SELECT WITH EXISTS syntax :

```
SELECT columns
FROM table1
WHERE
EXISTS (SELECT columns FROM table2 WHERE condition);
```

If any single record Exists
Then
Parent command show results

NOT EXISTS

```
SELECT columns
FROM table1
WHERE
NOT EXISTS (SELECT columns FROM table2 WHERE condition);
```

If not any single Record Exists
Then
Parent command show result.

Q. Define SQL ?

- Structural Query Language
- We use to create and manage the databases.
- It provide us with various built in fn that further help us to manage our database. we can say that SQL is a database transaction Language.
- It is an open source lang. and is free to use with a hug developer community of its own.

Q. What is RDBMS?

- stands for Relational Database Management System. It is an integral part of all the commercially functional databases.
- It essentially allows the ACID principles which help us maintain the integrity data.

Q. Some uses of SQL?

1) It helps us to manage the data efficiently.
2) Allows us to create databases.
3) Allows us to update the table by intersection and deletion of data.
4) Provide us with the data security and integrity.
5) Compatible with almost all technologies using various packages and plugins.

Q. What is the full form of ACID property.

- The ACID property is integral to all the commercial database.
- ACID stand for Atomicity, consistancy, Isolation & Durability.
- When any database follows all these parameters, it becomes fit for any commercial appln.

Q. What is a Query?

- A query is any statement we execute using SQL as the Language on our database to get the results.

- The query helps us to define what data we require. In the case of SQL, we only need to specify what output we require.

- The path of obtaining the result is set by the application itself in the most efficient way.

Q. What is the difference betn database and table

- A database is the collection of various table consisting of data from various resources.

- Table is a single unit located inside a database which is the primary source of data storage.

- Table is made up ~~for~~ of columns & attributes database is made up of tables.

Q. What is select state? Write the Query to get the first five records from the DataFlair table?

- The SELECT statement is the primary building block of any query in SQL.

- The SELECT statement is responsible for fetching the data as specified by the user under the WHERE clause

syntax:

SELECT col1, col2, col3... FROM table_name WHERE condition;

Query to find first five records from the DataFlair;

```
Use DataFlair;
SELECT * FROM DataFlair LIMIT 5;
```

Q. Define DISTINCT keyword & explain working?

- DISTINCT keyword is benificial to find the unique records within any selected attribute by the user.
- With the help of DISTINCT keyword and other aggregate functions, we can count the no. of unique records in our table.

- Syntax:

```
SELECT DISTINCT (col1), col2, col3,..... FROM tableName;
```

- Try to find the uniques names in our DataFlair Database.

Query:

```
use DataFlair;
SELECT DISTINCT (Name) FROM DataFlair;
```

O/P: only Name table columns will be printed.

Q. What are aggregate fn in SQL? Explain?

- SQL provides us with some built-in mathamatical functions to perform calculations efficiently and accurately.

- Popular aggregate fn:
  1) SUM() : Returns the some of the numerical attributes.
  2) AVG() : —"— Average value of the nume. attri.
  3) MAX() : —"— max value of the nume. attributes
  4) MIN() : —"— min. value of the numerical columns.
  5) COUNT() : —"— count of the records in any column of our table.

Query:

```
Use DataFlair
SELECT SUM (experience) AS total-Experience, AVG(experience)
AS Average-Experience FROM DataFlair;
```

O/P:

total_Experience	Average_Experience
45	5.3571

Q. What is aliasing & why is it necessary?

- Alias in SQL allows us to make the query result readable and easy to understand

- By using aliasing, we can temporarily rename the columns and the table as well to yeild the result in a more prasentable and readble format.

Syntax:

```
SELECT col1 AS alias1, col2 AS alias2, col3,....
FROM tableName WHERE condition;
```

Q. What are the Subqueries and what are the rules associated with them?

- Subquery in SQL is also known as Nested Query / Inner Query / Query within a query.
- In a subquery we nest a query in another query & consider the result produced by the inner query for running the outer query.
- Subquery come in handy when we need multiple filters on our data. With the help of subqueries, we can apply as a many filters as we want.

Syntax:

```
SELECT * FROM (SELECT * FROM tableName WHERE
                                    condition);
```

Q: What are the different types of relationships?

