**my is now Scaler Academy!**

[**KNOW MORE**](https://www.scaler.com/)**×**

SQL Interview Questions

right in..

**1. What is Database?**

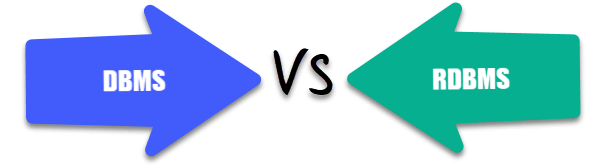
A database is an organized collection of data, stored and retrieved digitally from a remote or local computer system. Databases can be vast and complex, and such databases are developed using fixed design and modeling approaches.

**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.

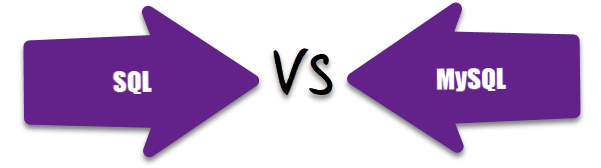


**4. What is SQL?**

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?

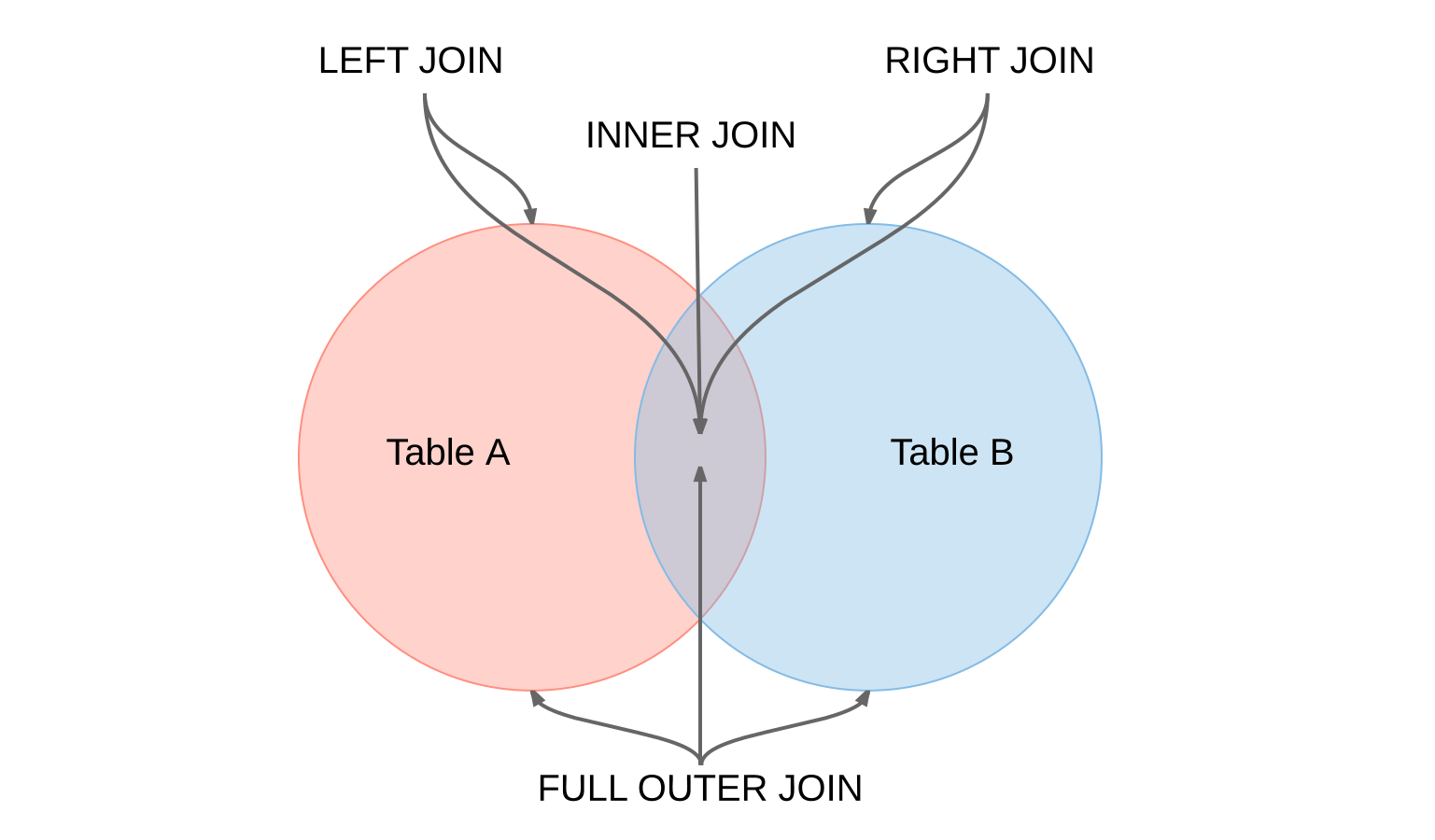


Q   =>   Write a SQL statement to add a FOREIGN KEY 'col\_fk' that references 'col\_pk' in 'table\_x'.



**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;

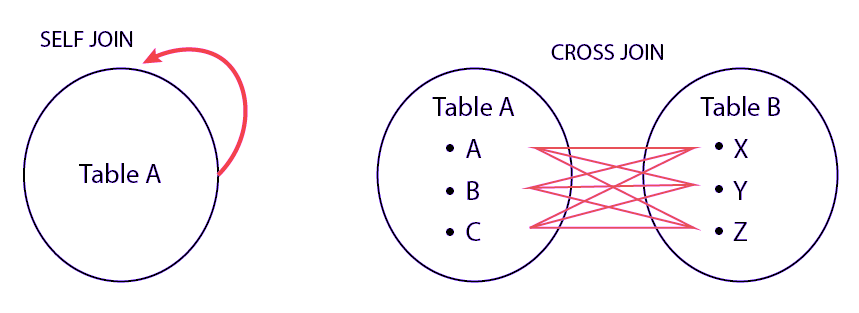
**13. What is a Cross-Join?**

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**

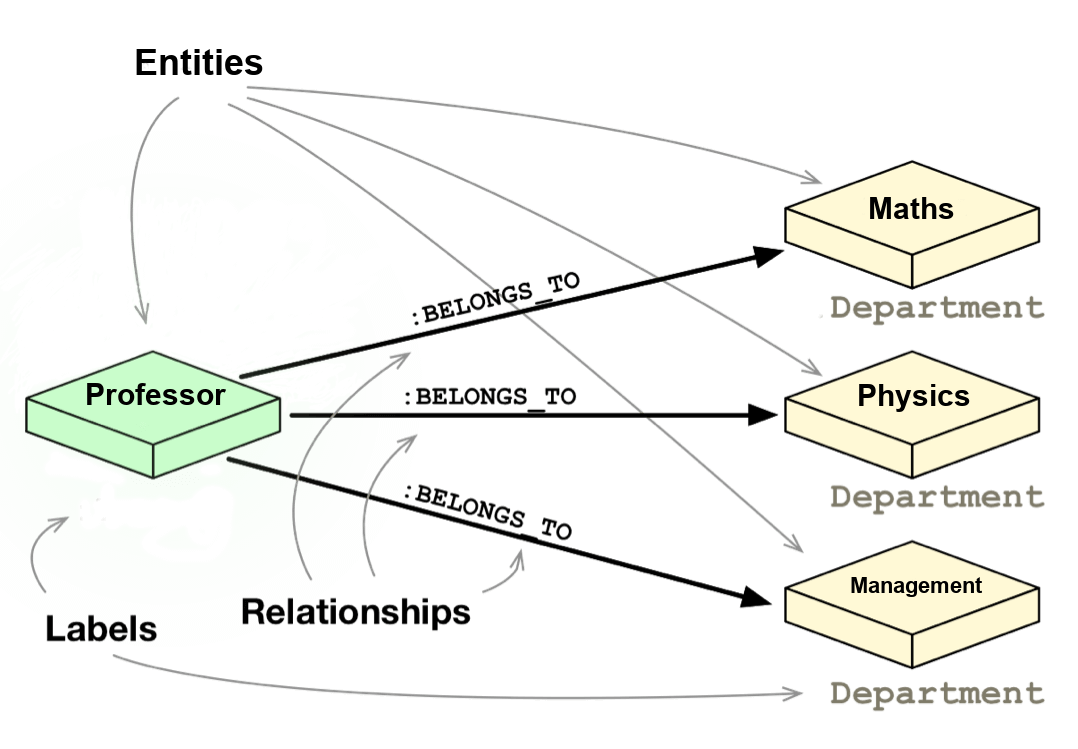
**CLOSE** db\_cursor */\* Close the cursor and deallocate the resources \*/*

