Oracle的视图和同义词

单世民

视图的含义

- 视图是对根据预定义的选择标准由一个或多个行的集合建立起来的动态表的静态定义。

- 使用视图的原因很多，比如：集中用户使用的数据、隐藏数据的复杂性、简化权限管理以及为向其他应用程序输出而重新组织数据等等

视图的种类

- Oracle中视图的种类
  - 关系视图
  - 内嵌视图
  - 对象视图
  - 物化视图

关系视图

- 关系视图就是经过存储的查询，可以将其输出看作是一个表。它就是基于关系数据的存储对象。可以将视图看作是虚拟表，可以像查询表一样地查询视图

创建视图

- 使用CREATE VIEW语句创建视图的部分语法形式如下：

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW
[USER.] view_name
[column1[, column2]…]
AS query
[WITH CHECK OPTION [CONSTRAINT constraint_name]
[WITH READ ONLY]
```

视图的删除

```
DROP VIEW view_name;
```

- 删除视图之后，如果再次使用它时，系统就会报出对象不存在的错误。视图被删除之后，与该视图有关的权限也随之删除。即使新建创建了该名称的视图，视图的权限也需要重新赋予。

获取有关视图的信息

- 检索视图的定义

```
SELECT * FROM user_views WHERE view_name='MY_VIEW';
```

WITH CHECK OPTION选项

- 此选项的作用可以这样理解：
  使用此选项后，通过视图进行的修改，必须也能通过该视图看到修改后的结果。比如INSERT操作，那么加入的记录在刷新视图后必须可以看到；如果修改，修改完的结果也必须能通过该视图看到；如果删除，当然只能删除视图里有显示的记录。

- 常见的错误在于在定义视图的字查询语句中使用了选择条件，而在插入时忽视了此选择条件

WITH READ ONLY选项

- 可以在创建视图时带上WITH READ ONLY 选项，以确保无DML操作发生。
- 任何对带只读约束的视图进行的插入或修改行的尝试，Oracle都将提示错误

对视图的验证

- 当创建视图时，Oracle将会验证视图的有效性。以后，改变基本表的特性有可能导致视图无效。例如，以下操作将导致视图无效：
  - 改变列的名称，或从基本表或视图中完全删除列。
  - 删除构建视图的基本表或视图。
  - 改变基本表或视图，使其无效，这样将导致视图变得无效。

- 为了解决上述问题，可以使用alter view view_name compile命令重新编译视图，或者使用create or replace view命令重新创建视图，或者修正视图所基于的基本表或视图。

通过视图的数据更新及删除

- 如果视图包含了下面的内容，那么不能通过视图删除基本表中的数据：
  - 分组函数，例如sum、avg、min、max等；
  - group by子句；
  - distinct关键字；
  - 包含了表达式；
  - rownum伪列。
- 在插入数据时，除了需要满足上面提到的条件之外，还需要保证那些没有包含在视图定义中的基表的列必须允许空值。
- 如果在视图定义中包含了with check option子句，那么对视图的修改除了前面的那些原则之外，还必须满足指定的约束条件。

内嵌视图

- 内嵌视图（ inline view，内建视图，内联视图）
  内嵌视图是一个带有别名(或相关名)的、可以在SQL语句中使用的子查询。

- 内嵌视图不是模式对象。从根本上来讲，内嵌视图就是嵌入到父查询中的查询，能够在任何可以使用表名称的地方使用。内嵌视图可以出现在select语句的from子句中，也可以出现在insert into、update、delete from等语句中。内嵌视图是临时的，它只存在于父查询的运行期间，但是它可以让开发人员有能力在整个查询的任何部分中使用视图结果。

内嵌视图

- 示例：
  得到每个单位具有最高工资的员工信息

```
SELECT a.EName, a.ESal, DName, b.maxsal
FROM Employee a,
   (SELECT DID, MAX(ESal) maxsal
          FROM Employee GROUP BY DID) b,
   Department dept
WHERE a.DID = b.DID
AND a.ESal = b.maxsal AND b.DID=dept.DID;
```

对象视图

- Oracle的对象—— 关系技术是构建在关系结构上的对象层。在对象层以下数据需要存储在关系表中，但是Oracle允许用户将这些数据封装在对象类型中。所以，为了获取特定雇员的信息，用户可以不再考虑从emp表的某些列中进行选择，而是可以考虑选择由存储在数据库中的对象模型化的单独客户。
- 如果已经创建了应用，而且只能够使用用户模式中的关系表，如果希望在不重新设计用户模式并且重新构建用户应用的前提下，利用Oracle中的对象—— 关系特性，那么应该怎么办呢？这里可以提供一个解决方案：对象视图。
- 用户可以基于用户的对象类型创建对象视图，然后可以像平常一样通过这些视图查询并且修改数据。

对象视图

```
create type emp_obj is object(
    id number(38),
    ename varchar2(30),
    email varchar2(30)
);
```

```
create table emp of emp_obj;
```

```
CREATE VIEW emp_email OF emp_obj
WITH OBJECT OID(id)
AS
SELECT emp.id,emp.email FROM emp;
```

物化视图

- 在Oracle 8i以前的版本中，这些对象被称为快照。从Oracle 8i以后，这些对象被重命名为物化视图，并且经过了加强，可以支持查询重写、刷新或提交。这些视图可以用于从数据仓库到分布式移动计算的各种任务和各种环境。
- 从本质上来看，物化视图就是在数据库中存储的查询结果。与在运行时确定结果的关系视图不同，物化视图的结果会预先计算并且存储。由于要存储结果，所以物化视图要占用空间，但是不会延缓用户对视图的使用。当正在查询大规模数据时，物化视图能够极大地增强用户应用的性能。

物化视图

```
create materialized view vw_material
as
select empno,count(*) total_emp
from emp
group by empno;
```

TOP N分析

- Top-N 查询是寻找一列中的n个最大或最小值
- Top-N 分析查询语句的两个基本构成：
  - 在子查询中使用order by字句来进行排序
  - 在主查询中限制结果集能显示的行数(使用ROWNUM 伪列)
- 示例：得到最先参加工作的3个员工的名单

```
SELECT rownum as SENIOR,tempemp.EName, tempemp.EHiredate
FROM (SELECT EName,EHiredate FROM Employee
      ORDER BY EHiredate) tempemp
WHERE rownum <= 3;
```

同义词

- 创建同义词可以简化对象访问(对象的另一个名字)。同义词能够：
  - 使用户更加方便的引用另一个用户的表
  - 使对象名字变短
- 创建一个同义词可以除去对象名必须带的方案限制 同义词可以用来替换表名、视图名、序列名和过程名或其它对象名(不包括包－package)。
- 一个私有同义词名字对于同一个用户必须与其所有的其它对象不同

使用同义词

- 创建同义词

```
CREATE [PUBLIC] SYNONYM synonym_name FOR object_name;
```

- 修改同义词

```
DROP [PUBLIC] SYNONYM synonym_name;
```

小结

- 视图
  - 关系视图
  - 内嵌视图
  - 对象视图
  - 物化视图
  - TOP-N分析
- 同义词

The End