COMPREHENSIVE GUIDE TO INTERVIEWS FOR DATA SCIENCE

SQL

ZEP ANALYTICS

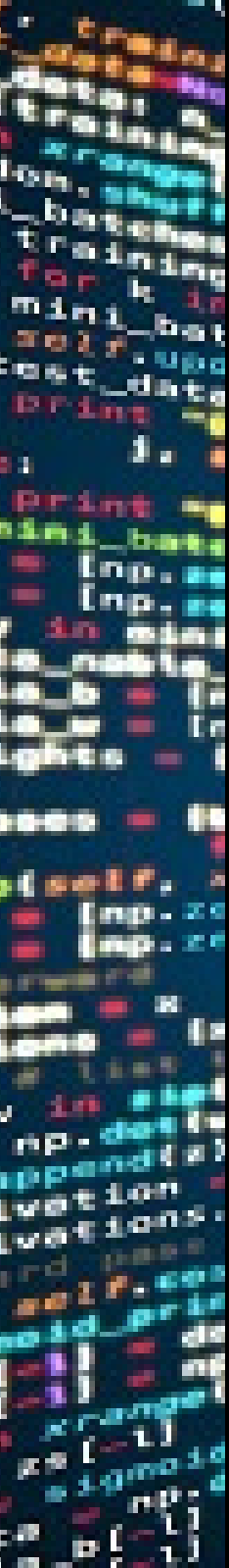
Introduction

We've curated this series of interview guides to accelerate your learning and your mastery of data science skills and tools.

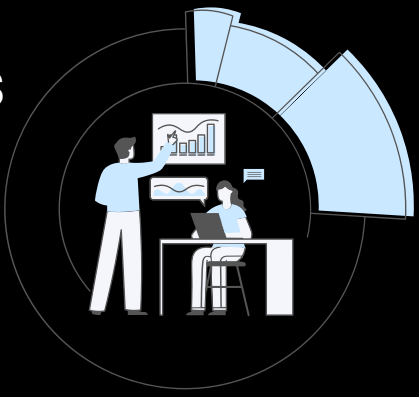
From job-specific technical questions to tricky behavioral inquires and unexpected brainteasers and guesstimates, we will prepare you for any job candidacy in the fields of data science, data analytics, or BI analytics.

These guides are the result of our data analytics expertise, direct experience interviewing at companies, and countless conversations with job candidates. Its goal is to teach by example - not only by giving you a list of interview questions and their answers, but also by sharing the techniques and thought processes behind each question and the expected answer.

Become a global tech talent and unleash your next, best self with all the knowledge and tools to succeed in a data analytics interview with this series of guides.



COMPREHENSIVE GUIDE TO INTERVIEWS FOR DATA SCIENCE



Data Science interview questions cover a wide scope of multidisciplinary topics. That means you can never be quite sure what challenges the interviewer(s) might send your way.

That being said, being familiar with the type of questions you can encounter is an important aspect of your preparation process.

Below you'll find examples of real-life questions and answers. Reviewing those should help you assess the areas you're confident in and where you should invest additional efforts to improve.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1. What is Relational Database Management System (RDMBS)?

RDBMS store data into a collection of tables, which is related by common fields between the columns of the table. It also provides relational operators to manipulate the data stored into the tables.
Example: SQL Server.

2. What is Structured Query Language?

SQL stands for Structured Query Language , and it is used to communicate with the Database. This is a standard language used to perform tasks such as retrieval, updates, insertion and deletion of data from a database.
Standard SQL Commands are Select.

3. What is a Database?

A Database is an organized form of data for easy access, storing, retrieval and managing of data. This is also known as structured form of data which can be accessed in many ways.
Example: School Management Database, Bank Management Database.

4. What is primary key?

A primary key is a combination of fields which uniquely specify a row. This is a special kind of unique key, and it has implicit NOT NULL constraint. This means, Primary key values cannot be NULL.

5. What is a unique key?

A Unique key constraint uniquely identifies each record in a database. This provides uniqueness for the column or set of columns. A Primary key constraint has automatic unique constraint defined on it. There can be many unique constraints defined per table, but only one Primary key constraint defined per table.

6. What is a foreign key?

A foreign key is one table which can be related to the primary key of another table. Relationships need to be created between two tables by referencing the foreign key with the primary key of another table.

7. Explain the difference between spreadsheets and databases.

- Spreadsheet:

A file that exists of cells in rows and columns and can help arrange, calculate and sort data. It can have numeric values, text, formulas and functions. It features columns and rows to keep inserted information legible and simple to understand. It is an electronic graph sheet.

- Database:

It is an organized collection of data arranged for ease and speed of search and retrieval. It contains multiple tables. A database engine can sort, change or serve the information on the database. Basically, it is a set of information which is held in a computer.

8. What are table and fields?

A table is a set of data that are organized in a model with Columns and Rows. Columns can be categorized as vertical, and Rows are horizontal. A table has a specified number of column called fields but can have any number of rows which are called records.

Example:

Table: Employee.

Field: Emp ID, Emp Name, Date of Birth.

Data: 201456, David, 11/15/1960.

9. Explain the various SQL languages.

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.

- Data Definition Language (DDL)

DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc. All the commands of DDL are auto-committed which means that it permanently saves all the changes in the database.

Some commands that come under DDL:

CREATE; ALTER; DROP; TRUNCATE

- Data Manipulation Language

DML commands are used to modify the database. It is responsible for all forms of changes in the database. The commands of DML are not auto-committed which means that it can't permanently save all the changes in the database.

