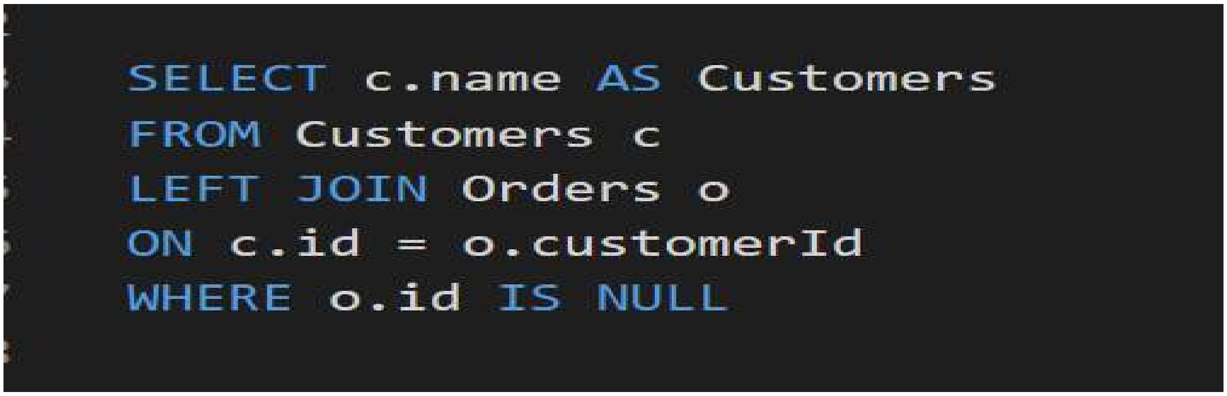
MOST ASKED SQL QUERIES IN THE INTERVIEW

Q1. Write an SQL query to report all customers who never order anything.

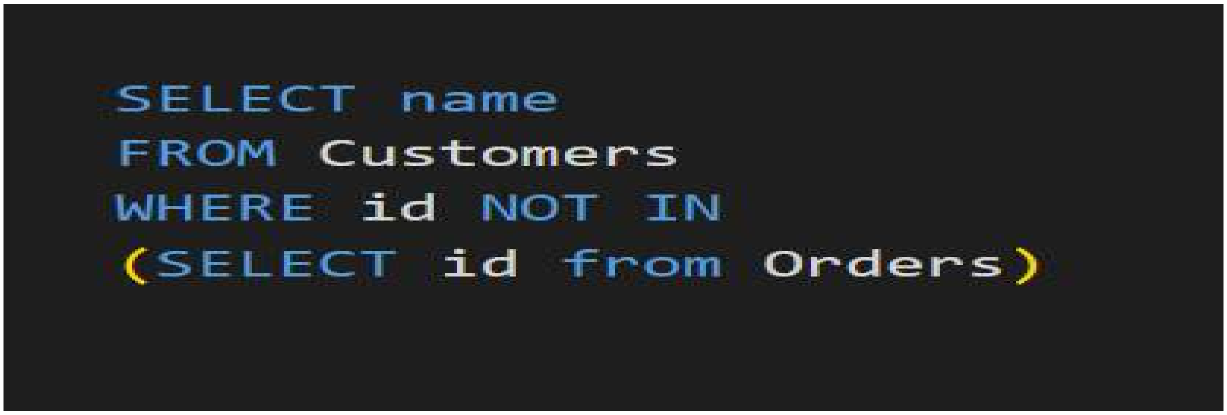
USING SELF JOIN



```
SELECT c.name AS Customers
FROM Customers c
LEFT JOIN Orders o
ON c.id = o.customerId
WHERE o.id IS NULL
```

