What is API testing?

API testing is a type of software testing where application programming interfaces (APIs) are tested to determine if they meet expectations for functionality, reliability, performance, and security. In simple terms, API testing is intended to reveal bugs, inconsistencies or deviations from the expected behavior of an API.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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 the creation of the table or after creating 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.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 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;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 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;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Differences between PUT and POST in rest API?

Use PUT when we want to modify a singular resource that is already a part of resources collection. PUT replaces the resource in its entirety. Use PATCH if request updates part of the resource. Use POST when you want to add a child resource under resources

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

What are different types of parameters REST API?

REST APIs have following types of parameters:

* Header parameters. Header parameters are included in the request header. ...
* Cookie Parameters. ...
* Path/Template parameters. ...
* Query string parameters. ...
* Request body parameters. ...
* Matrix parameters. ...
* Form Parameter. ...
* Plain Parameters.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**What is XML and JSON?**

**XML** (Extensible Markup Language):

* It is especially designed to store and transport data.
* It is similar to HTML but is more flexible than HTML because it allows users to create their own custom tags.
* It is used for representing structured information such as documents, data, configuration, etc.

**JSON** (JavaScript Object Notation):

* It is a lightweight format designed to store and transport data.
* It is easier to understand and is a standard text-based format used for representing structured data based on JavaScript object syntax.
* It is faster and easier to use

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

What is Status code in Postman?

### Status Code

A status code defines the status of the request. On entering URL, a mistake can be typed in the URL, or there may be a server-side problem. Status code is used to know about what went wrong and where you made a mistake. There are different status codes, and each of the status codes has different meanings.

Let's see some standard status codes:

**200:** This code is used for a successful request.

**201:** For a successful request and data was created.

**204:** For empty response.

**400:** This is used for **Bad Request**. If you enter something wrong or you missed some required parameters, then the request would not be understood by the server, and you will get 400 status code.

**401:** This is used for **Unauthorized Access**. If the request authentication failed or the user does not have permissions for the requested operations, then you will get a 401 status code.

**403:** This is for **Forbidden** or **Access Denied**.

**404:** This will come if the **Data Not Found**.

**405:** This will come if the method not allowed or if the requested method is not supported.

**500:** This code is used for **Internal Server Error**.

**503:** And this code is used for **Service Unavailable**.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Write the SQL query to get the third maximum salary of an employee from a table named employees.**

Employee table

|  |  |
| --- | --- |
| employee\_name | salary |
| A | 24000 |
| C | 34000 |
| D | 55000 |
| E | 75000 |
| F | 21000 |
| G | 40000 |
| H | 50000 |

SELECT \* FROM(

SELECT employee\_name, salary, DENSE\_RANK()

OVER(ORDER BY salary DESC)r FROM Employee)

WHERE r=&n;

To find 3rd highest salary set n = 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### What is a UNIQUE constraint?

The UNIQUE Constraint prevents identical values in a column from appearing in two records. The UNIQUE constraint guarantees that every value in a column is unique.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

What is distinct keyword in Java?

The SELECT DISTINCT statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

SELECT DISTINCT Syntax

SELECT DISTINCT *column1*,*column2, ...*  
FROM *table\_name*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

What is the difference between GROUP BY and ORDER BY

The Group By clause is used to group data based on the same value in a specific column. The ORDER BY clause, on the other hand, sorts the result and shows it in ascending or descending order. It is mandatory to use the aggregate function to use the Group By.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Difference between Primary Key and Foreign Key

[Primary Key](https://www.geeksforgeeks.org/types-of-keys-in-relational-model-candidate-super-primary-alternate-and-foreign/)**:**

A primary key is used to ensure data in the specific column is unique. It is a column cannot have NULL values. It is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

**Example:** Refer the figure –  
STUD\_NO, as well as STUD\_PHONE both, are candidate keys for relation STUDENT but STUD\_NO can be chosen as the primary key (only one out of many candidate keys).

[Foreign Key](https://www.geeksforgeeks.org/types-of-keys-in-relational-model-candidate-super-primary-alternate-and-foreign/)**:**  
A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It is a column (or columns) that references a column (most often the primary key) of another table.

**Example:** Refer the figure –  
STUD\_NO in STUDENT\_COURSE is a foreign key to STUD\_NO in STUDENT relation.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## What is the SQL COUNT(), AVG() and SUM() Functions

The COUNT() function returns the number of rows that matches a specified criterion.

### COUNT() Syntax

SELECT COUNT(column\_name)  
FROM table\_name  
WHERE condition;

The AVG() function returns the average value of a numeric column.

### AVG() Syntax

SELECT AVG(column\_name)  
FROM table\_name  
WHERE condition;

The SUM() function returns the total sum of a numeric column.

### SUM() Syntax

SELECT SUM(column\_name)  
FROM table\_name  
WHERE condition;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

What is get POST, PUT and Delete in API?

The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server. PUT. The PUT method replaces all current representations of the target resource with the request payload. DELETE. The DELETE method deletes the specified resource.