韩顺平的java教程中数据库的源代码（mysql版）

一：建库和建表：

1.创建数据库：

create database lsydb1 default character set utf8;

use lsydb1;

2.创建表（dept）：

create table dept (deptno int primary key, dname nvarchar(30), loc nvarchar(30)) default character set utf8;

3.创建表（emp）：

create table emp ( empno int primary key, ename nvarchar(30), job nvarchar(30), mgr int , hiredate datetime, sal decimal(6.2), comm decimal(6.2), deptno int ,foreign key(deptno) references dept(deptno)) default character set utf8;

4.向dept表插入数据（先插dept再插emp）：

insert into dept (deptno, dname, loc)

values (10, 'ACCOUNTING', 'NEW YORK'),

(20, 'RESEARCH', 'DALLAS'),

(30, 'SALES', 'CHICAGO'),

(40, 'OPERATIONS', 'BOSTON');

dept表如下：

+--------+------------+----------+

| deptno | dname | loc |

+--------+------------+----------+

| 10 | ACCOUNTING | NEW YORK |

| 20 | RESEARCH | DALLAS |

| 30 | SALES | CHICAGO |

| 40 | OPERATIONS | BOSTON |

+--------+------------+----------+

5.向emp表插入数据

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7369,'SMITH','CLERK',7902,'1980-12-17',800.00,NULL,20);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7499,'ALLEN','SALESMAN',7698,'1981-2-20',1600,300,30);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7521, 'WARD', 'SALESMAN', 7698, '1981-2-22', 1250, 500, 30);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7566, 'JONES', 'MANAGER', 7839, '1981-4-2', 2975, null, 20);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7654, 'MARTIN', 'SALESMAN', 7698, '1981-9-28', 1250, 1400, 30);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7698, 'BLAKE', 'MANAGER', 7839, '1981-5-1', 2850, NULL, 30);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7782, 'CLARK', 'MANAGER', 7839, '1981-6-9', 2450, NULL, 10);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7788, 'SCOTT', 'ANALYST', 7566, '1987-4-19', 3000, NULL, 20);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7839, 'KING', 'PRESIDENT', NULL, '1981-11-17', 5000, NULL, 10);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7844, 'TURNER', 'SALESMAN', 7698, '1981-9-8', 1500, 0, 30);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7876, 'ADAMS', 'CLERK', 7788, '1987-5-23', 1100, NULL, 20);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7900, 'JAMES', 'CLERK', 7698, '1981-12-3', 950, NULL, 30);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values ( 7902, 'FORD', 'ANALYST', 7566, '1981-12-3', 3000, NULL, 20);

insert into emp (empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7934, 'MILLER', 'CLERK', 7782, '1982-1-23', 1300, NULL, 10);

emp表：

+-------+--------+-----------+------+---------------------+------+------+--------+

| empno | ename | job | mgr | hiredate | sal | comm | deptno |

+-------+--------+-----------+------+---------------------+------+------+--------+

| 7369 | SMITH | CLERK | 7902 | 1980-12-17 00:00:00 | 800 | NULL | 20 |

| 7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 00:00:00 | 1600 | 300 | 30 |

| 7521 | WARD | SALESMAN | 7698 | 1981-02-22 00:00:00 | 1250 | 500 | 30 |

| 7566 | JONES | MANAGER | 7839 | 1981-04-02 00:00:00 | 2975 | NULL | 20 |

| 7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 00:00:00 | 1250 | 1400 | 30 |

| 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 00:00:00 | 2850 | NULL | 30 |

| 7782 | CLARK | MANAGER | 7839 | 1981-06-09 00:00:00 | 2450 | NULL | 10 |

| 7788 | SCOTT | ANALYST | 7566 | 1987-04-19 00:00:00 | 3000 | NULL | 20 |

| 7839 | KING | PRESIDENT | NULL | 1981-11-17 00:00:00 | 5000 | NULL | 10 |

| 7844 | TURNER | SALESMAN | 7698 | 1981-09-08 00:00:00 | 1500 | 0 | 30 |

| 7876 | ADAMS | CLERK | 7788 | 1987-05-23 00:00:00 | 1100 | NULL | 20 |

| 7900 | JAMES | CLERK | 7698 | 1981-12-03 00:00:00 | 950 | NULL | 30 |

| 7902 | FORD | ANALYST | 7566 | 1981-12-03 00:00:00 | 3000 | NULL | 20 |

| 7934 | MILLER | CLERK | 7782 | 1982-01-23 00:00:00 | 1300 | NULL | 10 |

+-------+--------+-----------+------+---------------------+------+------+--------+

二.基本查询操作：

1.

如何查询工资在2000到2500的员工情况

between是取两边的包括2000和2500.

