

# Database Management System

---



# Unit 6:Database Management System

- Introduction to Database, components and structure of DBMS – logical structure – the 3 level architecture and mapping among them. Comparison between traditional file based system and DBMS. Advantages and drawbacks of DBMS.
- Relational Model - What is relational model, Relational key constraints – candidate key, primary key, foreign key.ER Model – entities, attributes, relationship, and cardinality. Entity types, Entity sets Attributes and Keys Relationship types, Relationship Sets, converting ER diagram to relational tables. Database Schema

# Unit 6:Database Management System

- Database Anomalies, Codd Rules and Normalization theory, 1 NF, 2 NF, 3 NF and BCNF.
- Introduction to transaction and concept of concurrency control. Transaction and system concepts, desirable properties of transactions, transaction support in SQL. Concurrency control techniques, the locking protocol, serializable schedules, locks, 2 phase commit. Techniques, concurrency control based on timestamp ordering





# SQL

- **SQL is a database computer language designed for the retrieval and management of data in a relational databases like MySQL, MS Access, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres etc.**
- **SQL stands for Structured Query Language.**
- **SQL was developed in the 1970s by IBM Computer Scientists.**





# SQL

- **SQL is not a database management system, but it is a query language which is used to store and retrieve the data from a database or in simple words SQL is a language that communicates with databases.**
- **SQL was developed in the 1970s by IBM Computer Scientists and became a standard of the American National Standards Institute (ANSI) in 1986, and the International Organization for Standardization (ISO) in 1987.**





# SQL

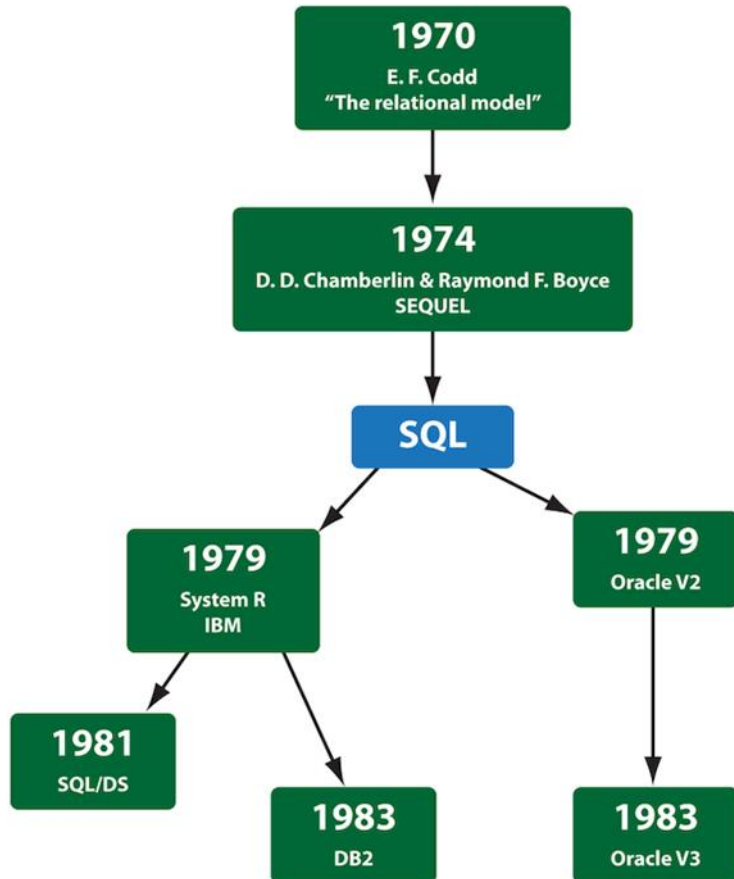
- Though SQL is an ANSI (American National Standards Institute) standard language, but there are many different dialects of the SQL language like MS SQL Server is using T-SQL and Oracle is using PL/SQL.







# BRIEF HISTORY OF SQL



- 1970 – Dr. Edgar F. "Ted" Codd of IBM is known as the father of relational databases. He described a relational model for databases.
- 1974 – Structured Query Language (SQL) appeared.
- 1978 – IBM worked to develop Codd's ideas and released a product named System/R.
- 1986 – IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software which later came to be known as Oracle.
- 1987 – SQL became the part of the International Organization for Standardization (ISO).





# SQL

- Though SQL is an ANSI (American National Standards Institute) standard language, but there are many different dialects of the SQL language like MS SQL Server is using T-SQL and Oracle is using PL/SQL.







# SQL

Data type	Meaning	Example
CHAR (n)	Fixed length character string. 'n' is the number of characters.	CHAR(5):“Ashok” “Vijay”
VARCHAR(n)	Variable length character string. 'n' is the maximum number of characters in the string.	VARCHAR(15): “Vijay Kumar” “Ashok Sen”
DATE	Date in the form of YYYY-MM-DD	DATE: '2014-03-20'
INTEGER	Integer number	23 56789





# SQL

Data type	Meaning	Example
DECIMAL (m, d)	Fixed point number m represents the number of significant digits that are stored for values and d represents the number of digits that can be stored following the decimal point. If d is zero or not specified then the value does not contains any decimal part.	DECIMAL(5,2) : 999.99 -567.78 DECIMAL (5) : 23456 99999



# SQL BASIC COMMANDS

- The standard SQL commands to interact with relational databases are **CREATE**, **SELECT**, **INSERT**, **UPDATE**, **DELETE** and **DROP**.





# SQL BASIC COMMANDS

## DDL - Data Definition Language

S.N.	Command & Description
1	<b>CREATE</b> Creates a new table, a view of a table, or other object in the database.
2	<b>ALTER</b> Modifies an existing database object, such as a table.
3	<b>DROP</b> Deletes an entire table, a view of a table or other objects in the database.





# SQL BASIC COMMANDS

## DML - Data Manipulation Language

S.N.	Command & Description
1	<b>SELECT</b> Retrieves certain records from one or more tables.
2	<b>INSERT</b> Creates a record.
3	<b>UPDATE</b> Modifies records.
4	<b>DELETE</b> Deletes records.





# SQL BASIC COMMANDS

## DCL - Data Control Language

S.N.	Command & Description
1	<b>GRANT</b> Gives a privilege to user.
2	<b>REVOKE</b> Takes back privileges granted from user.







# SQL BASIC COMMANDS

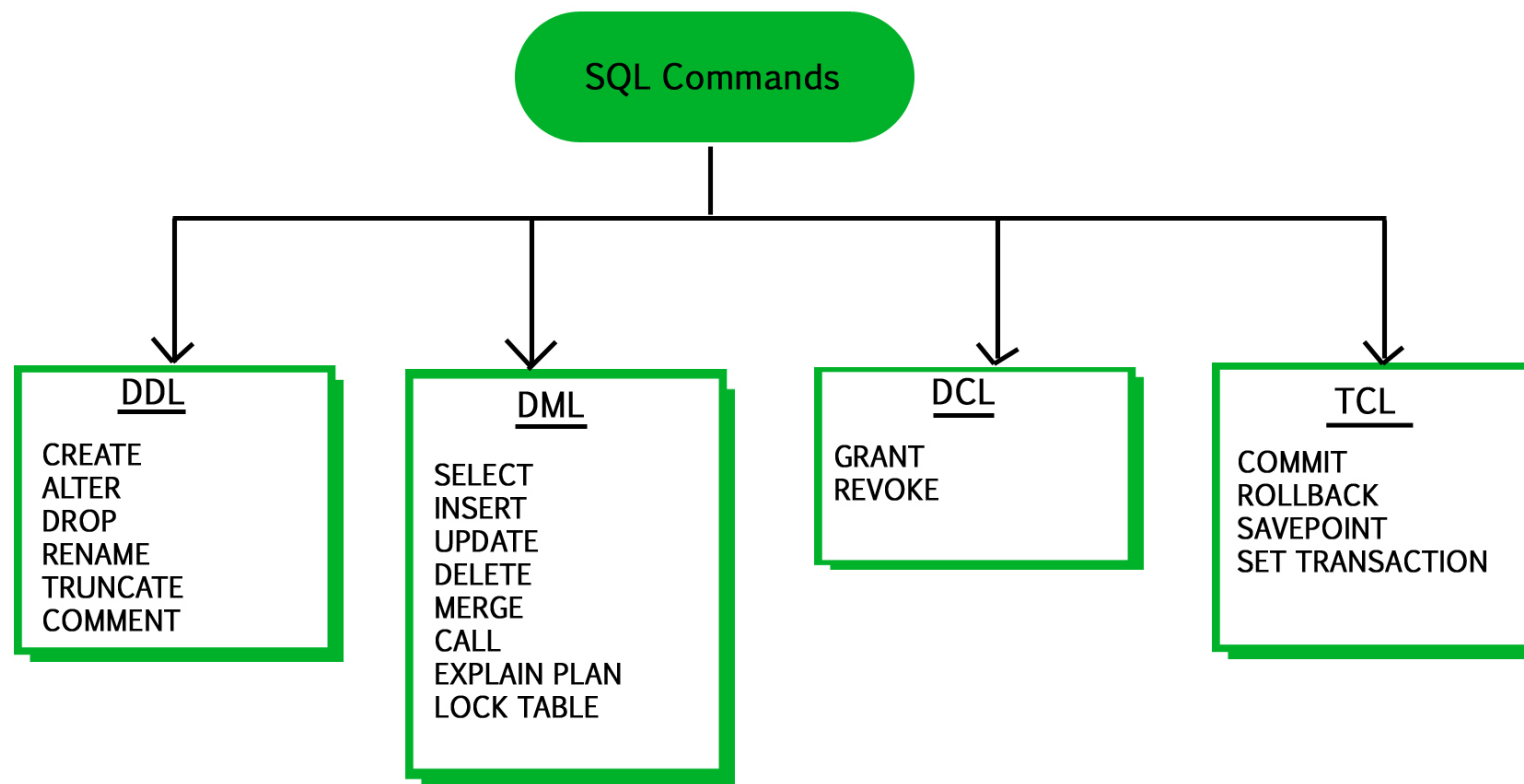
## TCL (Transaction Control Language) :

Transaction Control Language commands are used to manage transactions in the database. These are used to manage the changes made by DML-statements.





