

多表查询

多表查询

多表关系

概述

一对多（多对一）

多对多

一对一

多表查询概述

多表查询分类

连接查询

内连接

外连接

子查询

内连接

内连接查询语法

隐式内连接

显式内连接

内连接演示

外连接

外连接查询语法

左外连接

右外连接

外连接演示

自连接

自连接查询语法

自连接演示

联合查询-union

联合查询演示

子查询

子查询的介绍及分类

标量子查询

标量子查询演示

列子查询

列子查询演示

行子查询

行子查询演示

表子查询

表子查询演示

多表查询练习

小结

多表关系

概述

项目开发中，在进行数据库表结构设计时，会根据业务需求及业务模块之间的关系，分析并设计表结构，由于业务之间相互关联，所以各个表结构之间也存在着各种关系，基本上分为三种。

一对多（多对一）

案例：部门与员工的关系

关系：一个部门对应多个员工，一个员工对应一个部门

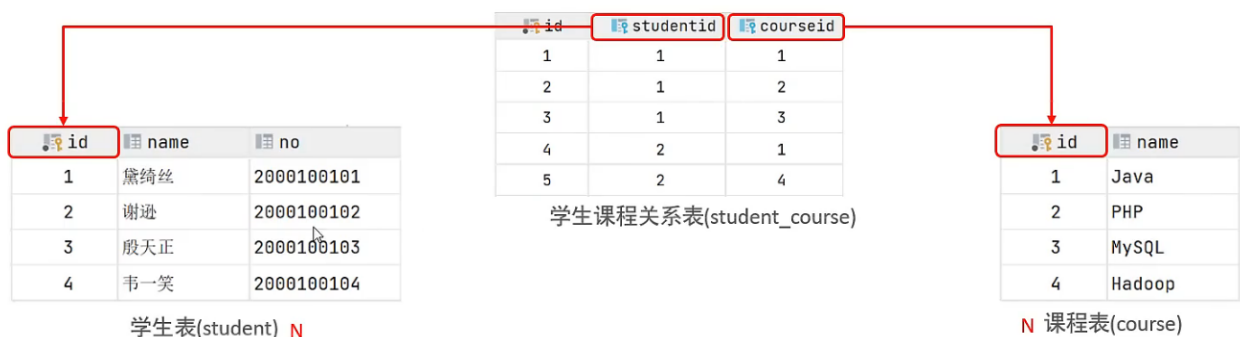
实现：在多的一方建立外键，指向一的一方的主键

多对多

案例：学生与课程的关系

关系：一个学生可以选修多门课程，一门课程也可以供多个学生选择

实现：建立第三张中间表，中间表至少包含两个外键，分别关联两方主键



实现上表中学生课程关系表

```

1 CREATE TABLE student_course(
2     id INT AUTO_INCREMENT COMMENT '主键' PRIMARY KEY,
3     studentid INT NOT NULL COMMENT '学生ID',
4     courseid INT NOT NULL COMMENT '课程ID',
5     CONSTRAINT fk_courseid FOREIGN KEY(courseid) REFERENCES
course(id),
6     CONSTRAINT fk_studentid FOREIGN KEY(studentid) REFERENCES
student(id),
7 );COMMENT '学生课程中间表';

```

一对一

案例：用户与用户详情的关系

关系：一对一关系，多用于单表拆分，将一张表的基础字段放在一张表中，其他详情字段放在另一张表中，以提升操作效率

实现：在任意一方加入外键，关联另外一方的主键，并且设置外键为唯一的(UNIQUE)



多表查询概述

- 指从多张表中查询数据
- 笛卡尔积：笛卡尔乘积是指在数学中，两个集合A和集合B的所有组合情况。（在多表查询时，需要消除无效的笛卡尔积）。

```

1 select * from emp, dept where emp.dept_id = dept.id;

```

多表查询分类

连接查询

内连接

- 相当于查询A、B交集部分数据

外连接

- 左外连接：查询左表所有数据，以及两张表交集部分数据
- 右外连接：查询右表所有数据，以及两张表交集部分数据

子查询

内连接

内连接查询的是两张表交集的部分

内连接查询语法

隐式内连接

```
1 SELECT 字段列表 FROM 表1, 表2 WHERE 条件...;
```

显式内连接

```
1 SELECT 字段列表 FROM 表1 [INNER] JOIN 表2 ON 连接条件...;
```

内连接演示

查询每一个员工的姓名，及关联的部门的名称（隐式内连接实现）

```
1 SELECT emp.name, dept.name FROM emp, dept WHERE emp.dept_id =  
dept.id;
```

