

JDBC: Java Data Base Connectivity

CSE 794R/ECE 694R

1

Recall: 3-Tier Architecture

- Keep these 3 levels logically separate
(and typically *physically* separate too)

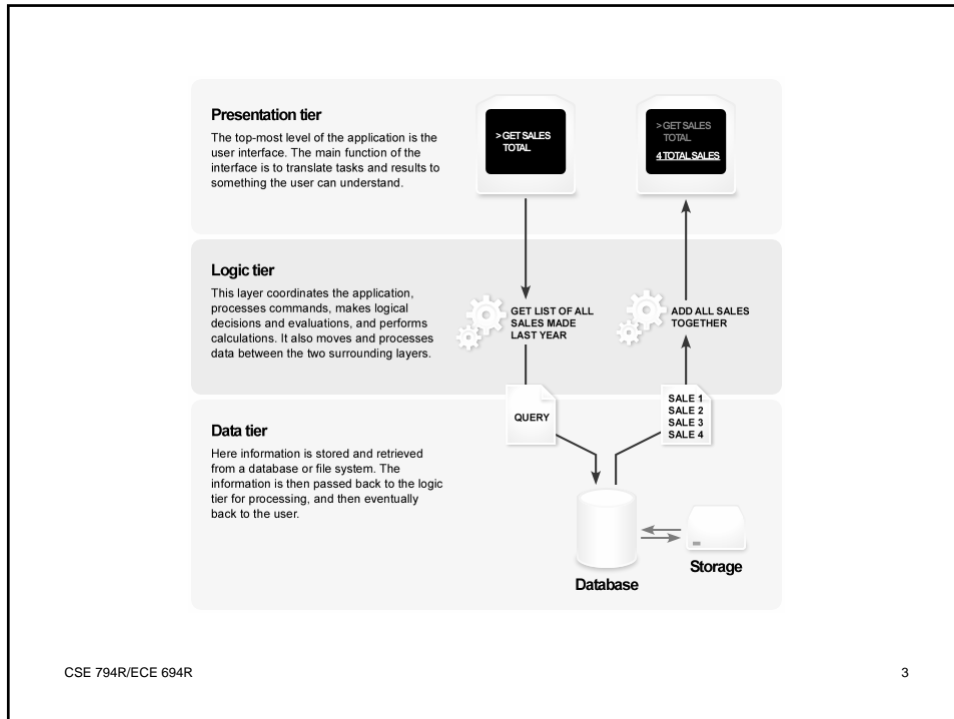
client
(http or rmi/corba)

application
(jdbc)

data

CSE 794R/ECE 694R

2



CSE 794R/ECE 694R

3

RDBMS

- Relational Database Management System
 - Organizes information as discrete tables (relations)
 - Provides mechanisms to use the database...
 - retrieve data from tables
 - insert data into tables
 - update data in tables
 - delete data from tables
 - ...and to administer the database
 - add new tables
 - change the structure of existing tables
 - remove existing tables

CSE 794R/ECE 694R

4

SQL

- Structured Query Language
 - from IBM research in 1974
 - e.g.,

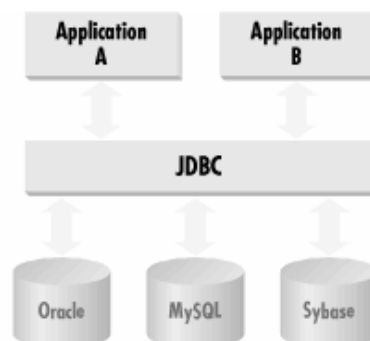
```
SELECT VendorID, VendorName
FROM VendorTable
WHERE DaysSincePurchase <= 90
      AND YTD Purchases > 10000
ORDER BY VendorID;
```
 - originally designed for interactive use

CSE 794R/ECE 694R

5

JDBC

- Java Database Connectivity
- Provides a Java API to access SQL databases



CSE 794R/ECE 694R

6

JDBC Classes

- java.sql.
 - DriverManager
 - getConnection
 - Connection
 - createStatement
 - Statement
 - execute, executeQuery, executeBatch
 - ResultSet
 - next, getString, getInt, getDate, getMetaData
 - ResultSetMetadata
 - getColumnCount, getColumnName, getColumnType

CSE 794R/ECE 694R

7

Common Steps

1. Instantiate proper driver
2. Open connection to database
3. Create a statement
4. Execute a query
5. Process the result

CSE 794R/ECE 694R

8

Instantiate Driver

- `Class.forName("driver class")`
- Driver class is vendor dependent, e.g.,
 - `sun.jdbc.odbc.JdbcOdbcDriver`
 - JDBC-ODBC bridge used to access ODBC sources
 - `com.mysql.jdbc.Driver`
 - driver to access MySQL database
 - `com.sybase.jdbc2.jdbc.SybDriver`
 - driver to access Sybase database

CSE 794R/ECE 694R

9

Open Connection

- `DriverManager.getConnection(url)` or `DriverManager.getConnection(url, user, pwd)` return `Connection`
- URL is `jdbc:<subprotocol>:<subname>`
 - e.g., `jdbc:odbc:someDB`
- `jdbc:<subprotocol>://<host>:<port>/<db>`
 - `jdbc:mysql://pc-dl672.cse.ohio-state.edu:24601/testDB`

CSE 794R/ECE 694R

10

Create Statement

- `Statement connection.createStatement()`
- Creates a `Statement` object for sending SQL statements to the database

Execute Query

- `boolean stmt.execute(sql)`
 - `sql` is a `String` containing an SQL command
 - returns `true` if first result is a `ResultSet`
- `ResultSet stmt.executeQuery(sql)`
 - `sql` must be an SQL command that returns a single `ResultSet`, usually a `SELECT`

Process Result

- **ResultSet**—a table of data representing a database result set
 - `boolean rs.next()`
 - `rs.getString(colIndex or colName)`
 - `rs.getInt()`
 - `rs.getDate()`
 - ...
 - `rs.getMetaData()`

Meta Data

- **ResultSetMetaData**—an object that can be used to get information about the types and properties of the columns in a **ResultSet** object
 - `rsmd.getColumnCount()`
 - `rsmd.getColumnLabel(col)`
 - `rsmd.getColumnType(col)`
 - ...