

## \* Q. Explain database languages

- To create db schema and database objects like table data definition language can be used
  - DDL statements are used to build & modify structure of your tables and other objects in database.
  - Set of DDL commands are as below
    - 1. Create stmt: create db objects.
    - 2. ALTER stmt: modify structure of db objects.
    - 3. DROP stmt: remove db objects
    - 4. RENAME stmt: rename db objects
    - 5. TRUNCATE stmt: empty db table
  - When you execute a DDL stmt, it takes effect immediately, as it is Autocommitted in to db. Hence no rollback operation (undo) can be performed with these set of commands.
  - Database objects are any data structure created in table db.
- Ex: table view, sequence.
- Data control language (DCL) is used to control various user actions (or privileges) in db.
  - To perform any operation in the db user needs privileges like creating table, sequence..
  - DCL is set of commands used to.
    - Grant: privilege gives some privileges to user for performing task on db.

→ Revoke : take back permission given to user.

- Privileges can be of many types:

→ System privileges : creating a table

→ Object privileges : executing a query

→ Ownership privileges: to execute query on table created by same user.

- Data manipulation language

- statements used for manipulating or managing data in db.

- DML commands are not auto-committed like DDL statements

- It means changes done by DML command can be rolled back

- DML is a set of command to:

1. INSERT Stmt:

2. DELETE Stmt:

3. UPDATE Stmt:

1. Insert Stmt:

- used to add records to existing table

- insert data into a table

- insert few values & per column names we can use generic syntax below.

INSERT INTO <Table name> (Column 1 ... Column N)  
VALUES (column 1, ..., column N)

- if all values for all columns of the table are to be added then also no need to specify the

columns name in query.

- But we need to make sure the order of values is in the same order as the columns in the table.

- `INSERT INTO <TABLE-NAME>  
VALUES ( column 1, ..., column N )`

- Example.

## 2. DELETE Stmt:

- used to delete some or all records from existing table.
- to delete data, SQL DELETE command is used
- to delete all the rows in table, we can use generic syntax:

`DELETE`

`FROM <TABLE-NAME>`

- To delete selected rows from table, we can specify where condition

`DELETE`

`FROM <Table-Name>`

`WHERE < condition >`

- give ex.

Q What is the view in SQL, how is it defined? Discuss the problem that may arise when we attempt to update a view. How views are implemented?

### i. Definition of views:

- i. Basically, a view is a single table derived from multiples tables or logically representing a subset of data.
- ii. These other tables can be base tables or defined view
- iii. A view is considered to be a virtual table in contrast to base table, whose tuples are physically stored in db.
- iv. There are 2 types of view

#### a) simple view:

- view which are based only on one table
- allow to perform DML Operations with some restrictions
- query defining simple view cannot have any join or grouping condition.

#### b) complex view:

- view which are based on more than one table
- do not allow DML operations
- query defining complex view can have join or grouping condition.

### 3. UPDATE STMT:

- used to modify the existing data present in db.
- SQL UPDATE cmd is used
- to update all the rows, the generic syntax is

UPDATE <table-name>

SET column1 = new value;

- to update selected rows from table, we can use the Where condition.

UPDATE <Table Name>

SET column1 = new value

Where <Condition>;

\*Q. What is the view in SQL, how is it defined? Discuss the problem that may arise when we attempt to update a view. How views are implemented?

## I. Definition of views.

- i. Basically, a view is a single table derived from multiple tables or logically representing subset of data.
- ii. These other tables can be base tables or defined view
- iii. A view is considered to be a virtual table in contrast to base table, whose tuples are physically stored in db.
- iv. There are 2 types of view
  - a) simple view:
    - view which are based only on one table
    - allow to perform DML Operations with some restrictions
    - query defining simple view cannot have any join or grouping condition.
  - b) complex view:
    - view which are based on more than one table
    - do not allow DML operations
    - query defining complex view can have join or grouping condition.

## 2. Implementation of view.

- i In SQL, the command to specify a view is CREATE VIEW. The view is given a virtual table name, a list of attribute names, and a query to specify the contents of a view
- Following is the syntax of the view

CREATE [OR REPLACE] VIEW < view-name >

AS

SUBQUERY

- SUBQUERY : The query which retrieve the columns of the table that query must have.

- ii - Consider a table TBL\_Employee with the fields Emp-ID, Emp- Name, EMP-DOB, Emp-Address, Emp-joindate, Emp-gender, Emp-email.

