

## 知识点列表

编号	名称	描述	级别
1	非关联子查询	学习非关联子查询的数据环境和语法	***
2	关联子查询	对比非关联子查询，掌握语法	**
3	集合操作	多个结果集之间的操作	*
4	表间关系	多表之间的关联关系	***
5	内连接和外连接	多表之间关联查询	***
6	左/右/全外连接	多表之间关联查询	***

注：    "\*\*\*"理解级别    "\*\*\*"掌握级别    "\*\*\*\*"应用级别

## 目录

1. 知识点小结 .....	4
1.1. 查询语句的基本格式 *** .....	4
1.2. 函数 .....	4
1.2.1. 组函数：count / avg / sum / max / min ** .....	4
1.2.2. 单行函数 .....	4
2. 子查询( 二 ) .....	8
2.1. 单行比较运算符 > < >= <= = <> .....	8
2.2. All * .....	11
2.3. Any * .....	12
2.4. In * .....	13
2.5. 单行比较运算符和 All、Any、in .....	16
2.6. 子查询的结果返回多列的情况 .....	16
2.7. 子查询出现在 having 短语中 .....	18
3. 关联子查询 ** .....	20
3.1. 关联子查询演示 .....	21
3.2. Exists 关键字 ** .....	22
4. 集合操作 * .....	26
4.1. 结果集操作 .....	26
4.1.1. 合集 .....	26
4.1.2. 交集 intersect .....	28
4.1.3. 差集 minus .....	28
5. 表间关联查询 ** .....	29

5.1. 表 emp_xxx 和表 dept_xxx 之间存在的参照关系 .....	29
5.2. 表 emp_ning 自身存在一种参照关系 .....	29
5.3. 主键( PK )和外键( FK ).....	30
5.4. 内连接 ** .....	30
5.4.1. 驱动表和匹配表.....	34
5.5. 外连接** .....	34
5.6. full outer join 全外连接 ** .....	40
6. 小结外连接.....	41

## 1. 知识点小结

### 1.1. 查询语句的基本格式 \*\*\*

```
SQL> select  字段 1, 字段 2, 字段 3, 表达式, 函数, ...
        from      表名
        where      条件
        group by   列名
        having     带组函数的条件
        order by   列名
```

### 1.2. 函数

#### 1.2.1. 组函数：count / avg / sum / max / min \*\*

记得：组函数忽略空值

#### 1.2.2. 单行函数

##### 1.2.2.1. 字符函数：upper / lower / initcap / length / lpad / rpad / replace / trim \* -- l 表示

left ; r 表示 right

- 1) **upper**            转换为大写
- 2) **lower**           转换为小写
- 3) **initcap**          转换为首字母大写
- 4) **length**          取长度
- 5) **lpad**            左补丁
- 6) **rpad**            右补丁
- 7) **replace**        字符替换
- 8) **trim**            去除前后的空格

**【案例 1】** 将 **ename** 字段设置为 10 个长度，如果不够左边用 “\*” 号补齐

```
SQL> select lpad( ename , 10 , '*' ) from emp_xxx ;
```

```
运行 SQL 命令行
SQL> select lpad(ename, 10, '*') from emp_xxx;

LPAD(ENAME,10,'*')
-----
*****amy
****张无忌
****刘苍松
*****李翊
****郭芙蓉
****张三丰
```

**【案例 2】** 将 ename 字段设置为 10 个长度，如果不够右边用 “#” 号补齐

```
SQL> select rpad(ename, 10, '#') from emp_xxx;
```

```
运行 SQL 命令行
SQL> select rpad(ename, 10, '#') from emp_xxx;

RPAD(ENAME,10,'#')
-----
amy#####
张无忌####
刘苍松####
李翊#####
郭芙蓉####
张三丰####
燕小六####
```

#### 1.2.2.2. 数字函数：round / trunc / mod \*\*

**【案例 3】** 求 salary 对 5000 取模

```
SQL> select salary, mod(salary, 5000) from emp_xxx;
```

```
运行 SQL 命令行
SQL> select salary , mod(salary, 5000) from emp_xxx;

SALARY MOD(SALARY, 5000)
-----
10000      0
 8000     3000
 9000     4000
 5000      0
15000      0
 5000      0
 3000     3000
 5000      0
 4000     4000
 4500     4500

已选择12行。
```

#### 1.2.2.3. 日期函数： months\_between / add\_months / last\_day \*\*

- 1) **months\_between**      两个日期之间的月份数
- 2) **add\_months**        给定一个日期，为该日期增加指定月份
- 3) **last\_day**          找出参数时间点所在月份的最后一天

#### 【案例 4】将 amy 的入职日期提前 2 个月

```
SQL> select ename , hiredate from emp_xxx
      where ename='amy';
SQL> update emp_xxx set hiredate=add_months(hiredate , -2)
      where ename='amy';
SQL> select ename , hiredate from emp_xxx
      where ename='amy';
```

```
运行 SQL 命令行
SQL> select ename , hiredate from emp_xxx
2  where ename='amy' ;

ENAME          HIREDATE
-----
amy            05-12月-11

SQL> update emp_xxx set hiredate=add_months(hiredate ,-2)
2  where ename='amy' ;

已更新 1 行。

SQL> select ename , hiredate from emp_xxx
2  where ename='amy' ;

ENAME          HIREDATE
-----
amy            05-10月-11
```

**【案例 5】这个月的最后一天是多少号？**

```
SQL> select sysdate , last_day(sysdate) from dual ;
```

```
运行 SQL 命令行
SQL> select sysdate , last_day(sysdate) from dual;

SYSDATE          LAST_DAY(SYSDA
-----
06-12月-11      31-12月-11
```

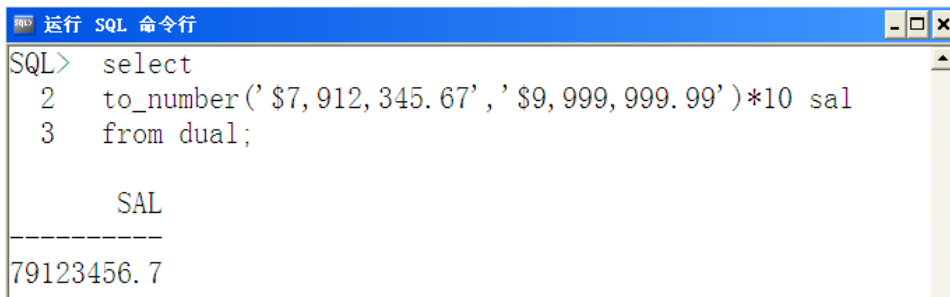
**1.2.2.4. 转换函数：to\_char / to\_date / to\_number \*\***

	<b>to_char</b>		<b>to_number</b>	
日期	----->	字符	----->	数字
	<-----		<-----	
	<b>to_date</b>		<b>to_char</b>	

● **to\_number 的用法**

**【案例 6】将 “\$7,912,345.67” 乘以 10，输出结果**

```
SQL> select to_number('$7,912,345.67', '$9,999,999.99') from dual;
```

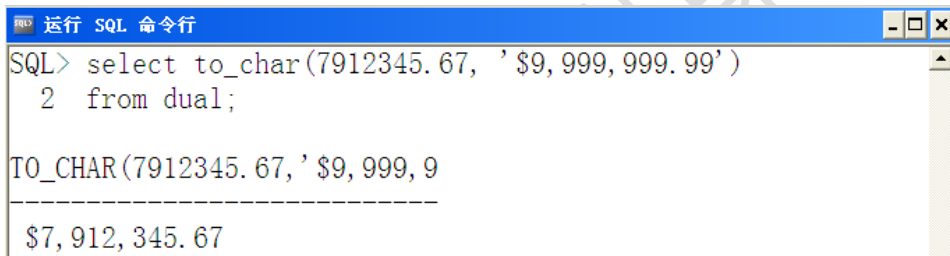


```
SQL> select
  2  to_number('$7,912,345.67', '$9,999,999.99')*10 sal
  3  from dual;

      SAL
-----
79123456.7
```

**【案例 7】将 7912345.67 按指定格式“ \$9,999,999.99 ”输出**

```
SQL> select to_char(7912345.67, '$9,999,999.99') from dual;
```



```
SQL> select to_char(7912345.67, '$9,999,999.99')
  2  from dual;

TO_CHAR(7912345.67, '$9,999,9
-----
$7,912,345.67
```

**1.2.2.5. 通用函数：nvl / coalesce / decode \***

单行函数除去四大类字符函数、数字函数、日期函数、转换函数，还有一些其它常用的函数。

✓ nvL / coalesce / decode

**1.2.2.6. 使用频率比较高的函数 \*\*\***

- 1) **单行函数**：upper / round / to\_char / to\_date / nvl
- 2) **组函数**：count / avg / sum / max / min

