# 视图的定义

视图(view)，也称虚表, 不占用物理空间，这个也是相对概念，因为视图本身的定义语句还是要存储在数据字典里的。视图只有逻辑定义。每次使用的时候,只是重新执行SQL。

  视图是从一个或多个实际表中获得的，这些表的数据存放在数据库中。那些用于产生视图的表叫做该视图的基表。一个视图也可以从另一个视图中产生。

  视图的定义存在数据库中，与此定义相关的数据并没有再存一份于数据库中。通过视图看到的数据存放在基表中。

  视图看上去非常象数据库的物理表，对它的操作同任何其它的表一样。当通过视图修改数据时，实际上是在改变基表中的数据；相反地，基表数据的改变也会自动反映在由基表产生的视图中。由于逻辑上的原因，有些Oracle视图可以修改对应的基表，有些则不能（仅仅能查询）。

  还有一种视图：物化视图（MATERIALIZED VIEW ），也称实体化视图，快照 （oracle 8 以前的说法） ，它是含有数据的，占用存储空间。

# 视图的作用

 1）提供各种数据表现形式, 可以使用各种不同的方式将基表的数据展现在用户面前, 以便符合用户的使用习惯。

2）隐藏数据的逻辑复杂性并简化查询语句, 多表查询语句一般是比较复杂的, 而且用户需要了解表之间的关系, 否则容易写错; 如果基于这样的查询语句创建一个视图, 用户就可以直接对这个视图进行"简单查询"而获得结果. 这样就隐藏了数据的复杂性并简化了查询语句.这也是oracle提供各种"数据字典视图"的原因之一。

3）执行某些必须使用视图的查询. 某些查询必须借助视图的帮助才能完成. 比如, 有些查询需要连接一个分组统计后的表和另一表, 这时就可以先基于分组统计的结果创建一个视图, 然后在查询中连接这个视图和另一个表就可以了。

4）提供某些安全性保证. 视图提供了一种可以控制的方式, 即可以让不同的用户看见不同的列, 而不允许访问那些敏感的列, 这样就可以保证敏感数据不被用户看见。

5）简化用户权限的管理. 可以将视图的权限授予用户, 而不必将基表中某些列的权限授予用户, 这样就简化了用户权限的定义。

# 创建视图

 create [ or replace ] [ force ] view [schema.]view\_name

                      [ (column1,column2,...) ]

                      as

                      select ...

                      [ with check option ]       [ constraint constraint\_name ]

                      [ with read only ];

说明：

1. or replace: 如果存在同名的视图, 则使用新视图"替代"已有的视图。
2. force: "强制"创建视图,不考虑基表是否存在,也不考虑是否具有使用基表的权限。
3. column1,column2,...：视图的列名, 列名的个数必须与select查询中列的个数相同; 如果select查询包含函数或表达式, 则必须为其定义列名.此时, 既可以用column1, column2指定列名, 也可以在select查询中指定列名。
4. with check option: 指定对视图执行的dml操作必须满足“视图子查询”的条件即,对通过视图进行的增删改操作进行"检查",要求增删改操作的数据, 必须是select查询所能查询到的数据,否则不允许操作并返回错误提示. 默认情况下, 在增删改之前"并不会检查"这些行是否能被select查询检索到。
5. with read only：创建的视图只能用于查询数据, 而不能用于更改数据。

## 1、创建简单视图

简单视图定义：是指基于单个表建立的，不包含任何函数、表达式和分组数据的视图。

[oracle@VM\_0\_15\_centos c]$ sqlplus / as sysdba

SQL\*Plus: Release 11.2.0.4.0 Production on 星期二 9月 24 11:44:21 2019

Copyright (c) 1982, 2013, Oracle. All rights reserved.

连接到:

Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production

With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> grant create view to scott;

授权成功。

SQL> connect scott/tiger;

已连接。

SQL> create view v\_emp as select empno,ename,job,hiredate,deptno from emp;

SQL> desc v\_emp

 Name                                      Null?    Type

 ----------------------------------------- -------- ----------------------------

 EMPNO                                     NOT NULL NUMBER(4)

 ENAME                                              VARCHAR2(10)

 JOB                                                VARCHAR2(9)

 HIREDATE                                           DATE

 DEPTNO                                             NUMBER(2)

SQL> select \* from v\_emp where deptno=10;

     EMPNO ENAME      JOB       HIREDATE      DEPTNO

----------  ----------     ---------    ---------

      7782 CLARK      MANAGER    09-JUN-81          10

      7839 KING       PRESIDENT   17-NOV-81         10

      7934 MILLER     CLERK       23-JAN-82          10

对简单视图进行DML操作

SQL> insert into v\_emp values(1,'a','aa','05-JUN-88',10);

