**一、数据库存储数据**

**字段、记录、表、约束（主键、外键、唯一键、非空、check、default、触发器）**

**1、表的相关数据**

**例：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **员工表** |  |  |  | 主键 | 外键 |
| 员工姓名 | 员工工资 | 员工性别 | 员工年龄 | **学生编号** | 部门编号 |
| 张三 | 1000 | 男 | 20 | **100000** | 1 |
| 李四 | 2000 | 男 | 20 | **100001** | 1 |
| 小娟 | 1500 | 女 | 26 | **100002** | 1 |
| 张三 | 1000 | 男 | 20 | **100003** | 2 |

**部门表**

|  |  |  |  |
| --- | --- | --- | --- |
| 部门姓名 | 部门人数 | 部门地址 | 部门编号 |
| 市场部 | 100 | 西安 | 1 |
| 研发部 | 50 | 北京 | 2 |

**字段（列、属性）：一个事物的某一个静态的特征。**

**记录（元组）：字段的组合，表示的是一个具体的事物。**

**表：记录的组合，表示的是同一类型事物的集合。**

**表和字段、记录的关系：**

**字段是事物的属性**

**记录是事物本身**

**表是事物的集合**

**2、create table 命令**

**通过图形化界面建表**

**create table （最后一个字段的后面建议不要写逗号）**

**3、什么是约束（**constraint**）**

**定义：对一个表中的属性操作的限制叫做约束**

**分类：**

**主键约束：不允许重复元素，避免了数据的冗余**

**外键约束：通过外键约束从语法上保证了本事物所关联的其他事物一定是存在的**

**事物和事物的关系是通过外键来体现的**

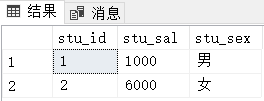
create table emp
(
emp\_id int constraint pk\_emp\_id primary key,
dept\_id int constraint fk\_dept\_id foreign key references dept(dept\_id)
)

**check约束：保证事物属性的取值在合法的范围之内**

**default约束：保证事物的属性一定会有一个值**

--创建表
create table student
(
stu\_id int primary key,
stu\_sal int check (stu\_sal > 1000 and stu\_sal <= 8000), --check约束
stu\_sex nchar(1) default('男') --()可以省略 在数据库中字符串必须用'括起来 --default约束
)

--插入数据
insert into student(stu\_id,stu\_sal) values(1,1000);
insert into student(stu\_id,stu\_sal,stu\_sex) values(2,6000,'女');



**唯一约束：保证了事物属性的取值不能重复，但允许为空**

**主键不能为空，唯一键仅允许一行为空（oracle 允许多行为空）**

**not null:属性值非空**

**4、表和约束的区别**

**数据库是通过表来解决事物的存储问题的**

**数据库是通过约束来解决事物取值的有效性和合法性的问题**

**建表的过程就是指定事物属性及其事物属性各种约束的过程**

**5、什么是关系**

**定义：表和表之间的关系**

**实现方式：通过设置不同形式的外键来体现表和表的不同关系**

**分类：（假设是A表和B表）**

**一对一：既可以把表A的主键充当表B的外键**

**也可以把表B的主键充当表A的外键**

**一对多：把表A的主键充当表B的外键（一A多B）**

**多对多：多读多需要用第三张表来表示（A的主键充当表C的外键；B的主键充当表C的外键）**

**班级和老师：班级一张表、老师一张表、班级和老师的关系一张表**

create table classTable
(
classTable\_id int primary key,
classTable\_num int not null,
classTable\_name nvarchar(100)
)
create table teacher
(
teacher\_id int primary key,
teacher\_name nvarchar(100)
)
create table class\_teacher
(
classTable\_id int constraint fk\_classTable\_id foreign key references classTable(classTable\_id),
teacher\_id int constraint fk\_teacher\_id foreign key references teacher(teacher\_id),
subject nvarchar(20),
constraint pk\_classTable\_id\_teacher\_id primary key(classTable\_id,teacher\_id,subject) --组合主键
)

