Using Relational Databases from a Program

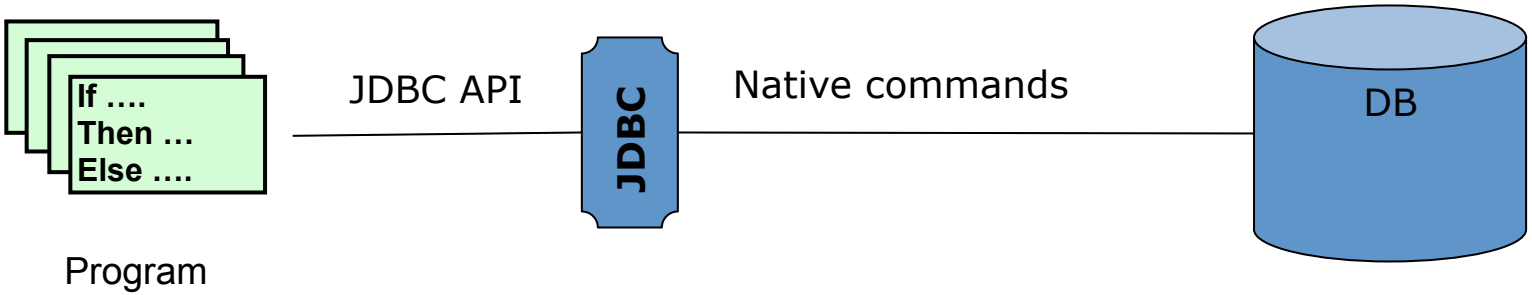
JDBC

Yannis Velegrakis
University of Trento

The Database

- If you want to install MySQL on your PC, it can be downloaded from
  - http://dev.mysql.com/downloads
- You should also download the MySQL Admin Tool
  - http://dev.mysql.com/downloads/gui-tools/5.0.html
  - Use localhost as hostname

JDBC Idea



- You will need the JDBC library to make your java program run. This can be found on the course web site on the lecture of the lab

Connect to the DB

- Open a DB connection
  - Remember to close the connection at the end of the program
    - dbConnection.close()

```
public static Connection connect(String host, int port, String dbName, String user, String passwd)
{
    Connection dbConnection = null;
    try
    {
        String dbString = null;
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        dbString = "jdbc:mysql://" + host + "/" + dbName;
        dbConnection = DriverManager.getConnection(dbString, user, passwd);
    }
    catch (Exception e)
    {
        System.err.println("Failed to connect with the DB");
        e.printStackTrace();
    }
    return dbConnection;
}
```

Connect to the DB

- Too many connections will crash the server
- Plus they are VERY SLOW
- Solution?
  - 1 connection for all the program
  - Connection pool

```
Connection con = connect("ares.science.unitn.it", 3306, "db_ppiw01", "ppiw01", "ppiwPass01");
String command;

command = "create table Students(sid varchar(25), name varchar(50), dateOfBirth varchar(25))";
executeStatement(command, con);

command = "insert into Students values (\"s1\", \"John\", \"01/01/1999\")";
executeStatement(command, con);
command = "insert into Students values (\"s2\", \"Mary\", \"02/02/2000\")";
executeStatement(command, con);
```

Send a command

- executeStatement("delete from Students where dateOfBirth=\"16/01/1978\"", con);

```
private static void executeStatement(String q, Connection con)
{
    try
    {
        Statement stmt = con.createStatement();
        stmt.execute(q);
        stmt.close();
    }
    catch (SQLException e)
    {
        System.out.println("Statement: \n" + q);
        e.printStackTrace();
        throw new RuntimeException("Query " + q);
    }
}
```

Sending a query

- executeQuery("select * from Students", con)

```
public static ResultSet executeQuery(String query, Connection con)
{
    try
    {
        Statement stmt = con.createStatement();
        ResultSet results = stmt.executeQuery(query);
        return results;
    }
    catch (Exception e)
    {
        e.printStackTrace();
        return null;
    }
}
```

- Why not a close command?

Reading the Results

- Result set is a list of tuples
- Each tuple has a series of attributes

Very Important

Why ?

Also this

```
command = "select * from Students";
ResultSet results = executeQuery(command, con);

try
{
    while (results.next())
    {
        String cid = results.getString(1);
        String stName = results.getString(2);
        //String age = results.getInt(1);
        System.out.println("cid=" + cid + " stName=" + stName);
    }
    Statement stmt = results.getStatement();
    results.close();
    stmt.close();
}
catch (Exception e)
{
    e.printStackTrace();
    throw new RuntimeException();
}


command = "drop table Students";
executeStatement(command, con);

try
{
    con.close();
}
catch (SQLException e)
{
    e.printStackTrace();
}
```

What about images?

- What options do we have ?
  - What are their advantages and disadvantages?
- BLOB Data type

Thank you for your attention!

Questions?