

**1. What is Database?**

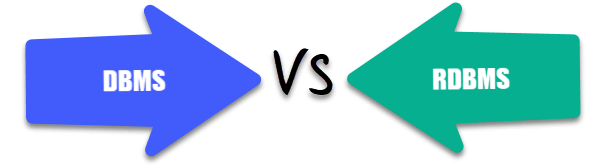
A database is an organized collection of data, stored and retrieved digitally from a remote or local computer system. Databases can be vast and complex, and such databases are developed using fixed design and modeling approaches.

**2. What is DBMS?**

DBMS stands for Database Management System. DBMS is a system software responsible for the creation, retrieval, updation and management of the database. It ensures that our data is consistent, organized and is easily accessible by serving as an interface between the database and its end users or application softwares.

**3. What is RDBMS? How is it different from DBMS?**

RDBMS stands for Relational Database Management System. The key difference here, compared to DBMS, is that RDBMS stores data in the form of a collection of tables and relations can be defined between the common fields of these tables. Most modern database management systems like MySQL, Microsoft SQL Server, Oracle, IBM DB2 and Amazon Redshift are based on RDBMS.

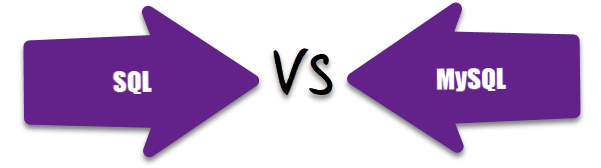


**4. What is SQL?**

SQL stands for Structured Query Language. It is the standard language for relational database management systems. It is especially useful in handling organized data comprised of entities (variables) and relations between different entities of the data.

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

SQL is a standard language for retrieving and manipulating structured databases. On the contrary, MySQL is a relational database management system, like SQL Server, Oracle or IBM DB2, that is used to manage SQL databases.



**6. What are Tables and Fields?**

A table is an organized collection of data stored in the form of rows and columns. Columns can be categorized as vertical and rows as horizontal. The columns in a table are called fields while the rows can be referred to as records.

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

Constraints are used to specify the rules concerning data in the table. It can be applied for single or multiple fields in an SQL table during creation of table or after creationg using the ALTER TABLE command. The constraints are:

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

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

The PRIMARY KEY constraint uniquely identifies each row in a table. It must contain UNIQUE values and has an implicit NOT NULL constraint.  
A table in SQL is strictly restricted to have one and only one primary key, which is comprised of single or multiple fields (columns).

**CREATE** **TABLE** Students ( */\* Create table with a single field as primary key \*/*

ID INT **NOT** **NULL**

Name VARCHAR(255)

**PRIMARY** **KEY** (ID)

);

**CREATE** **TABLE** Students ( */\* Create table with multiple fields as primary key \*/*

ID INT **NOT** **NULL**

LastName VARCHAR(255)

FirstName VARCHAR(255) **NOT** **NULL**,

**CONSTRAINT** PK\_Student

**PRIMARY** **KEY** (ID, FirstName)

);

**ALTER** **TABLE** Students */\* Set a column as primary key \*/*

**ADD** **PRIMARY** **KEY** (ID);

**ALTER** **TABLE** Students */\* Set multiple columns as primary key \*/*

**ADD** **CONSTRAINT** PK\_Student */\*Naming a Primary Key\*/*

**PRIMARY** **KEY** (ID, FirstName);

Q   =>   Write a SQL statement to add PRIMARY KEY 't\_id' to the table 'teachers'.



Q   =>   Write a SQL statement to add primary key constraint 'pk\_a' for table 'table\_a' and fields 'col\_b, col\_c'.



**9. What is a UNIQUE constraint?**

A UNIQUE constraint ensures that all values in a column are different. This provides uniqueness for the column(s) and helps identify each row uniquely. Unlike primary key, there can be multiple unique constraints defined per table. The code syntax for UNIQUE is quite similar to that of PRIMARY KEY and can be used interchangeably.

**CREATE** **TABLE** Students ( */\* Create table with a single field as unique \*/*

ID INT **NOT** **NULL** **UNIQUE**

Name VARCHAR(255)

);

**CREATE** **TABLE** Students ( */\* Create table with multiple fields as unique \*/*

ID INT **NOT** **NULL**

LastName VARCHAR(255)

FirstName VARCHAR(255) **NOT** **NULL**

**CONSTRAINT** PK\_Student

**UNIQUE** (ID, FirstName)

);

**ALTER** **TABLE** Students */\* Set a column as unique \*/*

**ADD** **UNIQUE** (ID);

**ALTER** **TABLE** Students */\* Set multiple columns as unique \*/*

**ADD** **CONSTRAINT** PK\_Student */\* Naming a unique constraint \*/*

**UNIQUE** (ID, FirstName);

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

A FOREIGN KEY comprises of single or collection of fields in a table that essentially refer to the PRIMARY KEY in another table. Foreign key constraint ensures referential integrity in the relation between two tables.  
The table with the foreign key constraint is labelled as the child table, and the table containing the candidate key is labelled as the referenced or parent table.

**CREATE** **TABLE** Students ( */\* Create table with foreign key - Way 1 \*/*

ID INT **NOT** **NULL**

Name VARCHAR(255)

LibraryID INT

**PRIMARY** **KEY** (ID)

**FOREIGN KEY** (Library\_ID) **REFERENCES** Library(LibraryID)

);

**CREATE** **TABLE** Students ( */\* Create table with foreign key - Way 2 \*/*

ID INT **NOT NULL PRIMARY KEY**

Name VARCHAR(255)

LibraryID INT **FOREIGN KEY** (Library\_ID) **REFERENCES** Library(LibraryID)

);

**ALTER** **TABLE** Students */\* Add a new foreign key \*/*

**ADD** **FOREIGN** **KEY** (LibraryID)

**REFERENCES** Library (LibraryID);

Q   =>   What type of integrity constraint does the foreign key ensure?

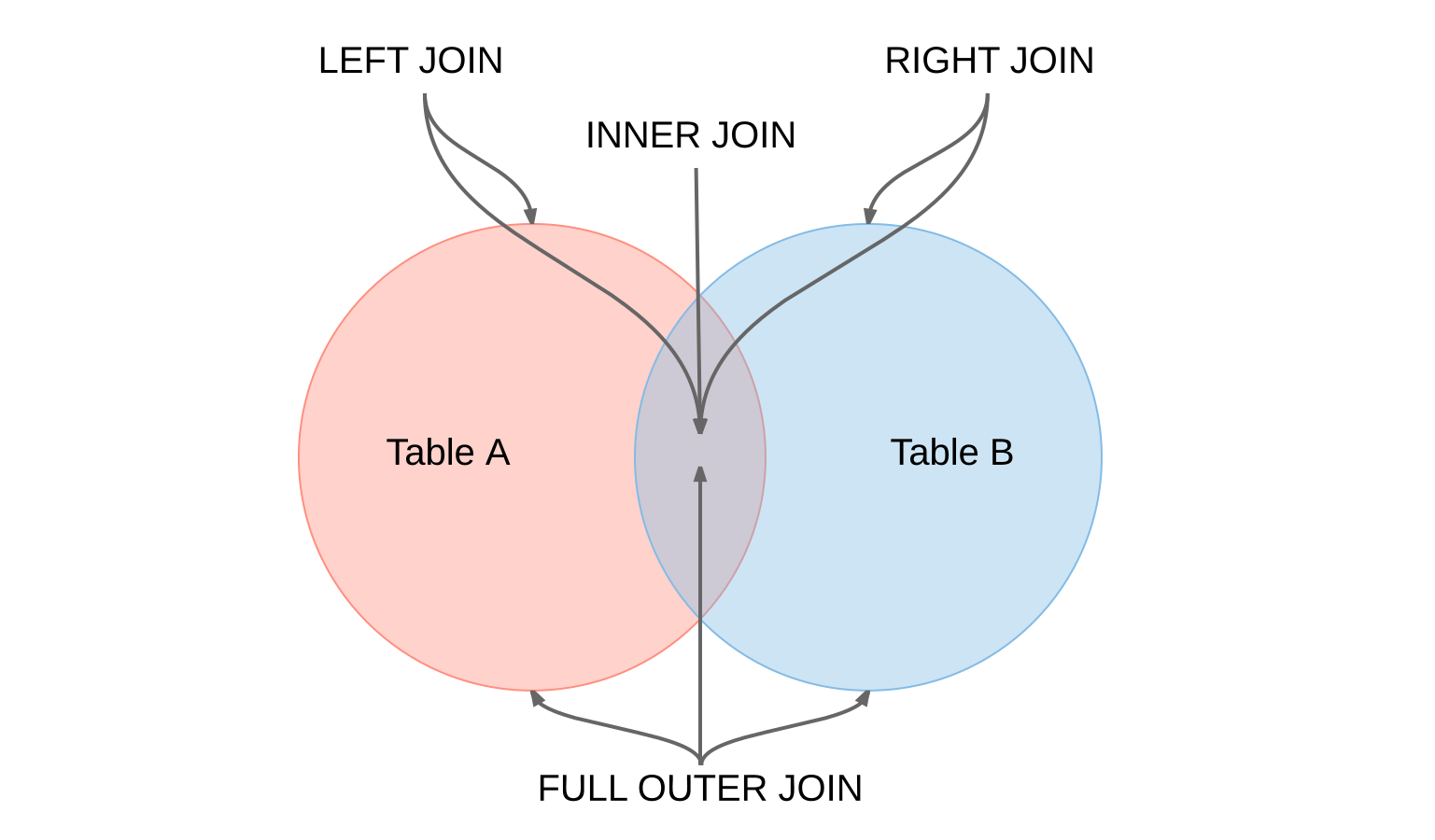


Q   =>   Write a SQL statement to add a FOREIGN KEY 'col\_fk' that references 'col\_pk' in 'table\_x'.



**11. What is a Join? List its different types.**

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



There are four different types of JOINs in SQL:

* **(INNER) JOIN**: Retrieves records that have matching values in both tables involved in the join. This is the widely used join for queries.
* **SELECT** \*
* **FROM** Table\_A
* **JOIN** Table\_B;
* **SELECT** \*
* **FROM** Table\_A
* **INNER JOIN** Table\_B;
* **LEFT (OUTER) JOIN**: Retrieves all the records/rows from the left and the matched records/rows from the right table.
* **SELECT** \*
* **FROM** Table\_A A
* **LEFT JOIN** Table\_B B
* **ON** A.col = B.col;
* **RIGHT (OUTER) JOIN**: Retrieves all the records/rows from the right and the matched records/rows from the left table.
* **SELECT** \*
* **FROM** Table\_A A
* **RIGHT JOIN** Table\_B B
* **ON** A.col = B.col;
* **FULL (OUTER) JOIN**: Retrieves all the records where there is a match in either the left or right table.
* **SELECT** \*
* **FROM** Table\_A A
* **FULL JOIN** Table\_B B
* **ON** A.col = B.col;

**12. What is a Self-Join?**

A **self JOIN**is a case of regular join where a table is joined to itself based on some relation between its own column(s). Self-join uses the INNER JOIN or LEFT JOIN clause and a table alias is used to assign different names to the table within the query.

**SELECT** A.emp\_id **AS** "Emp\_ID",A.emp\_name **AS** "Employee",

B.emp\_id **AS** "Sup\_ID",B.emp\_name **AS** "Supervisor"

**FROM** employee A, employee B

**WHERE** A.emp\_sup = B.emp\_id;

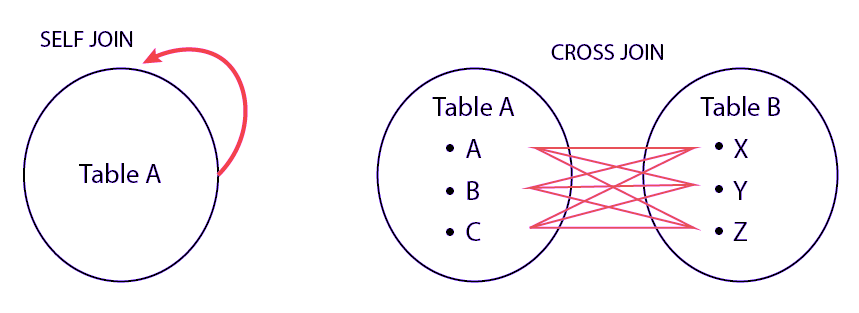
**13. What is a Cross-Join?**

Cross join can be defined as a cartesian product of the two tables included in the join. The table after join contains the same number of rows as in the cross-product of number of rows in the two tables. If a WHERE clause is used in cross join then the query will work like an INNER JOIN.

**SELECT** stu.name, sub.subject

**FROM** students **AS** stu

**CROSS** **JOIN** subjects **AS** sub;



Q   =>   Write a SQL statement to CROSS JOIN 'table\_1' with 'table\_2' and fetch 'col\_1' from table\_1 & 'col\_2' from table\_2 respectively. Do not use alias.



Q   =>   Write a SQL statement to perform SELF JOIN for 'Table\_X' with alias 'Table\_1' and 'Table\_2', on columns 'Col\_1' and 'Col\_2' respectively.



**14. What is an Index? Explain its different types.**

A database index is a data structure that provides quick lookup of data in a column or columns of a table. It enhances the speed of operations accessing data from a database table at the cost of additional writes and memory to maintain the index data structure.

**CREATE** **INDEX** index\_name */\* Create Index \*/*

**ON** table\_name (column\_1, column\_2);

**DROP** **INDEX** index\_name; */\* Drop Index \*/*

There are different types of indexes that can be created for different purposes:

* **Unique and Non-Unique Index**:

Unique indexes are indexes that help maintain data integrity by ensuring that no two rows of data in a table have identical key values. Once a unique index has been defined for a table, uniqueness is enforced whenever keys are added or changed within the index.

**CREATE** **UNIQUE** **INDEX** myIndex

**ON** students (enroll\_no);

Non-unique indexes, on the other hand, are not used to enforce constraints on the tables with which they are associated. Instead, non-unique indexes are used solely to improve query performance by maintaining a sorted order of data values that are used frequently.

* **Clustered and Non-Clustered Index**:

Clustered indexes are indexes whose order of the rows in the database correspond to the order of the rows in the index. This is why only one clustered index can exist in a given table, whereas, multiple non-clustered indexes can exist in the table.

The only difference between clustered and non-clustered indexes is that the database manager attempts to keep the data in the database in the same order as the corresponding keys appear in the clustered index.

Clustering index can improve the performance of most query operations because they provide a linear-access path to data stored in the database.

Q   =>   Write a SQL statement to create a UNIQUE INDEX "my\_index" on "my\_table" for fields "column\_1" & "column\_2".



**15. What is the difference between Clustered and Non-clustered index?**

As explained above, the differences can be broken down into three small factors -

1. Clustered index **modifies the way**records are stored in a database based on the indexed column. Non-clustered index creates a separate entity within the table which references the original table.
2. Clustered index is used for easy and **speedy retrieval**of data from the database, whereas, fetching records from the non-clustered index is relatively slower.
3. In SQL, a table can have **a single**clustered index whereas it can have multiple non-clustered indexes.

**16. What is Data Integrity?**

Data Integrity is the assurance of accuracy and consistency of data over its entire life-cycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data. It also defines integrity constraints to enforce business rules on the data when it is entered into an application or a database.

**17. What is a Query?**

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

**SELECT** fname, lname */\* select query \*/*

**FROM** myDb.students

**WHERE** student\_id = 1;

**UPDATE** myDB.students */\* action query \*/*

**SET** fname = 'Captain', lname = 'America'

**WHERE** student\_id = 1;

**18. What is a Subquery? What are its types?**

A subquery is a query within another query, also known as **nested query**or **inner query**. It is used to restrict or enhance the data to be queried by the main query, thus restricting or enhancing the output of the main query respectively. For example, here we fetch the contact information for students who have enrolled for the maths subject:

**SELECT** name, email, mob, address

**FROM** myDb.contacts

**WHERE** roll\_no **IN** (

**SELECT** roll\_no

**FROM** myDb.students

**WHERE** subject = 'Maths');

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

* A **correlated**subquery cannot be considered as an independent query, but it can refer the column in a table listed in the FROM of the main query.
* A **non-correlated**subquery can be considered as an independent query and the output of subquery is substituted in the main query.