# SQL BASIC COMMANDS



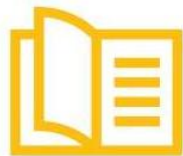


# SQL BASIC COMMANDS



Create

C



Read

R



Update

U



Delete

D

C ( CREATE )

INSERT

R ( READ )

SELECT

U ( UPDATE )

UPDATE

D ( DELETE )

DELETE





**Q. Which of the following is not a valid SQL type?**

- A. FLOAT**
- B. NUMERIC**
- C. DECIMAL**
- D. CHARACTER**





**Q. Which of the following is not a DDL command?**

- A. TRUNCATE**
- B. ALTER**
- C. CREATE**
- D. UPDATE**





**Q. Which of the following are TCL commands?**

- a) UPDATE and TRUNCATE**
- b) SELECT and INSERT**
- c) GRANT and REVOKE**
- d) ROLLBACK and SAVEPOINT**

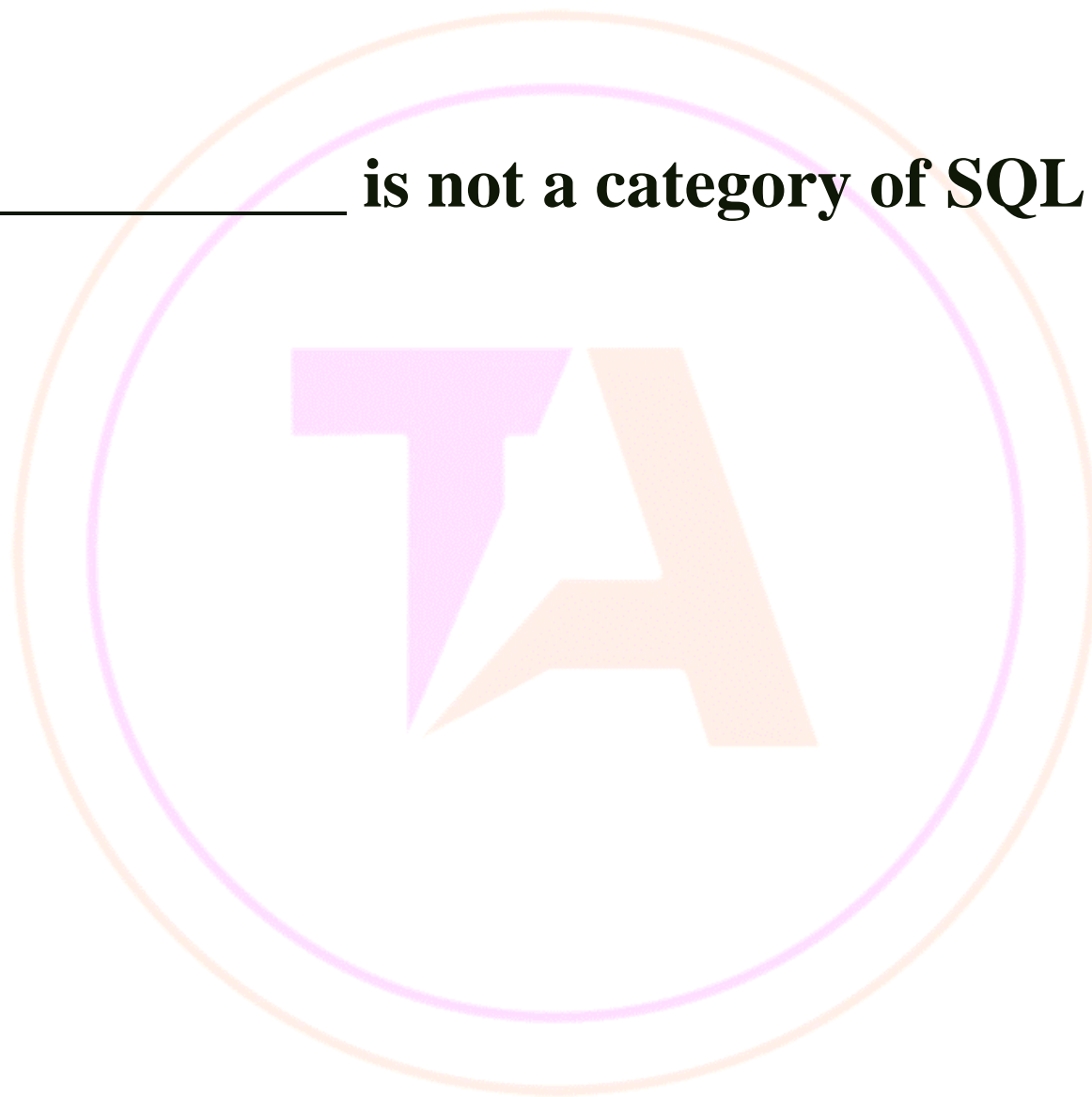






Q. \_\_\_\_\_ is not a category of SQL command.

- a) TCL
- b) SCL
- c) DCL
- d) DDL





# Some of The Most Important SQL Commands

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

sql keywords





# SELECT STATEMENT

- The SELECT statement is used to select data from a database.

## SELECT Syntax

```
SELECT column1, column2, ...  
FROM table_name;
```

Here, column1, column2, ... are the field names of the table you want to select data from. If you want to select all the fields available in the table, use the following syntax:

```
SELECT * FROM table_name;
```





# WHERE CLAUSE

- The WHERE clause is used to filter records.

## WHERE Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```





# AND, OR AND NOT OPERATORS

- The WHERE clause can be combined with AND, OR, and NOT operators.
- The AND and OR operators are used to filter records based on more than one condition:
- The AND operator displays a record if all the conditions separated by AND are TRUE.
- The OR operator displays a record if any of the conditions separated by OR is TRUE.
- The NOT operator displays a record if the condition(s) is NOT TRUE.





## **ORDER BY KEYWORD**

- **The ORDER BY keyword is used to sort the result-set in ascending or descending order.**
- **The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.**







# ORDER BY KEYWORD

## ORDER BY Syntax

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```





# INSERT INTO STATEMENT

- The **INSERT INTO** statement is used to insert new records in a table.
- It is possible to write the **INSERT INTO** statement in two ways:
  1. Specify both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```





# INSERT INTO STATEMENT

2. If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. However, make sure the order of the values is in the same order as the columns in the table. Here, the INSERT INTO syntax would be as follows:

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```





# UPDATE STATEMENT

The **UPDATE** statement is used to modify the existing records in a table.

UPDATE Syntax

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```





# LIMIT CLAUSE

The **LIMIT** clause is used to specify the number of records to return.

LIMIT Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;
```





# MIN() AND MAX() FUNCTIONS

- The MIN() function returns the smallest value of the selected column.
- The MAX() function returns the largest value of the selected column.

## MIN() Syntax

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

## MAX() Syntax

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```







# COUNT(), AVG() AND SUM() FUNCTIONS

- The COUNT() function returns the number of rows that matches a specified criterion.

COUNT() Syntax

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```





# COUNT(), AVG() AND SUM() FUNCTIONS

- The AVG() function returns the average value of a numeric column.

AVG() Syntax

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```





# COUNT(), AVG() AND SUM() FUNCTIONS

- The SUM() function returns the total sum of a numeric column.

SUM() Syntax

```
SELECT SUM(column_name)
FROM table_name
WHERE condition;
```





**Q. Which of these query will display the the table given above ?**

- a) Select employee from name**
- b) Select name**
- c) Select name from employee**
- d) Select employee**

---

Name

---

Annie

---

Bob

---

Callie

---

Derek

---





**Q. Here which of the following displays the unique values of the column?**

