

MySQL adatbázis elérése a kabinetben

linux alatt:

```
#> mysql -u <username>
```

```
mysql> use test      a test nevű adatbázist használjuk
Database changed
```

```
mysql> SELECT * FROM proba;  a test nevű adatbázisban egy
                             proba(int szám, char(20) szoveg) nevű
                             tábla van.
```

```
+-----+-----+-----+-----+
| id | value | szám | szoveg |
+-----+-----+-----+-----+
| 0 | value | 1 | szoveg |
| 1 | ertek | 42 | blabla |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

JDBC: (Linuxon)

Forrás a /pub/Adatbazisok/ könyvtárban: Jdbc.java
(Át kell nevezni Jdbc.java-ra)

Fordítás:

```
javac -classpath mysql-connector-java-5.0.8-bin.jar Jdbc.java
```

Futtatás:

```
java -classpath mysql-connector-java-5.0.8-bin.jar:$CLASSPATH Jdbc
```

Eredmény:

```
URL: jdbc:mysql://home.cab.u-szeged.hu:3306/test
Connection: com.mysql.jdbc.Connection@1d2068d
Display all results:
1 | szoveg
42 | blabla
```

MySQL elérése JDBC-vel (a példa program magyarázata):

Az adatbázis eléréséhez és az eredmények eltárolásához létre kell hozni néhány objektumot. Először regisztrálni kell a JDBC driver-t a MySQL-hez, ezzel mondjuk meg neki, hogy milyen driver-t használunk.

```
Class.forName("com.mysql.jdbc.Driver");
```

Ez nincs benne alapból a Java-ban, ehhez kell a `mysql-connector-java-5.0.8-bin.jar` fájl.

Az adatbázis URL címét érdemes előre egy sztringben eltárolni.

```
String url = "jdbc:mysql://home.cab.u-szeged.hu:3306/test";
```

Majd felhasználva ezt a sztringet kapcsolódhatunk az adatbázishoz, a második és harmadik paraméter rendre a felhasználónév és jelszó.

```
Connection con = DriverManager.getConnection(url, "", "");
```

A `Statement` objektummal manipulálhatjuk az adatbázisunkat, pl. megadhatunk egy SQL lekérdezést. A példa kedvéért egy egyszerű lekérdezést hajtunk végre.

```
stmt = con.createStatement();  
rs = stmt.executeQuery("SELECT * from proba");
```

az `executeQuery()` függvény egy `ResultSet` objektumot ad vissza, amely a lekérdezés eredményét (rekordokat) tároló halmaz.

A rekordokat a következőképpen tudjuk kiírni a képernyőre:

```
while(rs.next()){  
    String szam = rs.getString("Szam");  
    String szoveg = rs.getString("szoveg");  
    System.out.println(szam + " | " + szoveg);  
}
```

A tranzakció végén mindig le kell zárni a kapcsolatot:

```
con.close();
```

Az adatbázis-tranzakciókat mindig `try-catch` blokkba kell rakni, mivel egyáltalán nem biztos, hogy sikerül kapcsolódni az adatbázishoz, vagy, hogy a lekérdezés sikeres lesz. A `try-catch` blokk arra szolgál, hogy a hibákat le tudjuk kezelni.