2.模糊查询：

显示首字母为S的员工的姓名和工资

3.显示empno为123,345,800..的雇员情况。

使用 in关键字效率更高。

三.复杂查询操作：

1.显示员工最低和最高工资。

2.显示员工平均工资和总工资

3.显示高于平均工作雇员的姓名和工资,并显示平均工资。

(2).

4.显示员工人数：

group by:用于查询的结果分组统计。

having by:用于限制分组显示结果。

5.显示每个部门的平均工资和最高工资

mysql>结果下图：

+--------+-----------------------------+-----------------------------+

| deptno | 每个部门的平均工资 | 每个部门的最高工资 |

+--------+-----------------------------+-----------------------------+

| 10 | 2916.6667 | 5000 |

| 20 | 2175.0000 | 3000 |

| 30 | 1566.6667 | 2850 |

+--------+-----------------------------+-----------------------------+

6.显示每个部门的每种岗位的平均工资和最低工资：

结果图：

+-----------+----------+--------+-----------+

| avg(sal) | min(sal) | deptno | job |

+-----------+----------+--------+-----------+

| 1300.0000 | 1300 | 10 | CLERK |

| 2450.0000 | 2450 | 10 | MANAGER |

| 5000.0000 | 5000 | 10 | PRESIDENT |

| 3000.0000 | 3000 | 20 | ANALYST |

| 950.0000 | 800 | 20 | CLERK |

| 2975.0000 | 2975 | 20 | MANAGER |

| 950.0000 | 950 | 30 | CLERK |

| 2850.0000 | 2850 | 30 | MANAGER |

| 1400.0000 | 1250 | 30 | SALESMAN |

7.显示平均工资低于2000的部门和他的平均工资

（having 往往和group by结合使用，可以对分组查询结果进行筛选）

四.复杂查询（多表）

1.显示雇员的名字和部门的地点：

图：

+--------+---------+

| ename | loc |

+--------+---------+

| ALLEN | CHICAGO |

| WARD | CHICAGO |

| MARTIN | CHICAGO |

| BLAKE | CHICAGO |

| TURNER | CHICAGO |

| JAMES | CHICAGO |

+--------+---------+

2.显示部门号为10的部门名，员工名和工资。

图：

+------------+--------+------+

| dname | ename | sal |

+------------+--------+------+

| ACCOUNTING | CLARK | 2450 |

| ACCOUNTING | KING | 5000 |

| ACCOUNTING | MILLER | 1300 |

+------------+--------+------+

3.显示雇员名，雇员工资及所在部门的名字，并按部门排序。

图：

+--------+------+------------+

| ename | sal | dname |

+--------+------+------------+

| CLARK | 2450 | ACCOUNTING |

| KING | 5000 | ACCOUNTING |

| MILLER | 1300 | ACCOUNTING |

| SMITH | 800 | RESEARCH |

| JONES | 2975 | RESEARCH |

| SCOTT | 3000 | RESEARCH |

| ADAMS | 1100 | RESEARCH |

| FORD | 3000 | RESEARCH |

| ALLEN | 1600 | SALES |

| WARD | 1250 | SALES |

| MARTIN | 1250 | SALES |

| BLAKE | 2850 | SALES |

| TURNER | 1500 | SALES |

| JAMES | 950 | SALES |

+--------+------+------------+

自连接：同一张表的连接查询。

4.显示某员工的上级领导的姓名，比如“FORD”的上级。

+-------+

| ename |

+-------+

| JONES |

+-------+

5.显示公司每个员工名字和他上级的名字。

（1）.自连接。

图：

+--------+-------+

| ename | ename |

+--------+-------+

| SMITH | FORD |

| ALLEN | BLAKE |

| WARD | BLAKE |

| JONES | KING |

| MARTIN | BLAKE |

| BLAKE | KING |

| CLARK | KING |

| SCOTT | JONES |

| TURNER | BLAKE |

| ADAMS | SCOTT |

| JAMES | BLAKE |

| FORD | JONES |

| MILLER | CLARK |

外链接（左和右）

6.显示与SMITH同一部门的所有员工。(单行子查询)

