

# 视图

视图实际上是一个或多个表的预定义查询，视图的使用方法和表一样。

视图用于查看表的数据，不真实存储数据，只访问基表中的行。

一般视图都最好加只读（with read only）

```
--创建视图
--or replace 可写可不写，不写的话不可创建已有的视图名
create or replace view 视图名 as select * from 表名 with read only --附加只读状态，可写可不写

--查看视图
select * from 视图名

--删除视图
drop view 视图名;

--scott账号要使用system账号赋予权限才能创建视图
grant create any view to scott

/*当视图表结构和原表结构不一致的前提下，
原表插入数据的字段可为空，也可以通过对应视图表插入数据*/
--综合以上情况只要是视图表都最好加只读
create or replace view v_dept as select * from dept
insert into v_dept values(60,'a','a')
```

# 序列

序列是一个数据库项，用于生成一个整数序列，生成的序列用来填充数字型主键列。

数据库的对象之一，和表之间没有关联关系，用于数字型的主键列

```
--创建序列
create sequence 序列名
start with 开始数字
increment by 增量数
maxvalue 最大z值 --可写可不写
minvalue 最小值 --可写可不写

--例：创建一个序列，从1开始，每次增长1
create sequence seq_dept
start with 1
```

```

increment by 1

--调用序列
select seq_dept.nextval from dual--序列的下一个值，nextval第一次执行相当于激活序列
select seq_dept.currval from dual--查看序列的当前值
insert into dept values(seq_dept.nextval,'销售部门','桂林市')

--修改序列（只能修改增长值）
alter sequence seq_dept increment by 100
---没激活的序列直接修改增长值，那么该序列的初始值变成增长值

--删除序列
drop sequence seq_dept

```

## 循环序列

```

create sequence seq_aa
start with 5
minvalue 1
maxvalue 30
increment by 1

drop sequence seq_aa
select seq_aa.nextval from dual
alter sequence seq_aa cycle

```

## 事务

它是数据库在执行一系列操作时，保证所有的操作都正确完成，要么都执行，要么都不执行，保证数据的完整性

事务产生于会话的第一条DML语句（insert、delete、update）

必须具备以下四个属性，简称 **ACID** 属性：

- A：原子性（Atomicity）：事务是一个完整的操作。事务的各步操作是不可分的（原子的）；要么都执行，要么都不执行。
- C：一致性（Consistency）：一个查询的结果必须与数据库在查询开始时的状态保持一致（读不等待写，写不等待读）。
- I：隔离性（Isolation）：数据库中每一个用户的操作都是互不影响的，对于其他会话来说，未完成的（也就是未提交的）事务必须不可见。
- D：持久性（Durability）：事务一旦提交完成后，数据库就不可以丢失这个事务的结果，数据就永久的保存到数据库中。

事务采用隐性的方式，起始于session（会话窗口）的第一条DML语句，事务结束于：

- 1) COMMIT（提交）或ROLLBACK（回滚）
- 2) DDL语句被执行（提交）
- 3) DCL语句被执行（提交）
- 4) 用户退出SQLPLUS（正常退出是提交，非正常退出是回滚）
- 5) 服务器故障或系统崩溃（回滚）
- 6) shutdown immediate(回滚)

在一个事务里如果某个DML语句失败，之前其他任何DML语句将保持完好，而且不会提交！

## 锁表解锁

```
--1.查看被锁的表
select b.owner,b.object_name,a.session_id,a.locked_mode
from v$locked_object a,dba_objects b where b.object_id = a.object_id;
--2.查看那个用户那个进程造成死锁
select s.sid, q.sql_text
from v$sqltext q, v$session s where q.address = s.sql_address and s.sid = &sid
-- 这个&sid是第一步查询出来的
order by piece;--查看导致锁死的SQL
select b.username,b.sid,b.serial#,logon_time
from v$locked_object a,v$session b where a.session_id = b.sid order by
b.logon_time;
--3.杀掉进程
select 'alter system kill session ''' || sid || ',' || serial# || ''';'
"Deadlock" from v$session where sid in (select sid from v$lock where block =
1);
```

## 模拟死锁

```
--T1时刻
--事务1: session1执行
update emp set sal=sal*1.1 where deptno=10;
--事务2: session2执行
update emp set sal=3000 where deptno=20
--T2时刻
--事务1: session1执行
update emp set sal=3000 where deptno=20
--事务2: session2执行
update emp set sal=sal*1.1 where deptno=10;
--session3:system操作同上等待死锁
```