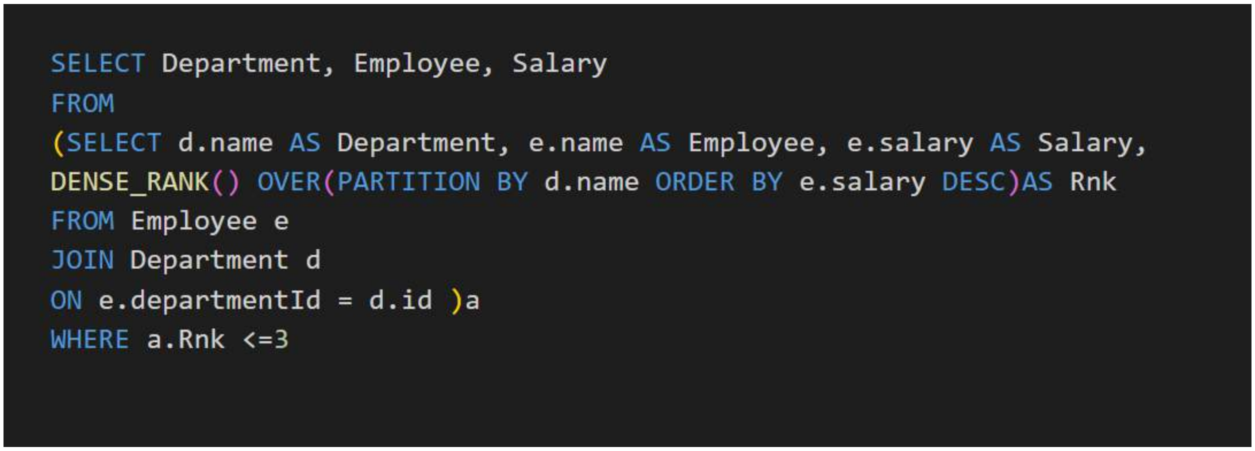
OR

USING SUBQUERY



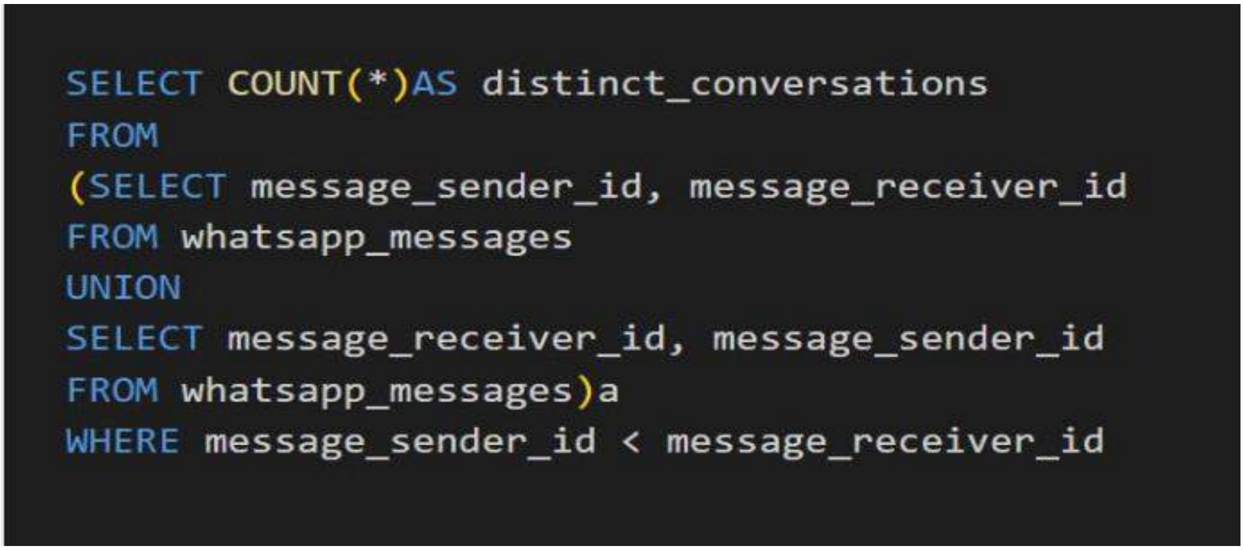
```
SELECT name
FROM Customers
WHERE id NOT IN
(SELECT id from Orders)
```

Q2. Write an SQL query to find the employees who are high earners in each of the departments.



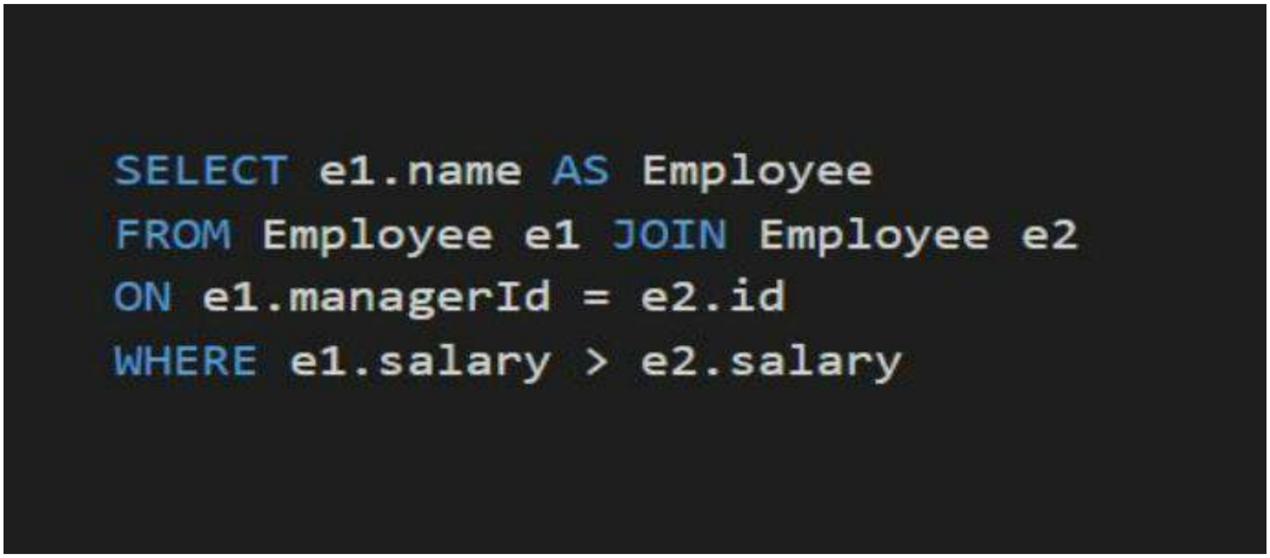
```sql
SELECT Department, Employee, Salary
FROM
(SELECT d.name AS Department, e.name AS Employee, e.salary AS Salary,
DENSE_RANK() OVER(PARTITION BY d.name ORDER BY e.salary DESC)AS Rnk
FROM Employee e
JOIN Department d
ON e.departmentId = d.id )a
WHERE a.Rnk <=3
```

Q3. Count the total number of distinct conversations on WhatsApp.
--> Two users share a conversation if at least 1 message exists between them.
--> Multiple messages between the same pair of users are considered a single conversation.



```sql
SELECT COUNT(*)AS distinct_conversations
FROM
(SELECT message_sender_id, message_receiver_id
FROM whatsapp_messages
UNION
SELECT message_receiver_id, message_sender_id
FROM whatsapp_messages)a
WHERE message_sender_id < message_receiver_id
```

Q4. Write an SQL query to find the employees who earn more than their managers.



```sql
SELECT e1.name AS Employee
FROM Employee e1 JOIN Employee e2
ON e1.managerId = e2.id
WHERE e1.salary > e2.salary
```

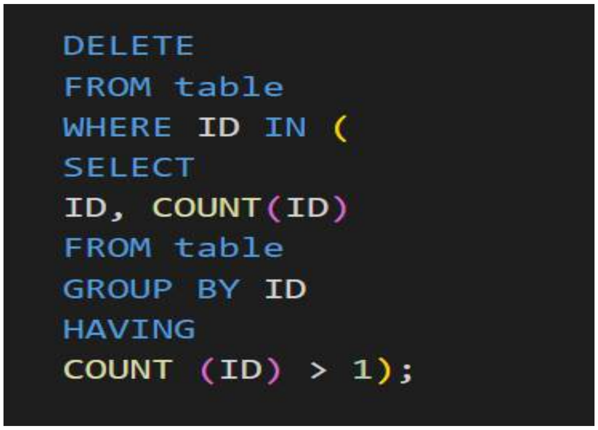
Q5. How to remove duplicate rows in SQL?

If the SQL table has duplicate rows, the duplicate rows must be removed.