7.显示和部门10的工作相同的雇员名字，岗位，工资和部门号。（多行子查询）

（部门包括10）

图：

+--------+-----------+------+--------+

| ename | job | sal | deptno |

+--------+-----------+------+--------+

| SMITH | CLERK | 800 | 20 |

| JONES | MANAGER | 2975 | 20 |

| BLAKE | MANAGER | 2850 | 30 |

| CLARK | MANAGER | 2450 | 10 |

| KING | PRESIDENT | 5000 | 10 |

| ADAMS | CLERK | 1100 | 20 |

| JAMES | CLERK | 950 | 30 |

| MILLER | CLERK | 1300 | 10 |

+--------+-----------+------+--------+

（部门不包括10）

图：

+-------+---------+------+--------+

| ename | job | sal | deptno |

+-------+---------+------+--------+

| SMITH | CLERK | 800 | 20 |

| JONES | MANAGER | 2975 | 20 |

| BLAKE | MANAGER | 2850 | 30 |

| ADAMS | CLERK | 1100 | 20 |

| JAMES | CLERK | 950 | 30 |

在from语句中使用子查询。

8.显示高于部门平均工资的员工信息。

图：

+-------+------+--------+-----------+

| ename | sal | deptno | myavg |

+-------+------+--------+-----------+

| KING | 5000 | 10 | 2916.6667 |

| JONES | 2975 | 20 | 2175.0000 |

| SCOTT | 3000 | 20 | 2175.0000 |

| FORD | 3000 | 20 | 2175.0000 |

| ALLEN | 1600 | 30 | 1566.6667 |

| BLAKE | 2850 | 30 | 1566.6667 |

+-------+------+--------+-----------+

9.显示第5到第10入职雇员（按时间的先后）

图：

+-------+--------+-----------+------+---------------------+------+------+--------+

| empno | ename | job | mgr | hiredate | sal | comm | deptno |

+-------+--------+-----------+------+---------------------+------+------+--------+

| 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 00:00:00 | 2850 | NULL | 30 |

| 7782 | CLARK | MANAGER | 7839 | 1981-06-09 00:00:00 | 2450 | NULL | 10 |

| 7844 | TURNER | SALESMAN | 7698 | 1981-09-08 00:00:00 | 1500 | 0 | 30 |

| 7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 00:00:00 | 1250 | 1400 | 30 |

| 7839 | KING | PRESIDENT | NULL | 1981-11-17 00:00:00 | 5000 | NULL | 10 |

| 7902 | FORD | ANALYST | 7566 | 1981-12-03 00:00:00 | 3000 | NULL | 20 |

| 7900 | JAMES | CLERK | 7698 | 1981-12-03 00:00:00 | 950 | NULL | 30 |

--左外连接和右外连接

（左外连接） ：左边的表记录全部显示，如果没有匹配记录就显示NULL

（右外连接） ：右边的表记录全部显示，如果没有匹配记录就显示NULL

例子如下图：



10.显示公司每位员工和他的上级名字，没有上级的名字也要显示。

图：

+--------+-------+

| ename | ename |

+--------+-------+

| SMITH | FORD |

| ALLEN | BLAKE |

| WARD | BLAKE |

| JONES | KING |

| MARTIN | BLAKE |

| BLAKE | KING |

| CLARK | KING |

| SCOTT | JONES |

| KING | NULL |

| TURNER | BLAKE |

| ADAMS | SCOTT |

| JAMES | BLAKE |

| FORD | JONES |

| MILLER | CLARK |

+--------+-------+

11.题目看下图：

（1）先创建goods表。

create table goods (goodsId nvarchar(50) primary key, goodsName nvarchar(80) not null, unitPrice decimal(8,2) check(unitPrice > 0), category nvarchar(3) check(catagory in("食物","日用品")), provider nvarchar(50)) default character set utf8;

图：

+-----------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+--------------+------+-----+---------+-------+

| goodsId | varchar(50) | NO | PRI | NULL | |

| goodsName | varchar(80) | NO | | NULL | |

| unitPrice | decimal(8,2) | YES | | NULL | |

| category | varchar(3) | YES | | NULL | |

| provider | varchar(50) | YES | | NULL | |

（2）创建customer表：

create table customer (customerId nvarchar(50) primary key, custName nvarchar(50) not null, address nvarchar(100), email nvarchar(100) unique, sex nchar(1) default "男" check(sex in("男","女")) , cardId nvarchar(18) ) default character set utf8;

图：

+------------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+-------+

| customerId | varchar(50) | NO | PRI | NULL | |

| custName | varchar(50) | NO | | NULL | |

| address | varchar(100) | YES | | NULL | |

| email | varchar(100) | YES | UNI | NULL | |

| sex | char(1) | YES | | 男 | |

| cardId | varchar(18) | YES | | NULL | |

+------------+--------------+------+-----+---------+-------+

（3）创建purchase表。

create table purchase (

customerId nvarchar(50) ,

goodsId nvarchar(50) ,

nums int check(nums>0) ,

CONSTRAINT `purchase\_fk\_customerId` FOREIGN KEY (`customerId`) REFERENCES `customer` (`customerId`),

CONSTRAINT `purchase\_fk\_goodsId'` FOREIGN KEY (`goodsId`) REFERENCES `goods` (`goodsId`)) default character set utf8;

图：

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+-------+

| customerId | varchar(50) | YES | MUL | NULL | |

| goodsId | varchar(50) | YES | MUL | NULL | |

| nums | int(11) | YES | | NULL | |

+------------+-------------+------+-----+---------+-------+