

多表查询

概述

笛卡尔积

连接查询

内连接

外连接

左外连接

右外连接

自连接

联合查询

子查询

标量子查询

列子查询

行子查询

表子查询

概述

▼ 查询 student2 和 stu表

SQL |

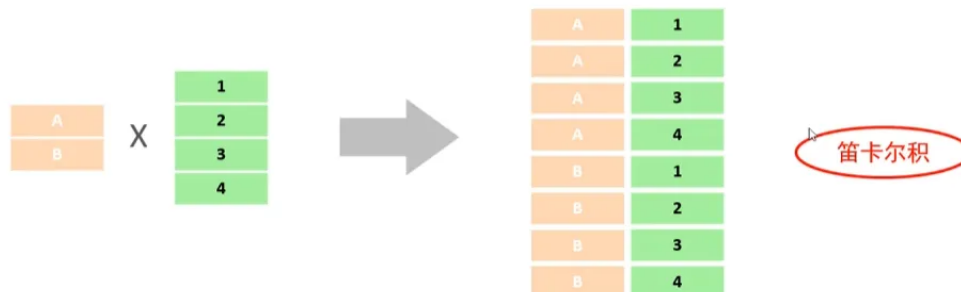
```
1 SELECT *FROM emp,dept WHERE emp.dept_id=dept.id;
```

id	name	age	job	salary	entrydate	ma
1	金庸	66	总裁	20000	2000-01-01	
1	金庸	66	总裁	20000	2000-01-01	
1	金庸	66	总裁	20000	2000-01-01	
1	金庸	66	总裁	20000	2000-01-01	
1	金庸	66	总裁	20000	2000-01-01	
2	张无忌	20	项目经理	12500	2005-12-05	
2	张无忌	20	项目经理	12500	2005-12-05	
2	张无忌	20	项目经理	12500	2005-12-05	
2	张无忌	20	项目经理	12500	2005-12-05	
2	张无忌	20	项目经理	12500	2005-12-05	
3	杨逍	33	开发	8400	2000-11-03	
3	杨逍	33	开发	8400	2000-11-03	
3	杨逍	33	开发	8400	2000-11-03	
3	杨逍	33	开发	8400	2000-11-03	
3	杨逍	33	开发	8400	2000-11-03	
4	韦一笑	48	开发	11000	2002-02-05	
4	韦一笑	48	开发	11000	2002-02-05	
4	韦一笑	48	开发	11000	2002-02-05	
4	韦一笑	48	开发	11000	2002-02-05	
4	韦一笑	48	开发	11000	2002-02-05	
5	常遇春	43	开发	10500	2004-09-07	
5	常遇春	43	开发	10500	2004-09-07	
5	常遇春	43	开发	10500	2004-09-07	
5	常遇春	43	开发	10500	2004-09-07	
5	常遇春	43	开发	10500	2004-09-07	

如图所示，查询出的数据竟然有 $5 \times 6 = 30$ 条，这是笛卡尔积现象

笛卡尔积

- 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集合 和 B集合的所有组合情况。(在多表查询时，需要消除无效的笛卡尔积)



要想避免笛卡尔积这一现象，需要添加判定条件

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

消息	摘要	结果 1	剖析	状态						
	id	name	age	job	salary	entrydate	managerid	dept_id	id(1)	name(1)
▶	1	金庸	66	总裁	20000	2000-01-01	(Null)	5	5	总经办
	2	张无忌	20	项目经理	12500	2005-12-05	1	1	1	研发部
	3	杨逍	33	开发	8400	2000-11-03	2	1	1	研发部
	4	韦一笑	48	开发	11000	2002-02-05	2	1	1	研发部
	5	常遇春	43	开发	10500	2004-09-07	3	1	1	研发部
	6	小昭	19	程序员鼓励	6600	2004-10-02	2	1	1	研发部

连接查询

● 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

自连接：当前表与自身的连接查询，自连接必须使用表别名

内连接

内连接查询语法：

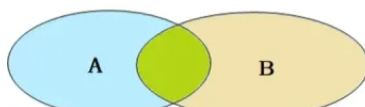
➤ 隐式内连接

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

➤ 显式内连接

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

内连接查询的是两张表交集的部分



```
1  -- 隐式内连接
2  SELECT emp.name,dept.name FROM emp,dept WHERE emp.dept_id=dept.id;
3  -- 显式内连接
4  SELECT emp.name,dept.name FROM emp INNER JOIN dept ON dept.id=emp.dept_id;
```

外连接

左外连接

➤ 左外连接

```
SELECT 字段列表 FROM 表1 LEFT [OUTER] JOIN 表2 ON 条件 ...;
```

```
1  -- 左外连接
2  SELECT emp.name,dept.name FROM emp left OUTER JOIN dept ON emp.dept_id=dept.id;
```

右外连接

➤ 右外连接

```
SELECT 字段列表 FROM 表1 RIGHT [OUTER] JOIN 表2 ON 条件 ...;
```

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

```
1  -- 右外连接
2  SELECT emp.name,dept.name FROM emp RIGHT OUTER JOIN dept ON emp.dept_id=dept.id;
```

自连接

自连接查询语法:

```
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。

必须要写别名

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

联合查询

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。

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

▼ UNION后面不加ALL，对查询的结果去重

SQL |

```
1 SELECT *from emp WHERE salary <10000  
2 UNION  
3 SELECT *from emp WHERE age>30;
```

UNIONON后面加上ALL则是将查询的结构简单拼接。

注意：对于联合查询的多张表列数一定要保持一致，字段类型也要保持一致。

子查询

- 概念：SQL语句中嵌套SELECT语句，称为**嵌套查询**，又称**子查询**。

```
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，分为：
 - 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

```
1 -- 查询销售部所有的员工信息  
2 SELECT * FROM emp WHERE dept_id=(SELECT id from dept WHERE name='研发部');  
3  
4 -- 查询杨道入职之后的员工信息  
5 SELECT *from emp WHERE entrydate>(SELECT entrydate FROM emp WHERE name='杨道');
```

列子查询

- 列子查询

子查询返回的结果是一列（可以是多行），这种子查询称为**列子查询**。

常用的操作符：IN、NOT IN、ANY、SOME、ALL

操作符	描述
IN	在指定的集合范围之内，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值都必须满足

行子查询

- 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**。

常用的操作符：=、<>、IN、NOT IN

▼ 查询与张无忌薪资与直属领导相同的信息

SQL |

```
1 SELECT *FROM emp WHERE salary=(SELECT salary FROM emp WHERE name='张无忌') AND managerid=(SELECT managerid FROM emp WHERE name='张无忌');
```

表子查询

子查询

- 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