

增删改查

结构：增create 删drop 改alter 查select

数据：增insert 删delete 改update 查select

alter table 能够实现的功能

alter都是基于表去修改

1.新增列

```
alter table 表名 add 列名 类型(长度)[约束&默认值];  
alter table student add City varchar2(20) not null;
```

2.修改列

```
alter table 表名 modify 列名 类型 [约束 默认值]; --修改表中的列  
alter table student modify City varchar(300) unique not null;  
alter table 表名 modify 列名 类型 null; --去掉列的非空约束  
alter table 表名 modify 列名 类型 not null; --给列添加非空约束  
alter table cjb modify xingming null
```

3.删除列

```
alter table 表名 drop column 列名; --删除表中的列  
alter table student drop column sname
```

4.新的列定义默认值

```
alter table student modify city default '北京';
```

5.重命名列

```
alter table 表名 rename column 旧列名 to 新列名;
```

6.将表更改为只读状态

```
alter table 表名 READ ONLY; --改为只读状态
```

```
alter table 表名 READ WRITE; --改回可编辑模式
```

7.添加约束

```
alter table 表名 add 表级约束语法; --给表添加表级约束
alter table bm2 add constraint bm2_zjys primary key(deptno);
alter table emp add constraint FK_DEPTNO foreign key (deptno) references DEPT
(DEPTNO);
alter table emp add constraint CK_sal check sal (sal>800)
--因非空约束不能使用表级约束语法，故添加非空约束需使用modify:
alter table bm2 modify loc varchar2(20) not null;
```

8.删除约束

```
alter table 表名 drop constraint 约束名; --删除一个约束
alter table bm2 drop constraint BM2_ZJYS
```

insert 数据插入

- 格式一：指定列插入数据，使用前提是不被插入的列可为空

```
insert into bm3(deptno,loc) values(50,'七星区');
insert into bm3(deptno,dname,loc) values(60,'a','a')
```

- 格式二：不指定列插入，默认所有列都需要值，如遇到不插入数据的列则用null值填充

```
insert into bm3 values(70,'t',null);
```

- 格式三：只插入select查询结果集的数据,注意查询的结果集结构要保证和表结构一致

```
insert into bm3(deptno,dname) select deptno,dname from scott.dept
insert into bm3 select 20,'z','z' from dual
insert into bm3 select deptno,ename,job from scott.emp
```

update 修改

```
update bm3 set deptno=80 where deptno=60
update bm3 set dname='销售部门',loc='八星区' where deptno=80
```

--把WARD的工资改成和SMITH一样

```
update yg2 set sal=(select sal from yg2 where ename='SMITH') where
ename='WARD'
```

delete 删除的是数据

delete、insert只能以行为单位进行操作

```
delete from yg2 where empno=7369
alter table yg2 drop column deptno
delete from yg2 where sal>(select sal from yg2 where empno=7499)
```

--删除表数据 DDL结构上的不能回滚 DML数据上的能回滚

drop table yg2--删除结构和数据，不能回滚

truncate table yg2--删除数据，不能回滚

delete from yg2 --删除数据，能回滚

速度：drop>truncate>delete

merge 融合

备份还原（仅一张表）

根据与源表联接的结果，对目标表执行插入、更新或删除操作。例如，根据在另一个表中找到的差异在一个表中插入、更新或删除行，可以对两个表进行同步。

merge也是数据上的属于DML

```
merge into 目标表
Using 备份表（源表）
On (values=values)
When matched then update 和 delete
When not matched then insert
```

--用源表sc1对目标表sc进行更新

merge into sc --目标表（需要更新的表）

using sc1 --源表（备份表）

on(sc.sno=sc1.sno **and** sc.cno=sc1.cno)

when matched then

update set sc.score=sc1.score--有值的时候去做更新

when not matched then

insert values(sc1.sno,sc1.cno,sc1.score)--没值的时候直接插入

update和delete同时带有where时，delete相当于继承上面的where条件再执行delete后面的条件

操作符

- 算术操作符: + - * / (加减乘除)
一般用与数字类型的运算, 和日期加减运算
- 连接操作符: || (把多个列合并成一个列)
- 逻辑操作符: and、or、not (与、或、非)
一般用于多个条件判断式的连接
优先级: not>and>or (可用小括号改变优先级)

--not 取反离自己最近的那个条件判断式

```
select * from emp where not sal>2000 or deptno=20 and job='MANAGER'
```

- 比较操作符: >、>=、<、<=、=、!=、<>、any、all、between..and..、like、in、is null
(都是放在where后面做条件比较)

```
select * from emp where sal>3000--数字的大于不包含本身
```

```
select * from emp where ename>'C'--字符的大于包含本身 (比的是数据的首字母A..Z)
```

--any: 只需满足与括号内任一内容比较的条件 (大于最小值, 小于最大值)

```
select * from emp where sal>any(select sal from emp where deptno=20) --求工资大于20号部门任一工资的员工数据
```

--all: 必须满足与括号内所有内容比较的条件 (大于最大值, 小于最小值)

```
select * from emp where sal>all(select sal from emp where deptno=20) --求工资大于所有20号部门的工资的员工数据
```

--in: 筛选出括号内的内容, 功能相当于 =any

```
select * from sc where sno in('s001','s002')
```

