



✦ ✦ 作者主页: [欽某](#) ✦ ✦

✦ ✦ 所属专栏: [MySql](#) ✦ ✦

准备工作, 创建一个雇员信息表 (来自oracle 9i的经典测试表)

- EMP员工表
- DEPT部门表
- SALGRADE工资等级表

mysql> select \* from dept;

deptno	dname	loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

CSDN @ 薇某

mysql> select \* from salgrade;

grade	losal	hisal
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

CSDN @ 薇某

mysql> select \* from emp;

empno	ename	job	mgr	hiredate	sal	comm	deptno
007369	SMITH	CLERK	7902	1980-12-17 00:00:00	800.00	NULL	20
007499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600.00	300.00	30
007521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250.00	500.00	30
007566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975.00	NULL	20
007654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250.00	1400.00	30
007698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850.00	NULL	30
007782	CLARK	MANAGER	7839	1981-06-09 00:00:00	2450.00	NULL	10
007788	SCOTT	ANALYST	7566	1987-04-19 00:00:00	3000.00	NULL	20
007839	KING	PRESIDENT	NULL	1981-11-17 00:00:00	5000.00	NULL	10
007844	TURNER	SALESMAN	7698	1981-09-08 00:00:00	1500.00	0.00	30
007876	ADAMS	CLERK	7788	1987-05-23 00:00:00	1100.00	NULL	20
007900	JAMES	CLERK	7698	1981-12-03 00:00:00	950.00	NULL	30
007902	FORD	ANALYST	7566	1981-12-03 00:00:00	3000.00	NULL	20
007934	MILLER	CLERK	7782	1982-01-23 00:00:00	1300.00	NULL	10

CSDN @ 薇某

## 多表查询

- 显示雇员名，雇员工资以及所在部门的名字

因为上面的数据来自EMP和DEPT，所以要联合查询

```
mysql> select ename, sal, emp.deptno from emp, dept
> where emp.deptno=dept.deptno;
```

ename	sal	deptno
-------	-----	--------

SMITH	800.00	20
ALLEN	1600.00	30
WARD	1250.00	30
JONES	2975.00	20
MARTIN	1250.00	30
BLAKE	2850.00	30
CLARK	2450.00	10
SCOTT	3000.00	20
KING	5000.00	10
TURNER	1500.00	30
ADAMS	1100.00	20
JAMES	950.00	30
FORD	3000.00	20
MILLER	1300.00	10

-----+  
14 rows in set (0.01 sec)

- 显示部门号为10的部门，员工名和工资

```
mysql> select dname,ename,sal from emp,dept
> where emp.deptno=dept.deptno and emp.deptno=10;
```

dname	ename	sal
ACCOUNTING	CLARK	2450.00
ACCOUNTING	KING	5000.00
ACCOUNTING	MILLER	1300.00

-----+  
3 rows in set (0.00 sec)

- 显示各个员工的姓名、工资、工资级别

```
mysql> select ename, sal, grade from emp, salgrade
> where emp.sal between losal and hisal;
```

ename	sal	grade
SMITH	800.00	1
ALLEN	1600.00	3
WARD	1250.00	2

```

| JONES | 2975.00 | 4 |
| MARTIN | 1250.00 | 2 |
| BLAKE | 2850.00 | 4 |
| CLARK | 2450.00 | 4 |
| SCOTT | 3000.00 | 4 |
| KING | 5000.00 | 5 |
| TURNER | 1500.00 | 3 |
| ADAMS | 1100.00 | 1 |
| JAMES | 950.00 | 1 |
| FORD | 3000.00 | 4 |
| MILLER | 1300.00 | 2 |
+-----+-----+-----+
14 rows in set (0.06 sec)

```

## 自连接

自连接是指在同一张表中连接查询

- 显示员工FORD的上级领导的编号和姓名（mgr是员工领导的编号-empno）

使用子查询：

```

mysql> select ename,empno from emp
       > where (select mgr from emp where ename='FORD')=empno;
+-----+-----+
| ename | empno |
+-----+-----+
| JONES | 007566 |
+-----+-----+
1 row in set (0.00 sec)

```

使用多表查询（自查询）：

```

mysql> select leader.ename, leader.empno from emp leader,emp worker
       > where leader.empno=worker.mgr and worker.ename='FORD';

