**SQL QA**

1. **How do you retrieve all records from a table named "employees" in SQL?**

Use the SELECT statement without any specific conditions.

SELECT \* FROM employees;

1. **What is the difference between the SQL statements SELECT and UPDATE?**

The SELECT statement is used to retrieve data from one or more database tables. It doesn't modify the data in the tables; instead, it reads and presents the data to the user or application.

SELECT first\_name, last\_name FROM employees WHERE department = 'HR';

The UPDATE statement is used to modify existing data in a database table. It allows you to change the values of one or more columns in existing rows based on specified conditions. UPDATE employees SET salary = 55000 WHERE employee\_id = 101;

1. **How do you perform a SQL JOIN operation between two tables?**

Performing a SQL JOIN operation between two tables allows you to combine data from both tables based on a related column. SQL offers different types of JOINs, including INNER JOIN, LEFT JOIN (or LEFT OUTER JOIN), RIGHT JOIN (or RIGHT OUTER JOIN), and FULL OUTER JOIN. Here, I'll explain how to use an INNER JOIN as it's one of the most commonly used JOIN operations.

INNER JOIN

SELECT employees.employee\_id, employees.first\_name, employees.last\_name, departments.department\_name

FROM employees

INNER JOIN departments ON employees.department\_id = departments.department\_id;

SELECT employee.id ,employee.name

, salary.salary

FROM employee

JOIN salary ON employee.id = salary.id;

SELECT e.id ,e.name,s.salary

FROM employee AS e

JOIN salary AS s ON e.id = s.id;

SELECT e.id ,e.name,s.salary

FROM employee e

JOIN salary s ON e.id = s.id;

SELECT st.id,st.name,st.subjectname,su.marks

FROM student st

JOIN Subject su ON st.id=su.id;

**Employee**

Name                                      Age                              Salary

1                                              20                                200

2                                              24                                300

**Department**

Name                                      Role

1                                              Data analyst

2                                              Software Engineer

**Left Join :**

**Query:**

Select \*

From Employee as e

Left Join Department as d

On e.Name=d.Name

No of Records : 2

Output:

Name             Age      Salary             Role

1                      20        200                  Data Analyst

2                      24        300                  Software Engineer

Outer Join:

**Employee**

Name                                      Age                              Salary

1                                              20                                200

2                                              24                                300

2                                              23                                220

5                                              10                                100

**Department**

Name                                      Role

1                                              Data analyst

2                                              Software Engineer

4                                              Engineer

Output:

Name             Age      Salary             Role

1                      20        200                  Data Analyst

2                      24        300                  Software Engineer

2                      23        220                  Software Engineer

4                      NULL   NULL               Engineer

5                      10        100                  Null

Nth Salary:

**Employee**

Name                                      Age                              Salary

1                                              20                                200

2                                              24                                300

2                                              23                                220

5                                              10                                100

SELECT salary, rank

FROM (

SELECT salary, DENSE\_RANK() OVER (ORDER BY salary DESC) AS rank

FROM employees

) AS ranked\_employees

WHERE rank = N;

Windows functions: rownumber, lead, lag, dense\_rank

**Sub Query:**

Name                                      Age                              Salary              dense\_rank

2                                              24                                300                       1

2                                              23                                220                       2

1                                              20                                200`                      3

5                                              10                                100                       4

**Final Output:**

Salary              dense\_rank

300                              1

SELECT m.student\_id, s.subject\_name, MAX(m.marks) AS max\_marks

FROM subjects s

Where student\_id=”sasi”

SELECT m.student\_id, s.subject\_name, MAX(m.marks) AS max\_marks

FROM subjects s

Where student\_id=”sasi”

Having max\_marks>30

**Table**

Sai       COSS    23

Sai       DB       20

Sai       DS        12

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif

1. **How can you find the total number of employees in a table named "employees"?**

Using count -SELECT COUNT(\*) AS total\_employees FROM employees;

1. **Write an SQL query to find the average salary of all employees in a table named "salaries."?**

Using AVG -SELECT AVG(salary) AS average\_salary FROM salaries;

1. **Write an SQL query to find the 5th highest-paid employee in a table named "salaries."?**

SELECT DISTINCT salary

FROM salaries

ORDER BY salary DESC

LIMIT 1 OFFSET 4;

•SELECT DISTINCT salary retrieves the distinct salary values from the "salaries" table.

