Views

Mirunalini.P

SSNCE

April 5, 2022

Table of Contents

Session Objective

- To learn about views
- To learn about updatable view condition
- To learn to drop the view

views

- Tables and views are database objects
- **Table** – basic unit of storage; composed of rows and columns
- **View** – logically represents subsets of data from one or more tables

What is a View?

- A view is a logical (virtual) table based on a table or another view.
- A view contains no data of its own.
- The tables on which a view is based are called **base tables**.

Why use Views?

- Views restrict access to the data because the view can display selective columns from the table – **data hiding**
- Used to make simple queries to retrieve the results of complicated queries – **simplified query formulations**
- Views can provide logical data independence

Creating a View

Embed a subquery within the CREATE VIEW statement.

```
CREATE [OR REPLACE] VIEW view_name
[(alias[, alias]...)]
AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
```

The subquery can contain complex SELECT syntax.

Creating a View

Create a view, that contains details of employees in department 50

```
CREATE VIEW empvu
AS SELECT employee_id,last_name,salary
FROM employees
WHERE department_id=50;
```

The structure of view can be viewd by DESC .

Guidelines – Creating a View

- The subquery that defines a view can contain complex SELECT syntax – Joins, Groups and subqueries.
- The subquery that defines the view cannot contain an ORDER BY clause.
- Use the OR REPLACE option to change the definition of the view.

Creating a View using column alias

The number of aliases must match the number of attributes selected in the subquery

```
CREATE VIEW salvu (ID_NUMBER, NAME, ANN_SAL)
AS SELECT employee_id, last_name, salary*12
FROM employees
WHERE department_id=50;
```

Example Relational Scheme

```
student(rollNo, name, degree, year, sex, deptNo, advis
department(deptId, name, hod, phone)
professor(empId, name, sex, startYear, deptNo, phone)
course(courseId, cname, credits, deptNo)
```

Creating a View – using complex SELECT clause

Create a view which contains name, employeeId and phone number of professors who joined before 2005, and working for CSE dept.

```
CREATE VIEW profBef05 (EMP_ID, NAME, CONTACT)
AS(SELECT empId, p.name, phone
FROM professor p, department d
WHERE p.deptNo = d.deptId
AND d.name = 'CSE'
AND p.startYear < 2005 );
```

Querying on Views

- Querying is allowed in views as like in base table.
- Obtain names of professors in CSE dept, who joined before 2005 and whose name starts with ' Ram'

```
SELECT name
FROM profBef05
WHERE name LIKE 'Ram%';
```

Operations on Views

- View definition is stored in data dictionary table.
- Update operations are usually restricted because
  - Updates on a view may modify many base tables.
  - There may not be a unique way of updating the base tables to reflect the updates on view.
  - View may contain some aggregate values.
  - Ambiguity where primary key of a base table is not included in view definition.

Example Relational Scheme

Dept (DEPTNO, DNAME, LOC)
Emp(EMPNO, ENAME, JOB , MGR, SAL , COMM, DEPTNO)

Restrictions on Updating Views

- Updates on views defined with GROUP BY clause and aggregate functions is not permitted, as a tuple in view will not have a corresponding tuple in base relation.
- Create a view Dept_Totsalary which contains total salary earned by the employees in the dept.

```
CREATE OR REPLACE VIEW Dept_Totsalary (DEPTNO,
TOT_salary)
AS(SELECT deptNo, SUM(salary)
FROM dept
GROUP BY deptNo);
```

1

Restrictions on Updating Views

- Updates on views which do not include primary key of base table, are also not permitted.
- Create a view Emp_job with emp name and Jobid

```
CREATE VIEW Emp_job
AS (SELECT name, job FROM emp);
```

Managing Views

```
CREATE VIEW emp_dept AS
        SELECT emp.empno, emp.ename,
        emp.deptno, emp.sal, dept.dname, dept.loc
        FROM emp, dept
        WHERE emp.deptno = dept.deptno
        AND dept.loc IN ('DALLAS', 'NEW YORK', 'BOSTON')
```

Key Preserved Tables

- The concept of a key-preserved table places the restrictions on modifying join views.
- A table is key-preserved if every key of the table can also be a key of the result of the join.