```
SELECT _____ dept_name  
FROM instructor;
```

- a) All
- b) From
- c) Distinct
- d) Name



Employee_id	Name	Salary
1001	Annie	6000
1009	Ross	4500
1018	Zeith	7000

**Q. This is Employee table.**

**Which of the following employee\_id will be displayed for the given query?**

**SELECT \* FROM employee WHERE employee\_id > 1009;**

- a) 1009, 1001, 1018
- b) 1009, 1018
- c) 1001
- d) 1018





**Q. SELECT name  
FROM instructor  
WHERE dept name = 'Physics'  
ORDER BY name;**

**By default, the order by clause lists items in \_\_\_\_\_ order.**

- a) Descending**
- b) Any**
- c) Same**
- d) Ascending**







**Q. Which of the following are the synonyms for Column and ROW of a table?**

**Row = [Tuple, Record]**

**Column = [Field, Attribute]**

**Row = [Tuple, Attribute]**

**Columns = [Field, Record]**

- A. 1 and 2**
- B. 3 and 4**
- C. Only 1**
- D. Only 2**





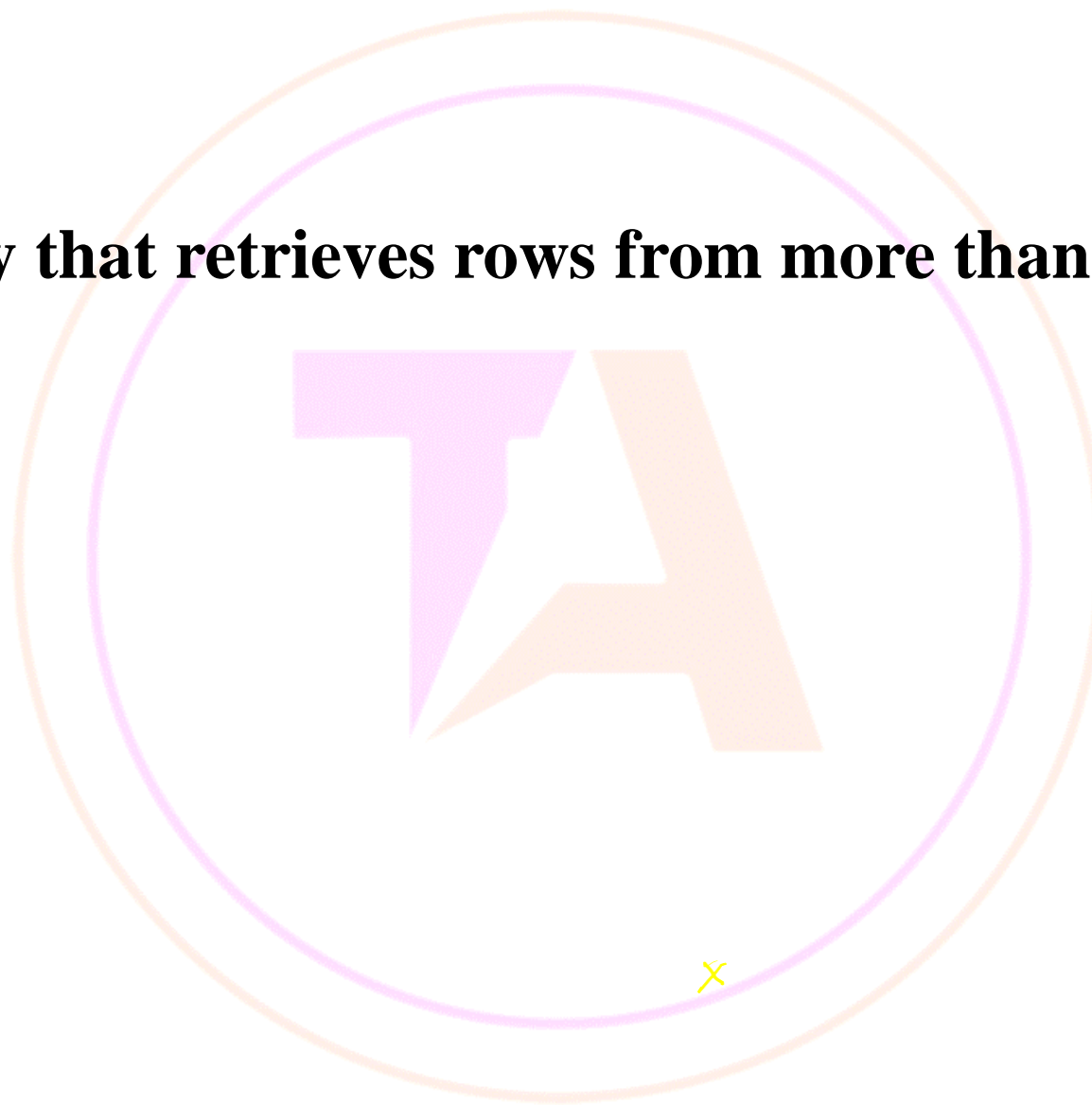
**Q. A\_\_\_\_\_ is a query that retrieves rows from more than one table or view**

**A. Start**

**B. End**

**C. Join**

**D. All of the above**





## Different Types of SQL JOINS

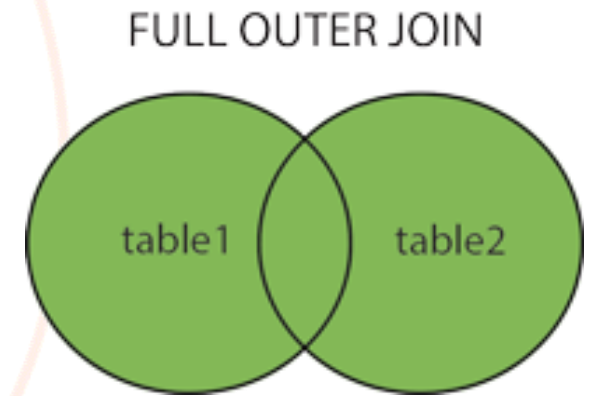
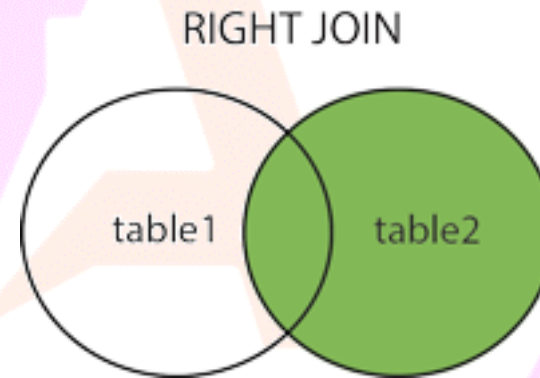
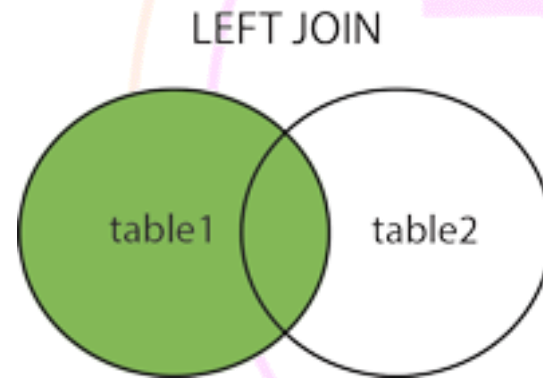
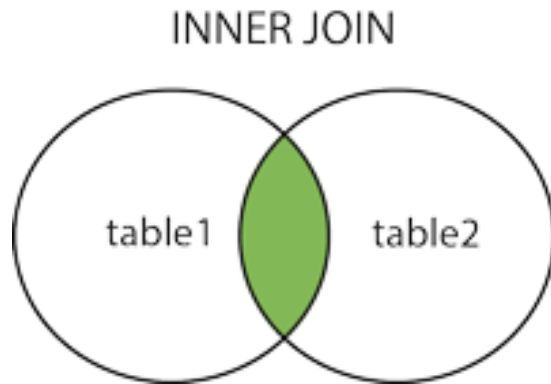
Here are the different types of the JOINS in SQL:

- ✓ **(INNER) JOIN:** Returns records that have matching values in both tables
- ✓ **LEFT (OUTER) JOIN:** Returns all records from the left table, and the matched records from the right table
- ✓ **RIGHT (OUTER) JOIN:** Returns all records from the right table, and the matched records from the left table
- ✓ **FULL (OUTER) JOIN:** Returns all records when there is a match in either left or right table





# Different Types of SQL JOINS





## A. INNER JOIN

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.

Syntax:

```
SELECT table1.column1,table1.column2,table2.column1,....  
FROM table1  
INNER JOIN table2  
ON table1.matching_column = table2.matching_column;
```

table1: First table.

table2: Second table

matching\_column: Column common to both the tables.





## A. INNER JOIN

```
SELECT StudentCourse.COURSE_ID, Student.NAME, Student.AGE FROM  
Student  
INNER JOIN StudentCourse  
ON Student.ROLL_NO = StudentCourse.ROLL_NO;
```

ROLL_NO	NAME	ADDRESS	PHONE	Age
1	HARSH	DELHI	XXXXXXXXXX	18
2	PRATIK	BIHAR	XXXXXXXXXX	19
3	RIYANKA	SILIGURI	XXXXXXXXXX	20
4	DEEP	RAMNAGAR	XXXXXXXXXX	18
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19
6	DHANRAJ	BARABAJAR	XXXXXXXXXX	20
7	ROHIT	BALURGHAT	XXXXXXXXXX	18
8	NIRAJ	ALIPUR	XXXXXXXXXX	19

Student

COURSE_ID	ROLL_NO
1	1
2	2
2	3
3	4
1	5
4	9
5	10
4	11

StudentCourse

COURSE_ID	NAME	Age
1	HARSH	18
2	PRATIK	19
2	RIYANKA	20
3	DEEP	18
1	SAPTARHI	19







## B. LEFT JOIN

```
SELECT Student.NAME, StudentCourse.COURSE_ID  
FROM Student  
LEFT JOIN StudentCourse  
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

ROLL_NO	NAME	ADDRESS	PHONE	Age
1	HARSH	DELHI	XXXXXXXXXX	18
2	PRATIK	BIHAR	XXXXXXXXXX	19
3	RIYANKA	SILIGURI	XXXXXXXXXX	20
4	DEEP	RAMNAGAR	XXXXXXXXXX	18
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19
6	DHANRAJ	BARABAJAR	XXXXXXXXXX	20
7	ROHIT	BALURGHAT	XXXXXXXXXX	18
8	NIRAJ	ALIPUR	XXXXXXXXXX	19

Student

COURSE_ID	ROLL_NO
1	1
2	2
2	3
3	4
1	5
4	9
5	10
4	11

StudentCourse

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
DHANRAJ	NULL
ROHIT	NULL
NIRAJ	NULL







## C. RIGHT JOIN

```
SELECT Student.NAME, StudentCourse.COURSE_ID  
FROM Student  
RIGHT JOIN StudentCourse  
ON StudentCourse.ROLL_NO = Student.ROLL_NO;
```