Some commands that come under DML:

INSERT; UPDATE; DELETE

- Data Control Language

DCL commands are used to grant and take back authority from any database user.
Some commands that come under DCL:
Grant; Revoke

- Transaction Control Language

TCL commands can only be used with DML commands like INSERT, DELETE and UPDATE. These operations are automatically committed in the database, which is why they cannot be used while creating tables or dropping them.
Some commands that come under TCL:
COMMIT; ROLLBACK; SAVEPOINT

- Data Query Language

DQL is used to fetch the data from the database.
It uses only one command:
SELECT

10. What is normalization?

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

11. What is denormalization?

Denormalization is a technique used to access the data from higher to lower normal forms of database. It is also a process of introducing redundancy into a table by incorporating data from the related tables.

12. Explain the different types of normalization.

Some types are:

- First Normal Form (1NF): This should remove all the duplicate columns from the table. Creation of tables for the related data and identification of unique columns.
- Second Normal Form (2NF): 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.
- Third Normal Form (3NF): This should meet all requirements of 2NF. Removing the columns which are not dependent on primary key constraints.
- Fourth Normal Form (4NF): Meeting all the requirements of third normal form and it should not have multi- valued dependencies.

13. What are views in SQL?

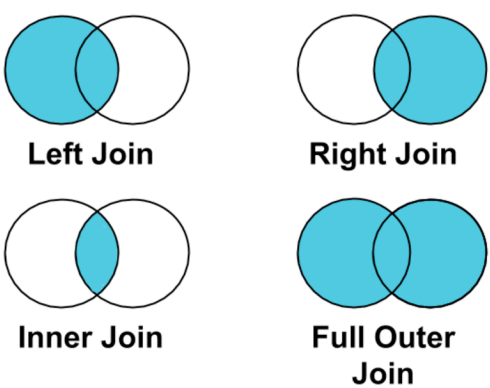
A view is a virtual table which consists of a subset of data contained in a table. Views are not virtually present, and it takes less space to store. View can have data of one or more tables combined, and it is depending on the relationship.

14. What is join? Explain the different types.

This is a keyword used to query data from more tables based on the relationship between the fields of the tables. Keys play a major role when JOINs are used.

There are various types of joins which can be used to retrieve data and it depends on the relationship between tables.

- Left Outer Join: If we want all the records from left table and only matching records from right table then will use left outer join/left join.
- Right Outer Join: If we want to display all the records from right table and only matching records from left table then will right outer join/right join.
- Full Outer Join: If we want to display all the records from both the tables then will use full outer join.
- Inner Join: If we want only the matching records from both the tables then will use Inner join/Simple join.



15. What are the different types of indexes?

An index is a performance tuning method of allowing faster retrieval of records from the table. An index creates an entry for each value and makes it faster to retrieve data.

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. Each table can have 999 non-clustered indexes.

16. What is a cursor in SQL?

A database Cursor is a control which enables traversal over the rows or records in the table. This can be viewed as a pointer to one row in a set of rows. Cursor is very much useful for traversing such as for retrieval, addition and removal of database records.

17. What is query?

A DB query is a code written in order to get the information back from the database. Queries can be designed in such a way that it matches with our expectation of the result set.

18. What is a subquery?

A subquery is a query within another 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.

19. What is a trigger?

A DB trigger is a code or programs 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 such as the Exam, Score and Attendance tables.

20. Differentiate between the DELETE and TRUNCATE commands.

- DELETE command is used to remove rows from the table, and WHERE clause can be used for conditional set of parameters. Commit and Rollback can be performed after delete statement.
- TRUNCATE removes all rows from the table. Truncate operation cannot be rolled back.

21. What are local and global variables?

- 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 to 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.

22. What are constraints?

Constraint can be used to specify the limit on the data type of table. Constraint can be specified while creating or altering the table statement.

23. 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.

24. What is auto increment?

Auto increment keyword allows the user to create a unique number to be generated when a new record is inserted into the table. AUTO INCREMENT keyword can be used in Oracle and IDENTITY keyword can be used in SQL SERVER.

25. What is a data warehouse?

Data warehouses are a central repository of data from multiple sources of information. This data is consolidated, transformed and made available for the mining and online processing. Warehouse data have subsets of data called Data Marts.

26. What is the difference between DROP and TRUNCATE statements?

TRUNCATE removes all the rows from the table, and it cannot be rolled back. DROP command removes a table from the database and operation cannot be rolled back.

27. What are aggregate and scalar functions?

Functions are methods used to perform data operations. SQL has many in-built functions used to perform string concatenations, mathematical calculations etc.

SQL functions are categorized into the following two categories: Aggregate Functions and Scalar Functions.

- Aggregate SQL Functions

The Aggregate Functions in SQL perform calculations on a group of values and then return a single value. Following are a few of the most commonly used Aggregate Functions:

Function	Description
SUM()	Used to return the sum of a group of values.
COUNT()	Returns the number of rows either based on a condition, or without a condition.
AVG()	Used to calculate the average value of a numeric column.
MIN()	This function returns the minimum value of a column.
MAX()	Returns a maximum value of a column.
FIRST()	Used to return the first value of the column.
LAST()	This function returns the last value of the column.

- Scalar SQL Functions

The Scalar Functions in SQL are used to return a single value from the given input value. Following are a few of the most commonly used Scalar Functions:

Function	Description
LCASE()	Used to convert string column values to lowercase
UCASE()	This function is used to convert a string column values to Uppercase.
LEN()	Returns the length of the text values in the column.
MID()	Extracts substrings in SQL from column values having String data type.
ROUND()	Rounds off a numeric value to the nearest integer.
NOW()	This function is used to return the current system date and time.
FORMAT()	Used to format how a field must be displayed.