Q   =>   Write a SQL query to update the field "status" in table "applications" from 0 to 1.



Q   =>   Write a SQL query to select the field "app\_id" in table "applications" less than 1000.



Q   =>   Write a SQL query to fetch the field "app\_name" from "apps" where "apps.id" is equal to the above collection of "app\_id".



**19. What is the SELECT statement?**

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

**SELECT** \* **FROM** myDB.students;

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

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

* **WHERE** clause in SQL is used to filter records that are necessary, based on specific conditions.
* **ORDER BY** clause in SQL is used to sort the records based on some field(s) in ascending (**ASC**) or descending order (**DESC**).
* **SELECT** \*
* **FROM** myDB.students
* **WHERE** graduation\_year = 2019
* **ORDER** **BY** studentID **DESC**;
* **GROUP BY** clause in SQL is used to group records with identical data and can be used in conjuction with some aggregation functions to produce summarized results from the database.
* **HAVING** clause in SQL is used to filter records in combination with the GROUP BY clause. It is different from WHERE, since WHERE clause cannot filter aggregated records.
* **SELECT** **COUNT**(studentId), country
* **FROM** myDB.students
* **WHERE** country != "INDIA"
* **GROUP** **BY** country
* **HAVING** **COUNT**(studentID) > 5;

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

The **UNION**operator combines and returns the result-set retrieved by two or more SELECT statements.  
The **MINUS**operator in SQL is used to remove duplicates from the result-set obtained by the second SELECT query from the result-set obtained by the first SELECT query and then return the filtered results from the first.  
The **INTERSECT**clause in SQL combines the result-set fetched by the two SELECT statements where records from one match the other and then returns this intersection of result-sets.

Certain conditions need to be met before executing either of the above statements in SQL -

* + Each SELECT statement within the clause must have the same number of columns
  + The columns must also have similar data types
  + The columns in each SELECT statement should necessarily have the same order

**SELECT** name **FROM** Students */\* Fetch the union of queries \*/*

**UNION**

**SELECT** name **FROM** Contacts;

**SELECT** name **FROM** Students */\* Fetch the union of queries with duplicates\*/*

**UNION ALL**

**SELECT** name **FROM** Contacts;

**SELECT** name **FROM** Students */\* Fetch names from students \*/*

**MINUS** */\* that aren't present in contacts \*/*

**SELECT** name **FROM** Contacts;

**SELECT** name **FROM** Students */\* Fetch names from students \*/*

**INTERSECT** */\* that are present in contacts as well \*/*

**SELECT** name **FROM** Contacts;

Q   =>   Write a SQL query to fetch "names" that are present in either table "accounts" or in table "registry".



Q   =>   Write a SQL query to fetch "names" that are present in "accounts" but not in table "registry".



Q   =>   Write a SQL query to fetch "names" from table "contacts" that are neither present in "accounts.name" nor in "registry.name".



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

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

**Working with SQL Cursor**

* + **DECLARE** a cursor after any variable declaration. The cursor declaration must always be associated with a SELECT Statement.
  + Open cursor to initialize the result set. The **OPEN** statement must be called before fetching rows from the result set.
  + **FETCH** statement to retrieve and move to the next row in the result set.
  + Call the **CLOSE** statement to deactivate the cursor.
  + Finally use the **DEALLOCATE** statement to delete the cursor definition and release the associated resources.

**DECLARE** @**name** **VARCHAR**(50) */\* Declare All Required Variables \*/*

**DECLARE** db\_cursor **CURSOR** **FOR** */\* Declare Cursor Name\*/*

**SELECT** name

**FROM** myDB.students

**WHERE** parent\_name **IN** ('Sara', 'Ansh')

**OPEN** db\_cursor */\* Open cursor and Fetch data into @name \*/*

**FETCH** next

**FROM** db\_cursor

**INTO** @**name**

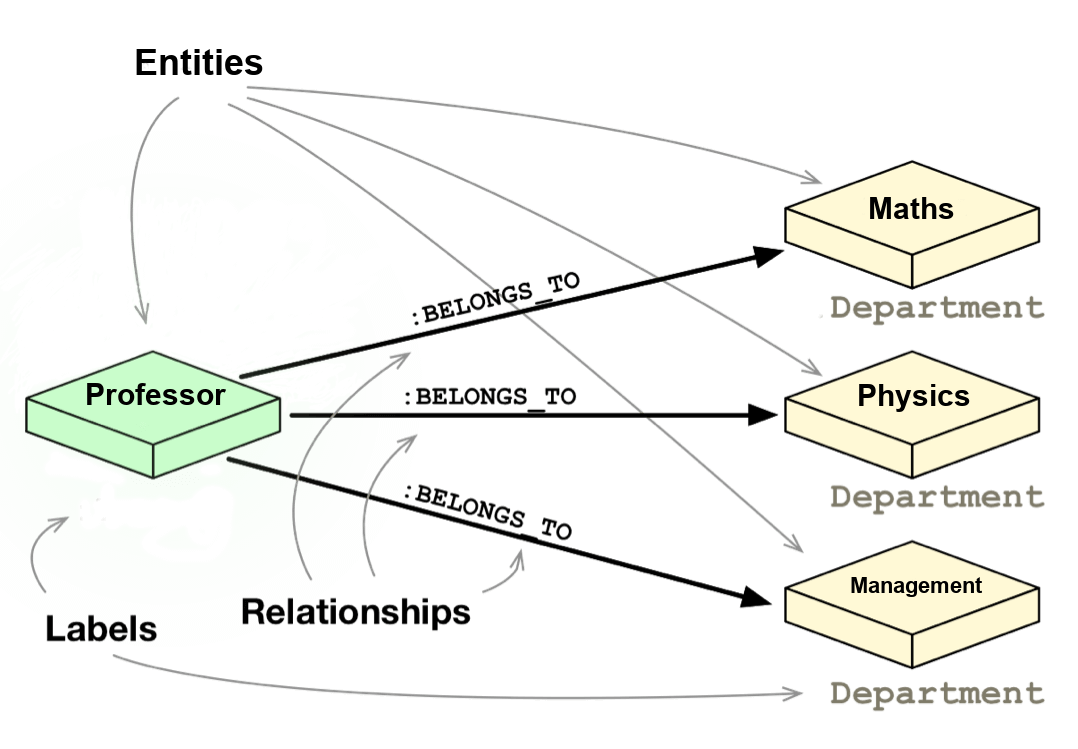
**CLOSE** db\_cursor */\* Close the cursor and deallocate the resources \*/*

**DEALLOCATE** db\_cursor

**23. What are Entities and Relationships?**

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

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



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

* + **One-to-One** - This can be defined as the relationship between two tables where each record in one table is associated with the maximum of one record in the other table.
  + **One-to-Many** & **Many-to-One** - This is the most commonly used relationship where a record in a table is associated with multiple records in the other table.
  + **Many-to-Many** - This is used in cases when multiple instances on both sides are needed for defining a relationship.
  + **Self Referencing Relationships** - This is used when a table needs to define a relationship with itself.

**25. What is an Alias in SQL?**

An alias is a feature of SQL that is supported by most, if not all, RDBMSs. It is a temporary name assigned to the table or table column for the purpose of a particular SQL query. In addition, aliasing can be employed as an obfuscation technique to secure the real names of database fields. A table alias is also called a **correlation name**.

An alias is represented explicitly by the **AS** keyword but in some cases the same can be performed without it as well. Nevertheless, using the AS keyword is always a good practice.

**SELECT** A.emp\_name **AS** "Employee" */\* Alias using AS keyword \*/*

B.emp\_name **AS** "Supervisor"

**FROM** employee A, employee B */\* Alias without AS keyword \*/*

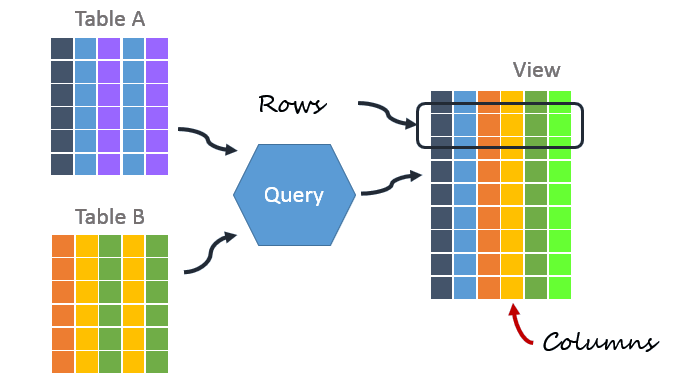
**WHERE** A.emp\_sup = B.emp\_id;

Q   =>   Write an SQL statement to select all from table "Limited" with alias "Ltd".



**26. What is a View?**

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



**27. What is Normalization?**

Normalization represents the way of organizing structured data in the database efficiently. It includes creation of tables, establishing relationships between them, and defining rules for those relationships. Inconsistency and redundancy can be kept in check based on these rules, hence, adding flexibility to the database.

**28. What is Denormalization?**

Denormalization is the inverse process of normalization, where the normalized schema is converted into a schema which has redundant information. The performance is improved by using redundancy and keeping the redundant data consistent. The reason for performing denormalization is the overheads produced in query processor by an over-normalized structure.

**29. What are the various forms of Normalization?**

Normal Forms are used to eliminate or reduce redundancy in database tables. The different forms are as follows:

* + **First Normal Form**

A relation is in first normal form if every attribute in that relation is a **single-valued attribute**. If a relation contains composite or multi-valued attribute, it violates the first normal form. Let's consider the following **students** table. Each student in the table, has a name, his/her address and the books they issued from the public library -

**Students Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | **Address** | **Books Issued** | **Salutation** |
| Sara | Amanora Park Town 94 | Until the Day I Die (Emily Carpenter), Inception (Christopher Nolan) | Ms. |
| Ansh | 62nd Sector A-10 | The Alchemist (Paulo Coelho), Inferno (Dan Brown) | Mr. |
| Sara | 24th Street Park Avenue | Beautiful Bad (Annie Ward), Woman 99 (Greer Macallister) | Mrs. |
| Ansh | Windsor Street 777 | Dracula (Bram Stoker) | Mr. |

As we can observe, the Books Issued field has more than one values per record and to convert it into 1NF, this has to be resolved into separate individual records for each book issued. Check the following table in 1NF form -

**Students Table (1st Normal Form)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | **Address** | **Books Issued** | **Salutation** |
| Sara | Amanora Park Town 94 | Until the Day I Die (Emily Carpenter) | Ms. |
| Sara | Amanora Park Town 94 | Inception (Christopher Nolan) | Ms. |
| Ansh | 62nd Sector A-10 | The Alchemist (Paulo Coelho) | Mr. |
| Ansh | 62nd Sector A-10 | Inferno (Dan Brown) | Mr. |
| Sara | 24th Street Park Avenue | Beautiful Bad (Annie Ward) | Mrs. |
| Sara | 24th Street Park Avenue | Woman 99 (Greer Macallister) | Mrs. |
| Ansh | Windsor Street 777 | Dracula (Bram Stoker) | Mr. |

* + **Second Normal Form**

A relation is in second normal form if it satisfies the conditions for first normal form and does not contain any partial dependency. A relation in 2NF has **no partial dependency**, i.e., it has no non-prime attribute that depends on any proper subset of any candidate key of the table. Often, specifying a single column Primary Key is the solution to the problem. Examples -

* + - **Example 1** - Consider the above example. As we can observe, Students Table in 1NF form has a candidate key in the form of [Student, Address] that can uniquely identify all records in the table. The field Books Issued (non-prime attribute) depends partially on the Student field. Hence, the table is not in 2NF. To convert it into 2nd Normal Form, we will partition the tables into two while specifying a new ***Primary Key*** attribute to identify the individual records in the Students table. The ***Foreign Key*** constraint will be set on the other table to ensure referential integrity.

**Students Table (2nd Normal Form)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_ID** | **Student** | **Address** | **Salutation** |
| 1 | Sara | Amanora Park Town 94 | Ms. |
| 2 | Ansh | 62nd Sector A-10 | Mr. |
| 3 | Sara | 24th Street Park Avenue | Mrs. |
| 4 | Ansh | Windsor Street 777 | Mr. |

**Books Table (2nd Normal Form)**

|  |  |
| --- | --- |
| **Student\_ID** | **Book Issued** |
| 1 | Until the Day I Die (Emily Carpenter) |
| 1 | Inception (Christopher Nolan) |
| 2 | The Alchemist (Paulo Coelho) |
| 2 | Inferno (Dan Brown) |
| 3 | Beautiful Bad (Annie Ward) |
| 3 | Woman 99 (Greer Macallister) |
| 4 | Dracula (Bram Stoker) |

* + - **Example 2** - Consider the following dependencies in relation R(W,X,Y,Z)

WX -> Y [W and X together determine Y]   
 XY -> Z [X and Y together determine Z]

Here, WX is the only candidate key and there is no partial dependency, i.e., any proper subset of WX doesn’t determine any non-prime attribute in the relation.

* + **Third Normal Form**

A relation is said to be in the third normal form, if it satisfies the conditions for second normal form and there is **no transitive dependency** between the non-prime attributes, i.e.,all non-prime attributes are determined only by the candidate keys of the relation and not by any other non-prime attribute.

* + - **Example 1** - Consider the Students Table in the above example. As we can observe, Students Table in 2NF form has a single candidate key Student\_ID (primary key) that can uniquely identify all records in the table. The field Salutation (non-prime attribute), however, depends on the Student Field rather than the candidate key. Hence, the table is not in 3NF. To convert it into 3rd Normal Form, we will once again partition the tables into two while specifying a new ***Foreign Key*** constraint to identify the salutations for individual records in the Students table. The ***Primary Key*** constraint for the same will be set on the Salutations table to identify each record uniquely.

**Students Table (3rd Normal Form)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_ID** | **Student** | **Address** | **Salutation\_ID** |
| 1 | Sara | Amanora Park Town 94 | 1 |
| 2 | Ansh | 62nd Sector A-10 | 2 |
| 3 | Sara | 24th Street Park Avenue | 3 |
| 4 | Ansh | Windsor Street 777 | 1 |

**Books Table (3rd Normal Form)**

|  |  |
| --- | --- |
| **Student\_ID** | **Book Issued** |
| 1 | Until the Day I Die (Emily Carpenter) |
| 1 | Inception (Christopher Nolan) |
| 2 | The Alchemist (Paulo Coelho) |
| 2 | Inferno (Dan Brown) |
| 3 | Beautiful Bad (Annie Ward) |
| 3 | Woman 99 (Greer Macallister) |
| 4 | Dracula (Bram Stoker) |

**Salutations Table (3rd Normal Form)**

|  |  |
| --- | --- |
| **Salutation\_ID** | **Salutation** |
| 1 | Ms. |
| 2 | Mr. |
| 3 | Mrs. |

* + - **Example 2** - Consider the following dependencies in relation R(P,Q,R,S,T)

P -> QR [P together determine C]   
 RS -> T [B and C together determine D]   
 Q -> S   
 T -> P

For the above relation to exist in 3NF, all possible candidate keys in above relation should be {P, RS, QR, T}.

* + **Boyce-Codd Normal Form**

A relation is in Boyce-Codd Normal Form if satisfies the conditions for third normal form and for every functional dependency, Left-Hand-Side is super key. In other words, a relation in BCNF has non-trivial functional dependencies in the form X –> Y, such that X is always a super key. For example - In the above example, Student\_ID serves as the sole unique identifier for the Students Table and Salutation\_ID for the Salutations Table, thus these tables exist in BCNF. Same cannot be said for the Books Table and there can be several books with common Book Names and same Student\_ID.

**30. What are the TRUNCATE, DELETE and DROP statements?**

**DELETE**statement is used to delete rows from a table.

**DELETE** **FROM** Candidates

**WHERE** CandidateId > 1000;

**TRUNCATE**command is used to delete all the rows from the table and free the space containing the table.

**TRUNCATE** **TABLE** Candidates;

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

**DROP** **TABLE** Candidates;

Q   =>   Write a SQL statement to wipe a table 'Temporary' from memory.



Q   =>   Write a SQL query to remove first 1000 records from table 'Temporary' based on 'id'.



Q   =>   Write a SQL statement to delete the table 'Temporary' while keeping its relations intact.