```

```
+-----+-----+
|  ename  | empno  |
+-----+-----+
|  JONES  | 007566 |
+-----+-----+
1 row in set (0.02 sec)
```

子查询

子查询是指嵌入在其他sql语句中的select语句，也叫嵌套语句

单行子查询

只返回一行记录的子查询

- 显示SMITH同一部门的员工

```
mysql> select * from emp where (select deptno from emp where
ename='SMITH')=deptno;
+-----+-----+-----+-----+-----+-----+-----+-----+
+
| empno  | ename  | job      | mgr   | hiredate           | sal      | comm  | deptno |
+-----+-----+-----+-----+-----+-----+-----+-----+
+
| 007369 | SMITH  | CLERK    | 7902  | 1980-12-17 00:00:00 | 800.00   | NULL  | 20     |
|
| 007566 | JONES  | MANAGER  | 7839  | 1981-04-02 00:00:00 | 2975.00  | NULL  | 20     |
|
| 007788 | SCOTT  | ANALYST  | 7566  | 1987-04-19 00:00:00 | 3000.00  | NULL  | 20     |
|
| 007876 | ADAMS  | CLERK    | 7788  | 1987-05-23 00:00:00 | 1100.00  | NULL  | 20     |
|
| 007902 | FORD   | ANALYST  | 7566  | 1981-12-03 00:00:00 | 3000.00  | NULL  | 20     |
|
+-----+-----+-----+-----+-----+-----+-----+-----+
+
5 rows in set (0.00 sec)
```

## 多行子查询

### 返回多行记录的子查询

- in关键字：查询和10号部门的工作岗位相同的雇员的名字、岗位、工资、部门号但是不包括10自己的

```
mysql> select ename, job, deptno from emp
       > where job in (select job from emp where deptno=10)
       > and deptno<>10;
```

ename	job	deptno
SMITH	CLERK	20
JONES	MANAGER	20
BLAKE	MANAGER	30
ADAMS	CLERK	20
JAMES	CLERK	30

5 rows in set (0.00 sec)

- all关键字：显示工资比部门30的所有员工的工资高的员工的姓名、工资、部门号

```
mysql> select ename, sal, deptno from emp
       > where sal > all (select sal from emp where deptno=30);
```

ename	sal	deptno
JONES	2975.00	20
SCOTT	3000.00	20
KING	5000.00	10
FORD	3000.00	20

4 rows in set (0.01 sec)

- any关键字：显示工资比部门30的任意员工的工资高的员工的姓名、工资、部门号（包含自己部门的员工）

```
mysql> select ename, sal, deptno from emp
      > where sal > any(select sal from emp where deptno=30);
```

ename	sal	deptno
ALLEN	1600.00	30
WARD	1250.00	30
JONES	2975.00	20
MARTIN	1250.00	30
BLAKE	2850.00	30
CLARK	2450.00	10
SCOTT	3000.00	20
KING	5000.00	10
TURNER	1500.00	30
ADAMS	1100.00	20
FORD	3000.00	20
MILLER	1300.00	10

12 rows in set (0.01 sec)

多列子查询

单行子查询是指子查询只返回单列，单行数据；多行子查询是指返回单列多行数据，都是针对单列而言的，而多列子查询则是指查询返回多个列数据的子查询语句

- 查询和SMITH的部门和岗位完全相同的所有雇员，不包含SMITH本人

```
mysql> select * from emp
      > where (deptno,job)=(select deptno,job from emp where ename='SMITH')
      > and ename<>'SMITH';
```

empno	ename	job	mgr	hiredate	sal	comm	deptno
007876	ADAMS	CLERK	7788	1987-05-23 00:00:00	1100.00	NULL	20

1 row in set (0.00 sec)

### 在from子句中使用子查询

子查询语句出现在from子句中。这里要用到数据查询的技巧，把一个子查询当做一个临时表使用。**在MySql中，一切皆为表。**

- 显示高于自己部门平均工资的员工的姓名、部门、工资、平均工资

```
mysql> select emp.ename, emp.deptno, emp.sal, format(avg_sal,2) from
> emp, (select avg(sal) avg_sal, deptno dpt from emp group by deptno) tmp
> where emp.sal>avg_sal and emp.deptno=tmp.dpt;
```

ename	deptno	sal	format(avg_sal,2)
FORD	20	3000.00	2,175.00
SCOTT	20	3000.00	2,175.00
JONES	20	2975.00	2,175.00
BLAKE	30	2850.00	1,566.67
ALLEN	30	1600.00	1,566.67
KING	10	5000.00	2,916.67

