BCNF and JDBC

CSE 444 section, July 8, 2010

Today

- BCNF decompositions
- JDBC for project 2

What is BCNF?

A relation R is in BCNF if:

If A_1,… A_n → B is a non-trivial dependency in R, then {A_1, …, A_n} is a superkey for R

Why do BCNF decompositions?

BCNF decomposition algorithm

BCNF_Decompose(R)

find X s.t.: X ≠X^+ ≠ [all attributes]

if (not found) **then** "R is in BCNF"

let Y = X^+ - X
let Z = [all attributes] - X^+
decompose R into R1(X ∪ Y) and R2(X ∪ Z)
continue to decompose recursively R1 and R2

BCNF example: table R(A, B, C, D, E)

Consider the following FDs:

- CD → E BAD
- D → B BAD
- A → CD

Which one are the bad dependences?

CD+ = BCDE

CD is not a superkey

D+ = BD

D is not a superkey

A+ = ABCDE

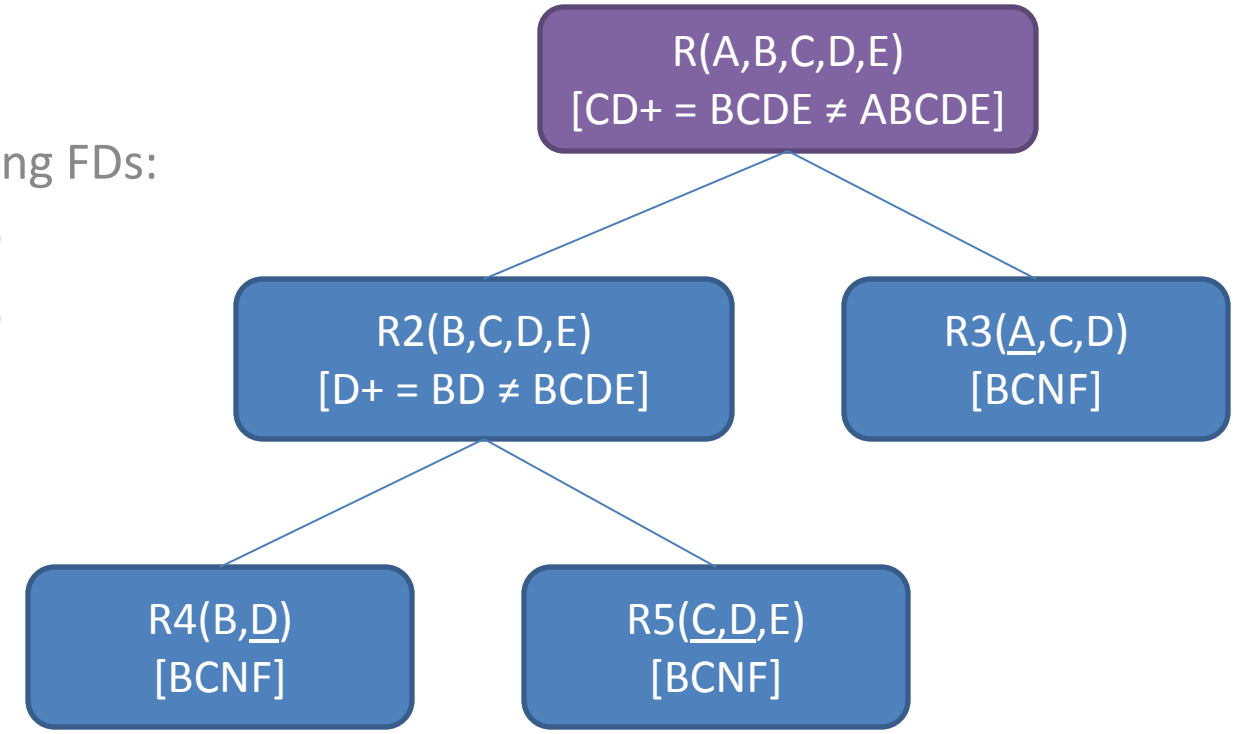
A is a superkey

Note: a set of attributes X is a superkey if X+ = ABCDE

BCNF example: table R(A, B, C, D, E)