**31. What is the difference between DROP and TRUNCATE statements?**

If a table is dropped, all things associated with the tables are dropped as well. This includes - the relationships defined on the table with other tables, the integrity checks and constraints, access privileges and other grants that the table has. To create and use the table again in its original form, all these relations, checks, constraints, privileges and relationships need to be redefined. However, if a table is truncated, none of the above problems exist and the table retains its original structure.

**32. What is the difference between DELETE and TRUNCATE statements?**

The TRUNCATE command is used to delete all the rows from the table and free the space containing the table.  
The DELETE command deletes only the rows from the table based on the condition given in the where clause or deletes all the rows from the table if no condition is specified. But it does not free the space containing the table.

**33. What are Aggregate and Scalar functions?**

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

* + **AVG()**- Calculates the mean of a collection of values.
  + **COUNT()**- Counts the total number of records in a specific table or view.
  + **MIN()**- Calculates the minimum of a collection of values.
  + **MAX()**- Calculates the maximum of a collection of values.
  + **SUM()**- Calculates the sum of a collection of values.
  + **FIRST()**- Fetches the first element in a collection of values.
  + **LAST()**- Fetches the last element in a collection of values.

*Note: All aggregate functions described above ignore NULL values except for the COUNT function.*

A **scalar function**returns a single value based on the input value. Following are the widely used SQL scalar functions:

* + **LEN()**- Calculates the total length of the given field (column).
  + **UCASE()**- Converts a collection of string values to uppercase characters.
  + **LCASE()**- Converts a collection of string values to lowercase characters.
  + **MID()**- Extracts substrings from a collection of string values in a table.
  + **CONCAT()**- Concatenates two or more strings.
  + **RAND()**- Generates a random collection of numbers of given length.
  + **ROUND()**- Calculates the round off integer value for a numeric field (or decimal point values).
  + **NOW()**- Returns the current data & time.
  + **FORMAT()**- Sets the format to display a collection of values.

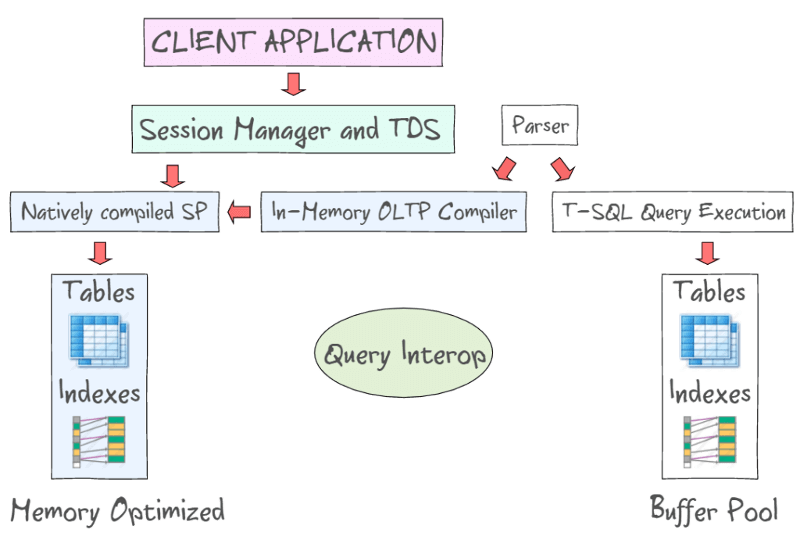
**34. What is User-defined function? What are its various types?**

The user-defined functions in SQL are like functions in any other programming language that accept parameters, perform complex calculations, and return a value. They are written to use the logic repetitively whenever required. There are two types of SQL user-defined functions:

* + **Scalar Function**: As explained earlier, user-defined scalar functions return a single scalar value.
  + **Table Valued Functions**: User-defined table-valued functions return a table as output.
    - **Inline**: returns a table data type based on a single SELECT statement.
    - **Multi-statement**: returns a tabular result-set but, unlike inline, multiple SELECT statements can be used inside the function body.

**35. What is OLTP?**

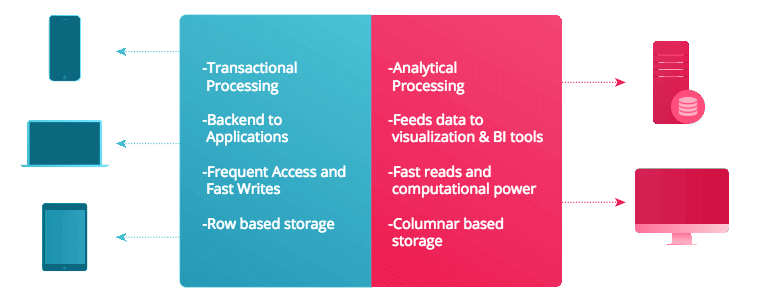
OLTP stands for Online Transaction Processing, is a class of software applications capable of supporting transaction-oriented programs. An essential attribute of an OLTP system is its ability to maintain concurrency. To avoid single points of failure, OLTP systems are often decentralized. These systems are usually designed for a large number of users who conduct short transactions. Database queries are usually simple, require sub-second response times and return relatively few records. Here is an insight into the working of an OLTP system [*Note - The figure is not important for interviews*] -



**36. What are the differences between OLTP and OLAP?**

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

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



**37. What is Collation? What are the different types of Collation Sensitivity?**

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

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

**38. What is a Stored Procedure?**

A stored procedure is a subroutine available to applications that access a relational database management system (RDBMS). Such procedures are stored in the database data dictionary. The sole disadvantage of stored procedure is that it can be executed nowhere except in the database and occupies more memory in the database server. It also provides a sense of security and functionality as users who can't access the data directly can be granted access via stored procedures.

**DELIMITER** $$

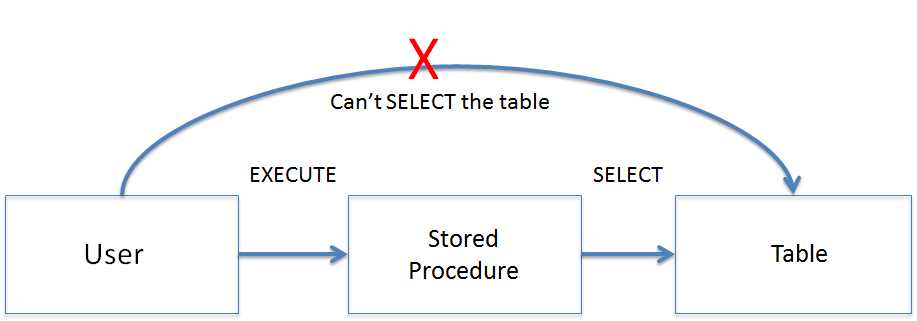
**CREATE** **PROCEDURE** FetchAllStudents()

**BEGIN**

**SELECT** \* **FROM** myDB.students;

**END** $$

**DELIMITER** ;



**39. What is a Recursive Stored Procedure?**

A stored procedure which calls itself until a boundary condition is reached, is called a recursive stored procedure. This recursive function helps the programmers to deploy the same set of code several times as and when required. Some SQL programming languages limit the recursion depth to prevent an infinite loop of procedure calls from causing a stack overflow, which slows down the system and may lead to system crashes.

**DELIMITER** $$ */\* Set a new delimiter => $$ \*/*

**CREATE** **PROCEDURE** calctotal( */\* Create the procedure \*/*

**IN** number INT, */\* Set Input and Ouput variables \*/*

**OUT** total INT

) **BEGIN**

**DECLARE** score INT **DEFAULT** **NULL**; */\* Set the default value => "score" \*/*

**SELECT** awards **FROM** achievements */\* Update "score" via SELECT query \*/*

**WHERE** id = number **INTO** score;

**IF** score **IS** **NULL** **THEN** **SET** total = 0; */\* Termination condition \*/*

**ELSE**

**CALL** calctotal(number+1); */\* Recursive call \*/*

**SET** total = total + score; */\* Action after recursion \*/*

**END** IF;

**END** $$ */\* End of procedure \*/*

**DELIMITER** ; */\* Reset the delimiter \*/*

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

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

**SELECT** \* **INTO** Students\_copy

**FROM** Students **WHERE** 1 = 2;

**41. What is Pattern Matching in SQL?**

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

* + **Using the % wildcard to perform a simple search**  
    The % wildcard matches zero or more characters of any type and can be used to define wildcards both before and after the pattern. Search a student in your database with first name beginning with the letter K:
  + **SELECT** \*
  + **FROM** students
  + **WHERE** first\_name **LIKE** 'K%'
  + **Omitting the patterns using the NOT keyword**  
    Use the NOT keyword to select records that don't match the pattern. This query returns all students whose first name does not begin with K.
  + **SELECT** \*
  + **FROM** students
  + **WHERE** first\_name **NOT LIKE** 'K%'
  + **Matching a pattern anywhere using the % wildcard twice**  
    Search for a student in the database where he/she has a K in his/her first name.
  + **SELECT** \*
  + **FROM** students
  + **WHERE** first\_name **LIKE** '%Q%'
  + **Using the \_ wildcard to match pattern at a specific position**  
    The \_ wildcard matches exactly one character of any type. It can be used in conjunction with % wildcard. This query fetches all students with letter K at the third position in their first name.
  + **SELECT** \*
  + **FROM** students
  + **WHERE** first\_name **LIKE** '\_\_K%'
  + **Matching patterns for specific length**  
    The \_ wildcard plays an important role as a limitation when it matches exactly one character. It limits the length and position of the matched results. For example -
  + **SELECT** \* */\* Matches first names with three or more letters \*/*
  + **FROM** students
  + **WHERE** first\_name **LIKE** '\_\_\_%'
  + **SELECT** \* */\* Matches first names with exactly four characters \*/*
  + **FROM** students

**WHERE** first\_name **LIKE** '\_\_\_\_'

# **SQL Server Interview Questions**

This guide will help you to crack any sql server interview round. If you are looking for sql interview questions you can visit [this page](https://www.interviewbit.com/sql-server-interview-questions/).

### 1. What is SQL Server?

SQL Server is a relational database management system. It provides the functionality of storing and retrieving data to applications. Microsoft developed SQL Server on April 24, 1989. There are several editions of SQL Server serving different audiences.

### 2. Which TCP/IP port does SQL Server run on?

By default SQL Server runs on port 1433. We can configure the server to listen to a specific port. This can be done by changing the TCP/IP properties from configuration-box in SQL Server.

### 3. Describe the different index configurations possible for a table?

Different index configurations possible for a table are:

1. **A clustered index:**

When only a single clustered index is present.

1. **A non-clustered index:**

When only a single non-clustered index is present.

1. **Many non-clustered indexes:**

When more than one non-clustered indexes is present.

1. **A clustered index and a non-clustered index:**

When a single clustered index and a single non-clustered index is present.

1. **A clustered index and many non-clustered indexes:**

When a single clustered index and more than one non-clustered indexes are present.

1. **No index:**

When there are no indexes present.

### 4. What is the difference between clustered and non-clustered index?

Indexing is a way of optimizing database query time. It helps in quickly locating and accessing the data in the database. It avoids full table scans.

#### Difference Between Clustered and Non-Clustered Index

|  |  |
| --- | --- |
| **Clustered Index** | **Non-Clustered Index** |
| It physically sorts the rows of a table based on the primary key or on a column that is unique and not null (generally we use primary key). | This is an index structure that is separate from the actual table which sorts one or more selected columns. Think of it as another table with few columns sorted together. For example, if you have a phone\_book table where it is sorted by the name of a person and you want to know the count of people living in a specific country. What you’ll do? You’ll make another table with columns ‘country’ and ‘count\_of\_people’ which is sorted by the country name. Now finding the number of people in a given country will be much faster otherwise you will have to do the full table scan to get the answer. This is a non-clustered index. |
| Querying data is fast. A typical use case can be where there are range-based queries on the primary key. | Querying data is faster when there is no index in a table. A scenario where it can be used is when you want to find no. of people from a particular country in the phonebook. Even if the phonebook is sorted by name, you would like to have some kind of mapping of country with no. of people living there for a faster answer to such queries. |
| There can only be one clustered index per table. | There can be many non-clustered indexes per table. |
| It doesn’t need extra disk space. | It requires extra space to store those indexes. |
| It is faster than the non-clustered index. | It is slower than the clustered index in terms of SELECT queries. |
| Updation and Insertion are slow as the sorted order has to be maintained (can be faster when insertion always happens at the last, e.g.: Index on ID col). | Updation and Insertion are slow as the sorted order has to be maintained. |

### 5. What are the two authentication modes in SQL Server?

Authentication means identifying a user based on its username and password.  
Two authentication modes on SQL Server are

1. **Windows Authentication:**

It is the default authentication mode in SQL Server. Trusted user and group accounts are authenticated when they login to the system. They do not have to present any additional credentials.

1. **Mixed Authentication:**

It supports Windows authentication as well as SQL Server authentication. Windows authentication is the same as above. SQL Server maintains a username and password for authentication of valid users.

You can choose an authentication mode by changing Server Authentication on Security page in Properties of SQL Server Management Studio.

### 6. What part does database design have to play in the performance of the SQL Server-based application?

Database design plays a vital role in the performance of SQL Server-based applications. Generally, when data is less, people do not care about database design. As when data is less there aren’t any noticeable performance issues because of bad database design. When data grows, there will be noticeable performance issues because of data redundancy. This is where normalization of the database comes into play. Right design principles ensure better performance at the later stage of software development. There will not be much redundant data. This will lead to reduced storage space. Also, it will reduce overhead to maintain consistency in the database.

### 7. What is a function in SQL Server?

Functions are part of SQL. A function in SQL Server is a group of statements that might take input, perform some task and return a result.  
There are two types of function in SQL Server:

* **System Defined Function:**

These functions are built-in ready-to-use and provided by SQL Server. Pass in input parameters if it takes one and get the result.  
Example: Below code show min, max, and sum of ‘salary’ column values from ‘employee’ table

**SELECT MIN**(salary) **AS** MinSalary, **MAX**(salary) **AS** MaxSalary, **SUM**(salary) **AS** TotalSalary

**FROM** employee

* **User Defined Function:**

These are the functions that are written by users.

**CREATE FUNCTION** getAverageSalary(@salary **int**)

**RETURNS int**

**AS**

**BEGIN RETURN**(**SELECT** @salary)

**END**

Q   =>   Write a SQL statement to get the average 'salary' as 'avg\_salary' from the table 'employees'.



### 8. What is CHECK constraint?

CHECK constraint is applied to any column to limit the values that can be placed in it. It helps to enforce integrity in the database.  
Suppose, your website caters to users between age 18 and 60 years. You can use CHECK to ensure that users who are creating an account have age in that range.

**CREATE TABLE** Users (

id **int NOT** NULL,

first\_name **varchar**(255) **NOT** NULL,

last\_name **varchar**(255) **NOT** NULL,

age **int CHECK** (age >= 18 **AND** age <= 60)

);

Q   =>   Write a SQL statement create a table 'students' with column 'id' of type 'int' and 'date\_of\_birth' where 'date\_of\_birth' should not be less than 2000 and not more than 2010. of type 'int'. 'id' should not be null.



### 9. What is a trigger in SQL Server?

The trigger is a special type of stored procedure. It gets invoked automatically when an event like INSERT, UPDATE, DELETE, etc. happens on the database server. You can use it, for example, to enforce referential integrity in the database. Suppose you want to delete an author from the ‘authors’ table. You can use triggers to delete all rows in the ‘books’ table which has ‘author\_id’ as of the deleted author.

Types of triggers:

* **DML trigger:**

DML trigger gets fired whenever a user tries to manipulate data using DML(Data Manipulation Language) event on the database server. DML events are INSERT, DELETE, or UPDATE.

* **DDL trigger:**

DDL trigger gets fired whenever a user tries to manipulate data using DDL(Data Definition Language) event on the database server. DDL events are CREATE, ALTER, DROP, etc.

* **Logon trigger:**

Logon trigger gets fired when a LOGON event is raised whenever a user’s session is created.