## 2. 子查询(二)

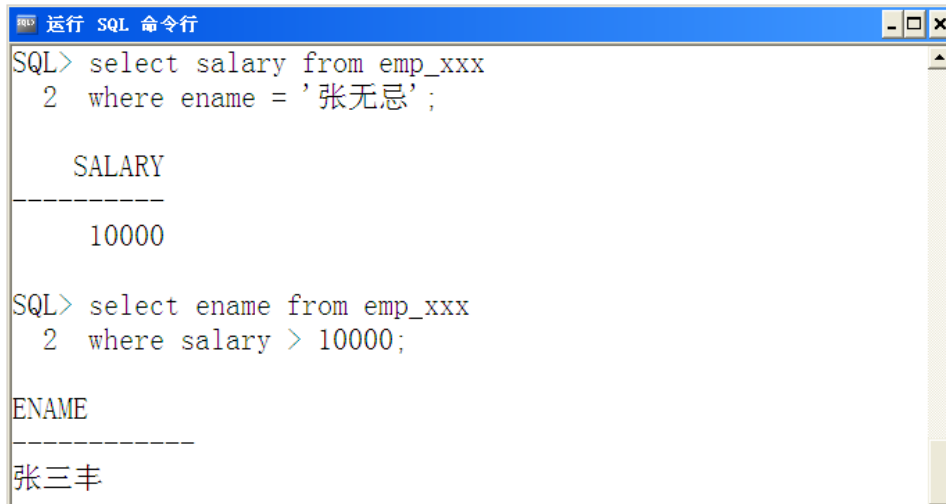
### 2.1. 单行比较运算符 > < >= <= = <>

**【例 8】谁的薪水比张无忌高？**



- **分步查询**

```
SQL> select salary from emp_xxx where ename = '张无忌';
SQL> select ename from emp_xxx where salary > 10000;
```



```
运行 SQL 命令行
SQL> select salary from emp_xxx
2  where ename = '张无忌';

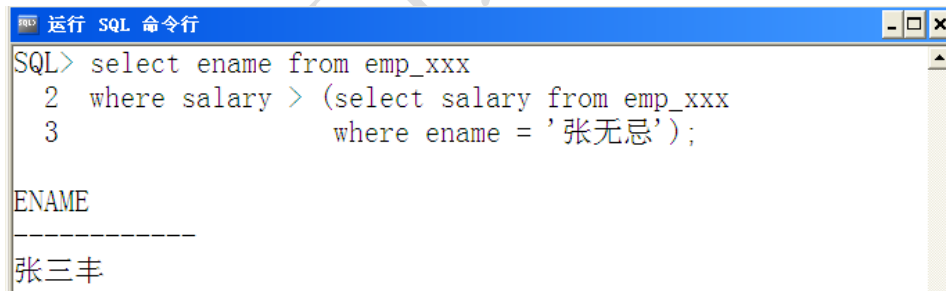
SALARY
-----
10000

SQL> select ename from emp_xxx
2  where salary > 10000;

ENAME
-----
张三丰
```

- **子查询(合并到一条 SQL 语句, 并不关心张无忌的薪水)**

```
SQL> select ename from emp_xxx
      where salary > (select salary from emp_xxx
                     where ename = '张无忌');
```



```
运行 SQL 命令行
SQL> select ename from emp_xxx
2  where salary > (select salary from emp_xxx
3                  where ename = '张无忌');

ENAME
-----
张三丰
```

**注意：**单行比较运算符都只能和一个值比较

**【案例 9】研发部有哪些职位？**

```
SQL> select distinct job from emp_xxx
      where deptno = (select deptno
                     from dept_xxx
                     where dname = 'developer');
```

```

运行 SQL 命令行
SQL> select distinct job from emp_xxx
2  where deptno = (select deptno
3                  from dept_xxx
4                  where dname = 'developer');

JOB
-----
Manager
Analyst
Programmer

```

【案例 10】谁的薪水比张无忌高？ --如果有多个叫张无忌的人则会出错

#### ● 准备数据

--准备数据：在 emp\_xxx 中再插入一个叫“张无忌”的人

```

SQL> insert into emp_xxx(empno, ename, salary)
      values(1014, '张无忌', 8000);
SQL> select ename, salary from emp_xxx;

```

```

运行 SQL 命令行
SQL> insert into emp_xxx(empno, ename, salary)
2  values(1014, '张无忌', 8000);

已创建 1 行。
SQL> select ename, salary from emp_xxx;

ENAME          SALARY
-----
amy
张无忌          10000
刘苍松          8000
李翊            9000
郭芙蓉          5000
张三丰          15000

```

燕小六	5000
陆无双	3000
黄蓉	5000
韦小宝	4000
郭靖	4500
余泽成	
张无忌	8000

已选择13行。

- **查询语句** --使用单行比较运算符会出错

```

运行 SQL 命令行
SQL> select ename from emp_xxx
  2  where salary > (select salary from emp_xxx
  3                    where ename = '张无忌');
where salary > (select salary from emp_xxx
                  *
第 2 行出现错误:
ORA-01427: 单行子查询返回多个行
  
```

## 2.2.All \*

**【案例 11】查询谁的薪水比所有叫张无忌的薪水都高？** --大于最大值

```

SQL> select ename , salary from emp_xxx where ename='张无忌' ;
SQL> select ename from emp_xxx
      where salary > ALL( select salary from emp_xxx
                          where ename = '张无忌' );
  
```

```

运行 SQL 命令行
SQL> select ename , salary from emp_xxx
2  where ename='张无忌';

ENAME          SALARY
-----
张无忌          10000
张无忌           8000

SQL> select ename from emp_xxx
2  where salary > ALL(select salary from emp_xxx
3                      where ename = '张无忌');

ENAME
-----
张三丰
  
```

## 2.3. Any \*

**【案例 12】** 哪些人的薪水比任何一个叫张无忌的员工工资高？ --大于最小值

```

SQL> select ename from emp_xxx
      where salary > ANY(select salary from emp_xxx
                        where ename = '张无忌');

--只要大于叫“张无忌”的人的薪水最小值就查出来
  
```

```

运行 SQL 命令行
SQL> select ename , salary from emp_xxx;

ENAME          SALARY
-----
amy
张无忌          10000
刘苍松           8000
李翊            9000
郭芙蓉           5000
张三丰          15000
燕小六           5000
  
```

陆无双	3000
黄蓉	5000
韦小宝	4000
郭靖	4500
余泽成	
张无忌	8000

已选择13行。

```
SQL> select ename from emp_xxx
2  where salary > ANY(select salary from emp_xxx
3                      where ename = '张无忌');
```

ENAME

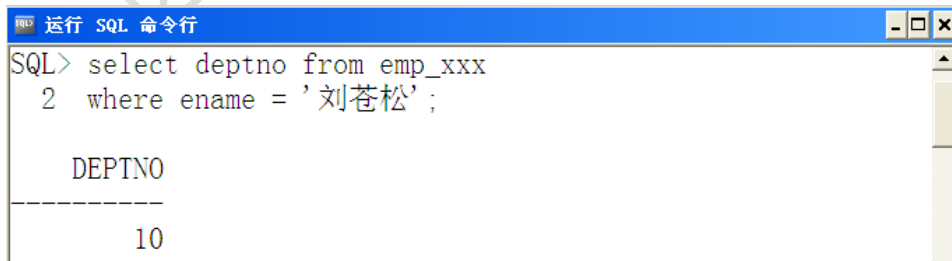
张无忌  
李翊  
张三丰

## 2.4.In \*

**【案例 13】谁和刘苍松同部门？列出除了刘苍松之外的员工名字**

- 分步查询，如果只有一个员工叫刘苍松 则无问题。

```
SQL> select deptno from emp_xxx
      where ename = '刘苍松';
SQL> select ename from emp_xxx
      where deptno = 10 and ename <> '刘苍松';
```



```
运行 SQL 命令行
SQL> select deptno from emp_xxx
2  where ename = '刘苍松';

DEPTNO
-----
10
```

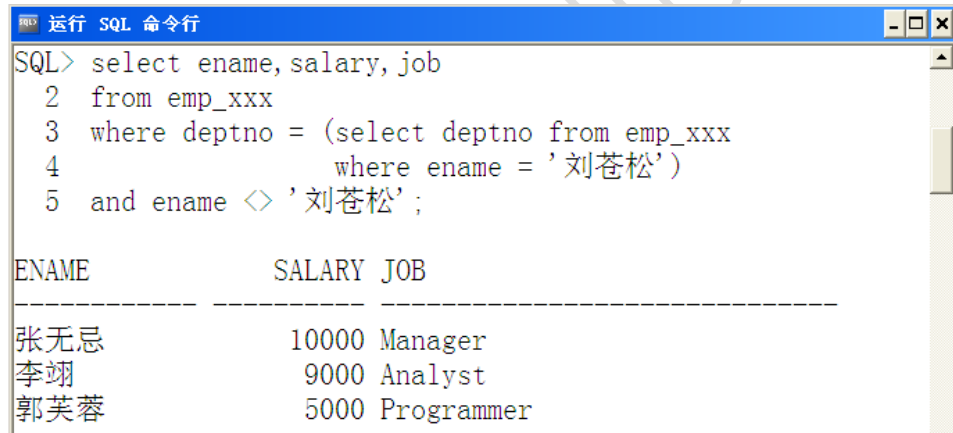
```
SQL> select ename from emp_xxx
2  where deptno = 10
3  and ename <> '刘苍松';
```

