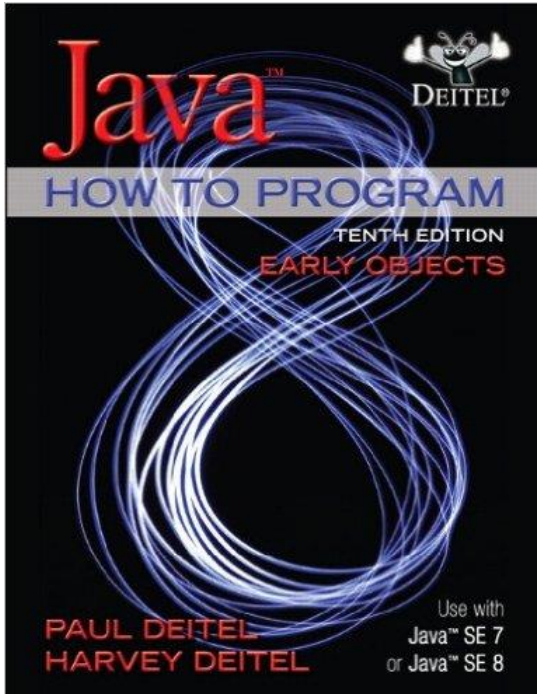


Adatbázis-kezelés Java alkalmazásokban

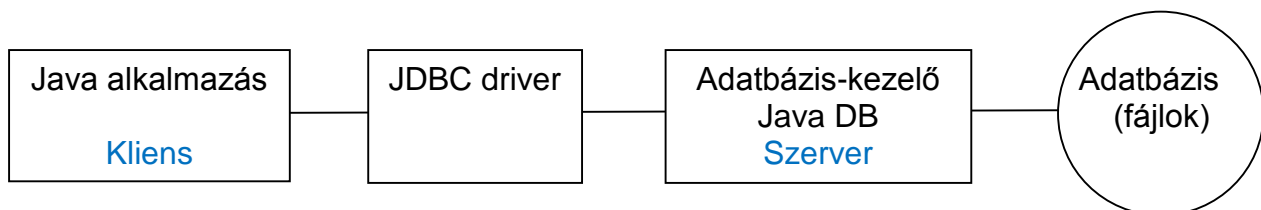
Az anyag példaprogramjai az alábbi könyvből származnak:



Java™ How To Program (Early Objects), Tenth Edition
by Harvey Deitel, Paul Deitel
Publisher: Prentice Hall
Published: February 2014
Chapter 24: Accessing Databases with JDBC

A könyvben szereplő Java forráskódokat és a Java Db adatbázisokat a NetBeans környezetbe integráltam. Az adatbázisok szerkezetét az Enterprise Architekt CASE eszközzel diagramokon ábrázoltam. A NetBeans adatbázis adminisztrátor felületéről leírást készítettem.

Kliens-szerver architektúra



A Java DB telepítése

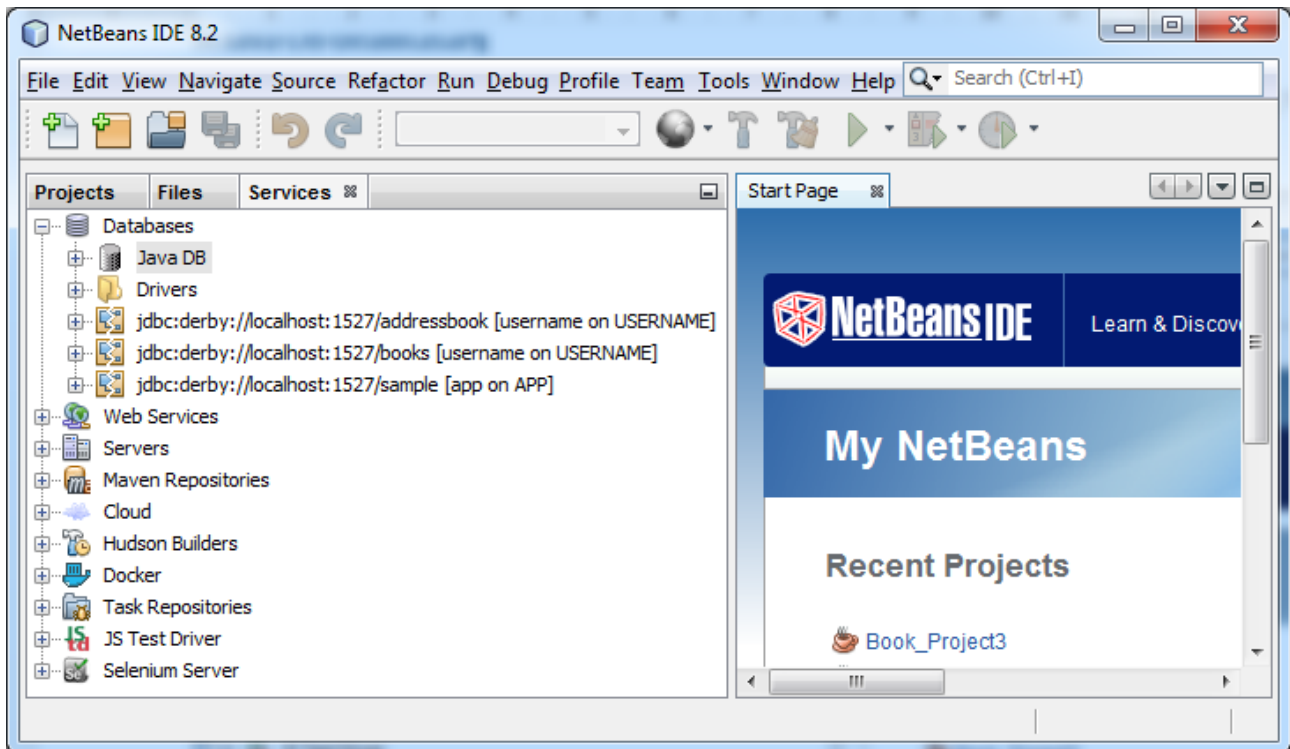
A NetBeans EE (Enterprise Edition) változatát telepítsük! A telepítő futtatása során jelöljük ki a Glassfish alkalmazásszerver telepítését! Magát a Glassfish-t nem fogjuk használni. A Glassfish-t azért kell telepíteni, mert vele együtt települ a Java DB relációs adatbázis-kezelő. A Java DB-t nevezik Apache Derby-nek is.

