

学习目标

- * 能够掌握数据定义语言DDL
 - * create table employee , alter table employee modify,add,drop,rename.. to
 - * drop table employee, truncate table employee,
- * 能够掌握数据操纵语言DML
 - * insert,update,delete
- * 能够掌握数据查询语言DQL
 - * select * from emp where ... order by(asc,desc),group by having
 - * sum,max,min,avg,count
 - * nvl
 - * like:_,%
 - * 分页：select rownum,ename from (select rownum rn,ename from emp where rn<6) where rn>2;
- * 能够理解JDBC的概述
 - * Java DataBase Connectivity
 - * sun：标准接口，数据库产商提供驱动（jar）
- * 能够掌握JDBC常见的API
 - * DriverManager
 - * Connection
 - * Statement
 - * ResutSet
- * 能够掌握JDBC的HelloWorld的开发
 - * 加载驱动
 - * 通过DriverManager获得链接Connection
 - * 通过链接Connection获得Statement
 - * 发送SQL并获取结果集
 - * 处理结果

* 释放资源

* 回顾

* XML的解析之SAX , Pull

- * SAXParser(parse,DefaultHandler)--->SAXParserFactory

- * DefaultHandler : startDocument,endDocument,startElement,endElement,characters

* VMware,XP,Oracle,远程连接

* Oracle

* SQL

- * DDL , DML , DQL , TCL , DCL,CCL

* Oracle数据类型

- * varchar2 , nvarchar2 , char , nchar , number (5 , 3) ,
date,timestamp,clob,nclob,blob

* 能够掌握数据定义语言DDL

```
1 * 准备
2 * sqlplus scott/tiger@192.168.1.121:1521/orcl
3 * 查询所有表
4 * select table_name from tabs;
5 * 查询表结构
6 * SET LONG 1000 -- 缓冲区用于显示LONG 的字节数只有80个字符
7 * set pagesize 50 设定页大小
8 * select dbms_metadata.get_ddl('TABLE','EMP') from dual;
9 * 'TABLE' 固定写法
10 * 查的表的表名，这里的表名必须大写
11
12 * create创建
13 --创建员工表
14 CREATE TABLE employee
15 (
16     emp_no CHAR(4) PRIMARY KEY NOT NULL, --工号，主键，非空
```

```

17     emp_name VARCHAR2(30) NOT NULL,--姓名,非空
18     emp_id VARCHAR2(18), --身份证号,代表18位整数
19     emp_age NUMBER(3,0) --年龄
20 );
21 -- ed --- /
22 --利用现有表创建新表, select 后边有多少字段, 新表将有多少个字段,主键不会被创建
23 CREATE TABLE emp_info AS SELECT emp_no,emp_age FROM employee;
24
25 * alter更改
26 --修改employee表, 添加emp_seat(座号)和emp_Address(住址) 两个列
27 ALTER TABLE employee ADD (emp_seat NUMBER(2,0) ,emp_address VARCHAR2(20));
28
29 --修改employee表的列的定义
30 ALTER TABLE employee MODIFY (emp_name VARCHAR2 (50),emp_address VARCHAR2(100));
31
32 --删除employee表中emp_ address和emp_seat列
33 ALTER TABLE employee DROP (emp_address,emp_seat);
34
35 --修改employee表列名
36 ALTER TABLE employee RENAME COLUMN emp_no TO rempno;
37
38 * drop
39 -- 删除employee表结构及数据
40 DROP TABLE employee;
41 * truncate
42 -- 只删除表数据, 每条删除的记录不写日志, 省资源, 效率比delete高
43 truncate table employee;
44
45 * 温馨提醒:
46 * 在没有备份情况下, 谨慎使用 drop 与 truncate

```

* 能够掌握数据操纵语言DML

* insert插入、delete删除、update更新

```

1 -- 创建员工表
2 CREATE TABLE employee
3 (
4     emp_no CHAR(5) PRIMARY KEY NOT NULL, --工号, 主键, 非空