•ORDER BY salary DESC sorts the salary values in descending order, which means the highest-paid employees will appear first.

•LIMIT 1 OFFSET 4 limits the result to one row, starting from the 5th row in the sorted list. The OFFSET 4 skips the first four rows (which are the four highest-paid employees), and LIMIT 1 then retrieves the 5th highest-paid employee's salary.

1. **What is the difference between the SQL WHERE and HAVING clauses, and when would you use each?**

The WHERE clause is used to filter rows before they are grouped and aggregated

SELECT column1, column2

FROM table\_name

WHERE condition;

The HAVING clause is used to filter rows after they have been grouped and aggregated using the GROUP BY clause. It typically involves aggregate functions like COUNT, SUM, AVG, etc., to filter groups based on their aggregated values

SELECT column1, COUNT(column2) as count\_column2

FROM table\_name

GROUP BY column1

HAVING COUNT(column2) > 5;

1. **What is a subquery in SQL, and provide an example when it can be used?**

In SQL, a subquery (also known as an inner query or nested query) is a query nested within another SQL query. A subquery is used to retrieve data that will be used in the main query as a condition to further filter the results. Subqueries are enclosed in parentheses and can appear in various parts of a SQL statement , such as the SELECT, FROM, WHERE, or HAVING clauses.

SELECT column1, column2

FROM table1

WHERE column3 = (SELECT column4 FROM table2 WHERE condition);

1. **Retrieve the names and ages of all customers who are older than 30 years, sorted in descending order of age?**

SELECT name, age

FROM customers

WHERE age > 30

ORDER BY age DESC;

1. **Calculate the total number of orders placed by each customer and display the results?**

SELECT customers.customer\_id, customers.customer\_name, COUNT(orders.order\_id) AS total\_orders

FROM customers

LEFT JOIN orders ON customers.customer\_id = orders.customer\_id

GROUP BY customers.customer\_id, customers.customer\_name

ORDER BY total\_orders DESC;

1. **When performing an INNER JOIN between two tables, Table A with 100 rows and Table B with 80 rows, how many rows would be in the result set?**

The number of rows in the result set for an INNER JOIN depends on the data and the specific join condition used. It could range from 0 rows (if there are no matches) to a maximum of 80 rows (the number of rows in Table B) if every row in Table A has a matching row in Table B with no duplicates.

1. **When using a FULL OUTER JOIN between Table A (with 100 rows) and Table B (with 80 rows), how many rows would be in the result set?**

Rows from Table A that have no matching rows in Table B will be included, with NULL values for the columns from Table B.

Rows from Table B that have no matching rows in Table A will also be included, with NULL values for the columns from Table A.

Rows that have matching rows in both Table A and Table B will be included in the result, with values from both tables.

1. **Write a SQL query to find the average age of employees in each department from a table named "Employees" with columns "EmployeeID," "Name," "Age," and "DepartmentID." ?**

SELECT DepartmentID, AVG(Age) AS AverageAge

FROM Employees

GROUP BY DepartmentID;

1. **Write a SQL query to retrieve the employees who have the highest salary within each department in an "Employees" table with columns "EmployeeID," "Name," "Salary," and "DepartmentID."?**

SELECT E.EmployeeID, E.Name, E.Salary, E.DepartmentID

FROM Employees E

INNER JOIN (

SELECT DepartmentID, MAX(Salary) AS MaxSalary

FROM Employees

GROUP BY DepartmentID

) MaxSalaries

ON E.DepartmentID = MaxSalaries.DepartmentID AND E.Salary = MaxSalaries.MaxSalary;

1. **Combine first name and last name into full name in employees table?**

SELECT EmployeeID, CONCAT(FirstName, ' ', LastName) AS FullName, Salary, DepartmentID

FROM Employees;

1. **What are the different types of SQL statements?**

Data Query Language (DQL) Statements:

SELECT: Retrieves data from one or more tables.

