

## 1、内连接之等值连接（最大特点是：连接条件是等量关系）

--找出每一个员工的部门名称，要求显示员工名和部门名

sql92: --已弃用

```
select
    e.ename,d.dname
from
    emp e,dept d
where
    e.deptno=d.deptno;
```

sql99: --常用

```
select
    e.ename,d.dname
from
    emp e
join                --此处本该是inner join，省略了inner，带着更好
    dept d
on
    e.deptno=d.deptno;
```

--sql99的语法

...

A

join

B

on

表的连接条件

where

记录数据的过滤条件

--由此可见sql99语法的结构更清晰些：因为表的连接条件关键字和记录过滤的条件关键字where分开了

## 2、内连接之非等值连接（最大特点是：连接条件是非等量关系）

--找出每个员工的工资等级，要求显示员工名、工资、工资等级

--从emp表中找出所有员工ename及其工资sal

mysql> select e.ename,e.sal from emp e;

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

```
14 rows in set (0.00 sec)
```

--从salgrade表中查询所有记录

```
mysql> select * from salgrade;
```

```
+-----+-----+-----+
| GRADE | LOSAL | HISAL |
+-----+-----+-----+
|      1 |    700 |   1200 |
|      2 |   1201 |   1400 |
|      3 |   1401 |   2000 |
|      4 |   2001 |   3000 |
|      5 |   3001 |   9999 |
+-----+-----+-----+
```

```
5 rows in set (0.12 sec)
```

--可以看到工资等级表中有五条记录，连接上面两表有14\*5=70条匹配数据，过滤无效记录通过sal字段和LOSAL及HISAL字段

如下：

```
mysql> select
-> e.ename,e.sal,s.grade
-> from
-> emp e
-> inner join
-> salgrade s
-> on
-> e.sal between s.losal and s.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 (0.01 sec)
```

### 3、自连接（最大特点：一张表看作两张表，自己连接自己）

--找出每个员工的上级领导，要求显示员工名和上级领导名

--先查询emp员工表的所有数据

```
mysql> select * from emp;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL      | COMM  | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+
| 7369  | SMITH  | CLERK    | 7902 | 1980-12-17 | 800.00   | NULL  | 20     |
| 7499  | ALLEN  | SALESMAN | 7698 | 1981-02-20 | 1600.00  | 300.00 | 30     |
| 7521  | WARD   | SALESMAN | 7698 | 1981-02-22 | 1250.00  | 500.00 | 30     |
| 7566  | JONES  | MANAGER  | 7839 | 1981-04-02 | 2975.00  | NULL  | 20     |
| 7654  | MARTIN | SALESMAN | 7698 | 1981-09-28 | 1250.00  | 1400.00 | 30     |
+-----+-----+-----+-----+-----+-----+-----+
```

7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
7788	SCOTT	ANALYST	7566	1987-04-19	3000.00	NULL	20
7839	KING	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1987-05-23	1100.00	NULL	20
7900	JAMES	CLERK	7698	1981-12-03	950.00	NULL	30
7902	FORD	ANALYST	7566	1981-12-03	3000.00	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10

+-----+-----+-----+-----+-----+-----+-----+-----+

--可以看到每个员工对应的上级是MGR字段,利用该表定义两个别名以使用MGR字段和EMPNO字段进行连接查询,就是员工的领导编号MGR与领导的编号EMPNO相同

```
select
    e1.ename as '员工',e2.ename as '领导'
from
    emp e1
inner join
    emp e2
on
```

ifnull(e1.mgr,7839)=e2.empno;--此处为e1.mgr=null的话将其赋为7893,如果不加ifnull的话由于KING中的MGR字段值是null,无法与领导表中的领导编号EMPNO匹配,就不会查询出员工KING的数据,因为连接的两张表是平等的,不一定要查出员工表(虽然这里两张表都是同一张表,但是为了找出连接条件要自己区分这两张表才能更好的找到关系)中的所有员工数据,除非员工表是主表,另一张是副表,就必须把主表中的所有数据查询到,不能因为副表没有可匹配的数据就忽略主表中的某个数据,但这是外连接。

+-----+-----+

员工	领导
----	----

+-----+-----+

SMITH	FORD
ALLEN	BLAKE
WARD	BLAKE
JONES	KING
MARTIN	BLAKE
BLAKE	KING
CLARK	KING
SCOTT	JONES
KING	KING
TURNER	BLAKE
ADAMS	SCOTT
JAMES	BLAKE
FORD	JONES
MILLER	CLARK

+-----+-----+

14 rows in set (0.00 sec)

