

H2 Day07_mysql进阶

H3 今日目标

1. 基本查询
2. 基本条件查询
3. 模糊查询
4. 排序查询
5. 分页查询
6. 分组查询
7. 聚合函数
8. 约束
9. 表关系
10. 连表查询
11. 子查询
12. 数据库的备份和还原

H3 1.基本查询

语法：

select 列名1,列名2,列名3... **from** 表名;

select * from 表名;查看表的所有数据

*****:所有列

#别名

在要显示的列后边 列名 **as** 别名 或者 列名 别名

#别名

select empno 编号,ename 姓名 **from** emp e;

H3 2.条件查询

select 列名 **from** 表名 **where** 条件;

条件规范:列名 符号 值;

符号: = ,>,>=,<,<=,!,<>(不等于)

#查询部门编号为30的员工信息

```
select * from emp where deptno=30;
```

#查询30部门以外其他部门信息

```
select * from emp where deptno!=30;
```

```
select * from emp where deptno<>30;
```

##关键字

or, and , in(),between A and B

or:表示如果两个条件都符合,那么都进行查询,如果只有一个符合,那么只查询一个

and:两个条件要同时都满足,如果不满足,没有符合的数据

in(值1,值2,值3...):分别取值1,值2,值3进行条件查询,也可以理解为是or的简写

between A and B:在A和B之间取值,包含A,B

#查询编号为7369,7499的员工编号,姓名,职位

```
select empno,ename,job from emp where empno=7369 or  
empno=7499;
```

#查询20部门中工资大于2000的员工信息

```
select * from emp where deptno=20 and sal>2000;
```

#查询1980年入职的员工信息

```
select * from emp where hiredate>='1980-01-01' and  
hiredate<='1980-12-31';
```

```
select * from emp where hiredate between '1980-01-  
01' and '1980-12-31';
```

H3 3.模糊查询

like 配合模糊查询使用有两个特殊的字符： **_** 任意一个字符
% 任意个字符

#用法: ... where 列名 **like** ...;

#查询名字中带有m的员工信息

#查询名字中带有m的员工信息

```
select * from emp where ename like '%m%';
```

#查询名字中第二个字母为m的员工信息

```
select * from emp where ename like '_m%';
```

H3 4.排序查询

order by 列名 **asc**(升序,默认)|**desc**(降序)|---> **order by**
列名 **asc|desc**,列名 **asc|desc**

#查看员工信息,按照工资的降序排列

```
SELECT * FROM emp ORDER BY sal DESC;
```

#查询20部门按照工资升序排列信息

```
SELECT * FROM emp WHERE deptno=20 ORDER BY sal;
```

#查询20部门按照工资升序排列信息,并按照编号的降序排列

```
SELECT * FROM emp WHERE deptno=20 ORDER BY sal,empno DESC;
```

H3 5.分页查询

limit m,n
m:起始位置的下标
n:截取的长度
-->比如取3-6行数据 **limit 2,4**

#查询前3行数据

```
SELECT * FROM emp LIMIT 0,3;
```

#简写,当第一个值为0的时候可以省略。

```
SELECT * FROM emp LIMIT 3;
```

H3 6.分组查询

group by 列名 **【having 条件】** ;

##==>如果在sql语句中存在分组,那么在select与from之间只能出现分组的列

#查询emp表中的职位信息;

select job from emp;

distinct(列):对列中的值去重复

#查询emp表中的职位信息;

SELECT DISTINCT(job) FROM emp;

SELECT job FROM emp GROUP BY job;

H3 7.聚合函数

sum(列): 求和,列的数据类型为数字

avg(列): 求平均,列的数据类型为数字

max(列): 求最大,列的数据类型为数字、日期

min(列): 求最小,列的数据类型为数字、日期

count(列): 计数,统计数量,任意列都可以,因此可以简写成
count(*)

#查询emp表总工资

SELECT SUM(sal) FROM emp;

#查询emp中人数

SELECT COUNT(*) FROM emp;

#查询最早入职的日期

SELECT MIN(hiredate) FROM emp;

#查询工资最高的员工信息

SELECT * FROM emp ORDER BY sal DESC LIMIT 1;

#查询各部门的最高工资

SELECT deptno,MAX(sal) FROM emp GROUP BY deptno; #先
分组,然后进行组内聚合

#执行优先级问题 **group by>聚合**

#查询各部门人数

SELECT deptno,COUNT(*) FROM emp GROUP BY deptno;

#查询人数大于4的部门

```
SELECT deptno,COUNT(*) 人数 FROM emp GROUP BY deptno  
HAVING 人数>4 ;
```

#查询部门编号中包含3且部门人数大于4的信息

```
SELECT deptno,COUNT(*) 人数 FROM emp GROUP BY deptno  
HAVING 人数>4 AND deptno LIKE '%3%';
```

```
SELECT deptno,COUNT(*) 人数 FROM emp WHERE deptno  
LIKE '%3%' GROUP BY deptno HAVING 人数>4;
```

#关键字的执行优先级: where > group by > 聚合 > having
where和having的区别?(面试题)

1.where后不能直接跟聚合,having可以

2.有group by 不一定有having,理论上有having就一定有group by

3.where的执行优先级要高于having

H3 8.约束

为了限制表中的列的数据,保证数据的完整性

1.主键约束 **primary key** 特点:有且仅有 ,目的能够快速检索到唯一一个对应的数据,

自增: 将主键列的维护交给系统。auto_increment,初始值1,每次获取列中最大的值进行+1

什么样的列可以作为主键列?