**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 \*/*

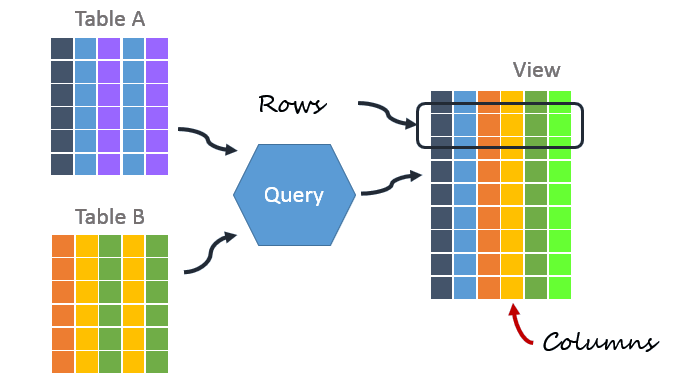
**WHERE** A.emp\_sup = B.emp\_id;

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.

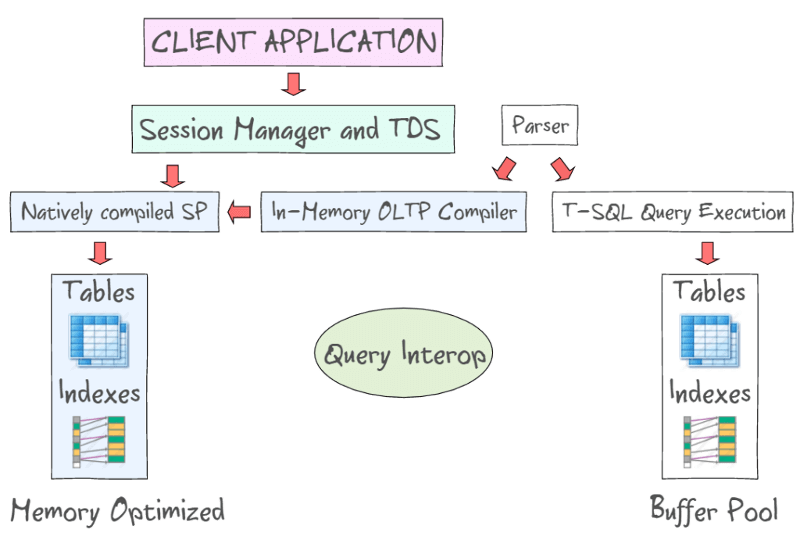
**34. What is User-defined function? What are its various types?**

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?**

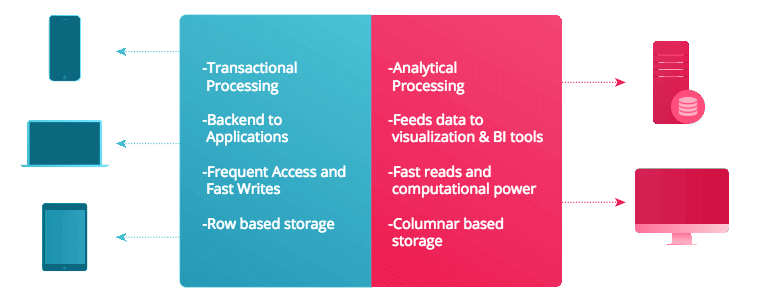
OLTP stands for Online Transaction Processing, is a class of software applications capable of supporting transaction-oriented programs. An essential attribute of an OLTP system is its ability to maintain concurrency. To avoid single points of failure, OLTP systems are often decentralized. These systems 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** $$