ENAME

-----  
张无忌  
李翊  
郭芙蓉

● 用子查询实现同一功能

```
SQL> select ename,salary,job
      from emp_xxx
      where deptno = (select deptno from emp_xxx
                      where ename = '刘苍松')
      and ename <> '刘苍松';
```



```
SQL> select ename, salary, job
2  from emp_xxx
3  where deptno = (select deptno from emp_xxx
4                  where ename = '刘苍松')
5  and ename <> '刘苍松';
```

ENAME	SALARY	JOB
张无忌	10000	Manager
李翊	9000	Analyst
郭芙蓉	5000	Programmer

**【案例 14】** 谁和刘苍松同部门？列出除了刘苍松之外的员工名字（如果子查询得到的结果是多个）

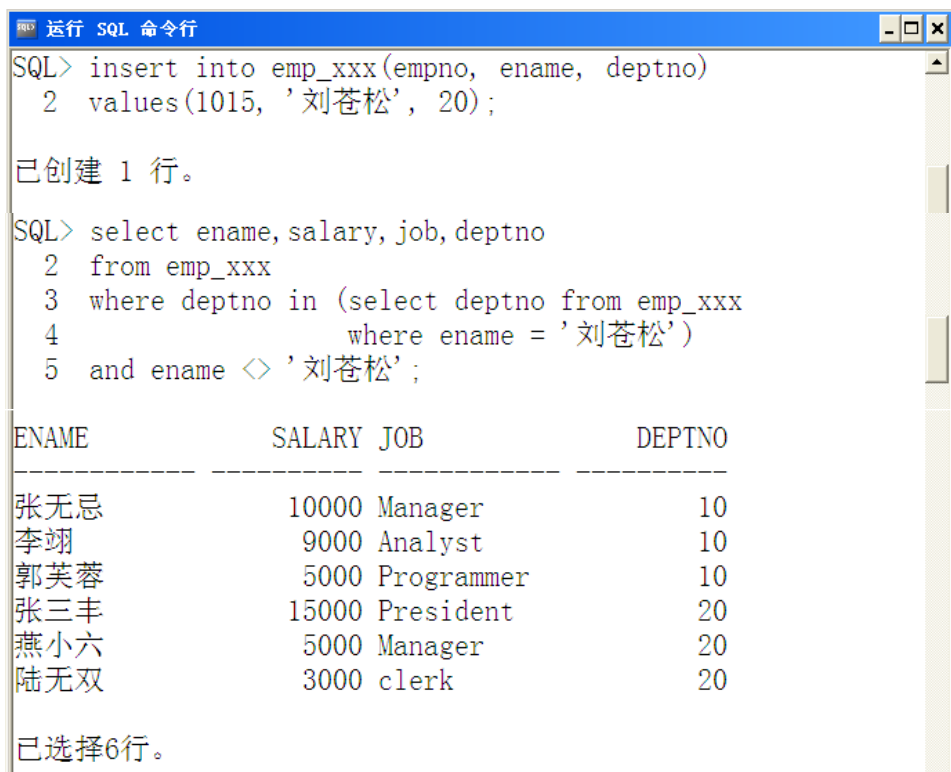
-- 数据准备：再添加 1 个“刘苍松”同学，部门号为 20

```
SQL> insert into emp_xxx(empno, ename, deptno)
      values(1015, '刘苍松', 20);
```

-- 如果子查询得到的结果是多个，不能使用单行比较运算符“=”，改为 in

```
SQL> select ename,salary,job,deptno
      from emp_xxx
      where deptno in (select deptno from emp_xxx
```

```
where ename = '刘苍松')
and ename <> '刘苍松';
```



```
运行 SQL 命令行
SQL> insert into emp_xxx(empno, ename, deptno)
2 values(1015, '刘苍松', 20);

已创建 1 行。

SQL> select ename, salary, job, deptno
2 from emp_xxx
3 where deptno in (select deptno from emp_xxx
4                  where ename = '刘苍松')
5 and ename <> '刘苍松';
```

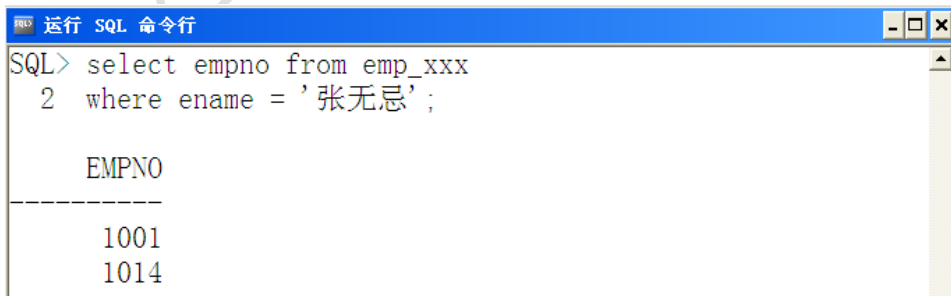
ENAME	SALARY	JOB	DEPTNO
张无忌	10000	Manager	10
李翊	9000	Analyst	10
郭芙蓉	5000	Programmer	10
张三丰	15000	President	20
燕小六	5000	Manager	20
陆无双	3000	clerk	20

已选择6行。

**【案例 15】谁是张无忌的下属？** 如果只有一个叫张无忌的员工，则无问题，如果有多个，需要用 in

- **步骤 1：查询张无忌下属的员工编号？**

```
SQL> select empno from emp_xxx where ename = '张无忌';
```



```
运行 SQL 命令行
SQL> select empno from emp_xxx
2 where ename = '张无忌';
```

EMPNO
1001
1014

- **步骤 2：查询员工编号是 1001 和 1004 的员工姓名**

```
SQL> select ename from emp_xxx
```

where mgr in (1001, 1014); -- mgr 代表该员工上司的 empno

```
运行 SQL 命令行
SQL> select ename from emp_xxx
2  where mgr in (1001, 1014);

ENAME
-----
刘苍松
李翊
郭芙蓉
```

● 步骤 3：谁是张无忌的下属？

```
SQL> select ename from emp_xxx
      where mgr in (select empno from emp_xxx
                    where ename = '张无忌');
```

```
运行 SQL 命令行
SQL> select ename from emp_xxx
2  where mgr in (select empno from emp_xxx
3                where ename = '张无忌');

ENAME
-----
刘苍松
李翊
郭芙蓉
```

## 2.5. 单行比较运算符和 All、Any、in

根据子查询返回的结果的行数选择使用：

- ✓ 返回一行      > < >= <= = <>
- ✓ 返回多行      >ALL >ANY <ALL <ANY in

## 2.6. 子查询的结果返回多列的情况

### 【案例 16】每个部门拿最高薪水的是谁？

● 步骤 1：每个部门的最高薪水是多少？



```
SQL> select deptno , max(salary)
      from emp_xxx
      where deptno is not null
      group by deptno ;
```

运行 SQL 命令行

```
SQL> select deptno, max(salary)
2  from emp_xxx
3  where deptno is not null
4  group by deptno;
```

DEPTNO	MAX(SALARY)
30	5000
20	15000
10	10000

● 步骤 2：每个部门拿最高薪水的是谁？

```
SQL> select ename, salary, job, deptno
      from emp_xxx
      where (deptno, salary) in ( select deptno, max(salary)
                                from emp_xxx
                                where deptno is not null
                                group by deptno ) ;
```

运行 SQL 命令行

```
SQL> select ename, salary, job, deptno
2  from emp_xxx
3  where (deptno, salary)
4      in (select deptno, max(salary)
5          from emp_xxx
6          where deptno is not null
7          group by deptno);
```

ENAME	SALARY	JOB	DEPTNO
张无忌	10000	Manager	10
张三丰	15000	President	20
黄蓉	5000	Manager	30