**CREATE** [**OR REPLACE** ] **TRIGGER** trigger\_name

{**BEFORE** | **AFTER** | **INSTEAD OF** }

{**INSERT** | **UPDATE** | **DELETE**}

[OF col\_name]

**ON** table\_name

[REFERENCING OLD **AS** o NEW **AS** n]

[FOR EACH ROW]

**WHEN** (condition)

**DECLARE**

Declaration-statements

**BEGIN**

Executable-statements

**EXCEPTION**

Exception-handling-statements

**END**;

### 10. What are the differences between local and global temporary tables?

Temporary tables are the tables that ephemeral in nature. Temporary tables are useful when we want to work with a small subset of records from a large table. Instead of filtering records, again and again, we can store that small subset of records in a temporary table. We can perform our queries on it.

#### Difference Between Local and Global Temporary Tables

|  |  |
| --- | --- |
| **Local Temporary Tables** | **Global Temporary Tables** |
| Local temporary tables are only visible to that session of SQL Server that has created it. | Global temporary tables are visible to all SQL Server sessions. |
| Local temporary tables are automatically dropped when the session of the SQL Server that has created it is closed. | Global temporary tables are dropped when the last session of SQL Server referencing to the global temporary tables is closed. |
| Local temporary tables are prefixed with a single pound ‘#’ symbol. | Global temporary tables are prefixed with double pounds ‘##’ symbol. |
| SQL Server appends some random numbers at the end of the local temporary table name. | SQL Server doesn’t append any random numbers at the end of the global temporary table name. |

### 11. What is the SQL Server Agent?

SQL Server Agent is a background tool for Microsoft SQL Server. It helps the database administrator(DBA) to schedule a job to run after a specific interval of time. These tasks can be scheduling backups, handling reporting services subscription or log-shipping tasks.

### 12. What are the different types of backups available in SQL Server?

Suppose your server is running fine. Your application is booming all over the internet. Then, due to short-circuiting your servers went on fire. Now, all the data is gone and there’s nothing to show. Scary? It should be. This is the reason we always want to back up our data. So that in case of any disaster like hardware or software failure there will not be any data loss.

There are several types of backup options.

* **Full backup:**

This backup includes all database objects, system tables, and data. Transactions that occur during the backup are also recorded.

* **Transaction log backup:**

This backup records the transactions since the previous backup. Previous backup can be transaction log backup or full backup (whichever happened last). It then truncates the transaction log. Transaction log backup represents the state of the transaction log at the time the backup is initiated rather than at the time the backup completes. Transaction log backup functions incrementally rather than differentially. When restoring transaction log backup, you must restore in the sequence of transaction log backups.

* **Differential backup:**

This backup backs the data that has been altered since the last full backup. Differential backup requires less time than a full database backup. Differential backups record transactions that occur during the differential backup process.

### 13. What are the scheduled tasks in SQL Server?

Scheduled tasks in SQL Server are predefined steps or tasks. SQL Server Agent automates these tasks. It executes them sequentially and at a scheduled time.

### 14. What is COALESCE in SQL Server?

In SQL Server, COALESCE function returns the first non-null value from a list. If all the values evaluate to null then it will also return null.

**SELECT COALESCE**(NULL, 2, 1, 3) */\* OUTPUT: 2 \*/*

Q   =>   What will the following SQL statement return? "SELECT COALESCE(NULL, 4, NULL, 5)"



### 15. How exceptions are handled in SQL Server programming?

Exceptions in SQL Server are handled using the try and catch block.

**BEGIN** TRY

*--code which might raise exception*

**END** TRY

**BEGIN** CATCH

*--code to run if error occurs in try block*

**END** CATCH

**Try** block contains the code which might raise exception. **Catch** block contains the code which runs in case an exception occurs in the try block.

**BEGIN** TRY

*--divide-by-zero error*

**SELECT** 5 / 0 **AS** Error;

**END** TRY

**BEGIN** CATCH

**SELECT**

ERROR\_NUMBER() **AS** ErrorNumber,

ERROR\_STATE() **AS** ErrorState,

ERROR\_SEVERITY() **AS** ErrorSeverity,

ERROR\_PROCEDURE() **AS** ErrorProcedure,

ERROR\_LINE() **AS** ErrorLine,

ERROR\_MESSAGE() **AS** ErrorMessage;

**END** CATCH;

The above code raises divide-by-zero error and handles it in the catch block. There are some special functions which are only usable inside the catch block. Outside the catch block they return NULL.

These functions are:

* **ERROR\_NUMBER():**As the name says, it returns the error number.
* **ERROR\_STATE():**It returns the state number of the error.
* **ERROR\_SEVERITY():**This function returns the severity value of the error.
* **ERROR\_PROCEDURE():**It returns the name of the stored procedure or function in which the error has occurred.
* **ERROR\_LINE():**Returns the line number at which the error has occurred.
* **ERROR\_MESSAGE():**Returns the message about the error.

### 16. What is the reporting services in SQL Server?

SQL Server Reporting Services(SSRS) is a set of tools that helps in generating reports. Businesses can use it for getting visual insights into data.  
A common use case can be a shopping store. The owner of the store might want to see how the sales of different products are performing. The owner might also want to see the performances in a particular quarter of a year. What could be better than a beautiful and detailed visualization for this? SSRS does that exact thing. It will help us visualize data, meeting our requirements. We can see charts, graphs, and whatnot.

### 17. What is log shipping?

Log shipping is a process in which we automate the back-up process of transaction log files. We back-up the files from a primary database server to a secondary(stand by) database server.  
To set up the log shipping process you must have sysadmin rights on the server.  
Log shipping helps in mitigating the risk of disasters. In case of a disaster like a production server failure, we can use the secondary server.

### 18. What is the master data services in SQL Server?

SQL Server Master Data Services is a product made by Microsoft to develop the Master Data Management(MDM) solutions. It is built on top of SQL Server for backend processing.  
The possibility of inconsistency in the database gives rise to the need for Master Data Management. It becomes especially important when there is a huge amount of data.  
Consider a scenario, there is a Credit Card company which maintains a database. There is a table “Customer” which has the “Address” attribute. There is another table “Bills” which also has the “Address” attribute for a customer. Now suppose a customer moves to a different location and changes his address immediately. It gets updated in the “Customer” table but not in the “Bills” table. Now, this is a disaster as all the bills of this particular customer will be sent to the wrong address.  
The disaster can be avoided by setting up a master customer data. All the tables would reference there for use. This way there will not be any inconsistency and extra memory consumption.

### 19. What are Hotfixes and Patches in SQL Server?

Hotfixes in SQL Server are the updates to fix issues which are not released publicly.  
Patches in SQL Server are the updates to fix known issues or bugs. Microsoft releases patches publicly.

### 20. What are the Magic Tables in SQL Server?

There are two magic tables in SQL Server: "inserted" and "deleted". These tables are called magic tables because they do not exist physically.

"inserted" table stores the data of last inserted row or “insert operation”. "deleted" table stores the data of last deleted row or “delete operation”.

**Note:** An update operation is actually a delete operation + an insert operation i.e., old data is deleted and new data is inserted.

For the sake of example, we assume there is a table with name Employee and columns ‘ID’ and ‘Name’.

SQL Server provides two ways to view data in magic tables.

* **Use Triggers:**

One way is to use triggers on delete, insert or update operations in a table to view the data in “inserted” and “deleted” magic tables.

**CREATE TRIGGER** Inserted\_Trigger

**ON** Employee

**FOR INSERT**

**AS**

**BEGIN**

**SELECT** \* **FROM** Inserted

**END**

Whenever an insert operation will be done on Employee table, this trigger will be called and this will show the “inserted” table.

* **Use OUTPUT clause:**

Use OUTPUT clause: Another way is to use the OUTPUT clause to view data in “inserted” or “deleted” magic tables. OUTPUT returns the copy of data inserted into or deleted from a table.

**INSERT INTO** Employee **OUTPUT** Inserted.\* **VALUES**('Robert')

## SQL Interview Questions for IBM :



**1.How to create a clone table with same structure of other table?**

**Answer:**

We can create clone table using following command:

Create table Employee1 as select \* from Employee where 1=2;

**2.If table named Employee has 50 records and Employee2 has 0 records what will be the output of following query?**

Select e1.\* from Employee e1,Employee e2 where e1.empno=e2.empno;

**Answer:**

Answer is ‘No Rows Found’

**3.What will be the output of following query?**

Select \* from (select ‘a’ union all select ‘b’) Q;

**Answer:**

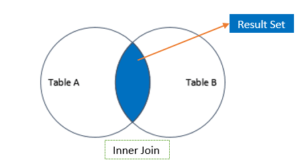
It will throw error because no values are selected in Subquery.

Error code-ORA-00923 from keyword not found expected values.

**4.What is inner join? Explain with Business use?**

**Answer:**

When 2 tables are connected such that it should retrieve only the matching records in both tables.Inner join select only the matching records between 2 tables.You can use Equal to(=) operator or Inner join keyword to apply inner join.This join is most widely used joins in real life applications,reporting,webapps,android apps.



Inner join is nothing but fetching common records from two or more tables.

[Click Here to get information about Joins..](http://www.complexsql.com/sql-joins-2/)

5.What is query to find distinct records without using distinct keyword?(Asked 90 % of Interview Questions for IBM)

**Answer:**

**select \* from Employee a where  rowid = (select max(rowid) from Employee b where  a.Employee\_no=b.Employee\_no);**

[CLICK HERE TO GET 20 COMPLEX INTERVIEW QUESTIONS..](http://www.complexsql.com/20-sql-queries-for-interview/)

6.What is query to display first 50% records from table?

**Answer:**

**Select rownum,E.\* from Employee E**

**minus**

**Select rownum,E.\* from Employee E where rownum<=(Select count(\*/2) from Employee);**

7.What is subquery?

**Answer:**

Subquery is query within query.The output of outer query is assigned to the column which is used in where condition of outer query.The subquery output is returning only one output value and based on that output value the outer query is executed.Subqueries are used in various real life scenarios like report development,Application logic development,Performance tuning of query.

Example:

Select \* from Employee where dept\_no In (Select dept\_no from department where department\_name=’Oracle’);

8.What is Correlated Subquery.

**Answer:**

Correlated Query is nothing but the subquery whose output is depending on the inner query used in that query.Correlated query is the query which is executed after the outer query is executed.The outer query is always dependent on inner query.The approach of the correlated subquery is bit different than normal subqueries.In normal subqueries the inner queries are executed first and then the outer query is executed but in Correlated Subquery outer query is always dependent on inner query so first outer query is executed then inner query is executed.Correlated Subqueries always uses operator like **Exist,Not Exist,IN,Not IN.**

**“Correlated Queries are also called as Synchronized queries…”**

9.Explain Steps in Correlated subquery excecution.

**Answer:**

**Execution Steps of Correlated Subqueries:**

1.Executes the outer Query

2.For Each row of outer query inner subquery is executed once

3.The result of correlated subquery determines whether the fetched row should be the part of our output results

4.The Process is Repeated for all Rows

**“It is not recommended to use Correlated Subqueries as it slows down the performance”**

[CLICK HERE TO GET INFORMATION ON CORRELATED QUERIES](http://www.complexsql.com/subqueries-in-sql-correlated-subquery-real-life-examples-of-subqueries-and-corelated-subqueries/)

10. Explain example of correlated subquery.(Asked 80 % of Interview Questions for IBM)

**Answer:**

Fetch the Employees who have not assigned a single department.

Select \* from Employee E where Not exist

(Select Department\_no From Department D where E.Employee\_id=D.Employee\_ID);

Execution of query:

Step 1:

Select \* from Employee E ;

It will fetch the all employees

Step 2:

The First Record of the Employee second query is executed and output is given to first query.

(Select Department\_no From Department D where E.Employee\_id=D.Employee\_ID);

Step 3:

Step 2 is repeated until and unless all output is been fetched.

11.What is Rank function as aggregate function?

**Answer:**

Rank function is used as aggregate function to return the rank of rows in the table within group of rows.If someone needs to find out the rank of specific row in the table then we will use the rank function.

Rank Function Syntax:

**RANK( expr1 [, expr2, … expr\_n ] ) WITHIN GROUP ( ORDER BY expr1 [, expr\_2, … expr\_n ] );**

[Click here to get Real life Example of Rank Function…](http://www.complexsql.com/rank-function-in-sql/)

12.What is Rank as analytical function in SQL?(Asked 80 % of Interview Questions for IBM)

**Answer:**

Rank function is used as analytical function in SQL/PLSQL/SQL server which is used to give the rank to the specific record in the table.Rank function is giving you ranking in ordered partitions.Means Ties are assigned to the same values after using the order by clause.So Rank function is not useful where same data is repeated again and again.It is useful in Unique data where user can make partition and order  the data properly.

Syntax of Rank:

**RANK () OVER (PARTITION BY expression ORDER BY expression);**

13.What is query to find the record no 15 from database?

**Answer:**

Select \* from ( Select Employee.\*, rownum Rn from Employee) Where Rn=15;

14.What is Rownum in Oracle?

**Answer:**

1. ROWNUM is magical column in Oracle which assigns the sequence number to the rows retreives in the table.
2. To limit the values in the table you can use rownum pseudocolumn
3. ROWNUM is nothing but logical sequence number given to the rows fetched from the table.
4. ROWNUM is logical number assigned temporarily to  the physical location of the row.
5. You can limit the values in the table using rownum
6. ROWNUM is also unique temparary sequence number assigned to that row.

[Click Here to get information on RowID and Rownum..](http://www.complexsql.com/rowid-rownum/)

15.What is faster in following queries?

select \* from Employee;

select employee\_num,Name from Employee;

**Answer:**

Select employee\_num,Name from Employee is faster.

16.Using Distinct keyword is good practice or not by considering the Performance?

**Answer:**

Using distinct keyword is not a good practice because it scans all table to fetch distinct records from the table.

[CLICK HERE TO GET BASIC SQL PERFORMANCE TUNING](http://www.complexsql.com/basics-sql-performance-tuning/)

17.What is difference between Simple view and Complex view?

**Answer:**

1.Simple View-Simple view is view created on single table

2.Complex View-Complex view is view created on more than 1 tables.

[CLICK HERE TO GET MORE INFORMATION ABOUT VIEW AND COMPLEX VIEW](http://www.complexsql.com/views-in-sql/)

18.What is Materialized View?

**Answer:**

Materialized view is also a logical structure which is stored physically on the disc.Like a view in Materialized view we are using simple select statement to create it.You should have **create materialized view** privileges to create a materialized view.Definition of materialized view(called as MV) has been stored in databases.Materialized views are useful in Data-warehousing concepts.

19.Explain Refresh options of Materialized view?

**Answer:**

1.Refresh on commit:

This option commited the data in materialized views immediately after data inserted and commited in table.This option is known as incremental refresh option.View is not fully refreshed with this option

2.Refresh on Demand:

Using this option you can add the condition for refreshing data in materialized views.

You can refresh the data using fast (incremental approach),Complete,Force options.

20.What is Query to Fetch last record from the table?

**Answer:**

Select \* from Employee where Rowid= select max(Rowid) from Employee;

# 50 SQL Query Questions and Answers for Practice

* [SQL Interview](https://www.techbeamers.com/sql-interview/)
* [Meenakshi Agarwal](https://www.techbeamers.com/author/meenakshi/)

If you want to improve SQL skills, then install a SQL package like MySQL and start practicing with it. To get you started, we’ve outlined a few **SQL query questions** in this post.

Solving practice questions is the fastest way to learn any subject. That’s why we’ve selected a set of [**50 SQL queries**](https://www.techbeamers.com/sql-query-questions-answers-for-practice/#sql-queries) that you can use to step up your learning. We’ve also given [**SQL scripts to create the test data**](https://www.techbeamers.com/sql-query-questions-answers-for-practice/#sql-scripts). So, you can use them to create a test database and tables.

Most of the SQL query questions we’ve filtered out of interviews held by top IT MNC like Flipkart and Amazon. So you’ll gain real-time experience by going through them.

Also, we recommend that you first try to form queries by yourself rather than just reading them from the post. Try to find answers on your own.

But you can’t start until the required sample data is not in place. You can check out the tables below that we’ve provided for practice. So first of all, you need to create the test data in your database software.

By the way, we have a bunch of other posts available for SQL interview preparation. So if you are interested, then follow the link given below.