Let's assume the following table as our dataset:

ID	Name	Age
1	A	21
2	B	23
2	B	23
4	D	22
5	E	25
6	G	26
5	E	25

The following SQL query removes the duplicate ids from the table:



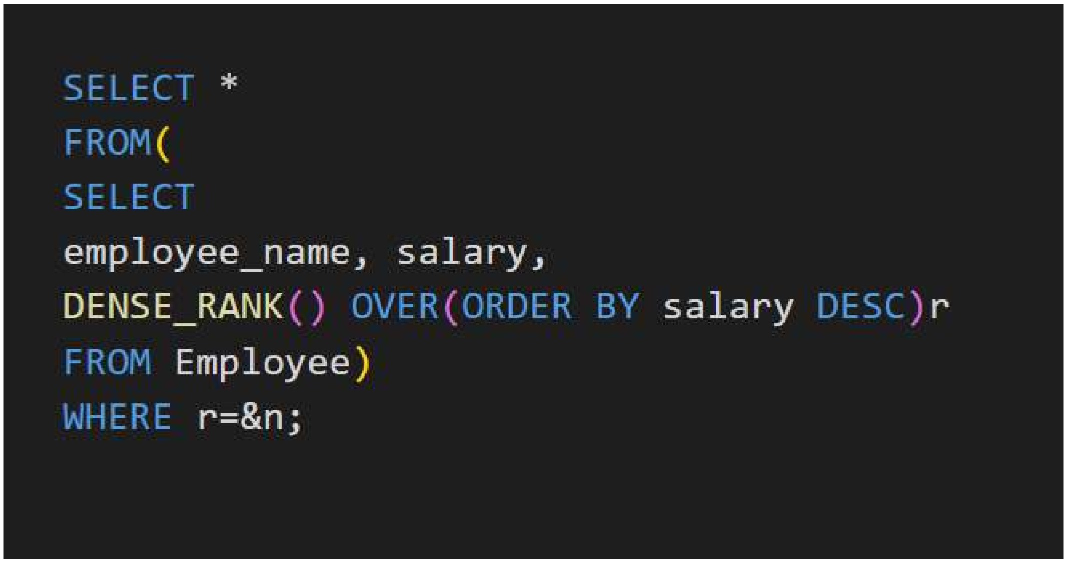
```sql
DELETE
FROM table
WHERE ID IN (
SELECT
ID, COUNT(ID)
FROM table
GROUP BY ID
HAVING
COUNT (ID) > 1);
```

Q6. How to find the nth highest salary in SQL?

The most typical interview question is to find the Nth highest pay in a table.
This work can be accomplished using the **dense rank**() function.

Employee table

employee_name	salary
A	24000
C	34000
D	55000
E	75000
F	21000
G	40000
H	50000



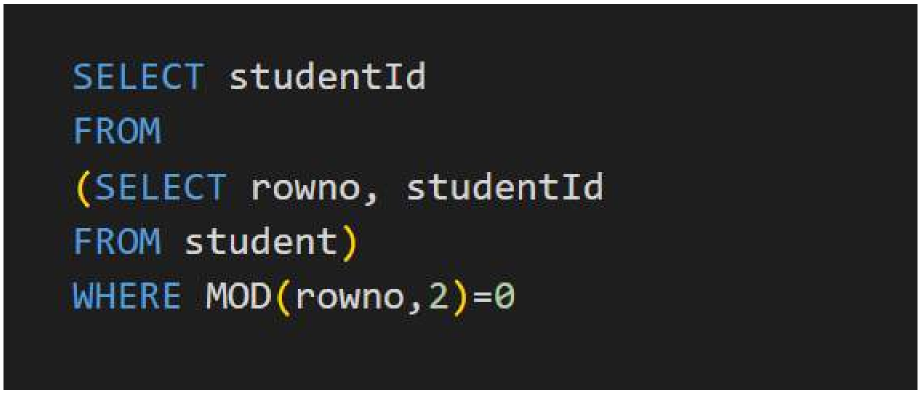
```
SELECT *
FROM(
SELECT
employee_name, salary,
DENSE_RANK() OVER(ORDER BY salary DESC)r
FROM Employee)
WHERE r=&n;
```

To find to the 2nd highest salary set n = 2
To find 3rd highest salary set n = 3 and so on.

Q7. How can you fetch alternate records from a table?

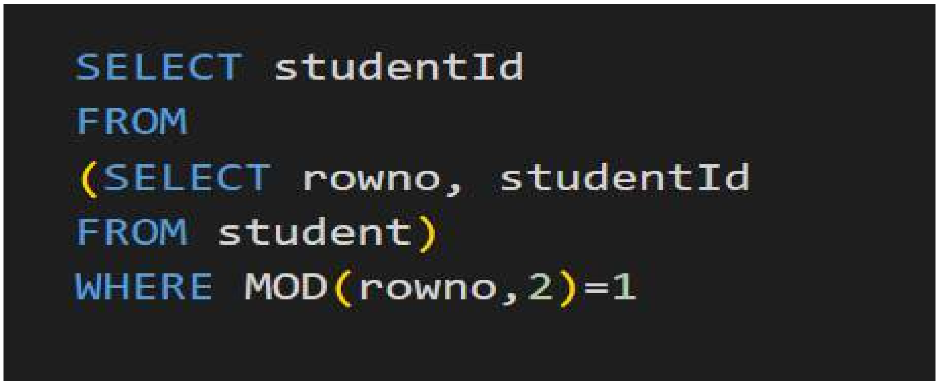
You can fetch alternate records i.e both odd and even row numbers.
For example-

To display even numbers, **use the following command**:



```
SELECT studentId
FROM
(SELECT rowno, studentId
FROM student)
WHERE MOD(rowno,2)=0
```

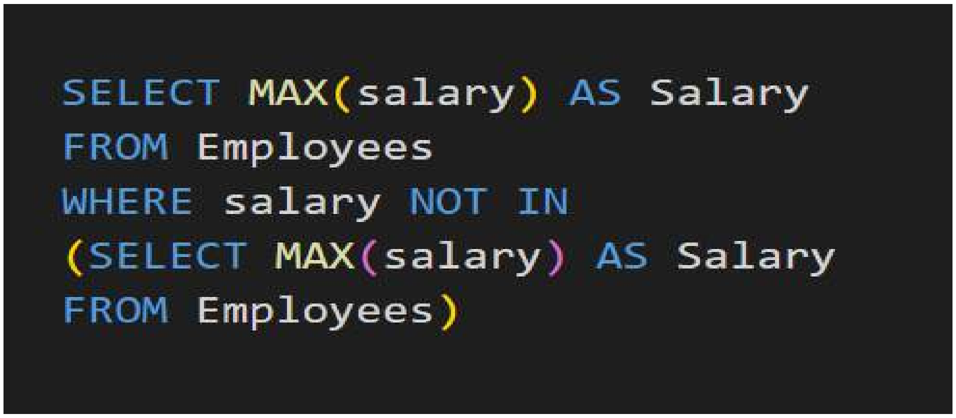
Now, **to display odd numbers**:



```
SELECT studentId
FROM
(SELECT rowno, studentId
FROM student)
WHERE MOD(rowno,2)=1
```

Q8. How to find the employee with second MAX Salary using a SQL query?