A NetBeans IDE grafikus felülete

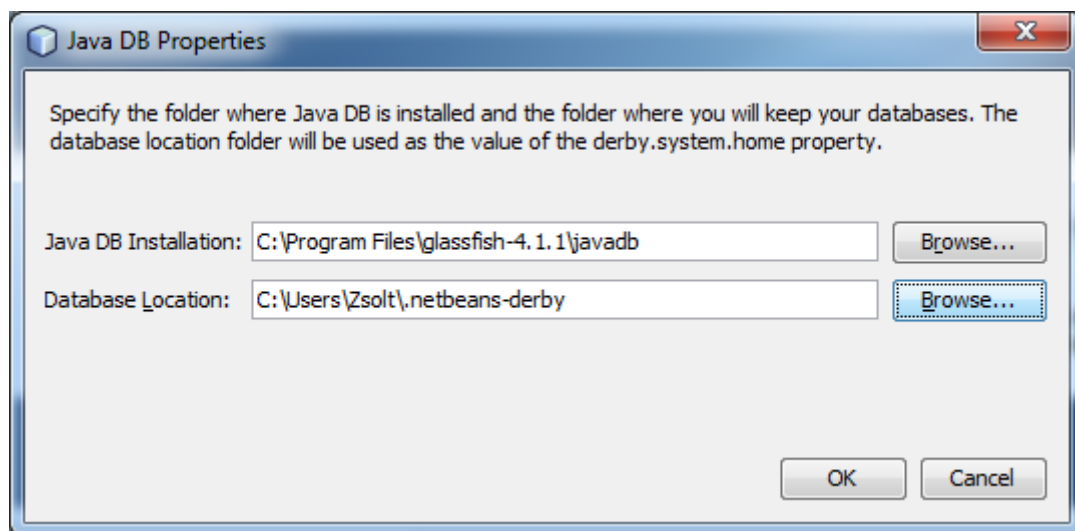
A Java DB eredetileg csak parancssoros felhasználói felületet biztosít az adatbázisok menedzseléséhez. A NetBeans-ben ezeket a funkciókat grafikus felhasználói felületen keresztül tudjuk elérni, párbeszéd-ablakok segítségével.

A Java DB beállítása

Kattintsunk a NetBeans-ben a **Services** fülre! Ha nem látszik a **Services** fül, akkor a **Window | Services** menüponttal tudjuk megjeleníteni.



Kattintsunk a jobb egérgombbal a **Java DB** ikonra, és válasszuk ki a **Properties...** menüpontot!



A **Java DB Installation** mezőben állítsuk be a Java DB elérési útvonalát! A Glassfish alá telepített változatot válasszuk ki!

A **Database Location** mezőben beállíthatjuk azt a könyvtárat, ahol tárolni akarjuk az adatbázis tartalmazó fájlokat. Használhatjuk az alapértelmezetten beállított könyvtárat is.

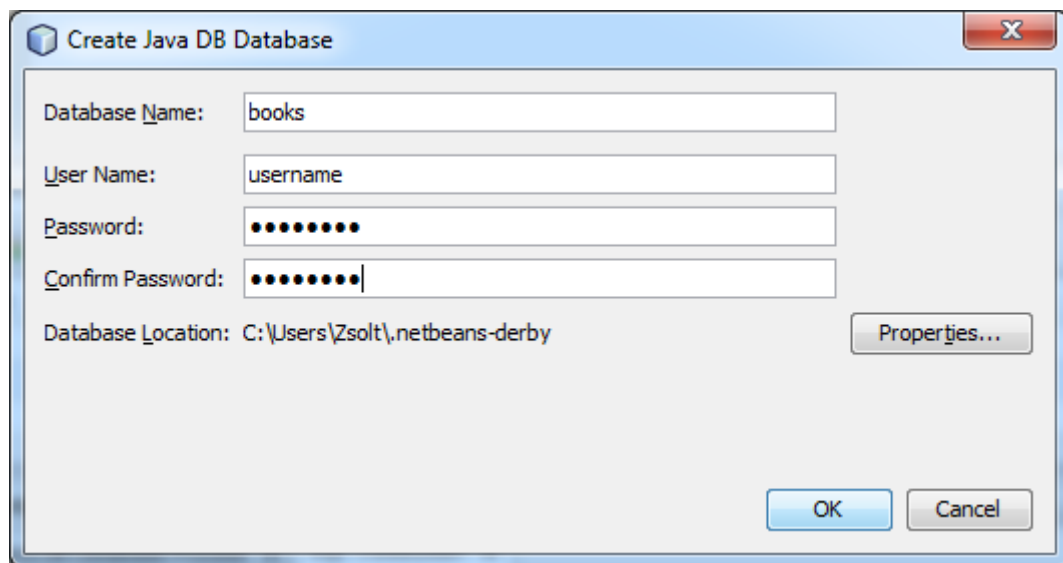
A Java DB adatbázis-szerver program elindítása

