

## JDBC – Java Database Connectivity

JDBC reprezintă un API pentru realizarea operațiilor cu baze de date (relaționale) din aplicații Java. Clasele JDBC se găsesc în pachetul `java.sql`, existând și o extensie în `javax.sql`.

În general prelucrarea unei operații SQL cu JDBC urmează o succesiune de pași:

- Stabilirea unei conexiuni la sursa de date (bază de date, sistem de fișiere etc).
- Crearea unei fraze SQL (un obiect `Statement` se crează cu ajutorul unui obiect `Connection`).
- Executarea interogării (regăsire date, apel procedură stocată, modificare date).
- Prelucrarea obiectului `ResultSet` rezultat.
- Închiderea conexiunii, pentru eliberarea resurselor folosite.

Există 3 feluri de fraze SQL:

- `Statement`: fraze SQL simple, fără parametri.
- `PreparedStatement`: extind `Statement` și pot conține parametri de intrare.
- `CallableStatement`: extind `PreparedStatement` și se folosesc pentru a executa proceduri stocate ce pot avea atât parametri de intrare cât și parametri de ieșire.

Execuția unei interogări se face prin apelul unor metode ale obiectului `Statement`:

- `execute`: poate returna unul sau mai multe obiecte `ResultSet`, care pot fi obținute prin apelarea repetată a metodei `Statement.getResultSet`.
- `executeQuery`: returnează u obiect `ResultSet`.
- `executeUpdate`: returnează numărul de linii afectate de fraza SQL de tip `INSERT`, `DELETE`, `UPDATE`.

Operațiile uzuale sunt descrise de următoarea secvență de cod:

```
try {
    stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(query);
    while (rs.next()) {
        String coffeeName = rs.getString("COF_NAME");
        int supplierID = rs.getInt("SUP_ID");
        float price = rs.getFloat("PRICE");
        int sales = rs.getInt("SALES");
        int total = rs.getInt("TOTAL");
        System.out.println(coffeeName + "\t" + supplierID + "\t" + price + "\t" + sales +
"\t" + total);
    }
}
```

Specificarea sursei de date se face prin indicarea unui șir de caractere de tip URL, cu următoarea structură:

- Pentru MySQL:

```
jdbc:mysql://[host] [, failoverhost... ][:port]/[database] [?propertyName1 [=propertyValue1] [&
propertyName2] [=propertyValue2] ...
```

- unde `host:port` implicit `127.0.0.1:3306`.

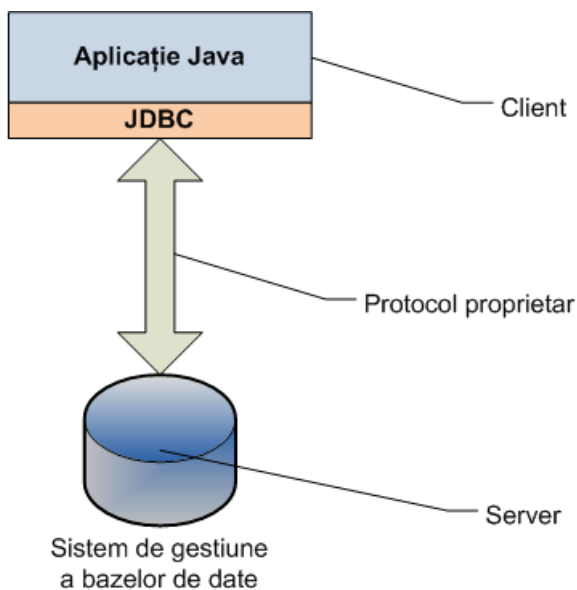
- Pentru Java DB:

```
jdbc:derby: [subsubprotocol:] [databaseName] [;attribute=value] *
```