Consider the following FDs:

- CD → E BAD
- D → B BAD
- A → CD

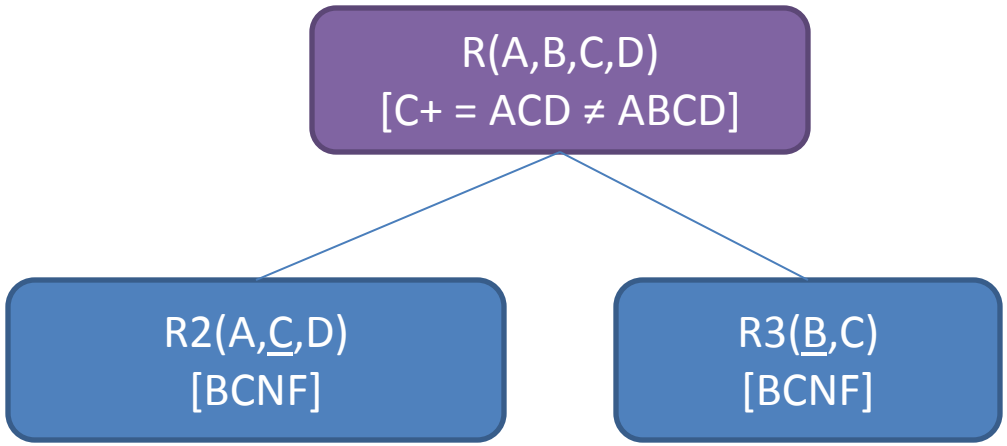


Note: a set of attributes X is a superkey if X+ = ABCDE

Another example: R(A,B,C,D)

Consider the following FDs:

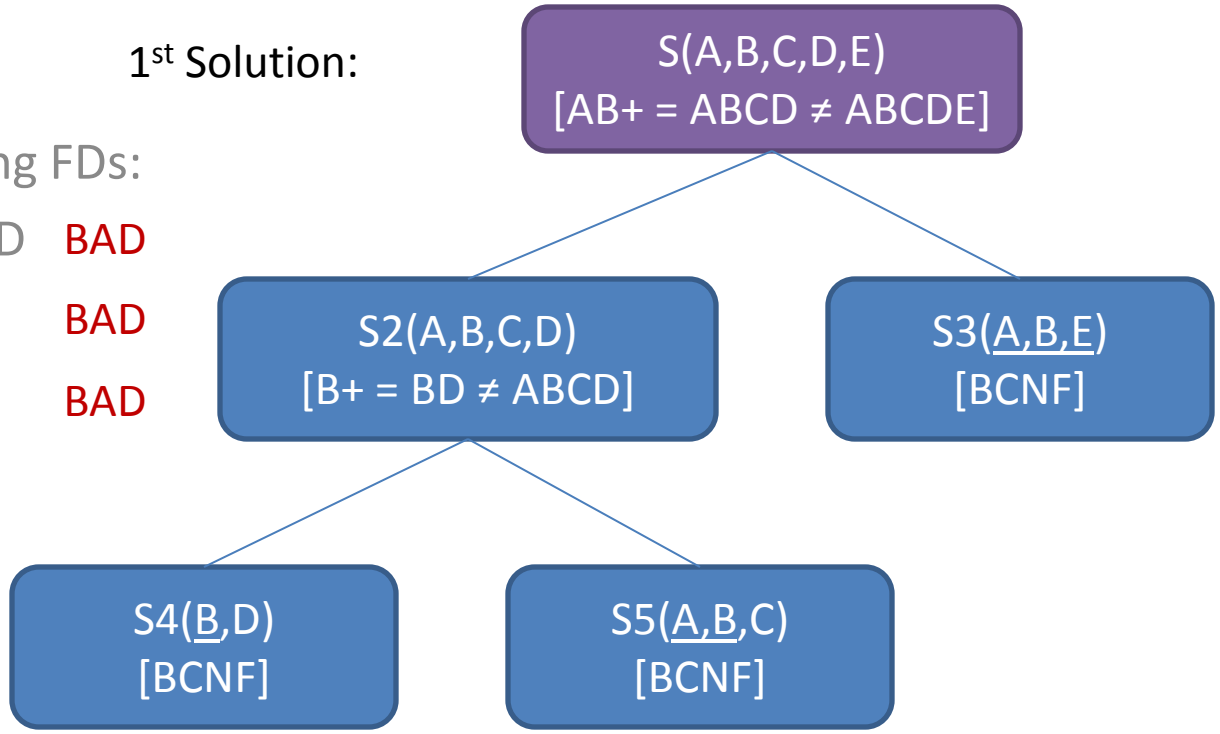
- C → D, C+ = ACD BAD
- C → A, C+ = ACD BAD
- B → C, B+ = ABCD



