

1 Views

1.1 Motivation

1. It is not always desirable for all users to see the entire set of relations in the database
2. Security considerations may require that only certain data in a relation be hidden from a user.
3. Sometimes, we may wish to create a personalized collection of “virtual” relations that is better matched to a certain user’s intuition of the structure of the enterprise.

1.2 Details

1. A view in SQL terminology is a single table that is derived from other tables.
 - These other tables can be base tables (whose tuples are always physically stored in the database) or previously defined views.
2. It is considered to be a virtual table.
3. It is possible to compute and store the results of these queries and then make the stored relations available to users.
 - However, if we did so, and the underlying data in the relations changed, the stored query results would then no longer match the result of re-executing the query on the relations.
4. In general, it is a bad idea to compute and store query results (although there are some exceptions).
5. When we define a view, the database system stores the *definition of the view* itself, rather than the *result of evaluation of the query expression that defines the view*.
6. Wherever a view relation appears in a query, it is replaced by the stored query expression. Thus, whenever we evaluate the query, the view relation is recomputed.
7. View names may appear in a query any place where a relation name may appear.
8. The **WITH** clause allows us to assign a name to a subquery for use as often as desired, but in **one particular query** only.
9. View extend this concept beyond a single query. It is possible to support a large number of views on top of any given set of actual relations.
10. Views differ from the **WITH** statement in that views, once created, remain available until explicitly dropped. The named subquery defined by with is local to the query in which it is defined.

1.3 Materialized Views

Definition 1 (Materialized Views). *Certain database systems allow view relations to be stored, but they make sure that, if the actual relations used in the view definition change, the view is kept **up-to-date**. Such views are called **materialized views**.*

Definition 2 (Materialized View Maintenance). *The process of keeping the materialized view up-to-date is called **materialized view maintenance** (or just **view maintenance**).*

Different strategies as to when a materialized view is updated are possible.

1. **Immediate Update:** This strategy updates a view as soon as the base tables are changed.
2. **Lazy Update:** This strategy updates the view when needed by a view query, *i.e.*, when the view is accessed.
3. **Periodic Update:** This strategy updates the view periodically. In this case, the contents of the materialized view may be stale, *i.e.*, not up-to-date, when it is used. This method should not be used if the application needs up-to-date data.

Some database systems permit the database administrator to control which of the aforementioned methods is used for each materialized view.

SQL does not define a standard way of specifying that a view is materialized, but many database systems provide their own SQL extensions for this task

Materialized view can be beneficial when –

- Applications use a view frequently.

- Applications demand fast response to certain queries that compute aggregates over large relations.

The benefits to queries from the materialization of a view must be weighed against the storage costs and the added overhead for updates.

1.4 Update View

Views are a useful tool for queries. However, they present serious problems if we express updates, insertions, or deletions with them. The difficulty is that a modification to the database expressed in terms of a view must be **translated** to a modification to the actual relations in the logical model of the database.

Modifications are generally not permitted on view relations, except in limited cases. Different database systems specify different conditions under which they permit updates on view relations.

In general, an SQL view is said to be updatable (*i.e.*, inserts, updates, or deletes can be applied on the view) if the following conditions are all satisfied by the query defining the view.

- The **FROM** clause has only one database relation.
- The **SELECT** clause contains only attribute names of the relation and does not have any expressions, aggregates, or **DISTINCT** specification.
- Any attribute not listed in the **SELECT** clause can be set to null; that is, it does not have a **NOT NULL** constraint and is not part of a primary key.
- The query does not have a **GROUP BY** or **HAVING** clause.