**Basic Interview Questions**

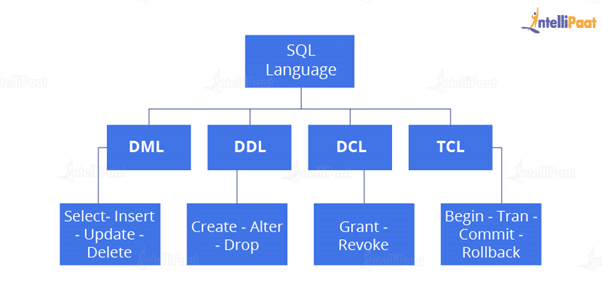
**1. State the differences between HAVING and WHERE clauses.**

|  |  |  |
| --- | --- | --- |
| **Basis for Comparison** | **WHERE** | **HAVING** |
| Implemented in | Row operations | Column operations |
| Applied to | A single row | The summarized row or groups |
| Used for | Fetching specific data from specific rows according to the given condition | Fetching the entire data and separating according to the given condition |
| Aggregate functions | Cannot have them | Can have them |
| Statements | Can be used with SELECT, UPDATE, and DELETE | Cannot be used without a SELECT statement |
| GROUP BY clause | Comes after the WHERE clause | Comes before the HAVING clause |

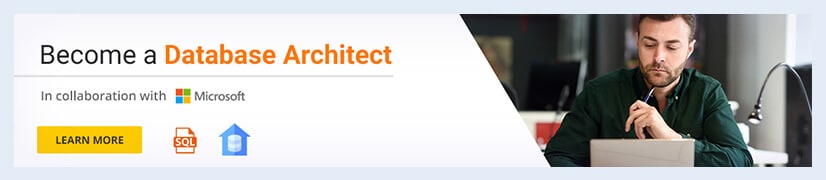
**2. What is SQL?**

SQL stands for ‘Structured Query Language’ and is used for communicating with the databases. According to ANSI, SQL is the standard query language for Relational Database Management Systems (RDBMS) that is used for maintaining them and also for performing different operations of data manipulation on different types of data. Basically, it is a database language that is used for the creation and deletion of databases, and it can be used to fetch and modify the rows of a table and also for multiple other things.

**3. Explain the different types of SQL commands.**



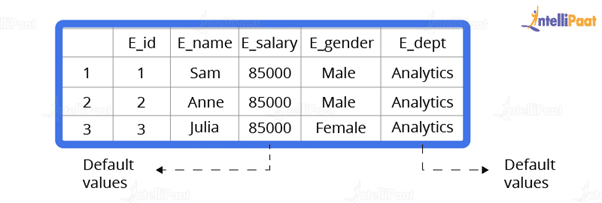
* **Data Definition Language**: DDL is that part of SQL which defines the data structure of the database in the initial stage when the database is about to be created. It is mainly used to create and restructure database objects. Commands in DDL are:
  + Create table
  + Alter table
  + Drop table
* **Data Manipulation Language**: DML is used to manipulate the already existing data in the database. That is, it helps users retrieve and manipulate the data. It is used to perform operations like inserting data into the database through the **insert**command, updating the data with the **update** command, and deleting the data from the database through the **delete** command.
* **Data Control Language:** DCL is used to control access to the data in the database. DCL commands are normally used to create objects related to user access and also to control the distribution of privileges among users. The commands that are used in DCL are **Grant**and **Revoke**.
* **Transaction Control Language:**It is used to control the changes made by DML commands. It also authorizes the statements to assemble in conjunction into logical transactions. The commands that are used in TCL are **Commit**, **Rollback**, **Savepoint**, **Begin**, and **Transaction**.

[](https://intellipaat.com/database-architect-training/)

**4. What is a default constraint?**

Constraints are used to specify some sort of rules for processing data and limiting the type of data that can go into a table. Now, let’s understand the default constraint.

**Default constraint**: It is used to define a default value for a column so that the default value will be added to all the new records if no other value is specified. For example, if we assign a default constraint for the E\_salary column in the below table and set the default value as 85000, then all the entries of this column will have a default value of 85000 unless no other value has been assigned during the insertion.



Now, let’s see how to set a default constraint. We will start off by creating a new table and adding a default constraint to one of its columns.

**Code**:

create table stu1(s\_id int, s\_name varchar(20), s\_marks int default 50)

select \*stu1

**Output**:

Output

Now, we will insert the records.

**Code**:

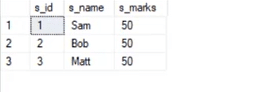
insert into stu1(s\_id,s\_name) values(1,’Sam’)

insert into stu1(s\_id,s\_name) values(2,’Bob’)

insert into stu1(s\_id,s\_name) values(3,’Matt’)

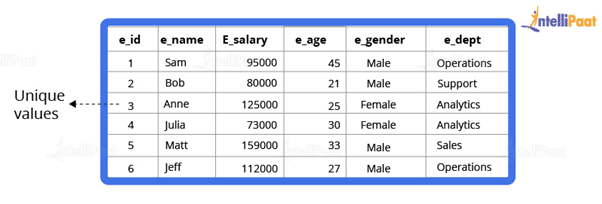
select \*from stu1

**Output**:



**5. What is a unique constraint?**

**Unique constraints** ensure that all the values in a column are different. For example, if we assign a unique constraint to the e\_name column in the below table, then every entry in this column should have a unique value.



First, we will create a table.

create table stu2(s\_id int unique, s\_name varchar(20))

Now, we will insert the records.

insert into stu2 values(1,’Julia’)

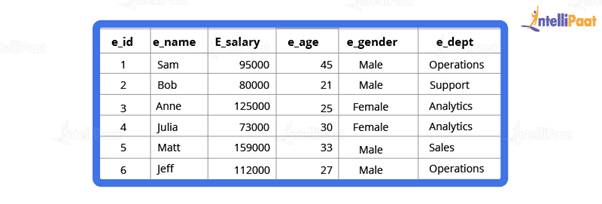
insert into stu2 values(2,’Matt’)

insert into stu2 values(3,’Anne’)

**Output**:



**6. How would you find the second highest salary from the below table?**

  
**Code**:

select \* from employee

select max(e\_salary) from employee where e\_salary not in (select max(e\_salary) from employee)

**Output**:

output 4

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

A primary key is used to uniquely identify all table records. It cannot have NULL values, and it must contain unique values. A table can have only one primary key that consists of single or multiple fields.