Note: a set of attributes X is a superkey if X+ = ABCD

A third example: S(A,B,C,D,E)

1st Solution:



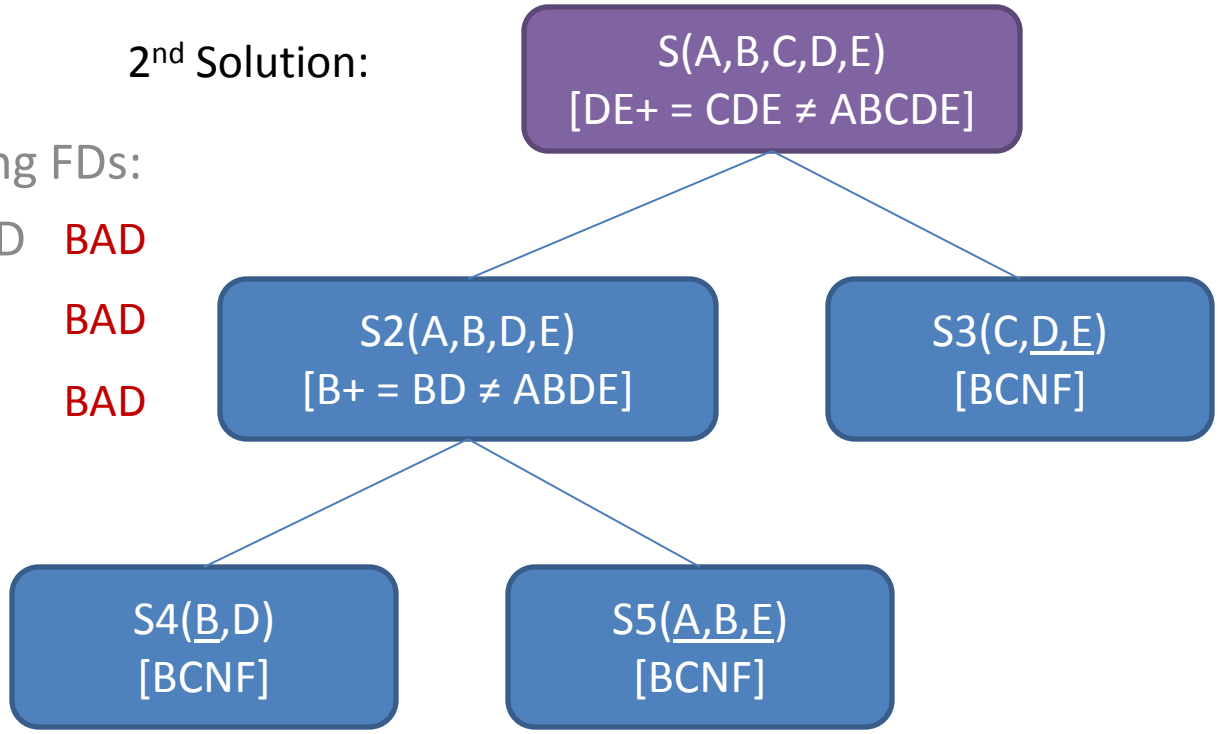
Consider the following FDs:

- AB → C, AB+ = ABCD BAD
- DE → C, DE+ = CDE BAD
- B → D, B+ = BD BAD

Note: a set of attributes X is a superkey if X+ = ABCDE

A third example: S(A,B,C,D,E)

2nd Solution:



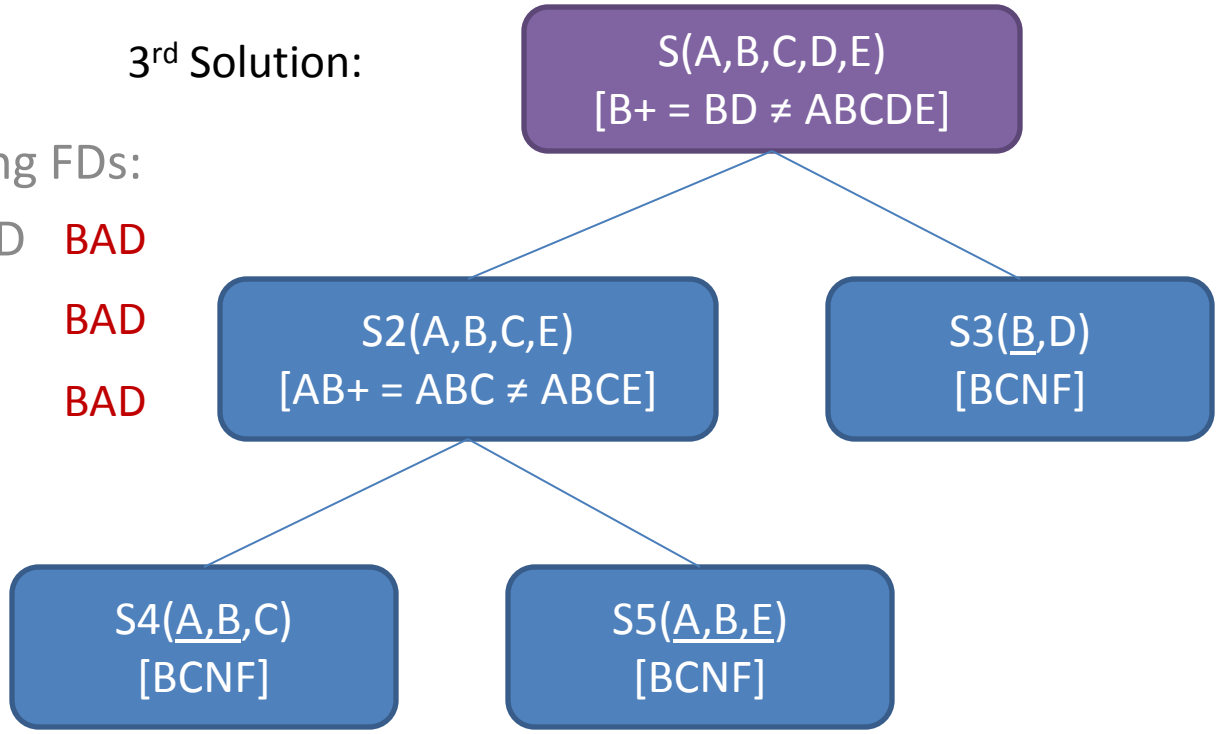
Consider the following FDs:

- AB → C, AB+ = ABCD BAD
- DE → C, DE+ = CDE BAD
- B → D, B+ = BD BAD

Note: a set of attributes X is a superkey if X+ = ABCDE

A third example: S(A,B,C,D,E)

3rd Solution:



Consider the following FDs:

- AB → C, AB+ = ABCD BAD
- DE → C, DE+ = CDE BAD
- B → D, B+ = BD BAD

Note: a set of attributes X is a superkey if X+ = ABCDE

A table with “real” data

JobInterviews

CandID	Date	Time	EmpID	RoomNo
C21	16-April-09	9.30 AM	E211	CSE550
C21	17-April-09	11.00 AM	E211	CSE550
C5	16-April-09	11.00 AM	E51	CSE218
C2	1-May-09	9.30 AM	E211	CSE218

Note: a set of attributes X is a superkey if X+ = all attributes

A table with “real” data

JobInterviews

CandID	Date	Time	EmpID	RoomNo
C21	16-April-09	9.30 AM	E211	CSE550
C21	17-April-09	11.00 AM	E211	CSE550
C5	16-April-09	11.00 AM	E51	CSE218
C2	1-May-09	9.30 AM	E211	CSE218

