

MySQL数据库第四部分讲义

重点：掌握多表连接查询和视图

一 多表连接查询

1. 什么是多表连接查询？
2. 为什么要使用多表连接查询？
3. 如何建立多表连接？
4. 等值连接
 - 4.1 动手做任务（TODO 3个任务）
5. 不等值连接
 - 5.1 动手做任务（TODO 3个任务）
6. 内连接
 - 6.1 动手做任务（TODO 2个任务）
7. 自连接
 - 7.1 动手做任务（TODO 1个任务）
8. 外连接
 - 8.1 动手做任务（TODO 1个任务）
9. 课后任务（TODO 共9个任务）

二 视图

- （一）理解并口述（技术要点及面试题）
 - [口述1] 什么是视图？
 - [口述2] 视图的作用及优点？
 - [口述3] 创建视图的语法？
 - [口述4] 视图分类有哪些？
- （二）动手做（TODO 7个任务）
 1. 如何创建简单视图？
 2. 如何创建复杂视图？
 3. 如何创建连接视图？
 4. 如何创建CHECK约束视图？
 5. 如何修改视图？
 6. 如何查看视图？
 7. 如何删除视图？
- （三）课后任务（TODO 3个任务）

MySQL数据库第四部分讲义

重点：掌握多表连接查询和视图

一 多表连接查询

1. 什么是多表连接查询？

- 多表查询：在查询时，需要涉及到两个以上表的查询。

2. 为什么要使用多表连接查询？

业务需求：前面几节课所有操作的数据来源都是一张表，但在实际业务中由于数据库的设计规范，需要查询的信息往往来源于多个表。

例如：查询“所有员工信息，以及他们所在部门的基本信息”，设计到的是两个数据源，一个是员工表，另一个是部门表。

解决方案：

```
SELECT * FROM emp,dept;
```

但是查询出的结果是一个“笛卡尔积”：第一个表中的所有记录和第二张表中的所有记录逐个配对，但结果大部分数据并不是我们想要的，想要获取到符合条件的数据，就需要用到“连接查询”。

注意：使用多表查询时必须弄清楚表之间的关联，这是多表查询的基础。需要通过建立两个表之间的关系，使用连接条件筛选后查询出想要的结果。

3. 如何建立多表连接？

(1)、通过主键-外键连接。

(2)、通过相同（相关）字段连接。

问题：

1)、主键与外键名称必须相同吗？不是必须相同，习惯相同

2)、外键取值能为空吗？（可以）

4. 等值连接

- 多张表的连接值相等。

WHERE子句条件中写的多表连接的条件中使用等号连接相同的两个字段。

表1.字段 = 表2.字段

- 连接条件的个数有N-1个（N表示数据表的数目）

面试要点：等值连接的连接条件

4.1 动手做任务（TODO 3个任务）

--任务1：所有员工信息，以及他们所在部门的基本信息

```
SELECT e.*,d.dname,d.loc
FROM emp e,dept d
WHERE e.deptno = d.deptno;
```

--任务2：查询10号部门的所有员工信息，显示员工姓名，部门编号，部门名称。

```
SELECT e.ename,d.deptno,d.dname
FROM emp e,dept d
WHERE e.deptno = d.deptno AND d.deptno = 10;
```

-- 任务3 查询20号部门和30号部门的员工的员工平均工资，部门号，所在部门名称。

```
SELECT AVG(e.sal),d.deptno,d.dname
FROM emp e,dept d
WHERE e.deptno = d.deptno AND e.deptno IN (20,30)
GROUP BY d.deptno;
```

5. 不等值连接

- 多张表的连接值不相等。

WHERE子句条件中写的多表连接的条件中要使用除了等号以外的运算符连接多张表的连接值。在连接条件中可以使用的运算符有：>,<,<=,>=, <>,BETWEEN...AND

5.1 动手做任务 (TODO 3个任务)

```
-- 任务1: 查询出员工的姓名, 薪水及薪水等级。
SELECT e.ename,e.sal,sg.grade
FROM emp e,salgrade sg
WHERE e.sal BETWEEN sg.losal AND sg.hisal;
-- 或者
SELECT e.ename,e.sal,sg.grade
FROM emp e,salgrade sg
WHERE e.sal >= sg.losal AND e.sal <= sg.hisal;
-- 任务2: 查询10号部门工资小于2000的员工的姓名, 工资及工资等级。
SELECT e.ename,e.deptno,e.sal,sg.grade
FROM emp e,salgrade sg
WHERE e.sal BETWEEN sg.losal AND sg.hisal AND e.deptno=10 AND e.sal<2000;
-- 任务3: 查询工资等级处于第四级别的员工的姓名。
SELECT e.ename,e.sal,sg.grade
FROM emp e,salgrade sg
WHERE e.sal BETWEEN sg.losal AND sg.hisal AND sg.grade=4;
```

