Chapter 10
Java and SQL

Wang Yang
wyang@njnet.edu.cn

Outline

- Concern Data - File & IO vs. Database &SQL
- Database & SQL
- How Connect Java to SQL - Java Model for Database
- Java Database Connectivity (JDBC) - API

Concern Data

File & IO vs. Database & SQL

Review of File & IO

- Experiment : Write Student Information into File
  - 姓名　　学号　　成绩
  - 张小灵 711011101 89
  - 李小乐 711011102 90
- We can use two methods
  - OutputStream
  - Writer

Review of File & IO

- OutputStream

```
DataOutputStream f = new DataOutputStream(
        new FileOutputStream("1.data"));
f.write("张小灵".getBytes());
f.writeInt(71101101);
f.writeInt(89);
f.write("李小乐".getBytes());
f.writeInt(71101102);
f.writeInt(90);
```

```
0000000 e5 bc a0 e5 b0 8f e7 81 b5 04 3c ea ad 00 00 00
0000010 59 e6 9d 8e e5 b0 8f e4 b9 90 04 3c ea ae 00 00
0000020 00 5a
```

Review of File & IO

- But if we want to change content
  - read all the record
  - change the specific record
  - write all the record back
  - rewrite All the record just for one record change

Review of File & IO

- We can use RandomAccessFile
  - use seek() to specific position
  - override the record

```
RandomAccessFile r = new RandomAccessFile("1.data", "w");
r.seek("张小灵".getBytes().length+4);
r.writeInt(91);
```

change 张小灵 score to 91

Review of File & IO

- But if we want to change record length

change 张小灵 name to 张灵

```
RandomAccessFile r = new RandomAccessFile("1.data", "rw");
r.write("张灵".getBytes());
r.close();
```

what we get is 张灵?

```
0000000 e5 bc a0 e7 81 b5 e7 81 b5
0000009 04 3c ea ad 00 00 00 59
```

Review of File & IO

- If multiple people want to modify the file at the same time
    - we must use synchronization the process
- If Someone insert wrong data, like -1 for score
    - we must write code to insure it will not be written
- if we want to use another language
    - we must learn different data type and process

What we concern is Data

- We only want write Student Information into a table
  - like excel, can easily manipulate the data
  - we only concern data,
    - not the media, :
      - binaryFile, TextFile, CSVFile, Excel,…
    - not the process
      - OutputStream, Writer, RandomAccessFile,…
    - not the programming language
      - Java, C, C++, …

	A	B	C
1	姓名	学号	成绩
2	张小灵	7110101	89
3	李小乐	7110102	91
4			
5			
6			
7			
8			

What we concern is Data

- We want to
  - create a table struct to describe data
    - 姓名： String：2～10 Character
    - 学号： String： 7Character， All Numeric
    - 成绩： Int　　： 0～100

What we concern is Data

- we want to manipulate data
  - insert data record
  - update data record
  - delete data record
  - query data record