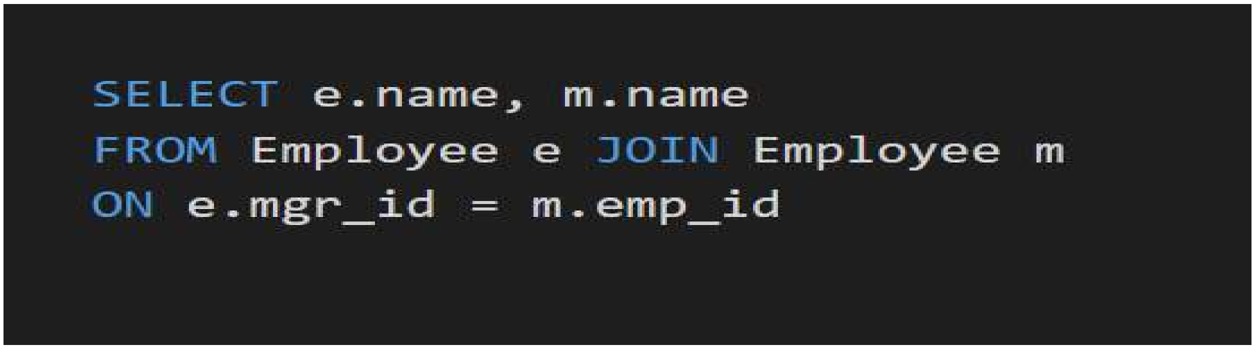
In order to find the second MAX salary, employee record with MAX salary needs to be eliminated. It can be achieved by using below SQL query.



```sql
SELECT MAX(salary) AS Salary
FROM Employees
WHERE salary NOT IN
(SELECT MAX(salary) AS Salary
FROM Employees)
```

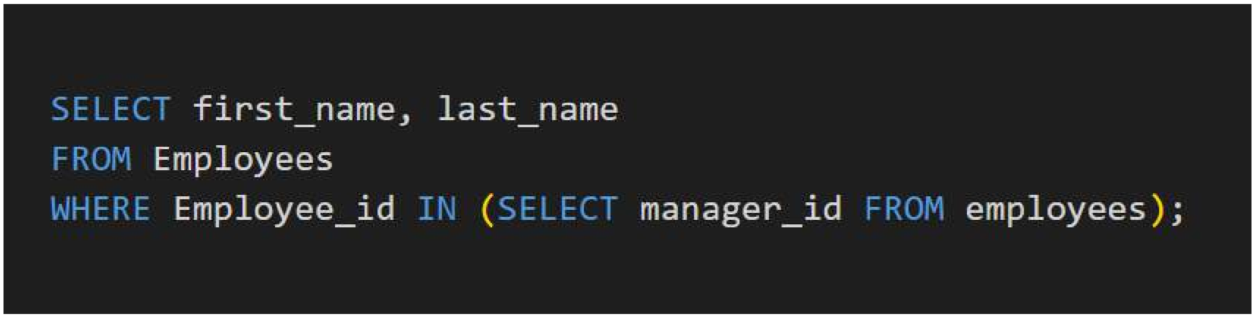
Q9. How do you find all employees who are also managers?

USING SELF JOIN



```sql
SELECT e.name, m.name
FROM Employee e JOIN Employee m
ON e.mgr_id = m.emp_id
```

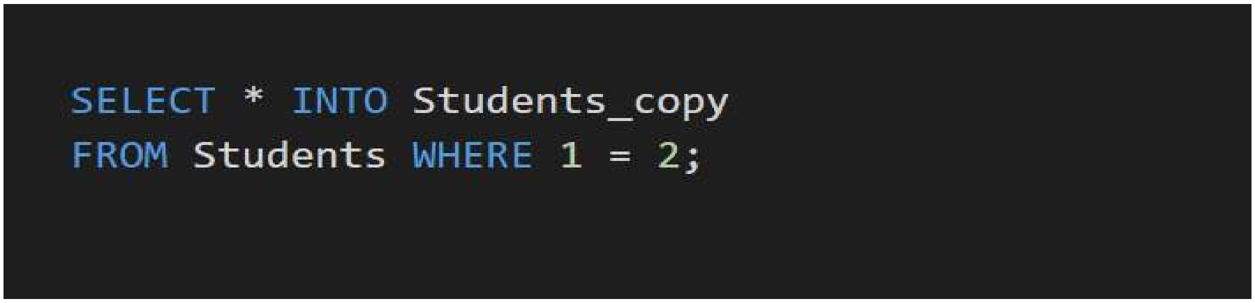
OR
USING SUB-QUERY



```sql
SELECT first_name, last_name
FROM Employees
WHERE Employee_id IN (SELECT manager_id FROM employees);
```

Q10. How to create empty tables with the same structure as another table?

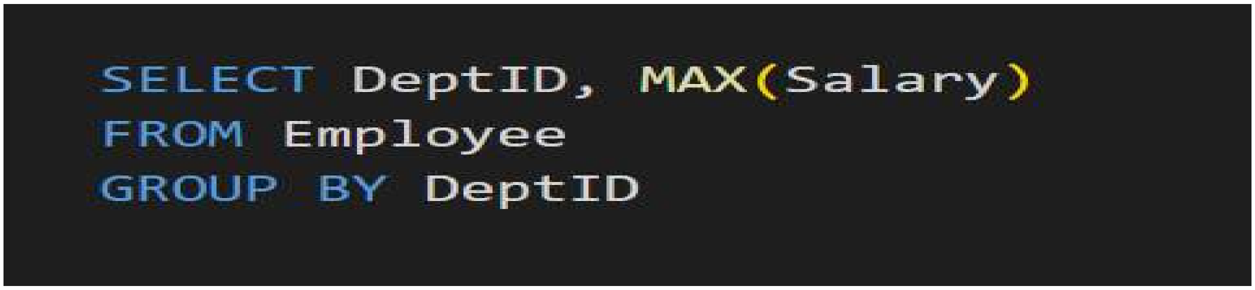
Creating empty tables with the same structure can be done smartly by fetching the records of one table into a new table using the INTO operator while fixing a WHERE clause to be false for all records. Hence, SQL prepares the new table with a duplicate structure to accept the fetched records but since no records get fetched due to the WHERE clause in action, nothing is inserted into the new table.



```sql
SELECT * INTO Students_copy
FROM Students WHERE 1 = 2;
```

Q11. Write an SQL Query to find Max Salary from each department.

