SQL Interview Questions

[Top 200 SQL Interview Questions with Answers]

Here is a list of possible interview questions of **SQL**, along with answers to help you prepare for your interview:

1. **What is SQL?**
    - SQL – Structured Query Language, an ANSI (American National Standards Institute) Standard language for accessing databases and It was initially developed at IBM in the 1970s.
    - SQL is the standard language to communicate with relational database management systems like Oracle, MS Access, MS SQL Server, MySQL, DB2, Sybase Etc…
    - SQL is used to Create New Databases, New Tables in a Database, Insert records in a Database, Update records in a Database, and Delete records in a Database.
    - SQL is used to retrieve data from a Database, Execute queries against a Database, Create stored procedures in a Database, Create views in a Database, Set permissions on tables, procedures, and views.
2. **When did SQL launch?**

   Structured Query Language was first launched by IBM in 1974 and it is Free Software (anybody can use with free of cost).
3. **Who should learn SQL?**

- Database Developers
- Database Testers
- Database Administrators

4. **What are the Uses of SQL?**

- Creating new databases
- Creating new tables in a database
- Inserting records in a database
- Updating records in a database
- Deleting records from a database
- Retrieving data from a database

- Executing queries against a database
- Creating stored procedures in a database
- Creating views in a database
- Setting permissions on tables, procedures, and views etc…

5. What is SQL Process?

When we are executing an SQL command for any RDBMS, the system determines the best way to carry out our request, and the SQL engine figures out how to interpret the task.

There are various components included in the process. These components are Query Dispatcher, Optimization engines, Classic Query Engine and SQL query engine, etc. Classic query engine handles all non-SQL queries but SQL query engine won't handle logical files.

6. Is SQL support programming?

Basically, SQL is a programming language (Structured Query Language) but no control flow statements in SQL, it is a command-based programming language.

7. What is the difference between SQL, SQL Server, and MySQL?

- SQL – Structured Query Language used to manage the relational databases like Oracle, MS SQL Server, MySQL, Sybase, etc,
- SQL Server and MySQL are relational database management systems used to store, retrieve, modify and administer a database using SQL.
- MS SQL Server is a commercial Database Management System whereas MYSQL is an Open-Source software.

8. What are the different subsets of SQL?

The important subsets of SQL are:

- DDL – Data Definition Language – It allows you to perform various operations on the database such as CREATE, ALTER, and DELETE objects.
- DML – Data Manipulation Language – It allows you to access and manipulate data. It helps you to insert, update, delete and retrieve data from the database.
- DCL – Data Control Language – It allows you to control access to the database. Example – Grant, Revoke access permissions.

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

A Database Management System (DBMS) is a software application that interacts with the user, applications, and the database itself to capture and analyze data.

A DBMS allows a user to interact with the database. The data stored in the database can be modified, retrieved and deleted and can be of any type like strings, numbers, images, etc.

There are two types of DBMS:

- Relational Database Management System/SQL Database: The data is stored in relations (tables). Example – Oracle.
- Non-Relational Database Management System/NoSQL Database: There is no concept of relations. Example – MongoDB.

10. What is Database Engine?

Software that stores and retrieves data in a database. It may be a self-contained program or the part of DBMS that performs the storage and retrieval operations.

11. What is the difference between DELETE and TRUNCATE statements?

Delete – Delete is a DML (Data Manipulation Language) command, it is used to delete a row in a table, we can roll back data after using delete statement, and It is slower than truncate statement.

Truncate – Truncate is a DDL (Data Definition Language) command, it is used to delete all the rows from a table, we cannot roll back data, and It is faster.

12. What do you mean by the table, field, and record in SQL?

A table refers to a collection of data in an organized manner in form of rows and columns where the rows are known as records and the columns are known as fields.

13. What are the important SQL language elements?

The important SQL Language elements are:

- Identifiers:

  Names of Database Objects such as Tables, Views, Columns, etc,
- Data Types:

  Define the type of data that is contained by a Column
- Constants:

  Symbols that represent specific data types
- Operators:

Perform Arithmetic, Comparison, and Logical Operations

- Functions:

  Built-in Functions to perform specific operations

- Clauses:

  Constituent components of statements and queries.

- Expressions:

  Produce scalar values, or tables containing columns and rows of data.

- Queries:

  Retrieve the data based on specific criteria, this an important element of SQL.

- Statements Etc…

14. What is Database Server?

Database Management Systems provide Database server functionality, database server provides database services to other computer programs or computers.

15. What is the difference between CHAR and VARCHAR data types in SQL?

Both Char and Varchar are used for characters datatype but varchar is used for character strings of variable length whereas Char is used for strings of fixed length.

CHAR Datatype:

It is a datatype in SQL which is used to store character strings of the fixed length specified. If the length of the string is less than set or fixed-length then it is padded with extra blank spaces so that its length became equal to the set length.

VARCHAR Datatype:

It is a datatype in SQL which is used to store character strings of variable length but a maximum of the set length specified. If the length of the string is less than set or fixed-length then it will store as it is without padded with extra blank spaces.

16. What is an Index in SQL? What are the different types of indexes in SQL?

In most SQL projects the data is in millions and billions. Because of the huge volume of the data performance of SQL statements degrades. So there are a lot of different ways to improve the performance of the application, the performance of reports, or the performance of SQL queries.

17. Definition Of Index:

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

There are different types of Indexes in SQL:

- Clustered Index
- Non-Clustered Index
- Unique Index
- Filtered Index
- Columnstore Index
- Hash Index etc,

18. 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.

19. What is a Query in SQL? What are the parameters of SQL Queries?

A query is a question or inquiry about a set of data. We use Structured Query Language (SQL) to retrieve meaningful and relevant information from databases. When building a structure, we pull data from tables and fields.

An SQL query typically includes Query parameter, Column list, Sort list, Options, and Format.

Example: SQL > Select * from employee where department_id=10 and salary>3500

20. What is key in SQL? What are the different types of Keys in SQL?

A Key in DBMS is an attribute or set of attributes that helps you to identify a row(tuple) in a relation(table). They allow you to find the relation between two tables.

21. Different types of Keys in SQL:

- **Super Key** – A super key is a group of single or multiple keys which identifies rows in a table.
- **Primary Key** – is a column or group of columns in a table that uniquely identifies every row in that table.
- **Candidate Key** – is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes.

- **Alternate Key** – is a column or group of columns in a table that uniquely identifies every row in that table.
- **Foreign Key** – is a column that creates a relationship between two tables. The purpose of foreign key is to maintain data integrity and allow navigation between two different instances of an entity.
- **Compound Key** – has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database.
- **Composite Key** – is a combination of two or more columns that uniquely identify rows in a table. The combination of columns guarantees uniqueness, though individual uniqueness is not guaranteed.
- **Surrogate Key** – An artificial key that aims to uniquely identify each record is called a surrogate key. These kinds of keys are unique because they are created when you don't have any natural primary keys.

22. What is the difference between DROP and TRUNCATE commands?

'**DROP'** command removes a table and it cannot be rolled back from the database whereas the '**TRUNCATE'** command removes all the rows from the table.

23. What is Normalization and what are the advantages of it?

Normalization in SQL is the process of organizing data to avoid duplication and redundancy. Some of the advantages of Normalization are:

- Better Database organization
- More Tables with smaller rows
- Efficient data access
- Greater Flexibility for Queries
- Quickly find the information
- Easier to implement Security
- Allows easy modification
- Reduction of redundant and duplicate data
- More Compact Database
- Ensure Consistent data after modification

24. 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 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.