What we concern is Data

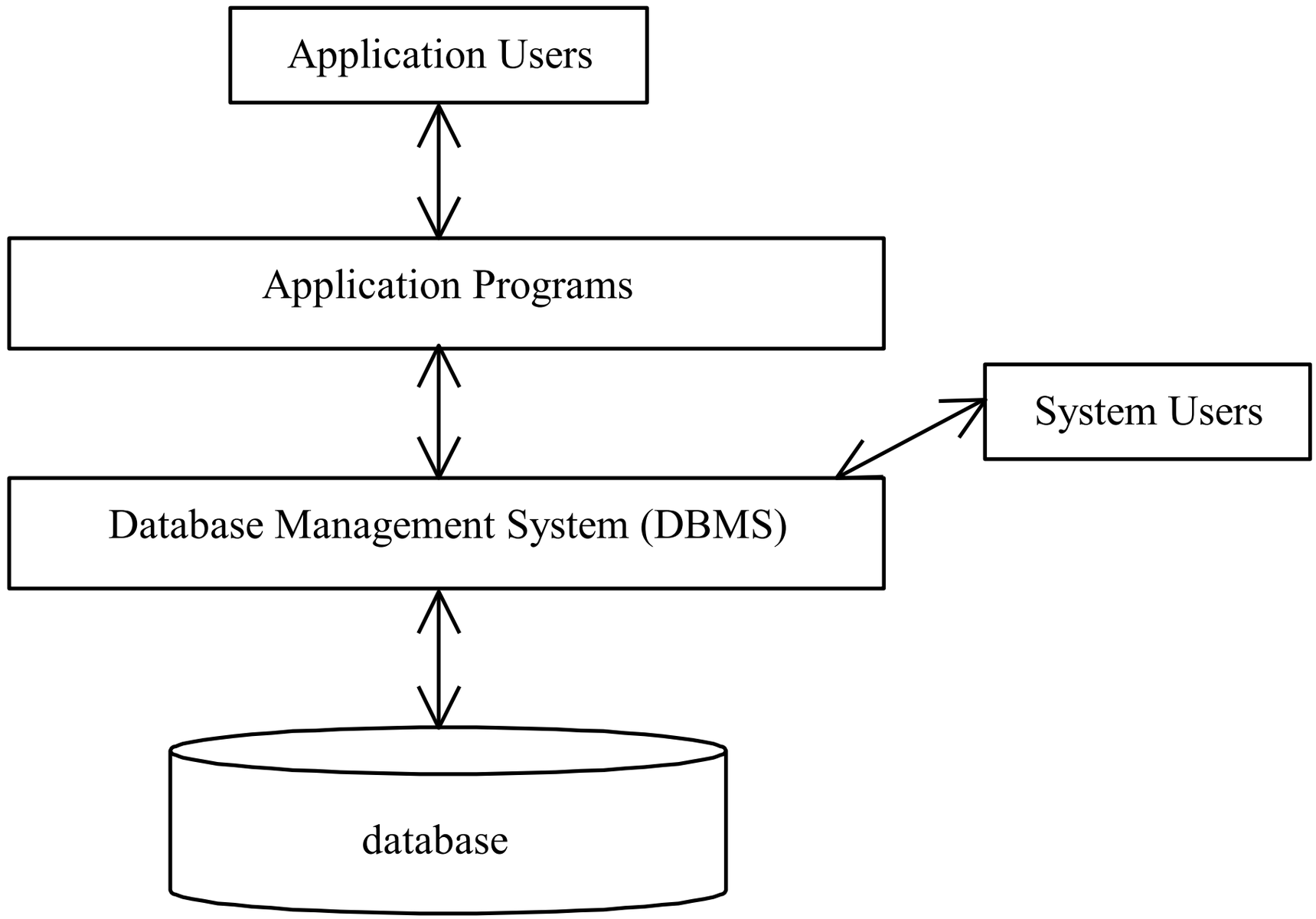
- We want there are a common system to store data — Database
- we want to learn only a common language to process it - SQL

Database & SQL

Basic Concepts in Database

- Database and DBMS
- RDBMS (关系型数据库管理系统)
    - Store data in rows and columns
    - Rows called Record (记录) , and columns called Field (字段)
    - A set of rows and columns are called Table (表)
    - A table usually represents an Entity (实体)
    - There are Relations (关系) between Entities, ER Map
    - Query through SQL（结构化查询语言）

