

## Maven projekt és a JDBC használata - 5. gyakorlat

1. A **Mysql Workbench** (esetleg más) hozzon létre egy adatbázist a neve: **cats**, majd egy **cats** nevű táblát.

```
Cats(`id` int(8) NOT NULL AUTO_INCREMENT, `name` varchar(64) DEFAULT NULL,
`breed` varchar(64) DEFAULT NULL, `age` int(8) DEFAULT NULL, `owner` varchar(64)
DEFAULT NULL, PRIMARY KEY (`id`));
```

2. Töltse fel a táblát!

```
INSERT INTO ``(`id`,`name`,`breed`,`age`,`owner`) VALUES (1,'Miska','Perzsa',5, 'Gal
Dora');
```

```
INSERT INTO `cats`(`id`,`name`,`breed`,`age`,`owner`) VALUES (2,'Napsugar','Sziami',2,
'Fekete Ferenc');
```

```
INSERT INTO `cats`(`id`,`name`,`breed`,`age`,`owner`) VALUES (3,'Lucifer','Oroszkek',7,
'Zöld Piroska');
```

```
INSERT INTO `cats`(`id`,`name`,`breed`,`age`,`owner`) VALUES (4,'Kokusz','Bombay',7,
'Fenyvesi Istvan');
```

3. Ellenőrizze le egy **Select** utasítással.

```
select * from cats;
```

4. Hozzon létre egy felhasználót és egy jelszót:

User: bedilasz - ez legyen mindenkinek a **neptunkod**

password: password

```
CREATE USER 'bedilasz'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON *.* TO 'bedilasz'@'localhost' WITH GRANT OPTION;
```

5. Ellenőrizzük le!

```
SELECT User FROM mysql.user;
```

6. Következik az adatbázis műveletek JDBC segítségével. Ajánlott fejlesztői környezet az Eclipse IDE vagy STS.

Hozzon létre a Maven segítségével egy projektet.

GroupID: hu.neptunkod

ArtifactID: mavendemo

Project name: mavendemo

Project location: d:\Maven munkafajlok\mavendemo – a path mindenkinek más.

## Adatbázis műveletek

Nyiss meg a POM.xml és illessze be a következő dependenciát.

```
<dependencies>
  <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.11</version>
  </dependency>
</dependencies>
```

**7.** Az **src/main/java/** mappán belül hozzon létre egy új package-t **CRUD** néven.

**8. Feladat:** Kérdezze le a cats táblát?

A CRUD package-ben hozzon létre egy új class-t **Select** névvel. Írja meg a kódot, amelyben 3 objektumot definiál. (Connection, PreparedStatement, ResultSet)

Adja meg adatbázis elérhetőségét: eszköz:adatbázis tipusa://server:port száma/adatbázis neve

Adja meg az adatbázis egy felhasználóját és annak jelszava:

**user name: neptunkod,**

**pw: password**

A try- catch közé adja meg *kapcsolatot*, a *lekérdezést*, az *eredményhalmaz kiíratását*,

A következő paramétert állítsa be: `preparedStatement.setInt(1, 1);`, tehát azokat írja ki, ahol a macska 1 évnél idősebb.

Futtassa a kódot.

**9.** Változtassa meg **age** paramétert (1, 5) a kódban. Futtassa a kódot. Eredmény: két cica kiírása.

**10.** Az CRUD package-ben hozzon létre egy **Insert** nevű class-t, amely egy rekord beszúrására szolgál – szűrjön be egy rekordot a **cats** táblába.

**10a.** Kérdezze le a táblát.

**11.** A CRUD package-ben hozzon létre egy **Update** class-t, amely a **Sziami** macska gazdáját cseréli le Piros Ferenc.

**11a.** Futtassa a kódot.

**12.** A CRUD package-ben hozzon létre egy **Delete** class-t, amelyben a törlésnek két feltételnek kell teljesülni: a *macska neve* és *gazdája*.

Törölje ki a cicák táblából minden olyan cicát, ahol a *cica neve*: Perzsa és a hozzá tartozó tulajdonos (Gal Dora)

**12a.** Futtassa le a **Select** utasítással a kódot.

---

### ***Tárolt eljárás készítése***

Nyissa meg a *MySQL Workbench* programot, majd a **cats** adatbázishoz, majd hozzon létre egy tárolt eljárást **Create *Stored Procedure*** (tárolt eljárás) a neve: ***storedProcedures***

**13. Feladat:** Készítsen egy egyszerű *tárolt eljárást*, ami visszaadja azokat a rekordokat (cicákat), akik a *gazdájának neve* megegyezik az eljárás *bemenő paraméterével*.

A tárolt eljárást a *MySQL Workbench*-ben készítse el.

```
CREATE DEFINER=`bedilasz`@`localhost` PROCEDURE `get_cats_for_owner`(IN the_owner VARCHAR(64))
```

```
BEGIN
```

```
SELECT * from cats where owner=the_owner;
```

```
END|
```

**13a.** Hozzon létre az *src/main/java* mappán belül egy ***storedProcedure*** package-t, majd ezen belül egy ***InResultSet*** nevű osztályt - ez fogja meghívni a tárolt eljárást.

A *statement* objektumunk most a *CallableStatement*, ennek a segítségével fogja tudni meghívni a tárolt eljárást.

A **call** kulcsszóval és az eljárás nevével előkészítjük a hívást. Értéket adunk az egyetlen bemenő paraméterünknek.

Ez az értéket írjuk át az **owner** mezőt, ***tulajdonosra***.

**14. Feladat:** A következő tárolt eljárás *kettő bemenő paraméterekkel* fog dolgozni.

Egy adott *macskafajta* összes példányának *gazdáját* meg fogja változtatni egy másik gazdára.

A tárolt eljárást a *MySQL Workbench*-ben készítse el.

**14a.** Készítsen a ***storedProcedure*** package-be egy új class-t ***InParameters*** néven. Ez az osztály az adatbázisban lévő összes *Perzsa* nevű macska tulajdonosa Gal Dora lesz.

**15. Feladat:** Készítsen egy olyan tárolt eljárást, ahol van egy *bemenő és egy kimenő* (In és Out) paraméter.

Ez az eljárás egy adott *macskafajta* előfordulásának számát fogja visszaadni.

A tárolt eljárás kap egy *breed*, visszaadja azt a számot, ahány van ebből a *fajból* az adatbázisban.

A tárolt eljárást a MySQL Workbench-ben készítse el.

**15a.** Feladat: Készítsen még egy class-t a *storedProcedure* package-ben **OutParameter** néven.

Ha jól írtuk meg a kódot, akkor *egy számot ad* vissza.

**Jó munkát!!!**