

# JDBC 4.0 annotációk

Talán sokunk nem tudja még (nem nézett utána vagy nem volt eddig szüksége rá), de a JDBC 4.0 sok újdonsága között az egyik legnagyobb újítás a BaseQuery interfész lesz, amelyet annotációkkal lehet majd kezelni. Az annotációk az ötös Java újításai között jelent meg, a BaseQuery interfész pedig (majd) a hatos Java és a JDBC 4.0 közös gyermeke lesz, erről olvashatunk cikket a [Worx.com](http://Worx.com) írásai között. A lekérdezéseinket egy BaseQuery interfészt kiterjeszt osztályba kell tennünk az alábbi módon:

## QueryAnnotationExample.java

```
public interface QueryAnnotationExample extends BaseQuery
{
    @Select(sql="SELECT ID, MODEL, MODEL_YEAR FROM CAR WHERE MODEL_YEAR = ?1")
    public DataSet<Car> getCarsModelYear( String year );
}
```

Természetesen a megoldás épít a JPA már meglévő annotációira, tehát kell egy Car osztály, amely JPA compliant állapotban van.

## Java forrás

```
QueryAnnotationExample qae = null;
try
{
    String url = "jdbc:derby://localhost:1527/wrox;create=true";
    Connection con = DriverManager.getConnection(url , "APP", "password");
    qae = con.createQueryObject(QueryAnnotationExample.class);
}
catch (SQLException e)
{
    e.printStackTrace();
}
Collection<Car> cars = qae.getCarsModelYear("1999");
```

Amint látjuk, az adatbázis kapcsolódás azonos módon történik, a lekérdezés folyamata azonban teljesen újszerű lesz. A lekérdezés során ugyanis létrehozunk egy annotációkkal megtdelt osztályunkból egy példányt, amelyből majd lekérdezzük, s a lekérdezés eredménye nem egy ResultSet lesz, hanem egy OOP szempontoknak teljes mértékben megfelel JPA entitás kollekció. Ebből a kollekcióból simán le tudunk kérni egy listát, át tudjuk adni a klienseknek XML szerializációval, satöbbi:

## Java forrás

```
for ( Car c : cars)
{
    System.out.println(" car id=" + c.getId() + " model="+c.getModel() +" year="+ c.getYear() );
}
```