① One-to-one:

In this relationship, two tables are accessible to each other and one column of table1 can be relationship with a column of table2. We can't map more than one column of any of the tables.

② One-to-Many & Many-to-One:

This is the most common relationship in the industry, here ony column of table is related to multiple columns in the other table.

③ Many-to-Many:

We have multiple columns of one table related to multiple tables of the second table.

④ Self Referencing Relationship:

- In this case, the table establishes a relationship with itself when required for use

Q. What is the difference bet^n SQL & MySQL

- SQL is the language for database transictions, which helps us to perform the CRUD operation on our tables.

- MySQL is a GUI tool for running the SQL commands and is a open source tool. MySQL is the interface for running the SQL commands whereas, SQL is the handling language.
- SQL helps us to run queries while MySQL helps us to visualize the query results adequately.

Q. What is PRIMARY KEY?

- The PRIMARY KEY constraint uniquely identifies each record in a table.

- PRIMARY KEY must contain UNIQUE values & cannot contain NULL values.

- A table can have only^ONE primary key; and in the table, this primary key consist of single or multiple columns.

Q. What is foreign key ?

- The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.
- A FOREIGN KEY is a field in one table, that refers to the PRIMARY KEY in another table.
- The table with the foreign key is called the child table and the table with primary key is called referenced or parent table.

Persons Table

PersonID	LastName	firstName	Age
1	Hansan	Ola	30
2	Savendson	Tove	23
3	Pettersen	Kari	20

Orders Table

OrderID	OrderNum.	PersonID
1	77895	3
2	44678	3
3	22458	2
4	24562	1

PRIMARY KEY

FOREIGN KEY

Q. What is the UNIQUE keyword in SQL?

- We use the UNIQUE keyword for the primary key attribute mostly. with the help of UNIQUE keyword, we can ensure that the data in the respective column is non-redundant.
- Also, a unique column help us to store the identification based columns like a contact number or mail-id's of users.
- The UNIQUE constraint ensure the all values in a column are different.

Q. Define BETWEEN clause/operator?

- The BETWEEN operator selects values within a given range. The values can be numbers, text or dates.

- BETWEEN operator inclusive: begin and end values are included.

Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

Q. What are views in SQL?

- Views are the temporary tables created by the user using SQL command which contains the required conditionals.

- With the help of views, we get the flexibility to decide which part of data or how many rows and columns of data we need in the newly created table.

- A view can contain rows and columns from one or multiple tables of the same database

Syntax:

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Q. What is a JOIN in SQL?

- A JOIN clause is used to combine rows from two or more tables, based on a related column betn them

Q. What are the Indexes in SQL?

- Indexes are set to make the queries efficient and are accessible to the system only.
- Indexes are not visible to users.
- Indexes can be created using a single column or by multiple columns.

Q. What are the different types of Indexes in SQL?

- We use the INDEX clause to create an index in our table. the index clause help us to query the data fast.
- 2 types

1) Implicit Indexes:

When the database creates indexes on its own we call them implicit indexes.

2) Composite Index:

- Created by the user, using multiple columns as a constraint.
- We use the composite index to maintain unique identification of data points in the database.

Q. What is the use of Auto Increment in SQL?

- In any database, we require a primary key to identify the data stored in our database.
- Sometimes in our data has no such unique attributes in such cases, we create a user defined attribute & set it to auto-increment so that we can uniquely identify the data in our database.

Q. What is normalization?

- Normalization is the method of organizing data in the tables or database in such a way that the data becomes functional
- we can achieve this state by creating multiple tables or by redefining the relationship bet^n the attributes.
- When the dataset is in the condition of normalization, the inconsistancy & the redundancy is removed and kept in cheack thus making the database better for analysis and deployment applications.

Q. What do you understand by NULL values in SQL?

- When we talk about the databases which are functional in the industry, many times we have a situation where some data is absent.
- In such cases, we use NULL value as a placeholder to avoid any logical error in the large appl^n
- A null value is neither equivalent to zero nor any other value. It simply means the data is not available.

Q- What do you understand by NULL

Q. What do you understand by NOT NULL in SQL? is it the same as NULL values?

- In every database, we have a primary key which is unique and helps us to identify each row. Thus, the primary key column can't have any empty or missing data values.
- To keep this in check we specify our primary key columns to be NOT NULL. i.e they can take null values as entry.
- No, NOT NULL is not similar to NULL values. NOT NULL is a constraint we apply on our attribute while NULL values are the placeholder for the missing data in our columns.