50 SQL Query Questions

### Prepare Sample Data To Practice SQL Skill.

#### Sample Table – Worker

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WORKER\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **SALARY** | **JOINING\_DATE** | **DEPARTMENT** |
| 001 | Monika | Arora | 100000 | 2014-02-20 09:00:00 | HR |
| 002 | Niharika | Verma | 80000 | 2014-06-11 09:00:00 | Admin |
| 003 | Vishal | Singhal | 300000 | 2014-02-20 09:00:00 | HR |
| 004 | Amitabh | Singh | 500000 | 2014-02-20 09:00:00 | Admin |
| 005 | Vivek | Bhati | 500000 | 2014-06-11 09:00:00 | Admin |
| 006 | Vipul | Diwan | 200000 | 2014-06-11 09:00:00 | Account |
| 007 | Satish | Kumar | 75000 | 2014-01-20 09:00:00 | Account |
| 008 | Geetika | Chauhan | 90000 | 2014-04-11 09:00:00 | Admin |

#### Sample Table – Bonus

|  |  |  |
| --- | --- | --- |
| **WORKER\_REF\_ID** | **BONUS\_DATE** | **BONUS\_AMOUNT** |
| 1 | 2016-02-20 00:00:00 | 5000 |
| 2 | 2016-06-11 00:00:00 | 3000 |
| 3 | 2016-02-20 00:00:00 | 4000 |
| 1 | 2016-02-20 00:00:00 | 4500 |
| 2 | 2016-06-11 00:00:00 | 3500 |

#### Sample Table – Title

|  |  |  |
| --- | --- | --- |
| **WORKER\_REF\_ID** | **WORKER\_TITLE** | **AFFECTED\_FROM** |
| 1 | Manager | 2016-02-20 00:00:00 |
| 2 | Executive | 2016-06-11 00:00:00 |
| 8 | Executive | 2016-06-11 00:00:00 |
| 5 | Manager | 2016-06-11 00:00:00 |
| 4 | Asst. Manager | 2016-06-11 00:00:00 |
| 7 | Executive | 2016-06-11 00:00:00 |
| 6 | Lead | 2016-06-11 00:00:00 |
| 3 | Lead | 2016-06-11 00:00:00 |

To prepare the sample data, you can run the following queries in your database query executor or on the SQL command line. We’ve tested them with MySQL Server 5.7 and MySQL Workbench 6.3.8 query browser. You can also download these Softwares and install them to carry on the SQL exercise.

#### SQL Script to Seed Sample Data.

CREATE DATABASE ORG;

SHOW DATABASES;

USE ORG;

CREATE TABLE Worker (

WORKER\_ID INT NOT NULL PRIMARY KEY AUTO\_INCREMENT,

FIRST\_NAME CHAR(25),

LAST\_NAME CHAR(25),

SALARY INT(15),

JOINING\_DATE DATETIME,

DEPARTMENT CHAR(25)

);

INSERT INTO Worker

(WORKER\_ID, FIRST\_NAME, LAST\_NAME, SALARY, JOINING\_DATE, DEPARTMENT) VALUES

(001, 'Monika', 'Arora', 100000, '14-02-20 09.00.00', 'HR'),

(002, 'Niharika', 'Verma', 80000, '14-06-11 09.00.00', 'Admin'),

(003, 'Vishal', 'Singhal', 300000, '14-02-20 09.00.00', 'HR'),

(004, 'Amitabh', 'Singh', 500000, '14-02-20 09.00.00', 'Admin'),

(005, 'Vivek', 'Bhati', 500000, '14-06-11 09.00.00', 'Admin'),

(006, 'Vipul', 'Diwan', 200000, '14-06-11 09.00.00', 'Account'),

(007, 'Satish', 'Kumar', 75000, '14-01-20 09.00.00', 'Account'),

(008, 'Geetika', 'Chauhan', 90000, '14-04-11 09.00.00', 'Admin');

CREATE TABLE Bonus (

WORKER\_REF\_ID INT,

BONUS\_AMOUNT INT(10),

BONUS\_DATE DATETIME,

FOREIGN KEY (WORKER\_REF\_ID)

REFERENCES Worker(WORKER\_ID)

ON DELETE CASCADE

);

INSERT INTO Bonus

(WORKER\_REF\_ID, BONUS\_AMOUNT, BONUS\_DATE) VALUES

(001, 5000, '16-02-20'),

(002, 3000, '16-06-11'),

(003, 4000, '16-02-20'),

(001, 4500, '16-02-20'),

(002, 3500, '16-06-11');

CREATE TABLE Title (

WORKER\_REF\_ID INT,

WORKER\_TITLE CHAR(25),

AFFECTED\_FROM DATETIME,

FOREIGN KEY (WORKER\_REF\_ID)

REFERENCES Worker(WORKER\_ID)

ON DELETE CASCADE

);

INSERT INTO Title

(WORKER\_REF\_ID, WORKER\_TITLE, AFFECTED\_FROM) VALUES

(001, 'Manager', '2016-02-20 00:00:00'),

(002, 'Executive', '2016-06-11 00:00:00'),

(008, 'Executive', '2016-06-11 00:00:00'),

(005, 'Manager', '2016-06-11 00:00:00'),

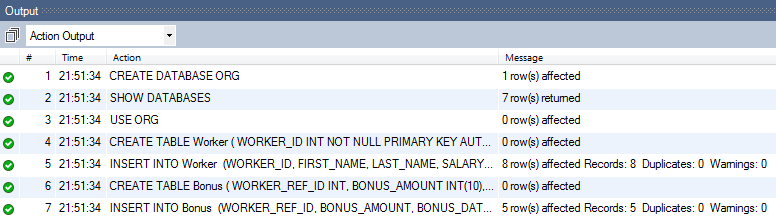
(004, 'Asst. Manager', '2016-06-11 00:00:00'),

(007, 'Executive', '2016-06-11 00:00:00'),

(006, 'Lead', '2016-06-11 00:00:00'),

(003, 'Lead', '2016-06-11 00:00:00');

Once above SQL would run, you’ll see a result similar to the one attached below.

**Creating Sample Data to Practice SQL Skill.**

### 50 SQL Query Questions and Answers for Practice.

#### Q-1. Write an SQL query to fetch “FIRST\_NAME” from Worker table using the alias name as <WORKER\_NAME>.

**Ans.**

The required query is:

Select FIRST\_NAME AS WORKER\_NAME from Worker;

#### Q-2. Write an SQL query to fetch “FIRST\_NAME” from Worker table in upper case.

**Ans.**

The required query is:

Select upper(FIRST\_NAME) from Worker;

#### Q-3. Write an SQL query to fetch unique values of DEPARTMENT from Worker table.

**Ans.**

The required query is:

Select distinct DEPARTMENT from Worker;

#### Q-4. Write an SQL query to print the first three characters of  FIRST\_NAME from Worker table.

**Ans.**

The required query is:

Select substring(FIRST\_NAME,1,3) from Worker;

#### Q-5. Write an SQL query to find the position of the alphabet (‘a’) in the first name column ‘Amitabh’ from Worker table.

**Ans.**

The required query is:

Select INSTR(FIRST\_NAME, BINARY'a') from Worker where FIRST\_NAME = 'Amitabh';

**Notes.**

* The INSTR method is in case-sensitive by default.
* Using Binary operator will make INSTR work as the case-sensitive function.

#### Q-6. Write an SQL query to print the FIRST\_NAME from Worker table after removing white spaces from the right side.

**Ans.**

The required query is:

Select RTRIM(FIRST\_NAME) from Worker;

#### Q-7. Write an SQL query to print the DEPARTMENT from Worker table after removing white spaces from the left side.

**Ans.**

The required query is:

Select LTRIM(DEPARTMENT) from Worker;

#### Q-8. Write an SQL query that fetches the unique values of DEPARTMENT from Worker table and prints its length.

**Ans.**

The required query is:

Select distinct length(DEPARTMENT) from Worker;

#### Q-9. Write an SQL query to print the FIRST\_NAME from Worker table after replacing ‘a’ with ‘A’.

**Ans.**

The required query is:

Select REPLACE(FIRST\_NAME,'a','A') from Worker;

#### Q-10. 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.

**Ans.**

The required query is:

Select CONCAT(FIRST\_NAME, ' ', LAST\_NAME) AS 'COMPLETE\_NAME' from Worker;

#### Q-11. Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending.

**Ans.**

The required query is:

Select \* from Worker order by FIRST\_NAME asc;

#### Q-12. Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending and DEPARTMENT Descending.

**Ans.**

The required query is:

Select \* from Worker order by FIRST\_NAME asc,DEPARTMENT desc;

#### Q-13. Write an SQL query to print details for Workers with the first name as “Vipul” and “Satish” from Worker table.

**Ans.**

The required query is:

Select \* from Worker where FIRST\_NAME in ('Vipul','Satish');

#### Q-14. Write an SQL query to print details of workers excluding first names, “Vipul” and “Satish” from Worker table.

**Ans.**

The required query is:

Select \* from Worker where FIRST\_NAME not in ('Vipul','Satish');

#### Q-15. Write an SQL query to print details of Workers with DEPARTMENT name as “Admin”.

**Ans.**

The required query is:

Select \* from Worker where DEPARTMENT like 'Admin%';

#### Q-16. Write an SQL query to print details of the Workers whose FIRST\_NAME contains ‘a’.

**Ans.**

The required query is:

Select \* from Worker where FIRST\_NAME like '%a%';

#### Q-17. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘a’.

**Ans.**

The required query is:

Select \* from Worker where FIRST\_NAME like '%a';

#### Q-18. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with ‘h’ and contains six alphabets.

**Ans.**

The required query is:

Select \* from Worker where FIRST\_NAME like '\_\_\_\_\_h';

#### Q-19. Write an SQL query to print details of the Workers whose SALARY lies between 100000 and 500000.

**Ans.**

The required query is:

Select \* from Worker where SALARY between 100000 and 500000;

#### Q-20. Write an SQL query to print details of the Workers who have joined in Feb’2014.

**Ans.**

The required query is:

Select \* from Worker where year(JOINING\_DATE) = 2014 and month(JOINING\_DATE) = 2;

#### Q-21. Write an SQL query to fetch the count of employees working in the department ‘Admin’.

**Ans.**

The required query is:

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

#### Q-22. Write an SQL query to fetch worker names with salaries >= 50000 and <= 100000.

**Ans.**

The required query is:

SELECT CONCAT(FIRST\_NAME, ' ', LAST\_NAME) As Worker\_Name, Salary

FROM worker

WHERE WORKER\_ID IN

(SELECT WORKER\_ID FROM worker

WHERE Salary BETWEEN 50000 AND 100000);

#### Q-23. Write an SQL query to fetch the no. of workers for each department in the descending order.

**Ans.**

The required query is:

SELECT DEPARTMENT, count(WORKER\_ID) No\_Of\_Workers

FROM worker

GROUP BY DEPARTMENT

ORDER BY No\_Of\_Workers DESC;

#### Q-24. Write an SQL query to print details of the Workers who are also Managers.

**Ans.**

The required query is:

SELECT DISTINCT W.FIRST\_NAME, T.WORKER\_TITLE

FROM Worker W

INNER JOIN Title T

ON W.WORKER\_ID = T.WORKER\_REF\_ID

AND T.WORKER\_TITLE in ('Manager');

#### Q-25. Write an SQL query to fetch duplicate records having matching data in some fields of a table.

**Ans.**

The required query is:

SELECT WORKER\_TITLE, AFFECTED\_FROM, COUNT(\*)

FROM Title

GROUP BY WORKER\_TITLE, AFFECTED\_FROM

HAVING COUNT(\*) > 1;

#### Q-26. Write an SQL query to show only odd rows from a table.

**Ans.**

The required query is:

SELECT \* FROM Worker WHERE MOD (WORKER\_ID, 2) <> 0;

#### Q-27. Write an SQL query to show only even rows from a table.

**Ans.**

The required query is:

SELECT \* FROM Worker WHERE MOD (WORKER\_ID, 2) = 0;

#### Q-28. Write an SQL query to clone a new table from another table.

**Ans.**

The general query to clone a table with data is:

SELECT \* INTO WorkerClone FROM Worker;

The general way to clone a table without information is:

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

An alternate way to clone a table (for MySQL) without is:

CREATE TABLE WorkerClone LIKE Worker;

#### Q-29. Write an SQL query to fetch intersecting records of two tables.

**Ans.**

The required query is:

(SELECT \* FROM Worker)

INTERSECT

(SELECT \* FROM WorkerClone);

#### Q-30. Write an SQL query to show records from one table that another table does not have.

**Ans.**

The required query is:

SELECT \* FROM Worker

MINUS

SELECT \* FROM Title;

#### Q-31. Write an SQL query to show the current date and time.

**Ans.**

Following MySQL query returns the current date:

SELECT CURDATE();

Following MySQL query returns the current date and time:

SELECT NOW();

Following SQL Server query returns the current date and time:

SELECT getdate();

Following Oracle query returns the current date and time:

SELECT SYSDATE FROM DUAL;

#### Q-32. Write an SQL query to show the top n (say 10) records of a table.

**Ans.**

Following MySQL query will return the top n records using the LIMIT method:

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

Following SQL Server query will return the top n records using the TOP command:

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

Following Oracle query will return the top n records with the help of ROWNUM:

SELECT \* FROM (SELECT \* FROM Worker ORDER BY Salary DESC)

WHERE ROWNUM <= 10;

#### Q-33. Write an SQL query to determine the nth (say n=5) highest salary from a table.

**Ans.**

The following MySQL query returns the nth highest salary:

SELECT Salary FROM Worker ORDER BY Salary DESC LIMIT n-1,1;

The following SQL Server query returns the nth highest salary:

SELECT TOP 1 Salary

FROM (

SELECT DISTINCT TOP n Salary

FROM Worker

ORDER BY Salary DESC

)

ORDER BY Salary ASC;

#### Q-34. Write an SQL query to determine the 5th highest salary without using TOP or limit method.

**Ans.**

The following query is using the correlated subquery to return the 5th highest salary:

SELECT Salary

FROM Worker W1

WHERE 4 = (

SELECT COUNT( DISTINCT ( W2.Salary ) )

FROM Worker W2

WHERE W2.Salary >= W1.Salary

);

Use the following generic method to find nth highest salary without using TOP or limit.