ROLL_NO	NAME	ADDRESS	PHONE	Age
1	HARSH	DELHI	XXXXXXXXXX	18
2	PRATIK	BIHAR	XXXXXXXXXX	19
3	RIYANKA	SILIGURI	XXXXXXXXXX	20
4	DEEP	RAMNAGAR	XXXXXXXXXX	18
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19
6	DHANRAJ	BARABAJAR	XXXXXXXXXX	20
7	ROHIT	BALURGHAT	XXXXXXXXXX	18
8	NIRAJ	ALIPUR	XXXXXXXXXX	19

Student

COURSE_ID	ROLL_NO
1	1
2	2
2	3
3	4
1	5
4	9
5	10
4	11

StudentCourse

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
NULL	4
NULL	5
NULL	4





**Q. Which of the following statement is TRUE about FULL OUTER JOIN created on two tables Table1 and Table2?**

- A. Retrieves all the unmatched rows of Table1**
- B. Retrieves all the unmatched rows of Table2**
- C. Retrieves both matched and unmatched rows of Table1 and Table2**
- D. Retrieves only matched rows of table1 and Table2**





**Q. Which join is equivalent to Cartesian Product?**

- A. INNER JOIN**
- B. OUTER JOIN**
- C. CROSS JOIN**
- D. NATURAL JOIN**



## Natural Join :

**Natural Join joins two tables based on same attribute name and datatypes. The resulting table will contain all the attributes of both the tables but only one copy of each common column.**

Roll_No	Name
1	A
2	B
3	C

Roll_No	Marks
2	70
3	50
4	85



**Natural Join :**

**SELECT \***  
**FROM Student S NATURAL JOIN Marks M;**

Roll_No	Name	Marks
2	B	70
3	C	50





## **Cross Join :**

**Cross Join will produce cross or Cartesian product of two tables if there is no condition specifies. The resulting table will contain all the attributes of both the tables including duplicate or common columns also.**





**Cross Join :**

**SELECT \***

**FROM Student S CROSS JOIN Marks M;**

Roll_No	Name	Roll_No	Marks
1	A	2	70
1	A	3	50
1	A	4	85
2	B	2	70
2	B	3	50
2	B	4	85
3	C	2	70
3	C	3	50
3	C	4	85







**Q. Joining a table with itself is called**

- A. Self Join**
- B. Outer Join**
- C. Join**
- D. Full join**





**Q. Which of the following Relational Algebra operations can be used when you want to keep all the tuples of the first relation irrespective of whether or not they have matching tuples in the second relation?**

- A. Natural join**
- B. Left-outer join**
- C. Right-outer-join**
- D. Theta join**





**Q. Which of the following is/are the type(s) of Outer Join?**

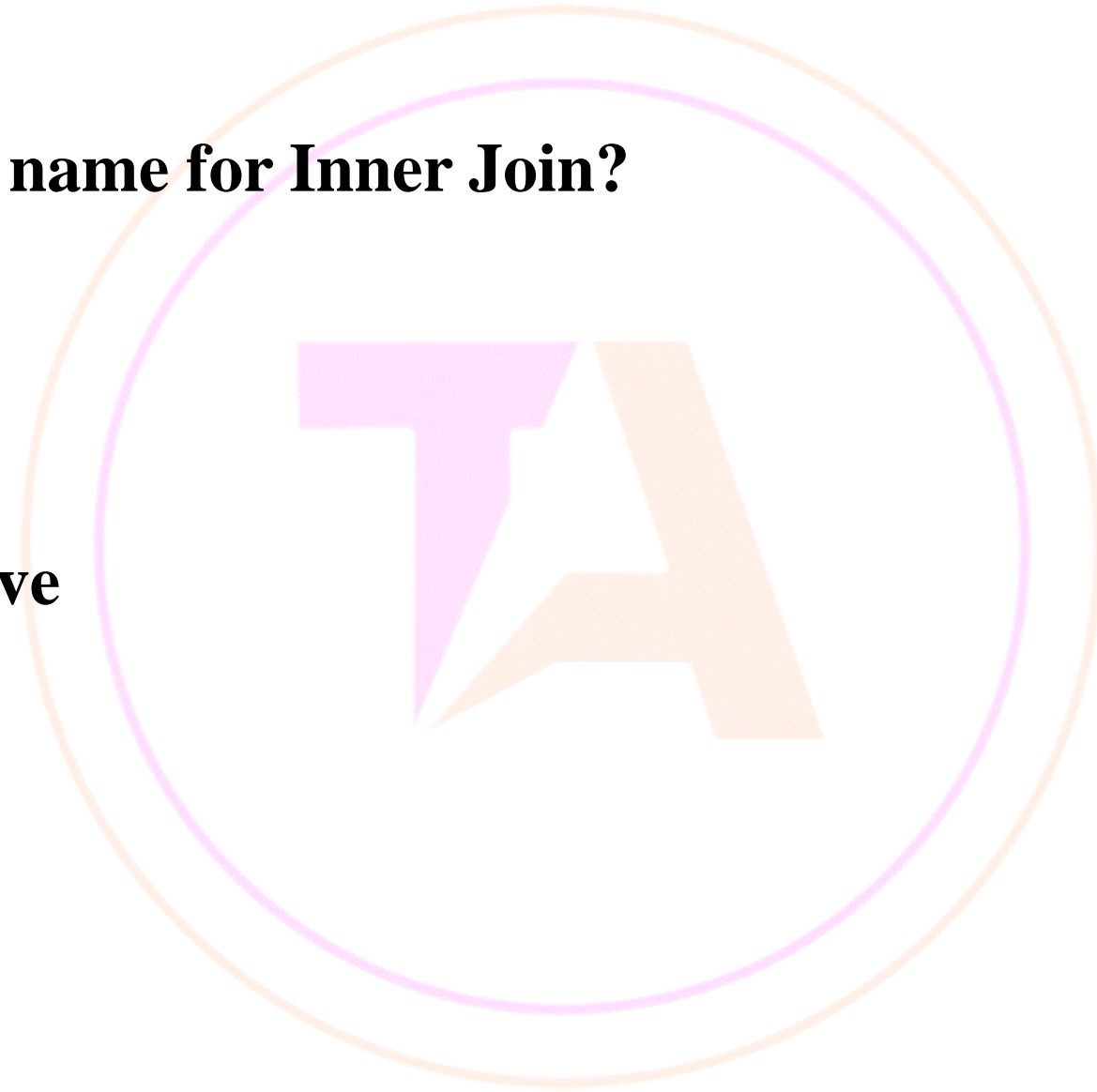
- A. Left Outer Join**
- B. Right Outer Join**
- C. Full Outer Join**
- D. All of the above**





**Q. What is another name for Inner Join?**

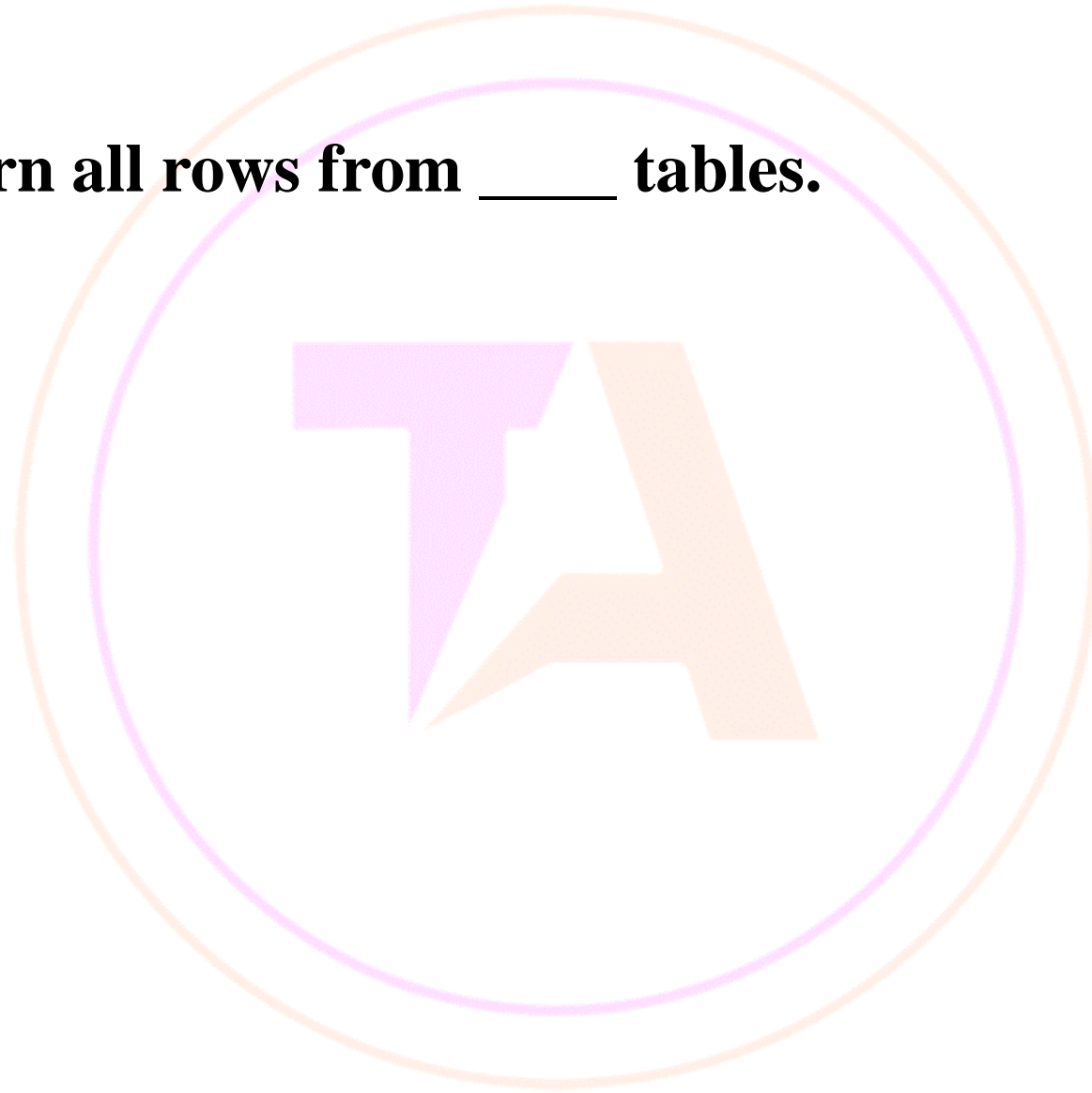
- A. Simple Join**
- B. Natural Join**
- C. Both A and B**
- D. None of the above**





