 }
330 }
331

```

```

SL
Add meg a táblát, amelynek szeretnéd az adatait:
auto
Rendszám BeszállítóID Típus Szín Motor Ár Évjárat
-----
jdc761 1 Ford Mondeo piros 1.8dt 14000000 2022-01-12 00:00:00.0
jbc767 2 Skoda Octavia kék 2.0 crtdi 16000000 2021-03-12 00:00:00.0
akx455 3 Opel Insignia fehér 1.7dti 19000000 2022-09-21 00:00:00.0
okl767 4 Mazda 6 piros 1.6b 11000000 2022-03-12 00:00:00.0
Szeretnél egy másik tábla adatait?(I/N)

```


-Dinamikus lekérdezés: a beszállító id-ját bekéri és ezután kilistázza az adott beszállító cégnevét, ha a raktárkészlete nagyobb, mint nulla.

```
332 public static void DinamikusLekerdezes() {
333     System.out.println("Beszállító ID-je: ");
334     String id = sc.next().trim();
335     String sqlp = "select cegnev from beszallito where raktarkeszlet>0 AND ID= '"+id+"'";
336     if(conn != null) {
337         try {
338             s=conn.createStatement();
339             s.executeQuery(sqlp);
340             rs=s.getResultSet();
341             while(rs.next()) {
342                 String cegnev = rs.getString("cegnev");
343                 System.out.println("Cégnév: "+cegnev);
344             }
345             rs.close();
346         } catch (Exception ex) {
347             System.out.println(ex.getMessage());
348         }
349     }
350     System.out.println("Szeretnél még egy műveletet?(I/N)");
351     String igen_nem = sc.next().trim();
352     SzeretnelMegEgyet(igen_nem);
353 }
354 }
```

```
DL
Beszállító ID-je:
3
Cégnév: Opel Hungaria Bt.
Szeretnél még egy műveletet?(I/N)
```

-Dinamikus Adattörölés: a beszállítót kitörli a megadott id alapján.

```
369 public static void DinamikusAdattorles() {
370     System.out.println("Törölendő beszállító: ");
371     String id = sc.next();
372     String sqlp="delete from "+ user +".BESZALLITO where ID=?";
373     if(conn != null) {
374         try {
375             ps=conn.prepareStatement(sqlp);
376             ps.setString(1, id);
377             ps.executeUpdate();
378             ps.close();
379             System.out.println(id + " azonosítójú beszállító törölve!\n");
380         } catch (Exception ex) {
381             System.err.println(ex.getMessage());
382         }
383     }
384     System.out.println("Szeretnél még egy műveletet?(I/N)");
385     String igen_nem = sc.next().trim();
386     SzeretnelMegEgyet(igen_nem);
387 }
388 }
389 }
```

```
DinamikusAdattorles -> DL
DA
Törölendő beszállító:
2
2 azonosítójú beszállító törölve!
```

Módosítható Kurzor: meg kell adni egy autó rendszámát és ez alapján megváltoztatja a színét lilára.

```
391 public static void ModosithatoKurzor() {
392     System.out.println("Rendszám: ");
393     String rsz = sc.next().trim();
394     String sqlp = "select szin from auto where rsza = '"+rsz+"'";
395     if(conn != null) {
396         try {
397             s=conn.createStatement(ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_UPDATABLE);
398             rs=s.executeQuery(sqlp);
399             while(rs.next()) {
400                 rs.updateString("szin", "lila");
401                 rs.updateRow();
402             }
403             System.out.println(rsz+" rendszámú autó színe lila lett!");
404         } catch (Exception ex) {
405             System.err.println(ex.getMessage());
406         }
407     }
408     System.out.println("Szeretnél még egy műveletet?(I/N)");
409     String igen_nem = sc.next().trim();
410     SzeretneLMegEgyet(igen_nem);
411 }
412 }
413 }
414 }
```

```
DinamikusAdattörlés - DA
MK
Rendszám:
jdk672
jdk672 rendszámú autó színe lila lett!
Szeretnél még egy műveletet?(I/N)
```