SQL Interview Q_A

1. **What is Database?**

   A collection of data and holds this data in the form of tables. The database provides us with the capability to access and manipulate this data.

2. **What are Tables, records, and Fields?**

   Tables holds the data in form of rows and columns it is like excel spreadsheet.
   Columns/ fields can be categorized as **vertical** and **rows/records** as **horizontal**.

3. **What is DBMS?**

   DBMS stands for **Database Management System**. DBMS is a system software responsible for the creation, retrieval, updation, and management of the database.

4. **Types of databases?**

   a.Relational database Management System (RDBMS): rows and columns and also have relation between them.
   Ex: MySQL, SQL Server, Postresql,sqlite,MariaDB,oracle, IBM DB2

   b.No-Sql database: key value ,document, graph
   Ex: Hbase,MangoDB,Cassandra

5. **What is the difference between SQL and MySQL?**

   SQL is a **structured query language** is used to query a **Relational database**.
   MySQL is a **Relational database management system**, like SQL Server, Oracle or IBM DB2, that is used to manage SQL databases.

6. **DDL and DML ?**

   DDL- data definition language – deals with table structure
   - **Create**
   - **Drop**
   - **Alter**
   - **truncate**

   DML-data manipulation language – deals with data directly

- **Insert**
- **Update**
- **Delete**
- **merge**

7. **Data Type?**

   Int – Numeric
   Varchar – Strings up to 255 characters.

8. **CRUD operations?**

   CREATE – creating records using insert statements.
   READ- reading records using select statements.
   UPDATE- updating records using update statements.
   DELETE- deleting records using delete statements.

9. **What are Constraints in SQL?**

   Constraints are used to limit the type of data that can go into a table. This ensure the accuracy of the the data is maintained. If there is any violation then action is aborted.

   - **NOT NULL** - Restricts NULL value from being inserted into a column.
   - **CHECK** - Verifies that all values in a field satisfy a condition.
   - **DEFAULT** - Automatically assigns a default value if no value has been specified for the field.
   - **UNIQUE** - Ensures unique values to be inserted into the field.
   - **INDEX** - Indexes a field providing faster retrieval of records.
   - **PRIMARY KEY** - Uniquely identifies each record in a table.
   - **FOREIGN KEY** - Ensures referential integrity for a record in another table.

10. **What is a Primary Key?**

    - The purpose of primary key is to make sure that uniquely identifies each row/record in a table.
    - A table can have only one primary key.
    - For primary key null values are not allowed and repeated values not allowed (only unique values allowed).
    - **Composite primary key** can have more than one or more column/field.
    - Auto increment key is best for primarykey

11. What is a UNIQUE Key?

- The purpose of unique key is to make sure that values in a column don't duplicate.
- We can have multiple unique keys in a table.
- Unique key can hold null.
- My sql db can hold any number of nulls, some db's can hold only one null.

12. Primary key Vs Unique key?

we can have only one Primary key and multiple unique keys in a table.

13. How do you make case sensitive select query?

By default, sql it is case insensitive. To make case sensitive we need to use **binary** keyword in query.

14. What is a Foreign Key?

- The foreign key constraint is used to prevent actions that would destroy links between two tables.
- A foreign key is a field in one table that refers to the primary key in another table.
- The table with foreign key is called child table.
- The table with primary key is called parent/reference table.

15. What is the difference between Drop, DELETE and TRUNCATE statements?

Drop:

- DROP command is used to remove an object from the database. If you drop a table, all the rows in the table are deleted and the table structure is removed from the database.

  EX : DROP TABLE Candidates;

Delete:

- It's DML command.
- It's use to delete individual records and we can also delete all records.
- It's Drops the records one after another so it takes more time then truncate, to drop all table records .

  EX : DELETE FROM Candidates WHERE CandidateId > 1000;

Truncate :

- it’s DDL command.
- it is also removes all records.
- truncate internally drop the table and recreates it.
- Truncate is efficient then delete.

EX : TRUNCATE TABLE Candidates;

16. What is a Join? List its different types.

is used to combine records (rows) from two or more tables in a SQL database based on a related column between the two.