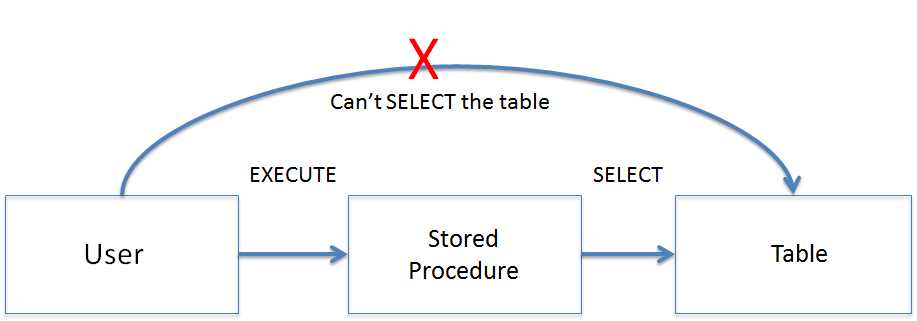
**CREATE** **PROCEDURE** FetchAllStudents()

**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** '\_\_\_\_'

**Practice SQL Questions asked in Interviews**

**Q - Which statement is true for a PRIMARY KEY constraint?**

|  |
| --- |
| Primary key defines a realtionship between two tables. |
| A table in SQL must have a primary key associated with it to uniquely identify its records. |
| A table in SQL is indexed by default based on its primary key. |
| Primary key may or may not be unique but can be comprised of multiple fields. |

**Q - Which statement is false for a FOREIGN KEY constraint?**

|  |
| --- |
| Foreign key defines a relationship between two tables. |
| Foreign Key is automatically created when two tables are joined. |
| Foreign Key uniquely identifies all the records in the referenced table. |
| Foreign key may or may not be unique but can be comprised of multiple fields. |

**Q - What is a Query?**

|  |
| --- |
| A SELECT or UPDATE statement in SQL. |
| A request for data from a table in the database. |
| A request to input data from the user. |
| A request for data from single or multiple tables in the database. |

**Q - What does SQL stand for?**

|  |
| --- |
| Structured Question Language |
| Strong Query Language |
| Structured Query Language |
| Strong Question Language |

**Q - Which statement is used to update data in the database?**

|  |
| --- |
| MODIFY |
| UPDATE |
| ALTER TABLE |
| SAVE AS |

**Q - Query to select all records with "bar" in their name?**

|  |
| --- |
| SELECT \* FROM people WHERE name = "%bar%"; |
| SELECT \* FROM people WHERE name LIKE "%bar%"; |
| SELECT \* FROM people WHERE name IN ("bar"); |
| SELECT \* FROM people WHERE name = "\_bar\_" |

**Q - Which statement is false for ORDER BY statement?**

|  |
| --- |
| Requires a ASC or DESC keyword explicitly to sort the result set. |
| Sorts the result set in descending order using the DESC keyword. |
| Can sort based on multiple columns. |
| None of the above. |

**Q - SQL query used to fetch unique values from a field?**

|  |
| --- |
| SELECT UNIQUE column\_name FROM table\_name; |
| SELECT DISTINCT column\_name FROM table\_name; |
| SELECT column\_name FROM table\_name WHERE COUNT(column\_name) = 1; |
| SELECT UNIQUE column\_name FROM table\_name WHERE COUNT(column\_name) = 1; |

**Q - What is the main advantage of a clustered index over a non-clustered index?**

|  |
| --- |
| It is easier to create and manipulate. |
| It requires extra memory but allows for speedy retrieval of records. |
| It does not require additonal memory and allows for speedy retrieval of records. |
| None of the above. |

**Q - Normalization which has neither composite values nor partial dependencies?**

|  |
| --- |
| Second Normal Formal |
| Third Normal Form |
| Boyce-Codd Normal Form |
| All of the above |

**Q - An SQL query to delete a table from the database and memory while keeping the structure of the table intact?**

|  |
| --- |
| DROP TABLE table\_name; |
| DROP FROM TABLE table\_name; |
| DELETE FROM TABLE table\_name; |
| TRUNCATE TABLE table\_name; |

**Q - Which of the following is known as a virtual table in SQL?**

|  |
| --- |
| SELF JOIN |
| INNER JOIN |
| VIEW |
| NONE |

Advanced SQL Interview Questions And Answers Pdf

**1. What is Failover clustering overview?**  
**Answer:**Failover clustering is mainly used for data availability. Typically, in a failover cluster, there are two machines.  
One machine provides the basic services and the second is available to run the service when the primary system fails.  
The primary system is monitored periodically to check if it works. This monitoring may be performed by the failover computer or an independent system also called a cluster controller. In an event of failure of the primary computer, the failover system takes control.

