VIEWS

View: Views are Virtual Relations or virtual tables, through which a selective portion of the data from one or more relations (or tables) can be seen.

The tables on which the view is based are called base tables.

Views do not exist physically.

Views are stored in data dictionary.

You can create a view by using the **CREATE VIEW** command:

The CREATE VIEW statement is a data definition command.
Views provide a level of security in the database because the view can restrict users to only specified columns and specified rows in a table. For example, if you have a company with hundreds of employees in several Departments, you could give the secretary of each department a view of only certain attributes and for the employees that belong only to that secretary's department

Syntax:

```
CREATE VIEW view_name AS
        SELECT column_list
        FROM table_name [where condition];
```

Example:

- CREATE table student (rollno int,sname varchar(50),gender char(50),gmail varchar(50),DOB date,password varchar(50));
- DESC student;
- INSERT INTO student VALUES
  (601,"karthik","M","karthik@gmail.com",'1990-05-18',"123456"),
  (602,"raju","M","raju@gmail.com",'1990-04-21',"996655"),
  (603,"rajitha","F","rajitha123@gmail.com",'1990-02-09',"111111");
- SELECT * FROM student;

CREATING SIMPLE VIEW:

- CREATE VIEW student_details AS SELECT rollno,sname,gender FROM student;
- SELECT * FROM student_details;

Data modifications like insert, delete, and update operations on base table, it will reflect on views.

- INSERT INTO student VALUES (604,"Mahesh","M","mahesh@hmail.com","1992-01-28",3333);
- SELECT * FROM student_details;

Data modifications like insert, delete, and update operations on views affects the actual relations in the database, upon which view is based.

- UPDATE student_details SET sname="Narendher" WHERE rollno=601;
- SELECT * FROM student_details; // Updated successfully
- SELECT * FROM student;
- INSERT INTO student_details VALUES (655,"srikanth","M");
- SELECT * FROM student_details;
- SELECT * from student;
- DELETE FROM student_details WHERE rollno=601;
- SELECT * FROM student;
- SELECT * FROM student;
- CREATE TABLE employee (eno int PRIMARY KEY,ename varchar(40));
- CREATE TABLE employee_phne (eno int, phone_no char(50),FOREIGN KEY(eno) REFERENCES employee(eno));
- INSERT INTO employee VALUES (101,"Mahipal"),(102,"Mahonar");
- INSERT INTO employee_phne VALUES(101,"9885749409"),(102,"9160600571");

CREATING COMPLEX VIEWS: view creation that involves multiple tables.

- CREATE VIEW employe_details AS SELECT employee.eno, employee.ename, employee_phne.phone_no FROM employee, employee_phne WHERE employee.eno=employee_phne.eno;
- SELECT * from employe_details;

UPDATE VIEWS:

ALTER VIEW statement is used to modify or update the already created VIEWwithout dropping it.

Syntax:

ALTER VIEW view_name AS SELECT columns FROM table WHERE conditions;

Example:

- ALTER VIEW employe_details AS SELECT employee.eno, employee_phne.phone_no FROM employee, employee_phne WHERE employee.eno=employee_phne.eno;
- SELECT * FROM employe_details;

DROP VIEW:

We can drop the existing VIEW by using the DROP VIEW statement.

Syntax:

DROP VIEW view_name:

Example:

- DROP VIEW employe_details;

Advantages:

Views improve security of the database by showing only intended data toauthorized users. They hide sensitive data.

Views make life easy as you do not have write complex queries time and again.