## 数据导入导出

数据的迁移和备份，导入导出，虚拟机cmd进行操作

exp 账号/密码@数据库名 file=文件存放路径及文件名 full=y --全库导出

命令行语句，不需要登录数据，只需要在命令行运行

按用户导出：

```
exp scott/oracle@orcl file=c:\test\exp_0514_scott.bak owner=scott
```

C:\Users\orastudy>exp scott/oracle@orcl file=C:\workspace\sql\exp\_0514\_scott.bak  
owner=scott

Export: Release 11.2.0.1.0 - Production on 星期三 5月 14 18:39:04 2025

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

连接到: Oracle Database 11g Release 11.2.0.1.0 - 64bit Production  
已导出 ZHS16GBK 字符集和 AL16UTF16 NCHAR 字符集

即将导出指定的用户...

. 正在导出 pre-schema 过程对象和操作  
. 正在导出用户 SCOTT 的外部函数库名  
. 导出 PUBLIC 类型同义词  
. 正在导出专用类型同义词  
. 正在导出用户 SCOTT 的对象类型定义

即将导出 SCOTT 的对象...

. 正在导出数据库链接  
. 正在导出序号  
. 正在导出簇定义  
. 即将导出 SCOTT 的表通过常规路径...

. 正在导出表	COURSE	导出了	10	行
. 正在导出表	DEPT	导出了	4	行
. 正在导出表	DEPT1	导出了	4	行
. 正在导出表	EMP	导出了	17	行
. 正在导出表	EMP2	导出了	18	行
. 正在导出表	GOODS	导出了	0	行
. 正在导出表	INCOME	导出了	0	行
. 正在导出表	OUTCOME	导出了	0	行
. 正在导出表	SALGRADE	导出了	5	行
. 正在导出表	SC	导出了	1583	行
. 正在导出表	SC1	导出了	4	行
. 正在导出表	SHUIGUO	导出了	21	行
. 正在导出表	STOCK	导出了	0	行
. 正在导出表	STUDENT	导出了	436	行
. 正在导出表	T	导出了	1	行
. 正在导出表	TEACHER	导出了	3	行
. 正在导出表	T_DIC	导出了	0	行

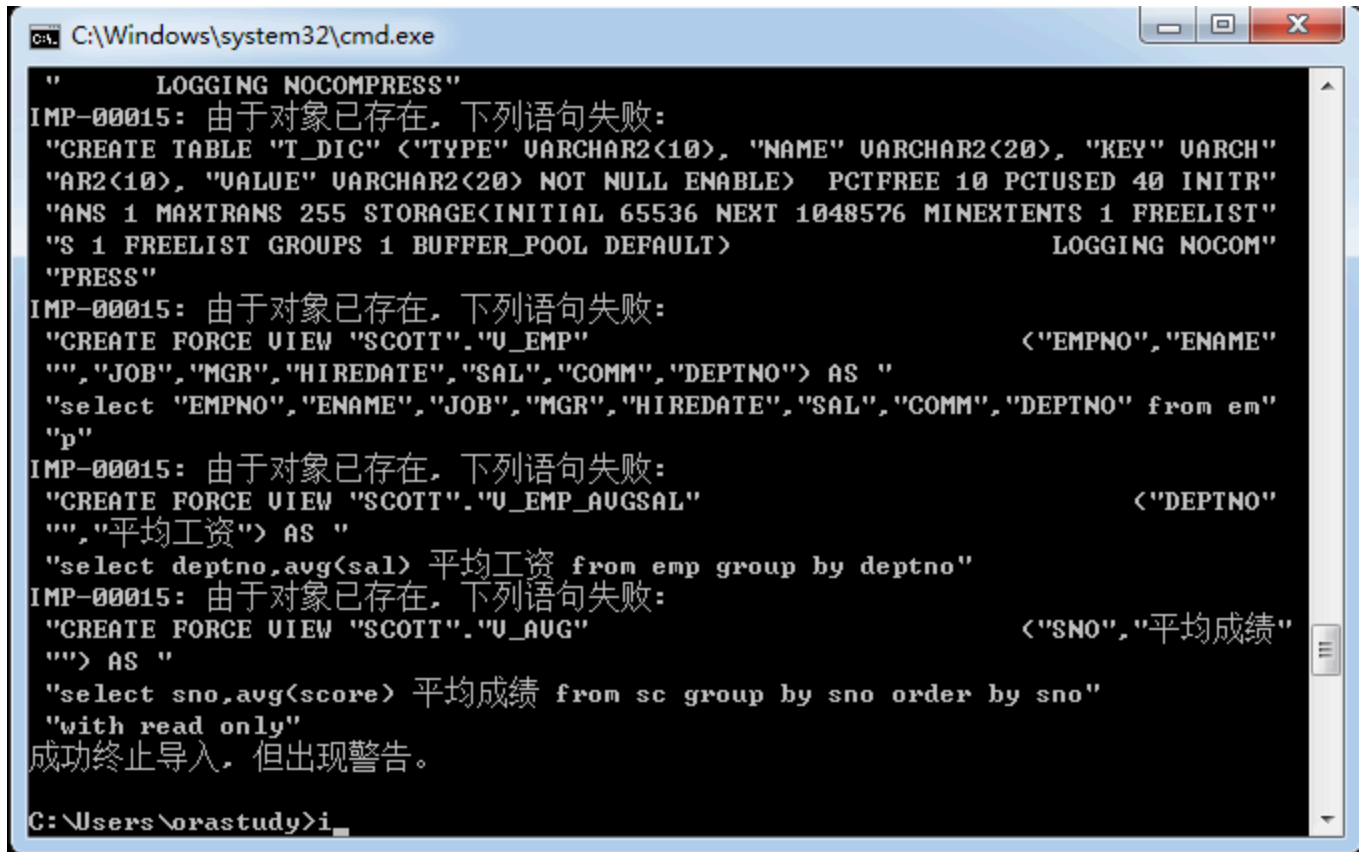
. 正在导出同义词  
. 正在导出视图  
. 正在导出存储过程  
. 正在导出运算符  
. 正在导出引用完整性约束条件  
. 正在导出触发器  
. 正在导出索引类型  
. 正在导出位图, 功能性索引和可扩展索引  
. 正在导出后期表活动  
. 正在导出实体化视图  
. 正在导出快照日志  
. 正在导出作业队列  
. 正在导出刷新组和子组  
. 正在导出维  
. 正在导出 post-schema 过程对象和操作  
. 正在导出统计信息