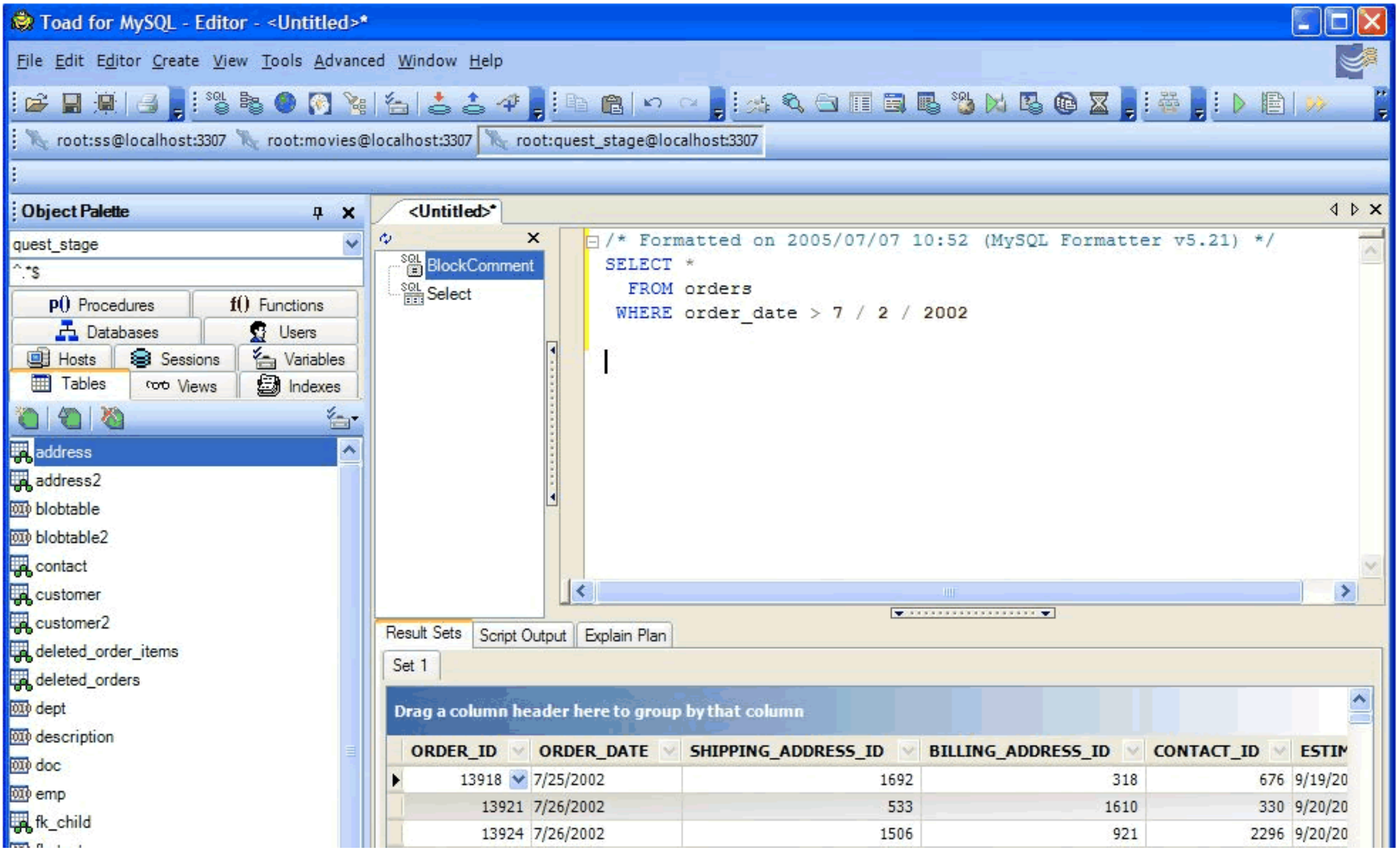
Basic Concepts in Database



Basic Concepts in Database

- Usual RDBMS
  - MySQL / PostgreSQL / Berkeley DB
  - Oracle
  - SQL Server
  - DB2

Basic Concepts in Database



Basic Concepts in SQL

- Structured Query Language
- SQL including
    - DDL – Data Definition Language
        - These queries are used to create database objects such as tables, indexes, procedures, constraints
        - Create , drop, alter,truncate,comment,rename
- DML – Data Manipulation Language
    - These queries are used to manipulate the data in the database tables.
    - Insert, update, delete, select (more available)
- DCL – Data Control Language
    - These are data control queries like grant and revoke permissions