Now, we will write a query for demonstrating the use of a primary key for the Employee table:

//

CREATE TABLE Employee (

ID int NOT NULL,

Employee\_name varchar(255) NOT NULL,

Employee\_designation varchar(255),

Employee\_Age int,

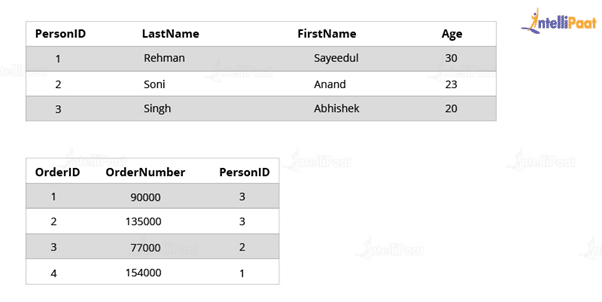
PRIMARY KEY (ID)

);

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

A foreign key is an attribute or a set of attributes that references to the primary key of some other table. So, basically, it is used to link together two tables.

Let’s create a foreign key for the below table:



CREATE TABLE Orders (

OrderID int NOT NULL,

OrderNumber int NOT NULL,

PersonID int,

PRIMARY KEY (OrderID),

FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)

)

**9. What is an Index?**

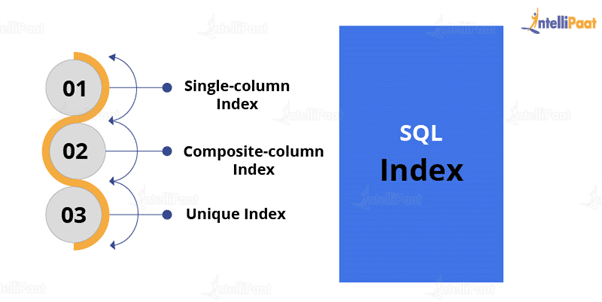
Indexes help speed up searching in the database. If there is no index on any column in the WHERE clause, then the SQL server has to skim through the entire table and check each and every row to find matches, which might result in slow operation on large data.

Indexes are used to find all rows matching with some columns and then to skim through only those subsets of the data to find the matches.

**Syntax**:

CREATE INDEX INDEX\_NAME ON TABLE\_NAME (COLUMN)

**10. Explain the types of Indexes.**

  
**Single-column Indexes**: A single-column index is created for only one column of a table.

**Syntax**:

CREATE INDEX index\_name

ON table\_name(column\_name);

**Composite-column Indexes**: A composite-column index is an index created for two or more columns of the table.

**Syntax**:

CREATE INDEX index\_name

ON table\_name (column1, column2)

**Unique Indexes**: Unique indexes are used for maintaining the data integrity of the table. They don’t allow multiple values to be inserted into the table.

**Syntax**:

CREATE UNIQUE INDEX index

ON table\_name(column\_name)

Now, let’s move on to the next question in this ‘Top SQL Interview Questions’ blog.

**Intermediate Interview Questions**

**11. State the differences between Clustered and Non-clustered indexes.**

* **Clustered index**: It is used to sort the rows of data by their key values. A clustered index is like the contents of a phone book. We can open the book at ‘David’ (for ‘David, Thompson’) and find information for all Davids right next to each other. Since the data is located next to each other, it helps a lot in fetching data based on range-based queries. Also, the clustered index is actually related to how the data is stored. There is only one clustered index possible per table.
* **Non-clustered index**: It stores data at one location and indexes at some other location. The index has pointers that point to the location of the data. As the index in the non-clustered index is stored in different places, there can be many non-clustered indexes for a table.

Now, we will see the major differences between clustered and non-clustered indexes:

| **Parameters** | **Clustered Index** | **Non-clustered Index** |
| --- | --- | --- |
| Used for | Sorting and storing records physically in memory | Creating a logical order for data rows. Pointers are used for physical data files |
| Methods for storing | Stores data in the leaf nodes of the index | Never stores data in the leaf nodes of the index |
| Size | Quite large | Comparatively, small |
| Data accessing | Fast | Slow |
| Additional disk space | Not required | Required to store indexes separately |
| Type of key | By default, the primary key of a table is a clustered index | It can be used with the unique constraint on the table that acts as a composite key |
| Main feature | Improves the performance of data retrieval | Should be created on columns used in Joins |

**12. State the differences between SQL and PL/SQL.**

|  |  |
| --- | --- |
| **SQL** | **PL/SQL** |
| SQL is a database structured query language. | It is a programming language for a database that uses SQL. |
| SQL is an individual query that is used to execute DML and DDL commands. | PL/SQL is a block of codes used to write the entire procedure or a function. |
| SQL is a declarative and data-oriented language. | PL/SQL is a procedural and application-oriented language. |
| It is mainly used for the manipulation of data. | It is used for creating an application. |
| It provides interaction with the database server. | It does not provide interaction with the database server. |
| It cannot contain PL/SQL code in it. | It can contain SQL in it because it is an extension of SQL. |

**13. What do you understand by a Character Manipulation function?**

Character manipulation functions are used for the manipulation of character data types.

Some of the character manipulation functions are:  
**UPPER:**It returns the string in uppercase.

**Syntax**:

UPPER(‘ string’)

**Example**:

SELECT UPPER(‘demo string’) from String;

**Output**:

DEMO STRING

**LOWER:**It returns the string in lowercase.

**Syntax**:

LOWER(‘STRING’)

**Example**:

SELECT LOWER (‘DEMO STRING’) from String

**Output**:

demo string

**INITCAP:** It converts the first letter of the string to uppercase and retains others in lowercase.

**Syntax**:

Initcap(‘sTRING’)

**Example**:

SELECT Initcap(‘dATASET’) from String

**Output**:

Dataset

**CONCAT:**It is used to concatenate two strings.

**Syntax**:

CONCAT(‘str1’,’str2’)

**Example**:

SELECT CONCAT(‘Data’,’Science’) from String

**Output**:

Data Science

**LENGTH:** It is used to get the length of a string.

**Syntax**:

LENGTH(‘String’)

**Example**:

SELECT LENGTH(‘Hello World’) from String

**Output**:

11

**14. What is AUTO\_INCREMENT?**

AUTO\_INCREMENT is used in SQL to automatically generate a unique number whenever a new record is inserted into a table.