SELECT Salary

FROM Worker W1

WHERE n-1 = (

SELECT COUNT( DISTINCT ( W2.Salary ) )

FROM Worker W2

WHERE W2.Salary >= W1.Salary

);

#### Q-35. Write an SQL query to fetch the list of employees with the same salary.

**Ans.**

The required query is:

Select distinct W.WORKER\_ID, W.FIRST\_NAME, W.Salary

from Worker W, Worker W1

where W.Salary = W1.Salary

and W.WORKER\_ID != W1.WORKER\_ID;

#### Q-36. Write an SQL query to show the second highest salary from a table.

**Ans.**

The required query is:

Select max(Salary) from Worker

where Salary not in (Select max(Salary) from Worker);

#### Q-37. Write an SQL query to show one row twice in results from a table.

**Ans.**

The required query is:

select FIRST\_NAME, DEPARTMENT from worker W where W.DEPARTMENT='HR'

union all

select FIRST\_NAME, DEPARTMENT from Worker W1 where W1.DEPARTMENT='HR';

#### Q-38. Write an SQL query to fetch intersecting records of two tables.

**Ans.**

The required query is:

(SELECT \* FROM Worker)

INTERSECT

(SELECT \* FROM WorkerClone);

#### Q-39. Write an SQL query to fetch the first 50% records from a table.

**Ans.**

The required query is:

SELECT \*

FROM WORKER

WHERE WORKER\_ID <= (SELECT count(WORKER\_ID)/2 from Worker);

#### Q-40. Write an SQL query to fetch the departments that have less than five people in it.

**Ans.**

The required query is:

SELECT DEPARTMENT, COUNT(WORKER\_ID) as 'Number of Workers' FROM Worker GROUP BY DEPARTMENT HAVING COUNT(WORKER\_ID) < 5;

#### Q-41. Write an SQL query to show all departments along with the number of people in there.

**Ans.**

The following query returns the expected result:

SELECT DEPARTMENT, COUNT(DEPARTMENT) as 'Number of Workers' FROM Worker GROUP BY DEPARTMENT;

#### Q-42. Write an SQL query to show the last record from a table.

**Ans.**

The following query will return the last record from the Worker table:

Select \* from Worker where WORKER\_ID = (SELECT max(WORKER\_ID) from Worker);

#### Q-43. Write an SQL query to fetch the first row of a table.

**Ans.**

The required query is:

Select \* from Worker where WORKER\_ID = (SELECT min(WORKER\_ID) from Worker);

#### Q-44. Write an SQL query to fetch the last five records from a table.

**Ans.**

The required query is:

SELECT \* FROM Worker WHERE WORKER\_ID <=5

UNION

SELECT \* FROM (SELECT \* FROM Worker W order by W.WORKER\_ID DESC) AS W1 WHERE W1.WORKER\_ID <=5;

#### Q-45. Write an SQL query to print the name of employees having the highest salary in each department.

**Ans.**

The required query is:

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;

#### Q-46. Write an SQL query to fetch three max salaries from a table.

**Ans.**

The required query is:

SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;

#### Q-47. Write an SQL query to fetch three min salaries from a table.

**Ans.**

The required query is:

SELECT distinct Salary from worker a WHERE 3 >= (SELECT count(distinct Salary) from worker b WHERE a.Salary >= b.Salary) order by a.Salary desc;

#### Q-48. Write an SQL query to fetch nth max salaries from a table.

**Ans.**

The required query is:

SELECT distinct Salary from worker a WHERE n >= (SELECT count(distinct Salary) from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;

#### Q-49. Write an SQL query to fetch departments along with the total salaries paid for each of them.

**Ans.**

The required query is:

 SELECT DEPARTMENT, sum(Salary) from worker group by DEPARTMENT;

#### Q-50. Write an SQL query to fetch the names of workers who earn the highest salary.

**Ans.**

The required query is:

SELECT FIRST\_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);

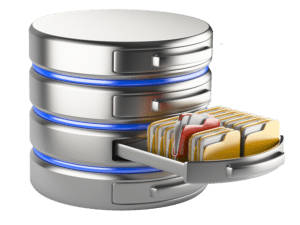
## **Q1. What is the difference between SQL and MySQL?**

|  |  |
| --- | --- |
| **SQL vs MySQL** | |
| **SQL** | **MySQL** |
| SQL is a standard language which stands for Structured Query Language based on the English language | MySQL is a database management system. |
| SQL is the core of the relational database which is used for accessing and managing database | MySQL is an RDMS (Relational Database Management System) such as SQL Server, Informix etc. |

## **Q2. What are the different subsets of SQL?**

* 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.

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

A database is a structured collection of data.

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: The data is stored in relations (tables). Example – MySQL.
* Non-Relational Database Management System: There is no concept of relations, tuples and attributes.  Example – Mongo

## **Q4. What do you mean by table and field in SQL?**

A table refers to a collection of data in an organised manner in form of rows and columns. A field refers to the number of columns in a table. For example:

***Table***: StudentInformation  
***Field***: Stu Id, Stu Name, Stu Marks

## **Q5. What are joins in SQL?**

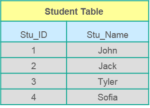
A JOIN clause is used to combine rows from two or more tables, based on a related column between them. It is used to merge two tables or retrieve data from there. There are 4 joins in SQL namely:

* Inner Join
* Right Join
* Left Join
* Full Join

## **Q6. What is the difference between CHAR and VARCHAR2 datatype in SQL?**

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

## **Q7. What is a Primary key?**

* APrimary keyis a column (or collection of columns) or a set of columns that uniquely identifies each row in the table.
* Uniquely identifies a single row in the table
* Null values not allowed

Example- In the Student table, Stu\_ID is the primary key.

## **Q8. What are Constraints?**

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

* NOT NULL
* CHECK
* DEFAULT
* UNIQUE
* PRIMARY KEY
* FOREIGN KEY

## **Q9. What is the difference between DELETE and TRUNCATE statements?**

|  |  |
| --- | --- |
| **DELETE vs TRUNCATE** | |
| **DELETE** | **TRUNCATE** |
| Delete command is used to delete a row in a table. | Truncate is used to delete all the rows from a table. |
| You can rollback data after using delete statement. | You cannot rollback data. |
| It is a DML command. | It is a DDL command. |
| It is slower than truncate statement. | It is faster. |

## **Q10. What is a Unique key?**

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

Apart from this SQL Interview Questions blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

## **Q11. What is a Foreign key?**

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

## **Q12. 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.

## **Q13. What is the difference between clustered and non clustered index in SQL?**

The differences between the clustered and non clustered index in SQL are :

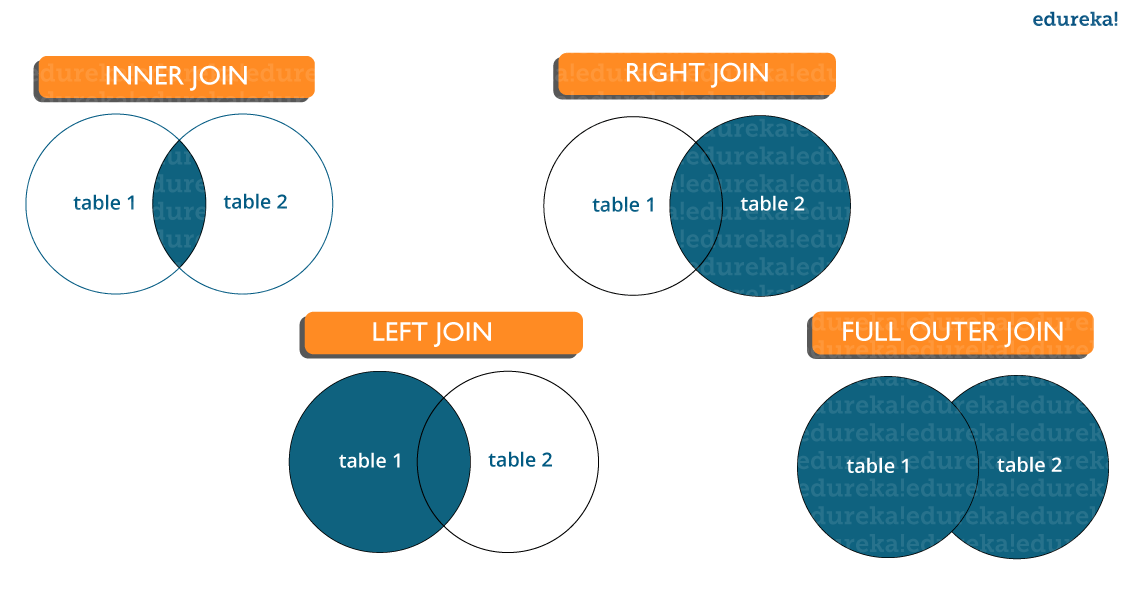
1. Clustered index is used for easy retrieval of data from the database and its faster whereas reading from non clustered index is relatively slower.
2. Clustered index alters the way records are stored in a database as it sorts out rows by the column which is set to be clustered index whereas in a non clustered index, it does not alter the way it was stored but it creates a separate object within a table which points back to the original table rows after searching.
3. One table can only have one clustered index whereas it can have many non clustered index.

## **Q14. Write a SQL query to display the current date?**

In SQL, there is a built-in function called GetDate() which helps to return the current timestamp/date.

## **Q15. List the different type of joins?**

There are various types of joins which are used to retrieve data between the tables. There are four types of joins, namely:

**Inner join:** Inner Join in MySQL 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 MySQL 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 MySQL 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.

### [80% INTERVIEW REJECTIONS HAPPEN IN FIRST 90 SECONDSBOOK A SLOT](https://www.edureka.co/mymock-interview-service" \t "_blank)

**Q16. What do you mean by Denormalization?**

Denormalization refers to a technique which is used to access data from higher to lower forms of a database. It helps the database managers to increase the performance of the entire infrastructure as it introduces redundancy into a table. It adds the redundant data into a table by incorporating database queries that combine data from various tables into a single table.

## **Q17. 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. 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 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).

## **Q18. What is an 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.

## **Q19. Explain different types of index.**

There are three types of index namely:

### ****Unique Index:****

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

### ****Clustered Index:****

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

#### ****Non-Clustered Index:****

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

## **Q20. What is Normalization and what are the advantages of it?**

Normalization is the process of organizing data to avoid duplication and redundancy. Some of the advantages 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

Apart from this SQL Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

## **Q21. What is the difference between DROP and TRUNCATE commands?**

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

## **Q22. Explain different types of Normalization.**

There are many successive levels of normalization. These are called **normal forms**. Each consecutive normal form depends on the previous one.The first three normal forms are usually adequate.

* First Normal Form (1NF) – No repeating groups within rows
* Second Normal Form (2NF) – Every non-key (supporting) column value is dependent on the whole primary key.
* Third Normal Form (3NF) – Dependent solely on the primary key and no other non-key (supporting) column value.

## **Q23. What is ACID property in a database?**

ACID stands for Atomicity, Consistency, Isolation, Durability. It is used to ensure that the data transactions are processed reliably in a database system.

**Atomicity:** Atomicity refers to the transactions that are completely done or failed where transaction refers to a single logical operation of a data. It means if one part of any transaction fails, the entire transaction fails and the database state is left unchanged.

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

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

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

## **Q24. What do you mean by “Trigger” in SQL?**

Trigger in SQL is are a special type of stored procedures that are 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.

## **Q25. What are the different operators available in SQL?**

There are three operators available in SQL, namely:

1. Arithmetic Operators
2. Logical Operators
3. Comparison Operators

Apart from this SQL Interview Questions blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

## **Q26.  Are NULL values same as that of zero or a blank space?**

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

## **Q27. What is the difference between cross join and natural join?**

The cross join produces the cross product or Cartesian product of two tables whereas the natural join is based on all the columns having the same name and data types in both the tables.

## **Q28. What is subquery in SQL?**

A subquery is a query inside another query where a query is defined to retrieve data or information back from the database. In a subquery, the outer query is called as the main query whereas the inner query is called subquery. Subqueries are always executed first and the result of the subquery is passed on to the main query. It can be nested inside a SELECT, UPDATE or any other query. A subquery can also use any comparison operators such as >,< or =.

## **Q29. What are the different types of a subquery?**

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

**Correlated subquery**: These are queries which 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 the column in a table.

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

## **Q30. List the ways to get the count of records in a table?**

To count the number of records in a table, you can use the below commands:

SELECT \* FROM table1

SELECT COUNT(\*) FROM table1

SELECT rows FROM sysindexes WHERE id = OBJECT\_ID(table1) AND indid < 2

Apart from this SQL Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

## **Q31. Write a SQL query to find the names of employees that begin with ‘A’?**

To display name of the employees that begin with ‘A’, type in the below command:

|  |  |
| --- | --- |
| 1 | SELECT \* FROM Table\_name WHERE EmpName like 'A%' |

## **Q32. Write a SQL query to get the third highest salary of an employee from employee\_table?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | SELECT TOP 1 salary  FROM(  SELECT TOP 3 salary  FROM employee\_table  ORDER BY salary DESC) AS emp  ORDER BY salary ASC; |

**Q33. What is the need for group functions in SQL?**

Group functions work on the set of rows and returns one result per group. Some of the commonly used group functions are: AVG, COUNT, MAX, MIN, SUM, VARIANCE.

## **Q34 . What is a Relationship and what are they?**

Relation or links are between entities that have something to do with each other. Relationships are defined as the connection between the tables in a database. There are various relationships, namely:

* One to One Relationship.
* One to Many Relationship.
* Many to One Relationship.
* Self-Referencing Relationship.

## **Q35.  How can you insert NULL values in a column while inserting the data?**

NULL values can be inserted in the following ways:

* Implicitly by omitting column from column list.
* Explicitly by specifying NULL keyword in the VALUES clause

## **Q36. What is the main difference between ‘BETWEEN’ and ‘IN’ condition operators?**

BETWEEN operator is used to display rows based on a range of values in a row whereas the IN condition operator is used to check for values contained in a specific set of values.

### Example of BETWEEN:

SELECT \* FROM Students where ROLL\_NO BETWEEN 10 AND 50;

**Example of IN:**

SELECT \* FROM students where ROLL\_NO IN (8,15,25);

**Q37. Why are SQL functions used?**

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

## **Q38. What is the need of 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.

## **Q39.** What do you mean by recursive stored procedure?

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

## **Q40. 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.

Apart from this SQL Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

## **Q41. What is the difference between ‘HAVING’ CLAUSE and a ‘WHERE’ CLAUSE?**

HAVING clause can be used only with SELECT statement. It is usually used in a GROUP BY clause and whenever GROUP BY is not used, HAVING behaves like a WHERE clause.  
Having Clause is only used with the GROUP BY function in a query whereas WHERE Clause is applied to each row before they are a part of the GROUP BY function in a query.

## **Q42. List the ways in which  Dynamic SQL can be executed?**

Following are the ways in which dynamic SQL can be executed:

* Write a query with parameters.
* Using EXEC.
* Using sp\_executesql.

## **Q43. What are the various levels of constraints?**

Constraints are the representation of a column to enforce data entity and consistency. There are two levels  of a constraint, namely:

* column level constraint
* table level constraint

## **Q44. How can you fetch common records from two tables?**

You can fetch common records from two tables using INTERSECT. For example:

|  |  |
| --- | --- |
| 1 | Select studentID from student. <strong>INTERSECT </strong> Select StudentID from Exam |

## **Q45. 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’)

Apart from this SQL Interview Questions blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

## **Q46. What are the different set operators available in SQL?**

Some of the available set operators are – Union, Intersect or Minus operators.

## **Q47. What is an ALIAS command?**

ALIAS name can be given to any table or a column. This alias name can be referred in WHERE clause to identify a particular table or a column.

For example-

Select emp.empID, dept.Result from employee emp, department as dept where emp.empID=dept.empID

In the above example, emp refers to alias name for employee table and dept refers to alias name for department table.

## **Q48. What are aggregate and scalar functions?**

Aggregate functions are used to evaluate mathematical calculation and returns a single value. These calculations are done from the columns in a table. For example- max(),count() are calculated with respect to numeric.

Scalar functions return a single value based on the input value. For example – UCASE(), NOW() are calculated with respect to string.

## **Q49. How can you fetch alternate records from a table?**

You can fetch alternate records i.e both odd and even row numbers. For example- To display even numbers, use the following command:

Select studentId from (Select rowno, studentId from student) where mod(rowno,2)=0

Now, to display odd numbers:

Select studentId from (Select rowno, studentId from student) where mod(rowno,2)=1

**Q50. Name the operator which is used in the query for pattern matching?**

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

1. % – It matches zero or more characters.

For example- select \* from students where studentname like ‘a%’

\_ (Underscore) – it matches exactly one character.  
For example- select \* from student where studentname like ‘abc\_’

Apart from this SQL Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka! Click below to know more.

## **Q51. How can you select unique records from a table?**

You can select unique records from a table by using the DISTINCT keyword.

Select DISTINCT studentID from Student

Using this command, it will print unique student id from the table Student.

*[Powered by Edureka](https://www.edureka.co/mymock-interview-service" \t "_blank)*[[](https://www.edureka.co/mymock-interview-service)](https://www.edureka.co/mymock-interview-service" \t "_blank)

### [NEED HELP FOR YOUR UPCOMING INTERVIEW?](https://www.edureka.co/mymock-interview-service" \t "_blank)

**[Take MYSQL Mock Interview](https://www.edureka.co/mymock-interview-service" \t "_blank)**

* **[Get Interviewed by Industry Experts](https://www.edureka.co/mymock-interview-service" \t "_blank)**
* **[Personalized interview feedback](https://www.edureka.co/mymock-interview-service" \t "_blank)**

[BOOK A SLOT](https://www.edureka.co/mymock-interview-service" \t "_blank)

## **Q52. How can you fetch first 5 characters of the string?**

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

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

## **Q53**. What is the main difference between SQL and PL/SQL?

SQL is a query language that allows you to issue a single query or execute a single insert/update/delete whereas PL/SQL is Oracle’s “Procedural Language” SQL, which allows you to write a full program (loops, variables, etc.) to accomplish multiple operations such as selects/inserts/updates/deletes.

## **Q54. What is a View?**

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

## **Q55. What are Views used for?**

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 same data.