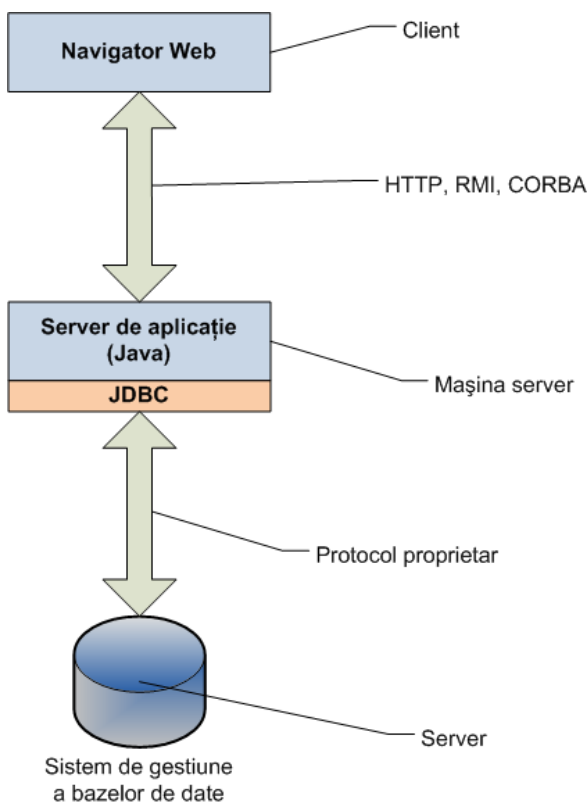
- `subsubprotocol` (unde e baza de date: într-un director, în memory, in class path, în JAR).
- `databaseName` is the name of the database to connect to.
- `attribute=value` listă opțională de attribute, pentru indicarea unor task-uri:
  - Crearea bazei de date din URL.
  - Criptarea bazei de date din URL.
  - Director pentru log.

- Utilizator și parolă de conectare.

Arhitectura JDBC în modelul cu 2 straturi este următoarea:



Arhitectura JDBC în modelul cu 2 straturi este următoarea:



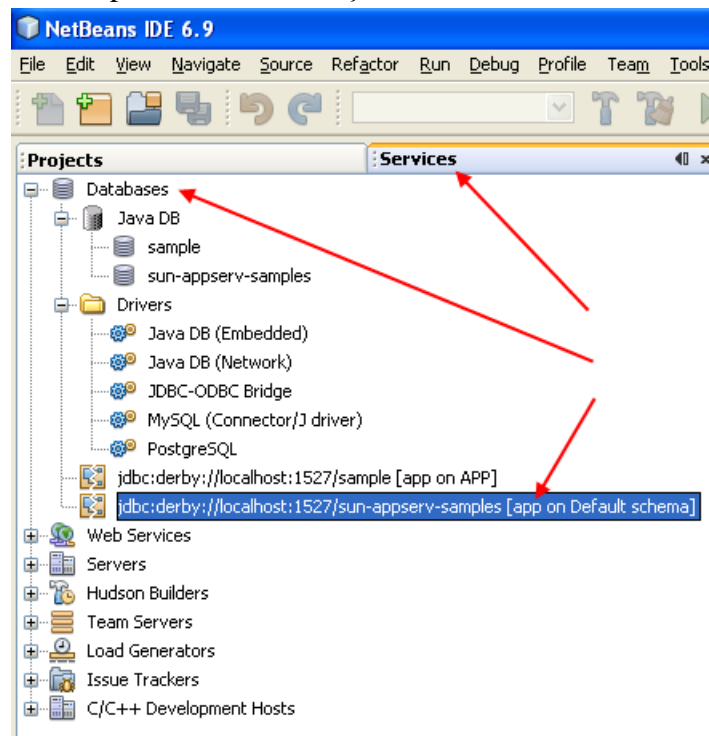
## 1. Derby (Java DB)

Derby este un motor de bază de date relațională pur Java, care folosește standard SQL și JDBC ca API, oferind o instanță încorporată în aplicație sau de server de rețea și o serie de utilitare în linie de comandă:

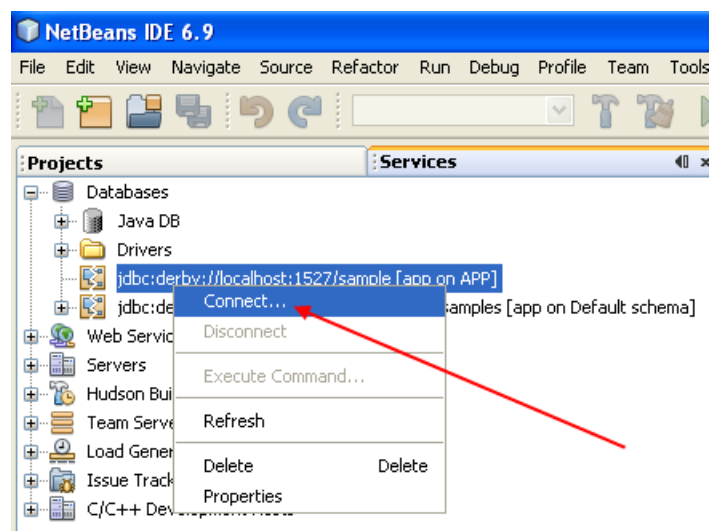
- ij (SQL scripting).
- dblook (schema dump).
- sysinfo (system info).

JDK 1.6 instalează un astfel de server în locația " C:\Program Files\Sun\JavaDB".

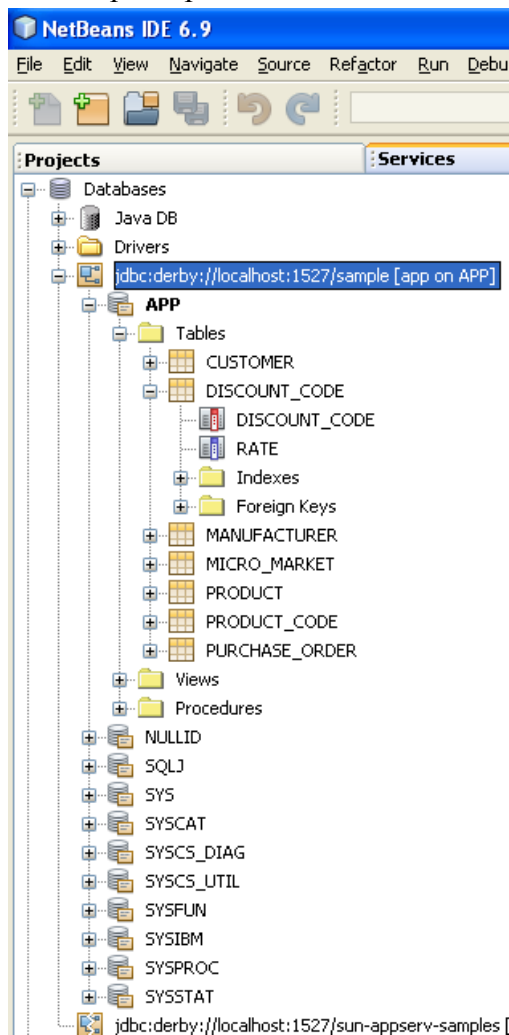
Mediul de dezvoltare NetBeans permite accesarea și controlarea acestui server:



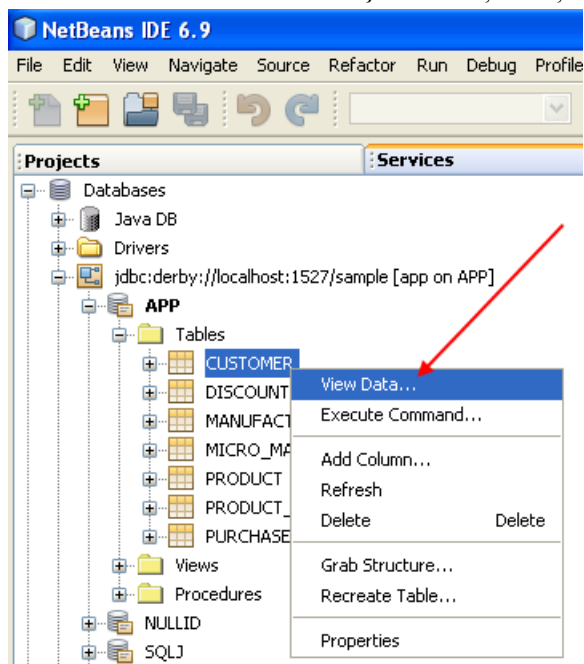
Startați manual acest server:



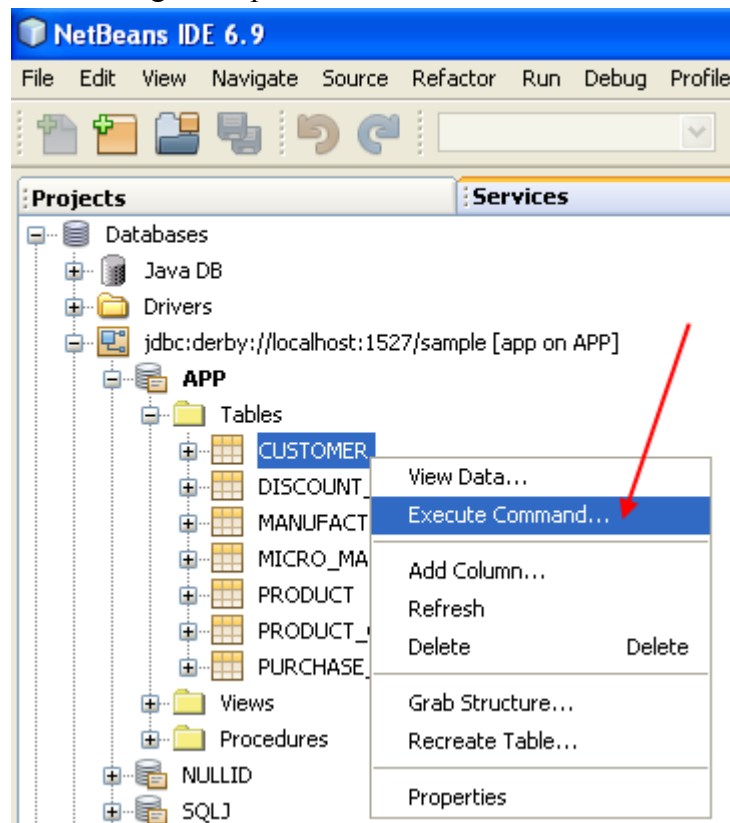
Analizați schemele care apar și tabelele principale ale acestora:



Observați modalitatea de vizualizare a datelor din tabele și analizați conținutul acestora:



Executați manual o serie de interogări asupra datelor existente:



## 2. IJ

Realizați setările de mediu necesare (variabilele DERBY\_HOME, PATH) și lansați utilitarul ij:

```
C:\ C:\WINDOWS\system32\cmd.exe - ij
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\GG>ij
ij version 10.4
ij> _
```