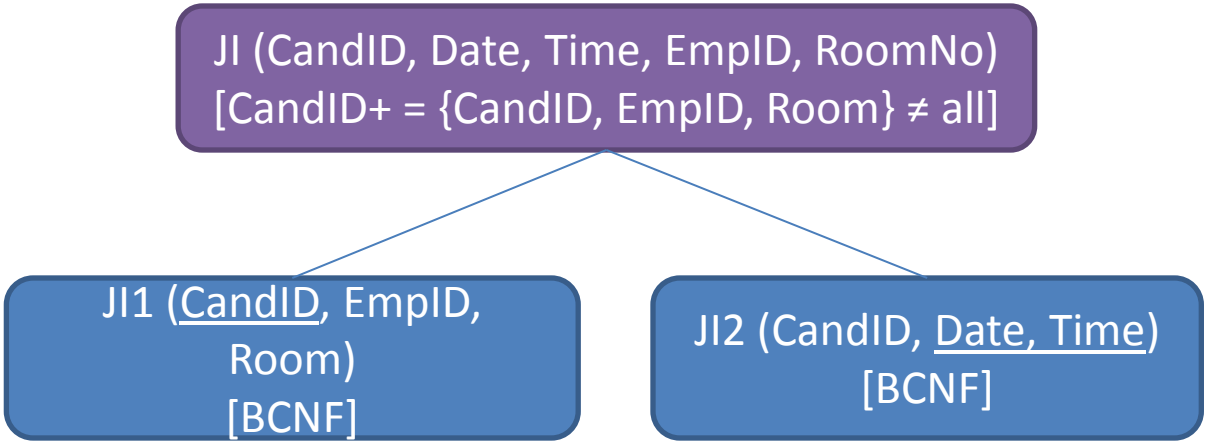
What are the FDs?

- CandID → EmpID, Room BAD
- Date, Time → CandID, EmpID, RoomNo
- Date, EmpID → CandID, Time, RoomNo
- more?

Note: a set of attributes X is a superkey if X+ = all attributes

A table with “real” data

JobInterviews (CandID, Date, Time, EmpID, RoomNo)



What are the FDs?

- CandID → EmpID, Room BAD
- Date, Time → CandID, EmpID, RoomNo
- Date, EmpID → CandID, Time, RoomNo
- more?

Note: a set of attributes X is a superkey if X+ = all attributes

Today

- BCNF decompositions
- JDBC for project 2

JDBC (Java Database Connectivity)

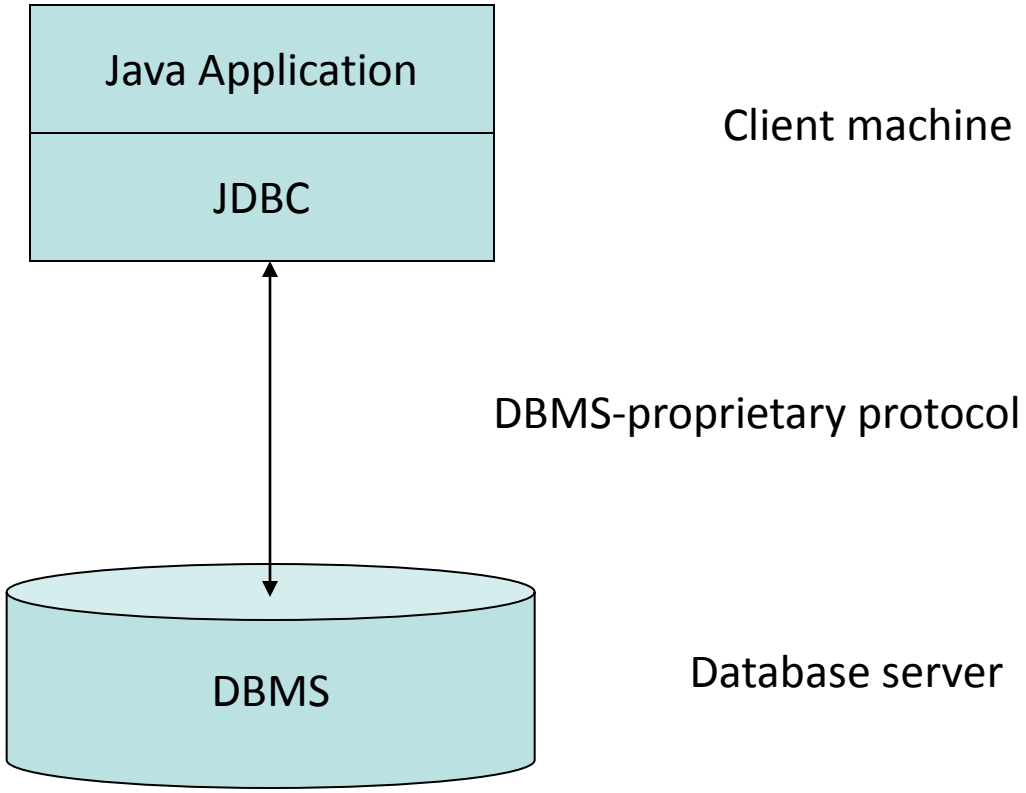
A Java API to access a database:

- connect to a data source
- send queries and update statements
- retrieve and process results

Documentation:

http://java.sun.com/javase/6/docs/technotes/guides/jdbc/

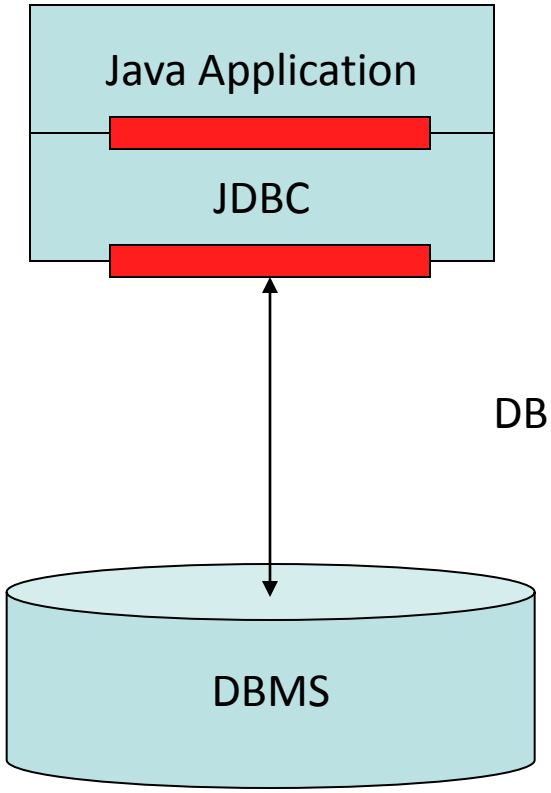
JDBC lets Java talk to your database



DBMS vendors make JDBC drivers…

JDBC API for apps

JDBC API for drivers

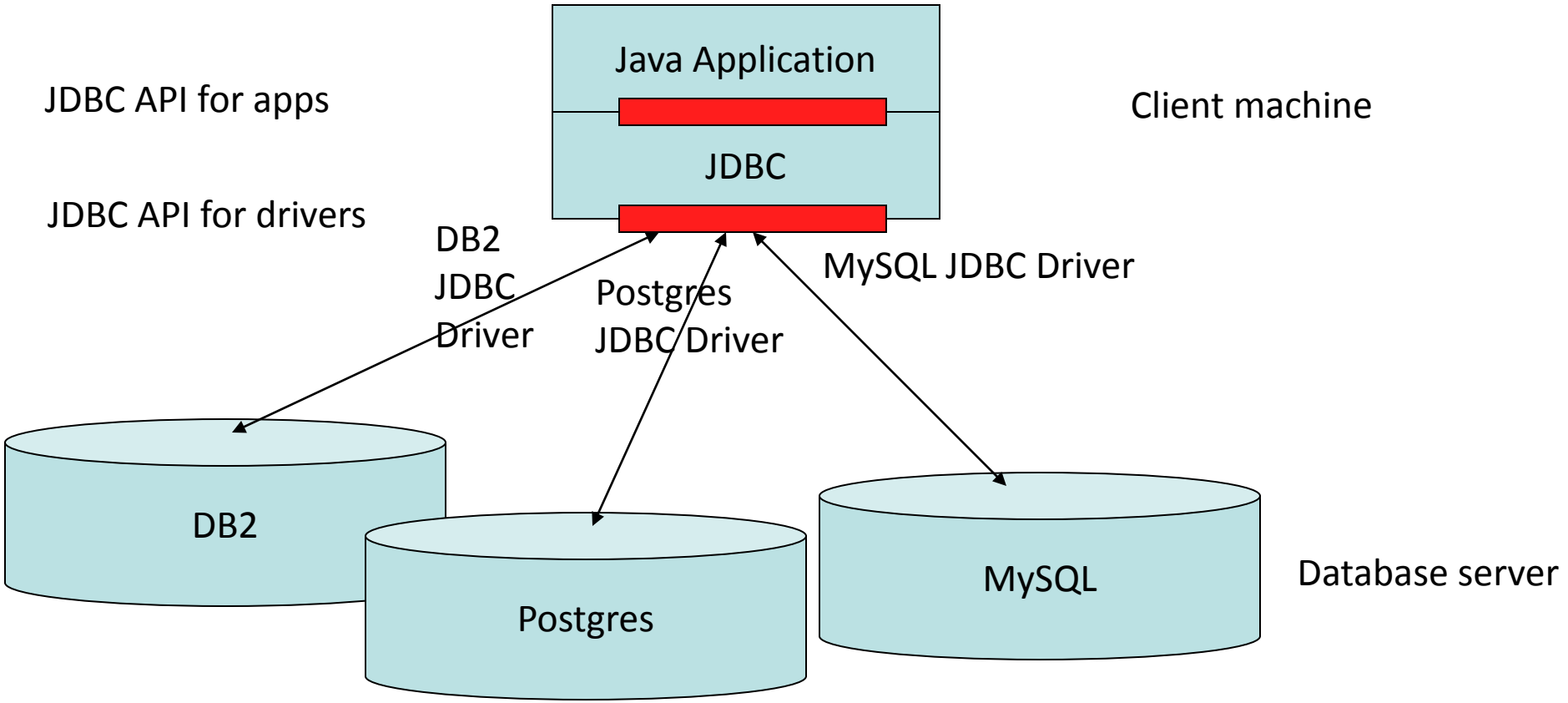


Client machine

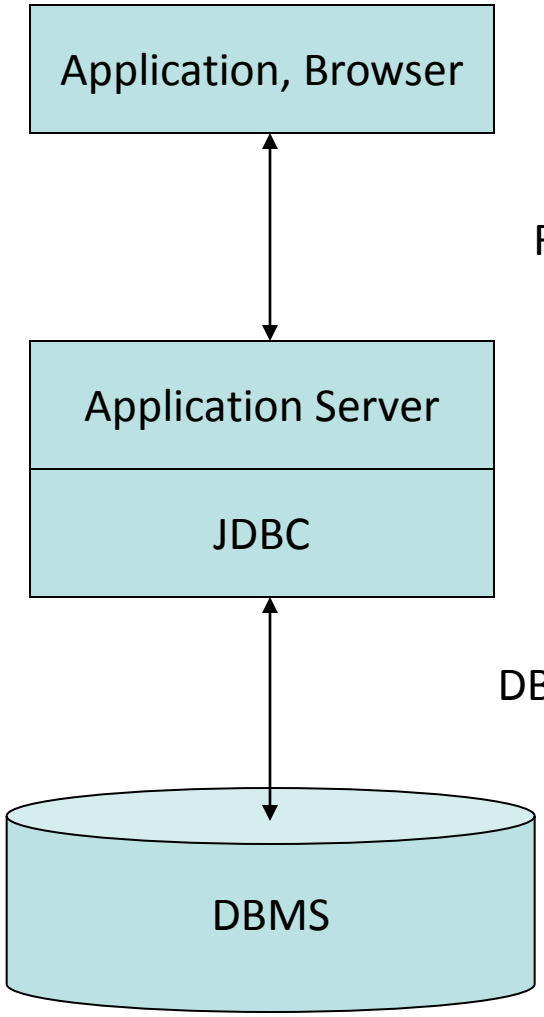
DBMS-proprietary protocol

Database server

… letting JDBC talk to *any* database



JDBC architecture
Three-tier model



connection pooling
distributed transactions
…

First, load the driver

- For Project 2, look in project2.tar.gz
  - SQL Server driver
    sqljdbc4.jar
  - PostgreSQL driver
    postgresql-8.4-701.jdbc4.jar
  - Already installed on Lab PCs (use 444shell.cmd)
- Put on class path, then tell Java to load it

```
Class.forName
("com.microsoft.sqlserver.jdbc.SQLServerDriver");
Class.forName ("org.postgresql.Driver");
```

  - Class.forName() optional in current versions of Java

JDBC example

```
Connection con = DriverManager.getConnection
   ("jdbc:sqlserver://iisqlsrv;database=imdb",
    "username", "password");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery
   ("SELECT a, b, c FROM Table1");

while (rs.next()) {
   int x = rs.getInt("a");
   String s = rs.getString("b");
   float f = rs.getFloat("c");
}
```

Modifying the database

Use Statement.executeUpdate():

```