**注意：**

- ✓ 子查询的条件是单列还是多列没关系，关键是要分清返回的是单行还是多行。

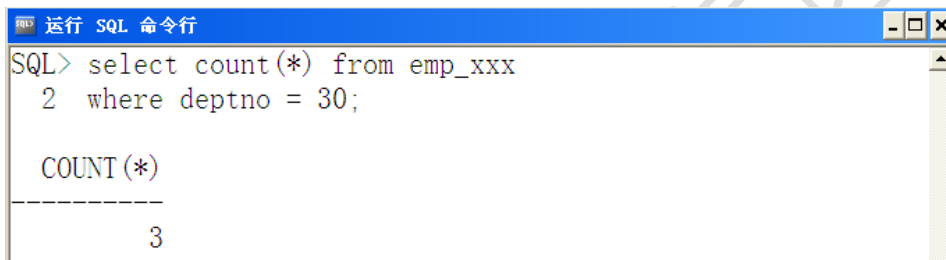
- ✓ 如果是单行，用单行比较运算符，=, >, < 这些
- ✓ 如果是多行，用 in, >all, >any, <all, <any 这些

## 2.7. 子查询出现在 having 短语中

### 【案例 17】哪个部门的人数比部门 30 的人数多？

- 步骤 1：查询部门 30 的人数是多少？

```
SQL> select count(*) from emp_xxx
      where deptno = 30 ;
```

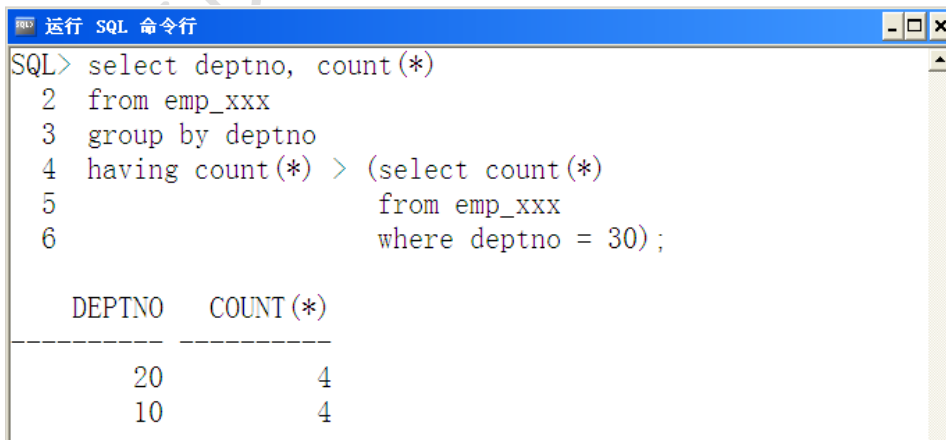


```
SQL> select count(*) from emp_xxx
2  where deptno = 30;

COUNT(*)
-----
3
```

- 步骤 2：哪个部门的人数比部门 30 的人数多？

```
SQL> select deptno, count(*)
      from emp_xxx
      group by deptno
      having count(*) > ( select count(*) from emp_xxx
                          where deptno = 30 );
```



```
SQL> select deptno, count(*)
2  from emp_xxx
3  group by deptno
4  having count(*) > (select count(*)
5                      from emp_xxx
6                      where deptno = 30);

DEPTNO    COUNT(*)
-----
20         4
10         4
```

**【案例 18】哪个部门的平均薪水比部门 20 的平均薪水高？**

```
SQL> select deptno , avg( nvl(salary, 0) ) avg_s
      from emp_xxx
      group by deptno
      having avg(nvl(salary,0)) > ( select avg(nvl(salary,0))
                                   from emp_xxx
                                   where deptno = 20 );
```

```
SQL> select deptno, avg(nvl(salary,0)) avg_s
2  from emp_xxx
3  group by deptno
4  having avg(nvl(salary,0)) >
5      (select avg(nvl(salary,0))
6      from emp_xxx
7      where deptno = 20);
```

DEPTNO	AVG_S
10	8000

**【案例 19】列出员工名字和职位，这些员工所在的部门平均薪水大于 5000 元**

● **步骤 1：查询平均工资大于 5000 元的部门号**

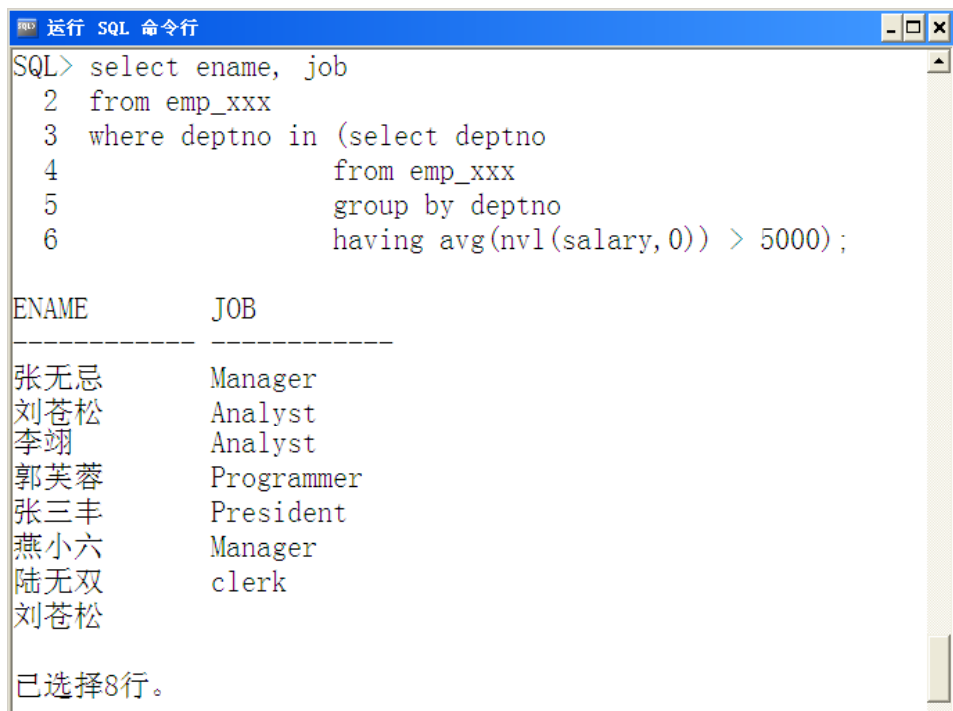
```
SQL> select deptno , avg( nvl(salary,0) )
      from emp_xxx
      group by deptno
      having avg( nvl(salary,0) ) > 5000;
```

```
SQL> select deptno, avg(nvl(salary,0))
2  from emp_xxx
3  group by deptno
4  having avg(nvl(salary,0)) > 5000;
```

DEPTNO	AVG(NVL(SALARY, 0))
20	5750
10	8000

● **步骤 2：列出员工名字和职位，这些员工所在的部门平均薪水大于 5000 元**

```
SQL> select ename, job
      from emp_xxx
      where deptno in (select deptno
                       from emp_xxx
                       group by deptno
                       having avg( nvl(salary,0)) > 5000 );
```



The screenshot shows a window titled "运行 SQL 命令行" (Run SQL Command Line). It contains the same SQL query as above. Below the query, the results are displayed in a table with two columns: "ENAME" and "JOB". The results are as follows:

ENAME	JOB
张无忌	Manager
刘苍松	Analyst
李翊	Analyst
郭芙蓉	Programmer
张三丰	President
燕小六	Manager
陆无双	clerk
刘苍松	

Below the table, it says "已选择8行。" (8 rows selected).

### 3. 关联子查询 \*\*

**【案例 20】哪些员工的薪水比公司的平均薪水低？**

```
SQL> select ename, salary
      from emp_xxx
      where salary < ( select avg(nvl(salary,0))
                       from emp_xxx );
SQL> select avg(nvl(salary,0)) from emp_xxx ;
```

```

运行 SQL 命令行
SQL> select ename, salary
  2   from emp_xxx
  3  where salary < (select avg(nvl(salary,0))
  4                    from emp_xxx);

ENAME          SALARY
-----
郭芙蓉          5000
燕小六          5000
陆无双          3000
黄蓉            5000
韦小宝          4000
郭靖            4500

已选择6行。

SQL> select avg(nvl(salary,0)) from emp_xxx;

AVG(NVL(SALARY,0))
-----
5464.28571
  
```

### 3.1. 关联子查询演示

子查询中不再是独立的 Sql 语句，需要依赖主查询传来的参数，这种方式叫关联子查询

**【案例 21】哪些员工的薪水比本部门的平均薪水低？不再和整个部门的平均薪水比较。**

```

SQL> select ename, salary, deptno
      from emp_xxx a
      where salary < ( select avg(nvl(salary,0))
                      from emp_xxx
                      where deptno = a.deptno );

--子查询不再是独立的 Sql 语句，需要依赖主查询传来的参数 a.deptno
  
```

```

运行 SQL 命令行
SQL> select ename, salary, deptno
2  from emp_xxx a
3  where salary < (select avg(nvl(salary,0))
4                  from emp_xxx
5                  where deptno = a.deptno);

