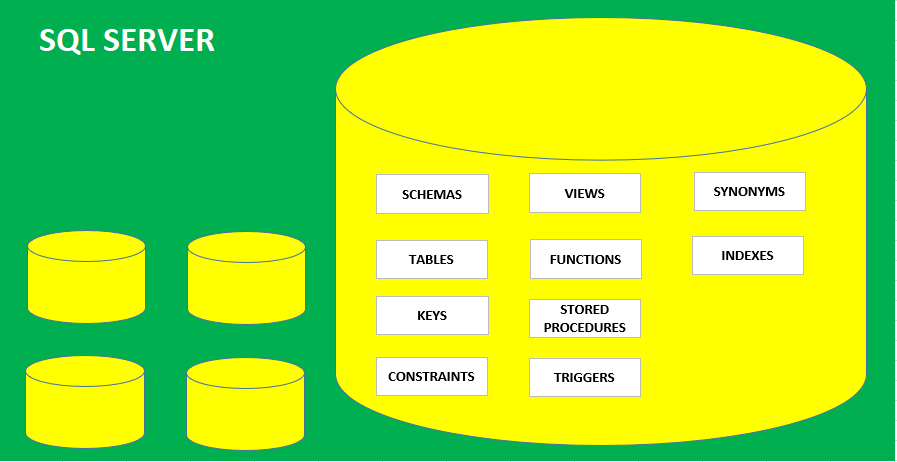
**VIEWS - BASICS**

****

**VIEWS** : DATABASE OBJECTS USED TO STORE **SELECT QUERY**.

VIEWS ARE ALSO CALLED VIRTUAL TABLES.

MEANS, END USERS FEEL THEM AS REGULAR TABLES BUT REALLY NOT.

REASON: VIEWS DO NOT STORE ANY DATA. VIEWS STORE ONLY **SELECT** QUERY.

**PURPOSE** : 1. TO IMPLEMENT ROW LEVEL SECURITY (**RLS**) BY DATABASE ADMINISTRATORS.

2. FOR REUSABILITY OF THE QUERY, FOR EASY REPORTING BY DATA ANALYSTS.

**RLS EXAMPLE:**

ASSUME A UNIVERSITY. DATA IS STORED IN A STUDENTS TABLE.

COMPUTERS STAFF SHOULD ACCESS ONLY COMPUTERS DATA.

ROBITICS STAFF SHOULD ACCESS ONLY ROBOTICS DATA.

ADMIN SHOULD ACCESS ALL STUDENTS DATA.

THIS KIND OF SECURED ACCESS TO STUDENTS TABLE CAN BE ACHIEVED BY VIEW.

**SYNTAX** CREATE VIEW <VIEW NAME>

AS

<< SELECT QUERY >>

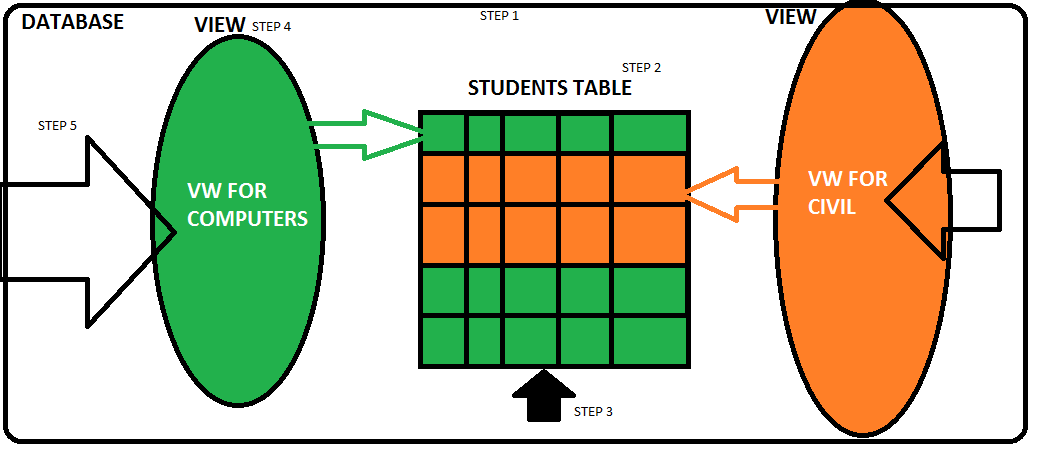
**OPERATIONS ON A VIEW:**

**SELECT** : WE CAN READ DATA USING A VIEW. UNDERLYING QUERY WILL BE EXECUTED.

**INSERT**: WE CAN INSERT DATA INTO A VIEW. BUT DATA GETS INSERTED TO BASE TABLE.

**UPDATE**: WE CAN UPDATE DATA INTO A VIEW. BUT DATA GETS UPDATED TO BASE TABLE.

**DELETE**: WE CAN DELETE DATA INTO A VIEW. DATA GETS DELETED FROM BASE TABLE.



**TO VERIFY THE VIEWS IN A DATABASE?**

FROM SSMS TOOL: IN OBJECT EXPLORER (LEFT) > EXPAND DATABASE > EXAND VIEWS > WE EE THE LIST.

**TO VERIFY THE DEFINITION OF A VIEW?**

FROM SSMS TOOL: IN OBJECT EXPLORER (LEFT) > EXPAND DATABASE > EXAND VIEWS > WE EE THE LIST. RIGHT CLICK REQUIRED VIEW: SCRIPT > CREATE > NEW QUERY WINDOW.