查询每一个员工的姓名，及关联的部门的名称（显式内连接实现）

```
1 表结构: emp, dept
2 链接条件: emp.dept_id = dept.id
3
4 SELECT emp.name, dept.name FROM emp INNER JOIN dept ON
   emp.dept_id = dept.id;
```

外连接

外连接查询语法

左外连接

```
1 SELECT 字段列表 FROM 表1 LEFT[OUTER] JOIN ON ...;
```

相当于查询表1（左表）的所有数据 包含 表1和表2交集部分的数据。

右外连接

```
1 SELECT FROM RIGHT[OUTER] JOIN ON ...;
```

相当于查询表2（右表）的所有数据 包含 表1和表2交集部分的数据。

外连接演示

查询emp表的所有数据，和对应的部门信息（左外连接）

```
1 SELECT e.*, d.name FROM emp e LEFT OUTER JOIN dept d ON e.dept_id
   = d.id;
```

查询dept表的所有数据，和对应的员工信息（右外连接）

```
1 SELECT d.*, e.* FROM emp e RIGHT OUTER JOIN dept d ON e.dept_id =
   d.id;
```

自连接

自连接查询语法

```
1 SELECT 字段列表 FROM 表A 别名A JOIN 表A 别名B ON 条件 ...;
```

自连接查询，可以是内连接查询，也可以是外连接查询。

自连接演示

在自连接SQL语句中，表必须起别名

1.查询员工及其所属领导的名字(内连接)

```
1 SELECT a.name, b.name FROM emp AS a, emp AS b WHERE a.managerid = b.id;
```

2.查询所有员工及其领导的名字，如果员工没有领导，也需要查询出来（外连接）

```
1 SELECT a.name AS '员工', b.name AS '领导' FROM emp AS a LEFT JOIN emp AS b ON a.managerid = b.id;
```

联合查询-union

对于union查询，就是把多次查询的结果合并起来，形成一个新的查询结果集。

```
1 SELECT 字段列表 FROM 表A ...  
2 UNION[ALL]  
3 SELECT 字段列表 FROM 表B ...;
```

对于联合查询的多张表的列数必须保持一致，字段类型也需要保持一致

union all会将全部的数据直接合并到一起，**union**会对合并之后的数据去重

联合查询演示

1.将薪资低于5000的员工，和年龄大于50岁的员工全部查询出来

```

1  /*UNION ALL --- 合并结果且不去重*/
2  SELECT * FROM emp WHERE salary < 5000
3  UNION ALL
4  SELECT * FROM emp WHERE age > 50;
5
6  /*UNION --- 合并结果且去重*/
7  SELECT * FROM emp WHERE salary < 5000
8  UNION
9  SELECT * FROM emp WHERE age > 50;

```

子查询

子查询的介绍及分类

概念：SQL语句中嵌套SELECT语句，称为嵌套语句，又称子查询。

```

1  SELECT * FROM t1 WHERE column1 = (SELECT column1 FROM t2);

```

子查询外部的语句可以是INSERT/UPDATE/DELETE/SELECT的任何一个

根据子查询结果不同，可分为四类

- 标量子查询（子查询结果为单个值）
- 列子查询（子查询结果为一列）
- 行子查询（子查询结果为一行）
- 表子查询（子查询结果为多行多列）

根据子查询位置，分为：WHERE之后、FROM之后、SELECT之后。

标量子查询

子查询返回的结果是单个值（数字、字符串、日期等），最简单的形式，这种子查询称为**标量子查询**。

常量操作符：=、<>、>、>=、<、<=

标量子查询演示

1.查询“销售部”的所有员工信息

```

1  /* a.查询“销售部”部门ID */
2  SELECT id FROM dept WHERE name = '销售部';
3
4  /* b.根据销售部门ID，查询员工信息 */
5  SELECT * FROM emp WHERE dept_id = 4;
6
7  SELECT * FROM emp WHERE dept_id = (SELECT id FROM dept WHERE name
   = '销售部');
```