28. What is alias in SQL?

SQL aliases are used to give a table, or a column in a table, a temporary name. Aliases are often used to make column names more readable. An alias only exists for the duration of that query. An alias is created with the AS keyword.

29. What is the difference between OLTP and OLAP?

- OLAP

Online Analytical Processing, a category of software tools which provide analysis of data for business decisions. OLAP systems allow users to analyze database information from multiple database systems at one time.
The primary objective is data analysis and not data processing.

- OLTP

Online transaction processing shortly known as OLTP supports transaction-oriented applications in a 3-tier architecture. OLTP administers day to day transaction of an organization.
The primary objective is data processing and not data analysis. Unlike OLAP systems, the goal of OLTP systems is serving real-time transactions.

30. What is collation? What are the various types of collation sensitivity?

Collation is defined as a set of rules that determine how character data can be sorted and compared.
ASCII value can be used to compare these character data.

- 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.

31. How can we create tables in SQL?

The command to create a table in SQL is extremely simple:
We will start off by giving the keywords, CREATE TABLE, then we will give the name of the table. After that in braces, we will list out all the columns along with their data types.

For example, if we want to create a simple employee table:

```
CREATE TABLE employee (
        name varchar(25),
        age int,
        gender varchar(25),
  ....
);
```

32. How can we insert data in SQL?

It is possible to write the INSERT INTO statement in two ways:

1. Specify both the column names and the values to be inserted:

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

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

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

33. How can we change a table name in SQL?

We will start off by giving the keywords ALTER TABLE, then we will follow it up by giving the original name of the table, after that, we will give in the keywords RENAME TO and finally, we will give the new table name.

For example, if we want to change the "employee" table to "employee_information", this will be the command:

```
ALTER TABLE employee
RENAME TO employee_information;
```

34. What is SQL server?

SQL server has stayed on top as one of the most popular database management products ever since its first release in 1989 by Microsoft Corporation. The product is used across industries to store and process large volumes of data. It was primarily built to store and process data that is built on a relational model of data.

SQL Server is widely used for data analysis and also scaling up of data. SQL Server can be used in conjunction with Big Data tools such as Hadoop.

SQL Server can be used to process data from various data sources such as Excel, Table, .Net Framework application, etc.

35. What is ETL in SQL?

ETL stands for Extract, Transform and Load. It is a three step process, where we would have to start off by extracting the data from sources. Once we collate the data from different sources, we have our raw data. This raw data has to be transformed into a tidy format, which will come in the second phase. Finally, we would have to load this tidy data into tools which would help us to find insights.

36. What are nested queries?

Triggers may implement DML by using INSERT, UPDATE, and DELETE statements. These triggers that contain DML and find other triggers for data modification are called Nested Triggers.

37. What is the difference between CHAR and VARCHAR2 data types in SQL server?

When stored in a database, varchar2 uses only the allocated space. E.g. if you have a varchar2(1999) and put 50 bytes in the table, it will use 52 bytes.
But when stored in a database, char always uses the maximum length and is blank-padded. E.g. if you have char(1999) and put 50 bytes in the table, it will consume 2000 bytes.

38. What is difference between SQL and PL/SQL?

SQL is a Structured Query Language to create and access databases whereas PL/SQL comes with procedural concepts of programming languages.

39. What is the difference between SQL and MySQL?

SQL is a Structured Query Language that is used for manipulating and accessing the relational database. On the other hand, MySQL itself is a relational database that uses SQL as the standard database language.

40. What is 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.

41. What are 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.

42. What is a CLAUSE?

SQL clause is defined to limit the result set by providing condition to the query. This usually filters some rows from the whole set of records.
Example – Query that has WHERE condition.

43. What is recursive stored procedure?

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

44. Explain UNION, MINUS and INTERACT 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 to return rows returned by both the queries.

45. What TCP/IP port does SQL Server run?

By default, SQL Server runs on port 1433.

46. Which operator is used in query for pattern matching?

LIKE operator is used for pattern matching, and it can be used with:

- % - Matches zero or more characters.
- _ (Underscore) – Matching exactly one character.

47. How can we select unique records from a Table?

Select unique records from a table by using DISTINCT keyword.

48. List and explain each of the ACID properties that collectively guarantee that database transactions are processed reliably.

ACID Properties are used for maintaining the integrity of database during transaction processing. ACID in DBMS stands for Atomicity, Consistency, Isolation, and Durability.

- Atomicity: A transaction is a single unit of operation. You either execute it entirely or do not execute it at all. There cannot be partial execution.
- Consistency: Once the transaction is executed, it should move from one consistent state to another.
- Isolation: Transaction should be executed in isolation from other transactions. During concurrent transaction execution, intermediate transaction results from simultaneously executed transactions should not be made available to each other.
- Durability: After successful completion of a transaction, the changes in the database should persist, even in the case of system failures.

49. What is the main difference in the 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);

50. What are SQL functions used for?

SQL functions are used for the following purposes:

- To perform some calculations on the data
- To modify individual data items
- To manipulate the output
- To format dates and numbers
- To convert the data types

51. What is the need for MERGE statement?

This statement allows conditional update or insertion of data into a table. It performs an UPDATE if a row exists, or an INSERT if the row does not exist.

52. List the ways in which dynamic SQL can be executed.

- Write a query with parameters.
- Using EXEC.
- Using sp_executesql.

53. 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')

54. Is semicolon used after sql? Justify why or why not.

Some database systems require a semicolon at the end of each SQL statement. Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.