Statement stmt = con.createStatement();

int rowsUpdated = stmt.executeUpdate (
    "UPDATE Actor " +
    "SET gender = 'F' " +
    "WHERE gender IS NULL"
);
```

- Works with any database modification, not just UPDATE
- Warning – will throw if you run it with a query!

Close all JDBC objects when done

```
Connection con = DriverManager.getConnection(...);
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery
   ("SELECT a, b, c FROM Table1");

// do work with rs...

rs.close();
stmt.close();
con.close();
```

```
Class.forName
  ("com.microsoft.sqlserver.jdbc.SQLServerDriver");

Connection con = null;

try {
    con = DriverManager.getConnection( … );


    …


} catch (Exception e) {
    e.printStackTrace();
} finally {
    con.close();
}
```

Parameterized queries - PreparedStatement

```
PreparedStatement pstmt = con.prepareStatement
    ("SELECT lname FROM persons WHERE id = ? ");
…
pstmt.setInt(1, 34);
ResultSet rs1 = pstmt.executeQuery();
…

pstmt.setInt(1, 63);
ResultSet rs2 = pstmt.executeQuery();

…
```

Parameterized queries - PreparedStatement

No need to worry about quotes ‘, “

```
PreparedStatement pstmt = con.prepareStatement
    ("SELECT website FROM shops
     WHERE name = ? OR owner = ? ");
…
pstmt.setString(1, "George's");
pstmt.setString(2, "Oh \"wow\"!");
…
```

Parameterized queries - PreparedStatement

No need to worry about quotes ‘, “

```
PreparedStatement pstmt = con.prepareStatement