Since the primary key is unique for each record, we add this primary field as the AUTO\_INCREMENT field so that it is incremented when a new record is inserted.

The AUTO-INCREMENT value is by default starts from 1 and incremented by 1 whenever a new record is inserted.

**Syntax:**

CREATE TABLE Employee(

Employee\_id int NOT NULL AUTO-INCREMENT,

Employee\_name varchar(255) NOT NULL,

Employee\_designation varchar(255)

Age int,

PRIMARY KEY (Employee\_id)

)

**15. What is the difference between DELETE and TRUNCATE commands?**

* **DELETE**: This query is used to delete or remove one or more existing tables.
* **TRUNCATE**: This statement deletes all the data from inside a table.

**The difference between DELETE and TRUNCATE commands are as follows:**

* TRUNCATE is a DDL command, and DELETE is a DML command.
* With TRUNCATE, we can’t really execute and trigger, while with DELETE we can accomplish a trigger.
* If a table is referenced by foreign key constraints, then TRUNCATE won’t work. So, if we have a foreign key, then we have to use the DELETE command.

**The syntax for the DELETE command**:

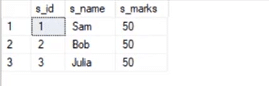
DELETE FROM table\_name

[WHERE condition];

**Example**:

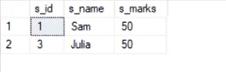
select \* from stu

**Output**:



delete from stu where s\_name=’Bob’

**Output**:



**The syntax for the TRUNCATE command**:

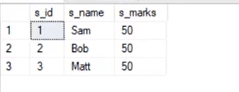
TRUNCATE TABLE

Table\_name;

**Example**:

select \* from stu1

**Output**:



truncate table stu1

**Output**:

output 8

This deletes all the records from the table.

**16. What is COALESCE function?**

COALESCE function takes a set of inputs and returns the first non-null value.

**Syntax**:

COALESCE(val1,val2,val3,……,nth val)

**Example**:

SELECT COALESCE(NULL, 1, 2, ‘MYSQL’)

**Output**:

1

**17. What do you understand by Normalization and Denormalization?**

Normalization and denormalization are basically two methods used in databases.

Normalization is used in reducing data redundancy and dependency by organizing fields and tables in databases. It involves constructing tables and setting up relationships between those tables according to certain rules. The redundancy and inconsistent dependency can be removed using these rules to make it more flexible.

Denormalization is contrary to normalization. In this, we basically add redundant data to speed up complex queries involving multiple tables to join. Here, we attempt to optimize the read performance of a database by adding redundant data or by grouping the data.

**18. What is wrong with the below-given SQL query?**

SELECT gender, AVG(age) FROM employee WHERE AVG(age)>30 GROUP BY gender

When we execute this command, we get the following error:

Msg 147, Level 16, State 1, Line 1

Aggregation may not appear in the WHERE clause unless it is in a subquery contained in a HAVING clause or a select list, the column being aggregated is an outer reference.

Msg 147, Level 16, State 1, Line 1

Invalid column name ‘gender’.

This basically means that whenever we are working with aggregate functions and we are using GROUP BY, we can’t use the WHERE clause. Therefore, instead of the WHERE clause, we should use the HAVING clause.

Also, when we are using the HAVING clause, GROUP BY should come first and HAVING should come next.

select e\_gender, avg(e\_age) from employee group by e\_gender having avg(e\_age)>30

**Output**:

Output 9

**19. What do you know about the stuff() function?**

The stuff function deletes a part of the string and then inserts another part into the string starting at a specified position.

**Syntax**:

STUFF(String1, Position, Length, String2)

Here, **String1** is the one that would be overwritten. **Position** indicates the starting location for overwriting the string. **Length**is the length of the substitute string, and **String2**is the string that would overwrite String1.

**Example**:

select stuff(‘SQL Tutorial’,1,3,’Python’)

This will change ‘SQL Tutorial’ to ‘Python Tutorial’

**Output**:

Python Tutorial

**20. What are Views? Give an example.**

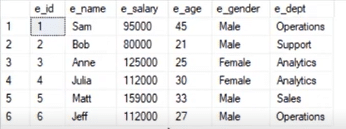
Views are virtual tables used to limit the tables that we want to display, and these are nothing but the result of a SQL statement that has a name associated with it. Since views are not virtually present, they take less space to store.

Let’s consider an example. In the below employee table, say, we want to perform multiple operations on the records with gender ‘Female’. We can create a view-only table for the female employees from the entire employee table.

Now, let’s implement it on the SQL server.

Below is our employee table:

select \* from employee



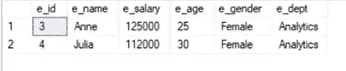
Now, we will write the syntax for view.

**Syntax**:

create view female\_employee as select \* from employee where e\_gender=’Female’

select \* from female\_employee

**Output**:



**21. What is a stored procedure? Give an example.**

A stored procedure is a prepared SQL code that can be saved and reused. In other words, we can consider a stored procedure to be a function consisting of many SQL statements to access the database system. We can consolidate several SQL statements into a stored procedure and execute them whenever and wherever required.

A stored procedure can be used as a means of modular programming, i.e., we can create a stored procedure once, store it, and call it multiple times as required. This also supports faster execution when compared to executing multiple queries.

**Syntax**:

CREATE PROCEDURE procedure\_name

AS

Sql\_statement

GO;

To execute we will use this:

EXEC procedure\_name

**Example**:

We are going to create a stored procedure that will help extract the age of the employees.

create procedure employee\_age

as

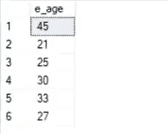
select e\_age from employee

go

Now, we will execute it.

exec employee\_age

**Output**:



**22. What do you know about Joins? Define different types of Joins.**

The Join clause is used to combine rows from two or more tables based on a related column between them. There are various types of Joins that can be used to retrieve data, and it depends upon the relationship between tables.