成功终止导出, 没有出现警告。

C:\Users\orastudy>

## 导入

```
imp scott/oracle@orcl file=c:\test\exp_0514_scott.bak full=y
```



```
C:\Windows\system32\cmd.exe
" LOGGING NOCOMPRESS"
IMP-00015: 由于对象已存在, 下列语句失败:
"CREATE TABLE "T_DIC" ("TYPE" VARCHAR2(10), "NAME" VARCHAR2(20), "KEY" VARCH
"AR2(10), "VALUE" VARCHAR2(20) NOT NULL ENABLE) PCTFREE 10 PCTUSED 40 INITR
"ANS 1 MAXTRANS 255 STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 FREELIST
"S 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT) LOGGING NOCOM
"PRESS"
IMP-00015: 由于对象已存在, 下列语句失败:
"CREATE FORCE VIEW "SCOTT"."V_EMP" <"EMPNO","ENAME"
"", "JOB", "MGR", "HIREDATE", "SAL", "COMM", "DEPTNO"> AS "
"select "EMPNO", "ENAME", "JOB", "MGR", "HIREDATE", "SAL", "COMM", "DEPTNO" from em
"p"
IMP-00015: 由于对象已存在, 下列语句失败:
"CREATE FORCE VIEW "SCOTT"."V_EMP_AVGSAL" <"DEPTNO"
"", "平均工资"> AS "
"select deptno, avg(sal) 平均工资 from emp group by deptno"
IMP-00015: 由于对象已存在, 下列语句失败:
"CREATE FORCE VIEW "SCOTT"."V_AVG" <"SNO", "平均成绩"
""> AS "
"select sno, avg(score) 平均成绩 from sc group by sno order by sno"
"with read only"
成功终止导入, 但出现警告。
C:\Users\orastudy>i_
```

## 作业练习

--上机练习17

--1. 创建一个视图, 查询出来emp表的所有数据, 并用insert into 给这个插入一条数据

```
create or replace view v_emp as select * from emp
insert into v_emp values(1111, 'HWZ', 'CLERK', null, trunc(sysdate), 888, null, 40)
delete from emp where empno=1111
```

--2. 创建一个查询emp表所有部门平均工资的视图, 并试图去修改这个视图的数据

```
create or replace view v_emp_avgsal as
select deptno, avg(sal) 平均工资 from emp group by deptno
select * from v_emp_avgsal
update v_emp_avgsal set 平均工资=round(平均工资) --非法
```

