**What are the different JOIN types and what do they do?**

The different join types in Oracle SQL are:

* **Inner join**: Returns records that exist in both tables.
* **Left join/left outer join**: Returns records that exist in the first table and shows NULL for those values that don’t exist in the second table.
* **Right join/right outer join**: Returns records that exist in the second table and shows NULL for those values that don’t exist in the first table.
* **Full join/full outer join**: Returns records that exist in both the first and second table, and shows NULL for those values that don’t exist in the corresponding table.
* **Cross join**: Returns all combinations of all records in both tables.
* **Natural join**: an inner join with two tables on columns that have the same names.
* **Self join**: A join from one table to another record in the same table.

### What is a “cross join”?

A cross join is a type of join where the results displayed contain the records from both tables in all possible combinations. There is no field used to perform the join.

For example, if table A has 10 records and table B has 8 records, then the cross join will result in 80 (or 10 x 8) records.

The result can also be called a “cartesian product”.

### What is a self join and why would you use one?

A self join is a type of join where a table is joined to itself.

You would use a self join when a table has a field that refers to another record in the same table. It’s often used in hierarchical structures, such as employee tables having a manager\_id column where the manager\_id refers to another employee record.

### What does UNION do? What’s the difference between UNION and UNION ALL?

Union allows you to combine two sets of results into one result.

It’s [different to UNION ALL](https://www.databasestar.com/sql-set-operators/) because UNION removes duplicate values and UNION ALL does not.

### What’s the difference between UNION and JOIN?

A join allows us to lookup data from one table in another table based on common fields (for example employees and departments). It requires us to have a field that is common in both tables.

A union allows us to combine the results of two queries into a single result. No join between the results is needed. Only the number and type of columns need to be the same.

### What is a subquery?

A subquery is a query within another query. This subquery can be in many places, such as in the FROM clause, the SELECT clause, or a WHERE clause.

It’s often used if you need to use the result of one query as an input into another query.

### 38. What is a correlated subquery?

A correlated subquery is a subquery that refers to a field in the outer query.

Subqueries can be standalone queries (non-correlated), or they can use fields in the outer query. These fields are often used in join conditions or in WHERE clauses.

### How can you perform conditional logic in an SQL statement?

This can be done using the [CASE statement](https://www.databasestar.com/sql-case-statement/).

What is a subquery?  
**A.**Subquery or Inner query or Nested query is a query in a query. A subquery is usually added in the WHERE clause of the sql statement. A subquery can be nested inside a **SELECT**, **INSERT**, **UPDATE**, or **DELETE** statement or inside another subquery.  Subqueries are an alternate way of returning data from multiple tables.

What are the advantages and disadvantages of using a subquery?  
**A**.

**Advantages**:

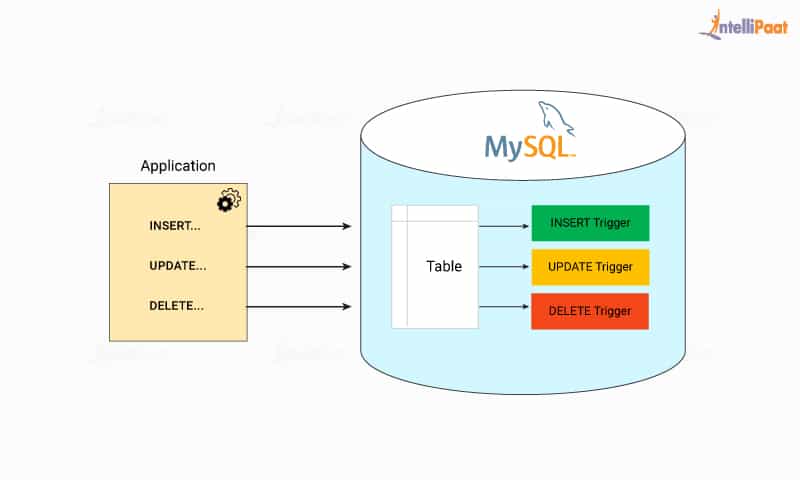
* Subqueries allow you to use the results of another query in the outer query.
* Subqueries in some complex SQL queries can simplify coding and improve maintainability by breaking down the complex query into a series of logical steps.
* In some cases, subqueries are easier to understand than complex joins and unions.

