Page No.

Date 20/09/2024

① What is Diff b/w DBMS & RDBMS

DBMS stores data as file.	RDBM stores data in TABULAR
NO relationship b/w data	Data is stored — RELATED to each other
Normalization is not present	Normalization is present
It deals with small quantity of data	large amount of data
Ex: XML	Ex: MySQL, PostgreSQL, Oracle.

② Constraints in SQL?

Sql constraints are used to specify rules for the data in table.

```
create TABLE Students(
    ID int NOT NULL PRIMARY KEY,
    Name varchar(255) NOT NULL,
    CourseID int FOREIGN KEY REFERENCES Course(CourseID),
    Age int NOT NULL CHECK (Age>=18),
    AdmissionDate date DEFAULT GETDATE(),
    CONSTRAINT uc_Student UNIQUE (ID, Name)
```

① PRIMARY KEY: is field which can uniquely identify each row in a table.

② NOT NULL: We can't store null values in table.column.

③ FOREIGN KEY: is a field which can uniquely identify each row in another table.

④ CHECK: helps to validate the values of column to meet a particular condition.

⑤ DEFAULT: Specifies default value of column when no value is specified by the user.

⑥ UNIQUE: constraint tell that all values in the column must be unique.

Page No.
Date / /

(3) Primary key	Unique key
can't accept Null values	can accept only one Null value
Creates clustered Index	Non-clustered Index
Only one primary key	more than one unique key in a table.

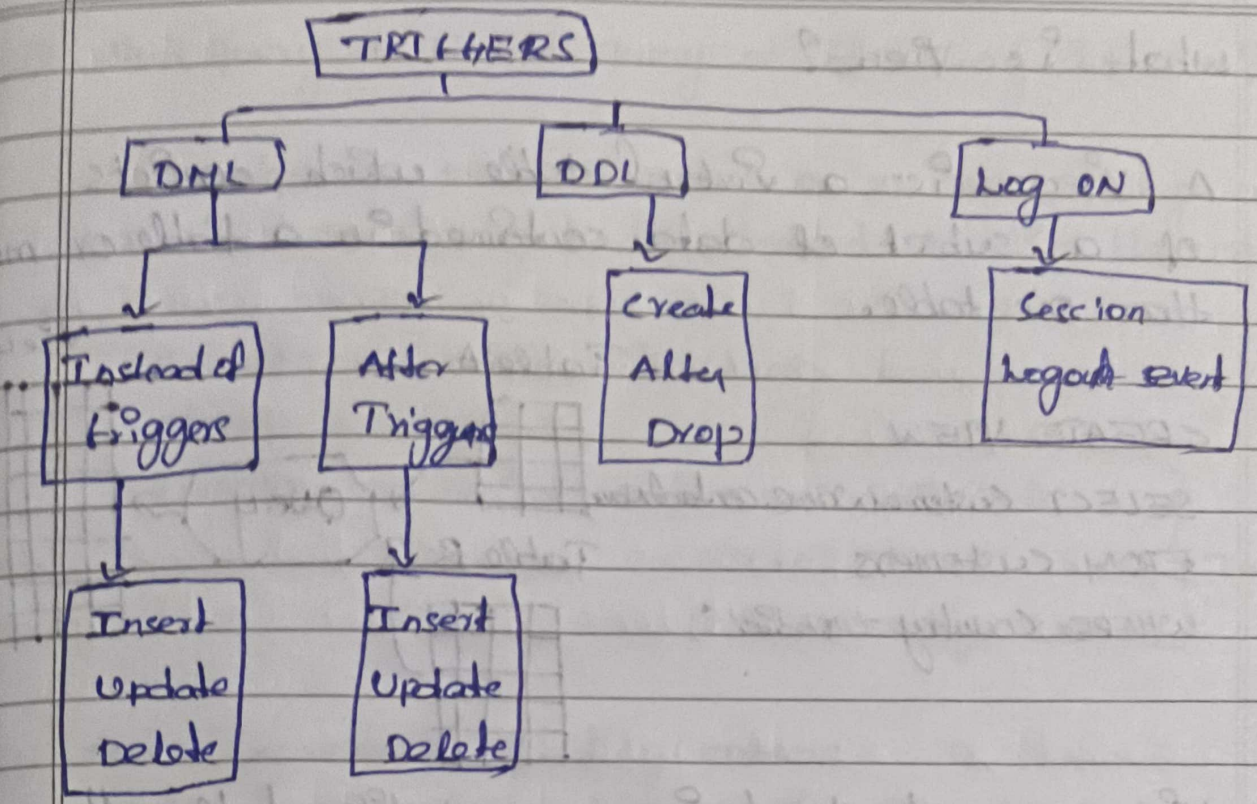
(4) Triggers and Trigge Types:

Triggers are stored programs which are automatically executed or fired when some event (inserts, delete, update) occur.