We can find the maximum salary for each department by grouping all records by DeptId and then using MAX() function to calculate the maximum salary in each group or each department.



```sql
SELECT DeptID, MAX(Salary)
FROM Employee
GROUP BY DeptID
```

MOST ASKED SQL QUESTIONS IN THE INTERVIEW

Q1. What is the difference between SQL and MySQL?

SQL

SQL is a standard language which stands for Structured Query Language based on the English language

SQL is the core of the relational database which is used for accessing and managing database

MySQL

MySQL is a database management system.

MySQL is an RDMS (Relational Database Management System) such as SQL Server, Informix, etc.

Q2. What are the different subsets of SQL?

There are four main subsets of the SQL language:

Data Control Language (**DCL**)
Data Definition Language (**DDL**)
Data Manipulation Language (**DML**)
Transaction control language (**TCL**)

Each set of the SQL language has a special purpose:

DCL is used to control access to data in a database such as to grant or revoke specified users' rights to perform specified tasks.

DDL is used to define data structures such as to create , alter, or drop tables.

DML is used to retrieve and manipulate data in the table such as to insert, delete, and update data.

TCL is a set of special commands that deal with the transactions within the database. Basically, they are used to manage transactions within the database.

Q3. **What do you mean by DBMS**? **What are its different types**?

There are three main types of DBMS data models:
Relational, **network**, **and hierarchical**.

Relational data model: Data is organized as logically independent tables.
Network data model: All entities are organized in graphical representations.
Hierarchical data model: Data is organized into a tree-like structure.

--

Q4. **List the different types of relationships in SQL**.

There are different types of relations in the database:

One-to-One – This is a connection between two tables in which each record in one table corresponds to the maximum of one record in the other.
One-to-Many and Many-to-One – This is the most frequent connection, in which a record in one table is linked to several records in another.
Many-to-Many – This is used when defining a relationship that requires several instances on each side.

--

Q5. **What is a Primary key**?

A Primary key in SQL is a column (or collection of columns) or a set of columns that **uniquely identifies each row in the table**.
Uniquely identifies a single row in the table
Null values are not allowed

--

Q6. **What are Constraints**?

Constraints in SQL are used to specify the limit on the data type of the table. It can be specified while creating or altering the table statement.
The sample of constraints are:

NOT NULL
CHECK
DEFAULT
UNIQUE
PRIMARY KEY
FOREIGN KEY

Q7. **What is a Unique key**?

--> Uniquely identifies a single row in the table.
--> Multiple values allowed per table.
--> Null values allowed.

Q8. **What is a Foreign key in SQL**?

--> Foreign key maintains referential integrity by enforcing a link between the data in two tables.
--> The foreign key in the child table references the primary key in the parent table.
--> The foreign key constraint prevents actions that would destroy links between the child and parent tables.

Q9. **Difference Between DELETE**, **TRUNCATE and DROP Commands**

DELETE

--> DELETE is a Data Manipulation Language (**DML**) command.

--> The DELETE command lets us either delete all the rows or delete them one by one, and the 'WHERE' clause allows us to implement it according to the requirement.

--> The DELETE command **does not free the tablespace** from memory.

--> The DELETE command deletes each record individually, making it **slower than a TRUNCATE command**.

--> **Data can be rolled back** with the DELETE command.

TRUNCATE

--> TRUNCATE is a Data Definition Language (**DDL**) command

--> The TRUNCATE command cannot be used to delete a single row as there is no 'WHERE' clause in this case.

--> The TRUNCATE command **does not free the tablespace** from memory.

--> The TRUNCATE command is **faster than both DROP and DELETE commands**.

--> **Data cannot be rolled back** with the TRUNCATE command.

DROP

--> DROP is a Data Definition Language (**DDL**) command.

--> The DROP command destroys the existence of the whole table.

--> DROP **release memory** from the tablespace.

--> **DROP is quick to execute but slower than TRUNCATE** because of its complexities.

--> **Data cannot be rolled back** with the DROP command since it permanently deletes the table.

Q10. **What is the main difference between 'BETWEEN' and 'IN' condition operators**?

BETWEEN operator is used to display rows based on a range of values in a row whereas the **IN** condition operator is used to check for values contained in a specific set of values.
Example of BETWEEN:
SELECT * FROM Students where ROLL_NO BETWEEN 10 AND 50;
Example of IN:
SELECT * FROM students where ROLL_NO IN (8,15,25);

Q11. **What is the difference between 'HAVING' CLAUSE and a 'WHERE' CLAUSE**?

The **HAVING** clause is used instead of WHERE with **aggregate functions**.
While the GROUP BY Clause groups rows that have the same values into summary rows.
The having clause is used with the where clause in order to find rows with certain conditions.
The **having** clause is always used **after** the **group By clause**.

Q12. **Are NULL values same as that of zero or a blank space**?

A **NULL** value is not at all same as that of zero or a blank space. NULL value represents a **value which is unavailable**, **unknown**, **assigned or not applicable** whereas a **zero is a number** and **blank space is a character**.

Q13. What are different types of joins in SQL?

Inner join: Inner Join in SQL is the most common type of join. It is used to return all the rows from multiple tables where the join condition is satisfied.

Left Join: Left Join in SQL is used to return all the rows from the left table but only the matching rows from the right table where the join condition is fulfilled.

Right Join: Right Join in SQL is used to return all the rows from the right table but only the matching rows from the left table where the join condition is fulfilled.

Full Join: Full join returns all the records when there is a match in any of the tables. Therefore, it returns all the rows from the left-hand side table and all the rows from the right-hand side table.

Self Join: A self-join is a join that is used to join a table with itself.
That is, each row of the table is joined with itself and all other rows depending on some conditions.

Q14. What are the rules to follow when using UNION operator?

You can combine two queries using UNION operator if they follow the below rules:
📌 Both queries must return the same no of columns.
📌 The columns in both queries must be in the same order.
📌 Data type of all the columns in both queries must be the same.

Q15. What is the difference between RANK, **DENSE_RANK and ROW_NUMBER window function**?

RANK() will assign a rank to each row within each partitioned result set.
--> If multiple rows have the same value then each of these rows will share the same rank.
--> However the rank of the following (next) rows will get skipped. Meaning for each duplicate row, one rank value gets skipped.

DENSE_RANK() will assign a rank to each row within each partitioned result set.
--> If multiple rows have the same value then each of these rows will share the same rank.
--> However the dense_rank of the following (next) rows will NOT get skipped.

ROW_NUMBER() will assign a unique row number to every row within each partitioned result set. It does not matter if the rows are duplicates or not.

Q16. What is the case when in SQL ?

The CASE statement is used to construct logic in which one column's value is determined by the values of other columns.

