Views in SQL

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table

Advantages of database view

The following are advantages of using database views.

- A database view allows you to simplify complex queries: a database view is defined by an SQL statement that associates with many underlying tables. You can use database view to hide the complexity of underlying tables to the end-users and external applications. Through a database view, you only have to use simple SQL statements instead of complex ones with many joins.
- A database view helps limit data access to specific users. You may not want a subset of sensitive data can be queryable by all users. You can use a database view to expose only non-sensitive data to a specific group of users.
- A database view provides extra security layer. Security is a vital part of any relational database management system. The database view offers additional protection for a database management system. The database view allows you to create the read-only view to expose read-only data to specific users. Users can only retrieve data in read-only view but cannot update it.
- A database view enables computed columns. A database table should not have calculated columns however a database view should. Suppose in the `orderDetails` table you have `quantityOrder`(the number of ordered products) and `priceEach` (price per product item) columns. However, the `orderDetails` table does not have a computed column to store total sales for each line item of the order. If it has, the database schema would not be a good design. In this case, you can create a computed column named `total` , which is a product of `quantityOrder` and `priceEach` to represent the calculated result. When you query data from the database view, the data of the computed column is calculated on the fly.
- A database view enables backward compatibility. Suppose you have a central database, which many applications are using it. One day, you decide to redesign the database to adapt to the new business requirements. You remove some tables and create new tables, and you don't want the changes to affect other applications. In this scenario, you can create database views with the same schema as the legacy tables that you will remove.

Disadvantages of database view

Besides the advantages above, there are several disadvantages of using database views:

- Performance: querying data from a database view can be slow especially if the view is created based on other views.
- Tables dependency: you create a view based on underlying tables of the database. Whenever you change the structure of these tables that view associated with, you have to change the view as well.