```
CREATE
    [OR REPLACE]
    [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
    [DEFINER = user]
    [SQL SECURITY { DEFINER | INVOKER }]
    VIEW view_name [(column_list)]
    AS select_statement
    [WITH [CASCADED | LOCAL] CHECK OPTION]
```

```
// Creating Table Query

Create table students(studentID int primary key, studentName varchar(200), city varchar(200), subject
varchar(200), marks int);

// Inserting values

Insert into students values(1,'rohith','hyderabad','science',49);

Insert into students values(2,'meghna','Dehradun','science',80);

Insert into students values(3,'Aakash','delhi','english',88);
Insert into students values(4,'Praneeth','Nagpur','Social',92);
Insert into students values(5,'Manyatha','Karnataka','Maths',99);
Insert into students values(6,'Anushka','Rajasthan','Physics',78);
Insert into students values(7,'Aamir','Delhi','Chemistry',89);
```

Student table :

```
select * from employees;

Create table students(studentID int primary key, studentName varchar(200), city varchar(200),subject
varchar(200), marks int);
desc students;
Insert into students values(1,'rohith','hyderabad','science',99);
Insert into students values(2,'meghna','Dehradun','science',80);
Insert into students values(3,'Aakash','Delhi','english',88);
Insert into students values(4,'Praneeth','Nagpur','Social',92);
Insert into students values(5,'Manyatha','Karnataka','Maths',99);
Insert into students values(6,'Anushka','Rajasthan','science',49);
Insert into students values(7,'Aamir','Delhi','Chemistry',89);


Insert into students values(10,'meghna','Dehradun','science',80);
```

```
// Creating a view from Table
CREATE
VIEW Student_view
AS SELECT StudentID, StudentName, City, Subject, Marks
FROM Students;

// Creating a View from View
CREATE
VIEW Student_view_new
AS SELECT StudentID, StudentName, City, Subject, Marks
FROM Student_view WHERE Marks > 80
WITH CHECK OPTION;
```

INERT INTO

– We can drop a table even if any dependent views are associated with it, but the views that are associated with it will not be dropped. They will still execute in the database only with the status as inactive object and all those views become active and start functioning provided the table is recreated.

– We can create a view based on other views. Usually, we create views based on tables, but it is also possible to create views based on views.

– Views can be updated. However, updating a view that is based on multiple tables, may not update the underlying tables correctly. To correctly update a view that is based on multiple tables we can make use "Instead OF triggers" in SQL Server.

– The WITH CHECK OPTION clause ensures that any data modifications made to the view can be successfully checked against the underlying data in the table. This means that the view must be created with a SELECT statement that is guaranteed to return only data that is compatible with the data in the underlying table.

———

UPDATING VIEWS

All views are not updatable. So, UPDATE command is not applicable to all views.

An updatable view is one which allows performing a UPDATE command on itself without affecting any other table.

There are certain conditions needed to be satisfied to update a view.

If any one of these conditions is not met, then we will not be allowed to update the view.

1. The SELECT statement which is used to create the view should not include GROUP BY clause or ORDER BY clause.

2. The SELECT statement should not have the DISTINCT keyword.

3. The View should have all NOT NULL values.

4. The view should not be created using nested queries or complex queries.

5. The view should be created from a single table. If the view is created using multiple tables then we will not be allowed to update the view.

Syntax:

UPDATE < view_name > SET<column1>=<value1>,<column2>=<value2>,.....

WHERE <condition>;

—--
// To make a view READ_ONLY
create view student_view3
as select distinct * from students;

Insert into student_view3 values(9,student view 3','hyderabad','science',90);
Output:
ERROR 1471 (HY000): The target table student_view3 of the INSERT is not insertable-into
—---
 // To make a view READ_ONLY
create view student_view_distinct
as select DISTINCT StudentName from students;
—---
// To make a view READ_ONLY
create view student_view_

as select SKIP * from students;

create view student_view_first
as select FIRST * from students;

Insert into student_view_distinct values('king');
Insert into student_view3 values(10,'student view 3','Pune','DBMS',20);

———

// Deleting entry from Distinct View
DELETE
FROM Student_view_distinct
WHERE StudentName='rohith';
Output:
ERROR 1288 (HY000): The target table Student_view_distinct of the DELETE is not updatable

———

// Updating
UPDATE Student_view_distinct SET 'Rohith' WHERE StudentName='rohith';

DELETE
FROM Student_view3
WHERE StudentName = 'rohith';

ERROR 1288 (HY000): The target table Student_view3 of the DELETE is not updatable

UPDATE Student_view_distinct SET StudentName = 'Rohit' WHERE StudentName='rohith';

ERROR 1288 (HY000): The target table Student_view_distinct of the UPDATE is not updatable