```

ENAME	SALARY	DEPTNO
郭芙蓉	5000	10
燕小六	5000	20
陆无双	3000	20
韦小宝	4000	30

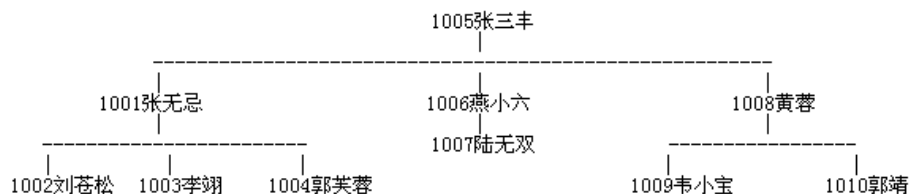
### 3.2. Exists 关键字 \*\*

【案例 22】哪些人是其他人的经理？（查找有下属的员工）

emp\_xxx 表

编码	姓名	职位	薪水	奖金	入职时间	经理	所在部
1001	张无忌	Manager	10000	2000	12-MAR-10	1005	10
1002	刘苍松	Analyst	8000	1000	01-APR-11	1001	10
1003	李翊	Analyst	9000	1000	11-APR-10	1001	10
1004	郭芙蓉	Programmer	5000	null	01-JAN-11	1001	10
1005	张三丰	President	15000	null	15-MAY-08	null	20
1006	燕小六	Manager	5000	400	01-FEB-09	1005	20
1007	陆无双	Clerk	4000	500	01-FEB-09	1006	20
1008	黄蓉	Manager	5000	800	01-MAY-09	1005	30
1009	韦小宝	Salesman	4000	Null	20-FEB-09	1008	30
1010	郭靖	Salesman	4500	null	10-MAY-09	1008	30

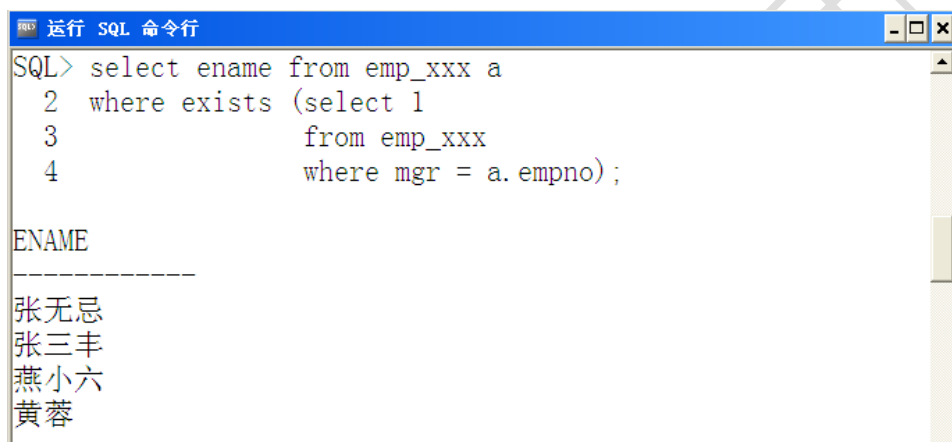
- ✓ 如图所示经理是编号为 1005、1001、1006、1008 的 4 个人
- ✓ 按表中的数据，人事结构图如下所示：



● **方法 1：使用关联子查询完成**

```
SQL> select ename from emp_xxx a
      where exists (select 1 from emp_xxx
                    where mgr = a.empno);
```

- ✓ exists 关键字判断子查询有没有数据返回，有则为 true，没有则为 false  
exists 不关心子查询的结果，所以子查询中 select 后面写什么都可以  
本例中我们写常量 "1"
- ✓ **sql 执行顺序**从主查询开始，把主查询中的 empno 数据传入子查询，作为条件中的参数

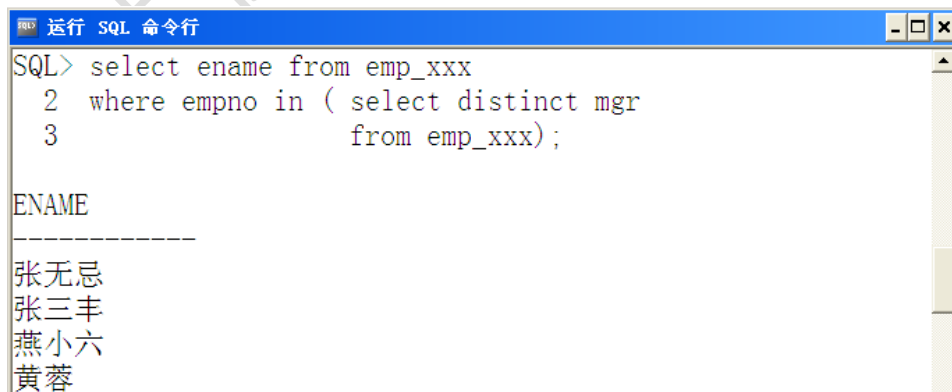


```
运行 SQL 命令行
SQL> select ename from emp_xxx a
2  where exists (select 1
3                from emp_xxx
4                where mgr = a.empno);

ENAME
-----
张无忌
张三丰
燕小六
黄蓉
```

● **方法 2：普通子查询**

```
SQL> select ename from emp_xxx
      where empno in (select distinct mgr
                     from emp_xxx);
```



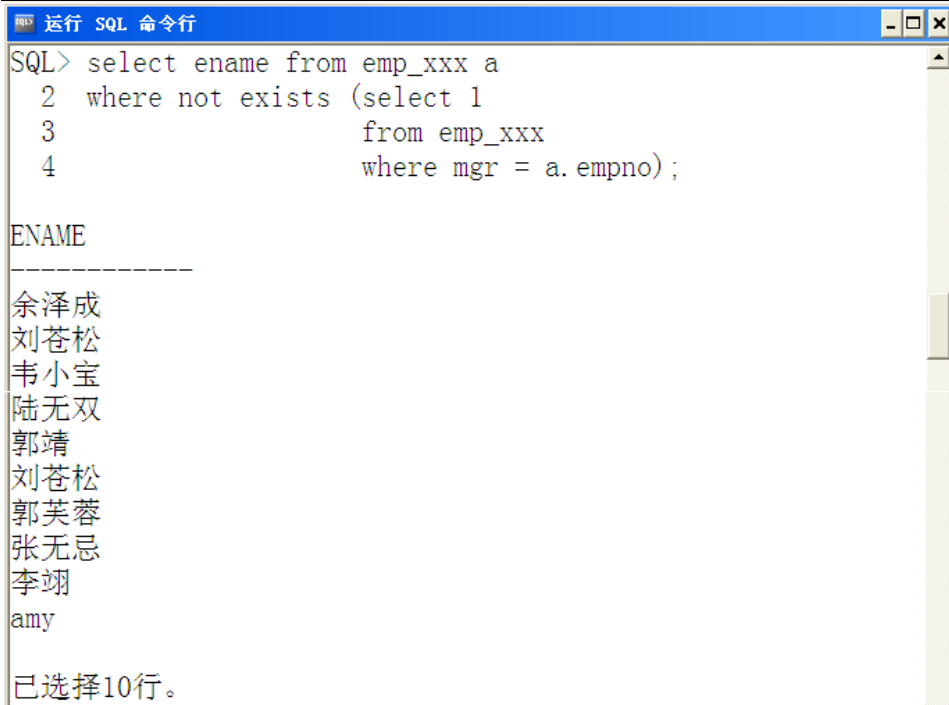
```
运行 SQL 命令行
SQL> select ename from emp_xxx
2  where empno in (select distinct mgr
3                from emp_xxx);

ENAME
-----
张无忌
张三丰
燕小六
黄蓉
```

### 【案例 23】哪些人不是别人的经理？

#### ● 方法 1：关联子查询

```
SQL> select ename from emp_xxx a
      where not exists (select 1 from emp_xxx
                        where mgr = a.empno);
```



```
运行 SQL 命令行
SQL> select ename from emp_xxx a
2  where not exists (select 1
3                      from emp_xxx
4                      where mgr = a.empno);

ENAME
-----
余泽成
刘苍松
韦小宝
陆无双
郭靖
刘苍松
郭芙蓉
张无忌
李翊
amy
已选择10行。
```

#### ● 方法 2：普通子查询

```
SQL> select ename from emp_xx
      where empno not in ( select distinct mgr
                          from emp_xxx
                          where mgr is not null);
--not in 要求结果不能有 null 值
```

- ✓ **注意：**not in (列表)：如果列表中有 null 值，将没有结果返回；in(列表)没有关系。  
如下所示，相当于：

```
SQL> select ename from emp_xxx
      where empno not in (1001,1005,1006,1008,null);
```



```

运行 SQL 命令行
SQL> select ename from emp_xxx
2  where empno not in (1001,1005,1006,1008,null);

未选定行

SQL> select ename from emp_xxx
2  where empno not in (1001,1005,1006,1008);

ENAME
-----
amy
刘苍松
李翊
郭芙蓉
陆无双
韦小宝
郭靖
余泽成
张无忌
刘苍松

已选择10行。