2.查询在“方东白”入职之后的员工信息

```

1  /* a.查询 方东白 的入职日期 */
2  SELECT entrydate FROM emp WHERE name = '方东白';
3
4  /* 查询指定入职日期之后的员工信息 */
5  SELECT * FROM emp WHERE entrydate > '2009-02-12';
6
7  SELECT * FROM emp WHERE entrydate > (SELECT entrydate FROM emp
   WHERE name = '方东白');
```

列子查询

子查询返回的结果是一列（可以是多列），这种子查询称为**列子查询**。

常用操作符：IN、NOT IN、ANY、SOME、ALL

操作符	描述
-----	----

操作符	描述
IN	在指定的集合范围子内，多选一
NOT IN	不在指定的集合范围之内
ANY	子查询返回列表中，有任意一个满足即可
SOME	与ANY等同，使用SOME的地方都可以使用ANY
ALL	子查询返回列表的所有制必须满足

列子查询演示

1.查询“销售部”和“市场部”的所有员工信息

```

1  /* a.查询“销售部”和“市场部”的所有员工信息 */
2  SELECT id FROM dept WHERE name = '销售部' OR name = '市场部';
3
4  /* b.根据部门ID，查询员工信息 */
5  SELECT * FROM emp WHERE dept_id IN (2, 4);
6
7  SELECT * FROM emp WHERE dept_id IN (SELECT id FROM dept WHERE
   name = '销售部' OR name = '市场部');

```

2.查询比财务部所有人工资都高的员工信息

```

1  /* a.查询财务部人员工资 */
2  SELECT id FROM dept WHERE name = '财务部';
3  SELECT salary FROM emp WHERE dept_id = 3;
4  SELECT salary FROM emp WHERE dept_id = (SELECT id FROM dept WHERE
   name = '财务部');
5
6  /* b.比财务部所有人工资都高的员工信息*/
7  SELECT * FROM emp WHERE salary > ALL (SELECT salary FROM emp
   WHERE dept_id = (SELECT id FROM dept WHERE name = '财务部'));

```

3.查询比研发部其中任意一人工资高的员工信息

```

1  /* a. 查询研发部所有人工资 */
2  SELECT salary FROM emp WHERE dept_id = (SELECT id FROM dept WHERE
   name = '研发部');
3
4  /* b. 比研发部其中任意一人工资高的员工信息 */
5  SELECT * FROM emp WHERE salary > ANY (SELECT salary FROM emp
   WHERE dept_id = (SELECT id FROM dept WHERE name = '研发部'));

```

行子查询

子查询返回的结果是一行（可以是多列），这种子查询称为行子查询。

常用操作符：=、<>、IN、NOT IN

行子查询演示

1. 查询与“张无忌”的薪资及直属领导相同的员工信息

```

1  /* a. 查询“张无忌”的薪资及直属领导 */
2  SELECT salary, managerid FROM emp WHERE name = '张无忌';
3
4  /* b. 查询与“张无忌”的薪资及直属领导相同的员工信息 */
5  SELECT * FROM Emp WHERE (salary, managerid) = (12500, 1);
6
7  SELECT * FROM Emp WHERE (salary, managerid) = (SELECT salary,
   managerid FROM emp WHERE name = '张无忌');

```

表子查询

子查询返回的结果是多行多列，这种子查询称为表子查询

常用的操作符：IN

表子查询演示

1. 查询于“鹿杖客”，“宋远桥”的职位和薪资相同的员工信息

```

1 a. 查询鹿杖客、宋远桥的职位和薪资
2 SELECT job, salary FROM emp WHERE name = '鹿杖客' OR name = '宋远
   桥';
3
4 b. 查询与鹿杖客、宋远桥的职位和薪资相同的员工信息
5 SELECT * FROM emp WHERE (job, salary) IN (SELECT job, salary FROM
   emp WHERE name = '鹿杖客' OR name = '宋远桥');

```

2. 查询入职日期是“2006-01-01”之后的员工信息，及其部门信息。

```

1 a. 查询入职日期是“2006-01-01”之后的员工信息
2 SELECT * FROM emp WHERE entrydate > '2006-01-01';
3
4 b. 查询这部分员工，对应的部门信息
5 SELECT e.*, d.* FROM (SELECT * FROM emp WHERE entrydate > '2006-
   01-01') e LEFT JOIN dept d ON e.dept_id = d.id;