**2. What is lock escalation?**  
**Answer:**Lock escalation is used to convert row locks and page locks into table locks thereby “escalating” the smaller or finer locks. This increases system performance as each lock is nothing but a memory structure. Too many locks would mean more consumption of memory. Hence, escalation is used.  
Lock escalation from SQL Server 7.0 onwards is dynamically managed by SQL Server. It is the process of converting a lot of low-level locks into higher-level locks.

**3. What is Built-in/Administrator?**  
**Answer:**The Builtin/Administrator account is basically used during some setup to join some machine in the domain. It should be disabled immediately thereafter. For any disaster recovery, the account will be automatically enabled. It should not be used for normal operations.

**4. What is the difference between cross join and natural join?**  
**Answer:**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.

**5. 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 such that once it is executed the database is opened for an attacker. This technique is usually used for attacking Data-Driven Applications to have access to sensitive data and perform administrative tasks on databases.

**For Example**: SELECT column\_name(s) FROM table\_name WHERE condition;

**6. List the ways in which Dynamic SQL can be executed?**  
**Answer:**  
Following are the ways in which dynamic SQL can be executed:

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

**7. What are Views used for?**  
**Answer:**A view refers to a logical snapshot based on a table or another view. It is used for the following reasons:

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

**8. What is Auto Increment in SQL?**  
**Answer:**Autoincrement 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 a PRIMARY KEY is used.

**9. What is the difference between Union and Union All command?**  
**Answer:**The fundamental difference between Union and Union All command is, Union is by default distinct i.e. it combines the distinct result set of two or more select [statements](https://en.wikipedia.org/wiki/Microsoft_SQL_Server). Whereas, Union All combines all the rows including duplicates in the result set of different select statements.

**10. What are transactions in SQL?**  
**Answer:**The transaction is a set of operations performed in a logical sequence. It is executed as a whole, if any statement in the transaction fails, the whole transaction is marked as failed and not committed to the database.

**11. What is SQL Sandbox in SQL Server?**  
**Answer:**SQL Sandbox is the safe place in SQL Server [Environment](https://www.microsoft.com/en-us/sql-server) where untrusted scripts are executed. There are 3 types of SQL sandbox, such as

Safe Access Sandbox: Here a user can perform SQL operations such as creating stored procedures, triggers, etc. but cannot have access to the memory and cannot create files.([Mulesoft Training Online](https://svrtechnologies.com/mulesoft-training/) )  
External Access Sandbox: User can have access to files without having a right to manipulate the memory allocation.  
Unsafe Access Sandbox: This contains untrusted codes where a user can have access to memory.

**12. What is the difference between clustered and non-clustered indexes?**  
**Answer:**One table can have only one clustered index but multiple nonclustered indexes.  
Clustered indexes can be read rapidly rather than non-clustered indexes.  
Clustered indexes store data physically in the table or view and non-clustered indexes do not store data in the table as it has separate structure from the data row.

**13. What are the different types of locks in the database?**  
**Answer:**  
**The different types of locks in the database are-**

Shared locks – Allows data to be read-only(Select operations), prevents the data to be updated when in the shared lock.  
Update locks – Applied to resources that can be updated. There can be only one update lock on a data at a time.  
Exclusive locks – Used to lock data being modified(INSERT, UPDATE, or DELETE) by one transaction thus ensuring that multiple updates cannot be made to the same resource at the same time.  
Intent locks – A notification mechanism using which a transaction conveys that intends to acquire a lock on data.  
Schema locks- Used for operations when schema or structure of the database is required to be updated.  
Bulk Update locks – Used in case of bulk operations when the TABLOCK hint is used.

**14. What is a Table in a Database?**  
**Answer:**A table is a database object used to store records in a field in the form of columns and rows that holds data.

**15. What is database testing?**  
**Answer:**Database testing involves checking the integrity of actual data in the front end with the data present in the database. It involves validating the data in the database, checking that there are no orphan records (record with a foreign key to a parent record that has been deleted”), no junk records are present, updating records in the database and verify the value in the front end.

**16. What is a Relationship and what are they?**  
**Answer:**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.

**17. How can you insert NULL values in a column while inserting the data?**  
**Answer:**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

**18. What is a View in SQL?**  
**Answer:**Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on a certain condition.

The CREATE VIEW statement of SQL is used for creating Views.