Data Definition Language (DDL) Statements:

CREATE TABLE: Creates a new table with specified columns and data types.

ALTER TABLE: Modifies an existing table, such as adding or dropping columns.

DROP TABLE: Deletes an existing table and all its data.

Data Manipulation Language (DML) Statements:

INSERT INTO: Adds new rows of data to a table.

UPDATE: Modifies existing data in a table.

DELETE FROM: Removes rows from a table.

MERGE: Performs an "upsert" operation, which combines insert and update operations based on a specified condition.

Data Control Language (DCL) Statements:

GRANT: Gives specific privileges or permissions on database objects to users or roles.

REVOKE: Removes specific privileges or permissions from users or roles.

Transaction Control Statements:

COMMIT: Saves all changes made during the current transaction.

ROLLBACK: Undoes all changes made during the current transaction.

SAVEPOINT: Sets a point within a transaction to which you can later roll back.

ROLLBACK TO: Rolls back to a specified savepoint within a transaction.

SET TRANSACTION: Sets properties for a transaction, such as isolation level.

1. **Explain the purpose of the SQL CASE statement?**

The SQL CASE statement is a powerful and flexible conditional expression used to perform conditional logic within SQL queries and update statements. It allows you to create conditional branches and specify different outcomes or values based on specified conditions. The primary purposes of the SQL CASE statement are:

SELECT

first\_name,

last\_name,

CASE

WHEN age >= 18 THEN 'Adult'

ELSE 'Minor'

END AS age\_group

FROM

people;

1. **Explain the purpose of the GROUP BY clause in SQL?**

The GROUP BY clause in SQL serves the purpose of grouping rows from a database table into summary rows based on the values in one or more columns. It is typically used in combination with aggregate functions to perform operations on groups of rows rather than on individual rows.

SELECT department, AVG(salary) AS avg\_salary

FROM employees

GROUP BY department;

1. **Explain the purpose of the HAVING clause in SQL?**

The primary purpose of the HAVING clause is to allow you to apply filtering criteria to groups of data after aggregation. Here's an explanation of its purpose:

SELECT department, AVG(salary) AS avg\_salary

FROM employees

GROUP BY department

HAVING avg\_salary > 50000;

1. **Difference between truncate, delete , drop?**

TRUNCATE is used to quickly remove all rows from a table, but it preserves the table structure and may reset identity columns.

DELETE is used to selectively remove rows from a table based on a condition, and it can be rolled back within a transaction.

DROP is used to completely remove a database object (table, index, etc.), and it is a destructive operation with no rollback capability.

1. **Explain purpose of SQL Distinct, How the same operation can be done without using Distinct Keyword?**  
   The DISTINCT keyword in SQL is used to retrieve unique values from a specific column or set of columns in a result set. Its primary purpose is to eliminate duplicate values, ensuring that each value appears only once in the result.

SELECT DISTINCT country FROM customers;

1. **Explain the difference between a unique key and a primary key?**

**GIT QA**

1. **How do you initialize a new Git repository?**

git init (clone) -> git add . ->git commit –m ”msg” -> git status -> git push

Step 1: git init used to initialize a new Git repository in the current directory

Step 2: git add if you have existing files that you want to include in the repository, you can add them using the git add command

Step 3: git commit -m "Initial commit" after adding files, you should commit them to create an initial snapshot of the project. Use the git commit command with a commit message

Step 4: git status used in Git to display the current state of your working directory and staging area (index) in relation to your Git repository. It provides information about which files have been modified, which files are staged for the next commit, and which files are untracked.