Comanda help oferă o serie de informații suplimentare:

```

C:\WINDOWS\system32\cmd.exe - ij
C:\Documents and Settings\GG>ij
ij version 10.4
ij> help;

Supported commands include:

PROTOCOL 'JDBC protocol' [ AS ident ];
-- sets a default or named protocol
DRIVER 'class for driver';
-- loads the named class
CONNECT 'url for database' [ PROTOCOL namedProtocol ] [ AS connectionName ];
-- connects to database URL
-- and may assign identifier
SET CONNECTION connectionName;
-- switches to the specified connection
SHOW CONNECTIONS;
-- lists all connections
AUTOCOMMIT [ ON | OFF ];
-- sets autocommit mode for the connection
DISCONNECT [ CURRENT | connectionName | ALL ];
-- drop current, named, or all connections;
-- the default is CURRENT

SHOW SCHEMAS;
-- lists all schemas in the current database
SHOW [ TABLES | VIEWS | PROCEDURES | SYNONYMS ] < IN schema >;
-- lists tables, views, procedures or synonyms
SHOW INDEXES < IN schema | FROM table >;
-- lists indexes in a schema, or for a table
DESCRIBE name;
-- lists columns in the named table

COMMIT;
-- commits the current transaction
ROLLBACK;
-- rolls back the current transaction

PREPARE name AS 'SQL-J text';
-- prepares the SQL-J text
EXECUTE < name | 'SQL-J text' > [ USING < name | 'SQL-J text' > ] ;
-- executes the statement with parameter
-- values from the USING result set row

REMOVE name;
-- removes the named previously prepared statement

RUN 'filename';
-- run commands from the named file

ELAPSED TIME [ ON | OFF ];
-- sets elapsed time mode for ij
MAXIMUM DISPLAY WIDTH integerValue;
-- sets the maximum display width for
-- each column to integerValue

ASYNC name 'SQL-J text';
-- run the command in another thread
WAIT FOR name;
-- wait for result of ASYNC'd command

GET [ SCROLL INSENSITIVE ] CURSOR name AS 'SQL-J query';
-- gets a cursor (JDBC result set) on the query
-- SCROLL cursors are only available
-- in JDBC 2.0 and higher.
-- (Cursor scroll type is ignored in JDBC 1.X.)
NEXT name;
-- gets the next row from the named cursor
FIRST name;
-- gets the first row from the named scroll cursor
LAST name;
-- gets the last row from the named scroll cursor
PREVIOUS name;
-- gets the previous row from the named scroll cursor
ABSOLUTE integerValue name;
-- positions the named scroll cursor at the absolute row number
-- (A negative number denotes position from the last row.)
RELATIVE integerValue name;
-- positions the named scroll cursor relative to the current row
-- (integer is number of rows)
AFTER LAST name;
-- positions the named scroll cursor after the last row
BEFORE FIRST name;
-- positions the named scroll cursor before the first row
GETCURRENTROWNUMBER name;
-- returns the row number for the current position of the named scroll cursor
-- (0 is returned when the cursor is not positioned on a row.)
CLOSE name;
-- closes the named cursor
LOCALIZED DISPLAY [ ON | OFF ];
-- controls locale sensitive data representation

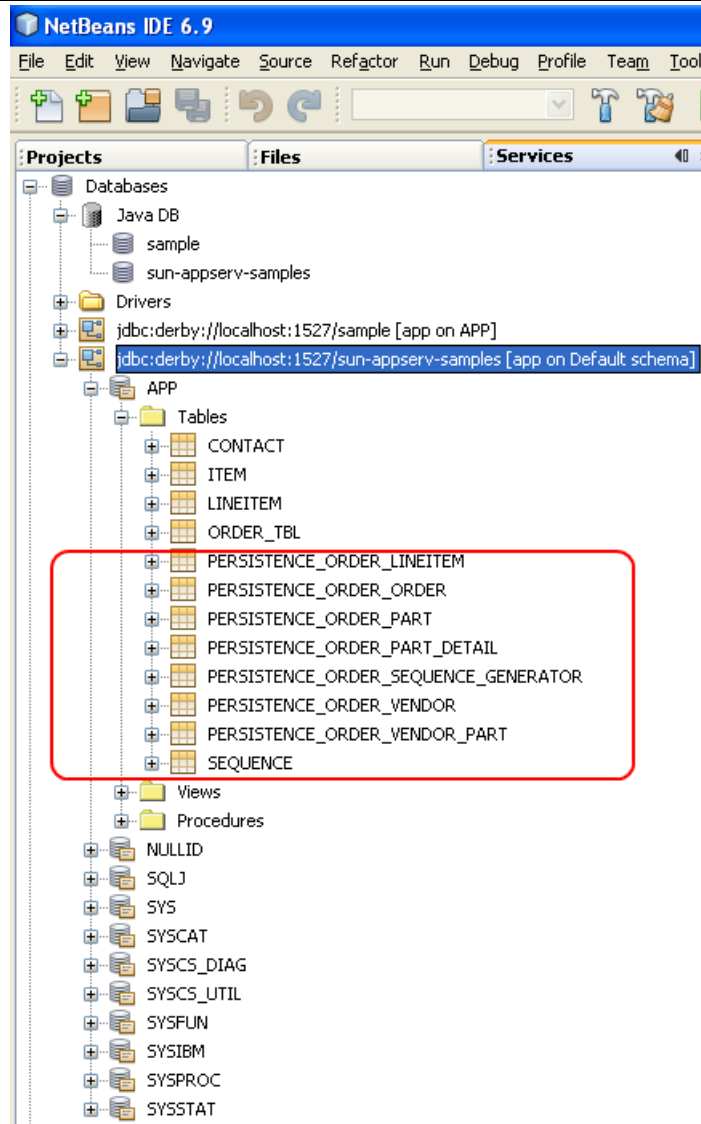
EXIT;
-- exits ij
HELP;
-- shows this message

Any unrecognized commands are treated as potential SQL-J commands and executed directly.
ij> _