There are four types of Joins:

* **Inner Join**: Inner Join basically returns records that have matching values in both tables.
* **Left Join**: Left Join returns rows that are common between the tables and all the rows of the left-hand-side table, i.e., it returns all the rows from the left-hand-side table even if there are no matches available in the right-hand-side table.
* **Right Join:**Right Join returns rows that are common between the tables and all the rows of the right-hand-side table, i.e., it returns all the rows from the right-hand-side table even if there are no matches available in the left-hand-side table.
* **Full Join:**Full Join returns all the rows from the left-hand-side table and all the rows from the right-hand-side table.

**23. Explain Inner Join.**

Inner Join basically gives us those records that have matching values in two tables.

Let us suppose, we have two tables Table A and Table B. When we apply Inner Join on these two tables, we will get only those records that are common to both Table A and Table B.

**Syntax**:

SELECT columns

FROM table1

INNER JOIN table2

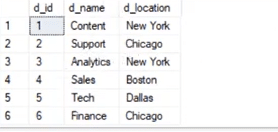
ON table1.column\_x=table2.column\_y;

**Example**:

select \* from employee

select \* from department

**Output**:



Now, we would have Inner Join in both of these tables, where the ‘e\_dept’ column in the employee table is equal to the ‘d\_name’ column of the department table.

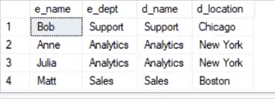
**Syntax**:

select employee.e\_name, employee.e\_dept, department.d\_name, department.d\_location

from employee inner join department

on

employee.e\_dept=department.d\_name

**Output**:  


After Inner Join, we have only those records where the departments match in both tables. As we can see, the matched departments are Support, Analytics, and Sales.

**24. State the differences between Views and Tables.**

|  |  |
| --- | --- |
| **Views** | **Tables** |
| It is a virtual table that is extracted from a database. | A table is structured with a set number of columns and a boundless number of rows. |
| Views do not hold data themselves. | Table contains data and stores the data in databases. |
| A view is also utilized to query certain information contained in a few distinct tables. | A table holds fundamental client information and the cases of a characterized object. |
| In a view, we will get frequently queried information. | In a table, changing the information in the database changes the information that appears in the view |

**25. What do you understand by a Temporary Table? Write a query to create a Temporary Table.**

A temporary table helps us store and process intermediate results. These temporary tables are created and can be automatically deleted when they are no longer used. They are very useful in places where we need to store temporary data.

**Syntax**:

CREATE TABLE #table\_name();

The below query will create a temporary table:

create table #book(b\_id int, b\_cost int)

Now, we will insert the records.

insert into #book values(1,100)

insert into #book values(2,232)

select \* from #book

**Output**:  
output 15

**26. Explain the difference between OLTP and OLAP.**

**OLTP:**It basically stands for **Online Transaction Processing** and we can consider it to be a category of software applications that is efficient for supporting transaction-oriented programs. One of the important attributes of the OLTP system is its potentiality to keep up the consistency.

The OLTP system often follows decentralized planning to keep away from single points of failure. This system is generally designed for a large audience of end-users to perform short transactions. Also, queries involved in such databases are generally simple, need fast response time, and in comparison, it returns only a few records. So, the number of transactions per second acts as an effective measure for those systems.

**OLAP:** OLAP stands for **Online Analytical Processing** and it is a category of software programs that are identified by a comparatively lower frequency of online transactions. For OLAP systems, the efficacy computing depends highly on the response time. Hence, such systems are generally used for data mining or maintaining aggregated historical data, and they are usually used in multi-dimensional schemas.

**27. What do you understand by Self Join?**

Self Join in SQL is used for joining a table with itself. Here, depending upon some conditions, each row of the table is joined with itself and with other rows of the table.

**Syntax**:

SELECT a.column\_name, b.column\_name

FROM table a, table b

WHERE condition

**Example**:  
Consider the customer table given below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Age** | **Address** | **Salary** |
| 1 | Anand | 32 | Ahmedabad | 2,000.00 |
| 2 | Abhishek | 25 | Delhi | 1,500.00 |
| 3 | Shivam | 23 | Kota | 2,000.00 |
| 4 | Vishal | 25 | Mumbai | 6,500.00 |
| 5 | Sayeedul | 27 | Bhopal | 8,500.00 |
| 6 | Amir | 22 | MP | 4,500.00 |
| 7 | Arpit | 24 | Indore | 10,000.00 |

We will now join the table using Self Join:

SQL> SELECT a.ID, b.NAME, a.SALARY

FROM CUSTOMERS a, CUSTOMERS b

WHERE a.SALARY < b.SALARY;

**Output**:

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Salary** |
| 2 | Anand | 1,500.00 |
| 2 | Abhishek | 1,500.00 |
| 1 | Vishal | 2,000.00 |
| 2 | Vishal | 1,500.00 |
| 3 | Vishal | 2,000.00 |
| 6 | Vishal | 4,500.00 |
| 1 | Sayeedul | 2,000.00 |
| 2 | Sayeedul | 1,500.00 |
| 3 | Sayeedul | 2,000.00 |
| 4 | Sayeedul | 6,500.00 |
| 6 | Sayeedul | 4,500.00 |
| 1 | Amir | 2,000.00 |
| 2 | Amir | 1,500.00 |
| 3 | Amir | 2,000.00 |
| 1 | Arpit | 2,000.00 |
| 2 | Arpit | 1,500.00 |
| 3 | Arpit | 2,000.00 |
| 4 | Arpit | 6,500.00 |
| 5 | Arpit | 8,500.00 |
| 6 | Arpit | 4,500.00 |

**28. What is the difference between Union and Union All operators?**

The **Union** operator is used to combine the result set of two or more select statements. For example, the first select statement returns the fish shown in Image A, and the second returns the fish shown in Image B. Then, the Union operator will return the result of the two select statements as shown in Image A U B. Also, if there is a record present in both tables, then we will get only one of them in the final result.