**Disadvantages**:

* When a subquery is used, the query optimizer of the database server may have to perform additional steps like sorting the results, etc. Hence, in some cases subqueries can be less efficient than using joins. So, favor joins to subqueries

**What are the TRIGGERS that can be used in MySQL tables?**

Following TRIGGERS are allowed in MySQL:



* BEFORE INSERT
* AFTER INSERT
* BEFORE UPDATE
* AFTER UPDATE
* BEFORE DELETE
* AFTER DELETE

How to create a Stored Procedure in MySQL?

A stored procedure is a group of SQL statements that we save in the database. The SQL queries, including INSERT, UPDATE, DELETE, etc. can be a part of the stored procedure. A procedure allows us to use the same code over and over again by executing a single statement. It stores in the database data dictionary.

We can create a stored procedure using the below syntax:

1. **CREATE** **PROCEDURE** procedure\_name [ (parameter datatype [, parameter datatype]) ]
2. **BEGIN**
3. Body\_section **of** SQL statements
4. **END**;

### How to execute a stored procedure in MySQL?

We can execute a stored procedure in MySQL by simply CALL query. This query takes the name of the stored procedure and any parameters we need to pass to it. The following is the basic syntax to execute a stored procedure:

1. CALL stored\_procedure\_name (argument\_list);

How to create a View in MySQL?

A view is a database object whose values are based on the base table. It is a **virtual table** created by a query by joining one or more tables. It is operated similarly to the base table but does not contain any data of its own. If any changes occur in the underlying table, the same changes reflected in the View also.

Following is the general syntax of creating a VIEW in MySQL:

1. **CREATE** [OR REPLACE] **VIEW** view\_name **AS**
2. **SELECT** columns
3. **FROM** tables
4. [**WHERE** conditions];

How to create a Trigger in MySQL?

A trigger is a procedural code in a database that automatically invokes whenever certain events on a particular table or view in the database occur. It can be executed when records are inserted into a table, or any columns are being updated. We can create a trigger in MySQL using the syntax as follows:

1. **CREATE** **TRIGGER** trigger\_name
2. [before | **after**]
3. {**insert** | **update** | **delete**}
4. **ON** table\_name [**FOR** EACH ROW]
5. **BEGIN**
6. --variable declarations
7. --trigger code
8. **END**;

How to import a CSV file in MySQL?

MySQL allows us to import the CSV (comma separated values) file into a database or table. A CSV is a plain text file that contains the list of data and can be saved in a tabular format. MySQL provides the LOAD DATA INFILE statement to import a CSV file. This statement is used to read a text file and import it into a database table very quickly. The full syntax to import a CSV file is given below:

1. **LOAD** DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/filename.csv'
2. **INTO** **TABLE** tablename
3. FIELDS TERMINATED **BY** ','
4. OPTIONALLY ENCLOSED **BY** '"'
5. LINES TERMINATED **BY** '\r\n'
6. **IGNORE** 1 **ROWS**;

### What is the usage of ENUMs in MySQL?

ENUMs are string objects. By defining ENUMs, we allow the end-user to give correct input as in case the user provides an input that is not part of the ENUM defined data, then the query won't execute, and an error message will be displayed which says "The wrong Query". For instance, suppose we want to take the gender of the user as an input, so we specify ENUM('male', 'female', 'other'), and hence whenever the user tries to input any string any other than these three it results in an error.

ENUMs are used to limit the possible values that go in the table:

**For example:**

CREATE TABLE months (month ENUM 'January', 'February', 'March'); INSERT months VALUES ('April').

Following are some differences between stored procedures and triggers:

* Event and actions needs to be identified at the time of creating a trigger. Whereas, at the time of creating a stored procedure, it is not important.
* Trigger can be run automatically when any event occurs. Whereas, stored procedures have to be run manually.
* A stored procedure can be called within a trigger. Whereas, a trigger cannot be called in a stored procedure.
* Triggers execute implicitly. Whereas, stored procedures execute explicitly.
* A trigger cannot be called from front end. Whereas, a stored procedure can be.