25. What are Entities and Relationships?

Entities: A person, place, or thing in the real world about which data can be stored in a database. Tables store data that represents one type of entity.

For example – A bank database has a customer table to store customer information. The customer table stores this information as a set of attributes (columns within the table) for each customer.

Relationships: Relation or links between entities that have something to do with each other.

For example – The customer's name is related to the customer account number and contact information, which might be in the same table. There can also be relationships between separate tables (for example, customer to accounts).

26. What are joins in SQL?

A JOIN clause is used to combine rows from two or more tables, based on a related column between them. It is used to merge two tables or retrieve data from there.

There are 4 types of joins,

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.

27. What do you mean by "Trigger" in SQL?

Trigger in SQL is a special type of stored procedure that is defined to execute automatically in place or after data modifications. It allows you to execute a batch of code when an insert, update, or any other query is executed against a specific table.

28. Are NULL values the 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. The NULL value represents a value that is unavailable, unknown, assigned, or not applicable whereas zero is a number and blank space is a character.

29. What is a 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 =.

There are two types of subqueries namely, Correlated and Non-Correlated.

Correlated subquery: These are queries that select the data from a table referenced in the outer query. It is not considered as an independent query as it refers to another table and refers to the column in a table.

Non-Correlated subquery: This query is an independent query where the output of the subquery is substituted in the main query.

30. What is the SELECT statement?

SELECT is a SQL command that is used for selecting data from a database. The data which is returned is saved in a result table, called the result-set.

Example: SELECT EMP_ID FROM EMPLOYEE;

Note: In the above SQL statement, 'EMP_ID' is a column and 'EMPLOYEE' is a Table.

31. What is a stored procedure?

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. Stored Procedure is a function that consists of many SQL statements to access the database system. Stored procedure supports faster execution instead of executing multiple queries. This reduces network traffic and provides better security to the data.

The disadvantage is that it can be executed only in the Database and uses more memory for storage.

32. What is CLAUSE in SQL?

SQL clause helps to limit the result set by providing a condition to the query. A clause helps to filter the rows from the entire set of records.

For example – WHERE, HAVING clause.

33. What is a View? What are Views used for?

A view is a virtual table that consists of a subset of data contained in a table. Since views are not present, it takes less space to store. The view can have data of one or more tables combined and it depends on the relationship. A view refers to a logical snapshot based on a table or another view.

It is used for the following reasons:

- Restricting access to data.
- Making complex queries simple.
- Ensuring data independence.
- Providing different views of the same data.

34. What is MS Access?

- MS Access was launched in 1992 by Microsoft Corporation as part of MS Office.
- Microsoft Access is an entry-level database management software. It is not only inexpensive but also a powerful database for small-scale projects.
- MS Access uses the Jet database engine which utilizes a specific SQL language dialect (sometimes referred to as Jet SQL).

- MS Access comes with the professional edition of MS Office package. MS Access is user-friendly database management system.

35. What is Oracle?

Oracle is a relational database management system developed by 'Oracle Corporation and launched in 1977. Oracle supports all major Operating systems including MS Windows. NetWare, UnixWare, OS/2, and most UNIX flavours.

36. What is MS SQL Server?

MS SQL Server is a Relational Database Management System developed by Microsoft Inc. Its primary query languages are T-SQL and ANSI SQL.

37. What is Sybase?

Sybase is a computer software company, their primary product is Sybase DBMS, which is a relational database management system based upon structured query language.

38. What is MySQL?

- MySQL is an open-source Database Management System, developed by Swedish company MySQL AB.
- MySQL Supports many different platforms including Microsoft Windows, the major Linux distributions, UNIX, and Mac OS X.
- MySQL has free and paid versions, depending on its usage (non-commercial/commercial) and features. MySQL comes with a very fast, multi-threaded, multi-user, and robust SQL database server.

39. What is DB2?

DB2 is the short name used for DATABASE 2. It is a relational database product developed by IBM. in 1983

40. What is DB/400?

It is one of the flavors of IBM DB2

41. What are the categories of operators available in SQL?

- Arithmetic operators
- Comparison operators
- Logical operators

42. What are Arithmetic operators in SQL?

Operator	Description
+ (Addition)	Adds values
– (Subtraction)	Subtracts Right side value from Left side value
* (Multiplication)	Multiplies values on either side of the operator
/ (Division)	Divides left hand operand by right hand operand
% (Modulus)	Divides left hand operand by right hand operand and returns remainder

43. What are Comparison operators in SQL?

For example x = 1, y= 2

Operator	Example
=	(x = y) is False
!=	(x != y) is True or (x <> y) is true.
>	(x > y) is False
<	(x < y) is True
>=	(x >= y) is False
<=	(x <= y) is True
!<	(x !< y) is False
!>	(x !> y) is True.

Note: Comparison Operators return Logical Results

44. What are Logical operators in SQL?

Operator	Description
NOT	Returns TRUE if the following condition is FALSE. Returns FALSE if it is TRUE.
AND	Returns TRUE if both component conditions are TRUE. Returns FALSE if either is FALSE.
OR	Returns TRUE if either component condition is TRUE. Returns FALSE if both are FALSE.

45. What is a Data Relationship and What are they?

Database Relationship is the connection between the tables in a database. There are 4 types of relationships, and they are:

- One to One Relationship
- One to Many Relationship
- Many to One Relationship
- Many to Many Relationship

46. What are Important Data Types in SQL?

Data Type	Syntax
character	char(x)
integer	integer
numeric	numeric(p,s)
decimal	decimal(p,s)
float	float(p)
date	date
time	time
character varying	varchar2(x)
bit	bit(x)
real	real
smallint	smallint

47. What are the Data Definition Language Commands and Operations?

Important Data Definition Language Commands

a. Create
b. Alter
c. Drop
d. Truncate
e. Rename

48. Important Data Definition Language Operations

a. Create a Database
b. Use Database
c. Rename a Database
d. Drop Database
e. Create a Table

f. Rename Table
g. Add a Column to exiting Table
h. Add multiple columns to existing Table
i. Modify an existing column
j. Rename a Column
k. Drop a Column
l. Truncate a Table
m. Drop a Table

Note: Download and Install Oracle 11g version (It is Free Edition) and practice SQL Commands and Operations.

49. How to Create a Database?

Syntax: Create Database databaseName;

Example: Create Database studentDB;

50. How to Select a Database?

Syntax: Use databaseName;

Example: Use studentDB;

51. How to Rename a Database?

Syntax: Alter Database databaseName Modify Name = newdatabseName;

Example: Alter Database studentDB Modify Name = Hyderabad Or Alter Database studentDB Modify Name = Hyderabad;

52. How to Drop a Database?

Syntax: Drop Database databaseName;

Example: Drop Database studentDB;

53. How to Create a Table?

Syntax: Create Table tableName(column1_name dataType(size),column2_name dataType(size),...);

Example: Create Table Students(STID int(10,STName char(50));

54. How to View Table info

Syntax: Select * from tablename;

Example: Select * from Students;

55. View Table Schema

Select * from INFORMATION_SCHEMA.COLUMNS where TABLE_NAME = 'Students';

56. How to Rename a Table?

Syntax: EXEC sp_rename 'old_tablename', 'new_tablename';

Example: EXEC sp_rename 'Students', 'newStudents';

57. How to Add a Column to an Existing Table?

Syntax: Alter Table table_name add column_name dataType(size);