Kattintsunk rá a **Java DB** ikonra, és válasszuk ki a **Start Server** menüpontot!
Az **Output** ablak tájékoztat, hogy az adatbázis-szerver program elindult.

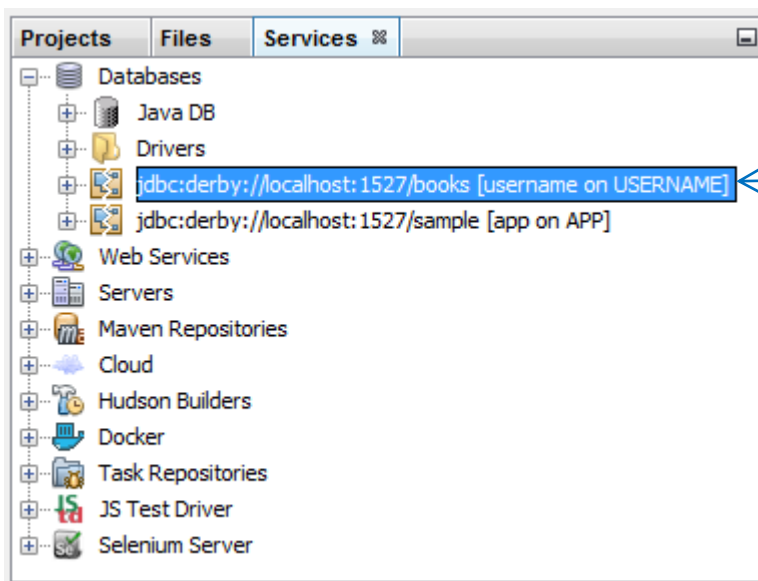
```
Output - Java DB Database Process
Thu Mar 16 00:50:54 CET 2017 : DRDA_SecurityInstalled.I
[Thu Mar 16 00:50:55 CET 2017 : Apache Derby Hálózati kiszolgáló - 10.10.2.0 - (1582446)
  elindult és kész a kapcsolatok fogadására a(z) {3} 1527 portján
```

Új adatbázis létrehozása

Kattintsunk rá a **Java DB** ikonra, és válasszuk ki a **Create Database...** menüpontot!

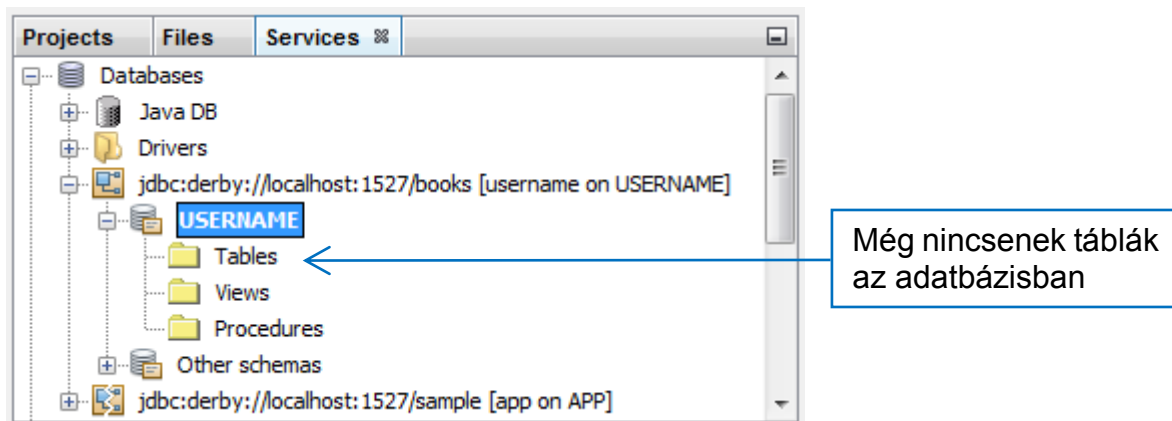


Töltsük ki a mezőket a fenti ábrán látható szövegekkel! A **Password** mezőbe az egyszerűség kedvéért most a password szót írjuk be! Az **OK** gomb megnyomásával hozzuk létre az adatbázist!



