# Mysql查询:

**-- count 统计个数;**

**-- sum 求和;**

**-- max 最大值;**

**-- AVG 平均值**

**-- min 最小值**

**-- select 查询可以嵌套**

**-- concat 拼接 类似于python中的字符串拼接+**

**-- 如下:(简单嵌套)**

**SELECT \* from employee where sal= (SELECT MAX(sal) from employee);**

## 简单查询:

**select \* from employee; #查询整张表  
select empno,ename,job as ename\_job from employee;**

## 精确条件查询:

### 等于=&&不等于!= or <>&&大于>&&小于<:

**select \* from employee where ename='后裔';**

**等于查询  
select \* from employee where sal != 50000;  
select \* from employee where sal <> 50000;**

**不等于查询  
select \* from employee where sal > 10000;**

**大于查询**

**Select \* from employee where sal < 10000;**

**小于查询**

## 模糊条件查询:

**select \* from employee where ename like '林%';**

**模糊匹配后面的,林字开头的姓名**

**select \* from employee where ename like '%林%';**

**模糊匹配所有包含林字的无论开头结尾或者中间.**

## 范围查询:

**关键字:where 字段名 between 开始数据 and 结束数据**

**select \* from employee where sal between 10000 and 30000;**

**注意: 数字格式的SQL数据可以不用加引号,其他非数字的数据都需要加引号.  
select \* from employee where hiredate between '2011-01-01' and '2017-12-1';**

## 离散查询:

**select \* from employee where ename in ('猴子','林俊杰','小红','小胡');**

**只要在此表的ename能找到 小明 林俊杰 小红 小胡 就会显示.**

## 清除重复值:

**select distinct(job) from employee;**

**清除job字段中值的重复**

## 统计查询（聚合函数）:

**count(code)或者count(\*)  
select count(\*) from employee;  
select count(ename) from employee;**

**统计几条数据.  
  
sum() 计算总和  
select sum(sal) from employee;  
  
max() 计算最大值  
select \* from employee where sal= (select max(sal) from employee);  
  
avg() 计算平均值  
select avg(sal) from employee;  
  
min() 计算最低值  
select \* from employee where sal= (select min(sal) from employee);  
  
concat函数： 起到连接作用  
select concat(ename,' 是 ',job) as aaaa from employee;**

# mysql查询子句之二group by分组查询（分组）:

**简介：详解group by的用法以及应用场景  
作用：把行 按 字段 分组  
语法：group by 列1，列2....列N  
适用场合：常用于统计场合，一般和聚合函数连用**

**select deptnu,count(\*) from employee group by deptnu;**

**# 统计所有的数据,并通过 group by 以deptnu 进行分组.  
select deptnu,job,count(\*) from employee group by deptnu,job;**

**# 统计所有的数据,并通过 group by 以deptnu , job 进行分组.  
  
select job,count(\*) from employee group by job;**

# mysql查询子句之三having条件查询（筛选）:

**简介：详解having的用法以及应用场景  
作用：对查询的结果进行筛选操作  
语法：having 条件 或者 having 聚合函数 条件 也可以用 != 或 = 或 < 或 >  
适用场合：一般跟在group by之后**

**select job,count(\*) from employee group by job having job ='文员';**

**#将查询出来的结果通过having去筛选**

**select deptnu,job,count(\*) from employee group by deptnu,job having count(\*)>=2;  
select deptnu,job,count(\*) as 总数 from employee group by deptnu,job having 总数>=2;**

**#将查询出来的结果的count条数重命名 总数 通过having去对 总数 筛选**

# mysql查询子句之四order by排序查询（排序）:

**简介：详解order by的用法以及应用场景  
作用：对查询的结果进行排序操作  
语法：order by 字段1,字段2 .....  
适用场合：一般用在查询结果的排序**

**select \* from employee order by sal;**

**#查询所有数据并以sal进行排序  
select \* from employee order by hiredate;**

**查询employee表所有数据并以hiredate进行正序排序  
select deptnu,job,count(\*) as 总数 from employee group by deptnu,job having 总数>=2 order by deptnu desc;**

**# 查询出employee表的deptnu, job数据并统计,将统计出的数据进行筛选后以deptnu进行倒序排列.  
select deptnu,job,count(\*) as 总数 from employee group by deptnu,job having 总数>=2 order by deptnu asc;**

**# 查询出employee表的deptnu, job数据并统计,将统计出的数据进行筛选后以deptnu进行正序排列.  
  
select deptnu,job,count(\*) as 总数 from employee group by deptnu,job having 总数>=2 order by deptnu;**