**Syntax**:

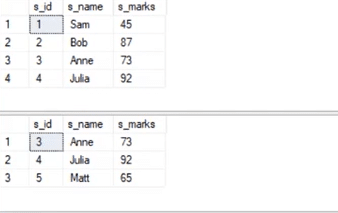
SELECT column\_list FROM table1

**Union**:

SELECT column\_list FROM table2

Now, we will execute it in the SQL server.

These are the two tables in which we will use the Union operator.

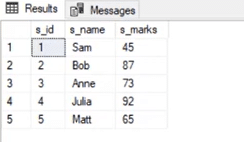


select \* from student\_details1

**Union**:

select \* from student\_details2

**Output**:



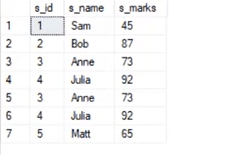
Now, **Union All** gives all the records from both tables including the duplicates.

**Syntax**:

select \* from student\_details1

**Union All**:

select \* from student\_details2

**Output**:  


**29. What is the use of the Intersect operator?**

The **Intersect** operator helps combine two select statements and returns only those records that are common to both the select statements. So, after we get Table A and Table B over here and if we apply the Intersect operator on these two tables, then we will get only those records that are common to the result of the select statements of these two.

**Syntax**:

SELECT column\_list FROM table1

INTERSECT

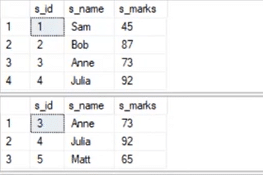
SELECT column\_list FROM table2

Now, let’s see an example for the INTERSECT operator.

select \* from student\_details1

select \* from student\_details1

**Output**:



select \* from student\_details1

intersect

select \* from student\_details2

**Output**:



**30. How can you copy data from one table into another?**

We have to copy this data into another table. For this purpose, we can use the INSERT INTO SELECT operator. Before we go ahead and do that, we would have to create another table that would have the same structure as the above-given table.

**Syntax**:

create table employee\_duplicate(

e\_id int,

e\_name varchar(20),

e\_salary int,

e\_age int,

e\_gender varchar(20)

e\_dept varchar(20)

)

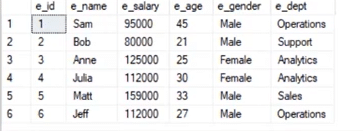
For copying the data, we would use the following query:

insert into employee\_duplicate select \* from employees

Let us have a glance at the copied table.

select \* from employee\_duplicate

**Output**:



**31. Describe how to delete duplicate rows using a single statement but without any table creation.**

Let’s create an Employee table where column names are ID, NAME, DEPARTMENT, and EMAIL. Below are the SQL scripts for generating the sample data:

CREATE TABLE EMPLOYEE

(

ID INT,

NAME Varchar(100),

DEPARTMENT INT,

EMAIL Varchar(100)

)

INSERT INTO EMPLOYEE VALUES (1,'Tarun',101,'tarun@intellipaat.com')

INSERT INTO EMPLOYEE VALUES (2,'Sabid',102,'sabid@intellipaat.com')

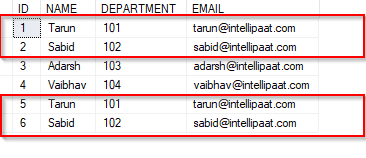
INSERT INTO EMPLOYEE VALUES (3,'Adarsh',103,'adarsh@intellipaat.com')

INSERT INTO EMPLOYEE VALUES (4,'Vaibhav',104,'vaibhav@intellipaat.com')

--These are the duplicate rows

INSERT INTO EMPLOYEE VALUES (5,'Tarun',101,'tarun@intellipaat.com')

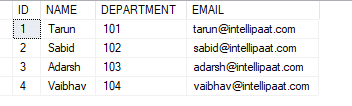
INSERT INTO EMPLOYEE VALUES (6,'Sabid',102,'sabid@intellipaat.com')



We can see the duplicate rows in the above table.

DELETE e1 FROM EMPLOYEE e1, EMPLOYEE e2 WHERE e1.name = e2.name AND e1.id > e2.id

The SQL query above will delete the rows, where the name fields are duplicated, and it will retain only those unique rows in which the names are unique and the ID fields are the lowest. That is, rows with IDs 5 and 6 are deleted, whereas rows with IDs 1 and 2 are retained.



**32. Can you identify the employee who is having the third-highest salary from the given Employee table (with salary-related data)?**

Consider the below Employee table. In the table, ‘Sabid’ has the third-highest salary (600000).

|  |  |
| --- | --- |
| **Name** | **Salary** |
| Tarun | 70000 |
| Sabid | 60000 |
| Adarsh | 30000 |
| Vaibhav | 80000 |

Below is a simple query to find out the employee who has the third-highest salary. The functions RANK, DENSE RANK, and ROW NUMBER are used to obtain the increasing integer value by imposing the ORDER BY clause in the SELECT statement, based on the ordering of rows. The ORDER BY clause is necessary when we use RANK, DENSE RANK, or ROW NUMBER functions. On the other hand, the PARTITION BY clause is optional.

WITH CTE AS

(

SELECT Name, Salary, RN = ROW\_NUMBER() OVER (ORDER BY Salary DESC) FROM EMPLOYEE

)

SELECT Name, Salary FROM CTE WHERE RN =3

**33. What is the difference between HAVING and WHERE clauses?**

The distinction between HAVING and WHERE clauses in SQL is that while the WHERE clause cannot be used with aggregates, we use the HAVING clause with the aggregated data. The WHERE clause works on the data from a row and not with the aggregated data.

Let’s consider the Employee table below.

|  |  |  |
| --- | --- | --- |
| **Name** | **Department** | **Salary** |
| Tarun | Production | 50000 |
| Tarun | Testing | 60000 |
| Sabid | Marketing | 70000 |
| Adarsh | Production | 80000 |
| Vaibhav | Testing | 90000 |

The following would select the data on a row-by-row basis:

SELECT Name, Salary FROM Employee WHERE Salary >=50000

**Output:**

|  |  |
| --- | --- |
| **Name** | **Salary** |
| Tarun | 50000 |
| Tarun | 60000 |
| Sabid | 70000 |
| Adarsh | 80000 |
| Vaibhav | 90000 |

The HAVING clause, on the other hand, operates on aggregated results.