Megjelenik a books adatbázis ikonja

Kattintsunk rá a jobb egérgombbal a `books` adatbázis ikonjára, és válasszuk ki a **Connect...** menüpontot! A + ikonok nyomogatásával nyissuk ki a `books` csomópontához tartozó ágakat!



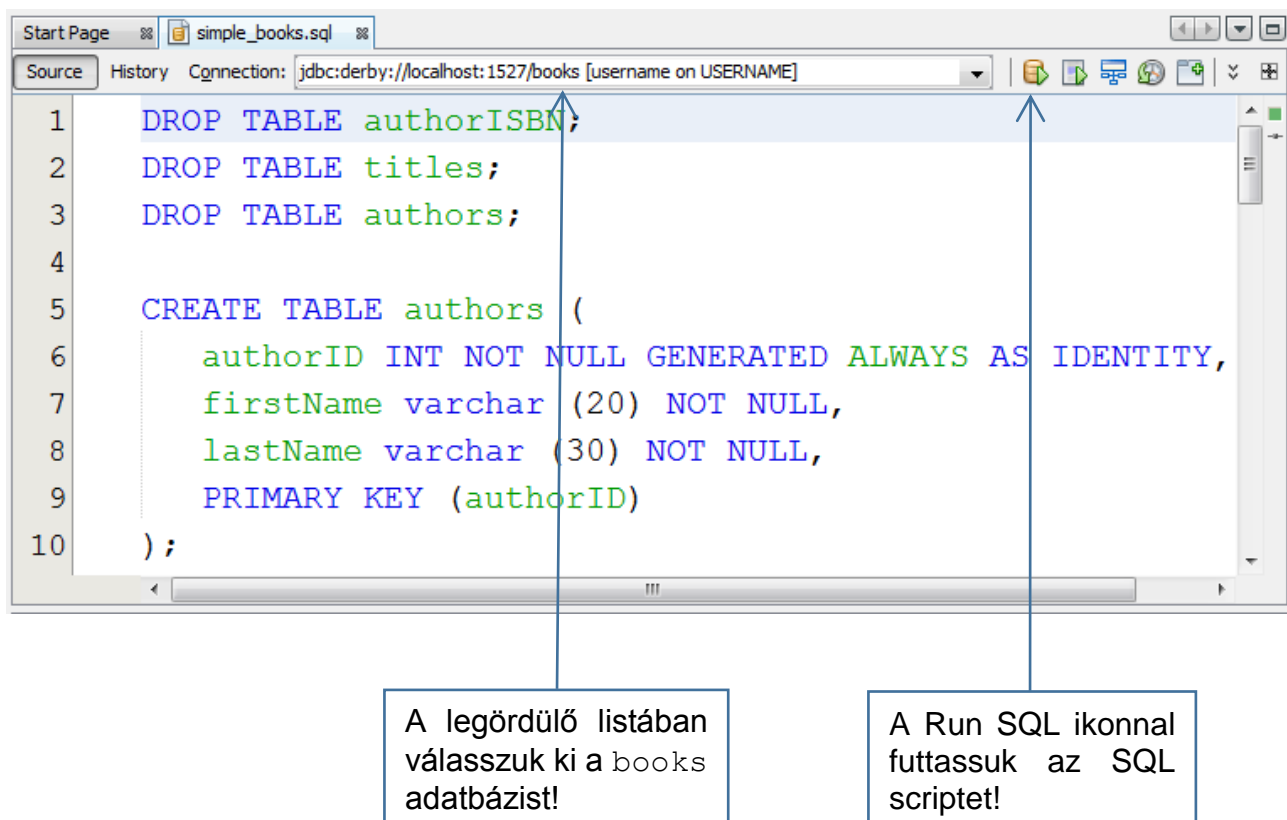
Táblák létrehozása az adatbázisban

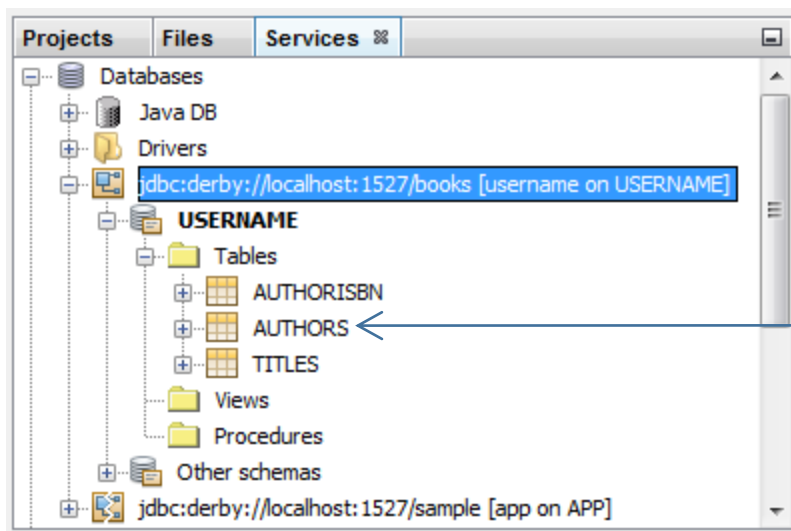
Új táblát kétféle módon hozhatunk létre:

Egyik lehetőség, hogy a jobb egérgombbal a **Tables** ikonra kattintunk, és a **Create Table...** menüpontot választjuk. A megjelenő párbeszédablakban megadhatjuk az új tábla szerkezetét.