```

```

5      emp_name VARCHAR2(30) NOT NULL,--姓名,非空
6      emp_id VARCHAR2(18), --身份证号,代表18位整数
7      emp_age NUMBER(3,0) --年龄
8 );
9
10 * insert 语句
11 insert into employee(emp_no,emp_name,emp_id,emp_age) values('lg001','张大','441!
12 insert into employee(emp_no,emp_name,emp_id,emp_age) values('lg002','张二','441!
13 insert into employee(emp_no,emp_name,emp_id,emp_age) values('lg003','张三','441!
14 insert into employee(emp_no,emp_name,emp_id,emp_age) values('lg004','张四','441!
15 commit;
16
17 * update 语句
18 --修改工号为'lg002'员工的名字为'李二'
19 update employee set emp_name='李二' where emp_no='lg002';
20 --所有员工的年龄都加1
21 update employee set emp_age=emp_age+1;
22
23 * select distinct
24 查询（结果）无重复数据行，distinct关键字使用：
25 --查询结果包括员工名字和年龄，如果存在员工名字和年龄都一样的多条记录将只返回一条
26 select distinct emp_name,emp_age from employee;
27
28 * delete
29 -- 删除并将该行的删除操作作为事务记录在日志中保存以便进行回滚操作。
30 delete from employee where emp_no='lg004'
31 --清空表，不写日志，省资源，效率高，属于数据定义语言
32 truncate table employee;
33
34 * 测试：
35 尝试--rollback;

```

* 能够掌握数据查询语言DQL

* DQL就是数据查询语言，数据库执行DQL语句不会对数据进行改变，而是让数据库发送结果集给客户端。

1 * 语法：

```

2 SELECT selection_list /*要查询的列名称*/
3   FROM table_list /*要查询的表名称*/
4  WHERE condition /*行条件*/
5  GROUP BY grouping_columns /*对结果分组*/
6  HAVING condition /*分组后的行条件*/
7  ORDER BY sorting_columns /*排序*/
8  LIMIT offset_start, row_count /*结果限定*/ (Oracle不支持,MYSQL支持)
9
10 * 创建学生表
11 CREATE TABLE stu (
12     sid CHAR(6) PRIMARY KEY NOT NULL,
13     sname VARCHAR2(50) NOT NULL,
14     age  NUMBER(3,0),
15     gender VARCHAR2(10)
16 );
17 * 插入数据
18 INSERT INTO stu VALUES('S_1001', 'liuYi', 35, 'male');
19 INSERT INTO stu VALUES('S_1002', 'chenEr', 15, 'female');
20 INSERT INTO stu VALUES('S_1003', 'zhangSan', 95, 'male');
21 INSERT INTO stu VALUES('S_1004', 'liSi', 65, 'female');
22 INSERT INTO stu VALUES('S_1005', 'wangWu', 55, 'male');
23 INSERT INTO stu VALUES('S_1006', 'zhaoLiu', 75, 'female');
24 INSERT INTO stu VALUES('S_1007', 'sunQi', 25, 'male');
25 INSERT INTO stu VALUES('S_1008', 'zhouBa', 45, 'female');
26 INSERT INTO stu VALUES('S_1009', 'wuJiu', 85, 'male');
27 INSERT INTO stu VALUES('S_1010', 'zhengShi', 5, 'female');
28 INSERT INTO stu VALUES('S_1011', 'xxx', NULL, NULL);
29
30 * 查询所有列
31 SELECT * FROM stu;
32 * 查询指定列
33 SELECT sid,sname,age,gender FROM stu;
34
35 * 条件查询
36 * 条件查询就是在查询时给出WHERE子句, 在WHERE子句中可以使用如下运算符及关键字:
37 * =、!=、<>、<、<=、>、>=;
38 * between and
39 * IN(set);
40 * IS NULL;
41 * AND;