Example: Alter Table newStudents add City char(50);

58. How to Add multiple columns to an Existing Table?

Syntax: Alter Table table_name add column1_name dataType(size), column2_name dataType(size); Or Alter Table table_name add column1_name dataType(size), column2_name dataType(size), ..;

Example: Alter Table newStudents add add1 char(100), add2 char(70); Or Alter Table newStudents add add3 char(100), add4 char(70), add5 char (100), phone int;

59. How to Modify an existing column?

Syntax: Alter Table table_name Alter Column column_name dataType(size);

Example: Alter Table newStudents Alter Column add1 varchar(150);

60. How to Rename a Column?

Syntax: EXEC sp_rename 'table_name.old_column_name', 'new_colum_name';

Example: ExEC sp_rename 'newStudents.phone', 'mobile'

61. How to Drop a Column?

Syntax: Alter Table table_name Drop Column column_name;

Example: Alter Table newStudents Drop Column City;

62. How to Truncate a Table?

Truncate Table command is used to delete complete data from an existing table

Syntax: Truncate Table table_name;

Example: Truncate Table newStudents;

63. How to Drop a Table?

Drop Table command is used to delete complete Table (Data and Table Structure) from the Database.

Syntax: Drop Table table_name;

Example: Drop Table newStudents;

64. How to add a new record into a Table?

Using INSERT INTO statement, we can insert new rows

Syntax: INSERT INTO TABLE_NAME (column1, column2, column3,…columnN) VALUES (value1, value2, value3,…valueN);

65. How to fetch data from a Database Table?

Using SELECT Statement, we can fetch data from a Database Table

Syntax: SELECT column1, column2, columnN FROM table_name; Or SELECT * FROM table_name;

66. Explain about IN Operator?

The IN operator implements comparison to a list of values, that is, it tests whether a value matches any value in a list of values. IN comparisons have the following general format:

value-1 [NOT] IN (value-2 [, value-3] …)

This comparison tests if value-1 matches value-2 or matches value-3, and so on. It is equivalent to the following logical predicate:

value-1 = value-2 [OR value-1 = value-3] …

67. Explain about FROM Clause in SQL?

The FROM clause always follows the SELECT clause. It lists the tables accessed by the query.

For example, SELECT * FROM S;

When the From List contains multiple tables, commas separate the table names. For example, SELECT sp.*, city FROM sp, s WHERE sp.sno=s.sno;

When the From List has multiple tables, they must be joined together.

68. What is the parameter substitution symbol used with the INSERT INTO command?

The parameter substitution symbol used with the INSERT INTO command is &.

69. What are the various uses of database triggers?

Database triggers can be used to enforce business rules, maintain derived values, and perform value-based auditing.

70. What is an event handler in SQL?

An event handler is a routine that is written to respond to a particular event.

71. What are the two methods of retrieving SQL?

The two methods of retrieving SQL are

1-Select

2-using Cursor.

72. What is a synonym? How is it used?

A synonym is used to reference a table or view by another name. The other name can then be written in the application code pointing to test tables in the development stage and to production entities when the code is migrated. The synonym is linked to the AUTHID that created it.

73. What is referential integrity?

Referential integrity refers to the consistency that must be maintained between primary and foreign keys, i.e. every foreign key value must have a corresponding primary key value.

74. Explain the EXPLAIN statement?

The 'explain' statement provides information about the optimizer's choice of access path of the SQL.

75. How is the SUBSTR keyword used in SQL?

SUBSTR is used for string manipulation with column name, first position and string length used as arguments. E.g. SUBSTR (NAME, 1 3) refers to the first three characters in the column NAME.

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

Group by controls the presentation of the rows, order by controls the presentation of the columns for the results of the SELECT statement.

77. What is a subselect? Is it different from a nested select?

A subselect is a select that works in conjunction with another select. A nested select is a kind of subselect where the inner select passes to the where criteria for the outer select.

78. What is the use of CASCADE CONSTRAINTS?

When this clause is used with the DROP command, a parent table can be dropped even when a child table exists.

79. How do you prevent output from coming to the screen?

The SET option TERMOUT controls output to the screen. Setting TERMOUT OFF turns off screen output. This option can be shortened to TERM.

80. Can the Primary key be a Foreign Key on the same table?

Yes, the Primary key is a Foreign Key on the same table.

81. How do you execute a host operating system command from within SQL?

By use of the exclamation point "!" (in UNIX and some other OS) or the HOST (HO) command.

82. What is a Cartesian product?

A Cartesian product is the result of an unrestricted join of two or more tables. The result set of a three-table Cartesian product will have x * y * z number of rows where x, y, z correspond to the number of rows in each table involved in the join.

83. How can variables be passed to a SQL routine?

By use of the & symbol. For passing in variables the numbers 1-8 can be used (&1, &2,…,&8) to pass the values after the command into the SQL PLUS session. To be prompted for a specific variable, place the ampersand variable in the code itself: "select * from dba_tables where owner=&owner_name;" . Use of double ampersands tells SQLPLUS to resubstitute the value for each subsequent use of the variable, a single ampersand will cause a repromoted for the value unless an ACCEPT statement is used to get the value from the user.

84. What command is used to get back the privileges offered by the GRANT command?

Revoke command is used to get back the privileges offered by the GRANT command.

85. What are the advantages of procedures?

- Loaded once and used many times.
- Performance better coz all SQL statements are sent in one go from the application to the database.
- Security (no object privileges are given directly).

- Invoker's rights are possible.
- Data integrity, productivity.

86. What is Parsing?

Parsing checks syntax checks privileges and allocating Private SQL Area.

87. What is Cursor?

Name or handle to a private SQL area where Oracle parses and fetches query results.

88. What is Datawarehouse?

Datawarehouse is a central repository of data from multiple sources of information. Those data are consolidated, transformed, and made available for mining and online processing. Warehouse data have a subset of data called Data Marts.

89. In what sequence SQL statements are processed?

The clauses of the select are processed in the following sequence

- FROM clause
- WHERE clause
- GROUP BY clause
- HAVING clause
- SELECT clause
- ORDER BY clause
- TOP clause

90. Write down the general syntax for a SELECT statement covering all the options.

Here's the basic syntax: (Also checkout SELECT in books online for advanced syntax)

- SELECT select_list
- [INTO new_table_]
- FROM table_source
- [WHERE search_condition]
- [GROUP BY group_by__expression]
- [HAVING search_condition]
- [ORDER BY order__expression [ASC | DESC]]

91. What is a correlated subquery?

When a subquery is tied to the outer query. Mostly used in self joins.

92. How to fetch common records from two tables?

Common records result set can be achieved by -.

Select studentID from student INTERSECT Select StudentID from Exam

93. How to select unique records from a table?

Select unique records from a table by using DISTINCT keyword.

Example: Select DISTINCT StudentID, StudentName from Student.

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

LIKE operator is used for pattern matching, and it can be used as -.

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

  Example:
- Select * from Student where studentname like 'a%';
- Select * from Student where studentname like 'ami_';

95. What are the types of subquery?

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

A correlated subquery cannot be considered as an independent query, but it can refer to the column in a table listed in the FROM the list of the main query.