A másik lehetőség, hogy a `CREATE TABLE` SQL paranccsal hozzuk létre az új táblát. Az SQL parancsot (vagy parancsokat) begépelhetjük egy `.sql` kiterjesztésű szöveges állományba, ezt SQL scriptnek nevezzük. Az SQL scriptet megnyithatjuk a NetBeans-ben és futtathatjuk.

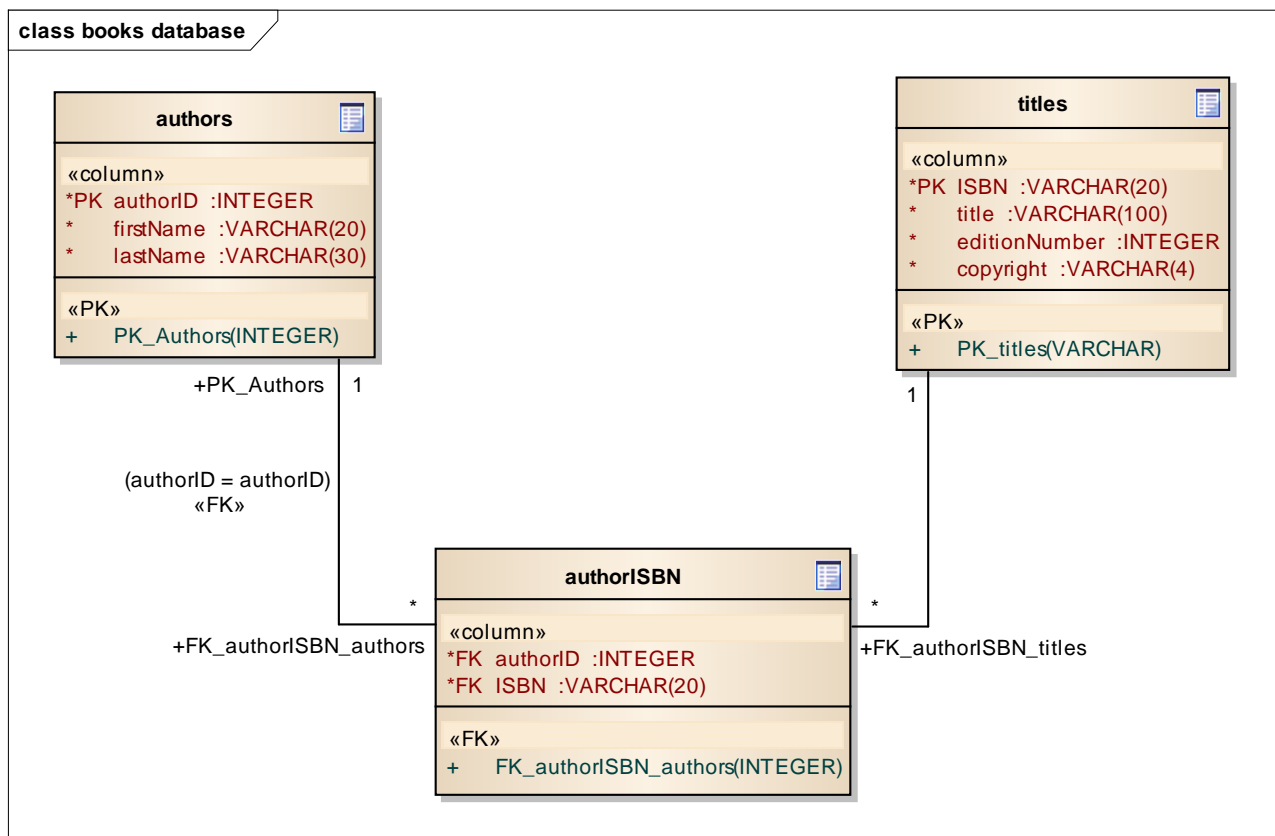
A NetBeans **File | Open File...** menüpontjával nyissuk meg a `simple_books.sql` fájlt!





A books adatbázisban három tábla jött létre.

A books adatbázis szerkezete (Enterprise Architect diagram):



Tábla feltöltése adatokkal

Egy táblába adatokat vihetünk fel SQL script segítségével. Az előbb lefuttatott `simple_books.sql` fájl tartalmaz SQL parancsokat, melyek feltöltik adatokkal a táblákat.

