

# Views

# View

A view is a named query that provides another way to present data in the database tables. A view is defined based on one or more tables which are known as base tables. When you create a view, you basically create a query and assign a name to the query. Therefore, a view is useful for wrapping a commonly used complex query.

Note that regular views do not store any data except the **materialized views**. In PostgreSQL, you can create special views called materialized views that **store data physically** and periodically refresh data from the base tables. The materialized views are handy in many scenarios, such as faster data access to a remote server and caching.

- Standard Views.
- Materialized Views.

# Standard Views

A view is a database object that is of a stored query. A view can be accessed as a virtual table in PostgreSQL. In other words, a PostgreSQL view is a logical table that represents data of one or more underlying tables through a SELECT statement. Notice that a view does not store data physically.

- `CREATE VIEW view_name AS query;`
- `ALTER VIEW old_name RENAME TO new_name;`
- `DROP VIEW [IF EXISTS] view_name [CASCADE | RESTRICT]`
- `DROP VIEW [ IF EXISTS ] view1, view2, ...;`

# Materialized views

PostgreSQL extends the view concept to the next level that allows views to store data physically. And these views are called **materialized views**.

**Materialized views** cache the result of a complex and expensive query and allow you to refresh this result periodically.

- `CREATE MATERIALIZED VIEW view_name AS query WITH [NO] DATA;`
- `REFRESH MATERIALIZED VIEW view_name;`
- `REFRESH MATERIALIZED VIEW CONCURRENTLY view_name; -- from postgresql 9.4`
- `DROP MATERIALIZED VIEW view_name;`

## Interview Questions