There are four different types of JOINs in SQL:

(INNER) JOIN: Only matching records from both left and right tables are considered. Non matching records are discarded.

LEFT (OUTER) JOIN:
All matching records from left and right table are considered.
And All non-matching records in the left table which does not have match in the right table will padded with nulls.

RIGHT (OUTER) JOIN:
All matching records from left and right table are considered.
And all non matching records in right table which does not match the right table will padded with the with nulls.

FULL (OUTER) JOIN:
All matching and non-matching records from both left and right join are considered.

17. What is a Self-Join?

A table is joined to itself based on some relation between its own columns.

18. What is a Cross-Join?

Cross join is a Cartesian product where number of rows in the first table multiplied by number of rows in the second table.

If a WHERE clause is used in cross join then the query will work like an INNER JOIN.

19. Natural Join in SQL?

The SQL NATURAL JOIN is a type of EQUI JOIN and is structured in such a way that, columns with the same name of associated tables will appear once only.

20. Difference between Where and Having Clause ?

Having:- is used after group by and filter on aggregated records.HAVING clause is the additional filter to the where clause.

Where:- is used to filter the individual records before aggregation (group by).WHERE clause doesn't work with the aggregations like SUM,AVG,MIN and MAX etc.

We can use where and having in same query. Where is more performant than having.

21. What is the difference between order and group by?

ORDER BY

The ORDER BY clause is used in SQL queries to sort the data returned by a query in ascending or descending order.
ASC denotes ascending order, while DESC denotes descending order

GROUP BY

The GROUP BY clause is used in SQL queries to organize data that have the same attribute values.
We can often use this clause in collaboration with aggregate functions like SUM, AVG, MIN, MAX, and COUNT.

22. What is the SELECT statement?

SELECT operator in SQL is used to select data from a database. The data returned is stored in a result table, called the result-set.
SELECT * FROM myDB.students;

23. What are some common clauses used with SELECT query in SQL?

Some common SQL clauses used in conjuction with a SELECT query are as follows:

Where>order by >group by>having

24. Differentiate between Rank, Row Number and Dense Rank?

These are used to give numbering based on one or more condition.
if there are no duplicates then row number ,rank,dense rank leads to similar results

only difference comes if there are duplicates.

row_number:

- It allows Numbering to duplicate records.
- When we use row number we should use order by clause.
- We can also use partition by clause is optional –then rownumber starts from 1 for each partition.

rank:

- for duplicates same rank is assigned and for the next entry it skips the next rank.(there are holes between ranks).
- Order by clause is mandatory.
- partition by clause is optional.

dense_rank:

- for duplicates same rank assigned and next entry starts from next rank(there should be no holes between ranks).
- Order by clause is mandatory.
- partition by clause is optional.

Scenario: when ever you don't have duplicates use row_number.

25. What is an Index? Explain its different types.

- Index is a data structure that provides a quick lookup of data in a column or columns of a table.
- An Index is created on one or more columns of a table.
- This can significantly speed up the query execution time, especially for large tables.
- An index is a performance tuning method of allowing faster retrieval of records from the table.

There are three types of indexes:

- **Unique Index:** This indexing does not allow the field to have duplicate values if the column is unique indexed.Unique index can be applied automatically when primary key is defined.
- **Clustered Index:** This type of index reorders the physical order of the table and search based on the key values. Each table can have only one clustered index.
- **Non-Clustered Index:** Non-Clustered Index does not alter the physical order of the table and maintains logical order of data.

A table can have a single clustred index where it can have multiple non-clustered indexes.

```
CREATE INDEX index_name

ON table_name (column_1, column_2);

DROP INDEX index_name;
```

26. What is Data Integrity?

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

27. What is a Query?

A query is a request for data or information from a database table or combination of tables. A database query can be either a select query or an action query.

```
/* select query */

SELECT fname, lname
FROM myDb.students
WHERE student_id = 1;

/* action query */

UPDATE myDB.students
SET fname = 'Captain', lname = 'America'
WHERE student_id = 1;
```

28. What is a Subquery? What are its types?

A subquery is a query within another query, also known as a nested query or inner query. The outer query is called as main query, and inner query is called subquery. SubQuery is always executed first, and the result of subquery is passed on to the main query.

