

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

INSTRUCTIONS FOR THE INTERVIEWER:

Please have enough copies of the candidate's CV printed and shared with each Cloudgress interviewer.

At the start of the technical interview, Please ask the candidate to self-rate their skills in the following areas, then after the interview, rate their skills according to our internal assessment.

Scale of 0 – 10 0 = No Experience 10 = Expert	SQL / Databases	Python / Django	REST API's
Candidate Self-rating:			
Cloudgress rating:			

In the interview, spend about equal time on each topic to last approximately 1 hour for the total interview. However, pay a little extra attention to SQL/Databases concepts as these are fundamental to Cloudgress.

Please choose selected questions from the following list to cover all required topics including some SQL and Python coding questions also.

IMPORTANT: If the candidate is taking more than 3 to 5 minutes to write a piece of SQL/Python code, then stop that question (as it should not take too long) and move on to the next question. Within Python, the exact syntax of the code is not as important as its logic, so you can recommend that they may use Pseudo Code instead of EXACT syntax if you want.

The candidate should be provided with the following during the interview:

- Blank paper to write code on
- Pencil or Pen

This document is NOT to be shared with any candidate, or with anyone outside of Cloudgress Inc.

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

1. **What is the last thing you learned? (5 minutes) When / What / Why / How / Challenges?** A true feel good response may reveal that a previous role required plenty of technical self learning, or general self-improvement with a desire to continuously from a technical standpoint

SQL Query Focused Questions

These are basic, general SQL based questions that should be quickly answered. Ask as many as possible, cut-off and continue to the next question if too much time is spent thinking. Answers should be in the context of SQL Server.

1. What is normalization in the context of database design?
Process of organizing data, reduces redundant data; What are 3 rules; usage of foreign key constraints, or provide an example
2. Assume a schema of Emp (Id, Name, DeptId) , Dept (Id, Name). If there are 10 records in the Emp table and 5 records in the Dept table, how many rows will be displayed in the result of the following SQL query:
Select * From Emp, Dept Explain your answer.

This will return 50 records due to the cross join

3. What is the difference between TRUNCATE and DELETE in SQL? What are some pros and cons of each, or a couple scenarios where you would use one versus the other?
TRUNCATE (DDL Command) is faster and deletes all rows in a table and does not generate logs (therefore cannot rollback). DELETE (DML Command) can be used to delete specific rows. - Delete is conditional, truncate performance is faster, or row by row.
4. What is the difference between a primary key and a unique key?
Primary Key won't allow nulls, both enforce uniqueness. Both must be unique.
5. Given the following test1: Write a query that will return the rows that have "Yellow" in at least one of the columns C1, C2, or C3 without using the keyword OR.

ID	C1	C2	C3
1	Red	Yellow	Blue
2	NULL	Red	Green
3	Yellow	NULL	Violet

```
select * from TableA where C1 = 'Yellow'
union select * from TableA where C2 = 'Yellow'
union select * from TableA where C3 = 'Yellow'
```

CLOUDGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 2

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

Alternative Solution: `select * from TableA where 'yellow' in (C1, C2, C3)`

6. Consider the table Employee below

Emp_Id	Emp_name	Salary	Manager_Id
10	Anil	50000	18
11	Vikas	75000	16
12	Nisha	40000	18
13	Nidhi	60000	17
14	Priya	80000	18
15	Mohit	45000	18
16	Rajesh	90000	–
17	Raman	55000	16
18	Santosh	65000	17

```
SELECT
    a.Manager_Id, b.Emp_name AS Manager,
    AVG(a.Salary) AS Average_Salary_Under_Manager
FROM
    Employee a, Employee b
WHERE
    a.Manager_Id = b.Emp_Id
GROUP BY a. Manager_Id, b.Emp_Name
```

Manager Id	Manager	Average Salary Under Manager
16	Rajesh	65000
17	Raman	62500
18	Santosh	53750

7. What are database indexes? What are the pros and cons?

A copy of a selected columns, designed to search efficient when used properly. Pros: data retrieval performance is improved, Cons: Storage space, overhead when inserting/updating.

8. What is the difference between UNION and UNION ALL in SQL?

UNION: eliminates duplicates UNION ALL: retains duplicates Both are used to combine the results of different SELECT statements.

CLOUDGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 3

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

9. How would you find out the total number of rows in a table?

Use SELECT COUNT(*) ... in query

10. How do you eliminate duplicate values in SELECT ?

Use SELECT DISTINCT ... in SQL query

11. In the WHERE clause what is difference between the key words BETWEEN and IN?

BETWEEN supplies a range of values while IN supplies a list of values.