```

**【案例 24】哪些部门没有员工？**

```

SQL> select deptno, dname from dept_xxx d
      where not exists (select 1
                        from emp_xxx
                        where deptno = d.deptno) ;

```

```

运行 SQL 命令行
SQL> select deptno, dname from dept_xxx d
2  where not exists (select 1
3                    from emp_xxx
4                    where deptno = d.deptno);

DEPTNO DNAME
-----
40 operations

```

## 4. 集合操作 \*

数据库中的查询语句的**结果集( ResultSet )**：集合 A 和集合 B

- ✓ 集合 A： {1,2,3,4,5}
- ✓ 集合 B： {1,3,5,7,9}
- ✓ A 与 B 的合集： {1,2,3,4,5,7,9}
- ✓ A 与 B 的交集： {1,3,5}
- ✓ A 与 B 的差集： A-B {2,4}

### 4.1. 结果集操作

- 1) 两个结果集必须结构相同
  - ✓ 当列的个数、列的顺序、列的数据类型一致时，我们称这两个结果集结构相同
  - ✓ 只有结构相同的结果集才能做集合操作
- 2) 合集 union 和 union all
  - ✓ union 和 union all 的区别
    - union 去掉重复记录，union all 不去重
    - union 排序，union all 不排序
  - ✓ 在满足功能的前提下，优选 union all
- 3) 交集 intersect
- 4) 差集 minus( 两个集合做减法 )

#### 4.1.1. 合集

- 1) union 去重，排序

#### 【案例 25】合集( union )演示

```
SQL> select ename , salary from emp_xxx
      where deptno = 10
      union
      select ename , salary from emp_xxx
      where salary > 6000 ;
```

```
运行 SQL 命令行
SQL> select ename, salary from emp_xxx
2  where deptno = 10
3  union
4  select ename, salary from emp_xxx
5  where salary > 6000;

ENAME          SALARY
-----
刘苍松          8000
张三丰          15000
张无忌          8000
张无忌          10000
李翊            9000
郭芙蓉          5000

已选择6行。
```

## 2) union all 不去重，不排序

### 【案例 26】合集(union)演示

```
SQL> select ename , salary from emp_xxx
      where deptno = 10
      union all
      select ename , salary from emp_xxx
      where salary > 6000 ;
```

```
运行 SQL 命令行
SQL> select ename, salary from emp_xxx
2  where deptno = 10
3  union all
4  select ename, salary from emp_xxx
5  where salary > 6000;

ENAME          SALARY
-----
张无忌          10000
刘苍松          8000
李翊            9000
```

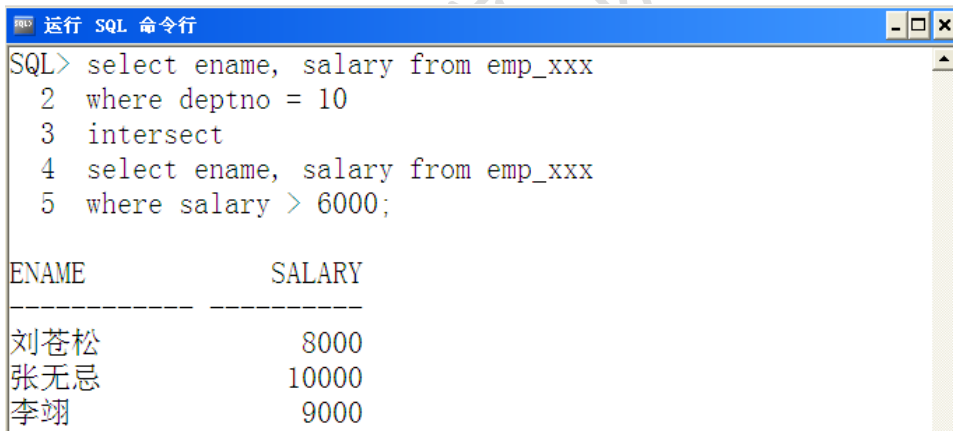
郭芙蓉	5000
张无忌	10000
刘苍松	8000
李翊	9000
张三丰	15000
张无忌	8000

已选择9行。

#### 4.1.2. 交集 intersect

##### 【案例 27】交集( intersect )演示

```
SQL> select ename , salary from emp_xxx
      where deptno = 10
      intersect
      select ename , salary from emp_xxx
      where salary > 6000 ;
```



```
SQL> select ename, salary from emp_xxx
2  where deptno = 10
3  intersect
4  select ename, salary from emp_xxx
5  where salary > 6000;
```

ENAME	SALARY
刘苍松	8000
张无忌	10000
李翊	9000

#### 4.1.3. 差集 minus

##### 【案例 28】差集( minus )演示

```
SQL> select ename , salary from emp_xxx
      where deptno = 10
      minus
      select ename , salary from emp_xxx
      where salary > 6000 ;
```

```

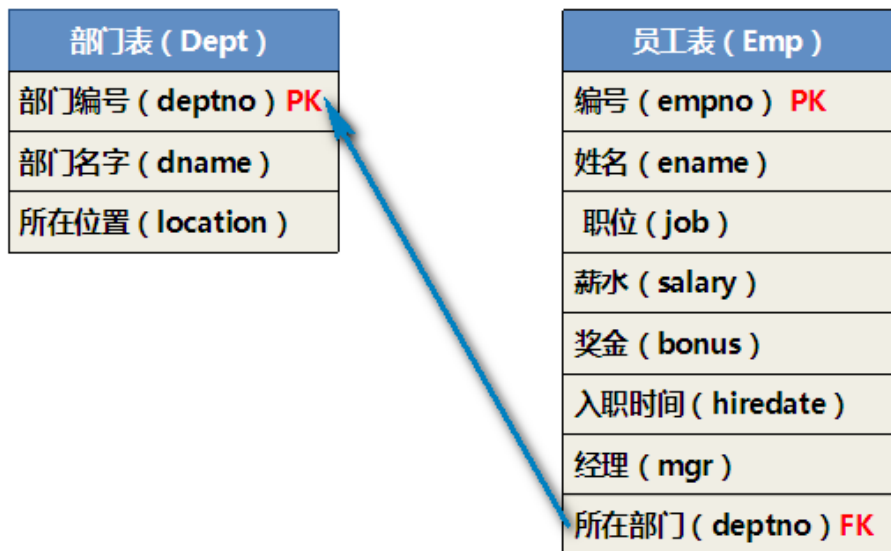
运行 SQL 命令行
SQL> select ename, salary from emp_xxx
2  where deptno = 10
3  minus
4  select ename, salary from emp_xxx
5  where salary > 6000;

ENAME          SALARY
-----
郭芙蓉          5000
    
```

## 5. 表间关联查询 \*\*

### 5.1. 表 emp\_xxx 和表 dept\_xxx 之间存在的参照关系


- 1) emp\_xxx 的所在部门( deptno )参照 dept\_xxx 的部门编码( deptno )
- 2) dept\_xxx 是主表( 父表 ), emp\_xxx 是从表( 子表 )



### 5.2. 表 emp\_ning 自身存在一种参照关系

员工的经理( mgr )列参照职员编码( empno )列

员工表 ( Emp )	
编号 ( empno )	PK
姓名 ( ename )	
职位 ( job )	
薪水 ( salary )	
奖金 ( bonus )	
入职时间 ( hiredate )	
经理 ( mgr )	
所在部门 ( deptno )	FK



### 5.3. 主键( PK )和外键( FK )

- 1) **主键**( Primary key,简称 PK ) --主键要求不重复，不能是空值
  - ✓ dept\_xxx 表的主键： 部门编码( deptno )
  - ✓ emp\_xxx 的主键： 职员编码( empno )
- 2) **外键**( Foreign key,简称 FK ) --外键参照主键的数据
  - ✓ emp\_xxx 的所在部门( deptno )是外键， 参照 dept\_xxx 的主键
  - ✓ emp\_xxx 的经理( mgr )列是外键， 参照 emp\_xxx 的主键

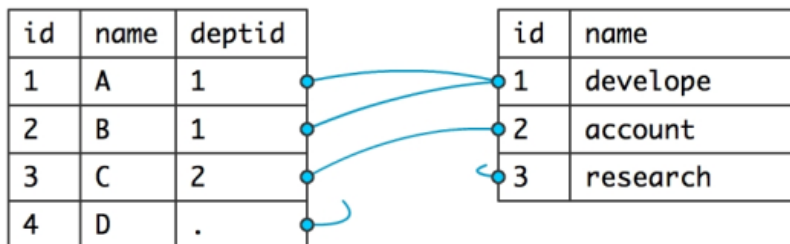
### 5.4. 内连接 \*\*

join 关键字用于连接两个表，语法如下：

表 1 join 表 2 on 条件

## 连接

内连接 join



id	name	deptid	name
1	A	1	develope
2	B	1	develope
3	C	2	account

### 【案例 29】列出员工的姓名和所在部门的名称和城市

```
SQL> select ename, dname, location
      from emp_xxx e join dept_xxx d
      on e.deptno = d.deptno ;
```