SELECT Department, SUM(Salary) AS total FROM Employee GROUP BY Department

**Output:**

|  |  |
| --- | --- |
| **Department** | **Total** |
| Marketing | 70000 |
| Production | 130000 |
| Testing | 150000 |

Now, let’s see the output when we apply HAVING in the above query.

SELECT Department, SUM(Salary) AS total FROM Employee GROUP BY Department HAVING SUM(Salary)>70000

**Output:**

|  |  |
| --- | --- |
| **Department** | **Total** |
| Production | 130000 |
| Testing | 150000 |

**34. Explain database white box testing and black box testing.**

The **white box test** method mainly deals with the internal structure of a particular database, where users hide specification details. The white box testing method involves the following:

* As the coding error can be detected by testing the white box, it can eliminate internal errors.
* To check for the consistency of the database, it selects the default table values.
* This method verifies the referential integrity rule.
* It helps perform the module testing of database functions, triggers, views, and SQL queries.

The **black box test** method generally involves interface testing, followed by database integration. It includes:

* Mapping details
* Verification of the incoming data
* Verification of the outgoing data from the other query functions

**1. Compare between SQL and PL/SQL.**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **SQL** | **PL/SQL** |
| What is it? | A single query or command execution | A full programming language |
| What does it comprise? | The data source for reports, web pages, etc. | An application language to build, format, and display reports, web pages, etc. |
| Characteristic | Declarative in nature | Procedural in nature |
| Used for | Manipulating data | Creating applications |

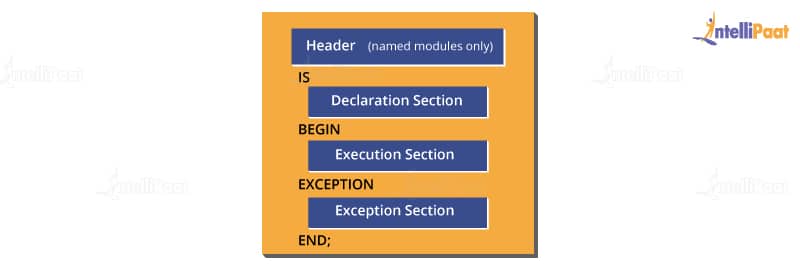
**2. What is PL/SQL?**

**Oracle PL/SQL** is a procedural language that has both interactive SQL and procedural programming language constructs such as iteration and conditional branching.

***Go through this***[***PL/SQL Tutorial***](https://intellipaat.com/blog/tutorial/oracle-plsql-tutorial/overview-of-plsql/)***to learn ‘What is PL/SQL?’***

**3. What is the basic structure of PL/SQL?**

PL/SQL uses a block structure as its basic structure. Anonymous blocks or nested blocks can be used in PL/SQL.



**4. Explain the uses of a database trigger.**

A PL/SQL program unit associated with a particular database table is called a database trigger. It is used for:

* Audit data modifications
* Log events transparently
* Enforce complex business rules
* Maintain replica tables
* Derive column values
* Implement Complex security authorizations

Any constant, variable, or parameter has a data type depending on which the storage constraints, format, and the range of values and operations are determined.

**5. How is a process of PL/SQL compiled?**

The compilation process includes syntax check, bind, and p-code generation processes. Syntax checking checks the PL/SQL codes for compilation errors. When all errors are corrected, a storage address is assigned to the variables that hold data. It is called Binding. P-code is a list of instructions for the PL/SQL engine. P-code is stored in the database for named blocks and is used the next time it is executed.

**6. What does a PL/SQL package consist of?**

A PL/SQL package consists of:

* PL/SQL table and record TYPE statements
* Procedures and functions
* Cursors
* Variables ( tables, scalars, records, etc.) and constants
* Exception names and pragmas for relating an error number with an exception
* Cursors

**7. What are the benefits of PL/SQL packages?**

PL/SQL packages provide several benefits as follows:

* **Enforced information hiding:** It offers the liberty to choose whether to keep data private or public.
* **Top-down design:** We can design the interface to the code hidden in the package before we actually implemented the modules.
* **Object persistence:** Objects declared in a package specification behave like global data for all PL/SQL objects in the application. We can modify the package in one module and then reference those changes to another module.
* **Object-oriented design:**The package gives developers stronghold over how the modules and data structures inside the package can be used.
* **Guaranteeing transaction integrity:**It provides a level of transaction integrity.
* **Performance improvement:**The RDBMS automatically tracks the validity of all program objects stored in the database and enhance the performance of packages.

**8. What are different methods to trace the PL/SQL code?**

Tracing the code is a crucial technique to measure its performance during the runtime. Different methods for tracing the code includes:

* DBMS\_APPLICATION\_INFO
* DBMS\_TRACE
* DBMS\_SESSION and DBMS\_MONITOR
* trcsess and tkprof utilities

**9. What is the difference between functions, procedures, and packages in PL/SQL?**

* **Function**: The main purpose of a PL/SQL function is to compute and return a single value. A function has a return type in its specification and must return a value specified in that type.
* **Procedure**: A procedure does not have a return type and should not return any value, but it can have a return statement that simply stops its execution and returns to the caller. A procedure is used to return multiple values; otherwise, it is generally similar to a function.
* **Package**: A package is a schema object which groups logically related PL/SQL types, items, and subprograms. You can also say that it is a group of functions, procedures, variables, and record TYPE statement. It provides modularity, due to which it aids application development. It is used to hide information from unauthorized users.

**10. What is a stored procedure?**

A **stored procedure** is a sequence of statements or a named PL/SQL block that performs one or more specific functions. It is similar to a procedure in other programming languages. It is stored in the database and can be repeatedly executed. It is stored as a schema object. It can be nested, invoked, and parameterized.