There are two types of subquery – Correlated and Non-Correlated.

A correlated subquery cannot be considered as an independent query, whereas a Non-Correlated sub query can be considered as independent query and the output of subquery are substituted in the main query.

29. What are UNION, MINUS and INTERSECT commands?

UNION operator is used to combine the results of two tables, and it eliminates duplicate rows from the tables**.**

MINUS operator is used to return rows from the first query but not from the second query.Matching records of first and second query and other rows from the first query will be displayed as a result set.

INTERSECT operator is used o return rows returned by both the queries.

30. What is the difference between UNION and UNION ALL?

The difference between UNION and UNION ALL is that UNION will omit duplicate records whereas UNION ALL will include duplicate records.

31. What is the difference between JOIN and UNION?

JOIN in SQL is used to combine data from many tables based on a matched condition between them. The data combined using JOIN statement results into new columns.

UNION in SQL is used to combine the result-set of two or more SELECT statements. The data combined using UNION statement results into new distinct rows.

32. Dimension tables and fact tables?

Fact Tables:

- A fact table/ reality table is a primary table in a dimensional model.
- A Fact Table contains Measurements/facts and Foreign key to dimension table
- A fact is a piece of information with a specific numerical value.
- Facts are accompanied by dimensions.
- There is less attributes than dimension table.
- There is more records than dimension table.
- Fact table forms a vertical table.
- It is used for analysis purpose and decision making.

Dimension tables:

- A dimension table contains dimensions of a fact.
- There is more attributes than fact table.
- There is less records than fact table.
- While dimension table forms a horizontal table.
- While the dimension table has a primary key, the table has a concatenated key.

33. What are Entities and Relationships?

Entity: An entity can be a real-world object, either tangible or intangible, that can be easily identifiable. For example, in a college database, students, professors, workers, departments, and projects can be referred to as entities. Each entity has some associated properties that provide it an identity.

Relationships: Relations or links between entities that have something to do with each other. For example - The employee's table in a company's database can be associated with the salary table in the same database.

34. List the different types of relationships in SQL.

One-to-One - This can be defined as the relationship between two tables where each record in one table is associated with the maximum of one record in the other table.

One-to-Many & Many-to-One - This is the most commonly used relationship where a record in a table is associated with multiple records in the other table. Many-to-Many - This is used in cases when multiple instances on both sides are needed for defining a relationship.

Self-Referencing Relationships - This is used when a table needs to define a relationship with itself.

35. 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.

```
SELECT A.emp_name AS "Employee" /* Alias using AS keyword */
B.emp_name AS "Supervisor"
FROM employee A, employee B /* Alias without AS keyword */
WHERE A.emp_sup = B.emp_id;
```

36. What is a View?

A view in SQL is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

37. What are Aggregate and Scalar functions?

An aggregate function performs operations on a collection of values to return a single scalar value. Aggregate functions are often used with the GROUP BY and HAVING clauses of the SELECT statement. Following are the widely used SQL aggregate functions:

AVG() - Calculates the mean of a collection of values.

COUNT() - Counts the total number of records in a specific table or view. MIN() - Calculates the minimum of a collection of values.

MAX() - Calculates the maximum of a collection of values.
SUM() - Calculates the sum of a collection of values.
FIRST() - Fetches the first element in a collection of values.
LAST() - Fetches the last element in a collection of values.
Note: All aggregate functions described above ignore NULL values except for the COUNT function.
A scalar function returns a single value based on the input value. Following are the widely used SQL scalar functions:
LEN() - Calculates the total length of the given field (column).
UCASE() - Converts a collection of string values to uppercase characters. LCASE() - Converts a collection of string values to lowercase characters. MID() - Extracts substrings from a collection of string values in a table. CONCAT() - Concatenates two or more strings.
RAND() - Generates a random collection of numbers of a given length. ROUND() - Calculates the round-off integer value for a numeric field (or decimal point values).
NOW() - Returns the current date & time.
FORMAT() - Sets the format to display a collection of values.