```

多表查询练习

1. 查询员工的姓名、年龄、职位、部门信息（隐式内连接）

```

1 表: emp, dept
2 连接条件: emp.dept_id = dept.id
3 SELECT e.name, e.age, e.job, d.name FROM emp e, dept d WHERE
   e.dept_id = d.id

```

2. 查询年龄小于30岁的员工姓名、年龄、职位、部门信息

```

1 表: emp, dept
2 连接条件: emp.dept_id = dept.id
3 SELECT e.name, e.age, e.job, d.name FROM emp e INNER JOIN dept d
   ON e.dept_id = d.id WHERE e.age < 30;

```

3. 查询拥有员工的部门ID、部门名称

```

1 表: emp, dept
2 连接条件: emp.dept_id = dept.id
3 SELECT DISTINCT d.id, d FROM emp e, dept d WHERE e.dept_id =
  d.id;

```

4.查询所有年龄大于40岁的员工，及其归属的部门名称；如果员工没有分配部门，也需要展示出来

```

1 表: emp, dept
2 连接条件: emp.dept_id = dept.id
3 SELECT e.*, d.name FROM emp e LEFT JOIN dept d ON e.dept_id =
  d.id WHERE e.age > 40;

```

5.查询所有员工的工资等级

```

1 表: emp, salgrade
2 连接条件: emp.salary >= salgrade.losal and emp.salary <=
  salgrade.hisal
3 SELECT e.*, s.grade, s.losal, s.hisal FROM emp e, salgrade s
  WHERE e.salary BETWEEN s.losal AND s.hisal

```

6.查询“研发部”所有员工的信息及工资等级

```

1 表: emp, salgrade, dept
2 连接条件: emp.salary >= salgrade.losal and emp.salary <=
  salgrade.hisal
3 查询条件: dept.name = '研发部'
4 SELECT e.*, s.grade
5 FROM emp e,
6      dept d,
7      salgrade s
8 WHERE e.dept_id = d.id
9      AND e.salary BETWEEN s.losal AND s.hisal
10     AND d.name = '研发部';

```

7.查询“研发部”员工的平均工资

```
1 SELECT AVG(e.salary) FROM emp e, dept d WHERE e.dept = d.id AND
   d.name = '研发部';
```

8.查询工资比“灭绝”高的员工信息

```
1 a.查询工资比“灭绝”的薪资
2 b.查询比其薪资高的员工信息
3 SELECT *
4 FROM emp
5 WHERE salary > (SELECT salary FROM emp WHERE name = '灭绝')
```

9.查询比平均薪资高的员工信息

```
1 SELECT *
2 FROM emp
3 WHERE salary > (SELECT AVG(salary) FROM emp)
```

10.查询低于本部门平均工资的员工信息

```
1 SELECT * FROM emp e2 WHERE e2.salary < (SELECT AVG(e1.salary)
   FROM emp e1 WHERE e1.dept_id = e2.dept_id)
```

11.查询所有的部门信息，并统计部门的员工人数

```
1 SELECT d.id, d.name, (SELECT COUNT(*) FROM emp e WHERE e.dept_id
   = d.id) '人数' FROM dept d;
```

12.查询所有学生的选课情况，展示出学生名称，学号，课程名字

```
1 表: student, course, student_course
2 连接条件: student.id = student_course.studentid, course.id =
   student_course.courseid
3 SELECT s.name, s.no, c.name
4 FROM student s, student_course sc, course c
5 WHERE s.id = sc.studentid
6       AND sc.courseid = c.id
```

小结

1.多表关系

- 一对多：在多的一方设置外键，关联一得到一方的主键
- 多对多：建立中间表，中间表包含两个外键，关联两张表的主键
- 一对一：用于表结构拆分，在其中任何一方设置外键(UNIQUE)，关联另一方的主键

2.多表查询

内连接

- 1 隐式: `SELECT ... 表A, 表B WHERE 条件 ...`
- 2 显式: `SELECT ... FROM 表A INNER JOIN 表B ON 条件 ...`

外连接

- 1 左外: `SELECT ... FROM 表A LEFT JOIN 表B ON 条件 ...`
- 2 右外: `SELECT ... FROM 表A RIGHT JOIN 表B 条件 ON 条件 ...`

自连接

- 1 `SELECT ... FROM 表A 别名1, 表A 别名2 WHERE 条件 ...`

子查询

- 1 标量子查询、列子查询、行子查询、表子查询

[返回文首](#)