Emp_dept View

EMPNO	ENAME	DEPTNO	SAL	DNAME	LOC
107	Diana	60	5000	Hardware	Boston
124	Kevin	50	5800	Mainte	Boston
141	Trenna	60	5900	Hardware	Boston
144	Peter	10	4026	HR	Dallas
149	Eleni	20	10500	Fin	Dallas
174	Ellen	20	11000	Fin	Dallas
176	Jona	30	8600	Devp	Newyork
200	Jenni	40	4400	testing	Newyork
143	Randall	10	4187	HR	Dallas
142	Curtis	50	3100	Mainte	Boston

- emp is a key-preserved table, because empno is a key of the emp table and key of the join.
- But dept is not a key-preserved table, because although deptno is a key of the dept table, it is not a key of the join

Update Operations on the View

- UPDATE statement that successfully modifies the emp_dept view:
- **UPDATE emp_dept SET sal = sal * 1.10**
  WHERE deptno = 10;
- The following UPDATE statement would be disallowed on the emp_dept view:
- **UPDATE emp_dept SET loc = 'BOSTON'**
  WHERE ename = 'SMITH';
- This statement fails with an error (ORA-01779 cannot modify a column which maps to a non key-preserved table),it attempts to modify non key-preserved table in the emp_dept view.
- All updatable columns of a join view must map to columns of a key-preserved table.

View with Check Option

- If the view is defined using the WITH CHECK OPTION clause, then all join columns and all columns taken from tables that are referenced more than once in the view are not modifiable.
- **UPDATE emp_dept SET deptno = 60 WHERE ename = 'Diana' with check option**;
- The statement fails because it is trying to update a join column

Delete Statement on views

- Deletion from a join view is possible if one and only one key-preserved table in the join is present
- The following DELETE statement works on the emp_dept view:
- **DELETE FROM emp_dept WHERE ename = 'SMITH';**
- The delete statement is legal because it can be translated to a DELETE operation on the base emp table
- If a view is defined using the WITH CHECK OPTION clause and the key-preserved table is repeated, rows cannot be deleted from such a view.

Delete Statement on views

- If a view is defined using the WITH CHECK OPTION clause and the key-preserved table is repeated, rows cannot be deleted from such a view.

```
CREATE VIEW emp_mgr AS
  SELECT e1.ename, e2.ename mname
  FROM emp e1, emp e2
  WHERE e1.mgr = e2.empno
  WITH CHECK OPTION;
```

Insert Statement on views

- Insert statement works because only one key-preserved base table is being modified
- Insert will happen only if it statisfies the constraints of the base table

1. INSERT INTO emp_dept (ename, empno, deptno)
   VALUES ('KURODA', 9010, 40);

2. INSERT INTO emp_dept (ename, empno, deptno)
   VALUES ('KURODA', 9010, 77);
   (Foriegn key violation)

3. INSERT INTO emp_dept (empno, ename, loc)
   VALUES (9010, 'KURODA', 'BOSTON');
   (Updation through non–key preserved table)

- The join view is defined using the WITH CHECK OPTION clause, then insertion cannot be performed.

User Updatable Operations

To check whether a view is modifiable, then you can select from the USER_UPDATABLE_COLUMNS view to see if it is.

```
SELECT COLUMN_NAME, UPDATABLE
        FROM USER_UPDATABLE_COLUMNS
        WHERE TABLE_NAME = 'EMP_DEPT';
```

Allowed Updates on Views

- The view defining table expression is a simple select expression and contains JOINs of key preserved table.
- The SELECT clause of select expression does not contain the DISTINCT keyword.
- That table reference identifies either a base table or a view that satisfies conditions.
- The select expression does not include aggregate functions, a GROUP BY clause, not contains non-preserved key and with check options (join columns and all columns of repeated tables)

Dropping a view

- An existing view can be dropped by means of DROP VIEW
- Dropping view has no effect on the tables on which the view was based.
- Views or other applications based on deleted views become invalid.

syntax:

```
DROP VIEW view_name;
DROP VIEW emp_dept;
```

Summary

- A view provides a different way of looking at the data in one or more tables;
- it is a named specification of a result table.
- view can be used in an insert, update, or delete operation depends on its definition.
- When the column of a view is directly derived from the column of a base table, that view column inherits any constraints that apply to the table column

Reference

 Fundamentals of Database systems 7^{th} Edition by Ramez Elmasri.