38. What is User-defined function? What are its various types?

The user-defined functions in SQL are like functions in any other programming language that accept parameters, perform complex calculations, and return a value. They are written to use the logic repetitively whenever required. There are two types of SQL user-defined functions:
Scalar Function: As explained earlier, user-defined scalar functions return a single scalar value.
Table-Valued Functions: User-defined table-valued functions return a table as output.
Inline: returns a table data type based on a single SELECT statement. Multi-statement: returns a tabular result-set but, unlike inline, multiple SELECT statements can be used inside the function body.

39. What is OLTP?

OLTP stands for Online Transaction Processing, is a class of software applications capable of supporting transaction-oriented programs. An essential attribute of an OLTP system is its ability to maintain concurrency. To avoid single points of failure, OLTP systems are often decentralized. These systems are usually designed for a large number of users who conduct short transactions. Database queries are usually simple, require sub-second response times, and return relatively few records. Here is an insight into the working of an OLTP

40. What are the differences between OLTP and OLAP?

OLTP stands for Online Transaction Processing, is a class of software applications capable of supporting transaction-oriented programs. An important attribute of an OLTP system is its ability to maintain concurrency. OLTP systems often follow a decentralized architecture to avoid single points of failure. These systems are generally designed for a large audience of end-users who conduct short transactions. Queries involved in such databases are generally simple, need fast response times, and return relatively few records. A number of transactions per second acts as an effective measure for such systems.

OLAP stands for Online Analytical Processing, a class of software programs that are characterized by the relatively low frequency of online transactions. Queries are often too complex and involve a bunch of aggregations. For OLAP systems, the effectiveness measure relies highly on response time. Such systems are widely used for data mining or maintaining aggregated, historical data, usually in multi dimensional schemas.

41. 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.

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

42. What is Pattern Matching in SQL?

SQL pattern matching provides for pattern search in data if you have no clue as to what that word should be. This kind of SQL query uses wildcards to match a string pattern, rather than writing the exact word. The LIKE operator is used in conjunction with SQL Wildcards to fetch the required information.

Using the % wildcard to perform a simple search

The % wildcard matches zero or more characters of any type and can be used to define wildcards both before and after the pattern. Search a student in your database with first name beginning with the letter K:

```
SELECT *
FROM students
WHERE first_name LIKE 'K%'
```

Omitting the patterns using the NOT keyword

Use the NOT keyword to select records that don't match the pattern. This query returns all students whose first name does not begin with K.

```
SELECT *
FROM students
WHERE first_name NOT LIKE 'K%'
```

Matching a pattern anywhere using the % wildcard twice

Search for a student in the database where he/she has a K in his/her first name.

```
SELECT *
FROM students
```

```
WHERE first_name LIKE '%Q%'
```

Using the _ wildcard to match pattern at a specific position

The _ wildcard matches exactly one character of any type. It can be used in conjunction with % wildcard. This query fetches all students with letter K at the third position in their first name.

```
SELECT *
FROM students
WHERE first_name LIKE '__K%'
```

43. Matching patterns for a specific length

The _ wildcard plays an important role as a limitation when it matches exactly one character. It limits the length and position of the matched results.

For example -

```
SELECT * /* Matches first names with three or more letters */
FROM students
WHERE first_name LIKE '___%'
SELECT * /* Matches first names with exactly four characters */
FROM students
WHERE first_name LIKE '____'
```

44. What is correlated and sub-queries in sql?

In a situation where inner query and outer query want to execute on a common group by condition. Then we use correlated sub-queries

45. How do you get the best revenue in a particular location ?Without using primary key. How you find duplicate fields in SQL ?

By using Unique Constraint

46. How can u load Text file data into MySQL database? Tell the script query?

```
LOAD DATA LOCAL INFILE ‘<PATH_OF_THE_FILE>’ INTO TABLE <TABLE_NAME>;
```

Ex:- LOAD DATA LOCAL INFILE ‘/home/hadoop/mysql/inputs’ INTO TABLE mytable;