Join Views

You can also create views that specify more than one base table or view in the `FROM` clause. These are called **join views**. The following statement creates the `division1_staff` view that joins data from the `emp` and `dept` tables:

```
CREATE VIEW division1_staff AS
      SELECT ename, empno, job, dname
      FROM emp, dept
      WHERE emp.deptno IN (10, 30)
         AND emp.deptno = dept.deptno;
```

An **updatable join view** is a join view where `UPDATE`, `INSERT`, and `DELETE` operations are allowed.

Updating a Join View

An updatable join view (also referred to as a ***modifiable join view***) is a view that contains more than one table in the top-level `FROM` clause of the `SELECT` statement, and is not restricted by the `WITH READ ONLY` clause.

The rules for updatable join views are shown in the following table. Views that meet these criteria are said to be inherently updatable.

Rule	***Description***
General Rule	Any `INSERT`, `UPDATE`, or `DELETE` operation on a join view can modify only one underlying base table at a time.
`UPDATE` Rule	All updatable columns of a join view must map to columns of a *key-preserved table*. If the view is defined with the `WITH CHECK OPTION` clause, then all join columns and all columns of repeated tables are not updatable.
`DELETE` Rule	Rows from a join view can be deleted as long as there is exactly one *key-preserved table* in the join.
`INSERT` Rule	An `INSERT` statement must not explicitly or implicitly refer to the columns of a non-key-preserved table. If the join view is defined with the `WITH CHECK OPTION` clause, `INSERT` statements are not permitted.

Key-Preserved Tables

```
CREATE TABLE dept (
      deptno        NUMBER(4) PRIMARY KEY,
      dname         VARCHAR2(14),
      loc           VARCHAR2(13));

CREATE TABLE emp (
      empno         NUMBER(4) PRIMARY KEY,
      ename         VARCHAR2(10),
      job           VARCHAR2(9),
      mgr           NUMBER(4),
      sal           NUMBER(7,2),
      comm          NUMBER(7,2),
      deptno        NUMBER(2),
      FOREIGN KEY (DEPTNO) REFERENCES DEPT(DEPTNO));
```

The following statement created the `emp_dept` join view which is referenced in the examples:

```
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');
```

The concept of a key-preserved table is fundamental to understanding 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. So, a key-preserved table has its keys preserved through a join.

Note: It is not necessary that the key or keys of a table be selected for it to be key preserved. It is sufficient that if the key or keys were selected, then they would also be keys of the result of the join.

The key-preserving property of a table does not depend on the actual data in the table. It is, rather, a property of its schema. For example, if in the `emp` table there was at most one employee in each department, then `deptno` would be unique in the result of a join of `emp` and `dept`, but `dept` would still not be a key-preserved table.

If you select all rows from `emp_dept`, the results are:

```
EMPNO      ENAME      DEPTNO  DNAME          LOC
---------- ---------- ------- -------------- -----------
7782        CLARK           10 ACCOUNTING     NEW YORK
7839        KING            10 ACCOUNTING     NEW YORK
7934        MILLER          10 ACCOUNTING     NEW YORK
7369        SMITH           20 RESEARCH       DALLAS
7876        ADAMS           20 RESEARCH       DALLAS
7902        FORD            20 RESEARCH       DALLAS
7788        SCOTT           20 RESEARCH       DALLAS
7566        JONES           20 RESEARCH       DALLAS
8 rows selected.
```

In this view, `emp` is a key-preserved table, because `empno` is a key of the `emp` table, and also a key of the result of the join. `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.

The views described in the following table can assist you to identify inherently updatable join views.

View	*Description*
`DBA_UPDATABLE_COLUMNS`	Shows all columns in all tables and views that are modifiable.

View	*Description*
ALL_UPDATABLE_COLUMNS	Shows all columns in all tables and views accessible to the user that are modifiable.
USER_UPDATABLE_COLUMNS	Shows all columns in all tables and views in the user's schema that are modifiable.

The updatable columns in view emp_dept are shown below.

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

COLUMN_NAME                    UPD
------------------------------ ---
EMPNO                          YES
ENAME                          YES
DEPTNO                         YES
SAL                            YES
DNAME                          NO
LOC                            NO

6 rows selected.
```

Restrictions on DML operations for views use the following criteria in the order listed:

1. If a view is defined by a query that contains SET or DISTINCT operators, a GROUP BY clause, or a group function, then rows cannot be inserted into, updated in, or deleted from the base tables using the view.
2. If a view is defined with WITH CHECK OPTION, a row cannot be inserted into, or updated in, the base table (using the view), if the view cannot select the row from the base table.
3. If a NOT NULL column that does not have a DEFAULT clause is omitted from the view, then a row cannot be inserted into the base table using the view.
4. If the view was created by using an expression, such as DECODE(deptno, 10, "SALES", ...), then rows cannot be inserted into or updated in the base table using the view.