96. What is SQL?

- SQL stands for Structured Query Language
- SQL was initially developed at IBM in the 1970s
- SQL is the standard language to communicate with relational database management systems, like Oracle, MS Access, MS SQL Server, MySQL, DB2, Sybase Etc…

Purpose of SQL

- SQL is used to Create New Databases
- SQL is used to Create New Tables in a Database
- SQL is used to Insert records in a Database
- SQL is used to Update records in a Database
- SQL is used to Delete records in a Database
- SQL is used to Retrieve data from a Database

- SQL is used to execute queries against a Database
- SQL can set permissions on tables, procedures and views
- SQL is used to Create stored procedures in a Database
- SQL is used to Create views in a Database

97. Who should learn SQL?

- Database Developers

  Design and deploy Database table structures, forms, reports and queries etc…
- Database Administrators (DBA)

  Keeping databases up to date and managing database access

  Writing Reports, documentation, and operating manuals
- Database Testers

  Verify Data Integrity

  Verify Data Manipulations (Add, Update, and Delete)

  Verify Data comparisons

98. What are the subsets of SQL?

SQL Commands can be classified into groups based on their nature, they are,

- Data Definition Language
- Data Manipulation Language
- Data Control Language

99. Important Commands and Operations in Data Definition Language [DDL].

Data Definition Language Commands are used to Create, Modify, and Drop the Structure of Database Objects like **Table, View, and Index**, etc…

- **Create**: To create databases and database objects

  Syntax: Create table < Table name> (col name 1 datatype (width), colname 2 datatype (width)…);

  Ex: SQL > create table student (Sno number (10), Sname varchar2 (20),.....);
- **Alter**: To modify existing database objects.

· **Alter table to add column:**

  Syntax: SQL> alter table < table name> add(colname datatype (width));

  Ex: SQL> alter table student add (group varchar (6));

- **To modify size (OR) column:**

  Syntax: SQL> alter table <table name> modify (column datatype (width));

  Ex: SQL> alter table student modify (sgroup varchar(10));

- **To Rename column:**

  Syntax: alter table tablename rename existing columname to new columname;

  Ex: alter table student rename column sid to StudentId;

- **Drop**: To drop databases and databases objects

  Syntax: ALTER TABLE tablename DROP column columname;

  Ex: SQL> ALTER TABLE student DROP column studentid;

- **Truncate**: To remove all records from a table

  Syntax: TRUNCATE TABLE table_name;

  Example: TRUNCATE TABLE EMPLOYEE

- **Rename:** To rename database objects(Table).

  Syntax: rename Old tablename to new tablename

  Example: SQL > rename student tostudent1;

100. Important DDL Operations.

- Create a Database
- Use Database
- Rename a Database
- Drop Database
- Create a Table
- Rename Table
- Add a Column to exiting Table
- Add multiple columns to existing Table
- Modify an existing column
- Rename a Column
- Drop a Column
- Truncate a Table
- Drop a Table

101. Important Commands and Operations in Data Manipulation Language [DML].

- **Select:** To select specific data from a database. To fetch records from the tables and views stored in the database, the Oracle SELECT statement is used.

· **To select all fields from a table:**

Syntax: SELECT * FROM table_name;

Example: SELECT * from students;

· **To select specific fields from a table:**

Syntax: SELECT field1,field2 FROM table_name;

Example: SELECT name, age FROM students

· **To select specific fields from a table using conditions:**

Syntax: SELECT field1,field2 FROM table_name WHERE conditions;

Example: SELECT name, age FROM students WHERE age>10

ORDER BY name;

· **To select specific fields from multiple tables:**

Syntax: SELECT expressions FROM table1 INNER JOIN table2 ON join_predicate;

Example: SELECT students.name, teachers.subject FROM students INNER JOIN teachers ON students.student_id = student_id ;

ORDER BY name;

- **Insert:** To insert new records in a table

· **Inserting values for all columns:**

Syntax: insert into tablename values(Column1value,Column2value,Co3umn1value,.....);

Ex: insert into employee values(1001,'krishna',50000,'05-may-2020');

· **Inserting values to specific columns:**

Syntax: insert into tablename(columnname,columname) values(columndata, cloumndata);

Ex: SQL> insert into employee (eid,ename) values(1003,'Supriya');

· **Inserting values for all columns at a time:**

Syntax:insert into tablename values(&column1name,&column2name. ...);

Ex: SQL> insert into employee values(&eid,'&ename',&salery,'&doj');

- **Update**: To update existing records.

Syntax: SQL>update < table name> set colname -values..... where < condition>;

Ex: SQL> update emp set da = sal * 10/100;

SQL> update emp set da = sal * 20/100 where ename = 'smith';

- Delete: To delete existing records from a table.

· **Delete the all rows from a table:**

Syntax: SQL>delete from <table name> where <condition>;

Ex: SQL> delete from student;

· **Delete particular row from a table:**

Ex: SQL> delete from student where Sno=101;

102. Important Commands and Operations in Data Control Language [DCL].

- Grant: To provide access to the Database objects to the users

  Syntax: SQL> grant privileges on <table name> to user name;

  Ex: SQL> grant select on emp to krishdb;

- Revoke: to remove user access rights to the database objects

  Syntax: SQL> revoke privileges on <table name> from user name;

  Ex: SQL> revoke select on emp from krishdb;

103. Important Commands and Operations in Data Control Language [TCL].

This language is used to transaction process, it consists of commit & Roll back commands.

- **Commit command:** It is used to make database changes permanently.

  Syntax: SQL> commit;

- **Roll back command:** It is used to cancel the committed transactions.

  Syntax: SQL> roll back;

104. Popular Database Management Systems.

- Oracle
- Microsoft SQL Server
- MySQL
- PostgreSQL
- MS Access

104. SQL Syntax.

- A database contains one or more tables. Each table is identified by a name, Tables contain records (rows) with data.

- Most of the actions we need to perform on a database are done with SQL statements.
- SQL keywords are NOT case sensitive: select is the same as SELECT
- All the SQL statements start with any of the keywords like SELECT, INSERT, UPDATE, DELETE etc…and all the statements end with a semicolon (;), the semicolon is the standard way to separate SQL Statements.

105. SQL Data Types.

A data type defines what kind of value a column can contain, we have to use data types while creating database tables, choose a particular data type for a table column based on our requirement.

Example:

- Character Data Types
- Numeric Data Types
- Date and Time Data Types etc…

Note: Data Types vary from One Database Management System to another

106. SQL Operators

Operators are used to perform Arithmetic, Comparison and Logical Operations.

- Arithmetic Operators
- Comparison Operators
- Logical Operators

107. SQL Functions

SQL has many built-in functions for performing processing on data.

- Aggregate Functions
- String Functions
- Date Functions Etc…

108. SQL Joins

The SQL Joins clause is used to combine records from two or more tables in a database.

Different SQL Joins:

- INNER JOIN (EQUI JOIN and NON-EQUIJOIN)
- Left Join
- Right Join

- Full Join
- Cross Join
- Self-Join

Popular Database Management Systems:

109. Oracle:

Oracle database is a relational database management system (RDBMS) developed by Oracle Corporation. Important Oracle editions are,

- Enterprise Edition: Offers all features, including superior performance and security.
- Standard Edition: Contains base functionality for users.
- Express Edition: The lightweight, free and limited Windows and Linux edition
- Oracle Lite: For mobile devices

110. Microsoft SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. Important Microsoft SQL Server editions are,

- Datacenter: SQL Server 2008 R2 Datacenter is a full-featured edition of SQL Server.
- Enterprise: SQL Server Enterprise Edition includes both the core database engine and add-on services.
- Standard: SQL Server Standard edition includes the core database engine, along with the stand-alone services.
- Web: SQL Server Web Edition is for Web hosting.
- Workgroup: SQL Server Workgroup Edition includes the core database functionality only.