At least one set of WHEN and THEN commands makes up the SQL Server CASE Statement.

The condition to be tested is specified by the WHEN statement.
If the WHEN condition returns TRUE, the THEN sentence explains what to do.

When none of the WHEN conditions return true, the ELSE statement is executed. The END keyword brings the CASE statement to a close.

```
CASE
WHEN condition1 THEN result1
WHEN condition2 THEN result2
WHEN conditionN THEN resultN
ELSE result
END;
```

Q17, **What is subquery in SQL**?

A subquery is a query inside another query where a query is defined to retrieve data or information back from the database.

In a subquery, the outer query is called as the main query whereas the inner query is called subquery.

Subqueries are always executed first and the result of the subquery is passed on to the main query.
It can be nested inside a SELECT, UPDATE or any other query.
A subquery can also use any comparison operators such as >,< or =.

Q18. What are the different types of a subquery?

There are several types of subqueries in MySQL, including:

📌 **Correlated subquery**: A subquery that references a column from a table in the outer query.

📌 **Non-correlated subquery**: A subquery that does not reference any column from the outer query.

📌 **Single-row subquery**: A subquery that returns only one row.

📌 **Multiple-row subquery**: A subquery that returns multiple rows.

📌 **Scalar subquery**: A subquery that returns a single value.

📌 **Derived table**: A subquery that is used as a table in the outer query.

📌 **Inline view**: A subquery that is used as a table in the outer query, but is not stored as a separate table in the database.

📌 **Common table expression**: A subquery that is defined using the WITH clause and can be used multiple times in a single query.

Q19. Explain different types of index in SQL.

There are three types of index in SQL namely:

Unique Index:
This index does not allow the field to have duplicate values if the column is unique indexed. If a primary key is defined, a unique index can be applied automatically.

Clustered Index:
This index reorders the physical order of the table and searches based on the basis of key values. Each table can only have one clustered index.

Non-Clustered Index:
Non-Clustered Index does not alter the physical order of the table and maintains a logical order of the data. Each table can have many nonclustered indexes.

Q20. Write an SQL query to find names of employees start with 'A'?

The **LIKE** operator of SQL is used for this purpose.
It is used to fetch filtered data by searching for a particular pattern in the where clause.
The required query is:

```
SELECT *
FROM Employees
WHERE EmpName like 'A%' ;
```

Q21. What is the difference between CHAR and VARCHAR2 **datatype in SQL**?

Both Char and Varchar2 are used for characters datatype but **varchar**2 is used for character strings of **variable length**
whereas Char is used for strings of **fixed length**.
For example, char(10) can only store 10 characters and will not be able to store a string of any other length
whereas varchar2(10) can store any length i.e 6,8,2 in this variable.

Q22. List some case manipulation functions in SQL?

There are three case manipulation functions in SQL, namely:

LOWER: This function returns the string in lowercase. It takes a string as an argument and returns it by converting it into lower case. **Syntax**:
LOWER(‘string’)

UPPER: This function returns the string in uppercase. It takes a string as an argument and returns it by converting it into uppercase. **Syntax**:
UPPER(‘string’)

INITCAP: This function returns the string with the first letter in uppercase and rest of the letters in lowercase.
Syntax:
INITCAP(‘string’)

Q23. What is a Datawarehouse?

Datawarehouse refers to a central repository of data where the data is assembled from multiple sources of information.
Those data are consolidated, transformed and made available for the mining as well as online processing.
Warehouse data also have a subset of data called Data Marts.

Q24. How can you fetch first 5 characters of the string?

There are a lot of ways to fetch characters from a string.
For example:

```
Select SUBSTRING(StudentName,1,5) as studentname
from student
```

Q25. What is normalization?

Normalization is the process of minimizing redundancy and dependency by organizing fields and table of a database. The main aim of Normalization is to add, delete or modify field that can be made in a single table.

Q26. What are all the different normalizations?

The normal forms can be divided into 4 forms, and they are explained below -.

1. **First Normal Form** (1**NF**): This should remove all the duplicate columns from the table. Creation of tables for the related data and identification of unique columns.
2. **Second Normal Form** (2**NF**): Meeting all requirements of the first normal form. Placing the subsets of data in separate tables and Creation of relationships between the tables using primary keys.
3.**Third Normal Form** (3**NF**): This should meet all requirements of 2NF. Removing the columns which are not dependent on primary key constraints.
4. **Fourth Normal Form** (4**NF**): Meeting all the requirements of third normal form and it should not have multivalued dependencies.

Q27. **How many Aggregate Functions are available there in SQL**?

SQL Aggregate Functions calculates values from multiple columns in a table and returns a single value.

There are 7 **aggregate functions we use in SQL**
AVG(): Returns the average value from specified columns
COUNT(): Returns number of table rows
MAX(): Returns largest value among the records
MIN(): Returns smallest value among the records
SUM(): Returns the sum of specified column values
FIRST(): Returns the first value
LAST(): Returns Last value

Q28. What is an SQL View?

A view is a virtual table whose contents are obtained from an existing table or tables, called base tables.
The retrieval happens through an SQL statement, incorporated into the view.
So, you can think of a view object as a view into the base table. The view itself does not contain any real data; the data is electronically stored in the base table.
The view simply shows the data contained in the base table.

--

Q29. What is an Alias in SQL?

An alias is a feature of SQL that is supported by most, if not all, RDBMSs.
It is a temporary name assigned to the table or table column for the purpose of a particular SQL query.
In addition, aliasing can be employed as an obfuscation technique to secure the real names of database fields.
A table alias is also called a correlation name .
An alias is represented explicitly by the AS keyword but in some cases the same can be performed without it as well.
Nevertheless, using the AS keyword is always a good practice.

--

Q30. What do you mean by data integrity?

Data Integrity defines the accuracy as well as the consistency of the data stored in a database. It also defines integrity constraints to enforce business rules on the data when it is entered into an application or a database.

Q31. What is ETL in SQL?

ETL is a process in Data Warehousing and it stands for Extract, Transform, and Load. It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area, and then finally, loads it into the Data Warehouse system. These are three database functions that are incorporated into one tool to pull data out from one database and put data into another database.

Q32. **What is the difference between COALESCE**() & **ISNULL**()?

COALESCE():
COALESCE function in SQL returns the first non-NULL expression among its arguments.
If all the expressions evaluate to null, then the COALESCE function will return null.
Syntax:
SELECT column(s), COALESCE(expression_1,….,expression_n)
FROM table_name;

ISNULL():
The ISNULL function has different uses in SQL Server and MySQL. In SQL Server,
ISNULL() function is used to replace NULL values.
Syntax:
SELECT column(s), ISNULL(column_name, value_to_replace)
FROM table_name;

Q33. **What is a stored procedure**?