**# 查询出employee表的deptnu, job数据并统计,将统计出的数据进行筛选后以deptnu进行正序排列.**注意顺序: 顺序：where ---- group by ----- having ------ order by

# mysql查询子句之五limit限制查询（限制）:

**简介：详解limit的用法以及应用场景  
作用：对查询结果起到限制条数的作用  
语法：limit n，m n:代表起始条数值，不写默认为0；m代表：取出的条数**

**注意: n是和编程语言中的计数一样,从0开始的  
适用场合：数据量过多时，可以起到限制作用**

**select \* from XD.employee limit 4,5;**

**#查看employee表的所有信息,取出第4条数据开始后的5条数据(注意这里的4开始指的是第五条数据,因为是从零开始的,机器与人的计数不一样)**

## Ifnull(列名,重命名)

# mysql查询之exists型子查询:

**exists型子查询后面是一个受限的select查询语句  
exists子查询，如果exists后的内层查询能查出数据，则返回 TRUE 表示存在；为空则返回 FLASE则不存在。**

**分为俩种：exists跟 not exists  
select 1 from employee where 1=1;**

**#1就是返回所有信息,where 1=1 这个是判断语句,1=1条件确实成立,呢么就是True 前面的查询数据就会执行,如果是where 1=2 此时条件不成立,就是false,前面的查询语句就会执行直接返回空.  
示例语句: select \* from 表名 a where exists (select 1 from 表名2 where 条件);  
eg:查询出公司有员工的部门的详细信息  
select \* from dept a where exists (select 1 from employee b where a.deptnu=b.deptnu);**

**select \* from dept a where not exists (select 1 from employee b where a.deptnu=b.deptnu);**

# mysql查询之左连接查询与右连接查询:

**简介：详解左右连接的用法以及应用场景  
左连接称之为左外连接 右连接称之为右外连接 这俩个连接都是属于外连接  
左连接关键字：left join 表名 on 条件 / left outer 表名 join on 条件**

**右连接关键字：right join 表名 on 条件/  
right outer 表名 join on 条件  
左连接说明： left join 是left outer join的简写，左(外)连接，左表(a\_table)的记录将会全部表示出来， 而右表  
(b\_table)只会显示符合搜索条件的记录。右表记录不足的地方均为NULL。  
右连接说明：right join是right outer join的简写，与左(外)连接相反，右(外)连接，左表(a\_table)只会显示符合  
搜索条件的记录，而右表(b\_table)的记录将会全部表示出来。左表记录不足的地方均为NULL。**

**eg:列出部门名称和这些部门的员工信息，同时列出那些没有的员工的部门  
dept部门表，employee员工信息表 salgrade工资等级表  
select a.dname,b.\* from dept a left join employee b on a.deptnu=b.deptnu;  
select b.dname,a.\* from employee a right join dept b on b.deptnu=a.deptnu;**

# mysql查询之内连接查询与联合查询:

## 内连接查询:

**详解内连接与联合查询的用法以及应用场景  
内连接：获取两个表中字段匹配关系的记录  
主要语法：INNER JOIN 表名 ON 条件;**

**eg:想查出员工张飞的所在部门的地址  
select a.addr from dept a inner join employee b on a.deptnu=b.deptnu and b.ename='张飞';**

**select a.addr from dept a,employee b where a.deptnu=b.deptnu and b.ename='张飞';**

## 联合查询：

**就是把多个查询语句的查询结果结合在一起**  
**主要语法1：... UNION ... （去除重复）**

**主要语法2：... UNION ALL ...（不去重复）  
union查询的注意事项：**

**(1)两个select语句的查询结果的“字段数”必须一致；  
(2)通常，也应该让两个查询语句的字段类型具有一致性；  
(3)也可以联合更多的查询结果；  
(4)用到order by排序时，需要加上limit（加上最大条数就行），需要对子句用括号括起来  
eg:对销售员的工资从低到高排序，而文员的工资从高到低排序  
(select \* from employee a where a.job = '销售员' order by a.sal limit 999999 ) union (select \* from employee b where b.job = '文员' order by b.sal desc limit 999999);**

# 具体环境:

## 数据库IP: 121.5.64.57

## 用户名: caibo

## ,密码: 123456

## 数据库名: SHIZHAN

## 具体表名信息:

**dept 部门表(部门的编号/名字/地址)**

**employee员工信息表(员工的编号/姓名/职位/员工的上级编号/入职日期/薪资信息/部门编号)**

**salgrade 薪资等级表 (薪资等级/薪资等级开始值/薪资等级结束值)**

# mysql查询之项目高级查询实战(一)

## 1.查出至少有一个员工的部门。显示部门编号、部门名称、部门位置、部门人数。

## 2. 列出薪金比安琪拉高的所有员工。

## 3. 列出所有员工的姓名及其直接上级的姓名。

## 4. 列出受雇日期早于直接上级的所有员工的编号、姓名、部门名称。

## 5. 列出部门名称和这些部门的员工信息，同时列出那些没有员工的部门。

## 6. 列出所有文员的姓名及其部门名称，所在部门的总人数。

# mysql查询之项目高级查询实战(二):

## 列出最低薪金大于15000的各种工作及从事此工作的员工人数。

## 列出在销售部工作的员工的姓名，假定不知道销售部的部门编号。

## 列出与诸葛亮从事相同工作的所有员工及部门名称。 列出薪金比 在部门30工作的员工的薪金 还高的员工姓名和薪金、部门名称。

## 列出每个部门的员工数量、平均工资。

## 5. 列出薪金高于公司平均薪金的所有员工信息，所在部门名称，上级领导，工资等级。