## **Q56. What is a Stored Procedure?**

A Stored Procedure is a function which consists of many SQL statements to access the database system. Several SQL statements are consolidated into a stored procedure and execute them whenever and wherever required which saves time and avoid writing code again and again.

## **Q57. List some advantages and disadvantages of Stored Procedure?**

### ****Advantages****:

A Stored Procedure can be used as a modular programming which means create once, store and call for several times whenever it is required. This supports faster execution. It also reduces network traffic and provides better security to the data.

### ****Disadvantage****:

The only disadvantage of Stored Procedure is that it can be executed only in the database and utilizes more memory in the database server.

## **Q58. List all the types of user-defined functions?**

There are three types of user-defined functions, namely:

* Scalar Functions
* Inline Table-valued functions
* Multi-statement valued functions

Scalar returns the unit, variant defined the return clause. Other two types of defined functions return table.

## **Q59. What do you mean by Collation?**

Collation is defined as a set of rules that determine how data can be sorted as well as compared. Character data is sorted using the rules that define the correct character sequence along with options for specifying case-sensitivity, character width etc.

## **Q60. What are the different types of Collation Sensitivity?**

Following are the different types of collation sensitivity:

* Case Sensitivity: A and a and B and b.
* Kana Sensitivity: Japanese Kana characters.
* Width Sensitivity: Single byte character and double-byte character.
* Accent Sensitivity.

Apart from this SQL Interview Questions Blog, if you want to get trained from professionals on this technology, you can opt for a structured training from edureka!

**Q61. What are Local and Global variables?**

### ****Local variables:****

These variables can be used or exist only inside the function. These variables are not used or referred by any other function.

### ****Global variables:****

These variables are the variables which can be accessed throughout the program. Global variables cannot be created whenever that function is called.

## **Q62. What is Auto Increment in SQL?**

Auto increment keyword allows the user to create a unique number to get generated whenever a new record is inserted into the table.  
This keyword is usually required whenever PRIMARY KEY is used.

AUTO INCREMENT keyword can be used in Oracle and IDENTITY keyword can be used in SQL SERVER.

## **Q63. What is a Datawarehouse?**

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

## **Q64. What are the different authentication modes in SQL Server? How can it be** changed?

Windows mode and Mixed Mode – SQL and Windows. You can go to the below steps to change authentication mode in SQL Server:

* Click Start> Programs> Microsoft SQL Server and click SQL Enterprise Manager to run SQL Enterprise Manager from the Microsoft SQL Server program group.
* Then select the server from the Tools menu.
* Select SQL Server Configuration Properties, and choose the Security page.

## **Q65. What are STUFF and REPLACE function?**

**STUFF Function**: This function is used to overwrite existing character or inserts a string into another string. Syntax:

STUFF(string\_expression,start, length, replacement\_characters)

where,  
string\_expression: it is the string that will have characters substituted

start: This refers to the starting position  
length: It refers to the number of characters in the string which are substituted.

replacement\_string: They are the new characters which are injected in the string.

**REPLACE function**: This function is used to replace the existing characters of all the occurrences. Syntax:

REPLACE (string\_expression, search\_string, replacement\_string)

Here every search\_string in the string\_expression will be replaced with the replacement\_string.

So this brings us to the end of the SQL interview questions blog. I hope this set of SQL Interview Questions will help you ace your job interview. **All the best for your interview!**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2019/06/Top-Interview-Questions-And-Answers-sql.png)

### Best SQL Interview Questions

**Q #1) What is SQL?**

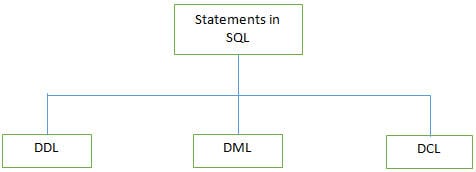
**Answer:** Structured Query Language SQL is a database tool that is used to create and access the database to support software applications.

**Q #2) What are tables in SQL?**

**Answer:**The table is a collection of record and its information at a single view.

**Q #3) What are the different types of statements supported by SQL?**

**Answer:**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/statements-supported-by-SQL.jpg)

**There are 3 types of SQL statements:**

**a) DDL (Data Definition Language):** It is used to define the database structure such as tables. It includes three statements such as CREATE, ALTER, and DROP.

**Some of the DDL Commands are listed below:**

**CREATE**: It is used for creating the table.

|  |
| --- |
| CREATE TABLE table\_name  column\_name1 data\_type(size),  column\_name2 data\_type(size),  column\_name3 data\_type(size), |

**ALTER:** The ALTER table is used for modifying the existing table object in the database.

|  |
| --- |
| ALTER TABLE table\_name   ADD column\_name datatype |

OR

|  |
| --- |
| ALTER TABLE table\_name  DROP COLUMN column\_name |

**b) DML (Data Manipulation Language):** These statements are used to manipulate the data in records. Commonly used DML statements are INSERT, UPDATE, and DELETE.

The SELECT statement is used as a partial DML statement, used to select all or relevant records in the table.

**c) DCL (Data Control Language):**These statements are used to set privileges such as GRANT and REVOKE database access permission to the specific user**.**

**Q #4) How do we use the DISTINCT statement? What is its use?**

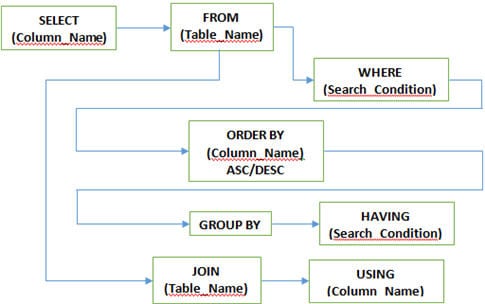
**Answer:**The DISTINCT statement is used with the SELECT statement. If the record contains duplicate values then the DISTINCT statement is used to select different values among duplicate records.

**Syntax:**

|  |
| --- |
| SELECT DISTINCT column\_name(s)   FROM table\_name; |

**Q #5) What are different Clauses used in SQL?**

**Answer:**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Clauses-used-in-SQL.jpg)

**WHERE Clause:**This clause is used to define the condition, extract and display only those records which fulfill the given condition.

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)   FROM table\_name   WHERE condition; |

**GROUP BY Clause:** It is used with SELECT statement to group the result of the executed query using the value specified in it. It matches the value with the column name in tables and groups the end result accordingly.

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)   FROM table\_name   GROUP BY column\_name; |

**HAVING clause:**This clause is used in association with the GROUP BY clause. It is applied to each group of results or the entire result as a single group. It is much similar as WHERE clause but the only difference is you cannot use it without GROUP BY clause

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)   FROM table\_name   GROUP BY column\_name   HAVING condition; |

**ORDER BY clause:**This clause is used to define the order of the query output either in ascending (ASC) or in descending (DESC). Ascending (ASC) is set as the default one but descending (DESC) is set explicitly.

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)   FROM table\_name   WHERE condition   ORDER BY column\_name ASC|DESC; |

**USING clause:**USING clause comes in use while working with SQL JOIN. It is used to check equality based on columns when tables are joined. It can be used instead of the ON clause in JOIN.

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)   FROM table\_name   JOIN table\_name   USING (column\_name); |

**Q #6) Why do we use SQL constraints? Which constraints we can use while creating a database in SQL?**

**Answer:**Constraints are used to set the rules for all records in the table. If any constraints get violated then it can abort the action that caused it.

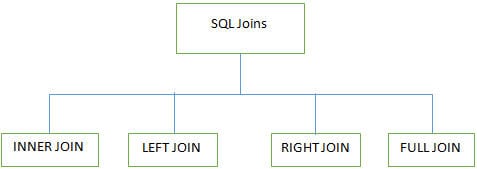
Constraints are defined while creating the database itself with the CREATE TABLE statement or even after the table is created once with the ALTER TABLE statement.

**There are 5 major constraints are used in SQL, such as**

* **NOT NULL:** That indicates that the column must have some value and cannot be left NULL.
* **UNIQUE:** This constraint is used to ensure that each row and column has a unique value and no value is being repeated in any other row or column.
* **PRIMARY KEY:** This constraint is used in association with NOT NULL and UNIQUE constraints such as on one or the combination of more than one column to identify the particular record with a unique identity.
* **FOREIGN KEY:** It is used to ensure the referential integrity of data in the table. It matches the value in one table with another using the PRIMARY KEY.
* **CHECK:** It ensures whether the value in columns fulfills the specified condition.

**Q #7) What are different JOINS used in SQL?**

**Answer:**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/SQL-Joins.jpg)

4 major types of Joins are used while working on multiple tables in SQL databases:

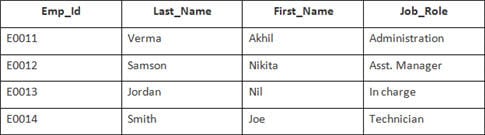
**INNER JOIN:**It is also known as SIMPLE JOIN which returns all rows from BOTH tables when it has at least one matching column.

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)   FROM table\_name1   INNER JOIN table\_name2   ON column\_name1=column\_name2; |

**For Example,**

In this example, we have a table **Employee** with the following data:

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Employee-table-1.jpg)

The second table's name is**Joining.**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/joining.jpg)

**Enter the following SQL statement:**

|  |
| --- |
| SELECT Employee.Emp\_id, Joining.Joining\_Date    FROM Employee    INNER JOIN Joining    ON Employee.Emp\_id = Joining.Emp\_id    ORDER BY Employee.Emp\_id; |

There will be 4 records selected. **Results are:**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/result-of-innerjoin.jpg)

**Employee** and **Orders** tables have a matching customer\_id value.

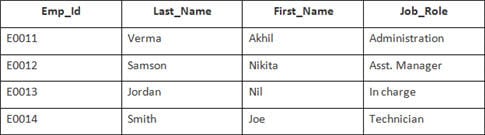
**LEFT JOIN (LEFT OUTER JOIN):**This join returns all rows from the LEFT table and its matched rows from a RIGHT table**.**

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)   FROM table\_name1   LEFT JOIN table\_name2   ON column\_name1=column\_name2; |

**For Example,**

In this example, we have a table **Employee** with the following data:

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Employee-table-1.jpg)

The second table's name is**Joining.**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/joining-1.jpg)

**Enter the following SQL statement:**

|  |
| --- |
| SELECT Employee.Emp\_id, Joining.Joining\_Date  FROM Employee  LEFT OUTER JOIN Joining  ON Employee.Emp\_id = Joining.Emp\_id  ORDER BY Employee.Emp\_id; |

There will be 4 records selected. **You will see the following results:**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/result-of-LEFT-OUTER-JOIN.jpg)

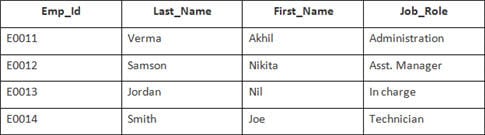
**RIGHT JOIN (RIGHT OUTER JOIN):**This joins returns all rows from the RIGHT table and its matched rows from the LEFT table**.**

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)  FROM table\_name1  RIGHT JOIN table\_name2  ON column\_name1=column\_name2; |

**For Example,**

In this example, we have a table **Employee** with the following data:

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Employee-table-1.jpg)

The second table's name is**Joining.**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/joining-1.jpg)

**Enter the following SQL statement:**

|  |
| --- |
| SELECT Employee.Emp\_id, Joining.Joining\_Date FROM Employee  RIGHT JOIN Joining  ON Employee.Emp\_id = Joining.Emp\_id  ORDER BY Employee.Emp\_id; |

**Output:**

| **Emp\_id** | **Joining\_Date** |
| --- | --- |
| E0012 | 2016/04/18 |
| E0013 | 2016/04/19 |
| E0014 | 2016/05/01 |

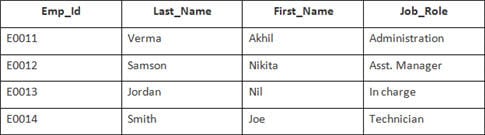
**FULL JOIN (FULL OUTER JOIN):**This joins returns all results when there is a match either in the RIGHT table or in the LEFT table**.**

**Syntax:**

|  |
| --- |
| SELECT column\_name(s)   FROM table\_name1   FULL OUTER JOIN table\_name2   ON column\_name1=column\_name2; |

**For Example,**

In this example, we have a table **Employee** with the following data:

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/Employee-table-1.jpg)

The second table's name is**Joining.**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/joining-1.jpg)

**Enter the following SQL statement:**

|  |
| --- |
| SELECT Employee.Emp\_id, Joining.Joining\_Date  FROM Employee  FULL OUTER JOIN Joining  ON Employee.Emp\_id = Joining.Emp\_id  ORDER BY Employee.Emp\_id; |

There will be 8 records selected. **These are the results that you should see.**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2016/07/result-of-FULL-OUTER-JOIN.jpg)

**Q #8) What are transactions and their controls?**

**Answer:**A transaction can be defined as the sequence task that is performed on databases in a logical manner to gain certain results. Operations like Creating, updating, deleting records performed in the database come from transactions.

In simple words, we can say that a transaction means a group of SQL queries executed on database records.

**There are 4 transaction controls such as**

* **COMMIT**: It is used to save all changes made through the transaction.
* **ROLLBACK**: It is used to roll back the transaction. All changes made by the transaction are reverted back and the database remains as before.
* **SET TRANSACTION**: Set the name of the transaction.
* **SAVEPOINT:**It is used to set the point where the transaction is to be rolled back.

**Q #9) What are the properties of the transaction?**

**Answer: Properties of the transaction are known as ACID properties. These are:**

* **Atomicity**: Ensures the completeness of all transactions performed. Checks whether every transaction is completed successfully or not. If not, then the transaction is aborted at the failure point and the previous transaction is rolled back to its initial state as changes are undone.
* **Consistency**: Ensures that all changes made through successful transactions are reflected properly on the database.
* **Isolation**: Ensures that all transactions are performed independently and changes made by one transaction are not reflected on others.
* **Durability**: Ensures that the changes made in the database with committed transactions persist as it is even after a system failure.

**Q #10) How many Aggregate functions are available in SQL?**

**Answer:**SQL Aggregate functions determine and calculate values from multiple columns in a table and return a single value.

**There are 7 aggregate functions in SQL:**

* **AVG():** Returns the average value from specified columns.
* **COUNT():** Returns number of table rows.
* **MAX():** Returns the largest value among the records.
* **MIN():** Returns smallest value among the records.
* **SUM():** Returns the sum of specified column values.
* **FIRST():** Returns the first value.
* **LAST():** Returns last value.

**Q #11) What are Scalar functions in SQL?**

**Answer:**Scalar functions are used to return a single value based on the input values.

**Scalar Functions are as follows:**

* **UCASE():**Converts the specified field in the upper case.
* **LCASE():**Converts the specified field in lower case.
* **MID():**Extracts and returns character from the text field.
* **FORMAT():**Specifies the display format.
* **LEN():**Specifies the length of the text field.
* **ROUND():**Rounds up the decimal field value to a number.

**Q #12) What are triggers**?

**Answer:**Triggers in SQL is kind of stored procedures used to create a response to a specific action performed on the table such as INSERT, UPDATE or DELETE. You can invoke triggers explicitly on the table in the database.

Action and Event are two main components of SQL triggers. When certain actions are performed, the event occurs in response to that action.

**Syntax:**

|  |
| --- |
| CREATE TRIGGER name {BEFORE|AFTER} (event [OR..]}  ON table\_name [FOR [EACH] {ROW|STATEMENT}]  EXECUTE PROCEDURE functionname {arguments} |

**Q #13) What is View in SQL?**

**Answer:**A View can be defined as a virtual table that contains rows and columns with fields from one or more tables.

**Syntax:**

|  |
| --- |
| CREATE VIEW view\_name AS  SELECT column\_name(s)  FROM table\_name  WHERE condition |

**Q #14) How we can update the view?**

**Answer:**SQL CREATE and REPLACE can be used for updating the view.

Execute the below query to update the created view.

**Syntax:**

|  |
| --- |
| CREATE OR REPLACE VIEW view\_name AS   SELECT column\_name(s)   FROM table\_name   WHERE condition |

**Q #15) Explain the working of SQL Privileges?**

**Answer:**SQL GRANT and REVOKE commands are used to implement privileges in SQL multiple user environments. The administrator of the database can grant or revoke privileges to or from users of database objects by using commands like SELECT, INSERT, UPDATE, DELETE, ALL, etc.