```

```

42 * OR;
43 * NOT;
44
45 * 查询性别为女，并且年龄大于50的记录
46 * SELECT * FROM stu WHERE gender='female' AND age>50;
47 * 查询学号为S_1001，或者姓名为liSi的记录
48 * SELECT * FROM stu WHERE sid='S_1001' OR sname='liSi';
49 * 查询学号为S_1001，S_1002，S_1003的记录
50 * SELECT * FROM stu WHERE sid IN('S_1001','S_1002','S_1003');
51 * 查询学号不是S_1001，S_1002，S_1003的记录
52 * SELECT * FROM stu WHERE sid NOT IN('S_1001','S_1002','S_1003');
53 * 查询年龄为null的记录
54 * SELECT * FROM stu WHERE age IS NULL;
55 * 查询年龄在20到50之间的学生记录
56 * SELECT * FROM stu WHERE age BETWEEN 20 AND 50;
57 * 查询性别非男的学生记录
58 * SELECT * FROM stu WHERE gender!='male';
59 * SELECT * FROM stu WHERE gender<>'male';
60 * SELECT * FROM stu WHERE NOT gender='male';
61 * 查询姓名不为null的学生记录
62 * SELECT * FROM stu WHERE sname IS NOT NULL;
63 * SELECT * FROM stu WHERE NOT sname IS NULL;
64
65 * 模糊查询（like, _,%）
66 * 模糊查询必须使用LIKE关键字。其中 “_”匹配任意一个字母，5个“_”表示5个任意字母。
67 * “%”匹配0~n个任何字母
68 * 查询姓名由5个字母构成的学生记录
69 * SELECT * FROM stu WHERE sname LIKE '_____';
70 * 查询姓名由5个字母构成，并且第5个字母为“i”的学生记录
71 * SELECT * FROM stu WHERE sname LIKE '____i';
72 * 查询姓名以“z”开头的学生记录
73 * SELECT * FROM stu WHERE sname LIKE 'z%';
74 * 查询姓名中第2个字母为“i”的学生记录
75 * SELECT * FROM stu WHERE sname LIKE '_i%';
76 * 查询姓名中包含“a”字母的学生记录
77 * SELECT * FROM stu WHERE sname LIKE '%a%';
78
79 * set line 150:
80 * 参数linesize用于设置行的长度，即每行不超过100个字符。如果超过，则换行显示，默认
81 * set pagesize 100;