**6、主键**

**定义：能够唯一标识一个事物的一个字段或多个字段的组合，被称为主键**

**附注：含有主键的表叫主键表**

**主键通常都是整数，不建议使用字符串当主键（如果主键是用于集群式服务，可以考虑用字符串当主键）。**

**主键的值通常都不允许修改，除非本记录被删除**

**主键不要定义成id，而要定义成表名Id或者表名\_id**

**要用代理主键，不要用业务主键：**

**任何一张表，强烈建议不要使用有业务含义的字段充当主键**

**我们通常都是在表中单独添加一个整型的编号充当主键字段**

**7、外键**

**定义：如果一个表中的若干个字段是来自另外若干个表的主键或唯一键，则这若干个字段就是外键**

**注意：外键通常是来自另外表的主键而不是唯一键，因为唯一键可能为null**

**外键不一定来自另外的表，也可能来自本表的主键**

**含有外键的表叫外键表，外键字段来自的那一张表叫做主键表**

**问题：先删主键表还是外键表？**

**答：先删外键表**

**如果先删主键表，会报错，因为这会导致外键表中数据引用失败**

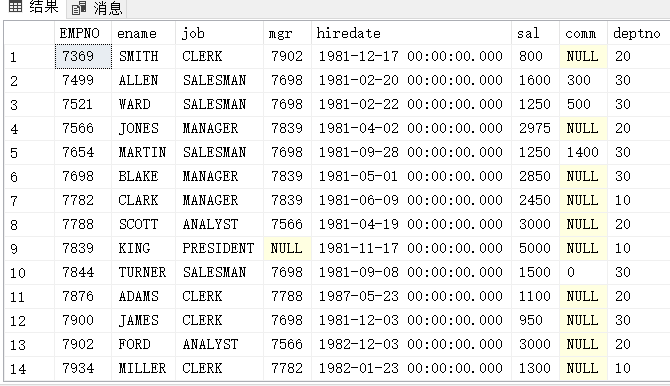
**二、数据库操作数据**

**insert、update、delete、T-SQL、存储过程、函数、触发器**

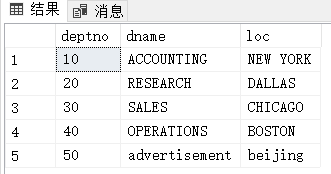
**三、数据库显示数据**

**★查询（select）**

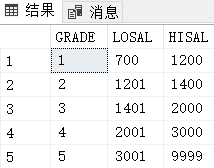
**emp表：**



**dept表：**

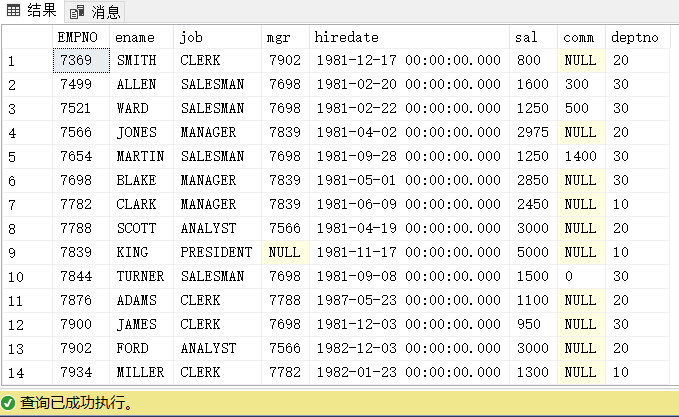


**SALGRADE表：**



**1、计算列**

select \* from emp;
select empno,ename from emp;
select ename,sal\*12 as "年薪" from emp; --as可以省略



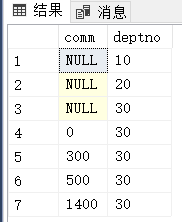


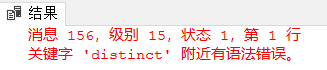


**2、distinct(不重复查询)**

select distinct deptno from emp;
select distinct comm, deptno from emp;
select deptno,distinct comm from emp; --逻辑错误

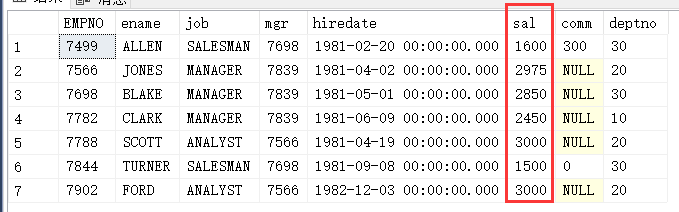






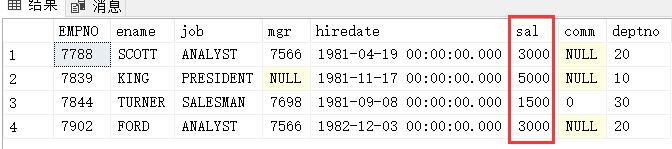
**3、between**

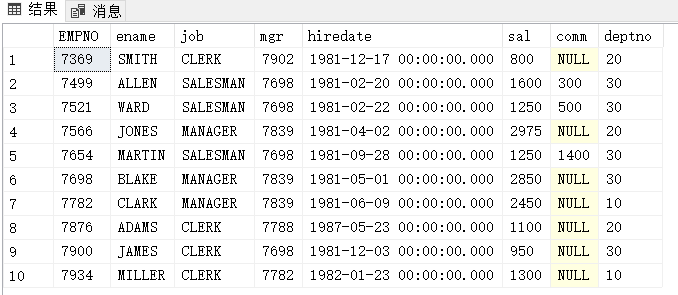
select \* from emp
where sal >= 1500 and sal <= 3000;
--等价于
select \* from emp
where sal between 1500 and 3000;