SQL> update v\_emp set ename='cc' where ename='KING';

SQL> delete v\_emp where ename='cc';

SQL> select \* from v\_emp where deptno=10;

     EMPNO ENAME      JOB       HIREDATE      DEPTNO

---------- ---------- --------- --------- ----------

      7782 CLARK      MANAGER   09-JUN-81         10

      7934 MILLER     CLERK     23-JAN-82         10

         1 a          aa        05-JUN-88         10

基表也发生了相应的更改

SQL> select empno,ename,job,hiredate,deptno from emp where deptno=10;

     EMPNO ENAME      JOB       HIREDATE      DEPTNO

---------- ---------- --------- --------- ----------

      7782 CLARK      MANAGER   09-JUN-81         10

      7934 MILLER     CLERK     23-JAN-82         10

         1 a          aa        05-JUN-88         10

SQL> select object\_name,object\_type from user\_objects;

OBJECT\_NAME                    OBJECT\_TYPE

------------------------------ -------------------

V\_EMP                         VIEW

SALGRADE                       TABLE

BONUS                          TABLE

PK\_EMP                         INDEX

EMP                            TABLE

DEPT                           TABLE

PK\_DEPT                        INDEX

7 rows selected.

创建只读视图

SQL> create view v\_emp\_readonly as select empno,ename,job,hiredate,deptno from emp with read only;

SQL> select \* from v\_emp\_readonly where deptno=10;

     EMPNO ENAME      JOB       HIREDATE      DEPTNO

---------- ---------- --------- --------- ----------

      7782 CLARK      MANAGER   09-JUN-81         10

      7934 MILLER     CLERK     23-JAN-82         10

         1 a          aa        05-JUN-88         10

只能查询，无法进行更改

SQL> delete v\_emp\_readonly where empno=1;

delete v\_emp\_readonly where empno=1

       \*

ERROR at line 1:

ORA-42399: cannot perform a DML operation on a read-only view

更新基表，只读视图也发生了相应的更改

SQL> update emp set empno=2 where ename='a';

SQL> select \* from v\_emp\_readonly where deptno=10;

     EMPNO ENAME      JOB       HIREDATE      DEPTNO

---------- ---------- --------- --------- ----------

      7782 CLARK      MANAGER   09-JUN-81         10

      7934 MILLER     CLERK     23-JAN-82         10

         2 a          aa        05-JUN-88         10

创建检查约束视图with check option

Sql>create view v\_emp\_check as select empno,ename,job,hiredate,deptno from emp where deptno=10 with check option;

SQL> insert into v\_emp\_check values('2','c','cc','02-JAN-55',10);

SQL> insert into v\_emp\_check values('3','d','dd','02-JAN-65',20);

insert into v\_emp\_check values('3','d','dd','02-JAN-65',20)

            \*

ERROR at line 1:

ORA-01402: view WITH CHECK OPTION where-clause violation

创建检查视图：对通过视图进行的增删改操作进行检查，要求增删改操作的数据必须是select查询所能查询到的数据

20号部门不在查询范围内，违反检查约束,所以无法插入；

SQL> delete v\_emp\_check where empno=2;

1 row deleted.

--------所删除的数据在查询范围内，不违反检查约束

## 2、创建连接视图

### 2.1、连接视图定义

是指基于多个表所创建的视图，即，定义视图的查询是一个连接查询。 主要目的是为了简化连接查询；

### 2.2、创建连接视图

示例1： 查询部门编号为10和30的部门及雇员信息

SQL> create view v\_dept\_emp

  2  as select a.deptno,a.dname,a.loc,b.empno,b.ename,b.sal from dept a,emp b where a.deptno=b.deptno and a.deptno in(10,30);

View created.

SQL> select \* from v\_dept\_emp;

    DEPTNO DNAME          LOC             EMPNO ENAME             SAL

---------- -------------- ------------- ---------- ---------- ----------

        30 SALES          CHICAGO             7499 ALLEN            1600

        30 SALES          CHICAGO             7521 WARD             1250

        30 SALES          CHICAGO             7654 MARTIN           1250

        30 SALES          CHICAGO             7698 BLAKE            2850

        10 ACCOUNTING    NEW YORK           7782 CLARK            2450

        30 SALES          CHICAGO             7844 TURNER           1500

        30 SALES          CHICAGO             7900 JAMES             950

        10 ACCOUNTING    NEW YORK           7934 MILLER           1300

        10 ACCOUNTING     NEW YORK               1 a

9 rows selected.