**GRANT Command**: This command is used to provide database access to users other than the administrator.

**Syntax:**

|  |
| --- |
| GRANT privilege\_name   ON object\_name   TO {user\_name|PUBLIC|role\_name}   [WITH GRANT OPTION]; |

In the above syntax, the GRANT option indicates that the user can grant access to another user too.

**REVOKE command**: This command is used to provide database deny or remove access to database objects.

**Syntax:**

|  |
| --- |
| REVOKE privilege\_name   ON object\_name   FROM {user\_name|PUBLIC|role\_name}; |

**Q #16) How many types of Privileges are available in SQL?**

**Answer: There are two types of privileges used in SQL, such as**

* **System privilege:**System privilege deals with the object of a particular type and provides users the right to perform one or more actions on it. These actions include performing administrative tasks, ALTER ANY INDEX, ALTER ANY CACHE GROUP CREATE/ALTER/DELETE TABLE, CREATE/ALTER/DELETE VIEW etc.
* **Object privilege:**This allows to perform actions on an object or object of another user(s) viz. table, view, indexes etc. Some of the object privileges are EXECUTE, INSERT, UPDATE, DELETE, SELECT, FLUSH, LOAD, INDEX, REFERENCES etc.

**Q #17) What is SQL Injection?**

**Answer:**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.

**For Example,**

|  |
| --- |
| SELECT column\_name(s) FROM table\_name WHERE condition; |

**Q #18) What is SQL Sandbox in SQL Server?**

**Answer:**SQL Sandbox is a safe place in the SQL server environment where untrusted scripts are executed. There are 3 types of SQL sandbox:

* **Safe Access Sandbox:** Here a user can perform SQL operations such as creating stored procedures, triggers etc. but cannot have access to the memory as well as cannot create files.
* **External Access Sandbox:**Users can access files without having the right to manipulate the memory allocation.
* **Unsafe Access Sandbox:**This contains untrusted codes where a user can have access to memory.

**Q #19) What is the difference between SQL and PL/SQL?**

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

**Q #20) What is the difference between SQL and MySQL?**

**Answer:**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.

**Q #21) What is the use of the NVL function?**

**Answer:**NVL function is used to convert the null value to its actual value.

**Q #22) What is the Cartesian product of the table?**

**Answer:**The output of Cross Join is called a Cartesian product. It returns rows combining each row from the first table with each row of the second table. **For Example,** if we join two tables having 15 and 20 columns the Cartesian product of two tables will be 15×20=300 rows.

**Q #23) What do you mean by Subquery?**

**Answer:**Query within another query is called as Subquery. A subquery is called inner query which returns output that is to be used by another query.

**Q #24) How many row comparison operators are used while working with a subquery?**

**Answer:**There are 3-row comparison operators that are used in subqueries such as IN, ANY and ALL.

**Q #25) What is the difference between clustered and non-clustered indexes?**

**Answer: The differences between the two are as follows:**

* One table can have only one clustered index but multiple non-clustered indexes.
* Clustered indexes can be read rapidly rather than non-clustered indexes.
* Clustered indexes store data physically in the table or view whereas, non-clustered indexes do not store data in the table as it has separate structure from the data row.

**Q #26) What is the difference between DELETE and TRUNCATE?**

**Answer: The differences are:**

* The basic difference in both is DELETE command is DML command and the TRUNCATE command is DDL.
* DELETE command is used to delete a specific row from the table whereas the TRUNCATE command is used to remove all rows from the table.
* We can use the DELETE command with WHERE clause but cannot use the TRUNCATE command with it.

**Q #27) What is the difference between DROP and TRUNCATE?**

**Answer:**TRUNCATE removes all rows from the table which cannot be retrieved back, DROP removes the entire table from the database and it also cannot be retrieved back.

**Q #28) How to write a query to show the details of a student from Students table whose  
name start with K?**

**Answer: Query:**

|  |
| --- |
| SELECT \* FROM Student WHERE Student\_Name like ‘K%’; |

Here ‘like’ operator is used to perform pattern matching.

**Q #29) What is the difference between Nested Subquery and Correlated Subquery?**

**Answer:**Subquery within another subquery is called Nested Subquery.  If the output of a subquery depends on column values of the parent query table then the query is called Correlated Subquery.

|  |
| --- |
| SELECT adminid(SELEC Firstname+' '+Lastname  FROM Employee WHERE   empid=emp. adminid)AS EmpAdminId FROM Employee; |

The result of the query is the details of an employee from the Employee table.

**Q #30) What is Normalization? How many Normalization forms are there?**

**Answer:**Normalization is used to organize the data in such a manner that data redundancy will never occur in the database and avoid insert, update and delete anomalies.

**There are 5 forms of Normalization:**

* **First Normal Form (1NF):** It removes all duplicate columns from the table. It creates a table for related data and identifies unique column values.
* **First Normal Form (2NF):** Follows 1NF and creates and places data subsets in an individual table and defines the relationship between tables using the primary key.
* **Third Normal Form (3NF):** Follows 2NF and removes those columns which are not related through the primary key.
* **Fourth Normal Form (4NF):** Follows 3NF and does not define multi-valued dependencies. 4NF is also known as BCNF.

**Q #31) What is a Relationship? How many types of Relationships are there?**

**Answer:**The relationship can be defined as the connection between more than one table in the database.

**There are 4 types of relationships:**

* One to One Relationship
* Many to One Relationship
* Many to Many Relationship
* One to Many Relationship

**Q #32) What do you mean by Stored Procedures? How do we use it?**

**Answer:**A stored procedure is a collection of SQL statements that can be used as a function to access the database. We can create these stored procedures earlier before using it and can execute them wherever required by applying some conditional logic to it. Stored procedures are also used to reduce network traffic and improve performance.

**Syntax:**

|  |
| --- |
| CREATE Procedure Procedure\_Name   (   //Parameters   )   AS   BEGIN   SQL statements in stored procedures to update/retrieve records   END |

**Q #33) State some properties of Relational databases?**

**Answer: Properties are as follows:**

* In relational databases, each column should have a unique name.
* The sequence of rows and columns in relational databases is insignificant.
* All values are atomic and each row is unique.

**Q #34) What are Nested Triggers?**

**Answer:**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.

**Q #35) What is a Cursor?**

**Answer:**A cursor is a database object which is used to manipulate data in a row-to-row manner.

**Cursor follows steps as given below:**

* Declare Cursor
* Open Cursor
* Retrieve row from the Cursor
* Process the row
* Close Cursor
* Deallocate Cursor

**Q #36) What is Collation?**

**Answer:**Collation is a set of rules that check how the data is sorted by comparing it. Such as character data is stored using correct character sequence along with case sensitivity, type, and accent.

**Q #37) What do we need to check in Database Testing?**

**Answer: In Database testing, the following thing is required to be tested:**

* Database connectivity
* Constraint check
* Required application field and its size
* Data Retrieval and processing with DML operations
* Stored Procedures
* Functional flow

**Q #38) What is Database White Box Testing?**

**Answer: Database White Box testing involves:**

* Database Consistency and ACID properties
* Database triggers and logical views
* Decision Coverage, Condition Coverage, and Statement Coverage
* Database Tables, Data Model, and Database Schema
* Referential integrity rules

**Q #39) What is Database Black Box Testing?**

**Answer: Database Black Box testing involves:**

* Data Mapping
* Data stored and retrieved
* Use of Black Box testing techniques such as Equivalence Partitioning and Boundary Value Analysis (BVA)

**Q #40) What are Indexes in SQL?**

**Answer:**The index can be defined as the way to retrieve the data more quickly. We can define indexes using CREATE statements.

**Syntax:**

|  |
| --- |
| CREATE INDEX index\_name   ON table\_name (column\_name) |

Further, we can also create a Unique Index using the following syntax:

|  |
| --- |
| CREATE UNIQUE INDEX index\_name   ON table\_name (column\_name) |

**UPDATE:**We have added few more short questions for practice.

**Q #41) What does SQL stand for?**

**Answer:** SQL stands for [Structured Query Language](http://en.wikipedia.org/wiki/SQL).

**Q #42) How to select all records from the table?**

**Answer:** To select all the records from the table we need to use the following syntax:

|  |
| --- |
| Select \* from table\_name; |

**Q #43) Define join and name different types of joins?**

**Answer:** Join keyword is used to fetch data from two or more related tables. It returns rows where there is at least one match in both the tables included in the join. [Read more here](http://www.w3schools.com/sql/sql_join.asp).  
**Type of joins are:**

1. Right join
2. Outer join
3. Full join
4. Cross join
5. Self join.

**Q #44) What is the syntax to add a record to a table?**

**Answer:** To add a record in a table INSERT syntax is used.

**For Example,**

|  |
| --- |
| INSERT into table\_name VALUES (value1, value2..); |

**Q #45) How do you add a column to a table?**

**Answer:**To add another column in the table, use the following command:

|  |
| --- |
| ALTER TABLE table\_name ADD (column\_name); |

**Q #46) Define the SQL DELETE statement.**

**Answer:** DELETE is used to delete a row or rows from a table based on the specified condition.  
**The basic syntax is as follows:**

|  |
| --- |
| DELETE FROM table\_name  WHERE <Condition> |

**Q #47) Define COMMIT?**

**Answer:** COMMIT saves all changes made by DML statements.

**Q #48) What is the Primary key?**

**Answer:** A Primary key is a column whose values uniquely identify every row in a table. Primary key values can never be reused.

**Q #49) What are Foreign keys?**

**Answer:** When a table’s primary key field is added to related tables in order to create the common field which relates the two tables, it called a foreign key in other tables. Foreign key constraints enforce referential integrity.

**Q #50) What is CHECK Constraint?**

**Answer:** A CHECK constraint is used to limit the values or type of data that can be stored in a column. They are used to enforce domain integrity.

**Q #51) Is it possible for a table to have more than one foreign key?**

**Answer:** Yes, a table can have many foreign keys but only one primary key.

**Q #52) What are the possible values for the BOOLEAN data field?**

**Answer:** For a BOOLEAN data field, two values are possible: -1(true) and 0(false).

**Q #53) What is a stored procedure?**

**Answer:** A stored procedure is a set of SQL queries that can take input and send back output.

**Q #54) What is identity in SQL?**

**Answer:** An identity column in where SQL automatically generates numeric values. We can define a start and increment value of the identity column.

**Q #55) What is Normalization?**

**Answer:** The process of table design to minimize the data redundancy is called normalization. We need to divide a database into two or more table and define the relationship between them.

**Q #56) What is a Trigger?**

**Answer:** The Trigger allows us to execute a batch of SQL code when table event occurs (INSERT, UPDATE or DELETE commands are executed against a specific table).

**Q #57) How to select random rows from a table?**

**Answer:** Using a SAMPLE clause we can select random rows.

**For Example,**

|  |
| --- |
| SELECT \* FROM table\_name SAMPLE(10); |

**Q #58) Which TCP/IP port does SQL Server run?**

**Answer:** By default SQL Server runs on port 1433.

**Q #59) Write a SQL SELECT query that only returns each name only once from a table?**

**Answer:** To get the result as each name only once, we need to use the DISTINCT keyword.

|  |
| --- |
| SELECT DISTINCT name FROM table\_name; |

**Q #60) Explain DML and DDL?**

**Answer:** DML stands for Data Manipulation Language. INSERT, UPDATE and DELETE  are DML statements.

DDL stands for Data Definition Language. CREATE, ALTER, DROP, RENAME are DDL statements.

**Q #61) Can we rename a column in the output of the SQL query?**

**Answer:** Yes, using the following syntax we can do this.

|  |
| --- |
| SELECT column\_name AS new\_name FROM table\_name; |

**Q #62) Give the order of SQL SELECT?**

**Answer:** Order of SQL SELECT clauses is: SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY. Only the SELECT and FROM clauses are mandatory.

**Q #63) Suppose a Student column has two columns, Name and Marks. How to get names and marks of the top three students.**  
**Answer:** SELECT Name, Marks FROM Student s1 where 3 <= (SELECT COUNT(\*) FROM Students s2 WHERE s1.marks = s2.marks)

**Q #64) What is SQL comments?**

**Answer:** SQL comments can be inserted by adding two consecutive hyphens (–).

**Q #65) Difference between TRUNCATE, DELETE and DROP commands?**

**Answer:**

* **DELETE** removes some or all rows from a table based on the condition. It can be rolled back.
* **TRUNCATE** removes ALL rows from a table by de-allocating the memory pages. The operation cannot be rolled back
* **DROP** command removes a table from the database completely.

**Q #66) What are the properties of a transaction?**

**Answer:** Generally, these properties are referred to as ACID properties. They are:

1. Atomicity
2. Consistency
3. Isolation
4. Durability.

**Q #67) What do you mean by ROWID?**

**Answer:** It’s an 18 character long pseudo column attached with each row of a table.

**Q #68) Define UNION, MINUS, UNION ALL, INTERSECT?**

**Answer:**

* **MINUS** – returns all distinct rows selected by the first query but not by the second.
* **UNION** – returns all distinct rows selected by either query
* **UNION ALL** – returns all rows selected by either query, including all duplicates.
* **INTERSECT** – returns all distinct rows selected by both queries.

**Q #69) What is a transaction?**

**Answer:** A transaction is a sequence of code that runs against a database. It takes the database from one consistent state to another.

**Q #70) What is the difference between UNIQUE and PRIMARY KEY constraints?**

**Answer: The differences are as follows:**

* A table can have only one PRIMARY KEY whereas there can be any number of UNIQUE keys.
* The primary key cannot contain Null values whereas the Unique key can contain Null values.

**Q #71) What is a composite primary key?**

**Answer:** The primary key created on more than one column is called composite primary key.

**Q #72) What is an Index?**

**Answer:** An Index is a special structure associated with a table to speed up the performance of queries. The index can be created on one or more columns of a table.

**Q #73) What is the Subquery?**

**Answer:** A Subquery is a subset of select statements whose return values are used in filtering conditions of the main query.

**Q #74) What do you mean by query optimization?**

**Answer:** Query optimization is a process in which a database system compares different query strategies and select the query with the least cost.

**Q #75) What is Collation?**

**Answer:** Set of rules that define how data is stored, how case-sensitivity and Kana character can be treated etc.

**Q #76) What is Referential Integrity?**

**Answer:** 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.

**Q #77) What is the Case function?**

**Answer:** Case facilitates if-then-else type of logic in SQL. It evaluates a list of conditions and returns one of the multiple possible result expressions.

**Q #78) Define a temp table?**

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

**Q #79) How can we avoid duplicating records in a query?**

**Answer:** By using the DISTINCT keyword, duplication of records in a query can be avoided.

**Q #80) Explain the difference between Rename and Alias?**

**Answer:** Rename is a permanent name given to a table or column whereas Alias is a temporary name given to a table or column.

**Q #81) What is a View?**

**Answer:** A view is a virtual table that contains data from one or more tables. Views restrict data access of the table by selecting only required values and make complex queries easy.

**Q #82) What are the advantages of Views?**

**Answer: Advantages of Views are:**

* Views restrict access to the data because the view can display selective columns from the table.
* Views can be used to make simple queries to retrieve the results of complicated queries. **For Example,** views can be used to query information from multiple tables without the user knowing.

**Q #83) List the various privileges that a user can grant to another user?**

**Answer:**  SELECT, CONNECT, RESOURCES.

**Q #84) What is schema?**

**Answer:** A schema is a collection of database objects of a User.

**Q #85) What is a Table?**

**Answer:** A table is the basic unit of data storage in the database management system. Table data is stored in rows and columns.

**Q #86) Does View contain Data?**

**Answer:** No, Views are virtual structures.

**Q #87) Can a View based on another View?**

**Answer:** Yes, A View is based on another View.

**Q #88) What is the difference between the HAVING clause and WHERE clause?**

**Answer:** Both specify a search condition but Having clause is used only with the SELECT statement and typically used with GROUP BY clause.  
If GROUP BY clause is not used then Having behaved like WHERE clause only.

**Q #89) What is the difference between Local and Global temporary tables?**

**Answer:**If defined inside a compound statement a local temporary table exists only for the duration of that statement but a global temporary table exists permanently in the DB but its rows disappear when the connection is closed.

**Q #90) What is CTE?**

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