--3. 创建一个只读视图, 查询学生的平均成绩, 并试图去修改数据

```
create or replace view v_avg as
select sno, avg(score) 平均成绩 from sc group by sno order by sno
with read only
update v_avg set 平均成绩=round(平均成绩) --非法
```

--4.创建一个序列，从1开始，每次增长1

```
create sequence seq_emp
start with 1
increment by 1
```

--5.使用序列，把序列是8的插入到emp表中作为主键

```
select seq_emp.nextval from dual
select seq_emp.currval from dual
insert into emp
values(seq_emp.currval, 'hwz', 'CLERK', 1111, trunc(sysdate), 888, null, 40)
select * from emp
```

--6.修改序列增长值为10，把序列是38的插入到emp表中作为主键

```
alter sequence seq_emp increment by 10
select seq_emp.nextval from dual
insert into emp
values(seq_emp.currval, 'hwz2', 'CLERK', 1111, trunc(sysdate), 666, null, 40)
```

--上机练习18

--创建字典表 t\_dic

```
create table t_dic(
    type varchar2(10),
    name varchar2(20),
    key varchar2(10) unique,
    value varchar2(20) not null
);
```

--批量插入数据

```
select * from t_dic for update
```

--创建货物表 goods

```
create table goods(
    id varchar2(32) primary key,
    type varchar2(10) references t_dic(key),
    name varchar2(100),
    inprice number(10,2),
    outprice number(10,2),
    tjdw varchar2(10) references t_dic(key),
    qualitydate number(4) check(qualitydate>0)
);
```

--批量插入数据

```
select * from goods for update
```

--创建货物库存表 stock

```
create table stock(  
    id varchar2(32) primary key,  
    goodsid varchar2(32) references goods(id),  
    sl number(4),  
    intime date default sysdate,  
    productime date  
);
```

--创建进帐表 income

```
create table income(  
    id varchar2(32) primary key,  
    goodsid varchar2(32) references goods(id),  
    stockid varchar2(32) references stock(id),  
    sl number(4) check(sl>0),  
    discount number(4,1),  
    allprice number(8,1),  
    outtime date default sysdate  
);
```

--创建出帐表 outcome

```
create table outcome(  
    id varchar2(32) primary key,  
    goodsid varchar2(32) references goods(id),  
    stockid varchar2(32) references stock(id),  
    sl number(4,1) check(sl>0),  
    allprice number(8,1),  
    intime date default sysdate  
);
```

```
drop table outcome
```

```
select * from t_dic --字典表  
select * from goods --货物表  
select * from stock --库存表  
select * from income --进帐表  
select * from outcome --出帐表
```

--1. 在货物库存表stock的goodsid建立索引

```
create index idx_stock_goodsid on stock(goodsid)
```

--2. 在货物类型表goods的货物名称列添加唯一约束

```
alter table goods add constraint wyys unique(name)
```

--3. 将货物类型表的各种物品，在库存表插入一条记录，要求每个字段都填写完整，保证数量都在100以上。

```
insert into stock(id,goodsid,sl,intime,productime)  
select
```



```
g.id id,  
g.id goodsid,  
floor(dbms_random.value(100, 300)) sl, -- 随机生成100-300的数量  
trunc(sysdate) - 30 intime, -- 进货日期（一个月前）  
trunc(sysdate - dbms_random.value(30,90)) productime -- 生产日期（1-3个月  
前）
```

```
from goods g order by g.id
```

```
delete from stock --删除数据
```

--4. 查询出货物表中过期的所有物品

```
select * from goods g join stock s on s.id=g.id where  
productime+qualitydate<sysdate
```

--5. 查询货物表中10天之内过期的所有物品

```
select * from goods g join stock s on s.id=g.id where  
productime+qualitydate+10<sysdate
```

--6. 于当前时间新进货脉动50瓶，可口可乐50瓶，在出账表outcome和库存表stock插入相应记录

```
create sequence seq_stock --库存序列  
start with 12  
increment by 1
```

```
drop sequence seq_stock  
select seq_stock.nextval from dual  
select seq_stock.currval from dual
```

```
create sequence seq_outcome --出账序列  
start with 1  
increment by 1
```

```
drop sequence seq_outcome  
select seq_outcome.nextval from dual  
select seq_outcome.currval from dual
```