111. MySQL

MySQL is an Open Source Relational SQL database management system used for developing web-based software applications. Important MySQL Editions are,

- Standard Edition: Standard Edition enables us to deliver high-performance and scalable Online Transaction Processing (OLTP) applications.
- Enterprise Edition: Enterprise Edition includes the most comprehensive set of advanced features and management tools.

- Cluster Carrier Grade Edition: Cluster enables users to meet the database challenges of next generation web, cloud, and communications services with uncompromising scalability, uptime and agility.

112. PostgreSQL

PostgreSQL is a powerful, open-source database management system. It runs on all major operating systems, including Linux, UNIX, Mac OS X, Solaris, and MS Windows.

113. MS Access

Microsoft Access is bundled as part of the Microsoft Office suite. It is only available on the PC version. It is a desktop database system because its functions are intended to be run from a single computer

114. What is Data?

In computing, data is information that has been translated into a form that is efficient for movement or processing. Data is defined as facts or figures, or information that's stored in or used by a computer.

115. What is a Database?

- A Database is a systematic collection of data.
- Databases support storage and manipulation of data.
- Databases make data management easy.

116. Database Objects in Relational databases

A database object in a relational database is a data structure used to either store or reference data. Most of the major database engines offer the same set of major database object types:

- Tables
- Indexes
- Sequences
- Views
- Synonyms

117. Database Server

- Database Server instance contains multiple databases
- Database contains multiple Tables
- Table contains multiple records (rows and columns)

118. Table and Record

- A Table in a Relational Database is a predefined format of rows and columns that define an entity.
- Each column contains a different type of attribute and each row corresponds to a single record.
- Each Table is provided with a name.

Example: Table name: Students

SID	Name	Address	Contact-No
1001	Venkat	Abcd, xyz	9876989890
1002	Ramya	Asdf, ghj	8888887676
1003	Basha	Ert,	yuio

119. What is DBMS?

- A Database management System is software designed to assist in maintaining and utilizing a large collection of data.
- The alternative to using a DBMS is to store the data in files and write application-specific code to manage it.

  Using a DBMS to manage data has many advantages:
- Data Independence
- Efficient Data Access
- Data Integrity and security
- Data Administration
- Concurrent Access and Data Recovery etc…

120. Types of Database Management System

Types of Database Management System

- Hierarchical DBMS
- Network DBMS
- Relational Database Management System (RDBMS)
- Example: Oracle, MS SQL Server, MySQL, DB2, etc…

- Non-Relational Database Management System
- Example: MongoDB, Apache Cassandra, Redis, Couchbase, and Apache HBase etc,

121. What are the types of databases?

- SQL/Relational databases
- NoSQL/Non-Relational databases
- Cloud databases
- Object-oriented databases
- Centralised database.
- Distributed database.
- Personal database.
- Commercial database.

122. What is Big Data?

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is data with so large size and complexity that none of the traditional data management tools can store it or process it efficiently. Big data is also data but with huge size.

Types of Big Data:

- Structured data
- Unstructured data
- Semi-structured data

123. SQL Language Elements

- **Identifiers**: Names of Database Objects such as Tables, Views, Columns etc…
- **Data Types**: Define the type of data that is contained by a Column
- **Constants**: Symbols that represent specific data types
- **Operators**: Perform Arithmetic, Comparison, and Logical Operations
- **Functions**: Built-in Functions to perform specific operations
- **Clauses**: Constituent components of statements and queries.
- **Expressions**: Produce scalar values, or tables containing of columns and rows of data.
- **Queries**: Retrieve the data based on specific criteria, this an important element of SQL.
- **Statements** etc…

Data Definition Language Queries:

124. Create a Database

Syntax: Create Database databaseName;

Example: Create Database practiceDB;

125. Use Database

Syntax: Use databaseName;

Example: Use practiceDB;

126. Rename a Database

Syntax: Alter Database databaseName Modify Name = newdatabseName;

Example: Alter Database practiceDB Modify Name = mydb;

127. Drop a Database

Syntax: Drop Database databaseName;

Example: Drop Database gcreddyDB;

128. Create a Table

Syntax: Create Table tableName(column1_name dataType(size),column2_name dataType(size),...);

Example: Create Table Students(STID int(5),STName char(50));

129. View Table info

Syntax: Select * from tablename;

Example: Select * from Students

130. View Table Schema

Syntax: desc tablename;

Example: desc Student;

131. Rename Table

Syntax: ALTER TABLE old_tablename RENAME TO new_tablename;

Example: ALTER TABLE Students RENAME TO newStudents;

132. Add a Column to the existing Table

Syntax: alter Table table_name add (column_name dataType(size));

Example: alter Table newStudents add (City char(50));

133. Add multiple columns to an existing Table

Syntax: Alter Table table_name add column1_name dataType(size), column2_name dataType(size);

Example: alter Table newStudents add add1 char(100), add2 char(70);

134. Modify an existing column

Syntax: alter table <table name> modify (column datatype (width));

Example: alter table student modify (sgroup varchar(10));

135. Rename a Column

Syntax: alter table tablename rename existing columname to new columname;

Example: alter table student rename column sid to StudentId;

136. Drop a Column

Syntax: Alter Table table_name Drop Column column_name;

Example: Alter Table newStudents Drop Column City;

137. Truncate a Table:

Truncate Table command is used to delete complete data from an existing table.

Syntax: Truncate Table table_name;

Example: Truncate Table newstudents;

138. Rename a Table:

It is a DDL Command, which is used to change the name of the data base table.

Syntax: rename Old tablename to new tablename

Example: SQL > rename student tostudent1;

139. Drop a Table:

Drop Table command is used to delete the complete Table (Data and Table Structure) from the Database.

Syntax: Drop Table table_name;

Example: Drop Table newStudents;

i. **Data Manipulation Language Queries:** Data Control Language etc…

Data Manipulation Language commands are used to store, modify, retrieve, and delete data from database tables. In this category we have Select, Insert, Update, Delete Commands…

140. Inserting values for all columns:

Syntax: insert into tablename alues(Column1value,Column2value,Co3umn1value,. .....);

Example: insert into employee values(1001,'krishna',50000,'05-may-2020');

o/p: 1 row created.

141. Inserting values to specific columns:

Syntax: insert into tablename(columnname,columname) values(columndata, cloumndata);

Example: SQL> insert into employee (eid,ename) values(1003,'Supriya');

o/p: 1 row created.

Note: The other columns are assigned with "Null values".

142. Inserting values for all columns at a time:

Syntax: insert into tablename values(&column1name,&column2name. ...);

Column1value:

Column2value:

Example: SQL> insert into employee values(&eid,'&ename',&salery,'&doj');

Enter value for eid: 1005

Enter value for ename: archana

Enter value for salery: 61454

Enter value for doj: 25-jan-1999

o/p: 1 row created.

143. Update command:

Syntax: SQL>update < table name> set colname -values ..... where < condition>;

Example: SQL> update emp set da = sal * 10/100;

SQL> update emp set da = sal * 20/100 where ename = 'smith';

144. Delete all rows from a table command:

Syntax: SQL>delete from <table name> ;

Example: SQL> delete from student;

145. Delete specific rows from a table command:

Syntax: SQL>delete from <table name> where <condition>;

SQL> delete from student where Sno=101;

NOTE: Delete command removes all rows but the structure of a table remains unchanged.