**11. What is a cursor? Why is it required?**

A **cursor** is a temporary work area created in system memory when a SQL statement is executed. A cursor contains information on a select statement and the row of data accessed by it. This temporary work area stores the data, retrieved from the database, to manipulate it. A cursor can hold more than one row but can process only one row at a time. A cursor is required to process rows individually for queries.

**12. Explain the Day-to-day activities in PL/SQL.**

1. Create database objects—tables, synonyms, sequences, etc.
2. To implement business rules, create procedures, functions, etc.
3. To impose business rules, create constraints, triggers, etc.
4. For data manipulation, create cursors

**13. How to display records having the maximum salary from an employee table?**

Select \* from emp where sal= (select max(sal) from emp)

**14. How to display the highest salary from an employee table?**

Use the following code for displaying the highest salary from an employee table:

Select max(sal) from emp;

**15. How to display the second highest salary from an employee table?**

Select max(sal) from emp where sal not in ( select max(sal) from emp

**16. What is a Join?**

**Join** is a keyword used to query data from multiple tables based on the relationship between the fields of tables. Keys play a major role in Joins.

**17. what is a View?**

* A **View** is a virtual table consisting of data contained in a table.
* Views do not need any memory space.
* Views can be created on multiple tables.

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

A **subquery** is a query within another query. The outer query is known as the main query and the inner query is called the subquery. A subquery is executed first, and the result of the subquery is passed to the main query.

There are two types of subqueries: correlated and non-correlated

**19. what is a trigger?**

A **trigger** is a database object that automatically executes in response to some events on the tables or views. It is used to apply the integrity constraint to the database objects.

**1. Compare MySQL vs SQL Server.**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **MySQL** | **SQL Server** |
| Developed by | Oracle | Microsoft |
| Programmed in | C and C++ | Mainly C++, but some parts in C |
| Platforms | Supports many platforms | Supports only Linux and Windows |
| Syntax | Complex Syntax | Simpler and easy-to-use syntax |

**2. What is SQL Server?**

SQL Server is one of the database management systems (DBMS) and is designed by Microsoft.  DBMS are computer software applications with the capability of interacting with users, various other applications, and databases. The objective of SQL Server is capturing and analyzing data and managing the definition, querying, creation, updating, and administration of the database.

**3. How and why use SQL Server?**

SQL Server is free and anyone can download and use it. The application uses SQL (Structured Query Language), and it is easy to use.

**4. What are the features of MySQL?**

MySQL provides cross-platform support, a wide range of interfaces for application programming, and has many stored procedures like triggers and cursors that help in managing the database.

**5. What is the Traditional Network Library for a system?**

In either Windows or POSIX systems, the named pipes provide ways of inter-process communications to connect different processes running on the same machine. It dispenses with the necessity of using the network stack, and data can be sent without affecting the performance. Servers set up named pipes to listen to requests. The client process needs to know the specific pipe name to send the request.

**6. What is the default port for MySQL Server?**

The default port for MySQL Server is 3306. Another standard default port is 1433 in TCP/IP for SQL Server.

**7. What do DDL, DML, and DCL stand for?**

DDL is the abbreviation for Data Definition Language dealing with database schemas, as well as the description of how data resides in the database. An example of this is the CREATE TABLE command. DML denotes Data Manipulation Language which includes commands such as SELECT, INSERT, etc. DCL stands for Data Control Language and includes commands like GRANT, REVOKE, etc.

**8. What is a join in MySQL?**

In MySQL, joins are used to query data from two or more tables. The query is made using the relationship between certain columns existing in the table. There are four types of joins in MySQL.

Inner join returns rows if there is at least one match in both tables. Left join returns all the rows from the left table even if there is no match in the right table. Right join returns all the rows from the right table even if no matches exist in the left table. Full join would return rows when there is at least one match in the tables.

**9. What are the common MySQL functions?**

Common MySQL functions are as follows:

* **NOWO:** The function for returning the current date and time as a single value
* **CURRDATEO:** The function for returning the current date or time
* **CONCAT (X, Y):** The function to concatenate two string values creating a single string output
* **DATEDIFF (X, Y):** The function to determine the difference between two dates

**10. What is the difference between CHAR and VARCHAR?**

When a table is created, CHAR is used to define the fixed length of the table and columns. The length value could be in the range of 1–255. The VARCHAR command is used to adjust the column and table lengths as required.

**11. What are Heap Tables?**

Basically, Heap tables are in-memory tables used for high-speed temporary storage. But, TEXT or BLOB fields are not allowed within them. They also do not support AUTO INCREMENT.

**12. What is the syntax for concatenating tables in MySQL?**

The syntax for concatenating tables is MySQL:

CONCAT (string 1, string 2, string 3)

**13. What is the limit of indexed columns that can be created for a table?**

The maximum limit of indexed columns that can be created for any table is 16.

**14. What are the different types of strings used in database columns in MySQL?**

In MySQL, the different types of strings that can be used for database columns are SET, BLOB, VARCHAR, TEXT, ENUM, and CHAR.

**15. How can a user get the current SQL version?**

The syntax for getting the current version of MySQL:

SELECT VERSION ();

**16. Is there an object-oriented version of MySQL library functions?**

Yes. MySQLi is the object-oriented version of MySQL, and it interfaces in PHP.

**17. What is the storage engine used for MySQL?**

Storage tables are named as table types. The data is stored in the files using multiple techniques such as indexing, locking levels, capabilities, and functions.

**18. What is the difference between the primary key and the candidate key?**

The primary key in MySQL is used to identify every row of a table in a unique manner. For one table, there is only one primary key. The candidate keys can be used to reference the foreign keys. One of the candidate keys is the primary key.

**19. What are the different types of tables in MySQL?**

MyISAM is the default table that is based on the sequential access method.

* **Heap** is the table that is used for fast data access, but the data will be lost if the table or the system crashes.
* **InnoDB** is the table that supports transactions using the COMMIT and ROLLBACK commands.
* **BDB** can support transactions similar to InnoDB, but the execution is slower.