```

```
82      * 设定页大小
83
84 * 查看雇员的月薪与佣金之和
85      * select empno,ename,sal,comm,sal+comm from emp;
86      * select empno,ename,sal,comm,sal+nvl(comm,0) from emp;
87      * 温馨提醒:
88      * 任何东西与NULL相加结果还是NULL, 使用函数nvl把NULL转换成数值0
89 * 给列名添加别名
90      * SELECT empno,ename,sal,sal+nvl(comm,0) AS total FROM emp;
91      * SELECT empno,ename,sal,sal+nvl(comm,0) total FROM emp;
92 * 查询所有学生记录, 按年龄升序排序
93      * SELECT * FROM stu ORDER BY age ASC;
94 * 查询所有学生记录, 按年龄降序排序
95      * SELECT * FROM stu ORDER BY age DESC;
96 * 查询所有雇员, 按月薪降序排序, 如果月薪相同时, 按编号升序排序
97      * SELECT * FROM emp ORDER BY sal DESC,empno ASC;
98 * 聚合函数
99      聚合函数是用来做纵向运算的函数:
100      * COUNT(): 统计指定列不为NULL的记录行数;
101      * MAX(): 计算指定列的最大值, 如果指定列是字符串类型, 那么使用字符串排序运算;
102      * MIN(): 计算指定列的最小值, 如果指定列是字符串类型, 那么使用字符串排序运算;
103      * SUM(): 计算指定列的数值和, 如果指定列类型不是数值类型, 那么计算结果为0;
104      * AVG(): 计算指定列的平均值, 如果指定列类型不是数值类型, 那么计算结果为0
105 * 查询emp表中记录数:
106      * SELECT COUNT(*) FROM emp;
107 * 查询emp表中有佣金的人数
108      * SELECT COUNT(comm) FROM emp;
109 * 查询emp表中月薪大于2500的人数
110      * SELECT COUNT(*) FROM emp WHERE sal>2500;
111 * 统计月薪与佣金之和大于2500元的人数
112      * SELECT count(*) from emp where (sal+nvl(comm,0))>2500;
113 * 查询有佣金的人数, 以及有领导的人数
114      * SELECT COUNT(comm),COUNT(mgr) FROM emp;
115 * 查询所有雇员月薪和:
116      * SELECT SUM(sal) FROM emp;
117 * 查询所有雇员月薪和, 以及所有雇员佣金和
118      * SELECT SUM(sal),SUM(comm) FROM emp;
119 * 查询所有雇员月薪+佣金和
120      * SELECT SUM(sal+nvl(comm,0)) FROM emp;
121 * 统计所有员工平均工资
```

```

122 * SELECT AVG(sal) FROM emp;
123 * 查询最高工资和最低工资
124 * SELECT MAX(sal),MIN(sal) FROM emp;
125 * 查询每个部门的部门编号和每个部门的工资和
126 * SELECT deptno,SUM(sal) FROM emp GROUP BY deptno;
127 * 查询每个部门的部门编号以及每个部门的人数
128 * SELECT deptno,COUNT(*) FROM emp GROUP BY deptno;
129 * 查询每个部门的部门编号以及每个部门工资大于1500的人数
130 * SELECT deptno,COUNT(*) FROM emp WHERE sal>1500 GROUP BY deptno;
131 * 查询工资总和大于9000的部门编号以及工资和
132 * SELECT deptno,SUM(sal) FROM emp GROUP BY deptno HAVING SUM(sal)>9000;
133
134 温馨提醒:
135 * having与where的区别:
136     * having是在分组后对数据进行过滤.
137     * where是在分组前对数据进行过滤
138     * having后面可以使用分组函数(统计函数)
139     * where后面不可以使用分组函数。
140     * WHERE是对分组前记录的条件，如果某行记录没有满足WHERE子句的条件，
141     那么这行记录不会参加分组；而HAVING是对分组后数据的约束。
142
143 * 分页
144 * rownum
145     * 给查询结果返回一个数值表示次序，可以用来限制返回的条数，例如分页查询。
146     * select rownum,ename from emp;
147     * 前5条
148     * select rownum,ename from emp where rownum<6;
149     * 查询某页的记录：第3到5条记录（总3条数据）
150     * 正常想法： select rownum,ename from emp where rownum>2 and rownum<6;
151     * 但可惜的是Oracle不支持rownum>2的写法，不会返回数据
152     * 正确做法：
153     * 子查询： select rownum rn,ename from emp where rownum<6，先查出结果，作为
154     * select rownum,ename from (select rownum rn,ename from emp where rownum
155

```

* 能够理解JDBC的概述

* JDBC: Java DataBase Connectivity

* 由sun公司提供一套Java操作数据库的接口标准，而且由各大厂商提供驱动(jar包)

* 用Oracle数据库，就由Oracle厂商提供驱动（jar）

* 用MySQL数据库，就由MySQL厂商提供驱动（jar）

* ...