6 rows in set (0.00 sec)

- 查找每个部门工资最高的人的姓名、工资、部门、最高工资

```
mysql> select ename, sal, emp.deptno, max_sal from
> emp,(select deptno,max(sal) max_sal from emp group by deptno) tmp
> where emp.deptno=tmp.deptno and max_sal=sal;
```

ename	sal	deptno	max_sal
BLAKE	2850.00	30	2850.00
SCOTT	3000.00	20	3000.00
KING	5000.00	10	5000.00
FORD	3000.00	20	3000.00

4 rows in set (0.00 sec)



- 显示每个部门的信息（部门名、编号、地址）和人员数量

使用子查询

```
mysql> select dept.deptno, dname, cnt, loc from
> dept, (select count(*) cnt,deptno from emp group by deptno) tmp
> where dept.deptno=tmp.deptno;

+-----+-----+-----+-----+
| deptno | dname      | cnt | loc      |
+-----+-----+-----+-----+
|      10 | ACCOUNTING |   3 | NEW YORK |
|      20 | RESEARCH   |   5 | DALLAS   |
|      30 | SALES      |   6 | CHICAGO  |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

使用多表（复杂、不建议）

```
mysql> select dept.deptno, dept.dname, dept.loc, count(*)
> from emp,dept where dept.deptno=emp.deptno
> group by dept.deptno,dept.dname,dept.loc;

+-----+-----+-----+-----+
| deptno | dname      | loc      | count(*) |
+-----+-----+-----+-----+
|      20 | RESEARCH   | DALLAS   |         5 |
|      30 | SALES      | CHICAGO  |         6 |
|      10 | ACCOUNTING | NEW YORK |         3 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

合并查询

在实际应用中，为了合并多个

select  
的执行结果，可以使用集合操作符  
union  
,  
union all

union

该操作符用于取得两个结果集的并集。当使用该操作符时，会自动去掉结果集中的重复行。

- 将工资大于2500或职位是MANAGER的人找出来

```
mysql> select ename, sal, job from EMP where sal>2500 union
      > select ename, sal, job from EMP where job='MANAGER';-- 去掉了重复记录
```

ename	sal	job
JONES	2975.00	MANAGER
BLAKE	2850.00	MANAGER
SCOTT	3000.00	ANALYST
KING	5000.00	PRESIDENT
FORD	3000.00	ANALYST
CLARK	2450.00	MANAGER

union all

该操作符用于取得两个结果集的并集。当使用该操作符时，不会去掉结果集中的重复行。

- 将工资大于25000或职位是MANAGER的人找出来

```
mysql> select ename, sal, job from EMP where sal>2500 union all
      > select ename, sal, job from EMP where job='MANAGER';
```

ename	sal	job
-------	-----	-----

ename	sal	job
JONES	2975.00	MANAGER
BLAKE	2850.00	MANAGER
SCOTT	3000.00	ANALYST
KING	5000.00	PRESIDENT
FORD	3000.00	ANALYST
JONES	2975.00	MANAGER
BLAKE	2850.00	MANAGER
CLARK	2450.00	MANAGER

## 内连接

内连接实际上就是利用where子句对两种表形成的笛卡尔积进行筛选，是再开发过程中使用得最多的连接查询

语法：

```
select 字段 from 表一 inner join 表二 on 连接条件 and 其他条件；
```

## 外连接

外连接分为左外连接和右外连接

### 左外连接

联合查询时，如果在左侧的表需要完全显示，我们就用左外连接

语法：

```
select 字段 from 表一 left join 表二 on 连接条件
```

## 右外连接

联合查询时，如果右侧的表需要完全显示，我们就右外连接

```
select 字段 from 表一 right join 表二 on 连接条件
```

**本期博客到这里就结束了，如果有什么错误，欢迎指出，如果对你有帮助，请点个赞，谢谢！**