**Q. Outer joins return all rows from \_\_\_\_ tables.**

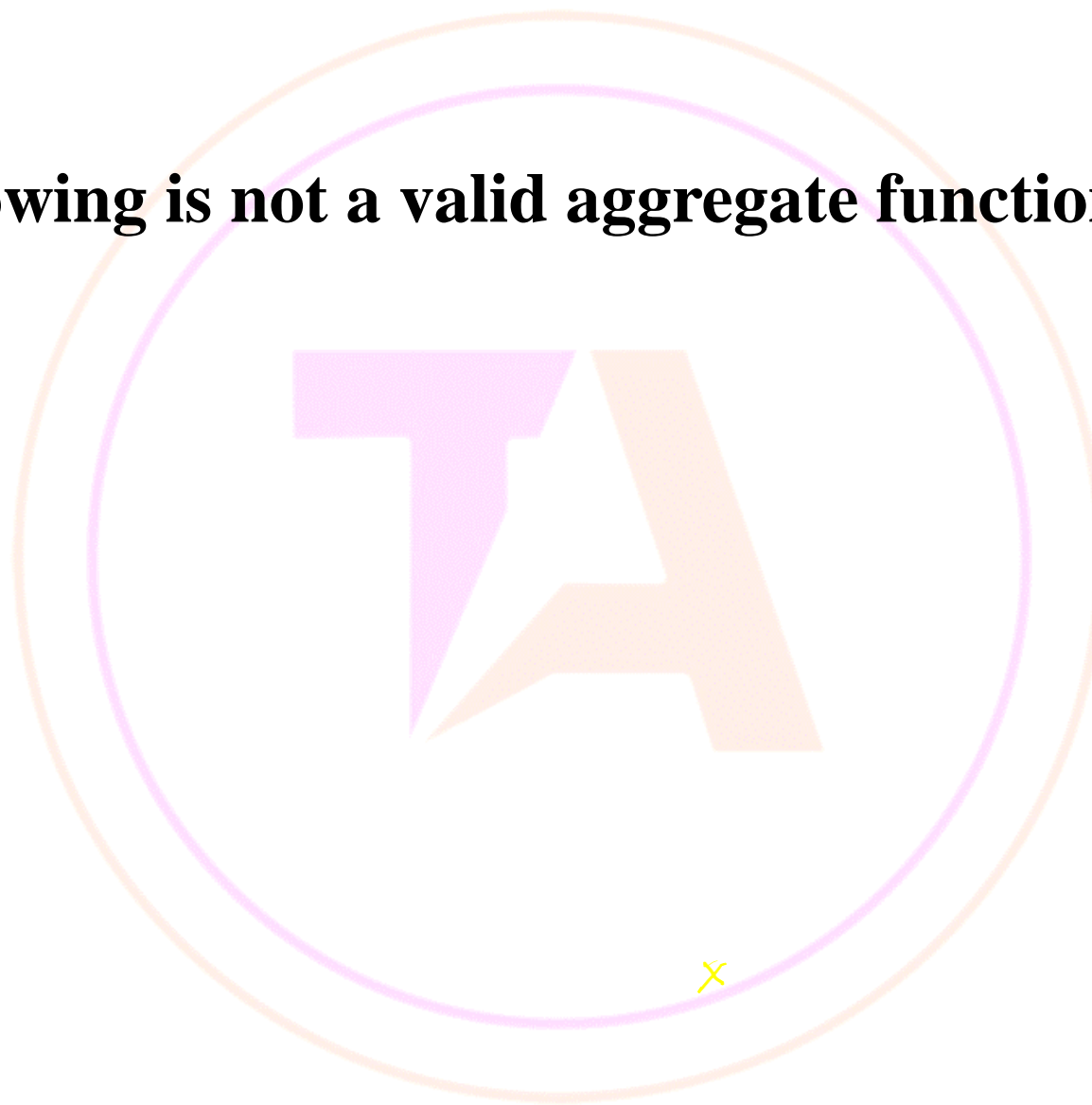
- A. First**
- B. Second**
- C. Both**
- D. None**





**Q. Which of the following is not a valid aggregate function?**

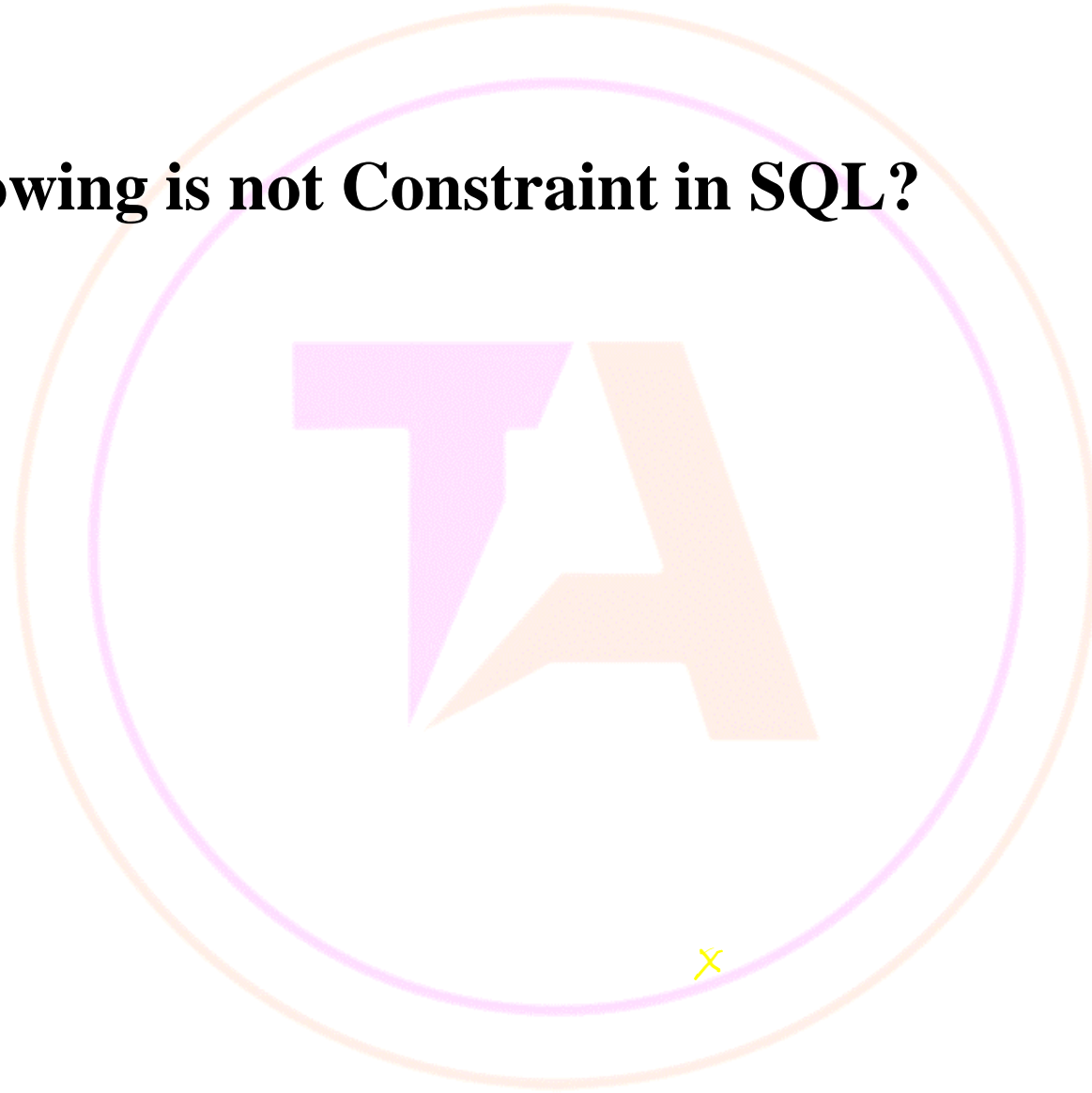
- A. COUNT**
- B. COMPUTE**
- C. SUM**
- D. MAX**