Q. What are the triggers in SQL?

- Triggers are programs which are available in the memory with uniques name made up of SQL queries with we need to fire on our database on and off. Triggers can be made for insert, update and delete statements in SQL. we have two types of trigger

1) Row-level-Triggers
2) Statement-level-Triggers.

Q. What is the SQL Query to display the current date?

- getDate() command.

ex: to get the date today?

SELECT sysdate() as Date_today;

o/p: // 2021-05-13 11:33:08

Q. What is a Datawarehouse & why is it important?

- A Datawarehouse is the collection of various databases which contain loads of data. warehousing is put to use when we need to process large blocks of data for our system execution.

- It helps in online transictions and the data mining process by consolidating the data available to us.

- With the help warehousing, we can easily find datarecords we need in analyzing and studing the pattern around us.

Q. What are user defined functions?

- When we move to industry-level software, most often we observe that the Data available to us is not following any fixed pattern.

- Analysis of such data becomes a task if each time we need to change the pattern.

- To avoid these hindrances, we often try to accomplish such task by defining flexible fn which can easily handle the variance in the data.

- User-defined fn are made from scratch by the user with such a Logic that it applies to the whole of the dataset.

- These can be evoked in the same way as system-defined functions.

Q. Write a query to create an empty table from an existing table?

- To create an empty table using an already existing table:

```
Use DataFlair ;
CREATE TABLE DataFlair_copy AS (SELECT * FROM DataFlair
                                WHERE 23=98);
SELECT * FROM DataFlair_copy;
```

	emp-id	name	location	experience

Q. What is **non-clustered - indexing** in SQL.

- We create non-clustered indexes to aid the process of searching the data in our table. They originate when multiple joins or conditions or various fitters are put to use in a single query.

- Non-clustered index does not affect the physical order of the table & keeps the logical order of data as it is.

- For each SQL table, we can create 999 non-clustered index in a session.

Q. What is a **self join** in SQL?

- A self joins in SQL tables are benificial when we need to join the table with itself. It help us to remove the hierarchy from the table, that is it helps us to convert the table to a flat system.

• A self join is a regular join, but the table is joined itse

syntax:

```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;          //T1 & T2 is different tabl
                          aliases for the same ta
```

Q. Write a Query for to count the number of unique records in a table?

```
Use DataFlair;
SELECT COUNT(distinct(emp_id) as noOfUniqueRecords
                    FROM DataFlair;
```

Q. Write a sql Query to find employees whose name starts with 'A' at DataFlair.

```
Use DataFlair WHERE name LIKE 'A%';
```

Q. What is the difference bet^n ~~REFUF~~ DELETE and TRUNCATE keywords in sql?

TRUNCATE - delete all Rows from our table, thus dislocating the disk space assign to the table.

DELETE - delete only the Rows from the table which fullfill the set condition ~~required~~ provided by the where clause & if no condition is input in the query it deletes all the rows in the table but in this case the disk space is still allocated to the table and is not free to be put the sum other use

Q Define clause in SQL?

. SQL clauses are the built in fn which help us to fetch the desired set of the result by providing a user-defined condition to the SQL query.

. It help us to filter out the rows from the entire set of records contained in our database table. some of the prominent clauses which we use in our queries are. where, having, order by etc.

Q. Write a Query to find the number of employees with experience between 2 to 5 years.

. we can do this by two methods.

Method1: Using the BETWEEN keyword.

```
USE ~~DBIAF1~~ DataFlair;
SELECT count(emp_id) as Experience2to5 FROM DataFlair
WHERE experience BETWEEN 2 AND 5;
```

Method2: Using the IN keyword. Query:

```
USE DataFlair;
SELECT count(emp_id) as Experience2to5 FROM DataFlair.
WHERE experience IN (2,3,4,5);
```

o/p:

Experience2to5
8

Q. Explain in detail OLTP?

OLTP - Online Transactional Processing is a type of Data processing which is focused on transaction oriented tasks and their occurrences.

OLTP deals with the tasks like the inserting, updating, deleting and many more in the small packets of data in database.

