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:

1. **Complexity:** Views help to reduce the complexity. Different views can be created on the same base table for different users.
2. **Security:** It increases the security by excluding the sensitive information from the view.
3. **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.
4. **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.
5. **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.
6. **Storage Capacity:** Views take very little space to store the data.
7. **Logical Data Independence:** View can make the application and database tables to a certain extent independent.

Disadvantages of View:

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

1. You cannot INSERT if the base table has any not null column that do not appear in view.
2. You cannot INSERT or UPDATE if any of the column referenced in the INSERT or UPDATE contains group functions or columns defined by expression.
3. You can't execute INSERT, UPDATE, DELETE statements on a view if with read only option is enabled.
4. You can't be created view on temporary tables.
5. You cannot INSERT, UPDATE, DELETE if the view contains group functions GROUP BY, DISTINCT or a reference to a psuedocolumn rownum.
6. You can't pass parameters to the SQL server views.
7. You can't associate rules and defaults with views.

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 |

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.

1. **SELECT** \* **FROM** DetailsView;

**Output:**

|  |  |
| --- | --- |
| **NAME** | **ADDRESS** |
| Stephan | Delhi |
| Kathrin | Noida |
| David | Ghaziabad |

Advertisement

3. **Creating View from multiple tables**

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

Advertisement

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:

1. **SELECT** \* **FROM** MarksView;

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

4. Deleting View

A view can be deleted using the Drop View statement.

**Syntax**

1. **DROP** **VIEW** view\_name;

**Example:**

Advertisement

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

1. **DROP** **VIEW** MarksView;

Advertisement

Significance of Views:

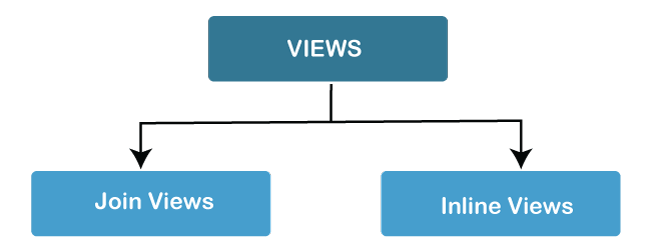
Views are highly significant, as they can provide advantages over tasks. Views can represent a subset of data contained in a table. Consequently they can limit the degree of exposure of the underlying base table to the outer world. They are used for security purpose in database and act as an intermediate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

