Views in SQL

* Views in SQL are 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.

The reasons for using Views”

* Views restrict access to the data because the view can display selective columns and rows from the table.
* Views provide groups of users with access to data according to their particular permissions.
* Views can be used to retrieve data from several tables, providing data independence for users.

Consider two tables: Student\_table and Student\_marks

Student\_table

|  |  |  |
| --- | --- | --- |
| Stud\_id | Name | Address |
| 1 | Stephan | Delhi |
| 2 | Kathrin | Noida |
| 3 | David | Ghaziabad |
| 4 | Alina | Gurugram |

Student\_marks

|  |  |  |  |
| --- | --- | --- | --- |
| Stud\_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.

SELECT \* FROM DetailsView;

Output:

|  |  |
| --- | --- |
| Name | Address |
| Stephan | Delhi |
| Kathrin | Noida |
| David | Ghaziabad |

## **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 |

## **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;

**DCL- Data control language (DCL)**

* DCL- Data control language (DCL) is used to access the stored data.
* It is mainly used for revoke and to grant the user the required access to a database.
* It is a part of the structured query language (SQL).
* It helps in controlling access to information stored in a database.
* It is the simplest among three commands.
* It provides the administrators, to remove and set database permissions to desired users as needed.
* These commands are employed to grant, remove and deny permissions to users for retrieving and manipulating a database.

Let us understand the difference between system and object privileges?

* In a database, every user account can be granted a number of *privileges*, which are also known as *permissions*.
* These privileges allow a particular user account to do certain things, like DELETE and UPDATE certain tables, CREATE a database, SELECT from a certain table, and many other things.
* In Oracle, Microsoft’s SQL Server, and in Sybase Adaptive Server privileges are further divided into two different categories: **1. system privileges and 2. object privileges.**

**How to create user in oracle DBMS?**

* Once connected as SYSTEM, simply issue the [CREATE USER](https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8003.htm) command to generate a new account.

Syntax: CREATE USER books\_admin IDENTIFIED BY MyPassword;

Here we’re simply creating a books\_admin account that is IDENTIFIED or authenticated by the specified password.

## **System privileges**

System privileges are privileges given to users to allow them to perform certain functions that deal with *managing the database and the server*. Most of the different types of permissions supported by the database vendors fall under the system privilege category

## Examples of Oracle system privileges

* CREATE USER. The CREATE USER permission, when granted to a database user, allows that database user to create new users in the database.
* CREATE TABLE. The CREATE TABLE permission, when granted to a database user, allows that database user to create tables in their own schema. This type of privilege is also available for other object types – like stored procedures and indexes.
* CREATE SESSION. The CREATE SESSION permission, when granted to a database user, allows that database user to connect to the database.

## **Object privileges**

|  |
| --- |
|  |

* Object privileges are privileges given to users so that they can perform certain actions upon certain database objects – where database objects are things like tables, stored procedures, indexes, etc.
* Some examples of object privileges include granting a particular database user the right to DELETE and/or SELECT from a particular table. This is done using the GRANT clause,.

**Hence,** system privileges are used for server and database privileges. But object privileges are used to grant privileges on database objects like tables, stored procedures, indexes, etc.

## **DDL Commands**

The Data Definition Language (DDL) commands are as follows −

### GRANT Command

It is used to grant a privilege(permissions) to a user. GRANT command allows specified users to perform specified tasks

**Syntax**

GRANT privilege\_name on objectname to user;

Here,

* privilege names are SELECT,UPDATE,DELETE,INSERT,ALTER,ALL
* objectname is table name
* user is the name of the user to whom we grant privileges

### REVOKE Command

It is used to remove a privilege(permissions) from a user. REVOKE helps the owner to cancel previously granted permissions.

**Syntax**

 REVOKE privilege\_name on objectname from user;

Here,

* privilege names are SELECT,UPDATE,DELETE,INSERT,ALTER,ALL
* objectname is table name
* user is the name of the user whose privileges are removing

### Example

GRANT SELECT, UPDATE ON employees TO Bhanu

Explanation − Firstly, to give the permissions to user, we have to use GRANT command. The privileges are SELECT because to view the records and UPDATE to modify the records. The objectname is table name which is Employee. The user name is bhanu.

REVOKE SELECT, UPDATE ON employees TO Bhanu

Explanation − Firstly, to revoke the permissions to user, we have to use REVOKE command. The privileges Need to revoke are SELECT because to view the records and UPDATE to modify the records. The objectname is table name which is Employee. The user name is Bhanu.