It deals with Large number of transictions done at a time by various users around the planet.

examples: 1) Use of online banking
2) Sending text msgs to one another
3) Call centre staff maintaining details of the customer. etc.

#

Q. Write a query to show the working of the SUM() function in SQL?

The SUM() function in SQL is put to use on the integer columns to return the total of all the entries.

```
Use DataFlair;
SELECT SUM(experience) as Total_Experience
FROM DataFlair;
```

o/p =>

Total_Experience
80

Q. What is the IFNULL() fn in SQL?

- IFNULL() fn in SQL is put to the use when we need to convert null value to some other user-specified value.
- IFNULL() fn is available in various version in SQL & MySQL servers.
- With the help of IFNULL(), we can easily perform the basic Data cleaning of our data thus speding up our process of data cleaning.
- If put to proper use it can save us from various computational expenses and increase the efficiency of Large system significantly.

Q. How do you count the number of records in our table?

- COUNT keyword / feature
- it returns no of records.

```
Use DataFlair;
SELECT COUNT(*) as noOfRecords FROM DataFlair;

noOfRecords
17
```

Q. How to find the first 3 characters of the name column for each employee in our table DataFlair?

```
Use DataFlair;
SELECTSUBSTRING(name, 1, 3) as firstThree FROM DataFlair;
```

Q. How to fetch alternate records from a SQL table?

① Even Records

```
Use DataFlair;
SELECT emp_id, name FROM (SELECT *, ROW_NUMBER()
OVER (ORDER BY emp_id) AS r FROM DataFlair) as
result WHERE mod(result.r, 2) = 0;
```

o/p:

	emp_id	name

② Odd Records

```
Use DataFlair;
SELECT emp_id, name FROM (SELECT *, ROW_NUMBER()
OVER (ORDER BY emp_id) AS r FROM DataFlair)
as result WHERE mod (result.r, 2) = 1;
```

	emp_id	name

Q. State the different types of user-defined functions in SQL.

- They are different from each other based on the o/p they send to the system.
- Various types:

1) Scalar function:- Returns the o/p of the i/p provided to them.
2) Inline table-valued fns:- fns provide the tables as the return value.
3) Multi statement value fn:- like the inline fns, these functions also provide the tables as the return value of the defined function.

Q. How to create Table in SQL?

1) Query to create the ~~table~~ Database

```
CREATE DATABASE database_name;
SHOW DATABASE;
USE database_name;
```

2) Query to create the table

```
CREATE TABLE worker(
—
—
—
);
```

3) Query to insert into the Table worker

```
INSERT INTO worker
(—,—,—,—,—,—)
VALUES(—,—,—,—,—,—)
(—,—,—,—,—,—)
(—,—,—,—,—,—);
```

Q. Write an SQL query for fetching "FIRST_NAME" from the WORKER table using <WORKER_NAME> as alias.

Select FIRST_NAME AS WORKER_NAME FROM Worker;

Q. Write a SQL Query for fetching the "FIRST_NAME" from WORKER table in upper case?

Select upper(FIRST_NAME) from worker;

Q. What is an SQL Query for fetching the unique values of the column DEPARTMENT from the WORKER table?

Select distinct DEPARTMENT from worker;

Q. Write an SQL query for printing the first the characters of the column FIRST_NAME.

Select substring(FIRST_NAME, 1, 3) FROM worker;

Q. What is an SQL query for finding the position of the alphabet ('A') in the FIRST_NAME column of Ayushi.

Select INSTR(FIRST_NAME, BINARY 'a') from worker where FIRST_NAME = 'Ayushi';

Q. What is an SQL Query for printing the FIRST_NAME from Worker Table after the removal of white spaces from right side.

Select RTRIM(NAME) FROM Worker;

Q. Write an SQL query for printing the DEPARTMENT from worker Table after the removal of white spaces from the left side.