**Q. Which of the following is not Constraint in SQL?**

- A. Primary Key**
- B. Not Null**
- C. Check**
- D. Union**







**Q. SELECT name  
FROM instructor  
WHERE salary IS NOT NULL;  
Selects?**

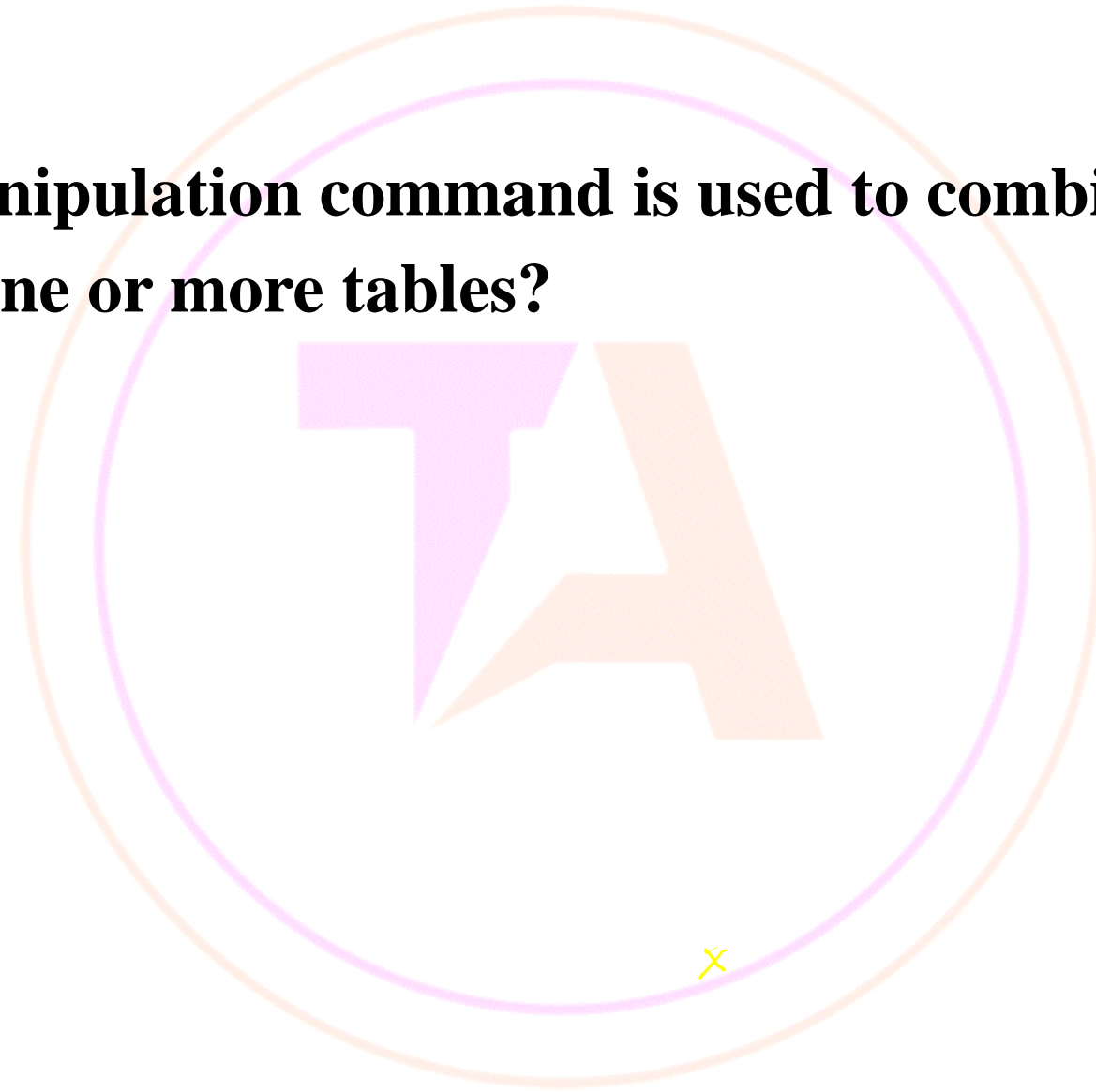
- a) Tuples with null value**
- b) Tuples with no null values**
- c) Tuples with any salary**
- d) All of the mentioned**





**Q. Which data manipulation command is used to combines the records from one or more tables?**

- A. SELECT**
- B. PROJECT**
- C. JOIN**
- D. PRODUCT**





**Q. Which of the following statement is true?**

- A. TRUNCATE free the table space while DELETE does not.**
- B. Both TRUNCATE and DELETE statements free the table's space.**
- C. Both TRUNCATE and DELETE statement does not free the table's space.**
- D. DELETE free the table space while TRUNCATE does not.**



Truncate	Delete
<ul style="list-style-type: none"> <li>• It removes all rows from a table. It is faster &amp; does not use as much undo space as a Delete</li> <li>• It is a DDL command so this command change structure of table</li> <li>• You cannot rollback in Truncate</li> <li>• In SQL, the auto increment counter gets reset with truncate</li> </ul>	<ul style="list-style-type: none"> <li>• It is used to remove rows from table. A WHERE clause can be used to only remove some rows</li> <li>• It is a DML command. It only remove rows from a table, leaving the table structure untouched</li> <li>• In DELETE, you can rollback</li> <li>• The auto increment counter cannot get reset with delete</li> </ul>



**Q. Which of the following is true about the HAVING clause?**

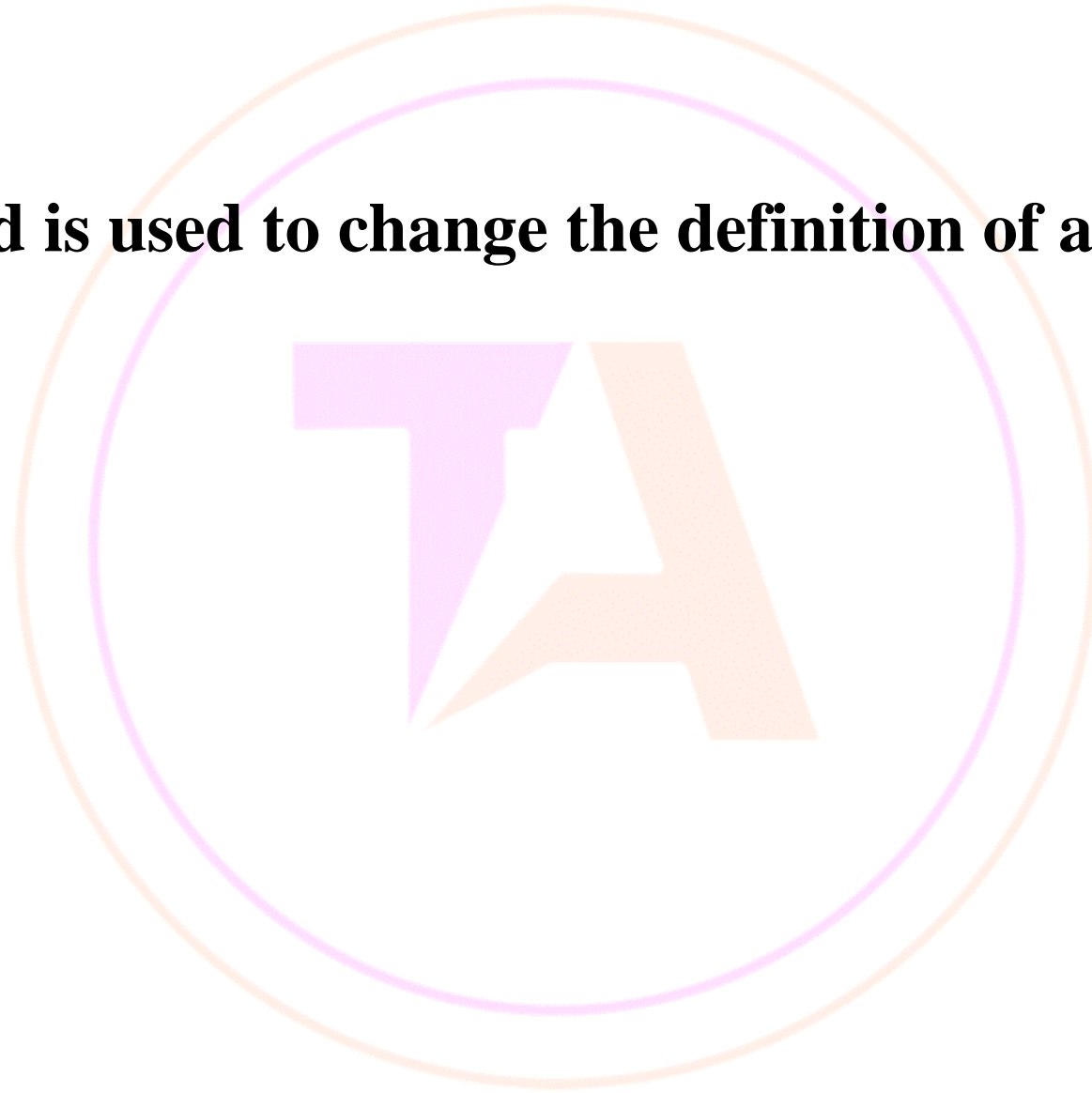
- A. Similar to the WHERE clause but is used for columns rather than groups.**
- B. Similar to WHERE clause but is used for rows rather than columns.**
- C. Similar to WHERE clause but is used for groups rather than rows.**
- D. Acts exactly like a WHERE clause.**





**Q. Which command is used to change the definition of a table in SQL?**

- A. CREATE**
- B. UPDATE**
- C. ALTER**
- D. SELECT**







**Q. Which of the following is the correct order of a SQL statement?**

- A. SELECT, GROUP BY, WHERE, HAVING**
- B. SELECT, WHERE, GROUP BY, HAVING**
- C. SELECT, HAVING, WHERE, GROUP BY**
- D. SELECT, WHERE, HAVING, GROUP BY**





**Q. Which of the following statement is correct regarding the difference between TRUNCATE, DELETE and DROP command?**



- I. DELETE operation can be rolled back but TRUNCATE and DROP operations cannot be rolled back.**
- II. TRUNCATE and DROP operations can be rolled back but DELETE operations cannot be rolled back.**
- III. DELETE is an example of DML, but TRUNCATE and DROP are examples of DDL.**
- IV. All are an example of DDL.**