55. What is candidate key?

A candidate key is a subset of a super key set where the key which contains no redundant attribute is none other than a Candidate Key. In order to select the candidate keys from the set of super key, we need to look at the super key set.

56. What is the difference between JOIN and UNION?

- JOIN

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

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.

57. 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. If we omit the sorting order, it sorts the summarized result in the ascending order by default. The ORDER BY clause, like the GROUP BY clause, could be used in conjunction with the SELECT statement. ASC denotes ascending order, while DESC denotes descending order.

The following is the syntax to use the ORDER BY clause in a SQL statement:

- SELECT expressions
- FROM tables
- [WHERE conditions]
- ORDER BY expression [ASC | DESC];

GROUP BY

The GROUP BY clause is used in SQL queries to organize data that have the same attribute values. Usually, we use it with the SELECT statement. It is important to remember that we have to place the GROUP BY clause after the WHERE clause. Additionally, it is paced before the ORDER BY clause.

We can often use this clause in collaboration with aggregate functions like SUM, AVG, MIN, MAX, and COUNT to produce summary reports from the database. It's important to remember that the attribute in this clause must appear in the SELECT clause, not under an aggregate function. If we do so, the query would be incorrect. As a result, the GROUP BY clause is always used in conjunction with the SELECT clause. The query for the GROUP BY clause is grouped query, and it returns a single row for each grouped object.

The following is the syntax to use GROUP BY clause in a SQL statement:

- SELECT column_name, function(column_name)
- FROM table_name
- WHERE condition
- GROUP BY column_name;

58. Write an SQL query to fetch employee names having a salary greater than or equal to 20000 and less than or equal to 10000.

By using BETWEEN in the where clause, we can retrieve the Employee Ids of employees with salary >= 20000 and <=10000.

- e.g.
- SELECT FullName
- FROM EmployeeDetails
- WHERE EmpId
- IN (SELECT EmpId FROM EmployeeSalary WHERE Salary BETWEEN 0 AND 10000)

59. What is SQL injection? When does SQL injection occur?

SQL Injection is a type of database attack technique where malicious SQL statements are inserted into an entry field of database in a way that once it is executed, the database is exposed to an attacker for the attack. This technique is usually used for attacking data-driven applications to have access to sensitive data and perform administrative tasks on databases.

60. What is ENUM?

An ENUM is a string object with a value chosen from a list of permitted values that are enumerated explicitly in the column specification at table creation time.

61. What is the difference between the ATAN and ATAN2 function?

- ATAN() Function

ATAN() function in MySQL is used to return the arc tangent of any number x. The arctangent of x is defined as the inverse tangent function of x when x is real ($x \in \mathbb{R}$).

- ATAN2() Function

ATAN2() function in MySQL is used for returning the arc tangent between specified two numbers, i.e., x and y. It returns the angle between the positive x-axis and the line from the origin to the point (y, x).

62. What is the difference between the CEIL, FLOOR and ROUND functions?

- ROUND - Rounds a positive or negative value to a specific length.
- CEILING - Evaluates the value on the right side of the decimal and returns the smallest integer greater than, or equal to, the specified numeric expression.
- FLOOR - Evaluates the value on the right side of the decimal and returns the largest integer less than or equal to the specified numeric expression.

63. What is the RAND() function?

The RAND() function will return a value between 0 (inclusive) and 1 (exclusive). The RAND() function will return a completely random number if no seed is provided, and a repeatable sequence of random numbers if a seed value is used.

64. What is the difference between LOCALTIMESTAMP and CURRENT_TIMESTAMP?

LOCALTIMESTAMP returns only time stamp value where as the function CURRENT_TIMESTAMP will return time stamp with Time Zone value.

65. Name three functions that specify current date and time.

SQL Server provides several different functions that return the current date time including: GETDATE(), SYSDATETIME(), and CURRENT_TIMESTAMP.

66. Which function returns the difference between two periods? What would the format of the output be?

DATEDIFF() is a basic SQL Server function that can be used to do date math. Specifically, it gets the difference between 2 dates with the results returned in date units specified as years, months, days, minutes, seconds as an int (integer) value.

67. How can we fetch common records from two tables?

Intersection A ∩ B of two sets A and B is the set, which contains all the elements of A, which also belong to B (or equivalently, all elements of B that also belong to A), but no other elements.

Let A={ Orange, pineapple, banana} and let B={ spoon,Orange, pineapple, mango}

A ∩ B = {Orange, pineapple}

- Select * from student
- Select * from student1
- (Select * from student) Intersect (Select * from student1)

68. How can we fetch alternate records from a table?

Records can be fetched for both Odd and Even row numbers.

- To display even numbers

Select employeeId from (Select row no, employeeId from employee) where mod(row no,2)=0

- To display odd numbers

Select employeeId from (Select rowno, employeeId from employee) where mod(row no,2)=1

69. How can we select unique records from a table?

Select unique records from a table by using the DISTINCT keyword.

70. What is the command used to fetch the first 5 characters of the string?

SELECT SUBSTRING('SQL Tutorial', 1, 5) AS ExtractString;

71. How to use LIKE in SQL?

The LIKE operator checks if an attribute value matches a given string pattern. Here is an example of LIKE operator:

SELECT * FROM employees WHERE first_name like 'Steven';

With this command, we will be able to extract all the records where the first name is like "Steven".

72. How can we copy a table in SQL?

We can use the SELECT INTO statement to copy data from one table to another. Either we can copy all the data or only some specific columns.