Select LTRIM(DEPARTMENT) from Worker;

Q. What is an SQL query for fetching the unique values from the DEPARTMENT column and thus printing is the length?

select distinct length(DEPARTMENT) from Worker&;

Q. Write a SQL query for printing the FIRST_NAME at replacing 'A' with 'a'

Select REPLACE(FIRST_NAME, 'a', 'A') from Worker;

Q. What is an SQL query for printing the FIRST_NAME and LAST_NAME into a column named COMPLETE_NAME (A space char should be used)

Select CONCAT(FIRST_NAME, ' ', LAST_NAME) AS 'COMPLETE_NAME' from Worker;

(#)

Q. What is an SQL query for printing all details of the worker table which ordered by FIRST_NAME ascending?

select * from Worker order by FIRST_NAME asc;

Q. Write an SQL query for printing the all details of the worker table which ordered by FIRST_NAME asceding and the DEPARTMENT in descending?

select * from Worker order by FIRST_NAME asc, DEPARTMENT desc

Q. What is an SQL query to print the details of the workers 'NIHARIKA' and 'PRIYANSHA'.

Select * from Worker where FIRST_NAME in ('NIHARIKA','PRIYANSHA');

Q. What is an SQL query printing all details of workers excluding the first names of 'NIHARIKA' and 'PRIYANSHU'.

select * from worker where FIRST_NAME in('NIHARIKA','PRIYANSHU');

Q. Write an SQL query for printing the details of DEPARTMENT names as "Admin".

Select * from Worker where DEPARTMENT Like 'Admin %.';

Q. What is an SQL query for printing the details of workers whose FIRST_NAME contains 'A'?

Select * from Worker where FIRST_NAME Like '%a%';

Q. What is an SQL query for printing the FIRST_NAME of workers whose name ends with 'A'?

Select * from worker where FIRST_NAME Like '%a';

Q. What is an SQL query for printing the details of the workers whose FIRST_NAME ends with 'H' and contains six alphabets?

select * from Worker where FIRST_NAME Like '____h';

Q. Write an SQL Query for printing the details of workers whose SALARY lies between 10000 and 20000.

```
Select * from Worker where SALARY between 10000 & 20000;
```

Q. Write an SQL Query for printing the details of workers who joined in Feb '2014.

```
Select * from Worker where year(JOINING_DATE) = 2014 and
month (JOINING_DATE) = 2;
```

Q. Write an SQL Query for fetching the ~~count~~ details of workers with salaries >= 5000 and <= 10000.

```
SELECT CONCAT (FIRST_NAME, ' ', LAST NAME) AS
Worker_Name , Salary
FROM Worker
WHERE WORKER_ID IN
(SELECT WORRER_ID FROM Worker
 WHERE Salary BETWEEN 5000 AND 10000);
```

Q. Write an SQL Query for fetching the count of workers in DEPARTMENT with 'Admin'.

```
SELECT COUNT(*) FROM worker WHERE DEPARTMENT
                              ='Admin';
```

Q. What is an SQL Query for fetching the no. of workers in each department in descending order?

```
SELECT DEPARTMENT, count(WORKER_ID) No_of_Workers
FROM worker
GROUP BY DEPARTMENT.
ORDER BY No_of_Workers DESC;
```

Q. What is an SQL Query for printing the details of workers who are also managers?

```
SELECT DISTINCT W.FIRST_NAME, T.WORKER_TITLE
FROM Worker W
INNER JOIN Title T
ON W.WORKER_ID = T.WORKER_REF_ID
AND T.WORKER_TITLE in ('Manager');
```

Q. Write an SQL Query for fetching the details of duplicate records in some fields.

```
SELECT WORKER_TITLE, AFFECTED_FROM, COUNT(*)
FROM Title
GROUP BY WORKER_TITLE, AFFECTED_FROM
HAVING COUNT(*)>1;
```

Q. What is an SQL Query for only showing odd rows?

```
SELECT * FROM Worker WHERE MOD(WORKER_ID, 2) <> 0;
```

Q. What is an SQL Query for only showing even rows?

```
SELECT * FROM Worker WHERE MOD(WORKER_ID, 2) = 0;
```

Q. Write an SQL Query for cloning a new table from another table.

```
SELECT * INTO WorkerClone FROM Worker;
```

OR

```
SELECT * INTO WorkerClone FROM Worker WHERE 1 = 0;
```

Q. Write an SQL Query for fetching the intersecting details of two tables.

```
(SELECT * FROM Worker)
INTERSECT
(SELECT * FROM Workerclone);
```