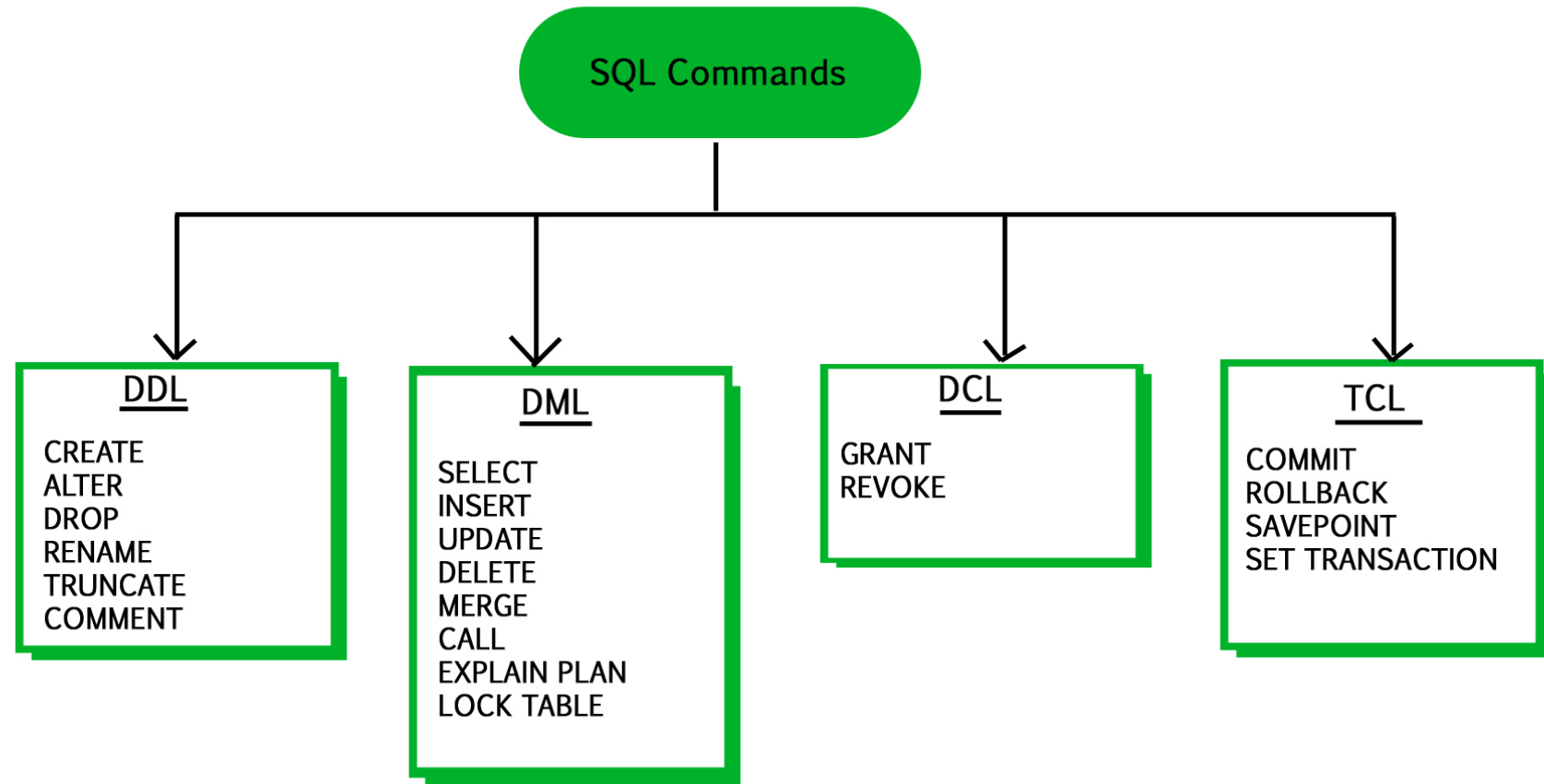
- A. I and III**
- B. II and III**
- C. II and IV**
- D. II and IV**





Aspect	TRUNCATE	DELETE	DROP
<b>Description</b>	Removes all data from a table	Removes specific rows from a table	Removes an entire table
<b>Rollback</b>	Cannot be rolled back	Can be rolled back	Cannot be rolled back
<b>WHERE Clause</b>	Not applicable	Optional	Not applicable
<b>Table Structure</b>	Preserves table structure	Preserves table structure	Removes table structure







**Q. Which command undo all the updates performed by the SQL in the transaction?**

- A. ROLLBACK**
- B. COMMIT**
- C. TRUNCATE**
- D. DELETE**





**Q. Which of the following command makes the updates performed by the transaction permanent in the database?**

- A. ROLLBACK**
- B. COMMIT**
- C. TRUNCATE**
- D. DELETE**





**Q. Table Employee has 10 records. It has a non-NULL SALARY column which is also UNIQUE.**

**The SQL statement**

**SELECT COUNT(\*) FROM Employee WHERE SALARY > ANY (SELECT SALARY FROM EMPLOYEE);**

**prints**

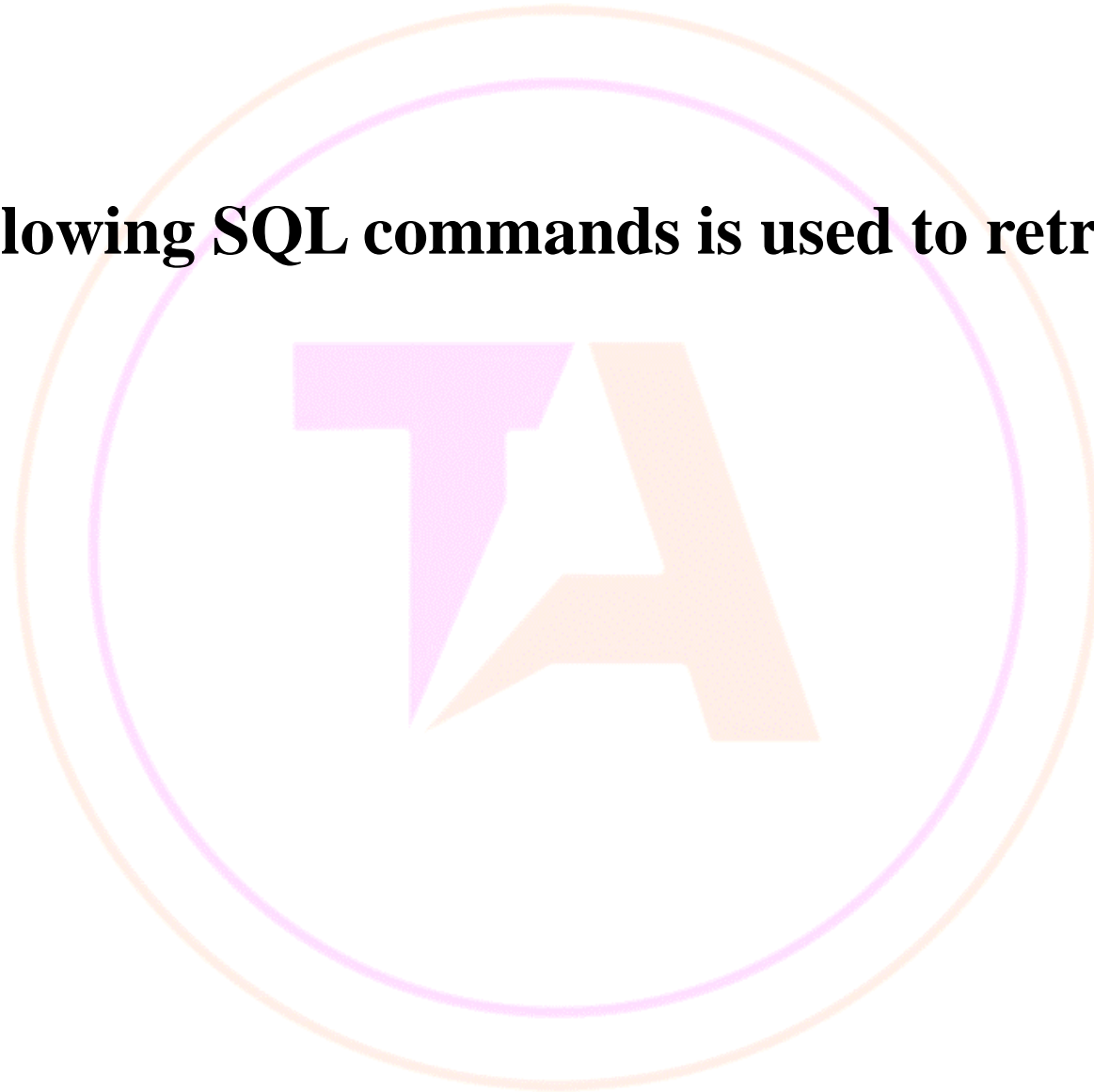
- A. 10**
- B. 9**
- C. 5**
- D. 0**





**Q. Which of the following SQL commands is used to retrieve data?**

- A. DELETE**
- B. INSERT**
- C. SELECT**
- D. JOIN**







**Q. What is a view?**

- A. A view is a special stored procedure executed when certain event occurs.**
- B. A view is a virtual table which results of executing a pre-compiled query. A view is not part of the physical database schema, while the regular tables are.**
- C. A view is a database diagram.**
- D. None of these**





**Q. Let X (a, b, c, d) and Y (p, q, r, s) be two relations in which a is the foreign key of X that refers to the primary key of Y. Which of the following operation causes the referential integrity constraint violation?**

