**SQL**

1. **建表语句**

create table tb\_user(

id int auto\_increment primary key comment 'ID唯一标识',

name varchar(10) not null unique comment '姓名' ,

age int check (age > 0 && age <= 120) comment '年龄' ,

status char(1) default '1' comment '状态',

gender char(1) comment '性别' ,

school\_id int comment ‘毕业学校’,

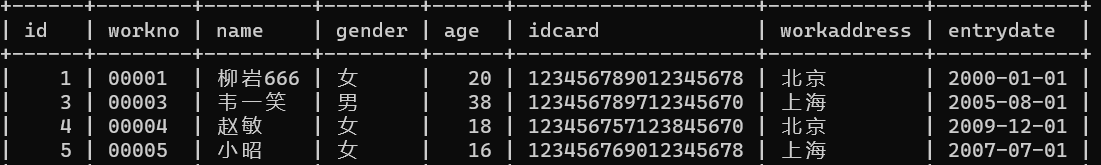
constraint foreignkeyname ke(school\_id) references school(id)

) comment ‘用户表’;

为已有表设置主键：alter table emp add primary key(id);

1. **统计每个月生日的人数,降序，从第一条记录开始显示10条**

select month(birthday) as b\_month, count(\*) from people group by b\_month order by b\_month asc/desc limit 0,10;



1. **查询所有员工的入职天数，并根据入职天数倒序排序**

**select name,datediff(curdate(),entrydate) as days from emp order by days desc;**

1. **查询每一个员工的姓名 , 及关联的部门的名称（内连接 隐+显）**

select emp.name , dept.name from emp , dept where emp.dept\_id = dept.id ;

select e.name, d.name from emp e (inner) join dept d on e.dept\_id = d.id;（显）

1. **查询emp表的所有数据, 和对应的部门信息 （左外连接）**

select e.\*, d.name from emp e left (outer) join dept d on e.dept\_id = d.id;

1. **查询dept表的所有数据, 和对应的员工信息(右外连接)**

select d.\*, e.\* from emp e right (outer) join dept d on e.dept\_id = d.id;

1. **查询所有员工 emp 及其领导的名字 emp , 如果员工没有领导, 也需要查询出来**

select a.name '员工', b.name '领导' from emp a left join emp b on a.managerid = b.id;

1. **将将薪资低于 5000 的员工 , 和 年龄大于 50 岁的员工全部查询出来.**

select \* from emp where salary < 5000

union (all)

select \* from emp where age > 50;

1. **查询 "销售部" 的所有员工信息**

select \* from emp where dept\_id = (select id from dept where name = '销售部');

查询指定入职日期之后入职的员工信息

select \* from emp where entrydate > (select entrydate from emp where name = '方东

白');

1. **查询比 财务部 所有人工资都高的员工信息**

select \* from emp where salary > all ( select salary from emp where dept\_id =

(select id from dept where name = '财务部') );

1. **查询与 "张无忌" 的薪资及直属领导相同的员工信息 ;**

select \* from emp where (salary,managerid) = (select salary, managerid from emp

where name = '张无忌');

1. **查询入职日期是 "2006-01-01" 之后的员工信息 , 及其部门信息**

select e.\*, d.\* from (select \* from emp where entrydate > '2006-01-01') e left

join dept d on e.dept\_id = d.id ;

1. **多表查询案例**
   1. **查询员工的姓名、年龄、职位、部门信息 （隐式内连接）**

select e.name , e.age , e.job , d.name from emp e , dept d where e.dept\_id = d.id;

* 1. **查询年龄小于30岁的员工的姓名、年龄、职位、部门信息（显式内连接）**

select e.name , e.age , e.job , d.name from emp e inner join dept d on e.dept\_id =

d.id where e.age < 30;

* 1. **查询拥有员工的部门ID、部门名称**

select distinct d.id , d.name from emp e , dept d where e.dept\_id = d.id;