- Following is the view vw\_EmployeeContactDetails which contains the name & email of employee

CREATE VIEW vw\_EmployeeContactDetails

SELECT EMP-Name, Emp-ID

FROM TBL\_Employee

- It results in creation of view.

- iii. - If we don't need the view anymore, we can use the drop DROP VIEW stmt

- Syntax is

DROP VIEW <view-name>.

3 Problems updating the view.

- i. Updating of views is complicated and can be ambiguous task.
- ii. In general, an update of view defined on a single table without any aggregate functions can be mapped to an update on the underlying base table under certain conditions.
- iii. For a view involving joins, an update operation may be mapped to ~~an~~ update operations in multiple ways. Hence, it is not often possible for the DBMS to determine which of the updates is intended.
- iv. Generally, a view update is feasible when only one possible update on the base relations can be accomplished the desired update effect on view.
- v. Whenever an update on the view can be mapped to more than one update on the underlying base relations, it is necessary to have a certain procedure for choosing one of the possible updates as the most likely one.

vi. Thus problems with updating a view can be summarized as follows.

- a view with a single defining table is updatable if the view attributes contain the primary key of the base relation, as well as attributes with the NOT NULL constraint do not have default values specified
- it is generally not possible to update view on defined multiple tables.
- it is not possible to update views defined using grouping and aggregate functions.

Q. What are triggers? Explain with example.

### Introduction

- A trigger is a procedure that is automatically invoked by the DBMS in response to specific alteration to the db
- A trigger defines an action the db should take when some db related event occur.
- A db trigger is a stored subprogram associated with a table
- Triggers are, in fact, written to be executed in response of any of the following events
  - A database manipulation (DML) statement
  - A database definition (DDL) statement
  - A database operation.

Triggers can be defined on the table, view, schema or db with which the event is associated.

### Components of Triggers (E-C-A) model.

- Event (E) - SQL statement that causes the trigger to activate
- Condition (C) - A condition must be satisfied for execution of trigger
- Action (A) - This is code or statement satisfied that execute when triggering condition is satisfied & trigger is activated on db table

Syntax for creating a trigger:

```
1 CREATE [ OR REPLACE ] TRIGGER trigger-name  
2 { BEFORE | AFTER | INSTEAD OF }  
3 { INSERT [OR] UPDATE [OR] DELETE }  
4 [ OF COLUMN_NAME ]  
5 ON TABLE_NAME  
6 [Referencing old AS o NEW AS n]  
7 [FOR EACH ROW]  
8 When [condition]  
9 DECLARE  
10 Declaration - statements  
11 BEGIN  
12 Executable - statements  
13 EXCEPTION  
14 Exception-handling - statements  
15 END;
```

Where,

- 1 → creates or replace an existing trigger with trigger-name
- 2 → specifies when trigger will be executed
- 3 → specifies DML operation
- 4 → column name that will be updated
- 5 → allows us to refer new and old values for various dml statements
- 7 → specifies a row-level trigger. i.e. the trigger will be executed for each row affected.
- 8 → provides a condition for rows for which the trigger would fire. valid only for row level triggers.

## Q) Types of triggers . operations.

a) Data dictionary for triggers.

Once triggers are created, their definitions can be viewed by selecting it from system tables as shown below:

Syntax:

```
SELECT *  
FROM user_triggers  
WHERE Trigger_name = '<Trigger Name>';
```

This statement will give you all the properties of trigger including trigger as well.

b) dropping triggers

To remove trigger from db we use DROP cmd.

Syntax:

```
DROP Trigger <Trigger Name>;
```

c) disabling Triggers

To deactivate trigger temporarily, this can be activating trigger by enabling it.

Syntax:

```
Alter trigger <Trigger Name> { Disable | Enable };
```

## Q. Advantages & disadvantages of trigger.

A:

- i) Triggers are useful for enforcing referential integrity, which preserves the defined relationship between tables when you add, update, or delete rows in those tables.
- ii) After finding that the new information is inconsistent with the db, raise an error that will cause the entire transaction to roll back.

DA:

A trigger hampers the performance of system as db operations will go on slower due to triggering action.

### i) Restrictions on triggers.

- we cannot modify the same table on which triggering event is written.
- we cannot modify a table which is connected to the triggering table by primary key or foreign key.

### ii) Mutating table errors.

- happens when the trigger is querying or modifying table whose modification activated the trigger, or a table that might need to be updated because of foreign key constraint with a cascade policy.

- Error: ORA-04091: table mutating, trigger function