### 2.3、连接视图上的DML操作

SQL> insert into v\_dept\_emp values(10,'aaa','aaaa',22,'a',5000);

insert into v\_dept\_emp values(10,'aaa','aaaa',22,'a',5000)

\*

ERROR at line 1:

ORA-01779: cannot modify a column which maps to a non key-preserved table

在视图上进行的所有DML操作，最终都会在基表上完成；

select 视图没有什么限制，但insert/delete/update有一些限制；

### 2.4、键值保存表

如果连接视图中的一个“基表的键”(主键、唯一键)在它的视图中仍然存在，并且“基表的键”仍然是“连接视图中的键”(主键、唯一键)；即，某列在基表中是主键|唯一键，在视图中仍然是主键或唯一键，则称这个基表为“键值保存表”。

 一般地，由主外键关系的2个表组成的连接视图，外键表就是键值保存表，而主键表不是。

### 2.5、连接视图的更新准则

 一、一般准则

1）任何DML操作，只能对视图中的键值保存表进行更新, 即，“不能通过连接视图修    改多个基表”;

2）在DML操作中，“只能使用连接视图定义过的列”;

3）“自连接视图”的所有列都是可更新(增删改)的

 二、insert准则

1. 在insert语句中不能使用“非键值保存表”中的列(包括“连接列”)；
2. 执行insert操作的视图，至少应该“包含”键值保存表中所有设置了约束的列；
3. 如果在定义连接视图时使用了WITH CHECK OPTION 选项，则“不能”针对连接视 图执行insert操作

 三、update准则

1. 键值保存表中的列是可以更新的；
2. 如果在定义连接视图时使用了WITH CHECK OPTION 选项，则连接视图中的连接列(一般就是“共有列”)和基表中的“其他共有列”是“不可”更新的，连接列和共有列之外的 其他列是“可以”更新的

 四：delete准则

如果在定义连接视图时使用了WITH CHECK OPTION 选项，依然“可以”针对连接视图执行delete操作 。

### 2.6、可更新连接视图

如果创建连接视图的select查询“不包含”如下结构，并且遵守连接视图的“更新准则”，则这样的连接视图是“可更新”的：

1. 集合运算符(union,intersect,minus)
2. DISTINCT关键字
3. GROUP BY，ORDER BY，CONNECT BY或START WITH子句
4. 子查询
5. 分组函数
6. 需要更新的列不是由“列表达式”定义的
7. 基表中所有NOT NULL列均属于该视图

## 3、创建复杂视图

复杂视图定义：是指包含函数、表达式、或分组数据的视图。主要目的是为了简化查询。主要用于执行查询操作，并不用于执行DML操作。

注意：当视图的select查询中包含函数或表达式时，必须为其定义列别名。

 示例1：查询目前每个岗位的平均工资、工资总和、最高工资和最低工资。

SQL> create view v\_emp\_job\_sal(job,avgsal,sumsal,maxsal,minsal)

  2  as select job,avg(sal),sum(sal),max(sal),min(sal) from emp group by job;

View created.

SQL> select \* from v\_emp\_job\_sal;

JOB           AVGSAL     SUMSAL     MAXSAL     MINSAL

--------- ---------- ---------- ---------- ----------

CLERK         1037.5       4150       1300        800

SALESMAN        1400       5600       1600       1250

aa

MANAGER   2758.33333       8275       2975       2450

ANALYST         3000       6000       3000       3000

## 4 、强制创建视图

强制视图定义：正常情况下，如果基表不存在，创建视图就会失败。但是可以使用force选项强制创建视图(前提：创建视图的语句没有语法错误！)，此时该视图处于失效状态。

SQL> create force view v\_test\_tab

  2  as select c1,c2 from test\_tab;

Warning: View created with compilation errors. 警告: 创建的视图带有编译错误。

SQL> select \* from v\_test\_tab;

select \* from v\_test\_tab

              \*

ERROR at line 1:

ORA-04063: view "SCOTT.V\_TEST\_TAB" has errors

SQL> select object\_name,status from user\_objects where object\_name='V\_TEST\_TAB';

 OBJECT\_NAME                    STATUS

------------------------------ -------

V\_TEST\_TAB                    INVALID ---------视图状态为不可用

SQL> create table test\_tab(c1 number(9) primary key,c2 varchar2(20),c3 varchar2(30));

 Table created.

