

27.5.3 Updatable and Insertable Views

Some views are updatable and references to them can be used to specify tables to be updated in data change statements. That is, you can use them in statements such as UPDATE, DELETE, or INSERT to update the contents of the underlying table. Derived tables and common table expressions can also be specified in multiple-table UPDATE and DELETE statements, but can only be used for reading data to specify rows to be updated or deleted. Generally, the view references must be updatable, meaning that they may be merged and not materialized. Composite views have more complex rules.

For a view to be updatable, there must be a one-to-one relationship between the rows in the view and the rows in the underlying table. There are also certain other constructs that make a view nonupdatable. To be more specific, a view is not updatable if it contains any of the following:

- Aggregate functions or window functions (SUM(), MIN(), MAX(), COUNT(), and so forth)
- `DISTINCT`
- `GROUP BY`
- `HAVING`
- UNION or UNION ALL
- Subquery in the select list

Nondependent subqueries in the select list fail for INSERT, but are okay for UPDATE, DELETE. For dependent subqueries in the select list, no data change statements are permitted.

- Certain joins (see additional join discussion later in this section)
- Reference to nonupdatable view in the `FROM` clause
- Subquery in the `WHERE` clause that refers to a table in the `FROM` clause
- Refers only to literal values (in this case, there is no underlying table to update)
- `ALGORITHM = TEMPTABLE` (use of a temporary table always makes a view nonupdatable)
- Multiple references to any column of a base table (fails for INSERT, okay for UPDATE, DELETE)

A generated column in a view is considered updatable because it is possible to assign to it. However, if such a column is updated explicitly, the only permitted value is `DEFAULT`. For information about generated columns, see Section 15.1.20.8, “CREATE TABLE and Generated Columns”.

It is sometimes possible for a multiple-table view to be updatable, assuming that it can be processed with the `MERGE` algorithm. For this to work, the view must use an inner join (not an outer join or a `UNION`). Also, only a single table in the view definition can be updated, so the `SET` clause must name only columns from one of the tables in the view. Views that use `UNION ALL` are not permitted even though they might be theoretically updatable.

With respect to insertability (being updatable with `INSERT` statements), an updatable view is insertable if it also satisfies these additional requirements for the view columns:

- There must be no duplicate view column names.
- The view must contain all columns in the base table that do not have a default value.
- The view columns must be simple column references. They must not be expressions, such as these:

```
3.14159
col1+ 3
UPPER(col2)
col3 / col4
(subquery)
```

MySQL sets a flag, called the view updatability flag, at `CREATE VIEW` time. The flag is set to `YES` (true) if `UPDATE` and `DELETE` (and similar operations) are legal for the view. Otherwise, the flag is set to `NO` (false). The `IS_UPDATABLE` column in the Information Schema `VIEWS` table displays the status of this flag. It means that the server always knows whether a view is updatable.

If a view is not updatable, statements such `UPDATE`, `DELETE`, and `INSERT` are illegal and are rejected. (Even if a view is updatable, it might not be possible to insert into it, as described elsewhere in this section.)

The updatability of views may be affected by the value of the `updatable_views_with_limit` system variable. See Section 7.1.8, “Server System Variables”.

For the following discussion, suppose that these tables and views exist:

```
CREATE TABLE t1 (x INTEGER);
CREATE TABLE t2 (c INTEGER);
CREATE VIEW vmat AS SELECT SUM(x) AS s FROM t1;
CREATE VIEW vup AS SELECT * FROM t2;
CREATE VIEW vjoin AS SELECT * FROM vmat JOIN vup ON vmat.s=vup.c;
```

`INSERT`, `UPDATE`, and `DELETE` statements are permitted as follows:

- INSERT: The insert table of an INSERT statement may be a view reference that is merged. If the view is a join view, all components of the view must be updatable (not materialized). For a multiple-table updatable view, INSERT can work if it inserts into a single table.

This statement is invalid because one component of the join view is nonupdatable:

```
INSERT INTO vjoin (c) VALUES (1);
```

This statement is valid; the view contains no materialized components:

```
INSERT INTO vup (c) VALUES (1);
```

- UPDATE: The table or tables to be updated in an UPDATE statement may be view references that are merged. If a view is a join view, at least one component of the view must be updatable (this differs from INSERT).

In a multiple-table UPDATE statement, the updated table references of the statement must be base tables or updatable view references. Nonupdated table references may be materialized views or derived tables.

This statement is valid; column *c* is from the updatable part of the join view:

```
UPDATE vjoin SET c=c+1;
```

This statement is invalid; column *x* is from the nonupdatable part:

```
UPDATE vjoin SET x=x+1;
```

This statement is valid; the updated table reference of the multiple-table UPDATE is an updatable view (*vup*):

```
UPDATE vup JOIN (SELECT SUM(x) AS s FROM t1) AS dt ON ...  
SET c=c+1;
```

This statement is invalid; it tries to update a materialized derived table:

```
UPDATE vup JOIN (SELECT SUM(x) AS s FROM t1) AS dt ON ...
```

```
SET s=s+1;
```

- DELETE: The table or tables to be deleted from in a DELETE statement must be merged views. Join views are not allowed (this differs from INSERT and UPDATE).

This statement is invalid because the view is a join view:

```
DELETE vjoin WHERE ...;
```

This statement is valid because the view is a merged (updatable) view:

```
DELETE vup WHERE ...;
```

This statement is valid because it deletes from a merged (updatable) view:

```
DELETE vup FROM vup JOIN (SELECT SUM(x) AS s FROM t1) AS dt ON ...;
```

Additional discussion and examples follow.

Earlier discussion in this section pointed out that a view is not insertable if not all columns are simple column references (for example, if it contains columns that are expressions or composite expressions). Although such a view is not insertable, it can be updatable if you update only columns that are not expressions. Consider this view:

```
CREATE VIEW v AS SELECT col1, 1 AS col2 FROM t;
```

This view is not insertable because `col2` is an expression. But it is updatable if the update does not try to update `col2`. This update is permissible:

```
UPDATE v SET col1 = 0;
```

This update is not permissible because it attempts to update an expression column:

```
UPDATE v SET col2 = 0;
```

If a table contains an `AUTO_INCREMENT` column, inserting into an insertable view on the table that does not include the `AUTO_INCREMENT` column does not change the value of `LAST_INSERT_ID()`, because the side effects of inserting default values into columns not part of the view should not be visible.

© 2024 Oracle