**4、in**

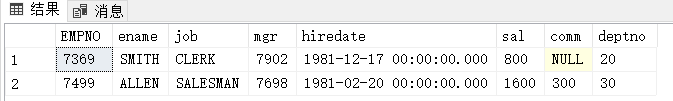
select \* from emp where sal in (1500,3000,5000); --等价于select \* from emp where sal =1500 or sal = 3000 or sal = 5000;
select \* from emp where sal not in (1500,3000,5000); --等价于select \* from emp where sal !=1500 and sal != 3000 and sal != 5000;
--等价于select \* from emp where sal <> 1500 and sal <> 3000 and sal <> 5000;

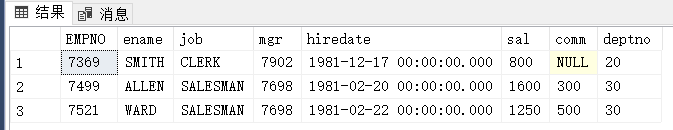




**5、top**

select top 2 \* from emp;
select top 15 percent \* from emp; -- 14 × 15％ = 2.1 → 3

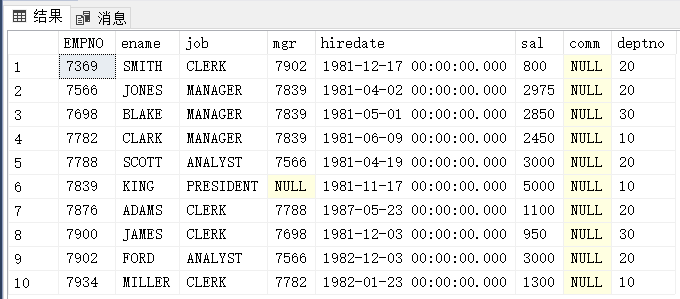


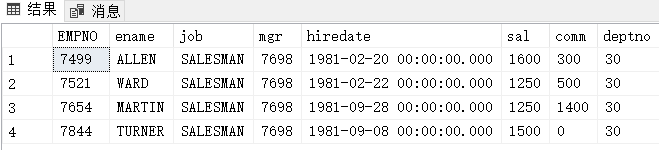


**6、null**

**只有is not和is两种运算**

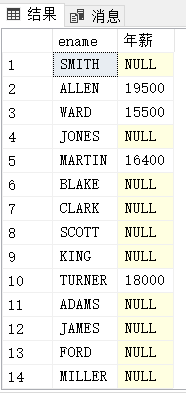
select \* from emp where comm is null;
select \* from emp where comm is not null;





**任何数字和null参与的数学运算结果都为null**

--输出员工的姓名、年薪（工资＋奖金）
select ename,sal\*12+comm as "年薪" from emp;