Step 5: git push used to pull request to notify others about changes you've pushed to a repository. It's a way to request that someone review and potentially merge your changes into the main or target branch.

1. **What command is used to add changes to the staging area in Git?**

To add changes to the staging area in Git, you can use the git add command. Like git add , git add . , git add \*.js, git add –all

1. **How do you commit changes in Git?**

Before committing, you need to stage the changes you want to include in the commit using the git add command.

Once your changes are staged, you can commit them using the git commit command:

git commit -m "Initial commit"

After running the git commit command, Git will create a new commit with the staged changes. You can view the commit details and the commit message by using git log: git log

**Push Your Commit (Optional)**: If you are working with a remote repository (e.g., on GitHub or GitLab), you may need to push your commits to the remote repository to make them available to others: git push origin branch\_name

1. **What is a Git branch, and why would you create one?**

A branch is a parallel line of development , Branches allow multiple developers to work on different features or issues simultaneously. Changes in one branch don't interfere with changes in other branches until you explicitly merge them. Git branches provide a flexible way to manage and organize development in a collaborative environment. They allow for concurrent work on various aspects of a project, enabling teams to develop features, fix issues, and experiment with minimal interference between different lines of development

1. **How can you merge one Git branch into another?**

Use the git merge command

Step 1 : check out the target branch - git checkout branchname

Step 2 : Merge the source branch –git merge name of the branch you want to merge into the current branch

Step 3 : Resolve Conflicts (if any)

Step 4 : Add and Commit the Merged Changes

git add .

git commit -m "Merge feature-branch into main"

Step 5 : Push the Changes

git push origin main

1. **What is merge conflict, how to resolve it?**

Merge conflicts happen when two or more branches have made conflicting changes to the same part of a file or when Git is unsure which changes should take precedence.

Step 1 : Identify the Conflict

Step 2 : Manually Resolve the Conflict

Step 3 : Add and Commit the Resolved File

Step 4 : Complete the Merge and Continue the Merge

git merge –continue

Step 5 : Test and verify

1. **How do you configure Git to use your name and email for commits?**

By using below commands

git config --global user.name "Your Name"

git config --global user.email “[youremail@example.com](mailto:youremail@example.com)”

1. **Difference between git & github?**

Git -Git is a distributed version control system. It is a tool used to track changes in source code and manage different versions of files in a project.

Git Hub-GitHub is a web-based platform that provides hosting for Git repositories. It's a service that allows you to store, share, and collaborate on Git repositories over the internet.

1. **What is Head in Git?**

Special reference It helps you identify where you are in your project's history.

Git show head

Git log

1. **What is the purpose of the .gitignore file & How to create?**

The .gitignore file is used in Git to specify which files or directories should be ignored when tracking changes in a Git repository. It's particularly useful for excluding files like build artifacts, logs, temporary files, and sensitive information that you don't want to include in your version control system.

**.gitignore**

1. **What is the difference between git pull and git fetch?**

The key difference is that git fetch only retrieves remote changes without merging them, while git pull retrieves remote changes and merges them into your current branch.

1. **How to check commit history in git?**

git log

1. **Provide the code for creating a branch and how to switch to it?**

Create a Branch: git branch new-feature

Switch to the New Branch: git checkout new-feature or git switch new-feature

Create and Switch to a New Branch in One Step:

git checkout -b new-feature or git switch -c new-feature

1. **Provide the one line code for creating and checking out in git?**

git checkout -b new-feature

1. **How do you create a new Git remote?**

git remote add <remote-name> <remote**-**url>

Ex -git remote add origin <https://github.com/username/repo.git>

1. **How can you find out which Git remote repositories are associated with your local repository?**

git remote -v

1. **How do you create a Git repository from an existing project or directory?**

Navigate to the Project Directory: cd /path/to/your/project

Initialize a Git Repository: git init

Add and Commit Your Files: git add .

Commit: git commit -m "Initial commit"

1. **What is Git's 'stash' feature, and when would you use it?**