146. To select all fields from a table:

Syntax: SELECT * FROM table_name;

Example: SELECT * from students;

147. To select specific fields from a table:

Syntax: SELECT field1,field2 FROM table_name;

Example: SELECT name, age FROM students

148. To select specific fields from a table using conditions:

Syntax: SELECT field1,field2 FROM table_name WHERE conditions;

Example: SELECT name, age FROM students WHERE age>10 ORDER BY name;

149. To select specific fields from multiple tables:

Syntax: SELECT expressions FROM table1 INNER JOIN table2 ON join_predicate;

Example: SELECT students.name, teachers.subject FROM students INNER JOIN teachers ON students.student_id = student_id ;ORDER BY name;

150. What is the difference between a left join and a right join?

A left join returns all rows from the left table and matching rows from the right table, while a right join returns all rows from the right table and matching rows from the left table.

151. What is a correlated subquery and how is it used?

A correlated subquery is a subquery that references a column from the outer query. It is used to retrieve data based on the values in the outer query, such as finding the most recent order for each customer.

152. What is a database index and how is it used?

A database index is a data structure used to speed up data retrieval by allowing the database engine to quickly locate specific data based on the values in one or more columns. It is used to improve query performance and reduce data access time.

153. What is a view and how is it used?

A view is a virtual table that is derived from one or more tables or other views in a database. It is used to simplify complex queries, limit data access, and provide a level of abstraction between the user and the underlying data.

154. What is a database trigger and how is it used?

A database trigger is a special type of stored procedure that automatically executes in response to certain events, such as inserting, updating, or deleting data in a table. It is used to enforce business rules, audit changes, or automate complex tasks.

155. What is normalization and why is it important in database design?

Normalization is the process of organizing data in a database to minimize redundancy and improve data integrity. It is important in database design because it helps to reduce data anomalies, improve data consistency, and simplify data maintenance.

156. What is a full outer join and how is it used?

A full outer join returns all rows from both tables being joined, along with matching rows from each table where available. It is used to combine data from two tables where not all rows have a matching value.

157. What is a constraint and how is it used?

A constraint is a rule that limits the type or value of data that can be inserted, updated, or deleted in a table. It is used to enforce data integrity and prevent data inconsistencies.

158. What is the difference between a clustered and a non-clustered index?

A clustered index determines the physical order of data in a table based on the values of the indexed columns, while a non-clustered index contains a separate data structure that maps the indexed columns to the corresponding rows in the table.

159. What is a materialized view and how is it used?

A materialized view is a precomputed result set that is stored on disk for fast data retrieval. It is used to speed up complex queries and reduce response time for data analysis.

160. What is a data warehouse and how is it different from a traditional database?

A data warehouse is a centralized repository that stores large amounts of historical and current data for analysis and decision-making. It is optimized for read-intensive workloads and data analysis, while a traditional database is optimized for transactional processing.

161. What is the difference between a group by and a having clause?

A group by clause is used to group the results of a query by one or more columns, while a having clause is used to filter the results of a group by based on some condition.

162. What is a foreign key and how is it used?

A foreign key is a column or combination of columns that references a primary key in another table. It is used to establish a relationship between two tables, ensuring data integrity and enforcing referential integrity constraints.

163. What is a self-join and how is it used?

A self-join is a join where a table is joined with itself using a common column. It is used to retrieve related data from the same table, such as finding all employees who have the same manager.

164. What is a primary key in SQL?

A primary key is a column or set of columns in a table that uniquely identifies each row in the table. Primary keys are used to establish relationships between tables and enforce data integrity.

165. What is the difference between a DELETE and TRUNCATE statement in SQL?

A DELETE statement is used to delete specific rows from a table, while a TRUNCATE statement is used to delete all data from a table, effectively resetting the table to its initial state.

166. How do you select distinct values from a column?

Example: SELECT DISTINCT column_name FROM table_name;

167. How do you sort a query by a specific column?

Example: SELECT column1, column2 FROM table_name ORDER BY column1;

168. How do you limit the number of rows returned in a query?

Example: SELECT column1, column2 FROM table_name LIMIT 10;

169. How do you join multiple tables in a query?

Example: SELECT * FROM table1 JOIN table2 ON table1.column_name = table2.column_name;

170. How do you create a primary key for a table?

Example: CREATE TABLE table_name (id INT PRIMARY KEY, column1 VARCHAR(255), column2 VARCHAR(255));

171. How do you create a foreign key for a table?

Example: CREATE TABLE table2 (id INT, FOREIGN KEY (id) REFERENCES table1(id));

172. How do you create an index in a table?

Example: CREATE INDEX index_name ON table_name (column_name);

173. How can you use the GROUP BY clause in SQL?

Example: SELECT column1, aggregate_function(column2) FROM table_name GROUP BY column

174. How can you sort the results of a query in SQL?

Example: SELECT column1, column2, ...FROM table_name ORDER BY column1 [ASC|DESC], column2 [ASC|DESC], ...;

175. How can you use the BETWEEN operator in SQL?

Example: SELECT column1, column2, ... FROM table_name WHERE column_name BETWEEN value1 AND value2;

176. How can you use the LIKE operator in SQL?

Example: SELECT column1, column2, ... FROM table_name WHERE column_name LIKE 'pattern';

177. How can you use the LIMIT clause in SQL?

Example: SELECT column1, column2, ...FROM table_name LIMIT number_of_rows;

178. How can you use the NOT IN operator in SQL?

Example: SELECT column1, column2, ... FROM table_name WHERE column_name NOT IN (value1, value2, ...);

179. How can you use the IS NULL operator in SQL?

Example: SELECT column1, column2, ...FROM table_name WHERE column_name IS NULL;

180. How can you use the UNION operator in SQL?

Example: SELECT column1, column2, ...FROM table1 UNION SELECT column1, column2, ... FROM table2;

181. How can you use the AVG() function in SQL?

Example: SELECT AVG(column_name) FROM table_name;

182. What is Data Integrity?

Data integrity is referred to the data's precision and consistency (validity) over its lifecycle. It ensures an organization's data's accuracy, completeness, consistency, and validity.

Every time data is copied or moved, it should remain unchanged and unaltered between updates. Error checking methods and validation procedures are typically responsible for ensuring the integrity of data transmitted or reproduced without the purpose of alteration.

183. What are ACID Properties?

Atomicity: means that the transaction takes place at once or doesn't happen at all;

Consistency: means that integrity constraints must be maintained so that the database is consistent before and after the transaction;

Isolation: ensures that each transaction is independent and occurs without interference;

Durability: ensures that after a transaction has completed execution, the updates and modifications to the database are stored in and written to disk, and they persist even if a system failure occurs.

184. Difference between CHAR and VARCHAR2

The CHAR data type stores character values. It is a fixed-length data type and has a maximum length of 2000 bytes of characters. Because char is a fixed size, it will often lead to memory wastage. (If you have a char(10), and want to store "Alex" in it will result in "Alex"). It has 6 additional blank characters.

The VARCHAR2 also stores character values, but it has variable length and can have a maximum length of 4000 bytes of characters. It helps in saving memory bytes.

185. SQL COUNT function

Example: SELECT COUNT(NAME) FROM employee WHERE manager_id = 1;

186. Different Types of Relationships?

One-to-one: occurs when each row in the user table has only one corresponding row in the password table.

One-to-many: one-to-many relationship occurs when one record in the class table is related to one or more records in the student table. However, one record in the student table cannot be related to more than one record in the class table.

Many-to-many: multiple records in one table (books table) are related to various records in another table (authors table).