This is how we can copy all the columns into a new table:

- SELECT *
- INTO newtable
- FROM oldtable
- WHERE condition;

If we want to copy only some specific columns, we can do it this way:

- SELECT column1, column2, column3, ...
- INTO newtable
- FROM oldtable
- WHERE condition;

73. If we drop a table, does it also drop related objects such as constraints, indexes, columns, default, views and sorted procedures?

Yes, SQL server drops all related objects, which exists inside a table like constraints, index, columns, defaults etc. However, dropping a table will not drop views and sorted procedures as they exist outside the table.

74. What is Live Lock?

A live lock is one wherein a request for an exclusive lock is repeatedly denied because a series of overlapping shared locks keep interfering.

75. Can you join a table by itself?

A table can be joined to itself using self join, when you want to create a result set that joins records in a table with other records in the same table.

76. Explain Equi join with an example.

When two or more tables has been joined using equal to operator then this category is called as equi join.

Example:

Select a.Employee_name, b.Department_name from Employee a, Employee b where a.Department_ID = b.Department_ID

77. Explain non-Equi join with an example.

When two or more tables are joining without an equal to condition then that join is known as Non Equi Join. Any operator can be used here, that is < >,!=,<,>,Between.

Example:

Select b.Department_ID, b.Department_name from Employee a, Department b where a.Department_id < > b.Department_ID;

78. State the difference between NVL and NVL2 functions.

Both the NVL(exp1, exp2) and NVL2(exp1, exp2, exp3) functions check the value exp1 to see if it is null. With the NVL(exp1, exp2) function, if exp1 is not null, then the value of exp1 is returned; otherwise, the value of exp2 is returned. With the NVL2(exp1, exp2, exp3) function, if exp1 is not null, then exp2 is returned; otherwise, the value of exp3 is returned.

79. What does this query achieve? GRANT privilege_name ON object_name TO {user_name|PUBLIC|role_name} [WITH GRANT OPTION]; ?

The given syntax indicates that the user can grant access to another user too.

80. Where is MyISAM table stored?

Each MyISAM table is stored on disk in three files.

- The ".frm" file stores the table definition.
- The data file has a '.MYD' (MYData) extension.
- The index file has a '.MYI' (MYIndex) extension.

81. What does myisamchk do?

It compresses the MyISAM tables, which reduces their disk or memory usage.

82. How can we store videos inside SQL server table?

By using FILESTREAM datatype, which was introduced in SQL Server 2008.

83. Write an SQL query to show the second highest salary from a table.

Below is the syntax to find 2nd highest salary in SQL:

- SELECT name, MAX(salary)
- FROM employees
- WHERE salary < (SELECT MAX(salary)
- FROM employees);

84. How would you select all the users whose phone number is NULL?

SELECT user_name FROM users WHERE ISNULL(user_phonenumber);

85. Write an SQL query to fetch three max salaries from a table.

SELECT TOP 1 salary FROM (SELECT TOP 3 salary FROM employee_table ORDER BY salary DESC) AS emp ORDER BY salary ASC;

86. Write an SQL query to create a new table with data and structure copied from another table.

Using SELECT INTO command- SELECT * INTO newTable FROM EmployeeDetails;

87. What are the differences between the HAVING clause, and the WHERE clause?

S. No.	Where Clause	Having Clause
1	The WHERE clause specifies the criteria which individual records must meet to be selected by a query. It can be used without the GROUP by clause	The HAVING clause cannot be used without the GROUP BY clause
2	The WHERE clause selects rows before grouping	The HAVING clause selects rows after grouping
3	The WHERE clause cannot contain aggregate functions	The HAVING clause can contain aggregate functions
4	WHERE clause is used to impose a condition on SELECT statement as well as single row function and is used before GROUP BY clause	HAVING clause is used to impose a condition on GROUP Function and is used after GROUP BY clause in the query
5	SELECT Column,AVG(Column_nmae)FROM Table_name WHERE Column > value GROUP BY Column_nmae	SELECT Columnq, AVG(Coulmn_nmae)FROM Table_name WHERE Column > value GROUP BY Column_nmae Having column_name>or<value

88. What does a BCP command do?

The Bulk Copy is a utility or a tool that exports/imports data from a table into a file and vice versa.

89. Can a view be active if the base table is dropped?

No, the view cannot be active if the parent table is dropped.

90. When should we use NoSQL and SQL?

SQL stands for structured query language and is majorly used to query data from relational databases. When we talk about a SQL database, it will be a relational database.
But when it comes to NoSQL database, we will be working with non-relational databases.

91. What is SYSTEM privilege?

Rights are given to a user, usually by the DBA, to perform a particular action on the database schema objects like creating tablespaces.

The following are examples of system privileges that can be granted to users:

- CREATE TABLE allows a grantee to create tables in the grantee's schema.
- CREATE USER allows a grantee to create users in the database.
- CREATE SESSION allows a grantee to connect to an Oracle database to create a user session.

92. What are object privileges?

An object-level privilege is a permission granted to a database user account or role to perform some action on a database object. These object privileges include SELECT, INSERT, UPDATE, DELETE, ALTER, INDEX on tables, and so on.

The following example is of object privileges that can be granted to users:
SELECT ON hr.employees TO myuser INSERT ON hr.employees TO myuser

93. Does the data stored in the stored procedure increase access time or execution time? Explain.

Data stored in stored procedures can be retrieved much faster than the data stored in the SQL database. Data can be precompiled and stored in stored procedures. This reduces the time gap between query and compiling as the data has been pre-compiled and stored in the procedure.