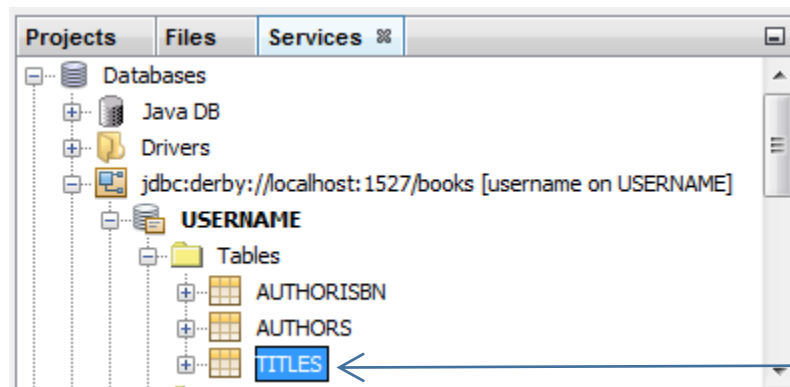
A books adatbázisba felvitt adatok:

books adatbázis			sok-sok kapcsolat			
authors			titles			
authorID	firstName	lastName	ISBN	title	editionNumber	copyright
1	Aladár	Kiss	11111	Java	1	2016
2	Béla	Nagy	22222	C++	1	2015
3	Katalin	Molnár	33333	Adatbázis-kezelés	2	2010
(authorID: auto increment)			44444	Szoftverfejlesztés	3	2000

KAPCSOLÓ TÁBLA	
authorISBN	
authorID	ISBN
1	11111
1	22222
1	33333
2	33333
3	33333
2	44444

Aladár Kiss: Java
Aladár Kiss: C++
Aladár Kiss, Béla Nagy, Katalin Molnár: Adatbázis-kezelés
Béla Nagy: Szoftverfejlesztés

Tábla listázása



Kattintsunk a jobb egérgombbal a TITLES ikonra, és válasszuk a **View Data...** menüpontot!

SELECT * FROM USERNAME.TI... ☒				
Max. rows: 100		Fetched Rows: 4		Matching Rows:
#	ISBN	TITLE	EDITIONNUMBER	COPYRIGHT
1	11111	Java	1	2016
2	22222	C++	1	2015
3	33333	Adatbázis-kezelés	2	2010
4	44444	Szoftverfejlesztés	3	2000

Eredménytábla lekérdezése SQL parancssal

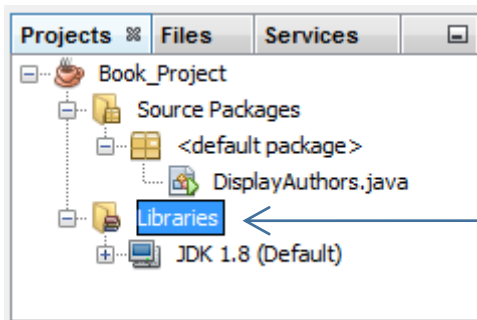
Ha egy lekérdezésben több táblát szeretnénk összekapcsolni, SQL parancsot használhatunk. A **File | Open File...** menüponttal nyissuk meg a szerzok_cimek.sql scriptet, majd futassuk! (Sok-sok kapcsolat megvalósítása kapcsolótáblával.)

```
Source History Connection: jdbc:derby://localhost:1527/books [username on USERNAME]
1 SELECT firstName, lastName, titles.ISBN, title
2 FROM authors
3 INNER JOIN authorISBN ON authors.authorID = authorISBN.authorID
4 INNER JOIN titles ON authorISBN.ISBN = titles.ISBN
```

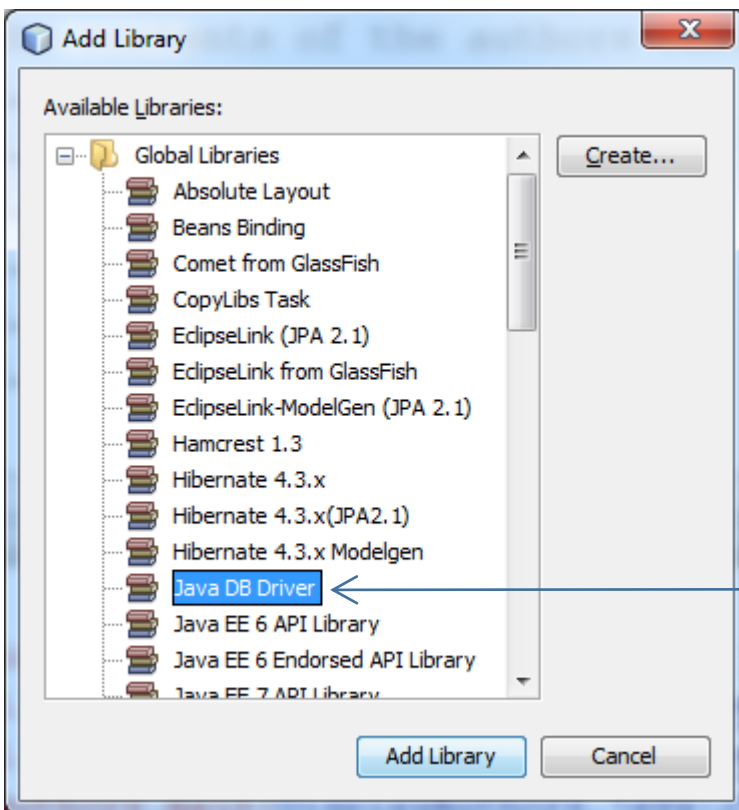
#	FIRSTNAME	LASTNAME	ISBN	TITLE
1	Aladár	Kiss	11111	Java
2	Aladár	Kiss	22222	C++
3	Aladár	Kiss	33333	Adatbázis-kezelés
4	Béla	Nagy	33333	Adatbázis-kezelés
5	Katalin	Molnár	33333	Adatbázis-kezelés
6	Béla	Nagy	44444	Szoftverfejlesztés