一、一般情况下,创建的表中列的值要求不重复的列可以作为主键列,

二、但是如果没有这么一个列,往往我们在表中添加一个id列作为表格的主键列,该列不具有任何意义。

三、一个表当中只能有一个主键

2.默认约束 **default** :当列中不添加数据的时候,取默认值,如果添加数据,直接使用添加的数据

3.不为空约束 **not null** :要求列的中不能为空

4.唯一约束 **unique**: 列的中值唯一,不重复

5.外键约束 **foreign key**,一张表约束另外一张表的数据

什么样的列能够作为外键列?

1. 能够使两个表产生关联的列就可以做为外键列
2. 一个表的外键往往是另外一个表的主键
3. 外键列中的值是可以重复的
4. 有外键列的表称为从表, 另外一个表叫主表

H3 9.表关系

1. 一对一

```
CREATE TABLE wife( #妻子表
    wid INT PRIMARY KEY,
    wname VARCHAR(50)
);
CREATE TABLE husband( #丈夫表
    hid INT PRIMARY KEY,
    hname VARCHAR(50),
    FOREIGN KEY(hid) REFERENCES wife(wid)
);
INSERT INTO wife VALUES(1, '赵丽颖');
INSERT INTO husband VALUES(1, '冯绍峰');
```

2. 一对多

员工和部门的例子

```
CREATE TABLE `emp` (
    `empno` INT(11) PRIMARY KEY,
    `ename` VARCHAR(10) DEFAULT NULL,
    `job` VARCHAR(9) DEFAULT NULL,
    `mgr` INT(11) DEFAULT NULL,
    `hiredate` DATE DEFAULT NULL,
    `sal` DECIMAL(7,2) DEFAULT NULL,
    `comm` DECIMAL(7,2) DEFAULT NULL,
    `deptno` VARCHAR(11) DEFAULT NULL,
    FOREIGN KEY(deptno) REFERENCES dept(deptno)
);
```

```

CREATE TABLE DEPT(
DEPTNO VARCHAR(11) PRIMARY KEY,
DNAME VARCHAR(14),
LOC VARCHAR(13)
);
3. 多对多
CREATE TABLE stu(
sid INT PRIMARY KEY,
sname VARCHAR(100)
);
-- 中间表 st
CREATE TABLE st(
stid INT PRIMARY KEY,
sid INT, #学生编号
tid INT, #教师编号
FOREIGN KEY(sid) REFERENCES stu(sid),
FOREIGN KEY(tid) REFERENCES teacher(tid)
);

CREATE TABLE teacher(
tid INT PRIMARY KEY,
tname VARCHAR(100)
);

```

H3 10.连表查询

连表查询：

1. 内连接查询

连接原理：

将两个表中从表的外键列的值与主表的主键列的值对应。

如果没有对应数据，则不显示

隐式内连接

select 列名 **from** 表名1,表名2 **where** 表名1.列名 = 表名2.列名....

显示内连接

select 列名 **from** 表名1 **join** 表名2 **on** 表名1.列名 = 表名2.列名 **where**...

#查询员工信息以及对应的部门信息

```
SELECT * FROM emp e JOIN dept d ON  
e.deptno=d.deptno;
```

```
CREATE TABLE grade(  
  id INT PRIMARY KEY,  
  lsal DECIMAL(6,2),  
  hsal DECIMAL(6,2),  
  lev CHAR(1)  
);
```

#查询emp表中员工信息及对应的工资等级信息

与emp表产生关联

```
SELECT * FROM emp e JOIN grade g ON sal BETWEEN lsal  
AND hsal;
```

2. 外连接查询

查询原理:

以主表为基准(将主表的数据全部显示),从表显示与主表对应的数据,如果没有对应的数据,那么则以null补齐

左连接查询 **left join ... on**

select 列名 **from** 表名1(主表) **left join** 表名2(从表) **on** 表名1.列名 = 表名2.列名 **where**...

右连接查询 **right join ... on**

select 列名 **from** 表名1(从表) **right join** 表名2(主表) **on** 表名1.列名 = 表名2.列名 **where**...

H3 11.子查询

一个sql语句中有多个select子句，也就意味着要执行多个查询，因此需要分先后，将先执行的select子句放入小括号中，因此当括号内部的查询我们成为子查询

select 列名 **from** 表名 **where** 列名 符号 (**select** 列名 **from** 表名 **where**....)

所以子查询常用的方式：将查询的结果作为另外一个查询的条件存在而使用

select 列名 **from** (**select** 列名 **from** 表名 **where**...) **where**(了解)

#查询smith所在部门的员工信息

#1.查询smith的部门

SELECT deptno **FROM** emp **WHERE** ename='smith';

#2.根据部门查询员工信息

SELECT * **FROM** emp **WHERE** deptno=?

#结合

SELECT * **FROM** emp **WHERE** deptno=(**SELECT** deptno **FROM** emp **WHERE** ename='smith');

#查询smith所在部门的其他员工信息

SELECT * **FROM** emp **WHERE** deptno=(**SELECT** deptno **FROM** emp **WHERE** ename='smith') **AND** ename<>'smith';

#查询smith所在部门的信息

SELECT * **FROM** dept **WHERE** deptno=(**SELECT** deptno **FROM** emp **WHERE** ename='smith');

H3 12.数据库的备份和还原

命令方式：备份

mysqldump -uroot -p密码 数据库名 > 文件目录路径\文件名.sql

版权所有 (C) 2007 Microsoft Corporation。保留所有权利。
C:\Users\Administrator>mysqldump -uroot -proot demo_db>d:\data.sql
C:\Users\Administrator>

注意:最后不要有分号

还原:

mysql -uroot -p密码 数据库名 < 文件目录路径\文件名.sql

```
C:\Users\Administrator>mysqldump -uroot -proot demo_db>d:\data.sql  
C:\Users\Administrator>mysql -uroot -proot demo<d:\data.sql  
C:\Users\Administrator>_
```