6. 内连接

- 内连接是一种常用的多表关联查询方式, 一般使用关键字INNER JOIN来实现。其中, INNER关键字可以省略, 当只使用JOIN关键字时, 语句只表示内连接操作。在使用内连接查询多个表时, 必须在FROM子句之后定义一个ON子句, 该子句用来指定两个表实现内连接的“连接条件”。

需要注意的是, 在内连接的检索结果中, 所有记录行都是满足连接条件的。内连接的语法格式如下:

```
SELECT column_list
FROM table_name1 [INNER] JOIN table_name2
ON join_condition;
```

- column_list: 字段列表
- table_name1和table_name2: 两个要实现内连接的表。
- join_condition: 实现内连接的条件表达式

- 内连接INNER JOIN...ON: 返回完全满足条件的记录。

面试要点: (1)、注意内连接的语法格式 (2)、注意内连接的条件是写在on子句后面

6.1 动手做任务 (TODO 2个任务)

```
-- 任务1 查询10号部门的所有员工信息, 显示员工姓名, 部门编号, 部门名称。
SELECT e.ename,d.deptno,d.dname
FROM emp e INNER JOIN dept d
ON e.deptno=d.deptno
WHERE d.deptno=10;
-- 任务2 查询在研发部('RESEARCH')工作员工的编号, 姓名, 工作部门, 工作所在地
SELECT e.empno,e.ename,d.dname,d.loc
FROM emp e JOIN dept d
ON e.deptno=d.deptno AND d.dname='RESEARCH';
```

7. 自连接

- 把一张表当成两张表用，自己和自己连接。给两张表取别名。

语法：SELECT select_list FROM table_name t1, table_name t2 WHERE search_condition; 面试要点: 用户可能会拥有“自引用式”的外键。“自引用式”外键是指表中的一个列可以是该表主键的一个外键。比如 :emp表中某一行的mgr列值（管理者列）可能是另一行的empno列值（员工列），因为管理者本身也是公司的员工。

7.1 动手做任务 (TODO 1个任务)

```
-- 任务： 显示BLAKE的上级领导的姓名
SELECT m.ename
FROM emp e,emp m
WHERE e.mgr = m.empno AND e.ename='BLAKE';
--或者
SELECT m.ename
FROM emp e INNER JOIN emp m
ON e.mgr=m.empno
WHERE e.ename='BLAKE';
```

8. 外连接

- 外连接：返回所有的匹配行和一些或全部不匹配的行，也就是会返回不满足条件的记录，这取决于所建立的外连接的类型。MySQL支持二种：(1). 左外连接 LEFT OUTER JOIN...ON (2). 右外连接 RIGHT OUTER JOIN...ON
- 注：OUTER可以省略。

面试要点：根据关联的表匹配返回的行记录与所想要查询的记录判断清楚到底需要用左外连接还是右外连接。

8.1 动手做任务 (TODO 1个任务)

```
-- 任务 统计各个部门的员工人数
-- 左外连接
SELECT d.deptno,COUNT(e.empno)
FROM dept d LEFT OUTER JOIN emp e
ON d.deptno = e.deptno
GROUP BY d.deptno;

-- 右外连接
SELECT d.deptno,COUNT(e.empno)
FROM emp e RIGHT OUTER JOIN dept d
ON d.deptno = e.deptno
GROUP BY d.deptno;
```

9. 课后任务 (TODO 共9个任务)

- ☒ 任务1 查询统计部门员工人数大于3的部门的员工人数。
- ☒ 任务2 显示所有员工的姓名及其所在部门的名称和工资
- ☒ 任务3 查询每个员工的信息及工资级别
- ☒ 任务4 查询没有员工的部门信息

利用之前创建的学生表，课程表，学生选修成绩表完成下面任务：

- ☑ 任务5 统计每门课所有学生的总成绩。
- ☑ 任务6 列出每个学生的姓名和所有课程平均成绩。
- ☑ 任务7 列出各科的平均成绩、最高成绩、最低成绩和选课人数。按课程号升序排序。
- ☑ 任务8 查询选修1002课程的学生姓名
- ☑ 任务9 查询MySQL不及格的学生姓名和成绩
- ☑ 任务10 查询平均分不及格的学生学号，姓名，平均分

二 视图

(一) 理解并口述（技术要点及面试题）

[口述1] 什么是视图？

- 视图是由一个或者多个表组成的虚拟表。实际就是SELECT 语句。

[口述2] 视图的作用及优点？

1. 视图的作用：（1）方便用户操作：要求所见即所需，无需添加额外的查询条件，直接查看，简化用户的数据查询和处理。（2）增加数据的安全性：通过视图，用户只能查看或修改指定的数据。（3）提高表的独立逻辑性：原有数据表结构的变化，不会影响视图，如果修改原有列，则只需修改视图即可。
2. 使用视图的优点：（1）为用户集中数据，简化用户的数据查询和处理。（2）屏蔽数据库的复杂性，用户不必了解数据库的复杂性。（3）简化用户权限的管理，只授予用户使用视图的权限。（4）便于数据共享，多个用户不必都定义所需的数据。（5）可以重新组织数据，以便关联到其他应用中。