Java alkalmazások

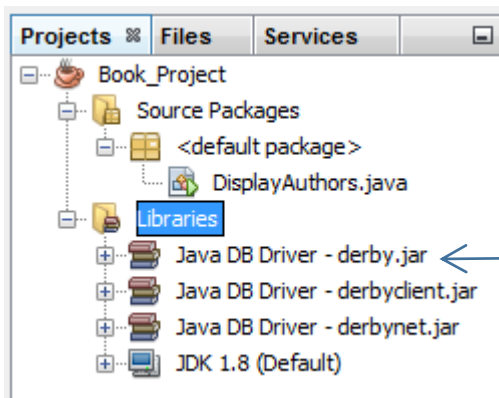
A JDBC driver hozzáadása egy új NetBeans projekthez



Kattintsunk a jobb egérgombbal a **Libraries** ikonra, és válasszuk az **Add Library...** menüpontot!



Válasszuk ki a Java DB Driver-t!

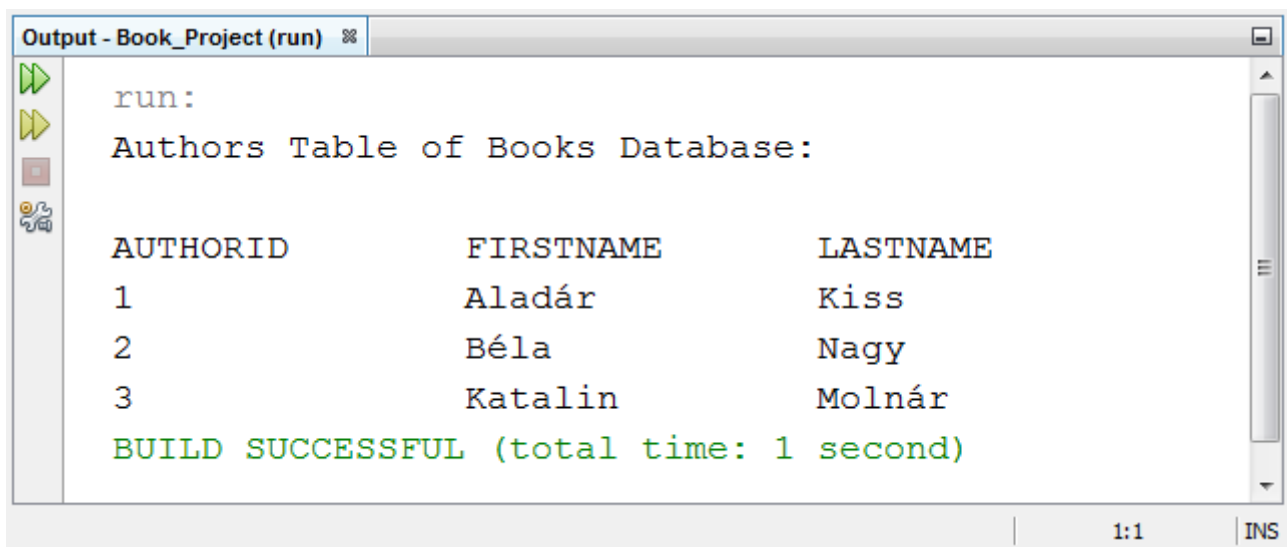


A hozzáadott könyvtárak

Book_Project - DisplayAuthors.java

- Készítsünk a NetBeans-ben egy új projektet, Book_Project néven!
- A projekt készítő 2. lépésében vegyük ki a **Create Main Class** pipát, azaz ne generáljunk indító osztályt! Így a projekt src mappája kezdetben üres marad.
- A projekt src mappájába másoljuk be a DisplayAuthors.java forrásfájlt!
- Adjuk hozzá a projekthez a JDBC drivert!
- Ha nem fut a Java DB adatbázis-szerver, akkor indítsuk el!
- Feltételezzük, hogy a books adatbázist már létrehoztuk és feltöltöttük adatokkal.
- Futtassuk a projektet!
- Futtatáskor állítsuk be **main class**-ként a DisplayAuthors.java-t!