Q. What is an SQL Query for showing the details of one table that another doesn't have.

```
SELECT * FROM Worker
MINUS
SELECT * FROM Title;
```

Q. What is the difference bet" DDL & DML

DDL - Data Defination Language

- Usage - DDL statements are used to create database, schema, constraints, users, tables etc
- Classification - No. classification
- Commands - CREATE, DROP, RENAME & ALTER, TRUNCATE

DML - Data Manipulation Language

- Usage: - DML statement is used to insert, update or delete the records.
- Classification - • procedural DML
  • Non-procedural DML
- Commands - INSERT, UPDATE and DELETE

Q. Having() and WHERE() clause?

HAVING():

The HAVING clause was added to SQL because the WHERE() keyword cannot be used with aggregate functions.

HAVING() syntax:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

WHERE():

- The WHERE clause is used to filter records
- It is used to extract only those records that fulfill a specific condition

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

Note- WHERE() clause is not only used in SELECT statements it is also used in UPDATE, DELETE etc.

Q Explain JOIN and it's types?

- JOIN is a clause is used to combine rows from two or more tables, based on related column between them.

① INNER JOIN

The INNER JOIN keyword selects records that have matching values in both tables.

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

② LEFT JOIN: Returns all records from the left table (table1), and the matching records (if any) from the Right table (table2).

③ RIGHT JOIN: Returns all records from the Right table (table2), and the matching records (if any) from the left table (table1).

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

④ CROSS JOIN: Returns all records from both tables (table1 and table2)

```
SELECT column_name(s)
FROM table1
CROSS JOIN table2;
```

⑤ SELF JOIN: Regular join, but the table is joined with itself.

```
SELECT column_name(s)
FROM table1 T1, table1 T2
WHERE condition;
```

T1 & T2 - different table aliases for the same table.

Q. GROUP BY:

- Group By statement groups rows that have the same values into summary rows, like "find the number of clusters in each country"

- Group By statements is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AUG()) to group the result-set by one or more columns.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Q What is the diff^n bet^n Grant & Revoke

- Grant & Revoke belong to these type of commands of the Data controlling language. DCL is a component of SQL commands.

1) Grant :

SQL Grant command is specifically use to provide ~~pre~~ privileges to database objects for an user. This command users to grant permissions for other users too.

syntax :

grant privilege_name on object_name
to { user_name | public | role_name }

privilege_name → is which permission has granted
object_name → name of the database obj.
user_name → user to which access should be provided
public → permit access to all users

2) Revoke :

Revoke command withdraws user privileges on database objects if any granted. It does operations opposite to the Grant command. When a privilege is revoked from a particular user U, then the privileges granted to all other users by user U will be revoked.

syntax :

revoke privilege_name on object_name
from { user_name | public | role_name }

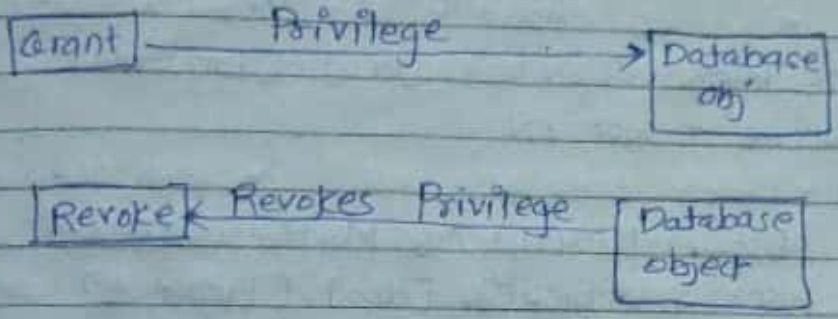
Examples

grant insert,
select on accounts to Ram

By above command user ram has grauted permissions on accounts database objects like he can query or insert into accounts.

revoke insert,
select on accounts from Ram

By above command user ram's permission like query or insert on accounts database object has been removed.



Grant & Revoke command.

- Grant
- Revoke

1) This DCL command grauts permission to the user on the database objects.	1) This DCL command removes permission if any granted to the users on database objects
2) It Assign access rights to users	2) It revokes access rights of users.
3) For each user you need to specify the permissions	3) If access for one user is removed; All the particular permissions provided by that user to others will be removed.