CRUD Operations in SQL

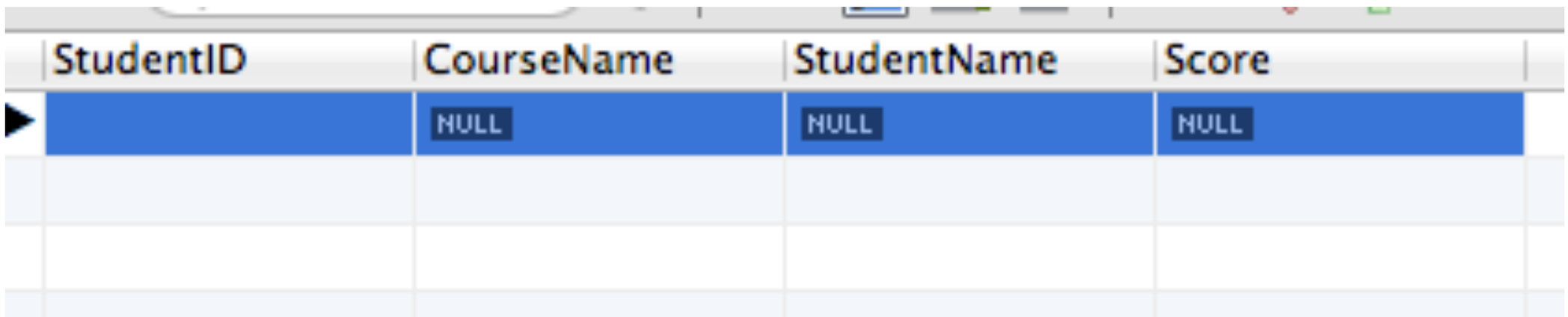
- CRUD means
  - Create, Read, Update, Delete
    - Create : create the table and insert data into table
    - Read : select data from table
    - Update : update data in the table
    - Delete : delete data from the table

Create Statement

- Data
- Query

```
create table Course (
  StudentName varchar(10),
  StudentID char(7),
  CourseName varchar(10),
  score integer,
  primary key (studentID));
```

- result



Insert Statement

- Data

ColumnName	DataType
StudentName	varchar(10)
StudentID	char(7)
Score	int

- Query

```
INSERT into Course (StudentName, StudentID, Score) VALUES (”张小灵”,”7110101”, 91) ;
```

- result

StudentID	StudentName	Score
7110101	张小灵	91

Insert Statement

- First assign which fields you want to insert
  - you can just insert one field or two fields
    - other fields will be null or default value
- Then you give values for the fields
  - String must be quoted
  - don’t use Chinese “ or Chinese ()

Select Statement

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
SELECT StudentName, Score FROM Course;
```

- Result

StudentName	Score
张小灵	91
李小乐	89
王小天	85
赵小宝	91

Select Where Condition

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
SELECT StudentName, Score FROM Course
Where Score > 90;
```

- Result

StudentName	Score
张小灵	91
赵小宝	91

Select Where Combined Condition

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
SELECT StudentName, Score FROM Course
Where Score > 90 and StudentID < “7110103” ;
```

- Result

StudentName	Score
张小灵	91

Select Where In

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	

- Query

```
SELECT StudentName, Score FROM Course
Where StudentName in (“张小灵”, “李小乐”);
```

- Result

StudentName	Score
张小灵	91
李小乐	89

Select Where Between

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
SELECT StudentName, Score FROM Course Where
SutdentID between ‘7110101’ and ‘7110103’;
```

- Result

StudentName	Score
张小灵	91
李小乐	89
王小天	85

Select Where Like

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
SELECT StudentName, Score FROM Course
Where StudentName Like “张%”;
```

- Result

StudentName	Score
张小灵	91

Select Where Order by

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
SELECT StudentName, Score FROM Course
Order by Score DESC;
```

- Result

StudentName	Score
张小灵	91
赵小宝	91
李小乐	89
王小天	85

Select Distinct

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
SELECT Distinct(Score) FROM Course;
```

- Result

Score
91
89
85

Select Where Count

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
SELECT Count (Distinct(Score)) FROM Course;
```

- Result

Count(Distinct(Score))
3

Update Statement

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
UPDATE Course SET Score=95 Where StudentID=“7110102”;
```

- Result

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	95
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

Delete Statement

- Data

StudentID	StudentName	Score
7110101	张小灵	91
7110102	李小乐	89
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

- Query