--库存表插入

```
insert into stock  
select seq_stock.nextval,id,50,trunc(sysdate),trunc(sysdate)-20  
from goods where id in (select id from goods where name = '脉动' or name='可口可乐')
```

--出账插入

```
insert into outcome --脉动  
select seq_outcome.nextval,id,seq_outcome.nextval,50,inprice*50,trunc(sysdate)  
from goods where id in (select id from goods where name = '脉动' or name='可口可乐')
```

```
delete from outcome
```

--7. 于当前时间卖出泰山一包，在进账表插入相应记录，在库存表修改相应记录

```
select * from income --进帐表
```

```
create sequence seq_intcome --出账序列
```

```
start with 1
```

```
increment by 1
```

```
drop sequence seq_income
```

```
select seq_income.nextval from dual
```

```
select seq_income.currval from dual
```

--进帐插入

```
insert into income(id, goodsid, stockid, sl, discount, allprice, outtime)
```

```
values(
```

```
    seq_income.nextval, -- id
```

```
    (select id from goods where name = '泰山'), -- 商品id
```

```
    (select s.id from stock s join goods g on s.goodsid = g.id where g.name =  
'泰山'), -- 库存id
```

```
    1, -- 销售数量1包
```

```
    1, -- 不打折
```

```
    (select outprice from goods where name = '泰山')*1, -- 总价
```

```
    sysdate -- 销售时间
```

```
);
```

--更新库存

```
update stock set sl = sl - 1 where goodsid = (select id from goods where name  
= '泰山') and sl > 0; -- 确保库存充足
```

--8. 查询库存内所有物品的总价，进货价计算

```
select s.*, outprice*sl 总价, inprice*sl 进货价 from stock s join goods g on  
g.id=s.id
```

--9. 查询酒水类物品的仓库保有量最大的物品id和总价

```
select * from (  
    select g.id, g.outprice * sum(s.sl) 总价  
    from stock s  
    join goods g on s.goodsid = g.id  
    where g.type = 'g02'  
    group by g.id, g.outprice  
    order by sum(s.sl) desc  
)
```

```
where rownum = 1;
```

--10. 查询价格大于5元低于10元的所有物品名称

```
select name from goods where outprice between 5 and 10
```

--11. 查询名称中有山字的所有物品

```
select name from goods where name like '%山%'
```

--12. 查询进货日期是今天的所有进货记录

```
select * from outcome where intime = trunc(sysdate)
```

--13. 查询名称长度最大的物品名称

```
select name from(  
    select name from goods order by length(name) desc  
) where rownum = 1;
```

--14. 汇总各类型物品的库存，其中烟和酒水单独统计，其他物品展示为其他

```
select case when  
    g.type in('g01','g02') then (g.name)  
    else '其他' end 名称, sum(s.sl) 库存  
from stock s join goods g on g.id=s.goodsid  
group by case when  
    g.type in('g01','g02') then (g.name)  
    else '其他' end  
order by 库存
```

--15. 更新可比克薯片的名称，将名称改为可比克薯片（袋装）

```
update goods set name='可比克薯片（袋装）' where name='可比克薯片'
```

--16. 将各类物品根据价格倒序排列展示

```
select * from goods order by outprice desc
```

--17. 查询出今天的销售额，毛利润

```
select sum(allprice) 销售额, sum(allprice-inprice*sl) 毛利润  
from income i join goods g on g.id=i.goodsid
```

--18. 根据每种物品类型（类型包括很多物品）昨天销售的累计数量，求出最大的数量的类型名称及总数量

```
with 昨日销售 as (  
    select g.type as 类型代码, sum(i.sl) as 销售总量  
    from income i join goods g on i.goodsid = g.id  
    where trunc(i.outtime) = trunc(sysdate) - 1 -- 昨天的销售记录  
    group by g.type  
)  
select d.name 类型名称, y.销售总量  
from 昨日销售 y join t_dic d on y.类型代码 = d.key  
where y.销售总量 = (select max(销售总量) from 昨日销售);
```

19. 查询出烟类，近10天的销售情况，展示出物品名称，卖出数量，总价格，毛利润

20. 根据天数分组，汇总近10天每天的销售情况，毛利润和总价格
21. 创建视图，展示当月销售物品中，分类统计各类物品的销售数量，总进价，总卖出价，毛利润
22. 在物品表里查出利润率最大的物品
23. 因为某人讲价，于当前时间，以8折卖出馒头50个，在进账表和库存表做出相应修改