```

### 3. Aplicația order

Deschideți în NetBeans aplicația "order" folosită la laboratorul anterior. Identificați tabelele asociate entităților aplicației.



Lansați aplicația și adăugați comenzi și produse.

### Order Java Persistence Example - DAI Lab

#### View All Orders

Order ID	Shipment Info	Status	Last Updated	Items	Value	Discount	Sum	Actions
1111	333 New Court, New City, CA 90000	N	Fri Nov 19 05:18:11 EET 2010	3	738.53	10%	664.677	Delete
1234	Test lab DAI	N	Fri Nov 19 05:21:19 EET 2010	2	117.9900000000000001	20%	94.3920000000000001	Delete
4312	333 New Court, New City, CA 90000	N	Fri Nov 19 05:18:11 EET 2010	3	2011.44	0%	2011.44	Delete

#### Create New Order

Order fields

Order ID:

Shipment Info:

Status: Pending

Discount: 0%

#### Find Vendors

Search for vendors:

## Line Items for Order 1234

Item ID	Part Number	Vendor Part Number	Revision	Quantity	Remove From Order
1	9876-4321-02	2	2	4	<input type="button" value="Remove 1"/>
2	5456-6789-03	3	3	1	<input type="button" value="Remove 1"/>

Available items

Part Number	Vendor Part Number	Revision	Add To Order
1234-5678-01	1	1	<input type="button" value="Add"/>
5456-6789-03	3	3	<input type="button" value="Add"/>
9876-4321-02	2	2	<input type="button" value="Add"/>
ABCD-XYZW-FF	4	5	<input type="button" value="Add"/>
SDFG-ERTY-BN	5	7	<input type="button" value="Add"/>

Comparați cele afișate de aplicație cu informațiile existente în baza de date.

The screenshot shows the NetBeans IDE 6.9 interface. On the left, the 'Projects' pane shows a database project named 'APP' with a 'Tables' folder containing several tables, including 'PERISTENCE\_ORDER\_ORDER'. The main window displays an SQL Command window with the following query:

```
select * from APP.PERISTENCE_ORDER_ORDER
```

The results pane below the query shows the following data:

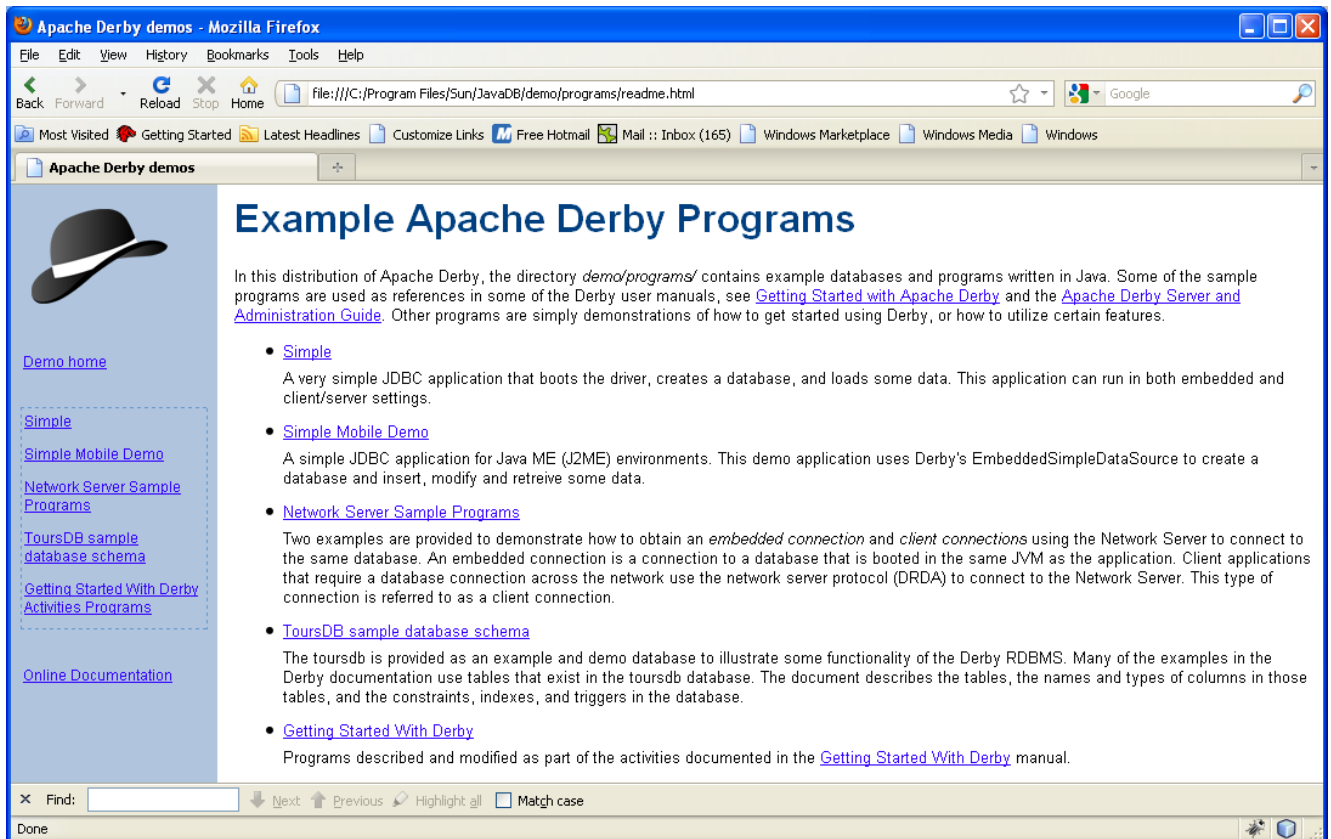
#	ORDERID	SHIPMENTINFO	STATUS	LASTUPDATE	ITEMCOUNT	DISCOUNT
1	4312	333 New Court, New City, CA 90000	N	2010-11-19 05:25:28.89	3	0
2	1111	333 New Court, New City, CA 90000	N	2010-11-19 05:25:28.89	3	10
3	1234	Test lab DAI	N	2010-11-19 05:25:46.531	3	20

Adăugați prin aplicație în tabela "Order" câmpul "CreationDate".