94. What is CTE?

A CTE or common table expression is an expression that contains temporary result set which is defined in a SQL statement.

95. Does view contain data?

No, Views are virtual structures.

96. Define a temp table.

A temp table is a temporary storage structure to store the data temporarily.

97. What is the difference between the RANK() and DENSE_RANK() function?

The only difference between the RANK() and DENSE_RANK() functions is in cases where there is a "tie"; i.e., in cases where multiple values in a set have the same ranking. In such cases, RANK() will assign non-consecutive "ranks" to the values in the set (resulting in gaps between the integer ranking values when there is a tie), whereas DENSE_RANK() will assign consecutive ranks to the values in the set (so there will be no gaps between the integer ranking values in the case of a tie).

For example, consider the set {25, 25, 50, 75, 75, 100}. For such a set, RANK() will return {1, 1, 3, 4, 4, 6} (note that the values 2 and 5 are skipped), whereas DENSE_RANK() will return {1,1,2,3,3,4}.

98. What is referential integrity?

Set of rules that restrict the values of one or more columns of the tables based on the values of the primary key or unique key of the referenced table.

99. What does query optimization imply?

Query optimization is a process in which a database system compares different query strategies and selects the query with the least cost.

100. What are nested triggers?

Triggers may implement data modification logic by using INSERT, UPDATE, and DELETE statements. These triggers that contain data modification logic and find other triggers for data modification are called Nested Triggers.

104. What is CTE in SQL server?

CTEs are Common Table Expressions that are used to create temporary result tables from which data can be retrieved/ used. The standard syntax for a CTE with a SELECT statement is:

- WITH RESULT AS
- (SELECT COL1, COL2, COL3
- FROM EMPLOYEE)
- SELECT COL1, COL2 FROM RESULT

CTEs can be used with Insert, Update or Delete statements as well.

Few examples of CTEs are given below:

- Query to find the 10 highest salaries with result as:

(select distinct salary, dense_rank() over (order by salary desc) as salary rank from employees)
select result. salary from result where the result.salaryrank = 10

- Query to find the 2nd highest salary with the result as:

(select distinct salary, dense_rank() over (order by salary desc) as salaryrank from employees)
select result. salary from result where the result.salaryrank = 2

In this way, CTEs can be used to find the nth highest salary within an organisation.

101. What is schema in SQL server?

Our database comprises of a lot of different entities such as tables, stored procedures, functions, database owners and so on. To make sense of how all these different entities interact, we would need the help of schema. So, you can consider schema to be the logical relationship between all the different entities which are present in the database.

Once we have a clear understanding of the schema, this helps in a lot of ways:

- We can decide which user has access to which tables in the database.
- We can modify or add new relationships between different entities in the database.

Overall, you can consider a schema to be a blueprint for the database, which will give you the complete picture of how different objects interact with each other and which users have access to different entities.

102. Write a query to fetch 50% records from an EmployeeInfo table.

```
Select top 50 percent * from Employee;
```

103. Write a query to add email validation to your database.

```
SELECT * FROM student
WHERE s_email LIKE '%@gmail.com';
```

105. Suppose you have a sample table of Workers with columns Worker_id, first_name,last_name, salary, join_date, department. We have another table bonus with columns worker_ref_id, bonus date, bonus_amt. We also have another table called title and it has cols like worker_ref_id, worker_title, affected_from.

Write an SQL query to print the FIRST_NAME and LAST_NAME from Worker table into a single column COMPLETE_NAME. A space char should separate them.

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

Write an SQL query to fetch duplicate records having matching data in some fields of a table.

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

Write an SQL query to print the name of employees having the highest salary in each department.

```
SELECT t.DEPARTMENT,t.FIRST_NAME,t.Salary
from(SELECT max(Salary) as TotalSalary,DEPARTMENT
from Worker group by DEPARTMENT) as TempNew
Inner Join Worker t on
TempNew.DEPARTMENT=t.DEPARTMENT and
TempNew.TotalSalary=t.Salary;
```

Write an SQL query that fetches the unique values of DEPARTMENT from Worker table and prints its length.

```
Select distinct length(DEPARTMENT) from Worker;
```

106. Write a query to fetch the top N records.

The SELECT TOP clause allows you to limit the number of rows or percentage of rows returned in a query result set.

Because the order of rows stored in a table is unspecified, the SELECT TOP statement is always used in conjunction with the ORDER BY clause. Therefore, the result set is limited to the first N number of ordered rows.

The following shows the syntax of the TOP clause with the SELECT statement:

- SELECT TOP (expression) [PERCENT][WITH TIES]
- FROM table_name
- ORDER BY column_name;

In this syntax, the SELECT statement can have other clauses such as WHERE, JOIN, HAVING, and GROUP BY.

Social Media Company Interview Qs (e.g. Facebook)

Column Name	Data Type
id	int
time_id	datetime
user_id	varchar
customer_id	varchar
client_id	varchar
event_type	varchar
event_id	int

1. Find the new users which are defined as users that have started using the services for the first time.

We can find this by finding the minimum date from the 'time_id' column for each user, which gives the date they started using services.

```
SELECT user_id,
min(time_id) as new_user_start_date
FROM fact_events
GROUP BY user_id
```

2. Calculate the count of new users by month by extracting the month from the date and counting unique users.

```
SELECT date_part('month', new_user_start_date) AS
month,
  count(DISTINCT user_id) as new_users
FROM (SELECT user_id, min(time_id) as
new_user_start_date FROM fact_events
    GROUP BY user_id) sq
GROUP BY month
```