- 结果集( 结果集中的数据一定是在两个表中能找到匹配数据的记录 )

```
运行 SQL 命令行
SQL> select ename, dname, location
      2  from emp_xxx e join dept_xxx d
      3  on e.deptno = d.deptno;

ENAME      DNAME      LOCATION
-----
张无忌     developer   beijing
刘苍松     developer   beijing
李翊       developer   beijing
郭芙蓉     developer   beijing
```

张三丰	account	shanghai
燕小六	account	shanghai
陆无双	account	shanghai
黄蓉	sales	guangzhou
韦小宝	sales	guangzhou
郭靖	sales	guangzhou

已选择10行。

● 通过查看结果集，我们可以得出结论：

- 1) 子表( emp\_xxx )中的外键值( deptno )为 null 的数据不包含在结果集中  
amy 和余泽成的部门号( deptno )为 null

```
运行 SQL 命令行
SQL> select count(*) from emp_xxx;

COUNT(*)
-----
12

SQL> select ename from emp_xxx where deptno is null;

ENAME
-----
amy
余泽成
```

- 2) 父表( dept\_xxx )中主键值( deptno )没有被参照的数据不包含在结果集中  
部门号为 40 的部门没有员工

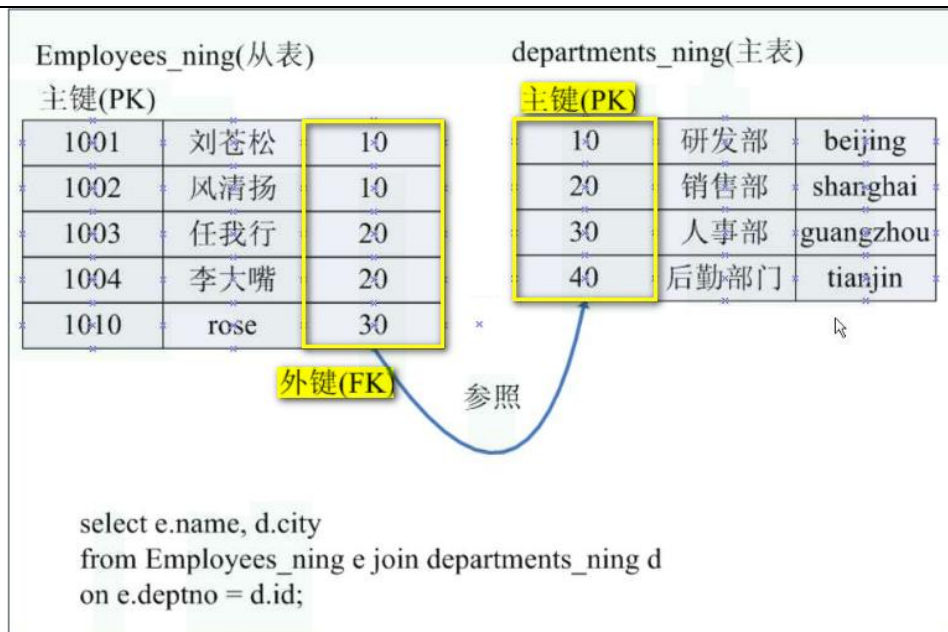
```
运行 SQL 命令行
SQL> select * from dept_xxx;

DEPTNO DNAME      LOCATION
-----
10 developer  beijing
20 account   shanghai
30 sales     guangzhou
40 operations tianjin
```

3) 1 对多关系

1 个部门( Department )对应多个员工( Employee )





【案例 30】列出员工的姓名和他的上司的姓名( 自连接 )

```
SQL> select t1.ename, t2.ename
      from emp_xxx t1 join emp_xxx t2
      on t1.mgr = t2.empno;

--t1 表示从表, t2 表示主表
-- 没有上司的员工不会列出
```

```
运行 SQL 命令行
SQL> select t1.ename, t2.ename
      2  from emp_xxx t1 join emp_xxx t2
      3  on t1.mgr = t2.empno;

ENAME      ENAME
-----
郭芙蓉      张无忌
李翊        张无忌
刘苍松      张无忌
```

黄蓉	张三丰
燕小六	张三丰
张无忌	张三丰
陆无双	燕小六
郭靖	黄蓉
韦小宝	黄蓉

已选择9行。

#### 5.4.1. 驱动表和匹配表

**表 1 join 表 2 on 条件**

- 1) 表 1 叫做**驱动表**，表 2 叫做**匹配表**
- 2) 等值连接方式下，驱动表和匹配表位置可以互换，不影响结果集

```
SQL> select ename, dname, location
      from emp_xxx t1 join dept_xxx t2    --t1 驱动表, t2 匹配表
      on t1.deptno = t2.deptno ;
```

**等价于**

```
SQL> select ename, dname, location
      from dept_xxx t1 join emp_xxx t2    --t1 驱动表, t2 匹配表
      on t1.deptno = t2.deptno ;
```

- 3) 执行方式：不论谁做驱动表，都会遍历驱动表，在匹配表中查找匹配数据

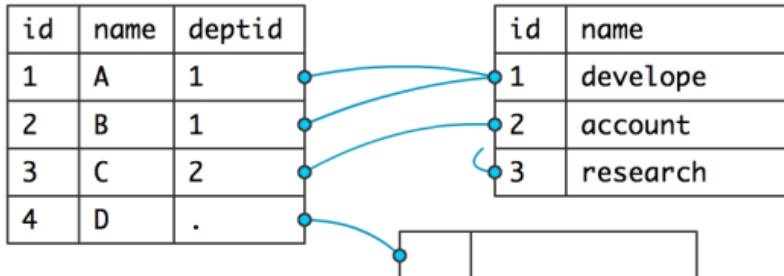
#### 5.5. 外连接\*\*

- 1) 左外连接语法结构：表 1 left outer join 表 2 on 条件
- 2) 右外连接语法结构：表 1 right outer join 表 2 on 条件
- 3) 外连接的特征：
  - ✓ 如果驱动表在匹配表中找不到匹配记录，则匹配一行空行
  - ✓ **外连接的结果集 = 内连接的结果集 + 驱动表在匹配表中匹配不上的记录和空值**
  - ✓ 外连接的本质是驱动表中的数据一个都不能少
    - left outer join 以左边的表为驱动表
    - right outer join 以右边的表为驱动表

##### ● 左外连接 left outer join

## 连接

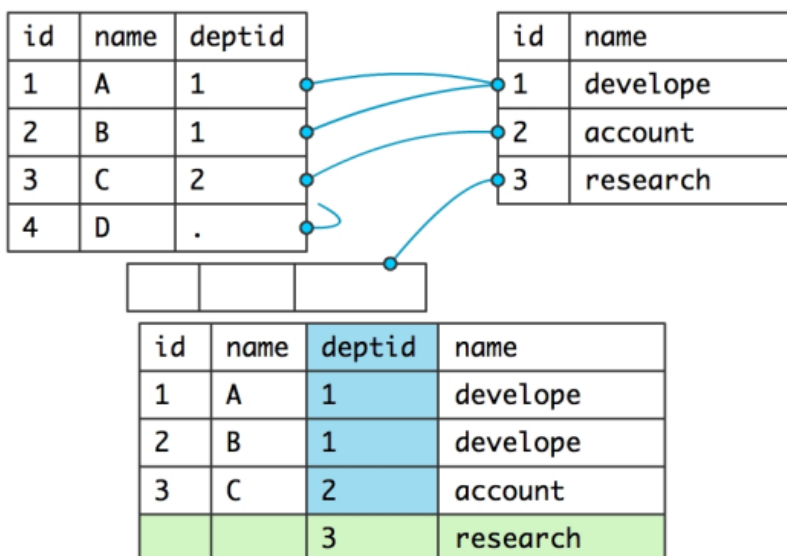
左外链接 left outer join



- 右外连接 right outer join

## 连接

右外连接 right outer join



**【案例 31】列出员工的姓名和他所在部门的名字，把没有部门的员工也查出来**

以员工表为驱动表。

● **方法 1( left outer join )**

```
-- 结果集中包括有部门的员工和没有部门的员工
-- 驱动表：emp_xxx
-- 匹配表：dept_xxx
-- left outer join 以左边的表为驱动表
```

```
SQL> select e.empno , ename , d.deptno , d.dname , d.location
      from emp_xxx e left outer join dept_xxx d
      on e.deptno = d.deptno ;
```

```
运行 SQL 命令行
SQL> select e.empno, ename, d.deptno, d.dname, d.location
2  from emp_xxx e left outer join dept_xxx d
3  on e.deptno = d.deptno;
```