## 4. Exemple JDBC

Urmăriți exemplele care însoțesc Java DB.



**Example Apache Derby Programs**

In this distribution of Apache Derby, the directory `demo/programs/` contains example databases and programs written in Java. Some of the sample programs are used as references in some of the Derby user manuals, see [Getting Started with Apache Derby](#) and the [Apache Derby Server and Administration Guide](#). Other programs are simply demonstrations of how to get started using Derby, or how to utilize certain features.

- [Simple](#)  
A very simple JDBC application that boots the driver, creates a database, and loads some data. This application can run in both embedded and client/server settings.
- [Simple Mobile Demo](#)  
A simple JDBC application for Java ME (J2ME) environments. This demo application uses Derby's EmbeddedSimpleDataSource to create a database and insert, modify and retrieve some data.
- [Network Server Sample Programs](#)  
Two examples are provided to demonstrate how to obtain an *embedded connection* and *client connections* using the Network Server to connect to the same database. An embedded connection is a connection to a database that is booted in the same JVM as the application. Client applications that require a database connection across the network use the network server protocol (DRDA) to connect to the Network Server. This type of connection is referred to as a client connection.
- [ToursDB sample database schema](#)  
The toursdb is provided as an example and demo database to illustrate some functionality of the Derby RDBMS. Many of the examples in the Derby documentation use tables that exist in the toursdb database. The document describes the tables, the names and types of columns in those tables, and the constraints, indexes, and triggers in the database.
- [Getting Started With Derby](#)  
Programs described and modified as part of the activities documented in the [Getting Started With Derby](#) manual.

## 5. Resurse

Resurse utile:

- <http://download.oracle.com/javase/tutorial/jdbc/index.html>
- <http://download.oracle.com/javase/tutorial/jdbc/overview/index.html>
- <http://download.oracle.com/javase/tutorial/jdbc/basics/index.html>
- <http://www.oracle.com/technetwork/java/overview-141217.html>
- <http://www.jdbc-tutorial.com/>
- [http://books.google.ro/books?id=YmZgOkGK3oC&printsec=frontcover&dq=jdbc&source=bl&ots=G X4wPPbElg&sig=F\\_Bg83PUFRyEG3io3\\_-2FtB-Qko&hl=en&ei=n7LITOfBpDCswaNmJ21Cw&sa=X&oi=book\\_result&ct=result&resnum=4&ved=0CCcQ6AEwAzgK#v=onepage&q&f=false](http://books.google.ro/books?id=YmZgOkGK3oC&printsec=frontcover&dq=jdbc&source=bl&ots=G X4wPPbElg&sig=F_Bg83PUFRyEG3io3_-2FtB-Qko&hl=en&ei=n7LITOfBpDCswaNmJ21Cw&sa=X&oi=book_result&ct=result&resnum=4&ved=0CCcQ6AEwAzgK#v=onepage&q&f=false)
- <http://www.herongyang.com/JDBC/>
- <http://onjava.com/pub/a/onjava/2006/08/02/jjdbc-4-enhancements-in-java-se-6.html>
- <http://www.oracle.com/technetwork/java/download-141179.html>
- <file:///C:/Program%20Files/Sun/JavaDB/demo/programs/readme.html>
- <http://download.oracle.com/javaee/6/tutorial/doc/>
- <http://download.oracle.com/javaee/6/tutorial/doc/bnbpy.html>
- <http://download.oracle.com/javaee/6/jvaserverfaces/2.0/docs/pdldocs/facelets/h/tld-summary.html>
- <http://www.coreservlets.com/JSF-Tutorial/jsf1/>
- <http://www.coreservlets.com/JSF-Tutorial/jsf2/>
- <https://javaetutorial.dev.java.net/servlets/ProjectDocumentList>
- <http://download.oracle.com/javaee/6/jvaserverfaces/2.0/docs/pdldocs/facelets/>
- <http://netbeans.org/kb/docs/web/jsf20-intro.html>