**Basic Syntax:**

CREATE VIEW view\_name AS  
SELECT column1, column2…..  
FROM table\_name  
WHERE condition;

view\_name: Name for the View  
table\_name: Name of the table  
condition: Condition to select rows  
For more details on how to create and use view, please refer to this article.

**19. What is a join in SQL? What are the types of joins?**  
**Answer:**An SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are:  
INNER JOIN: The INNER JOIN keyword selects all rows from both the tables as long as the condition satisfies. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be same.

**LEFT JOIN**: This joins returns all the rows of the table on the left side of the join and matching rows for the table on the right side of the join. The rows for which there is no matching row on the right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN

**RIGHT JOIN**: RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. The rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as the RIGHT OUTER JOIN.

**FULL JOIN**: FULL JOIN creates the result-set by combining the result of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both the tables. The rows for which there is no matching, the result-set will contain NULL values.  
To know about each of these joins in details, refer this.

**20. What is RDBMS?**  
**Answer:**An RDBMS or Relational Database Management System is a type of DBMS having relationships between the tables using indexes and different constraints like primary key, foreign key, etc. The use of indexes and constraints helps in faster retrieval and better management of data within the databases. ( [oracle apex training online](https://svrtechnologies.com/apex-training/) )

**21. What is the difference between DBMS and RDBMS?**  
**Answer:**The primary difference between DBMS and RDBMS is, in RDBMS we have relations between the tables of the database. Whereas in DBMS there is no relation between the tables(data may even be stored in files).  
RDBMS has primary keys and data is stored in tables. DBMS has no concept of primary keys with data stored in navigational or hierarchical form.  
RDBMS defines integrity constraints in order to follow ACID properties. While DBMS doesn’t follow ACID properties.

**22. What is a field in a table?**  
**Answer:**A field is an entity used for storing a particular type of data within a table like numbers, characters, dates, etc.

**23. How to avoid duplicate records in a query?**  
**Answer:**The SQL SELECT DISTINCT query is used to return only unique values. It eliminates all the duplicated values.([Selenium Training](https://svrtechnologies.com/selenium-training/) )  
View Detailed Post.

**24. What is the difference between an inner and outer join?**  
**Answer:**An inner join returns rows when there is at least some matching data between two (or more) tables that are being compared.  
An outer join returns rows from both tables that include the records that are unmatched from one or both the tables.

**25. What is SQL?**  
**Answer:**SQL stands for Structured Query Language. It is a language used to interact with the database, i.e to create a database, to create a table in the database, to retrieve data or update a table in the database, etc. SQL is an ANSI(American National Standards Institute) standard. Using SQL, we can do many things, for example – we can execute queries, we can insert records in a table, we can update records, we can create a database, we can create a table, we can delete a table, etc.

**26. Explain how to send email from SQL database?**  
**Answer:**SQL Server has a feature for sending mails. Stored procedures can also be used for sending mail on demand. With SQL Server 2005, MAPI client is not needed for sending mails.  
The following is the process for sending emails from the database.

Make sure that the SQL Server Mail account is configured correctly and enable Database Mail.  
Write a script to send an e-mail. The following is the script.([Sap Fico Online Training](https://svrtechnologies.com/sap-fico-online-training/) )  
USE [YourDB]  
EXEC msdb.dbo.sp\_send\_dbmail  
abc@intellipaat.com;pqr@intellipaat.com’  
@body = ‘ A warm wish for your future endeavor’,  
@subject = ‘This mail was sent using Database Mail’ ;

**27. What are the different types of SQL commands?**  
**Answer:**SQL commands are the set of commands used to communicate and manage the data present in the database. The different type of SQL commands are-

DDL – Data Definition Language  
DML – Data Manipulation Language  
DCL – Data Control Language  
TCL – Transactional Control Language

**28. What is the temp table?**  
**Answer:**A temp table is a temporary storage structure to store the data temporarily.