Stored Procedures are created to perform one or more DML operations on databases.
It is nothing but a group of SQL statements that accepts some input in the form of parameters and performs some tasks and may or may not return a value.

Q34. What do you understand by query optimization?

The phase that identifies a plan for evaluation query which has the least estimated cost is known as query optimization.
The advantages of query optimization are as follows:
--> The output is provided faster
--> A larger number of queries can be executed in less time
--> Reduces time and space complexity

Q35. What is a trigger?

A DB trigger is a code or program that automatically execute with
response to some event on a table or view in a database. Mainly, trigger helps to maintain the integrity of the database.
Example: When a new student is added to the student database, new
records should be created in the related tables like Exam, Score and
Attendance tables.

Q36. What is the difference between group by and window functions?

Result set size: **GROUP BY** reduces the number of rows in the result set by aggregating rows into groups, while **WINDOW FUNCTIONS** do not reduce the number of rows in the result set and add new columns to it.

Aggregation: **GROUP BY** aggregates rows based on the values in one or more columns, while **WINDOW FUNCTIONS** perform calculations across related rows in an ordered partition of the data.

Use case: **GROUP BY** is typically used for basic aggregation, such as counting, summing, or averaging values, while **WINDOW FUNCTIONS** are used for advanced calculations, such as cumulative sums, running totals, or rankings, that would be difficult or impossible to achieve with GROUP BY alone.

Q37. **What is the difference between union and union all**?

The difference between UNION and UNION ALL is in the way they handle duplicates in the result set:

UNION: The UNION operator **removes duplicates** from the result set and returns only distinct rows.

UNION ALL: The UNION ALL operator **returns all rows**, including duplicates, from the result set.

In other words, UNION combines the result sets and eliminates any duplicate rows, while UNION ALL combines the result sets and retains any duplicate rows.

Q38. **What do you mean by table and field in SQL**?

In SQL, a **table** is a database object that represents a set of related data stored in rows and columns. A **table** is essentially a two-dimensional structure that consists of rows and columns, where each row represents a single record and each column represents a data field within that record.

A **field** or **column** is a vertical entity within a table that holds a specific type of data, such as text, numbers, or dates. Each field has a unique name and a defined data type, such as INT, VARCHAR, or DATE.

Q39. What is the ACID property in a database?

ACID stands for **Atomicity**, **Consistency**, **Isolation**, **Durability**.
It is used to ensure that the data transactions are processed reliably in a database system.
--> **Atomicity**: Atomicity refers to the transactions that are completely done or failed where transaction refers to a single logical operation of a data.
It means if one part of any transaction fails, the entire transaction fails and the database state is left unchanged.

--> **Consistency**: Consistency ensures that the data must meet all the validation rules.
In simple words, you can say that your transaction never leaves the database without completing its state.

--> **Isolation**: The main goal of isolation is concurrency control.

--> **Durability**: Durability means that if a transaction has been committed, it will occur whatever may come in between such as power loss, crash or any sort of error.

Q40. **Name the operator which is used in the query for pattern matching**?

LIKE operator is used for pattern matching, and it can be used as -.
1. % – It matches zero or more characters.
For example- select * from students where studentname **like** '**a**%'
2. _ (Underscore) – it matches exactly one character.
For example- select * from student where studentname **like** '**abc_**'

Q41. **What are STUFF and REPLACE function**?

STUFF Function: This function is used to overwrite existing character or inserts a string into another string.
Syntax:
STUFF(string_expression,start, length, replacement_characters)
where,
string_expression: it is the string that will have characters substituted
start: This refers to the starting position
length: It refers to the number of characters in the string which are substituted.
replacement_string: They are the new characters which are injected in the string.

REPLACE function: This function is used to replace the existing characters of all the occurrences.
Syntax:
REPLACE (string_expression, search_string, replacement_string)
Here every search_string in the string_expression will be replaced with the replacement_string.

Q42. What is Cross-Join?

Cross join defines as Cartesian product where number of rows in the first table multiplied by number of rows in the second table. If suppose,
WHERE clause is used in cross join then the query will work like an INNER JOIN.

Q43. **What is user defined functions**?

User defined functions are the functions written to use that logic whenever required.
It is not necessary to write the same logic several times.
Instead, function can be called or executed whenever needed.

Q44. **What are all types of user defined functions**?

Three types of user defined functions are.
--> Scalar Functions.
--> Inline Table valued functions.
--> Multi statement valued functions.

Scalar returns unit, variant defined the return clause.
The other two types return table as a return.

Q45. What are aggregate and scalar functions?

For doing operations on data SQL has many built-in functions, they are categorized into two categories and further sub-categorized into seven different functions under each category.
The categories are:
Aggregate functions: These functions are used to do operations from the values of the column and a single value is returned.
Scalar functions: These functions are based on user input, these too return a single value.

Q46. **What are Nested Triggers**?

A trigger can also contain **INSERT**, **UPDATE**, **and DELETE** logic within itself, so when the trigger is fired because of data modification it can also cause another data modification, thereby firing another trigger.
A trigger that contains data modification logic within itself is called a nested trigger.

Q47. **Explain the difference between OLTP and OLAP**.

OLTP: It stands for **Online Transaction Processing**, and we can consider it to be a category of software applications that is efficient for supporting transaction-oriented programs.
One of the important attributes of the OLTP system is its potential to keep up the consistency.
The OLTP system often follows decentralized planning to keep away from single points of failure.
This system is generally designed for a large audience of end-users to perform short transactions.
Also, queries involved in such databases are generally simple, need fast response time, and in comparison, return only a few records.
So, the number of transactions per second acts as an effective measure for those systems.

OLAP: OLAP stands for **Online Analytical Processing**, and it is a category of software programs that are identified by a comparatively lower frequency of online transactions. For OLAP systems, the efficiency of computing depends highly on the response time. Hence, such systems are generally used for data mining or maintaining aggregated historical data, and they are usually used in multi-dimensional schemas.

Q48. What are the rules to follow when using UNION operator?

When using the UNION operator in SQL, **there are a few rules to follow**:

--> **Number of columns**: The number of columns in each SELECT statement must be the same. The columns must also have the same data type in each SELECT statement.

--> **Column names**: The column names in the result set are taken from the first SELECT statement.

--> **Order of rows**: The rows in the result set are returned in the order they appear in the SELECT statements.