187. What is a Live Lock?

Locks are essential parts of databases. If our application scales, it will receive many requests, which may result in a situation where two or more sessions will try to edit the same row. That may result in some data integrity problems.

A Live lock happens when a request for an exclusive lock is denied continuously because a sequence of other shared locks keeps interfering with each other and changing the status. Therefore no one will complete the transaction.

An excellent example of a live lock would be two cars meeting on a one-way street, both of them will turn left or right at the same time to let the other car pass, but as these actions are done at the same time, neither of them will advance.

188. SQL Query to Find Names Starting with a Substring?

select name from employee where name like 'a%'

189. SQL Query to Find and Delete Duplicate Records from a Table?

Example: DELETE employee1 FROM employee_2 employee1 INNER JOIN employee_2 employee2 WHERE employee1.id < employee2.id AND employee1.email = employee2.email;

190. How to Remove Duplicate Rows from a Query?

SELECT DISTINCT fetches all the distinct data from a query. For example, let's say that we want all the distinct countries where our employees are located. For this task, we will use the following SQL query:

Example: SELECT DISTINCT country FROM employee

191. How to Use LIKE operator in SQL?

The LIKE operator is used in WHERE clauses to check if a value fits a given string pattern. Here is an example of a LIKE operator:

Example: SELECT * FROM employe WHERE name like "Alexandru";

With this command, we can pull all the records where the first name is like "Alexandru". The result will look like this:

192. What is a Schema in SQL?

- Our database holds a lot of different objects, such as tables, relations, triggers, indexes, their relationships and so on. So, we usually make a database schema to create a clear vision of our database and organize all tables and relations between them.
- So, we can view schema as the logical relationship between all the different tables in the database. Overall, we can consider a schema as a blueprint for the database. Usually, the schema is the first thing created when creating a database.

193. What are UNION, MINUS and INTERACT commands?

- The **UNION** operator merges the results of two tables and eliminates duplicate rows from the tables.
- The **MINUS** returns the rows from the first query but not from the second query.
- The **INTERSECT** returns the rows that are returned by both of the queries.

194. Differences between Views and Tables?

Views and tables may seem pretty similar, but they have some differences. Both of them consist of rows and columns, but they differ in terms of persistence.

A **table** stores and retrieves data from persistent storage whenever the user wants. On the other side, the view is only a virtual table based on the result of some SQL queries, meaning that this will disappear after the session ends.

195. Difference between ORDER BY and GROUP BY clauses?

Using **ORDER BY**, we can sort the result set in ascending or descending order.

GROUP BY clause is used to group the rows with the same value.

196. Difference between WHERE and HAVING?

With the **WHERE** clause, we can filter the result set according to given conditions. It can be used with SELECT, UPDATE, and DELETE statements.

Example: SELECT ID, NAME FROM EMPLOYEE WHERE ID >= 100

HAVING clause is used as a filter with the GROUP BY statement. Those grouped rows that will satisfy the given condition in HAVING will appear in the final result. HAVING Clause can only be used with SELECT statement.

Example: SELECT ID, NAME FROM EMPLOYEE GROUP BY NAME HAVING AGE > 18;

197. What are constraints? Different types of Constraints?

Constraints are the rules imposed on the database contents and processes to ensure data integrity. There are different types of constraints:

Domain constraint specifies the attribute's domain or set of values (if we define a number column, we cannot insert 'A' in the column).

Tuple Uniqueness constraint specifies that all the rows must be necessarily unique in any table.

Key constraint defines that every primary key should be unique and not null.

Entity Integrity constraint specifies that no attribute that composes the primary key must contain a null value in any relation.

Referential Integrity constraint establishes that all the foreign key values must either be in the relation to the primary key or be null.

198. What is a View in a Database?

A view is a virtual table in the database that contains data from one or more tables, but it takes less memory because it is not physically stored in the database. The names of the views are always unique. Database views are saved in the database as named queries and can be used to save frequently used complex queries. Views are an excellent tool to restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.

Example: CREATE VIEW EMPLOYEE_VIEW AS SELECT ID, NAME, DOB FROM EMPLOYEE;

199. Difference between SQL and PL/SQL?

→ Even though SQL and PL/SQL (Procedural Language extension to SQL) are pretty similar, there are several differences in how they work, they differ in performance, error handling capabilities, and interaction with the databases. For example:

→ SQL is a query language with few keywords. And PL/SQL is a programming language using SQL for a database and has variables, data types, loops etc. PL/SQL also offers error and exception handling features that do not exist in SQL.

→ SQL is declarative; PL/SQL is procedural.

→ While SQL executes one query at a time, PL/SQL can run an entire block or multiple operations in a single execution.

→ PL/SQL does not interact directly with the database server, while SQL does.

→ Generally, SQL is used for DDL and DML statements, whereas PL/SQL is used to write program blocks, functions, procedures, triggers and packages.

200. Difference between Relational and Non-relational Databases

A **relational database** is organized in tables. Usually, the data within these tables have relationships with one another or dependencies.

A **non-relational database** is document-oriented, meaning all information gets stored in more of a laundry list order. As a result, we will have all our data listed within a single construct or document.

201. What is a Query?

A query, in SQL, is a command used to request data or update information from a database table or combination of tables. Users can write and execute queries to retrieve, delete or update information in a database. Usually, a query response returns data from different tables within the database, but this does not apply every time (for example, a delete query will not return anything).

For example, let's say that we have an employee table. We want to get all employees from the table that are born before 1990. For this case, we will use a SELECT statement because all we want is to retrieve data from a table.

Example: SELECT * FROM employees WHERE year_born < 1990

202. What is a Primary Key?

- A primary key refers to a column or a set of columns containing values uniquely identifying each row in a table. A database table uses a primary key to uniquely insert, update, restore or delete data from a database table.
- Generally, a primary key is a sequentially generated long number and its whole purpose is to uniquely identify a particular row in the table. It does not have any business value in itself.
- We can define a primary key in the employee table as follows. In this example, we are marking ID column as the primary key for the table Employee.

Example: CREATE TABLE Employee (id INT PRIMARY KEY, name VARCHAR(45),email VARCHAR(45),manager_id INT,country VARCHAR(45));

203. What is a Unique Key?

A unique key refers to a value of one column (or combination of more than one column) that uniquely identifies a record in the database table. not like primary key, a unique key has a business meaning and puts a certain constraint on the table as defined by the business.

For example, for an EMPLOYEE table, am EMAIL can be a unique key because each employee is guaranteed to have a unique key throughout the organization.

Example: CREATE TABLE Employee (id INT PRIMARY KEY,name VARCHAR(45), email VARCHAR(45) UNIQUE, manager_id INT, country VARCHAR(45));

If the table and the column already exist, we can add the constraint in the following way:

Example: ALTER TABLE Employee ADD UNIQUE (email);

204. Difference between Primary key and Unique Key?

- A primary key column cannot contain NULL values, but the unique key can accept one NULL value.
- Every database table can have just one primary key, but multiple unique keys.
- The primary key generates a clustered index, but the unique key generates a non-clustered index.

205. What is a Foreign Key?

A FOREIGN KEY is a column (or collection of columns) in one table that links the PRIMARY KEY in another table.