**29. How to make a remote connection in a database?**  
**Answer:**  
The following is the process to make a remote connection in the database:

Use SQL Server Surface Area Configuration Tool for enabling the remote connection in the database.  
Click on Surface Area Configuration for Services and Connections.  
Click on SQLEXPRESS/Database Engine/RemoteConnections.  
Select the radio button: Local and Remote Connections and select ‘Using TCP/IP only’ under Local and Remote Connections.([Azure Training](https://svrtechnologies.com/azure-training/) )  
Click on OK button / Apply button

**30. What is the difference between delete, truncate and drop command?**  
**Answer:**The difference between the Delete, Truncate and Drop command is – Delete command is a DML command, it removes rows from a table based on the condition specified in the where clause, being a DML statement we can rollback changes made by delete command.  
Truncate is a DDL command, it removes all the rows from a table and also frees the space held, unlike delete command. It takes a lock on the table while delete command takes a lock on rows of the table.  
Drop is a DDL command, it removes the complete data along with the table structure(unlike truncate command that removes only the rows).

**31. Can a table contain multiple PRIMARY KEY’s?**  
**Answer:**The short answer is no, a table is not allowed to contain multiple primary keys but it allows to have one composite primary key consisting of two or more columns.

**32. What do you mean by Denormalization?**  
**Answer:**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.

**33. What is the largest value that can be stored in a BYTE data field?**  
**Answer:**The largest number that can be represented in a single byte is 11111111 or 255. The number of possible values is 256 (i.e. 255 (the largest possible value) plus 1 (zero), or 28).

**34. What are Operators available in SQL?**  
**Answer:**SQL Operator is a reserved word used primarily in an SQL statement’s WHERE clause to perform operations, such as arithmetic operations and comparisons. These are used to specify conditions in an SQL statement.

**There are three types of Operators.**

* Arithmetic Operators
* Comparison Operators
* Logical Operators
* View Detailed Post

**35. List out the ACID properties and explain?**  
**Answer:**Following are the four properties of ACID. These guarantees that the database transactions are processed reliably.

* Atomicity
* Consistency
* Isolation
* Durability

**36. What are the different types of subquery?**  
**Answer:**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.  
Ques.31. Define the select into a statement.  
Ans. Select into statement is used to directly select data from one table and insert into other, the new table gets created with the same name and type as of the old table-

**37. What are the ACID properties?**  
**Answe**r: ACID properties refer to the four properties of transactions in SQL-

Atomicity – All the operations in the transaction are performed as a whole or not performed at all.  
Consistency – State of database changes only on successfully committed transaction.  
Isolation – Even with concurrent execution of the multiple transactions, the final state of the DB would be same as if transactions got executed sequentially. In other words, each transaction is isolated from one another.  
Durability – Even in the state of crash or power loss the state of the committed transaction remain persistent.

**38. What are locks in SQL?**  
**Answer:**Locks in SQL are used for maintaining database integrity in case of concurrent execution of the same piece of data.

**39. How many types of Privileges are available in SQL?**  
**Answer:**There are two types of privileges used in SQL, such as

System Privilege: System privileges deal with an object of a particular type and specifies the right to perform one or more actions on it which include Admin allows a user to perform administrative tasks, ALTER ANY INDEX, ALTER ANY CACHE GROUP creates/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.

**40. Write the Syntax for STUFF function in an SQL server?**  
**Answer:**

* STUFF (String1, Position, Length, String2)
* String1 – String to be overwritten
* Position – Starting location for overwriting
* Length – Length of substitute string
* String2- String to overwrite.

**41. Is it possible to import data directly from T-SQL commands without using SQL Server Integration Services? If so, what are the commands?**  
**Answer:**Yes, six commands are available to import data directly in the T-SQL language. These commands include :

**BCP**: The bulk copy (BCP) command of Microsoft SQL Server provides you with the ability to insert large numbers of records directly from the command line. In addition to being a great tool for command-line aficionados, BCP is a powerful tool for those seeking to insert data into a SQL Server database from within a batch file or other programmatic method.

**Bulk Insert:** The BULK INSERT statement was introduced in SQL Server 7 and allows you to interact with BCP (bulk copy program) via a script.  
O**penRowSet:** The OPENROWSET function can be referenced in the FROM clause of a query as if it were a table name. The OPENROWSET function can also be referenced as the target table of an INSERT, UPDATE, or DELETE statement, subject to the capabilities of the OLE DB provider. Although the query might return multiple result sets, OPENROWSET returns only the first one.  
**OPENDATASOURCE** Provides ad hoc connection information as part of a four-part object name without using a linked server name.

**OPENQUERY**: Executes the specified pass-through query on the specified linked server. This server is an OLE DB data source. OPENQUERY can be referenced in the FROM clause of a query as if it were a table name.