A program kilistázza a konzolra az Authors tábla sorait.



```

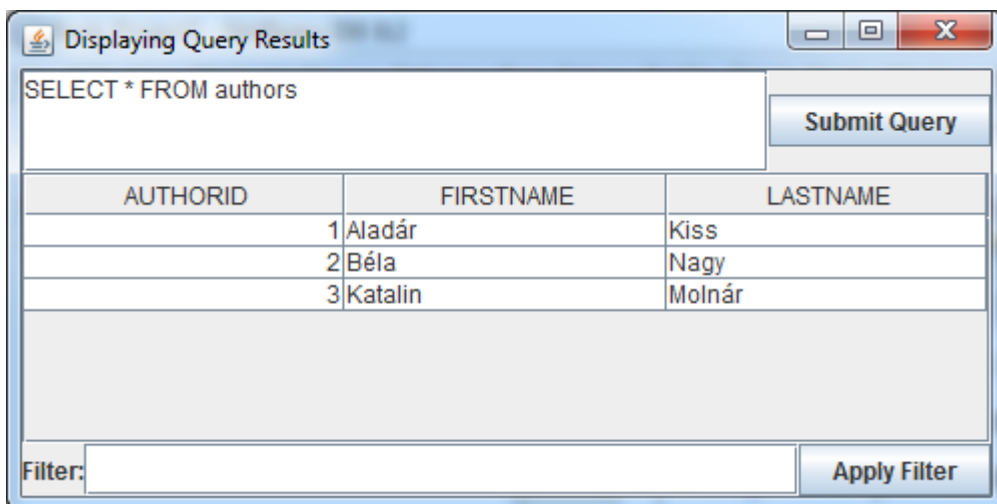
run:
Authors Table of Books Database:

AUTHORID      FIRSTNAME      LASTNAME
1              Aladár         Kiss
2              Béla           Nagy
3              Katalin        Molnár

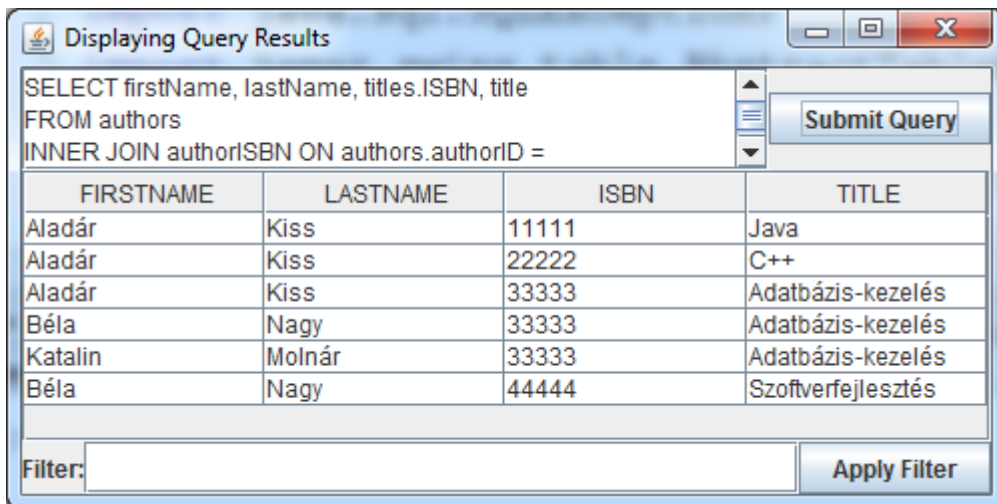
BUILD SUCCESSFUL (total time: 1 second)
    
```

Book_Project2 ResultSetTableModel.java, DisplayQueryResults.java

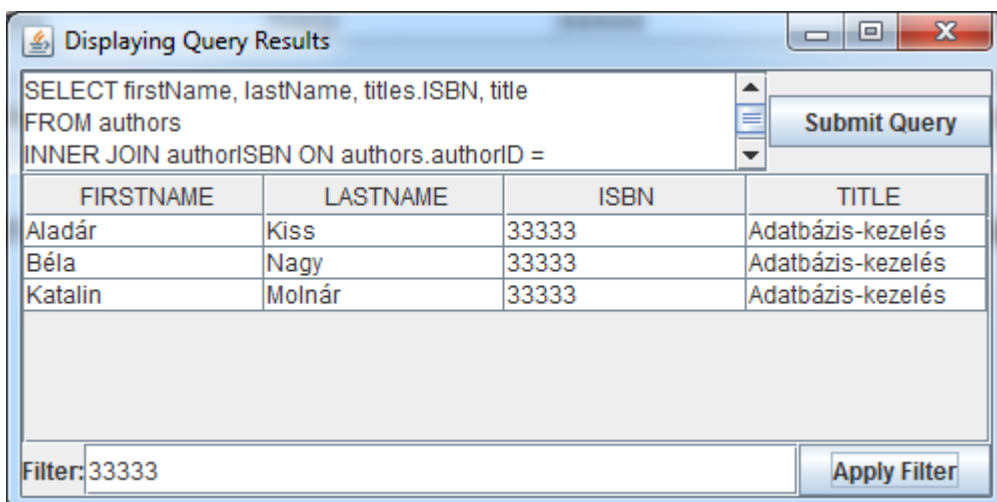
- Készítsünk egy új projektet Book_Project2 néven!
- Tegyük bele a ResultSetTableModel.java, DisplayQueryResults.java fájlokat!



- Nyissuk meg a Jegyzetömbbel a `szerezok_cimek.sql` scriptet!
- Az SQL scriptet a vágólapon keresztül másoljuk be az alkalmazás felső részén lévő beviteli mezőbe!
- Nyomjuk meg a **Submit Query** gombot!



- A **Filter** mezőbe írjuk be: 33333, majd nyomjuk meg az **Apply Filter** gombot!



Book_Project3 – JdbcRowSetTest.java

- Készítsünk egy új projektet `Book_Project3` néven!
- Tegyük bele a `JdbcRowSetTest.java` fájlt!

Ez a `Book_Projekt1` másik megoldása, a `JdbcRowSet` osztály használatával.