**7、order by**

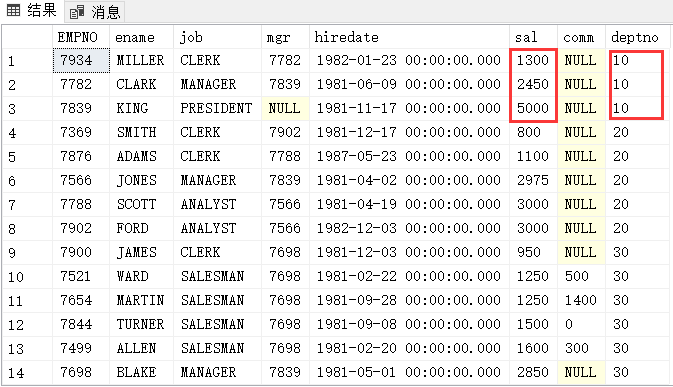
**order by a,b; --a、b都是升序**

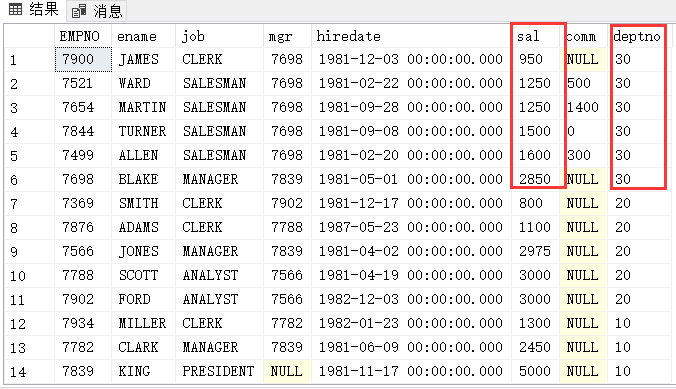
**order by a desc,b; --a降序b升序**

**order by a,b desc; --a升序b降序**

**order by a desc,b desc; --a、b都是降序**

select \* from emp order by deptno,sal;
select \* from emp order by deptno desc,sal;

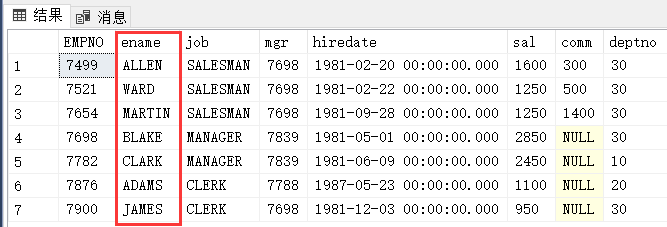


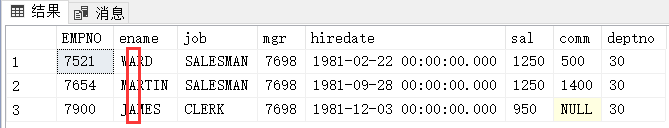


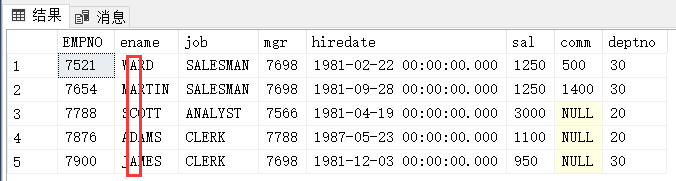
**8、模糊查询**

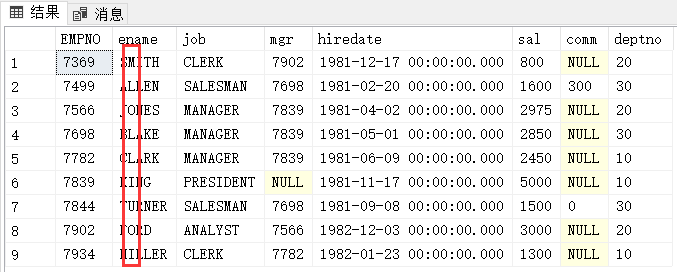
**注意：匹配的条件必须用单引号括起来，不能省略也不能改用双引号**

select \*
from emp
where ename like '%A%'; --包含A
select \* from emp where ename like '\_A%'; --第二个字母为A
select \* from emp where ename like '\_[A-G]%'; --第二个字母为A-G中的一个
select \* from emp where ename like '\_[^A-G]%'; --第二个字母是A-G之外的一个
select \* from emp where ename like '\_[A,C]%'; --第二个字母是A或C
select \* from emp where ename like '%\%%' escape '\'; --第二个字符为%
select \* from emp where ename like '%\\_%' escape '\'; --第二个字符为\_









**9、聚合函数**

**函数的分类**

**单行函数：每一行返回一个值**

**lower()**

**upper()**

select lower(ename) from emp; --将名字转化为小写查询

**多行函数：多行返回一个值 --聚合函数是多行函数**

**聚合函数的分类**

**max(）**

**min()**

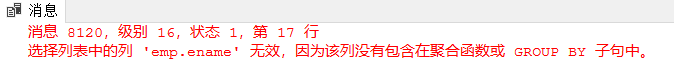
**avg()**

**count()**

**注意的问题：判断如下sql语句是否正确**

**select max(sal)** "最高工资"**,min(sal)** "最低工资"**,count(\*)** "员工人数" **from emp;**
select max(sal),lower(ename) from emp; -- 错误，单行函数和多行函数不能混用





select count(deptno) from emp; --返回重复且非空数据
select count(distinct deptno) from emp; --返回不重复且非空数据
select count(comm) from emp; --comm为空的记录不计数



**10、group by**

**11、having**

**12、连接查询**

**定义**

**分类**

**内连接**

**外连接**

**完全连接**

**交叉连接**

**自连接**

**联合**

**13、嵌套查询（oracle讲）**