**What are the advantages of using a Stored Procedures?**  
Ans: Following are the main advantage of using a SP

* Reduce network usage between clients and servers – stored procedures perform intermediate processing on the database server reducing unnecessary data transfer across the network
* Improved security – database administrator can control the users who access the stored procedure
* Reduced development cost and increased reliability
* Stored procedures are tunable to improve the performance. When same stored procedure executed again, it can use the previously cached execution plans
* Separate or abstract server side functions from the client side
* Stored procedures can encapsulate logic. You can change stored procedure code without affecting clients.
* Access to other database objects in a secure and uniform way
* Can prevent SQL injection attacks
* Unit testable
* Encapsulation of business logic – less chances to data become corrupted through faulty client programs.

**9). What are the disadvantages of using a Stored Procedures?**  
Ans: Following are the main disadvantage of using a SP

* Writing and maintaining stored procedures requires more specialized skills.
* There are no debuggers available for stored procedures
* Stored procedure language may differ from one database system to another.
* Poor exception handling
* Tightly coupled to the database system
* Not possible to use objects
* Sometimes it is hard to understand the logic written in dynamic SQL

**What are different types of Stored Procedures?**

**Answer:**

There are two types of Stored Procedures:

1.User Defined Procedures : This category includes the code written by developers.

2.System Stored Procedures:  These are stored procedures which are already written scripts in Oracle. User just needs to run that procedure.

3.Extended Procedure

4.CLR Stored Procedure

**What is difference between Stored Procedure and Function?**

**Answer:**

Following is difference between Stored Procedure and functions.

|  |  |  |
| --- | --- | --- |
|  | Stored Procedure | Functions |
| Compilation | Stored in database in compiled format.  Note: Compiled indicates, Execution plan will be made by sql at the time it created and stored in DB. | Will compiled at run time |
| Return type | It can directly return only integers  Return type is not must | It can return any scalar or table  Return type is must |
| Multiple return values | It can also return more than one values (of any data type) indirectly with the help of out parameters | It won’t support out parameters |
| DML Statements | Can have DML statements. | Cannot have DML statements.  Note: In case of multi-table valued functions it can contain DML statements affecting Table Variables. |
| Execution | Stored procedure can execute function.  Cannot be the part of Select query as a column.  Stored Procedures cannot be used in the SQL statements anywhere in the WHERE/HAVING/SELECT | Function cannot execute stored procedure.  Can be the part of select query as a column.  Functions be used in the SQL statements anywhere in the WHERE/HAVING/SELECT |

**What are different parameters of Stored Procedures?( 100% asked Stored Procedure Interview Questions )**

**Answer:**

We can pass parameters to procedures in three ways.  
1) IN-parameters  
2) OUT-parameters  
3) IN OUT-parameters

**Can Procedure contain return value?**

**Answer:**

Procedure may or may not contain return value. Function must return the value.

**What is the purpose of normalization in DBMS?**

**Answer:** Normalization is the process of analyzing the relational schemas which are based on their respective functional dependencies and the primary keys in order to fulfill certain properties.

**The properties include:**

* To minimize the redundancy of the data.
* To minimize the Insert, Delete and Update Anomalies.

**What is 1NF in the DBMS?**

**Answer:** 1NF is known as the **First Normal Form**.

This is the easiest form of the normalization process which states that the **d**omain of an attribute should have only atomic values. The objective of this is to remove the duplicate columns that are present in the table.

**Q #20) What is 2NF in the DBMS?**

**Answer:**2NF is the **Second Normal Form**.

Any table is said to have in the 2NF if it satisfies the following 2 conditions:

* A table is in the 1NF.
* Each non-prime attribute of a table is said to be functionally dependent in totality on the primary key.

**Q #21) What is 3NF in the DBMS?**

**Answer:** 3NF is the **Third Normal Form.**

Any table is said to have in the 3NF if it satisfies the following 2 conditions:

* A table is in the 2NF.
* Each non-prime attribute of a table is said to be non-transitively dependent on every key of the table.

**Q #22) What is BCNF in the DBMS?**

**Answer:** BCNF is the **Boyce Codd Normal Form**which is stricter than the 3NF**.**

Any table is said to have in the BCNF if it satisfies the following 2 conditions:

* A table is in the 3NF.
* For each of the functional dependency X->Y that exists, X is the super key of a table.
* **What is De-normalization?**
* As the name indicates, de-normalization is the reverse process of normalization. It’s the controlled introduction of redundancy in to the database design.  
  It helps improve the query performance as the number of joins could be reduced.