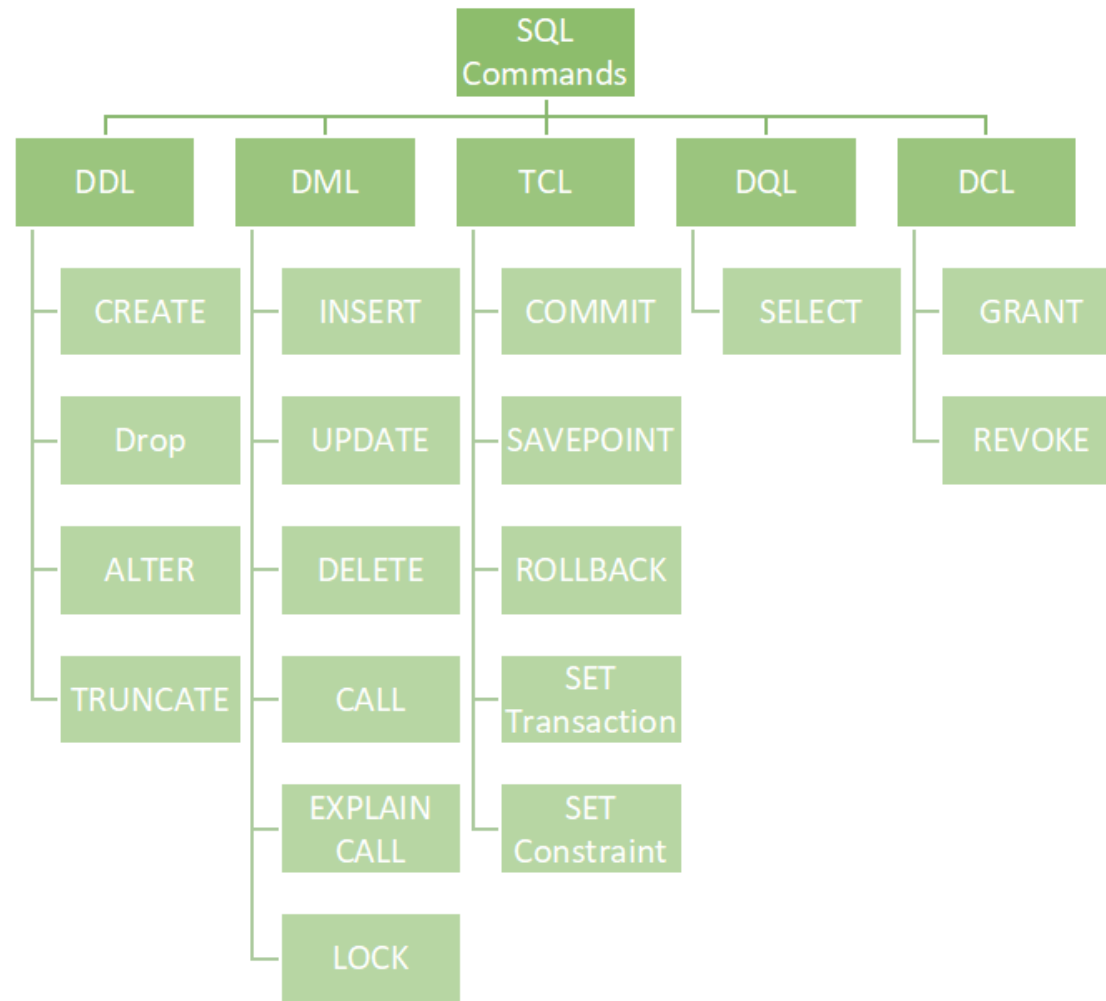
- A. Insert into Y**
- B. Delete from X**
- C. Insert into X**
- D. Both 3 and 5**
- E. Delete from Y**



**Q. Which of the following represents the three basic SQL DML commands?**

- A. Drop, Update, Delete**
- B. Insert, Alter, Delete**
- C. Insert, Update, Create**
- D. Insert, Update, Delete**





# Transaction in DBMS

When the data of users is stored in a database, that data needs to be accessed and modified from time to time. This task should be performed with a specified set of rules and in a systematic way to maintain the consistency and integrity of the data present in a database. In DBMS, this task is called a transaction. It is similar to a bank transaction, where the user requests to withdraw some amount of money from his account. Subsequently, several operations take place such as fetching the user's balance from the database, subtracting the desired amount from it, and updating the user's account balance. This series of operations can be called a transaction.



# What does a Transaction mean in DBMS?

Transaction in Database Management Systems (DBMS) can be defined as a set of logically related operations. It is the result of a request made by the user to access the contents of the database and perform operations on it. It consists of various operations and has various states in its completion journey. It also has some specific properties that must be followed to keep the database consistent.



# Operations of Transaction

A user can make different types of requests to access and modify the contents of a database. So, we have different types of operations relating to a transaction. They are discussed as follows:

## i) Read(X)

A read operation is used to read the value of X from the database and store it in a buffer in the main memory for further actions such as displaying that value. Such an operation is performed when a user wishes just to see any content of the database and not make any changes to it. For example, when a user wants to check his/her account's balance, a read operation would be performed on user's account balance from the database.





# Operations of Transaction

## ii) Write(X)

A write operation is used to write the value to the database from the buffer in the main memory. For a write operation to be performed, first a read operation is performed to bring its value in buffer, and then some changes are made to it, e.g. some set of arithmetic operations are performed on it according to the user's request, then to store the modified value back in the database, a write operation is performed.

For example, when a user requests to withdraw some money from his account, his account balance is fetched from the database using a read operation, then the amount to be deducted from the account is subtracted from this value, and then the obtained value is stored back in the database using a write operation



# Operations of Transaction

## iii) Commit

This operation in transactions is used to maintain integrity in the database. Due to some failure of power, hardware, or software, etc., a transaction might get interrupted before all its operations are completed.

This may cause ambiguity in the database, i.e. it might get inconsistent before and after the transaction. To ensure that further operations of any other transaction are performed only after work of the current transaction is done, a commit operation is performed to the changes made by a transaction permanently to the database.



# Operations of Transaction

## iv) Rollback

This operation is performed to bring the database to the last saved state when any transaction is interrupted in between due to any power, hardware, or software failure.

In simple words, it can be said that a rollback operation does undo the operations of transactions that were performed before its interruption to achieve a safe state of the database and avoid any kind of ambiguity or inconsistency.



# Transaction Schedules

When multiple transaction requests are made at the same time, we need to decide their order of execution. Thus, a transaction schedule can be defined as a chronological order of execution of multiple transactions. There are broadly two types of transaction schedules discussed as follows,

## i) Serial Schedule

In this kind of schedule, when multiple transactions are to be executed, they are executed serially, i.e. at one time only one transaction is executed while others wait for the execution of the current transaction to be completed. This ensures consistency in the database as transactions do not execute simultaneously. But, it increases the waiting time of the transactions in the queue, which in turn lowers the throughput of the system, i.e. number of transactions executed per time. To improve the throughput of the system, another kind of schedule are used which has some more strict rules which help the database to remain consistent even when transactions execute simultaneously.



# Transaction Schedules

## ii) Non-Serial Schedule

To reduce the waiting time of transactions in the waiting queue and improve the system efficiency, we use nonserial schedules which allow multiple transactions to start before a transaction is completely executed.

This may sometimes result in inconsistency and errors in database operation. So, these errors are handled with specific algorithms to maintain the consistency of the database and improve CPU throughput as well. Serial Schedules are also sometimes referred to as parallel schedules as transactions execute in parallel in this kind of schedules.



## Serializable

**Serializability in DBMS is the property of a nonserial schedule that determines whether it would maintain the database consistency or not. The nonserial schedule which ensures that the database would be consistent after the transactions are executed in the order determined by that schedule is said to be Serializable Schedules. The serial schedules always maintain database consistency as a transaction starts only when the execution of the other transaction has been completed under it. Thus, serial schedules are always serializable.**



**A transaction is a series of operations, so various states occur in its completion journey. They are discussed as follows:**

**i) Active**

**It is the first stage of any transaction when it has begun to execute. The execution of the transaction takes place in this state. Operations such as insertion, deletion, or updation are performed during this state. During this state, the data records are under manipulation and they are not saved to the database, rather they remain somewhere in a buffer in the main memory.**





**A transaction is a series of operations, so various states occur in its completion journey. They are discussed as follows:**

## **ii) Partially Committed**

**This state of transaction is achieved when it has completed most of the operations and is executing its final operation. It can be a signal to the commit operation, as after the final operation of the transaction completes its execution, the data has to be saved to the database through the commit operation. If some kind of error occurs during this state, the transaction goes into a failed state, else it goes into the Committed state.**



**A transaction is a series of operations, so various states occur in its completion journey. They are discussed as follows:**

**iii) Committed**

**This state of transaction is achieved when all the transaction-related operations have been executed successfully along with the Commit operation, i.e. data is saved into the database after the required manipulations in this state. This marks the successful completion of a transaction.**



**A transaction is a series of operations, so various states occur in its completion journey. They are discussed as follows:**

#### **iv) Failed**

**If any of the transaction-related operations cause an error during the active or partially committed state, further execution of the transaction is stopped and it is brought into a failed state. Here, the database recovery system makes sure that the database is in a consistent state.**



**A transaction is a series of operations, so various states occur in its completion journey. They are discussed as follows:**

**v) Aborted**

**If the error is not resolved in the failed state, then the transaction is aborted and a rollback operation is performed to bring database to the the last saved consistent state. When the transaction is aborted, the database recovery module either restarts the transaction or kills it.**