Example of After(DML) Trigger

```
CREATE TRIGGER TR_UPD_LOCATIONS ON LOCATION
                      ↓                 ↓
                 Trigger Name       Table Name
FOR UPDATE
NOT FOR REPLICATION
AS
BEGIN
    INSERT INTO LocationList
    SELECT LocationID
           ~~,generate~~,getdate()
      FROM INSERTED
END
```

(boxed statements) → T-SQL that runs original specified DML Event.



Example of instead of (DML) Trigger

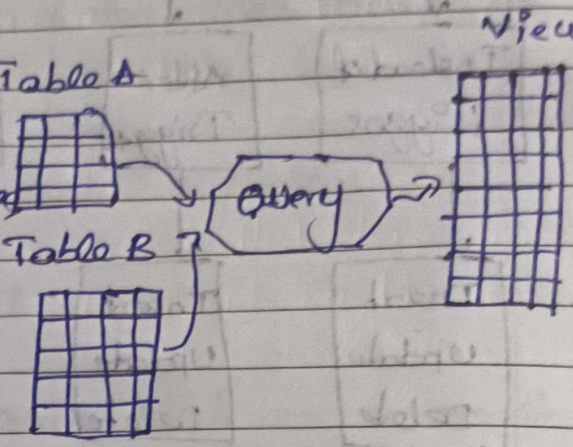
```
CREATE TRIGGER [dbo].[TRG_VW_EMPDETAILS]   → Trigger Name
ON [dbo].[VW_EMPDETAILS]                   → View Name
INSTEAD OF INSERT
AS
BEGIN
  --Logic Here
END
```

An Instead of trigger that allows you to deploy skip an INSERT, DELETE, or UPDATE statement to a table or a View and execute other statements defined in the trigger instead.

Page No.
Date / /

⑤ What is View?

A View is a virtual table which consists of a subset of data contained in a table or more than one table.




```
CREATE VIEW
SELECT CustomerName, ContactName
FROM Customers
WHERE Country = "India";
```

Views are not stored in memory like tables then why to views.

1. Indexed Views to improve the performance.
2. Extra Security — DBA can hide the actual table names and expose views for Read Operations.

⑥ Having clause and where clause.

WHERE

where clause is used before GROUP BY clause
WHERE clause is used after GROUP BY clause.

```
SELECT COUNT(CustomerID), Country
FROM Customers
→ WHERE Country = "India"
GROUP BY Country
→ HAVING COUNT(CustomerID) > 5;
```

HAVING clause is used after the GROUP BY clause
HAVING clause can contain aggregate function.

```
select EmpName from Employee
GROUP BY EmpName
→ HAVING SUM(EmpSalary) < 20000
```

Page No.
Date / /

⑦ Sub query or Nested query or Inner query in SQL?

A subquery or Inner query or a Nested query within another sql query and embedded within the WHERE clause.

```
SELECT LastName, FirstName
FROM Employees
WHERE OfficeCode IN(SELECT OfficeCode
                    FROM Offices
                    WHERE country="USA");
```
} Inner Query

⑧ Auto Increment / Identity column

Auto-incremented allows a unique number to be generated automatically when a new record is inserted into a table.

Mostly it is the primary key only

```
CREATE TABLE Persons (
    PersonId int IDENTITY(1,1) PRIMARY KEY,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int
);
```

Page No.
Date / /

SQL-JOINS:

What are Joins in SQL

A Join clause is used to COMBINE rows from two or more tables, based on a related column btw them

Types of Joins:

INNER JOIN: Retrieves records that have matching values in both tables involved in the join.

```
Select * FROM Table_A
JOIN Table_B;
```

```
Select * From Table_A
INNER JOIN Table_B
```

LEFT(OUTER)JOIN: Retrieves all the records/rows from the left table and the matched records/rows from the left table.

```
SELECT *
FROM Table_A A
LEF JOIN Table_B B
ON A.col = B.col;
```

```
SELECT Employee.Emp_id, Joining.Joining_Date
FROM Employee
LEFT OUTER JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

RIGHT JOIN

This joins returns all rows from the RIGHT table and its matached rows from a LEFT table

```
SELECT Employee.Emp_id, Joining.Joining_Date
FROM Employee
LEFT OUTER JOIN Joining
ON Employee.Emp_id = Joining.Emp_id
ORDER BY Employee.Emp_id;
```

Date / /

FULL JOIN: This joins return all value there is match either in the RIGHT table or in the LEFT Table

```
SELECT Employee.Emp_Id, Joining.Joining_Date
FROM Employee
FULL OUTER JOIN Joining
ON Employee.Emp-id = Joining.Emp_id
ORDER By Employee.Emp-id;
```

Third-Higlest salary

```
SELECT salary
FROM(
    SELECT salary, DENSE_RANK() OVER(ORDER BY salary DESC) as rank
rank FROM Employees
    )ranked_salaries
    WHERE rank =3
```

Second Higlest salary

```
SELECT MAX(Salary)
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
```

Find duplicate records in a Table

```
SELECT email, COUNT(*)
FROM users
GROUP BY email
HAVING COUNT(*)>1;
```