4) When the access is decentralized granting permissions will be easy

4) If decentralized access removing the granted permission default.

Q. Different ways to SQL delete duplicates rows from a SQL Table

① delete duplicates by using Group By & having clause

```
DELETE FROM [SampleDB].[dbo].[Employee]
    WHERE ID NOT IN
    (
      SELECT MAX(ID) As MaxRecordID
      FROM [sampleDB].[dbo].[Employee]
      GROUP BY [FirstName],
               [LastName],
               [Country]
    );
```

All tables:

ID	FirstName	LastName
1	Raj	Gupta
2	Raj	Gupta
3	Raghav	Kumar
4	Mahaes	Barry
5	James	Barry
6	James	Barry

	MaxRecordID
1	6
2	3
3	2

```
Select *
FROM [Sample DB].[dbo].[Employee]
```

	ID	FirstName	LastName	Country
1	1	Raj	Gupta	India
2	4	James	Barry	UK
3	5	James	Barry	UK

After Removing Duplicates

	ID	FirstName	LastName	Country
1	2	Raj	Gupta	India
2	5	Mohan	kumar	USA
3	6	James	Barry	UK

Q. What is the diffn betn delete & Truncate

- Delete is a DML command
- Truncate is a DDL command
- Truncate can be used for delete entire data of the table without maintaining integrity
- Delete can be used for delete specific Data.

Key	Delete	Truncate
1) Basic	• It is used to delete specific data	• It is used to delete the entire data of the table.
2) Where clause	• We can use with where clause	• It can't be used with where clause
3) Locking	• It locks the table row before deleting the row.	• It locks the entire table
4) Rollback	• We can Rollback the changes	• We can't rollback the changes
5) Performance	• It is slower than truncate	• It is faster than delete.
	• Query	• Query
	DELETE FROM tableName WHERE condition;	TRUNCATE TABLE tableName;

Q. Diff^n bet^n Delete & DROP

Key	DELETE	DROP
① Purpose	DELETE command removes some or all tuples/ records from relation/table	DROP command, removes all named element of schema like relation/table constrains or entire schema.
② Language	DML	DDL
③ clause	WHERE clause is used to add filtering	NO WHERE clause available
④ Rollback	Delete command can be rollbacked as it works on data buffer	Drop command can't be rollbacked as it works directly on data.
⑤ memory space	Table memory space is not a free if all records are deleted using delete command.	Drop command may cause memory fragmentation.
⑥ Interaction	SQL directly interact with database server	PL/SQL does not directly interact with dB server
⑦ Problem	DELETE command may face shortage of memory	DROP command may cause memory fragmentation
⑧ Orientation	SQL is Data oriented Language	PL/SQL is application oriented Language
⑨ Objective	SQL is used to write queries create & execute DDL & DML statement	PL/SQL is used to write program blocks, fns, procedure triggers & Packages

Scanned with CamScanner

Q Diff^n bet^n JOIN and UNION

- Union is a set operator that can be used to combine the result set of two different SELECT statement. In the union member of columns and data types should be the same

- Types:
  - Union
  - Union All

- Relational Database, tables are associated with each other & we use Foreign key to maintain relationships between tables. We use JOIN clause to retrive data from associated tables. The join condition indicates how column each table are matched against each other

There are two types of joins clause in SQL
- Inner JOIN
- Outer JOIN

Key	JOIN	UNION
1) Basic	It can be used to retrive matched records bet^n both tables or more tables	It can be used to combine the result set of two different SELECT statement
2) Data type	Result set can have diff^n types of data types	Data types should be same as the result set of each select statement
3) Duplicate	It doesn't remove duplicate data.	It removes duplicate rows between the various select statement.

Example UNION

```
SELECT columnlist
FROM tableA
UNION
SELECT columnlist
FROM tableB
```

Example JOIN

```
SELECT columnlist
FROM TableA
INNER JOIN
TableB ON join condition
```

Q. diff^n bet^n Inner Join & Outer Join

- There are 2 types of JOIN
  - INNER
  - OUTER
    - LEFT
    - RIGHT

Example of INNER JOIN

```
SELECT columnlist
FROM TableA
INNER JOIN
TableB ON join condition
```

Example of OUTER JOIN

```
SELECT columnlist
FROM tableA
FULL OUTER JOIN tableB
ON tableA.column_name = tableB.
                        column_name
WHERE condition;
```

• key	• Inner Join	• Outer Join
• ① Basic	• It can be used to retrive only matched records bet^n both the tables.	• It is used to retrive all matching records of the tables as well as matching records of the tables.
• ② Returns	• It doesn't return anything when match is not found	• It returns null in the column value.
• ③ Performance	• It is faster than outer join	• It is slower than inner join because of the larger result set.