EMPNO	ENAME	DEPTNO	DNAME	LOCATION
1004	郭芙蓉	10	developer	beijing
1003	李翊	10	developer	beijing
1002	刘苍松	10	developer	beijing
1001	张无忌	10	developer	beijing
1007	陆无双	20	account	shanghai
1006	燕小六	20	account	shanghai
1005	张三丰	20	account	shanghai
1010	郭靖	30	sales	guangzhou
1009	韦小宝	30	sales	guangzhou
1008	黄蓉	30	sales	guangzhou
1013	余泽成			
1012	amy			

已选择12行。

● **方法 2( right outer join )**

```
-- 结果集中包括有部门的员工和没有部门的员工
-- 驱动表：emp_xxx
```

```
-- 匹配表 : dept_xxx
-- right outer join 以右边的表为驱动表
```

```
SQL> select e.empno , e.ename , d.deptno , d.dname , d.location
      from dept_xxx d right outer join emp_xxx e
      on e.deptno = d.deptno ;
```

```
运行 SQL 命令行
SQL> select e.empno, e.ename, d.deptno, d.dname, d.location
2  from dept_xxx d right outer join emp_xxx e
3  on e.deptno = d.deptno;

EMPNO ENAME          DEPTNO DNAME          LOCATION
-----
1004 郭芙蓉          10 developer    beijing
1003 李翊              10 developer    beijing
1002 刘苍松            10 developer    beijing
1001 张无忌            10 developer    beijing
1007 陆无双            20 account      shanghai
1006 燕小六            20 account      shanghai
1005 张三丰            20 account      shanghai
1010 郭靖              30 sales         guangzhou
1009 韦小宝            30 sales         guangzhou
1008 黄蓉              30 sales         guangzhou
1013 余泽成
1012 amy
```

已选择12行。

**【案例 32】**列出员工的姓名和他所在部门的名字，把没有员工的部门也查出来

以部门表为驱动表。

● 方法 1( left outer join )

```
-- 结果集中包括有员工的部门和没有员工的部门
-- 驱动表 : dept_xxx
-- 匹配表 : emp_xxx
-- left outer join 以左边的表为驱动表
```

```
SQL> select e.empno , e.ename , d.deptno , d.dname , d.location
```

```
from dept_xxx d left outer join emp_xxx e
on e.deptno = d.deptno;
```



运行 SQL 命令行

```
SQL> select e.empno, e.ename, d.deptno, d.dname, d.location
2  from dept_xxx d left outer join emp_xxx e
3  on e.deptno = d.deptno;
```

EMPNO	ENAME	DEPTNO	DNAME	LOCATION
1001	张无忌	10	developer	beijing
1002	刘苍松	10	developer	beijing
1003	李翊	10	developer	beijing
1004	郭芙蓉	10	developer	beijing
1005	张三丰	20	account	shanghai
1006	燕小六	20	account	shanghai
1007	陆无双	20	account	shanghai
1008	黄蓉	30	sales	guangzhou
1009	韦小宝	30	sales	guangzhou
1010	郭靖	30	sales	guangzhou
		40	operations	tianjin

已选择11行。

● 方法 2( right outer join )

```
-- 结果集中包括有员工的部门和没有员工的部门
-- 驱动表 : dept_xxx
-- 匹配表 : emp_xxx
-- right outer join 以右边的表为驱动表
```

```
SQL> select e.empno , e.ename , d.deptno , d.dname , d.location
      from emp_xxx e right outer join dept_xxx d
      on e.deptno = d.deptno ;
```

```
运行 SQL 命令行
SQL> select e.empno, e.ename, d.deptno, d.dname, d.location
2  from emp_xxx e right outer join dept_xxx d
3  on e.deptno = d.deptno;
```

EMPNO	ENAME	DEPTNO	DNAME	LOCATION
1001	张无忌	10	developer	beijing
1002	刘苍松	10	developer	beijing
1003	李翊	10	developer	beijing
1004	郭芙蓉	10	developer	beijing
1005	张三丰	20	account	shanghai
1006	燕小六	20	account	shanghai
1007	陆无双	20	account	shanghai
1008	黄蓉	30	sales	guangzhou
1009	韦小宝	30	sales	guangzhou
1010	郭靖	30	sales	guangzhou
		40	operations	tianjin

已选择11行。

### 【案例 33】哪些部门没有员工？

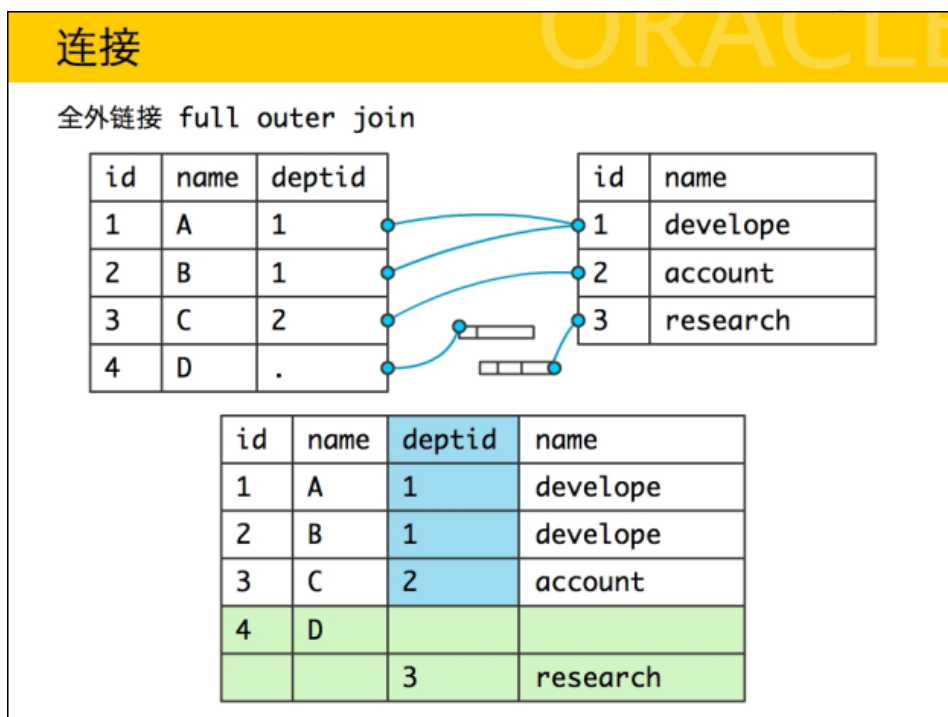
```
--第 1 步：把部门表做驱动表，所有的部门都列出来
--第 2 步：添加 where 条件“匹配表的主键 is null”
SQL> select e.empno, e.ename, d.deptno, d.dname, d.location
      from dept_xxx d left outer join emp_xxx e
      on e.deptno = d.deptno
      where e.empno is null;
```

```
运行 SQL 命令行
SQL> select e.empno, e.ename, d.deptno, d.dname, d.location
2  from dept_xxx d left outer join emp_xxx e
3  on e.deptno = d.deptno
4  where e.empno is null;
```

EMPNO	ENAME	DEPTNO	DNAME	LOCATION
		40	operations	tianjin

## 5.6. full outer join 全外连接 \*\*

- 1) 全外连接可以把两个表中的记录全部查出来
- 2) 全外连接的结果集 = 内连接的结果集 +  
驱动表中在匹配表中找不到匹配记录的数据和 null 值 +  
匹配表中在驱动表中找不到匹配记录的数据和 null 值
- 3) 驱动表和匹配表可以互换



### 【案例 34】全外连接

```
SQL> select e.empno , e.ename , d.deptno , d.dname
      from emp_xxx e full outer join dept_xxx d
      on e.deptno = d.deptno ;
```



```

运行 SQL 命令行
SQL> select e.empno, e.ename, d.deptno, d.dname
2  from emp_xxx e full outer join dept_xxx d
3  on e.deptno = d.deptno;

  EMPNO ENAME          DEPTNO DNAME
-----
    1004 郭芙蓉             10 developer
    1003 李翊              10 developer
    1002 刘苍松             10 developer
    1001 张无忌             10 developer
    1007 陆无双             20 account
    1006 燕小六             20 account
    1005 张三丰             20 account
    1010 郭靖              30 sales
    1009 韦小宝             30 sales
    1008 黄蓉              30 sales
    1013 余泽成
    1012 amy
                                40 operations

已选择13行。

```

## 6. 小结外连接

- 1) 外连接的结果集 = 内连接的结果集 + 驱动表在匹配表中找不到匹配记录的数据和空值
- 2) 驱动表和匹配表不能互换
- 3) 写外连接的方式：
  - a) 先写内连接
  - b) **使用 left 或 right 不重要**，关键是确定谁做驱动表，
- 4) 匹配表 pk is null 的应用：[哪些部门没有员工？](#)