--> **NULL values**: NULL values are treated as duplicates when using the UNION operator.

--> **Compatibility**: The SELECT statements being combined with the UNION operator must be compatible with each other in terms of data types, expressions, and aggregate functions.

Q49. **What is the difference between UNION and JOIN** ?

UNION: The UNION operator is used to combine the results of two or more SELECT statements into a single result set.
The SELECT statements must have the same number of columns with the same data types.
The UNION operator eliminates duplicates by default, so only unique rows are returned in the result set.

JOIN: The JOIN operator is used to combine rows from two or more tables based on a related column between them.
The result set includes all rows from both tables where the related column values match.
There are several types of JOINs, such as INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN, each of which returns a different subset of the data based on the join conditions.

Q50. **What is a Composite Primary Key** ?

A Composite primary key is a set of columns whose values uniquely identify every row in a table. What it means is that, table which contains composite primary key will be indexed based on columns specified in the primary key. This key will be referred in Foreign Key tables.

Q51. What are local and global variables and their differences?

Local variables are the variables which can be used or exist inside the function.
They are not known to the other functions and those variables cannot be referred or used.
Variables can be created whenever that function is called.

Global variables are the variables which can be used or exist throughout the program.
Same variable declared in global cannot be used in functions.
Global variables cannot be created whenever that function is called.

Q52. **What are all types of user defined functions**?

Three types of user defined functions are.
--> Scalar Functions.
--> Inline Table valued functions.
--> Multi statement valued functions.
Scalar returns unit, variant defined the return clause.
Other two types return table as a return.

Q53. **What is Online Transaction Processing (OLTP)**?

Online Transaction Processing (OLTP) manages transaction based
applications which can be used for data entry, data retrieval and data
processing. OLTP makes data management simple and efficient. Unlike OLAP systems goal of OLTP systems is serving real-time transactions.
Example – Bank Transactions on a daily basis.

Q54. What is SQL Injection?

SQL Injection is a type of database attack technique where malicious SQL statements are inserted into an entry field of database such that once it is executed the database is opened for an attacker. This technique is usually used for attacking Data-Driven Applications to have an access to sensitive data and perform administrative tasks on databases.
For Example: SELECT column_name(s) FROM table_name WHERE condition;

Q55. **What is the use of the Intersect operator**?

The Intersect operator helps combine two select statements and returns only those records that are common to both the select statements. So, after we get Table A and Table B over here and if we apply the Intersect operator on these two tables, then we will get only those records that are common to the result of the select statements of these two.

Q56. **What are UNION, MINUS and INTERSECT commands?**

The **UNION** operator is used to combine the results of two tables while also removing duplicate entries.
The **MINUS** operator is used to return rows from the first query but not from the second query.
The **INTERSECT** operator is used to combine the results of both queries into a single row.
Before running either of the above SQL statements, **certain requirements must be satisfied** –
--> Within the clause, each SELECT query must have the same amount of columns.
--> The data types in the columns must also be comparable.
In each SELECT statement, the columns must be in the same order.

--

Q57. What is Cursor? **How to use a Cursor**?

A database cursor is a control structure that allows for the traversal of records in a database. Cursors, in addition, facilitates processing after traversal, such as retrieval, addition, and deletion of database records.
They can be viewed as a pointer to one row in a set of rows.
Working with SQL Cursor:

1. DECLARE a cursor after any variable declaration.
The cursor declaration must always be associated with a SELECT Statement.

2. Open cursor to initialize the result set.
The OPEN statement must be called before fetching rows from the result set.

3. FETCH statement to retrieve and move to the next row in the result set.

4. Call the CLOSE statement to deactivate the cursor.

5. Finally use the DEALLOCATE statement to delete the cursor definition and release the associated resources.

--

Q58. What is Collation? What are the different types of Collation Sensitivity?

Collation refers to a set of rules that determine how data is sorted and compared.
Rules defining the correct character sequence are used to sort the character data.
It incorporates options for specifying case sensitivity, accent marks, kana character types, and character width.
Below are the different types of collation sensitivity:

Case sensitivity: A and a are treated differently.
Accent sensitivity: a and á are treated differently.
Kana sensitivity: Japanese kana characters Hiragana and Katakana are treated differently.
Width sensitivity: Same character represented in single-byte (half-width) and double-byte (full-width) are treated differently.

Q59. What is the difference between data warehouse and data lake?

Here are the key differences between a data warehouse and a data lake in a nutshell:

🎯 **Data warehouse**: A centralized repository of integrated data, optimized for complex queries and analytics, with a predefined schema and structured data.
🎯 **Data lake**: A centralized repository of raw data, stored in its native format, designed for flexible data exploration and analysis, with a more flexible schema and support for structured, semi-structured, and unstructured data.
In summary, a data warehouse is more structured and optimized for analysis, while a data lake is more flexible and optimized for exploration.

Q60. If you have more than one table, what would you use JOIN or SUBQUERY and why?

Whether to use a join or a subquery depends on the specific requirements of the query and the data being queried.
A **join** is used to combine data from two or more tables based on a common field or set of fields. Joins can be used to extract data from large tables with high efficiency, and the database engine can optimize the query by using indexes or other performance-enhancing techniques.
On the other hand, a **subquery** is typically used to retrieve a specific set of records from one or more tables, based on a set of conditions or criteria. Subqueries are useful when you want to retrieve data from a table based on a condition that cannot be expressed using a simple WHERE clause.

Q61. **What is the difference between self join and cross join**?

Here are three key differences between self join and cross join:

Purpose: A self join is used to combine data from a table with itself, while a cross join is used to combine data from two different tables.

Output: A self join produces a new table that has the same columns as the original table, but with the rows combined based on the join condition. A cross join produces a new table that has columns from both input tables, with each row in one table matched to every row in the other table.

Join conditions: In a self join, the join condition is usually based on a column in the table that is being compared to itself. In a cross join, there is no join condition because every row in one table is matched to every row in the other table.

Q62. What do you mean by recursive stored procedure?

Recursive stored procedure refers to a stored procedure that calls by itself until it reaches some boundary condition.
This recursive function or procedure helps the programmers to use the same set of code n number of times.

Q63. **Write SQL Query to display the current date**?

SQL has a built-in function called GetDate() which returns the current timestamp.
SELECT GetDate();

Q64. **What is the use of the Intersect operator**?

The Intersect operator helps combine two select statements and returns only those records that are common to both select statements. So, after we get Table A and Table B over here and if we apply the Intersect operator on these two tables, then we will get only those records that are common to the result of the select statements of these two.