```
DELETE FROM Course Where Score>91;
```

- Result

StudentID	StudentName	Score
7110101	张小灵	91
7110103	王小天	85
7110104	赵小宝	91
NULL	NULL	NULL

TEST

	CourseName	StudentCnt	Semester
	面向对象程序设计1	30	第一学期
	高级面向对象程序设计	60	第二学期
▶	数据结构	50	第一学期
		NULL	NULL

- Write following SQL query

  for Table “CourseInfo”

  - Query for all courseName
  - Query for the number of course with studentCnt > 50
  - Query for CourseName, StudentCnt with course name containing “对象”, and rank the result descending according to StudentCnt
  - Insert into table a new record (any record will do)
  - Modify “面向对象程序设计1”课程的 studentCnt to 100
  - Delete all records related to “面向对象”

SELECT CourseName FROM CourseInfo;

SELECT count(*) FROM CourseInfo where StudentCnt > 50;

SELECT CourseName, StudentCnt FROM CourseInfo where CourseName like "%对象%" order by StudentCnt DESC

INSERT INTO CourseInfo (CourseName, StudentCnt, Semester) values ("软件测试", 40, “第一学期")

UPDATE CourseInfo Set StudentCnt=100 where CourseName=“面向对象程序设计1”

DELETE from CourseInfo

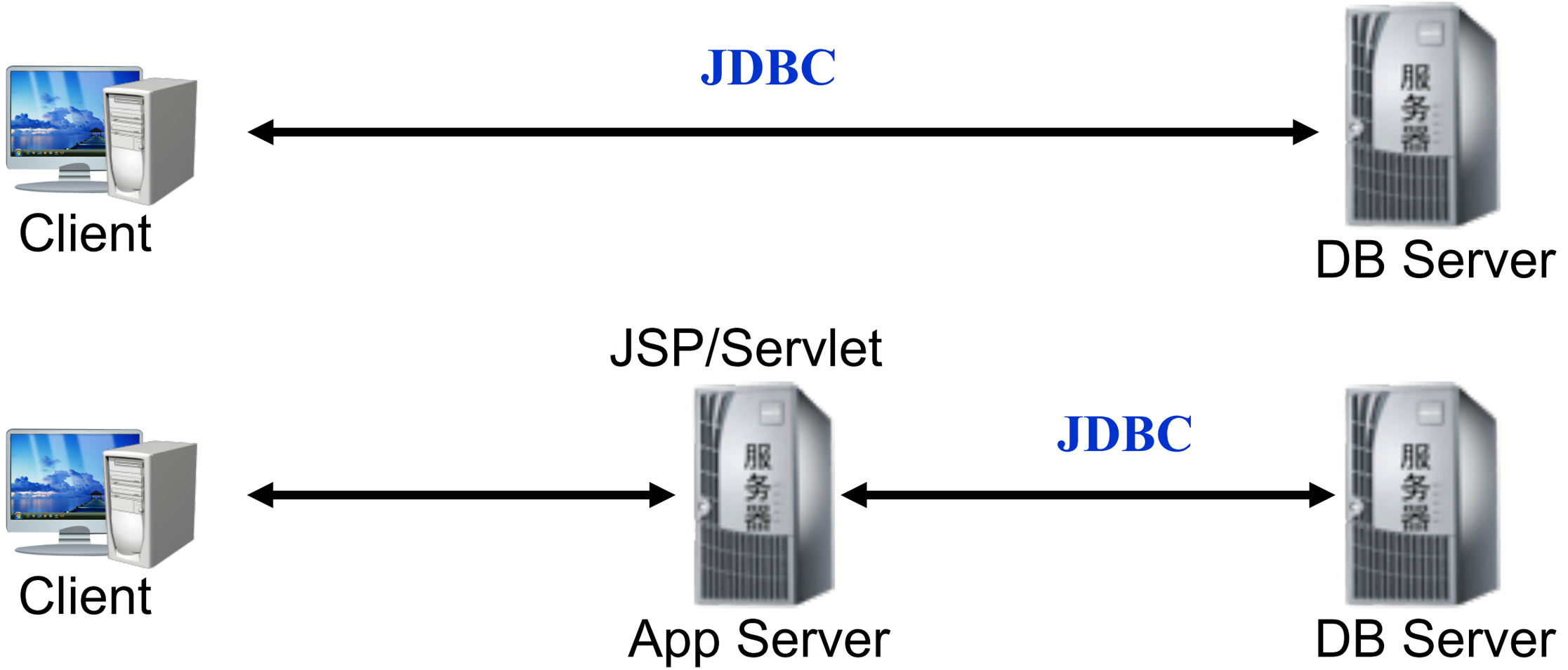
TEST

CourseName	StudentCnt	Semester
面向对象程序设计1	30	第一学期
高级面向对象程序设计	60	第二学期
数据结构	50	第一学期
	NULL	NULL

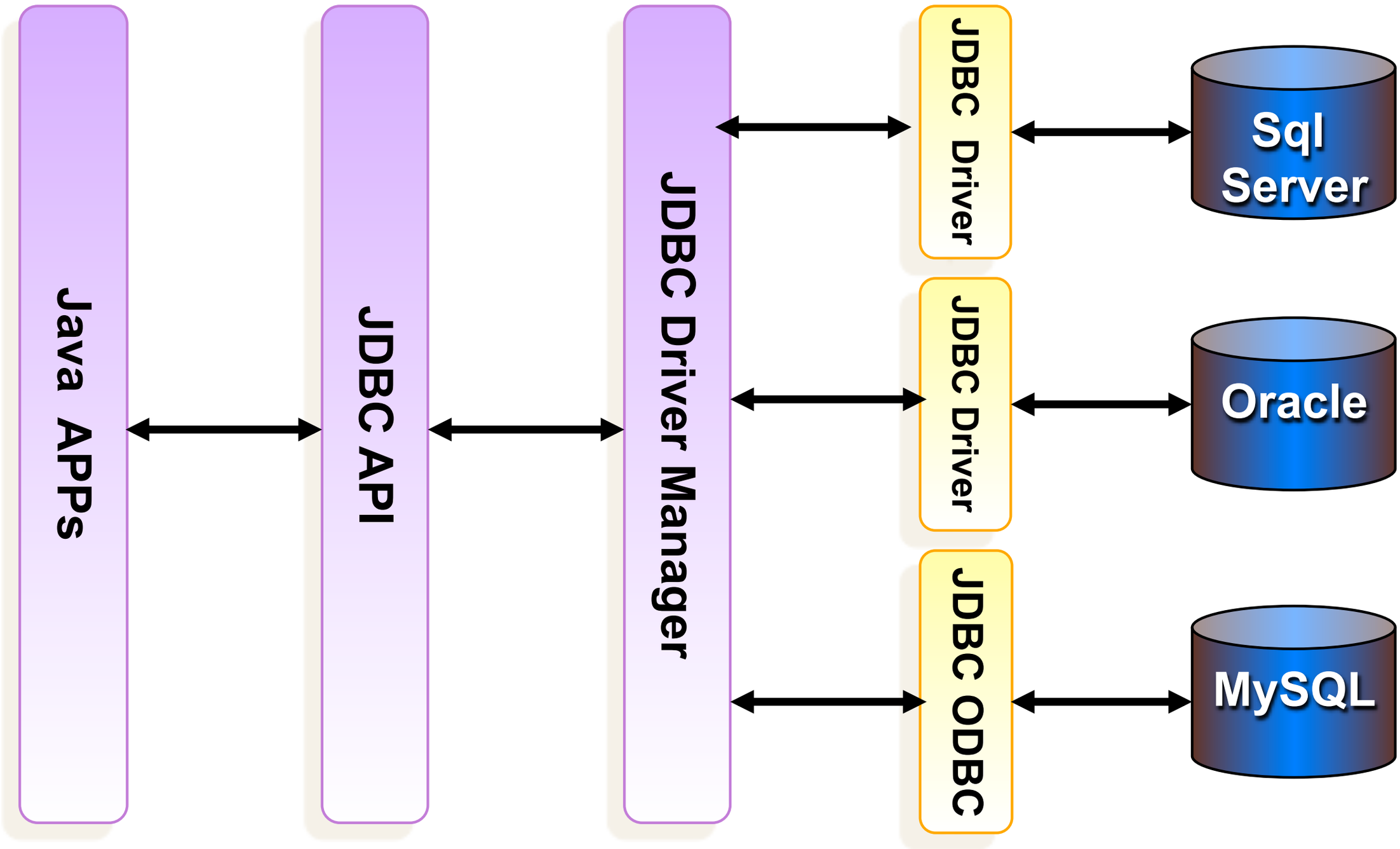
How Connect Java to SQL Java Model for Database

JDBC

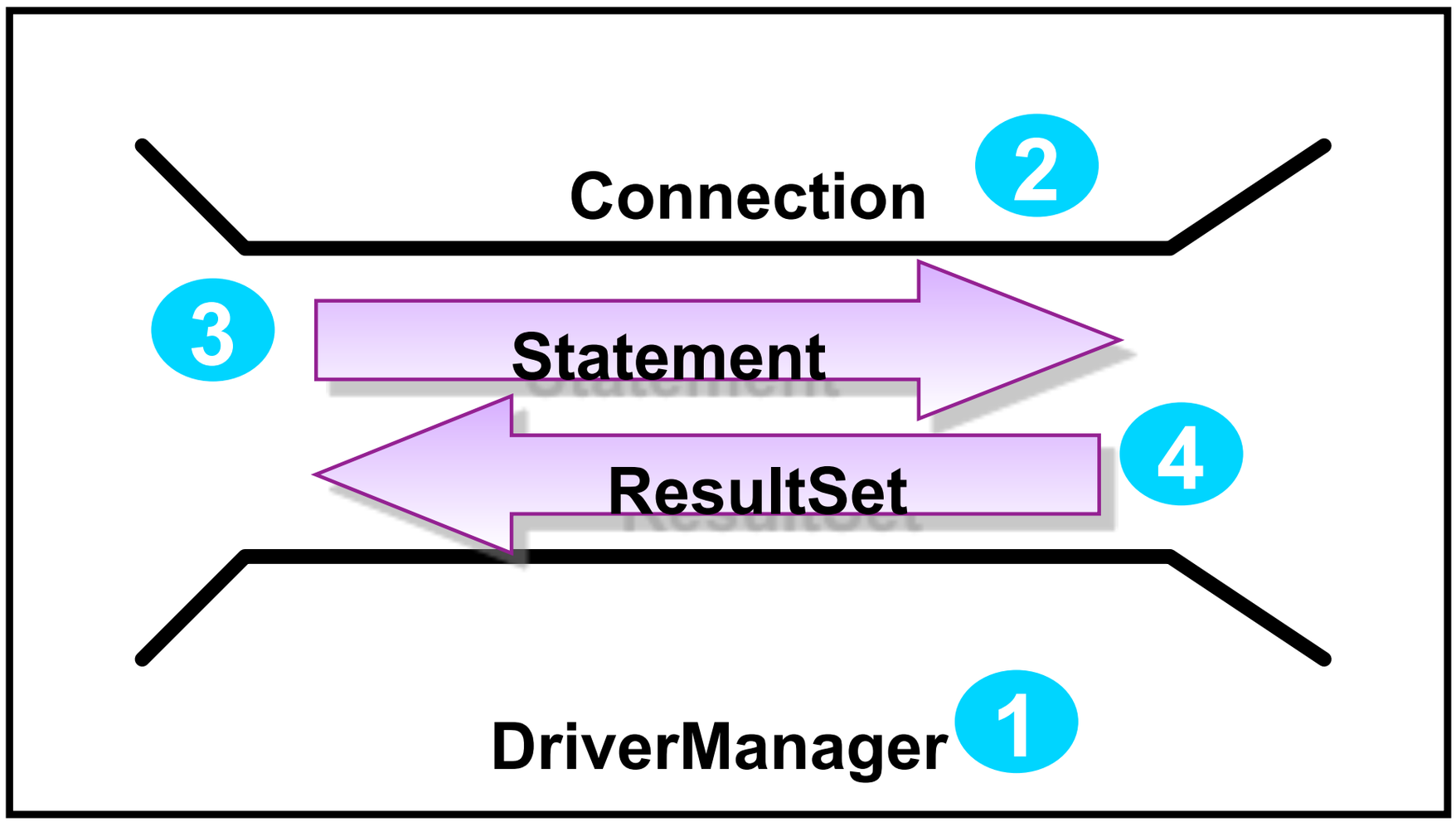
- JDBC - Java Database Connectivity
- Provides API for database access



JDBC Principle



JDBC Principle



Client

DB Server

Java Database Connectivity API

JDBC API

- Provider: Sun
- Package
  - java.sql
  - javax.sql
- Major Classes and Interfaces
  - DriverManager Class
  - Connection Interface
  - Statement Interface
  - ResultSet Interface

Establish a Connection

- **import java.sql.*;**
- Load the vendor specific driver
  - **Class.forName("com.mysql.jdbc.Driver");**
    - Dynamically loads a driver class, for Mysql database
- Make the connection
  - **String dbURL = "jdbc:mysql://localhost:3306/" +**
  - **"MyDB?user=your_username&password=your_password";**
  - **Connection connection = DriverManager.getConnection(dbURL);**
    - Establishes connection to database by obtaining a Connection object

Create JDBC statement(s)

- **String query = “Select * From Course”;**
- **Statement stmt = conn.createStatement() ;**
  - Creates a Statement object for sending SQL statements to the database

execute and get Result