SQL> select \* from v\_test\_tab;

 no rows selected

SQL> select object\_name,status from user\_objects where object\_name='V\_TEST\_TAB';

OBJECT\_NAME                    STATUS

------------------------------ -------

V\_TEST\_TAB                    VALID  --------------视图状态为可用

# 四 更改视图

在对视图进行更改(或重定义)之前，需要考虑如下几个问题：

1. 由于视图只是一个虚表，其中没有数据，所以更改视图只是改变数据字典中对该视图的定义信息，视图的所有基础对象都不会受到任何影响。
2. 更改视图之后，依赖于该视图的所有视图和PL/SQL程序都将变为INVALID(失效)状态。
3. 如果以前的视图中具有with check option选项，但是重定义时没有使用该选项，则以前的此选项将自动删除。

## 1、更改视图的定义

 方法——执行create or replace view语句。这种方法代替了先删除(“权限也将随之删除”)后创建的方法，会保留视图上的权限，但与该视图相关的存储过程和视图会失效。

示例1：

将视图改为改为只读

SQL> create or replace view v\_emp

  2  as

  3  select empno,ename,job,hiredate,deptno from emp with read only;

View created.

## 2、视图的重新编译

   语法：alter view 视图名 compile;  
  作用：当视图依赖的基表改变后，视图会“失效”。为了确保这种改变“不影响”视图和依赖于该视图的其他对象，应该使用 alter view 语句“明确的重新编译”该视图，从而在运行视图前发现重新编译的错误。视图被重新编译后，若发现错误，则依赖该视图的对象也会失效；若没有错误，视图会变为“有效”。  
  权限：为了重新编译其他模式中的视图，必须拥有alter any table系统权限。  
  注意：当访问基表改变后的视图时，oracle会“自动重新编译”这些视图。

 示例1：

SQL> select last\_ddl\_time,object\_name,status from user\_objects where object\_name='V\_TEST\_TAB';

LAST\_DDL\_ OBJECT\_NAME                    STATUS

--------- ------------------------------ -------

23-AUG-14 V\_TEST\_TAB                    VALID     -----视图的状态:有效

SQL> alter table test\_tab modify(c2 varchar2(30)); ——修改基表，c2列的长度

SQL> select last\_ddl\_time,object\_name,status from user\_objects where object\_name='V\_TEST\_TAB';

LAST\_DDL\_ OBJECT\_NAME                    STATUS

--------- ------------------------------ -------

23-AUG-14 V\_TEST\_TAB                    INVALID     ——视图的状态:失效

SQL> alter view v\_test\_tab compile;                 ---重新编译

SQL> select last\_ddl\_time,object\_name,status from user\_objects where object\_name='V\_TEST\_TAB';

LAST\_DDL\_ OBJECT\_NAME                    STATUS

--------- ------------------------------ -------

23-AUG-14 V\_TEST\_TAB                    VALID         ---视图有效

思考：若上述代码修改的不是列长，而是表名，结果又会如何？

<警告：更改的视图带有编译错误；视图状态：失效>

# 五 删除视图

可以删除当前模式中的任何视图；

如果要删除其他模式中的视图，必须拥有DROP ANY VIEW系统权限；

视图被删除后，该视图的定义会从词典中被删除，并且在该视图上授予的“权限”也将被删除。

 视图被删除后，其他引用该视图的视图及存储过程等都会失效。

示例1：drop view v\_test\_tab;

# 六 查看视图

使用数据字典视图  
1 dba\_views——DBA视图描述数据库中的所有视图  
2 all\_views——ALL视图描述用户“可访问的”视图  
3 user\_views——USER视图描述“用户拥有的”视图   
4 dba\_tab\_columns——DBA视图描述数据库中的所有视图的列(或表的列)  
5 all\_tab\_columns——ALL视图描述用户“可访问的”视图的列(或表的列)  
6 user\_tab\_columns——USER视图描述“用户拥有的”视图的列(或表的列)   
示例1：查询当前方案中所有视图的信息

# 七 在视图上执行DML操作的步骤和原理

第一步：将针对视图的SQL语句与视图的定义语句(保存在数据字典中)“合并”成一条SQL语句

第二步：在内存结构的共享SQL区中“解析”(并优化)合并后的SQL语

第三步：“执行”SQL语句

示例：假设视图v\_emp的定义语句如下：

create view v\_emp

as

select empno,ename,loc from employees emp,departments dept

where

  emp.deptno=dept.deptno and dept.deptno=10;

当用户执行如下查询语句时：

select ename from v\_emp where empno=9876;

oracle将把这条SQL语句与视图定义语句“合并”成如下查询语句：

select ename from employees emp,departments dept

where

 emp.deptno=dept.deptno and dept.deptno=10 and empno=9876;

然后，解析(并优化)合并后的查询语句，并执行查询语句;

T\_LOCALOBTMIND 七天之内 当前表

T\_LOCALOBTMIND\_HIS 七天之前 历史表

T\_LOCALOBTMIND \_ARH\_2018 归档表

T\_LOCALOBTMIND \_ARH\_2017 归档表

T\_LOCALOBTMIND \_ARH\_2016 归档表