--is null:判断为空值

```
select * from emp where comm is null
```

```
select * from emp where comm is not null --非空
```

--between..and..包含边界值, 边界值得从小到大

```
select * from emp where sal between 950 and 3000
```

like 模糊搜索

在where子句中使用like关键字查询数据的方式也称为**字符串模式匹配**或**字符串模糊查询**, like关键字需要使用通配符在字符串内查找指定的模式, 所以要了解常用的通配符。

Like关键字可以使用以下两个通配符:

- %: 代表0或者多个字符
 - _: 代表一个且只能是一个字符
- 例如:

“k%”代表以k开头的任意长度的字符串；

“%k”代表以k结尾的任意长度的字符串；

“%k%”代表含有k字母的任意长度的字符串；

“_kk”代表3个字符长度且后面2个字符是kk的字符串

要查询的字符串中含有“%”或“_”时，可以使用转义（escape）关键字实现：

```
select * from EMP where ename like '%\_%' escape '\';
```

上述查询语句中使用了“\”为转义字符，即在“\”之后的“_”字符已不是通配符，而是他本来的含义，即下划线。因此该查询的结果为：含有“_”，任意长度字符的字符串。

以上规则可复合使用（见练习17-20）

上机练习5

--学生管理系统

--1.将c002课程的成绩增加5分

```
update sc set score=score+5 where cno='c002'
```

--2.将c001课程成绩小于80分的同学的成绩增加10分

```
update sc set score=score+10 where cno='c001' and score<80
```

--3.增加一个学生:学号's013', 姓名: '王麻子', 年龄: 28, 性别: 男

```
insert into student values('s013', '王麻子', 28, '男')
```

--4.创建一张和sc表相同的表，并将s001和s002学生的选课信息插入新表中

```
create table sc1 as  
select * from sc where sno='s001' or sno='s002'
```

--5.将所有c001课程成绩不及格的同学的分数改为60分

```
update sc set score=60 where cno='c001' and score<60
```

--6.删除“s002”同学的“c001”课程的信息记录

```
delete from sc where sno='s002' and cno='c001'
```

--重新插入“s002”同学的“c001”课程的信息记录

```
insert into sc values('s002', 'c001', 80.90);
```

--7.用sc1去更新sc，当学生和选修课程相同的时候，把sc1的分数更新到sc，不同的时候就插入到sc

```
merge into sc  
using sc1  
on(sc.sno=sc1.sno and sc.cno=sc1.cno)  
when matched then  
update set sc.score=sc1.score  
when not matched then
```

```
insert values(sc1.sno,sc1.cno,sc1.score)
```

--操作符练习

--1. 查询名字是BLAKE的人的编号, 名字, 工资

```
select empno,ename,sal from emp where ename='BLAKE'
```

--2. 查询编号是7782的员工的编号, 名字, 工资, 奖金

```
select empno,ename,sal,comm from emp where empno=7782
```

--3. 查询职位是销售(SALESMAN)的人的名字, 职位, 入职日期

```
select ename,job,hiredate from emp where job='SALESMAN'
```

--4. 查询部门是10的人的编号, 名字, 部门编号

```
select empno,ename,deptno from emp where deptno=10
```

--5. 查询工资大于1500, 并且小于2500的人的编号, 名字, 工资

```
select empno,ename,sal from emp where sal>1500 and sal<2500
```

--6. 查询工资小于2000的人的名字, 工资, 奖金

```
select ename,sal,comm from emp where sal<2000
```

--7. 求工作是CLERK的或者工资小于2000员工姓名, 工作, 工资

```
select ename,job,sal from emp where job='CLERK' or sal<2000
```

--8. 求工资小于800或者大于1500的员工姓名, 工作, 工资

```
select ename,job,sal from emp where sal<800 or sal>1500
```

--9. 求工作是CLERK, 并且工资小于950或者大于1500的员工姓名, 工作, 工资

```
select ename,job,sal from emp where job='CLERK' and (sal<950 or sal>1500)
```

--10. 求10号部门工资大于1000和20号部门工资大于1500的员工姓名

```
select ename from emp where deptno=10 and sal>1000 or deptno=20 and sal>1500
```

--11. 求工资在1500到2000之间的员工姓名

```
select ename from emp where sal between 1500 and 2000
```

--12. 求部门编号是10号, 20号, 30号中任意一个部门的员工姓名

```
select ename from emp where deptno in(10,20,30)
```

--13. 求姓名中包含'M'的员工姓名

```
select ename from emp where ename like '%M%'
```

--14. 求员工姓名第二位是'M'的员工姓名

```
select ename from emp where ename like '_M%'
```

--15. 求姓名中包含A和L的姓名

```
select ename from emp where ename like '%A%' and ename like '%L%'
```

--16. 求姓名是五位的员工信息

```
select * from emp where ename like '_____'
```

--17. 求姓名中第3位是%的员工姓名

```
select ename from emp where ename like '__\%%' escape '\'
```

--18. 求第一位是_, 倒数第二位也是_的员工姓名

```
select ename from emp where ename like '\_%' and ename like '%\__' escape '\'
```

--19. 求以_开头, 第三位也是下划线_, 第六位是%的员工姓名

```
select ename from emp where ename like '\__\___\%%' escape '\'
```

--20. 求名字中不包含M的员工姓名

```
select ename from emp where ename not like '%M%'
```