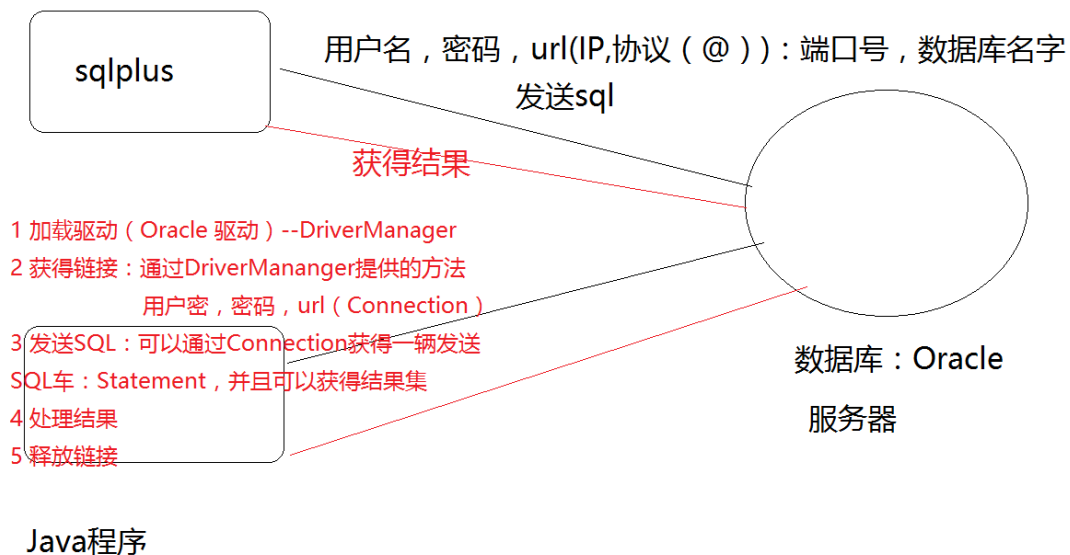
* 能够掌握JDBC常见的API

* DriverManager

* Connection

* Statement

* ResultSet



* 能够掌握JDBC的HelloWorld的开发

```
1 public static void main(String[] args) throws SQLException {
2     // 1 加载驱动（Oracle 驱动）--DriverManager
3     DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
4     // 2 获得链接：通过DriverMananger提供的方法
5     //     用户密，密码，url（Connection）
6     String user="scott";
7     String password="tiger";
```

```

8      String url="jdbc:oracle:thin:@192.168.1.121:1521:orcl";
9      Connection conn = DriverManager.getConnection(url, user, password);
10 //      3 发送SQL: 可以通过Connection获得一辆发送SQL车: Statement, 并且可以获得结果:
11      Statement statement = conn.createStatement();
12      String sql="select * from emp";
13      ResultSet rs=statement.executeQuery(sql);
14 //      4 处理结果
15      while(rs.next()) {
16          String empNo=rs.getString(1);// 数组下标是从开始, JDBC不从零开始,从1开
17          String eName=rs.getString(2);
18          String job=rs.getString(3);
19          String mgr=rs.getString(4);
20          String hireDate=rs.getString(5);
21          String sal=rs.getString(6);
22          String comm=rs.getString(7);
23          String deptNo=rs.getString(8);
24          System.out.println(empNo+":"+eName+":"+job+":"+mgr+":"+hireDate+":");
25      }
26 //      5 释放链接
27      rs.close();
28      conn.close();
29  }

```

30 结果:

```

31
32 7369:SMITH:CLERK:7902:1980-12-17 00:00:00:800:null:20
33 7499:ALLEN:SALESMAN:7698:1981-02-20 00:00:00:1600:300:30
34 7521:WARD:SALESMAN:7698:1981-02-22 00:00:00:1250:500:30
35 7566:JONES:MANAGER:7839:1981-04-02 00:00:00:2975:null:20
36 7654:MARTIN:SALESMAN:7698:1981-09-28 00:00:00:1250:1400:30
37 7698:BLAKE:MANAGER:7839:1981-05-01 00:00:00:2850:null:30
38 7782:CLARK:MANAGER:7839:1981-06-09 00:00:00:2450:null:10
39 7788:SCOTT:ANALYST:7566:1987-04-19 00:00:00:3000:null:20
40 7839:KING:PRESIDENT:null:1981-11-17 00:00:00:5000:null:10
41 7844:TURNER:SALESMAN:7698:1981-09-08 00:00:00:1500:0:30
42 7876:ADAMS:CLERK:7788:1987-05-23 00:00:00:1100:null:20
43 7900:JAMES:CLERK:7698:1981-12-03 00:00:00:950:null:30
44 7902:FORD:ANALYST:7566:1981-12-03 00:00:00:3000:null:20
45 7934:MILLER:CLERK:7782:1982-01-23 00:00:00:1300:null:10

```

46

47