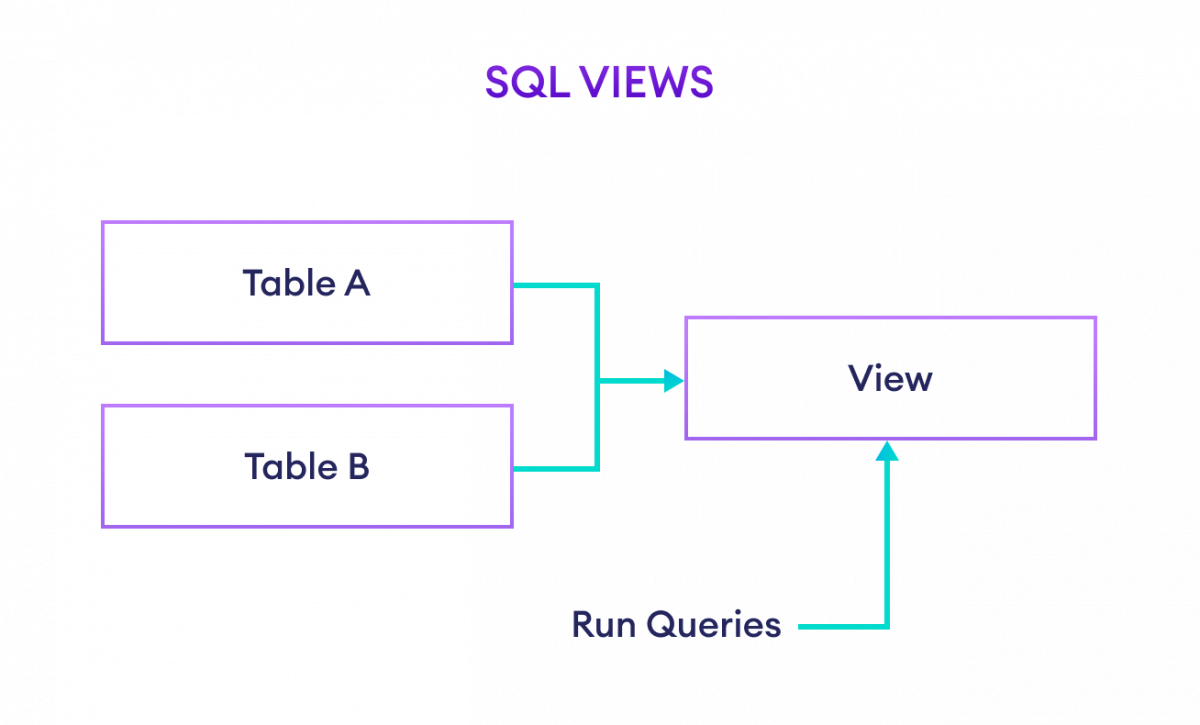
Advertisement

Types of Views:

There are two types of views.

Advertisement



1. **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.
2. **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.
3. **SQL Views**
4. In SQL, views contain rows and columns similar to a table, however, views don't hold the actual data.
5. You can think of a view as a virtual table environment that's created from one or more tables so that it's easier to work with data.
6. Example: SQL Views
7. **Creating a View in SQL**
8. We can create views in SQL by using the CREATE VIEW command. For example,
9. CREATE VIEW us\_customers AS
10. SELECT customer\_id, first\_name
11. FROM Customers
12. WHERE Country = 'USA';
13. [Run Code](https://www.programiz.com/sql/online-compiler)
14. Here, a view named us\_customers is created from the customers table.
15. Now to select the customers who lives in **USA**, we can simply run,
16. SELECT \*
17. FROM us\_customers;
18. [Run Code](https://www.programiz.com/sql/online-compiler)
19. **Updating a View**
20. It's possible to change or update an existing **view** using the CREATE OR REPLACE VIEW command. For example,
21. CREATE OR REPLACE VIEW us\_customers AS
22. SELECT \*
23. FROM Customers
24. WHERE Country = 'USA';
25. [Run Code](https://www.programiz.com/sql/online-compiler)
26. Here, the us\_customers view is updated to show all the fields.
27. **Deleting a View**
28. We can delete views using the DROP VIEW command. For example,
29. DROP VIEW us\_customers;
30. [Run Code](https://www.programiz.com/sql/online-compiler)
31. Here, the SQL command deletes the view named us\_customers.
32. **Note:** If the view is not available, the above command throws an error.
33. **Views for Complex Queries**
34. Suppose **A** and **B** are two tables and we wan't to select data from both of the tables. For that, we have to use [SQL JOINS](https://www.programiz.com/sql/join).
35. However using the JOIN each time could be a tedious task. For that, we can create a view to fetch records easily.
36. Let's create a view,
37. CREATE VIEW order\_details AS
38. SELECT Customers.customer\_id, Customers.first\_name, Orders.amount
39. FROM Customers
40. JOIN Orders
41. ON Customers.customer\_id = Orders.customer\_id;
42. [Run Code](https://www.programiz.com/sql/online-compiler)
43. Now, to select the data, we can run
44. SELECT \*
45. FROM order\_details;
46. [Run Code](https://www.programiz.com/sql/online-compiler)\\\\

views

select \* from student

CREATE VIEW DetailsView

AS

SELECT name, city FROM student WHERE studid < 4;

select \* from detailsview

---------------------------------------------------------------------

multiple tables view

CREATE VIEW MarksView AS

SELECT Student.NAME, Student1.stud\_name, Student1.course\_enrolled

FROM Student, Student1

WHERE Student.NAME = Student1.stud\_name

select \* from MarksView

select \* from student

select \* from student1

------------------------------------------------------------------------------

CREATE VIEW view\_customers

AS

SELECT custid, name FROM Customer WHERE salary >=2000

select \* from view\_customers

----------------------------------------------------------------------------------

Update the views :

create or alter view view\_customers2

as

SELECT custid, name FROM Customer WHERE salary <=5000