```
ResultSet rs = stmt.executeQuery(query);

while (rs.next()) {
    System.out.println(rs.getString("StudentName"));
}
```

张小灵
李小乐
王小天
赵小宝

Close all the resource

- **rs.close();**
- **stmt.close();**
- **conn.close();**
- release all the resources of ResultSet, Statement, and Connection

JDBC Process

```
try{
    // 加载及注册JDBC驱动程序
    Class.forName(...);
    //创建JDBC连接
    Connection connection = ......
    //创建Statement
    Statement statement  = ......
    //创建查询并处理查询结果
    ResultSet rs = .....
}
catch (ClassNotFoundException e) {System.out.println("无法找到驱动类");}
catch (SQLException e) {e.printStackTrace();}
finally {
    try {
        rs.close();
        statement.close();
        connection.close();
    } catch (Exception e) {    e.printStackTrace();}
}
```

Two Way of Executing Statement

- executeUpdate
  - For queries with no results returned, usually Insert \ Delete \ Update
- executeQuery
  - For queries with results returned, usually Select

```
Statement stmt = conn.createStatement();
stmt.executeUpdate("DELETE FROM Course where Score>90 ");

Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM Course");
```

PreparedStatement Interface

- Derived from Statement interface
- For repeatedly executed SQL
- Usually for queries with no result, such as Insert \ Delete \ Update
- Together with addBatch() and executeBatch()

PreparedStatement Interface

```
String query = "INSERT INTO account (username, money, interest)
VALUES (?,?,?)";
PreparedStatement pst = connection.prepareStatement(query);
pst.setString(1, “张三”);
pst.setInt(2, 100);
pst.setInt(3, 10);
pst.addBatch();
pst.clearParameters();

pst.setString(1, “李四”);
pst.setInt(2, 200);
pst.setInt(3, 20);
pst.addBatch();
pst.clearParameters();

pst.executeBatch();
pst.close();
```

Reference

- W3C School SQL教程
  - http://www.w3school.com.cn/sql/
- SQL语句教程
  - http://www.1keydata.com/cn/sql/sql.php
- JDBC Data Access API – JDBC Technology Homepage
  - http://www.oracle.com/technetwork/java/javase/jdbc/index.html
- JDBC Database Access – The Java Tutorial
  - http://docs.oracle.com/javase/tutorial/jdbc/index.html
- JDBC Documentation
  - http://docs.oracle.com/javase/7/docs/technotes/guides/jdbc/