SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Re-accredited by NAAC with A+ grade, Accredited by NBA(CSE, IT, ECE, EEE & Mechanical)
Approvedy by AICTE, New Delhi, Recognized by UGC, Affiliated to Anna University, Chennai

Department of MCA

DBMS Views (Virtual Tables) in SQL

Course Name : 23CAT603 - DATA BASE MANAGEMENT SYSTEM

Class : I Year / I Semester

Unit IV – Views (Virtual Tables) in SQL

Views (Virtual Tables) in SQL

Views in SQL

Views in SQL are considered as a virtual table. A view also contains rows and columns.

To create the view, we can select the fields from one or more tables present in the database.

A view can either have specific rows based on certain condition or all the rows of a table.

Advantages of View:

Complexity: Views help to reduce the complexity. Different views can be created on the same base table for different users.

Security: It increases the security by excluding the sensitive information from the view.

Query Simplicity: It helps to simplify commands from the user. A view can draw data from several different tables and present it as a single table.

Consistency: A view can present a consistent, unchanged image of the structure of the database. Views can be used to rename the columns without affecting the base table.

Data Integrity: If data is accessed and entered through a view, the DBMS can automatically check the data to ensure that it meets the specified integrity constraints.

Storage Capacity: Views take very little space to store the data.

Logical Data Independence: View can make the application and database tables to a certain extent independent.

Views (Virtual Tables) in SQL

Disadvantages of View:

The DML statements which can be performed on a view created using single base table have certain restrictions are:

You cannot INSERT if the base table has any not null column that do not appear in view.

You cannot INSERT or UPDATE if any of the column referenced in the INSERT or UPDATE contains group functions or columns defined by expression.

You can't execute INSERT, UPDATE, DELETE statements on a view if with read only option is enabled.

You can't be created view on temporary tables.

You cannot INSERT, UPDATE, DELETE if the view contains group functions GROUP BY, DISTINCT or a reference to a psuedocolumn rownum.

You can't pass parameters to the SQL server views.

You can't associate rules and defaults with views.

Views (Virtual Tables) in SQL

Sample table:

Student_Detail

STU_ID	NAME	ADDRESS
1	Stephan	Delhi
2	Kathrin	Noida
3	David	Ghaziabad
4	Alina	Gurugram

Student_Marks

STU_ID	NAME	MARKS	AGE
1	Stephan	97	19
2	Kathrin	86	21
3	David	74	18
4	Alina	90	20
5	John	96	18

SQL Views

1. Creating view

A view can be created using the **CREATE VIEW** statement. We can create a view from a single table or multiple tables.

Syntax:

```
CREATE VIEW view_name AS
SELECT column1, column2.....
FROM table_name
WHERE condition;
```

2. Creating View from a single table

In this example, we create a View named DetailsView from the table Student_Detail.

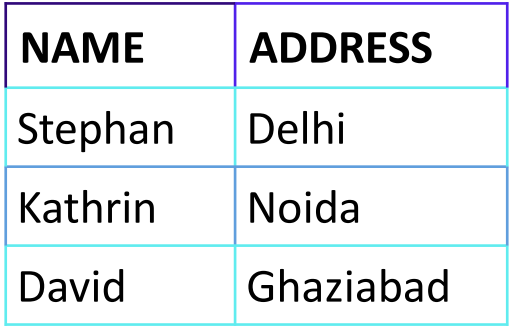
Query:

```
CREATE VIEW DetailsView AS
SELECT NAME, ADDRESS
FROM Student_Details
WHERE STU_ID < 4;
```

Just like table query, we can query the view to view the data.

```
SELECT * FROM DetailsView;
```

Output:



NAME	ADDRESS
Stephan	Delhi
Kathrin	Noida
David	Ghaziabad

SQL Views

3. Creating View from multiple tables

View from multiple tables can be created by simply include multiple tables in the SELECT statement.

In the given example, a view is created named MarksView from two tables Student_Detail and Student_Marks.

Query:

```
CREATE VIEW MarksView AS
SELECT Student_Detail.NAME, Student_Detail.ADDRESS, Student_Marks.MARKS
FROM Student_Detail, Student_Mark
WHERE Student_Detail.NAME = Student_Marks.NAME;
```

To display data of View MarksView:

```
SELECT * FROM MarksView;
```

NAME	ADDRESS	MARKS
Stephan	Delhi	97
Kathrin	Noida	86
David	Ghaziabad	74
Alina	Gurugram	90

SQL Views

4. Deleting View

A view can be deleted using the Drop View statement.

Syntax

DROP VIEW view_name;

Example:

If we want to delete the View **MarksView**, we can do this as:

DROP VIEW MarksView;

Significance of Views:

Views are highly significant, as they can provide advantages over tasks. Views can represent a subset of data contained in a table. Consequently they can limit the degree of exposure of the underlying base table to the outer world. They are used for security purpose in database and act as an intermediate between real table schemas and programmability. They act as aggregate tables.

SQL Views

CREATE VIEWS in SQL

We can create a view using **CREATE VIEW** statement. A View can be created from a single table or multiple tables.

Syntax:

```
CREATE VIEW view_name AS
SELECT column1, column2…..
FROM table_name
WHERE condition;
```

Parameters:

view_name: Name for the View

table_name: Name of the table

condition: Condition to select rows

SQL CREATE VIEW Statement Examples

Let's look at some examples of CREATE VIEW Statement in SQL to get a better understanding of how to create views in SQL.

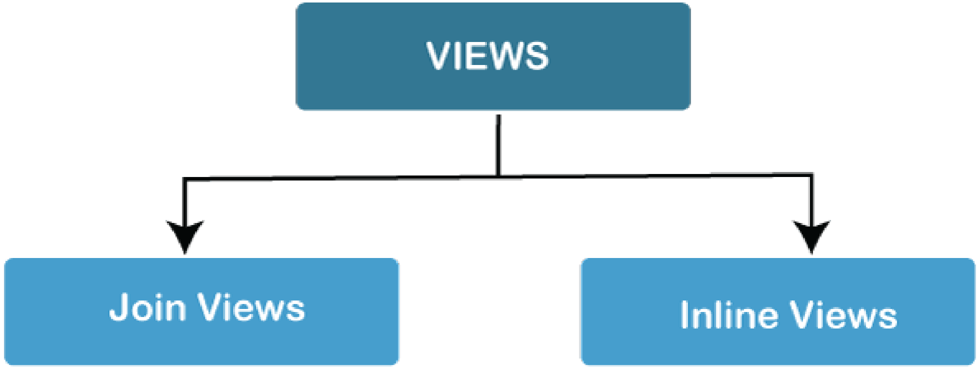
SQL Views

Types of Views:

There are two types of views.

Join View: A join view is a view that has more than one table or view in its from clause and it does not use any Group by Clause, Rownum, Distinct and set operation.

Inline View: An inline view is a view which is created by replacing a subquery in the from clause which defines the data source that can be referenced in the main query. The sub query must be given an alias for efficient working.



VIEWS

Join Views

Inline Views

References

1. https://www.javatpoint.com/dbms-sql-view