**Assignment 10 : A view to list all active users along with their basic information.**

An SQL view to list all active users along with their basic information:

SQL

CREATE VIEW Active Users (

User\_ id INT PRIMARY KEY,

username VARCHAR(255) NOT NULL,

email\_ address VARCHAR(255) NOT NULL,

-- Add other basic information columns as needed

) AS

SELECT u.id AS user\_ id,

u. user name,

u. email\_ address,

-- Add selections for other basic information columns from relevant tables

FROM users u

INNER JOIN user\_ status us ON u.id = us. user\_ id

WHERE us. status = 'active';

**Explanation:**

1. **CREATE VIEW:** This statement defines the creation of a view named Active Users.
2. **Columns:**
   * User\_ id INT PRIMARY KEY: Specifies the user ID as the primary key for the view.
   * username VARCHAR(255) NOT NULL: Defines the username column.
   * email\_ address VARCHAR(255) NOT NULL: Defines the email address column.
   * **(Optional):** You can add additional columns for other basic user information you want to include in the view by selecting them from relevant tables.
3. **AS Clause:**
   * This clause defines the logic for populating the view.
4. **SELECT Statement:**
   * u.id AS user\_ id: Selects the user ID from the users table (aliased as u) and renames it to user\_ id for clarity.
   * u. user name, u. email\_ address: Selects username and email address from the users table.
   * **(Optional):** Add selections for other basic information columns from relevant tables, ensuring they have compatible data types with the view's columns.
5. **FROM Clause:**
   * FROM users u: Specifies the users table as the main source for user data (aliased as u).
6. **INNER JOIN:**
   * INNER JOIN user\_ status us ON u.id = us. user\_ id: This joins the user\_ status table (aliased as us) with the users table based on the condition that the user\_ id in both tables must match.
7. **WHERE Clause:**
   * WHERE us.status = 'active': Filters the results to only include users with an active status in the user\_status table.

**Benefits of using a View:**

* **Simplified Queries:** Users can query the ActiveUsers view with a simple SELECT \* statement to retrieve basic information about active users.
* **Improved Security:** If the underlying tables have sensitive information, the view can limit access to specific columns, enhancing data security.
* **Modular Design:** Views promote modularity by encapsulating the logic for identifying active users and their basic information.

**Note:**

* Replace users and user\_status with the actual names of your user and user status tables in your database schema.
* Adjust column names and data types based on your specific user data structure.