    ("SELECT website FROM shops
     WHERE name = ? OR owner = ? ");
…
pstmt.setString(1, "George's");
pstmt.setString(2, "Oh \"wow\"!");
…
```

Single quotes without escaping!

Parameterizing lets plan be cached

```
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery
   ("SELECT website FROM shops
      WHERE name = 'George\'s' OR …");
```

Must escape single quotes.
What if this came from user?

Transactions

- A transaction is a logical group of SQL statements
- Each transaction is guaranteed to execute *as if* it was the only code running on the database
- We will talk more about them in lecture

Transactions in JDBC – option 1

Execute the SQL code to start, end transactions:

```
PreparedStatement pBeginTx   =
   con.prepareStatement("BEGIN TRANSACTION");
PreparedStatement pCommitTx   =
   con.prepareStatement("COMMIT TRANSACTION");
PreparedStatement pRollbackTx =
   con.prepareStatement("ROLLBACK TRANSACTION");
…
pBeginTx.executeUpdate();
// transaction started
…

if (ok) pCommitTx.executeUpdate();
else pRollbackTx.executeUpdate();
// transaction finished or reverted
```

Transactions in JDBC – option 2

Use JDBC methods to work with transactions:

```
con.setAutoCommit(false);
// From now on, everything is in a transaction
…

if (ok) con.commit();
else con.rollback();
// Old transaction done/reverted, new one started
…

con.setAutoCommit(true);
// Now each statement executes by itself again
```