**Linked Servers**: Configure a linked server to enable the SQL Server Database Engine to execute commands against OLE DB data sources outside of the instance of SQL Server. Typically linked servers are configured to enable the Database Engine to execute a Transact-SQL statement that includes tables in another instance of SQL Server, or another database product such as Oracle.

**42. In what version of SQL Server were synonyms released?**  
**Answer:**How do synonyms work and explain its use cases? Synonyms were released with SQL Server 2005.  
Synonyms enable the reference of another object (View, Table, Stored Procedure or Function) potentially on a different server, database or schema in your environment. In simple words, the original object that is referenced in the whole code is using a completely different underlying object, but no coding changes are necessary. Think of this as an alias as a means to simplify migrations and application testing without the need to make any dependent coding changes.  
Synonyms can offer a great deal of value when converting underlying database objects without breaking front end or middle-tier code. This could be useful during a re-architecture or upgrade project.  
Become Master of SQL by going through this SQL training course.

**43. What is Normalization and what are the advantages of it?**  
**Answer:**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.

**44. 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 cannot be retrieved back.

**45. 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 is depending on column values of the parent query table then the query is called Correlated Subquery.

SELECT admin(SELEC Firstname+’ ‘+Lastname FROM Employee WHERE  
empid=emp. admin)AS EmpAdminId FROM Employee

This query gets details of an employee from the Employee table.

**46. What is Normalization? How many Normalization forms are there?**  
**Answer:**Normalization is used to organize the data in such 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. 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 do not define multi-valued dependencies. 4NF also known as BCNF

**47. What is Relationship? How many types of Relationship are there?**  
**Answer:**The relationship can be defined as the connection between more than one tables 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

**48. State some properties of Relational databases?**  
**Answer:**In relational databases, each column should have a unique name  
The sequence of rows and columns in relational databases are insignificant  
All values are atomic and each row is unique

**49. What are Nested Triggers?**  
**Answer:**Triggers may implement data modification logic by using INSERT, UPDATE, and DELETE statement. These triggers that contain data modification logic and find other triggers for data modification are called Nested Triggers.

**50. What is the 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

**51. 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.

**52. What do we need to check in Database Testing?**

**Answer:**Generally, in Database Testing following thing is to need to be tested

* Database Connectivity
* Constraint Check
* Required Application Field and its size
* Data Retrieval and Processing With DML operations
* Stored Procedures
* Functional flow

**53. Define join and name different types of joins?**  
**Answer:**Join keyword is used to fetch data from related two or more tables. It returns rows where there is at least one match in both the tables included in the join. Read more here. ()

**Type of joins are:**

* Right Join
* Outer Join
* Full Join
* Cross Join
* Self Join.

**54. What are Entities and Relationships?**  
**Answer:**

**Entities**: A person, place, or thing in the real world about which data can be stored in a database. Tables store data that ([Sql Server Training](https://svrtechnologies.com/sql-server-training/) )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).

**55. What is the difference between NULL value, Zero, and Blank space?**  
**Answer:**As I mentioned earlier, Null value is a field with no value which is different from zero value and blank space.  
A null value is a field with no value.  
Zero is a number  
Blank space is the value we provide. The ASCII value of space is CHAR(32).

**56. What is the difference between 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.

**57. What is the need for group functions in SQL?**  
**Answer:**Group functions work on the set of rows and returns one result ([Servicenow Training](https://svrtechnologies.com/servicenow-training/) )per group. Some of the commonly used group functions are AVG, COUNT, MAX, MIN, SUM, VARIANCE.

**58. What are transaction and its 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 performed like Creating, updating, deleting records in the database comes from transactions.

In simple word, 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 such as 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 from where the transaction is to be rolled back

**59. How many Aggregate Functions are available there in SQL?**  
**Answer:**SQL Aggregate Functions calculates values from multiple columns in a table and returns a single value. ( [hadoop training videos](https://svrtechnologies.com/hadoop-training/" \t "_blank) )

**There are 7 aggregate functions we use in SQL**

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

**60. What is Stored procedure?**  
**Answer:**A Stored Procedure is a collection of SQL statements that have been created and stored in the database to perform a particular task. The stored procedure accepts input parameters and processes them and returns a single value such as a number or text value or a result set (set of rows).