Q. How to find **maximum salary** from given table.

Sr. No.	Name	Position	Office	Salary

Employee_db

- Query to find the **highest** salary

```
SELECT * FROM [DBO].[EMPLOYEE]
ORDER BY SALARY DESC
SELECT MAX(SALARY) FROM EMPLOYEE
```

- Query to find **second highest** salary

```
SELECT * FROM EMPLOYEE ORDER BY SALARY DESC
SELECT MAX (SALARY) FROM EMPLOYEE.
WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE)
```

```
SELECT *FROM EMPLOYEE ORDER BY SALARY DESC
SELECT DISTINCT TOP 2 SALARY
FROM EMPLOYEE
ORDER BY SALARY DESC
```

- Nth Number of highest salary using DENSE_RANK function

```
SELECT SALARY,
        DENSE_RANK() OVER (ORDER BY SALARY DESC) AS
                                    SALARY RANK,
FROM EMPLOYEE
```

- To find Nth highest salary using CTE

```
SELECT *FROM [DBO].[EMPLOYEE] ORDER BY SALARY DESC
GO
WITH RESULT AS
(
    SELECT SALARY,
        DENSE_RANK() OVER (ORDER BY SALARY DESC)
                                    AS DENSERANK
    FROM EMPLOYEE
)
SELECT TOP 1 SALARY
FROM RESULT
WHERE DENSERANK = 3
```

Q. What is mean by Methodology in SQL

- Methods are procedures and functions that define the operations permitted on data defined using data cartridge.

- A method is procedure or function that is a part of the object type of defination, and that can operate on the attributes of the type. Such methods are also called member methods & they take the keyword MEMBER when you specify them as a component of the object type

  - method specification.
  - method names.
  - method names overloading.

Datacenters : MH : PUNE
WESTERN MUMBAI
: CHENNAI
} for India for different Region Datacenters

→ DDL

Data Definition Language

Alter, Drop, Create, Truncate, Rename.

→ DML

Data Manipulation Language

Insert, Delete, Update, Select

→ DCL

Data Control Language

Grant, Revoke

→ TCL

Transiction Control Language

Commit, Rollback, Savepoint.

- Deployment tools:

  - public
  - private
  - Hybrid
  - community cloud.

- Azure Database (SQL)
- Deployment models

⇒ SQL Database (PaaS)

→ Managed Instance
→ Single Database
→ Elastic Pool

Paas
↓
Platform as a service (cloud computing model)

Saas
↓
Software as a service (cloud provider hosts application & makes them available)

- Service Tiers
  - General purpose / standard
  - Bussiness critical / Premium
  - Hyperscale

- Database divided into 3 types
  - → RDBMS
  - → NOSQL
  - → OLAP

OLTP	OLAP
↓	↓
• Industry specific	• Subject specific
• Transaction	• Aggregate data from transaction

- Advantages of NOSQL
  - It support query language
  - It handles large volume of data
  - It provides fast performance
  - It provides horizontal scalability

SQL	NOSQL
• Structured data	• Un-structured data
• can't use	• Big data
• Connect tables	• can't connect
• vertically scalability	• Horizontal Scalability

- Cosmos DB
  - It is NoSQL database
  - Schema work
    - Schema → Create, Add, Insert, Update
      (empid empname empdept empsal)

- Coosistency Level of Databases:
  (Based on performance & availability)

  → Eventual: (Primary & Secondary node used for writing & Reading)
  → Strong: (latest copy of data & gives all users But gives less
  veletively performance)
  → Coosistent Prefix: (Client can read data in the same order
  as it is written)
  → Session: (users who committed the data will be able to see)
  → Bounded staleness: (which data woout be replicated into
  the secondary node)

- Application Cosmos DB

  - IOT
  - Website data
  - Retail & marketing
  - Application Data