**20. Can you use MySQL with Linux operating system?**

Yes. The syntax for using MySQL with Linux operating system is as follows:

etc/init.d/mysqlstart

**21. What is the use of ENUM in MySQL?**

The use of ENUM will limit the values that can go into a table. For instance, a user can create a table giving specific month values and other month values would not enter into the table.

**22. 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

**23. What is the difference between LIKE and REGEXP operators in MySQL?**

LIKE is denoted using the ‘%’ sign. For example:

SELECT \* FROM user WHERE user name LIKE “%NAME”

On the other hand, the use of REGEXP is as follows:

SELECT \* FROM user WHERE username REGEXP “^NAME”;

**24. How to use the MySQL slow query log?**

Information that is provided on the slow query log could be huge in size. The query could also be listed over a thousand times. In order to summarize the slow query log in an informative manner, one can use the third-party tool ‘pt-query-digest’.

**25. How can one take an incremental backup in MySQL?**

A user can take an incremental backup in MySQL using Percona XtraBackup.

**26. How can you change the root password if it is lost?**

In such cases when the password is lost, the user should start the DB with skip-grants-table and then change the password. Thereafter, with the new password, the user should restart the DB in a normal mode.

**27. How to resolve the problem of the data disk that is full?**

When the data disk is full and overloaded, the way out is to create and soft link and move the .frm and the .idb files into that link location.

**28. What is the difference between the DELETE TABLE and TRUNCATE TABLE commands in MySQL?**

Basically, DELETE TABLE is a logged operation, and every row deleted is logged. Therefore, the process is usually slow. TRUNCATE TABLE also deletes rows in a table, but it will not log any of the rows deleted.  The process is faster here in comparison. TRUNCATE TABLE can be rolled back and is functionally similar to the DELETE statement without a WHERE clause.

**29. What are the types of joins in MySQL?**

There are four types of joins in MySQL. Inner join returns the rows if there is at least one match in two tables. Left join returns all the rows from the left table even if there is no match in the right table. Right join returns all the rows from the right table even if no matches exist in the left table. Full join would return rows when there is at least one match in the tables.

**30.What are the storage models of OLAP?**

The storage models in OLAP are MOLAP, ROLAP, and HOLAP.

**31. How to define the testing of network layers in MySQL?**

For this, it is necessary to review the layered architecture and determine hardware and software configuration dependencies with respect to the application put to test.

**32. What is the difference between primary key and unique key?**

While both are used to enforce the uniqueness of the column defined, the primary key would create a clustered index, whereas the unique key would create a non-clustered index on the column. The primary key does not allow ‘NULL’, but the unique key does.

**33. What is meant by transaction? What are ACID properties?**

Transaction is a logical unit of work where either all or none of the steps should be performed. ACID is the abbreviation for Atomicity, Consistency, Isolation, and Durability that are properties of any transaction.

**34. How can one restart SQL Server in the single user or the minimal configuration modes?**

The command line SQLSERVER.EXE used with ‘–m’ will restart SQL Server in the single user mode and the same with ‘–f’ will restart it in the minimal configuration mode.

**35. What is the difference between BLOB and TEXT?**

BLOBs are binary large object holding huge data. Four types of BLOBs are TINYBLOB, BLOB, MEDIBLOB, and LONGBLOB. TEXT is a case-sensitive BLOB. Four types of TEXT are TINY TEXT, TEXT, MEDIUMTEXT, and LONG TEXT.

**36. What is the basic MySQL architecture?**

The logical architecture of MySQL is made of ‘connection manager’, ‘query optimizer’, and ‘pluggable engines’.

**37. What are the advantages and disadvantages of using MySQL?**

There are various advantages and disadvantages of using MySQL. Some of them are given below:

**Advantages**

* MySQL helps in the secure management of databases. By using it, we can securely execute database transactions.
* It is fast and efficient in comparison to other database management systems as it supports varieties of storage engines.
* As its transaction processing is high, MySQL can execute millions of queries.

Besides, some of the features that make MySQL unique are deadlock identification, execution of multiple transactions, efficient processing, and easy management.

**Disadvantages**

* Scalability in MySQL is a redundant task.
* MySQL serves good for large databases mostly.
* There are issues of the instability of software.

**38. What are the differences between a primary key and a foreign key?**

|  |  |
| --- | --- |
| **Primary Key** | **Foreign Key** |
| It helps in the unique identification of data in a database | It helps establish a link between tables |
| There can be only one primary key for a table | There can be more than one foreign key for a table |
| Primary key attributes cannot have duplicate values in a table | Duplicate values are acceptable for a foreign key |
| Null values are not acceptable | Null values are acceptable |
| We can define primary key constraints for temporarily created tables | It cannot be defined for temporary tables |
| The primary key index is automatically created | The index is not created automatically |

**39. What is the TIMESTAMP data type?**

Timestamp in SQL Server helps in row versioning. Row versioning is a type of concurrency that allows retaining the value until it is committed in the database. It shows the instant time of any event. It consists of both the date and time of the event. Also, timestamp helps in backing up data during the failure of a transaction.

While we insert, update, or delete a record, the date and time automatically get inserted.

* Format of timestamp: YYYY-MM-DD HH:MM: SS
* Range of timestamp: “1970-01-01 00:00:01” UTC to “2038-01-19 03:14:07” UTC

**40. What is the function of mysqldump?**

As the name suggests, mysqldump is used to dump one or more created databases. It performs backups for data or transfers the data from SQL Server to another. Also, it helps in producing the initial database schema by logical backups. Moreover, unlike triggers, mysqldump does not backup the stored procedures or functions by default.

**Syntaxes**

For a single database:

mysqldump [options] db\_name [tables]

For multiple databases:

mysqldump [options] –databases db1 [db2 db3...]

For all databases:

mysqldump [options] –all-databases

**41. What is an access control list?**

Every organization has some crucial data specific to its business. This data needs secure access so that any consequence due to data loss can be avoided. For this, organizations create a sequence of permissions that are linked to various data objects. These lists are known as the access control list (ACL).

ACL serves the basis for the server’s security that helps troubleshoot the connection problems for users. These are also known as grant tables that are cached by MySQL. MySQL verifies a user for authentication and grants permissions in a sequence when the user executes a command.