* 1. **查询所有年龄大于40岁的员工, 及其归属的部门名称; 如果员工没有分配部门, 也需要展示出 来(外连接)**

select e.\*, d.name from emp e left join dept d on e.dept\_id = d.id where e.age >

40 ;

* 1. **查询所有员工的工资等级**

-- 方式一

select e.\* , s.grade , s.losal, s.hisal from emp e , salgrade s where e.salary >=

s.losal and e.salary <= s.hisal;

-- 方式二

select e.\* , s.grade , s.losal, s.hisal from emp e , salgrade s where e.salary

between s.losal and s.hisal;

* 1. **查询 "研发部" 所有员工的信息及 工资等级**

select e.\* , s.grade from emp e , dept d , salgrade s where e.dept\_id = d.id and ( e.salary between s.losal and s.hisal ) and d.name = '研发部';

* 1. **查询 "研发部" 员工的平均工资**

select avg(e.salary) from emp e, dept d where e.dept\_id = d.id and d.name = '研发 部';

* 1. **查询工资比 "灭绝" 高的员工信息。**

select \* from emp where salary > ( select salary from emp where name = '灭绝' );

* 1. **查询比平均薪资高的员工信息**

select \* from emp where salary > ( select avg(salary) from emp );

* 1. **查询低于本部门平均工资的员工信息**

select \* from emp e2 where e2.salary < ( select avg(e1.salary) from emp e1 where e1.dept\_id = e2.dept\_id );

* 1. **查询所有的部门信息, 并统计部门的员工人数**

select d.id, d.name , ( select count(\*) from emp e where e.dept\_id = d.id ) '人数'

from dept d;

* 1. **查询所有学生的选课情况, 展示出学生名称, 学号, 课程名称**

select s.name , s.no , c.name from student s , student\_course sc , course c where s.id = sc.studentid and sc.courseid = c.id ;

1. **函数**
   1. count();max();min();sum();avg():
   2. concat(s1,s);lower(s);upper(s);lpad(s,n,pad);rpad(s,n,p);trim(str);substring(str,start,len);//start从1开始
   3. cell(x)上；floor(x)下;mod(x,y);rand();round(x,y)
   4. curdate();curtime();now();year(date);month(date);

day(date);date\_add(date,interval expr type);//加时间段

datediff(date1,date2);//date1-date2

1. **count(\*)包括null字段，count（字段）不记录null字段**
2. **where条件**

比较符：>,>=,<,<=,=,<>或!=,between..and..包含边界,in(1,4)符合1或4，like ‘\_%’\_占一位，%占多位,is null；逻辑：and-&&，or-||，not-！

group by-分组查询

分组查询的字段一般为聚合函数和分组字段，查询其他字段无任何意义。

按多段分组记录条数就是分组内所有种类的笛卡尔乘积

1. 流程函数

流程函数也是很常用的一类函数，可以在SQL语句中实现条件筛选，从而提高语句的效率。

|  |  |
| --- | --- |
| **函数** | **功能** |
| if(value , t , f) | 如果value为true，则返回t，否则返回f |
| ifnull(value1 , value2) | 如果value1不为空，返回value1，否则返回value2 |
| case when[ val1 ] then [res1] ...  else [ default ] end | 如果val1为true，返回res1，... 否则返回default默认值 |
| case [ expr ] when [ val1 ] then  [res1] ... else [ default ] end | 如果expr的值等于val1，返回  res1，... 否则返回default默认值 |

1. 语法
2. 添加外键

案例:

为emp表的dept\_id字段添加外键约束,关联dept表的主键id。

1 CREATE TABLE 表名(

2

字段名 数据类型,

3

...

4

[CONSTRAINT] [外键名称] FOREIGN KEY (外键字段名) REFERENCES 主表 (主表列名)

5 );

1 ALTER TABLE 表名 ADD CONSTRAINT 外键名称 FOREIGN KEY (外键字段名) REFERENCES 主表 (主表列名) ;

1 alter table emp add constraint fk\_emp\_dept\_id foreign key (dept\_id) references

dept(id);