3. Calculate all users (existing and new) for each month. This will give us existing users once we subtract out the new users.

```
SELECT date_part('month', time_id) AS month,
       count(DISTINCT user_id) as all_users
FROM fact_events
GROUP BY month
```

4. Join the two tables together by month.

```
with all_users as (
    SELECT date_part('month', time_id) AS month,
           count(DISTINCT user_id) as all_users
    FROM fact_events
    GROUP BY month),
new_users as (
    SELECT date_part('month', new_user_start_date) AS
month,
           count(DISTINCT user_id) as new_users
    FROM
        (SELECT user_id,
         min(time_id) as new_user_start_date
         FROM fact_events
         GROUP BY user_id) sq
    GROUP BY month
)
SELECT
  *
FROM all_users au
JOIN new_users nu ON nu.month = au.month
```

5. Calculate user shares.

```
with all_users as (
    SELECT date_part('month', time_id) AS month,
           count(DISTINCT user_id) as all_users
    FROM fact_events
    GROUP BY month),
new_users as (
    SELECT date_part('month', new_user_start_date) AS
month,
           count(DISTINCT user_id) as new_users
    FROM
        (SELECT user_id,
          min(time_id) as new_user_start_date
         FROM fact_events
         GROUP BY user_id) sq
    GROUP BY month
)
SELECT
  au.month,
  new_users / all_users::decimal as share_new_users,
  1- (new_users / all_users::decimal) as
share_existing_users
FROM all_users au
JOIN new_users nu ON nu.month = au.month
```

Audio Streaming Service Company Interview Qs (e.g. Spotify)

Song

Column Name	Data Type
id	int
name	varchar
artist_id	int

Daily_Plays

Column Name	Data Type
date	datetime
country	varchar
song_id	int
plays	int

Artist

Column Name	Data Type
id	int
name	varchar

1. Write a query to return top 5 songs in the UK yesterday.

```
SELECT
   S.song_id,
   S.name
FROM song_plays P
INNER JOIN song S
ON P.song_id = S.id
WHERE P.country = 'UK'
AND P.date = CURRENT_DATE - 1
ORDER BY daily_plays DESC
LIMIT 5;
```

2. Write a query to return the top 5 artists in the US and UK yesterday.

```
WITH artist_ranking AS (
  SELECT
    A.artist_id,
    MAX(A.artist_name) AS artist_name,
    MAX(P.country) AS country
    ROW_NUMBER() OVER(PARTITION BY country ORDER
BY SUM(plays) DESC) AS ranking
  FROM daily_plays P
  INNER JOIN song S
  ON P.song_id = S.id
  INNER JOIN artist A ON
  A.artist_id = S.artist_id
  WHERE P.country IN ('UK', 'US')
  AND P.date = CURRENT_DATE - 1
  GROUP BY A.artist_id
)
SELECT artist_id, artist_name, country, ranking
FROM artist_ranking
WHERE ranking <= 5
LIMIT 5;
```

e-Commerce Company Interview Qs (e.g. Amazon)

1. Assume you are given the below table on purchases from users. Write a query to get the number of people that purchased at least one product on multiple days.

Purchases

Column Name	Data Type
purchase_id	int
user_id	int
product_id	int
quantity	int
price	float
purchase_time	datetime

```
SELECT *
FROM
 (SELECT p.user_id,
 COUNT (DISTINCT purchase_id) as purchase_frequency
 FROM purchase_p
 GROUP BY p.user_id)
 PIVOT
 (COUNT (user_id)
 for purchase_frequency in ('1' one, '2' two, '3' three)
 );
```

2. Assume you are given the table alongside for the session activity of user. Write a query to assign ranks to users by the total session duration for the different session types they have had between a start date (2020-01-01) and an end date (2020-02-01).

sessions

Column Name	Data Type
session_id	int
user_id	int
session_type	varchar
duration	int
start_time	datetime

```
SELECT ss.*,
rank() over (partition by ss.user_id order by
ss.total_duration desc) as rank_order
FROM (select s.user_id,
s.session_type,
sum(s.duration) as total_duration
FROM sessions.s
WHERE s.start_time between '01-jan-20' and '01-feb-20'
GROUP BY s.user_id,
s.session_type)ss
```

3. How many customers placed an order and what is the average order amount?

Column Name	Data Type
id	int
customer_id	int
seller_id	int
order_timestamp_utc	datetime
amount	float
city_id	int

```
SELECT count(DISTINCT customer_id),
       avg(amount)
FROM orders
```

Entertainment Streaming Company Interview Qs (e.g. Netflix)

1. Return the share of monthly active users in United States (U.S). Active users are the ones with the "open" status in the table.

Column Name	Data Type
user_id	int
name	varchar
status	varchar
country	varchar

```
SELECT active_users /total_users::float AS
active_users_share
FROM
  (SELECT count(user_id) total_users,
        count(CASE
                WHEN status = 'open' THEN 1
                ELSE NULL
              END) AS active_users
   FROM fb_active_users
   WHERE country = 'USA') subq
```

2. Using the table given, list the top 10 users who accumulated the most sessions where they had more streaming sessions than viewing. Return the user_id, number of streaming sessions, and the number of viewing sessions.