## 4、外连接

外连接与内连接的区别

内连接：

假设A和B表进行连接,使用内连接的话,凡是A表和B表能匹配上的记录都查询出来,无法匹配的自动过滤掉。两张表没有主副之分,这两张表是平等的

外连接：

AB表使用外连接进行连接查询,这两张表有一张必须是主表,一张是副表;主要查询的是主表中的内容,捎带 查询副表中的数据,当副表中的数据没有和主表中的数据匹配上,副表自动模拟出null与之匹配。

外连接的分类：

左外连接（左连接）：表示左边的表是主表。

右外连接（右连接）：表示右边的表是主表。

其中左连接有右连接的写法，右连接也有左连接的写法

--找出每个员工的上级领导

```
mysql> select e.ename,e.empno,e.mgr from emp e;--此时将emp表看作员工表e
```

ename	empno	mgr
SMITH	7369	7902
ALLEN	7499	7698
WARD	7521	7698
JONES	7566	7839
MARTIN	7654	7698
BLAKE	7698	7839
CLARK	7782	7839
SCOTT	7788	7566
KING	7839	NULL
TURNER	7844	7698
ADAMS	7876	7788
JAMES	7900	7698
FORD	7902	7566
MILLER	7934	7782

14 rows in set (1.60 sec)

--根据MGR中的编号筛选出是领导的员工编号并作为领导表（这是没法实现的，但是是自连接为了找出连接条件用该办法）

```
mysql> select ename,empno from emp b;
```

ename	empno
JONES	7566
BLAKE	7698
CLARK	7782
SCOTT	7788
KING	7839
FORD	7902

14 rows in set (0.00 sec)

--筛选后的emp表作为领导表b

--这样我们就可以找出连接条件：员工表e中的e.MGR与领导表b中的b.empno相匹配就是一条可以输出记录，然而KING中的MGR是null，领导表b中没有empno数据与之匹配，如果使用内连接，KING员工这条数据就不会被查询出，但KING是老板，是不能缺少的，所以需要使用外连接将员工表e作为主表，主要查询主表中的数据，且一定要查询出来，外连接查询语句如下

```
mysql> select
```

```
-> e1.ename as '员工',e2.ename as '领导'
```

```
-> from
```

```
-> emp e1
```

```
-> left outer join--左连接，左边的是主表，且outer可以省略，表示外连接
```

```
-> emp e2
```

```
-> on
```

```
-> e1.mgr=e2.empno;
```

员工	领导
----	----

```
+-----+-----+
| SMITH | FORD |
| ALLEN | BLAKE |
| WARD | BLAKE |
| JONES | KING |
| MARTIN | BLAKE |
| BLAKE | KING |
| CLARK | KING |
| SCOTT | JONES |
| KING | NULL |
| TURNER | BLAKE |
| ADAMS | SCOTT |
| JAMES | BLAKE |
| FORD | JONES |
| MILLER | CLARK |
+-----+-----+
```

14 rows in set (1.37 sec)

```
mysql> select
-> e1.ename as '员工',e2.ename as '领导'
-> from
-> emp e2
-> right join--右连接，右边的是主表
-> emp e1
-> on
-> e1.mgr=e2.empno;
```

--题目，找出哪个部门没有员工

--部门表是主表，员工表是副表

```
select
  d.*
  from
  dept d
  left outer join
  emp e
  on
  d.deptno=e.deptno
  where
  e.epmno is null;
```

```
mysql> select
-> d.*,e.*
-> from
-> dept d
-> left outer join
-> emp e
-> on
-> d.deptno=e.deptno;
```

```
+-----+-----+-----+
| dname      | deptno | ename |
+-----+-----+-----+
| RESEARCH   | 20     | SMITH |
| SALES       | 30     | ALLEN |
| SALES       | 30     | WARD  |
| RESEARCH   | 20     | JONES |
| SALES       | 30     | MARTIN |
| SALES       | 30     | BLAKE |
| ACCOUNTING | 10     | CLARK |
| RESEARCH   | 20     | SCOTT |
| ACCOUNTING | 10     | KING  |
```

SALES		30	TURNER	
RESEARCH		20	ADAMS	
SALES		30	JAMES	
RESEARCH		20	FORD	
ACCOUNTING		10	MILLER	
OPERATIONS		40	NULL	

+-----+-----+-----+

15 rows in set (0.00 sec)

--然后

select

-> d.\*

-> from

-> dept d

-> left outer join

-> emp e

-> on

-> d.deptno=e.deptno

->where

->e.empno is null;--或者e.ename is null

+-----+-----+-----+

DEPTNO		DNAME		LOC	
--------	--	-------	--	-----	--

+-----+-----+-----+

	40		OPERATIONS		BOSTON	
--	----	--	------------	--	--------	--

+-----+-----+-----+

1 row in set (0.00 sec)