[口述3] 创建视图的语法？

视图的创建语法：CREATE VIEW 视图名(列名1，列名2，列名3...) AS SELECT 语句 [WITH CHECK OPTION]; 备注：定义视图使用WITH CHECK OPTION，表示更新视图时要保证在该视图的权限范围之内，检查约束。创建视图原则：定义视图的查询可以采用复杂的SELECT语法，可以包含连接、分组、子查询。定义视图的查询中不能使用ORDER BY子句，因为ORDER BY 子句可以在从视图中查询数据时使用。

[口述4] 视图分类有哪些？

1. 简单视图 是指基于单个表所建立的，不包含任何函数，表达式以及分组数据的视图。在该视图上可以执行DML语句（即可执行增、删、改操作）。
2. 复杂视图 指包含函数、表达式或者分组函数的视图，使用复杂视图的主要目的是为了简化查询操作。在该视图上执行DML语句时必须符合特定条件。复杂视图有部分可以DML操作，若复杂视图中有分组函数、GROUP BY子句、DISTINCT等则不可以做DML操作。注：在定义复杂视图时必须为函数或表达式定义别名。
3. 连接视图 是指基于多个表所建立的视图，使用连接视图的主要目的是为了简化连接查询。一般来说不会在该视图上执行INSERT、UPDATE、DELETE操作。
4. CHECK约束视图 WITH CHECK OPTION用于在视图上定义CHECK约束,即在该视图上执行INSERT或UPDATE操作时,数据必须符合约束条件。

(二) 动手做（TODO 7个任务）

1. 如何创建简单视图？

-- 任务1 创建一个简单视图，查询员工的姓名，工资，部门号

```

--不指定视图列名
CREATE VIEW v_emp1
AS
SELECT ename e,sal s,deptno d
FROM emp;
-- 查看视图v_emp1的信息。
DESC v_emp1;
SELECT * FROM v_emp1;

-- 指定视图的列名
CREATE VIEW v_emp11(e_name,e_sal,e_deptno)
AS
SELECT ename e,sal s,deptno d
FROM emp;
-- 查看视图v_emp1的信息。
DESC v_emp11;
SELECT e_name,e_sal FROM v_emp11;

```

2. 如何创建复杂视图?

```

-- 任务2 创建一个复杂视图查询各个部门的部门名称和各部门的平均工资
CREATE VIEW v_deptAvgSal
AS
SELECT d.dname,AVG(e.sal) avgsal
FROM emp e,dept d
WHERE e.deptno=d.deptno
GROUP BY d.dname;
-- 查看视图
DESC v_deptAvgSal;
SHOW CREATE VIEW v_deptAvgSal;
SELECT * FROM v_deptAvgSal;

```

3. 如何创建连接视图?

```

-- 任务3 创建一个连接视图v_emp, 可以通过该视图查看所有员工的员工号, 员工姓名, 工资, 部门号, 部门名称, 工资等级。
CREATE VIEW v_emp
AS
SELECT e.empno,e.ename,e.sal,e.deptno,d.dname,sg.grade
FROM emp e,dept d,salgrade sg
WHERE e.deptno=d.deptno AND e.sal BETWEEN sg.losal AND sg.hisal;

--查看视图
SELECT * FROM v_emp;

```

4. 如何创建CHECK约束视图?

```
-- 任务4 创建一个CHECK约束视图，查询员工的姓名，工资，部门号
CREATE VIEW v_emp2
AS
SELECT ename e,sal s,deptno d
FROM emp
WITH CHECK OPTION;
```

5. 如何修改视图？

语法：CREATE OR REPLACE VIEW view_name AS SELECT 语句;

```
-- 任务5 修改视图v_emp2
CREATE OR REPLACE VIEW v_emp2
AS
SELECT empno,ename,sal,job,deptno
FROM emp;
```

6. 如何查看视图？

查看视图的结构的语法：DESC 视图名; 查看视图的详细结构的语法：SHOW CREATE VIEW 视图名; 查看视图里的数据内容与查看数据表一样的语法：SELECT * FROM 视图名;

```
-- 任务6：查看视图v_deptAvgSal的结构和内容信息。
DESC v_deptAvgSal;
SHOW CREATE VIEW v_deptAvgSal;
SELECT * FROM v_deptAvgSal;
```

7. 如何删除视图？

语法：DROP VIEW 视图名;

```
-- 任务7 删除视图v_emp2;
DROP VIEW v_emp2;
```

(三) 课后任务 (TODO 3个任务)

- ☒ 任务1 创建视图“雇员表”包含了30号部门的雇员编号、姓名、以及薪水并使用了别名。
- ☒ 任务2 创建一视图查询各个部门员工的最低工资，最高工资，平均工资，所在部门名称。
- ☒ 任务3 创建一个视图查询每个学生不及格的课程名称和成绩。