Column Name	Data Type
user_id	int
session_start	datetime
session_end	datetime
session_id	int
session_type	varchar

```
SELECT user_id,
       count(CASE
                 WHEN session_type='streamer' THEN 1
                 ELSE NULL
             END) AS streaming,
       count(CASE
                 WHEN session_type='viewer' THEN 1
                 ELSE NULL
             END) AS VIEW
FROM twitch_sessions
GROUP BY user_id
HAVING count(CASE
                 WHEN session_type='streamer' THEN 1
                 ELSE NULL
             END) > count(CASE
                              WHEN session_type='viewer' THEN 1
                              ELSE NULL
                          END)
LIMIT 10
```

Financial Institution Interview Qs (e.g. HSBC)

Column Name	Data Type
id	int
balance	float
interest_rate	float
rate_type	varchar
loan_id	int

1. Write a query that return the rate_type, loan_id and balance of each loan type, and a column that shows what percentage of the submission's total balance each loan constitutes.

```
SELECT s1.loan_id,
       s1.rate_type,
       sum(s1.balance) AS balance,
       sum(s1.balance)::decimal/total_balance AS
balance_share
FROM submissions s1
LEFT JOIN
  (SELECT rate_type,
          sum(balance) AS total_balance
   FROM submissions
   GROUP BY rate_type) s2 ON s1.rate_type =
s2.rate_type
GROUP BY s1.loan_id,
         s1.rate_type,
         s2.total_balance
ORDER BY s1.rate_type, s1.loan_id
```

Online Marketplace Interview Qs (e.g. AirBnB)

1. Find the average number of bathrooms and bedrooms for each city and property type.

Column Name	Data Type
id	int
price	float
property_type	object
amenities	object
accommodates	object
bathrooms	int
bed_type	object
cancellation_policy	object
cleaning_fee	bool
city	object
host_identity_verified	object
host_response_rate	object
host_since	object
neighbourhood	object
number_of_reviews	int
review_scores_rating	float
zipcode	int
bedrooms	int
beds	int

```
Select city,
property_type,
avg(bathrooms) as average_bathrooms,
avg(bedrooms) as average_bedrooms
from airnb_search_details
group by city,
property_type;
```

2. Find the min, avg and max log price per review qualification.
The review qualification is categorized by the number of reviews as defined below, along with the associated price
0 reviews : NO
1 to 5 reviews : FEW
5 to 15 reviews : SOME
15 to 40 reviews : MANY
More than 40 reviews : ALOT

```
Select b.qualification_category,
min(b.price),
avg(b.price),
max(b.price)
from
(select a.*,
case when a.number_of_reviews = 0 then 'NO'
    when a.number_of_reviews between 1 and 5 then 'FEW'
        when a.number_of_reviews between 5 and 15
then 'SOME'
        when a.number_of_reviews between 15 and 40
then 'MANY'
        when a.number_of_reviews > 40 then 'ALOT'
        else 'NA' end as qualification_category
from airbnb_search_details a) b
group by qualification_category;
```

Software Company Interview Qs (e.g. Microsoft)

Worker

Column Name	Data Type
worker_id	int
first_name	varchar
last_name	varchar
salary	int
joining_date	datetime
department	varchar

Bonus

Column Name	Data Type
worker_ref_id	int
bonus_date	datetime
bonus_amount	int

Title

Column Name	Data Type
worker_ref_id	int
worker_title	varchar
affected_from	datetime

1. Write query to show the top n (say 10) records of a table in 3 different ways.

- Using LIMIT Method

```
SELECT * FROM Worker ORDER BY Salary DESC LIMIT 10;
```

- Using TOP command

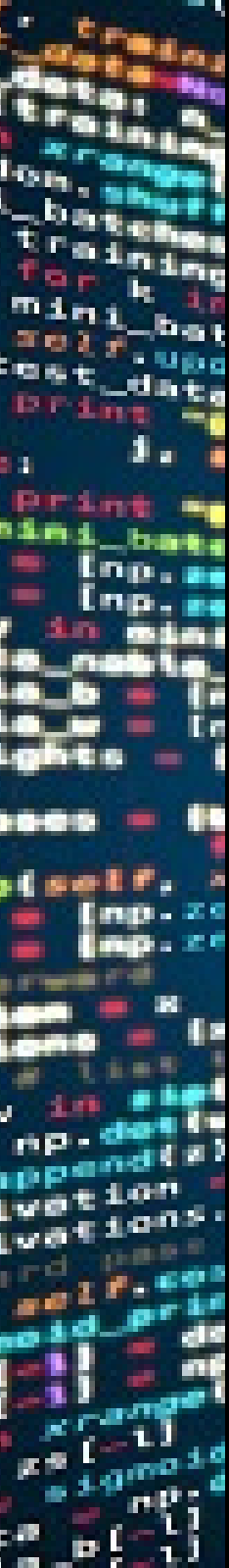
```
SELECT TOP 10 * FROM Worker ORDER BY Salary DESC;
```

- Using ROWNUM

```
SELECT * FROM (SELECT * FROM Worker ORDER BY Salary DESC) WHERE ROWNUM <= 10;
```

2. Write an SQL query to print the name of the employees having the highest salary in each department.

```
SELECT t.department, t.first_name, t.salary from (SELECT MAX(Salary) as total_salary, department FROM Worker GROUP BY Department) as TempNew
INNER JOIN Worker t on TempNew.Department= t.Department
and TempNew.TotalSalary = t.Salary;
```



This brings our list of 120+ SQL interview questions to an end.

We believe these series of guides will help you “expect the unexpected” and enter your first data analytics interview with confidence.

We, at Zep provide a platform for Education, where your demand gets fulfilled. You demand we fulfil all your learning needs without costing you extra.

Ready to take the next steps?

Zep offers a platform for education to learn, grow & earn.

Become a part of the team at Zep

Why don't you start your journey as a tech blogger and enjoy unlimited perks and cash prizes every month.

Explore

zepanalytics.com