We can add a foreign key to our employee table on the manager_id column because a manager is also an employee. We will link the manager_id column with the id from the same table. If we are creating a new table we can implement the foreign key in the following way:

Example: CREATE TABLE Employee (id INT PRIMARY KEY, name VARCHAR(45), email VARCHAR(45) UNIQUE, manager_id INT, country VARCHAR(45), FOREIGN KEY (manager_id) REFERENCES Employee(id));

And if we already have the table and the column created we can add the constraint altering the table in the following way:

Example: ALTER TABLE employee ADD FOREIGN KEY (manager_id) REFERENCES Employee(id);

206. What are Database Normalization and Denormalization?

Database normalization is an essential process that consists of structuring a relational database by a sequence of so-called normal forms to ease redundancy and improve data integrity. To say that the data from our database is normalized, it should meet the following two requirements:

→ No redundancy of data. The data is held in just one place and is not duplicated in other tables.

→ Data dependencies are logical and separated. All related data are stored together.

→ There are a few rules for database normalization. Each rule is called a "normal form." If the first rule is observed, the database is said to be in "first normal form." If the first three rules are observed, the database is considered to be in "third normal form." The third normal form is considered the highest level necessary for most applications.

Here is a **list** of Normal Forms in SQL:

1NF (First Normal Form): Each table cell should contain a single value. Each record needs to be unique.

2NF (Second Normal Form): In addition to 1NF, all non-key attributes are fully dependent on the primary key. It helps to eliminate redundant data.

3NF (Third Normal Form): In addition to 2NF, there are no transitive functional dependencies. In 3NF, values in a record that are not part of that record's key do not belong in the table.

BCNF (Boyce-Codd Normal Form): is just a more strong form of 3NF. Sometimes also referred to as 3.5NF.

4NF (Fourth Normal Form): In addition to BCNF, the database has no multi-valued dependency.

5NF (Fifth Normal Form) In addition to 4NF, the database does not contain any join dependency, joining should be lossless.

On the opposite, **denormalization** is the exact opposite process of normalization. Here, we intentionally add redundancy to the data to improve the specific application's performance and protect data integrity.

Although denormalization is not recommended, it may be necessary in some very specific cases.

207. Arithmetic Operators:

Arithmetic operators can perform arithmetical operations on numeric operands involved.

Operator	Description	Example
+ Addition	Adds values on either side of the operator	a + b will give 30
- Subtraction	Subtracts right hand operand from left hand operand	a - b will give -10
* Multiplication	Multiplies values on either side of the operator	a * b will give 200
/ Division	Divides left hand operand by right hand operand	b / a will give 2
% Modulus	Divides left hand operand by right hand operand and returns remainder	b % a will

208. Comparison Operators:

A comparison (or relational) operator is a mathematical symbol that is used to compare between two values. Comparison operators are used in conditions that compare one expression with another. The result of a comparison can be TRUE, FALSE, or UNKNOWN (an operator that has one or two NULL expressions returns UNKNOWN).

Operator	Description	Example
=	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(a = b) is not true. (10=20)
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(a != b) is true. (10!=20)
<>	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(a <> b) is true. (10<>20)
>	Checks if the value of left operand is greater than the value of right operand, if yes then	(a > b) is not true. (10>20)
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(a < b) is true. (10<20)

>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(a >= b) is not true. (10>=20)
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(a <= b) is true. (10<=20)
!<	Checks if the value of left operand is not less than the value of right operand, if yes then condition becomes true.	(a !< b) is false. (10!<20)
!>	Checks if the value of left operand is not greater than the value of right operand, if yes then condition becomes true.	(a !> b) is true. (10!>20)

209. **Logical Operators:**

There are three Logical Operators namely, AND, OR, and NOT. These operators compare two conditions at a time to determine whether a row can be selected for the output. When retrieving data using a SELECT statement, You can use logical operators in the WHERE clause, which allows you to combine more than one condition.

OPERATOR	DESCRIPTION
ALL	The ALL operator is used to compare a value to all values in another value set.
AND	The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause.
ANY	The ANY operator is used to compare a value to any applicable value in the list according to the condition.
BETWEEN	The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value.
EXISTS	The EXISTS operator is used to search for the presence of a row in a specified table that meets certain criteria.
IN	The IN operator is used to compare a value to a list of literal values that have been specified.
LIKE	The LIKE operator is used to compare a value to similar values using wildcard operators.
NOT	The NOT operator reverses the meaning of the logical operator with which it is used. [Eg: NOT EXISTS, NOT BETWEEN, NOT IN, etc. This is a negate operator.]
OR	The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause.
IS NULL	The NULL operator is used to compare a value with a NULL value.
UNIQUE	The UNIQUE operator searches every row of a specified table for uniqueness (no duplicates).

210. 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.

Syntax for column: Column_name AS alias_name;

Here, column_name: original name of the column and alias_name: temporary name

SQL> SELECT C.ID AS CUSTID, C.NAME AS CUSTNAME, O.OID AS ORDERID

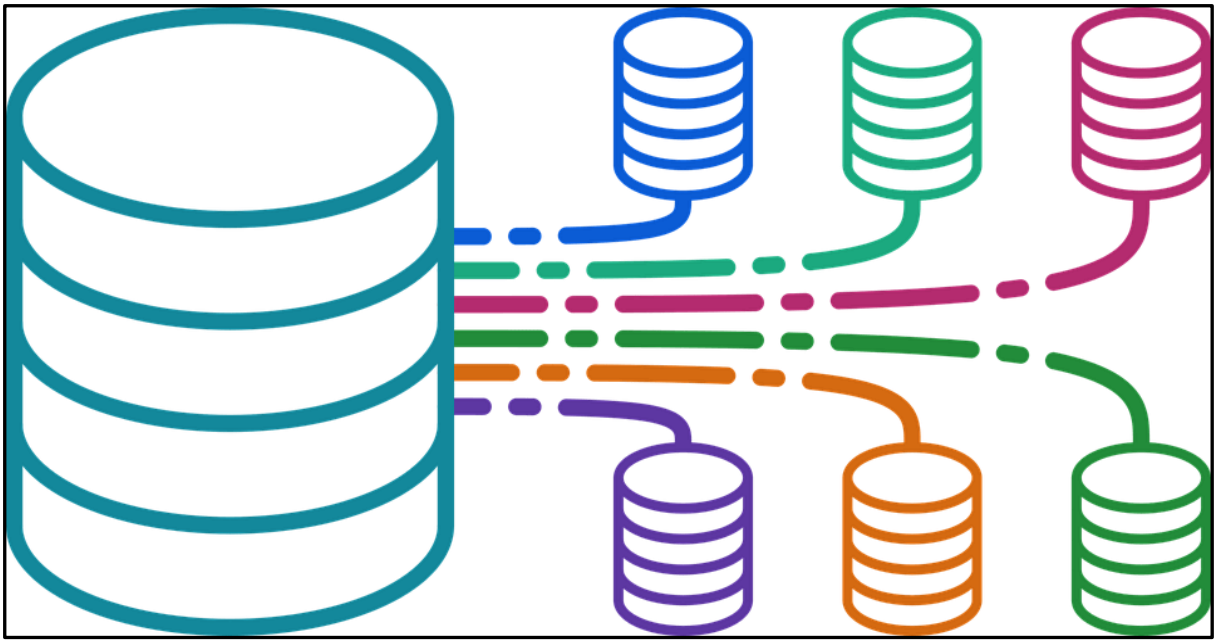
FROM CUSTOMERS C JOIN ORDERS O ON C.ID = O.ID;

Syntax for table:

CREATE TABLE new_table_name AS SELECT * FROM existing_table_name;

SQL> CREATE TABLE orderNew AS SELECT * FROM orders;

Table created.



******THE END********