12. What are a few of the different joins? Explain a couple of them?

INNER JOIN

OUTER JOIN

LEFT OUTER JOIN

RIGHT OUTER JOIN

FULL OUTER JOIN

13. What are Temp DB Files in SQL Server? What are they typically used for? Not looking for an exact definition, but some discussion on what it may be used for or how they are used

14. What is an Identity Column in SQL Server

Numeric incremental value typical in a primary key field that is used in a CREATE statement, that is automatically populated

15. Challenge the interviewee what a general approach they would take to solving a performance issue in a long running process, such as one in a nightly batch cycle. (3-5 minutes)

Look for the type of questions they would ask and potential solutions.

- An example task may be a File Export Task that involves a long, complicated query returning highvolume records, plus a loop to iterate through each row and ETL within the inner loop that involves multiple database hits. The inner loop may involve determining if an update or an insert is required for each record.
- Look for suggestions on how to both identify the bottleneck and solve for the issue. Responses may improve query optimization, separate into smaller batches, multi threading, data cache within the loop to save database calls, reducing number of SQL calls within the loop.
- Not trying to find a specific solution, more so trying to determine if there has been any experiences in the past.

16. What is Index Rebuilding? When and why would you perform one? What is the difference between rebuilding and reorganizing? **Deletes and rebuilds index entirely for performance. Industry practice is to rebuild when fragmentation reaches 30%. Reorganizing is lighter, involves moving pages around and can be performed online**

CLOUDGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 4

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

Python

Easy Questions

1. What is the difference between .py and .pyc files?

.py files are Python source files. .pyc files are the compiled bytecode files that is generated by the Python compiler.

2. Explain inheritance in Python.

Inheritance is a feature by which one class can gain all the attributes and methods of another class. This facilitates code reusability and a program one written can be reused by inheriting the parent class attributes without having to rewrite them every time. The class from which we are inheriting is called super-class and the class that is inherited is called a sub-class / derived class.

- Single Inheritance – where a derived class acquires the members of a single superclass.
- Multi-level inheritance – a derived class d1 is inherited from base class base1, and d2 is inherited from base2.
- Hierarchical inheritance – from one base class you can inherit any number of child classes
- Multiple inheritances – a derived class is inherited from more than one base class.

3. What is the difference between lists and tuples?

Lists are mutable, i.e., they can be edited	Tuples are immutable (they are lists that cannot be edited)
Lists are usually slower than tuples	Tuples are faster than lists
Lists consume a lot of memory	Tuples consume less memory when compared to lists

Lists are less reliable in terms of errors as unexpected changes are more likely to occur	Tuples are more reliable as it is hard for any unexpected change to occur
Lists consist of many built-in functions.	Tuples do not consist of any built-in functions.
Syntax: list_1 = [10, 'Intellipaat', 20]	Syntax: tup_1 = (10, 'Intellipaat', 20)

Difficult Questions

1. What is pickling and unpickling in Python? Pros and Cons?

Pickling is just the Python way of saying *serializing*. Pickling lets you serialize an object into a string (or anything else you choose) in order to be persisted on storage or sent over network. Unpickling is the process of restoring the original object from a pickled string. Used in file/database, maintain program state across sessions, or transport data over the network.

2. What is GIL and what are some of the ways to get around it?

CLoudGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 5

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

GIL stands for the Global Interpreter Lock and it is a mechanism Python is using for concurrency. It is built in deep into Python system and it is not possible to get rid of it. The major downside of GIL is that it makes threading not truly concurrent. It locks the interpreter, and even though it looks like you are working with threads, they are not executed at the same time, resulting in performance losses. Here are some ways of getting around it:

- multiprocessing module. It lets you spawn new Python processes and manage them the same way you would manage threads
- asyncio module. It effectively enables asynchronous programming and adds the async/await syntax. While it does not solve the GIL problem, it will make the code way more readable and clearer.
- Stackless Python. This is a fork of Python without GIL.

3. Write a python program to check Prime Number? (5 mins)

```
if num > 1:
for i in range(2, num):
if (num % i) == 0:
print(num, "is not a prime number")
break
else:
print(num, "is a prime number")
else:
```

```
print(num, "is not a prime number")
```

4. Write a python program to check To Check If A Number Is A Perfect Square (5-7 mins)
Without the square root function

```
def isPerfectSquare(n) :  
    i = 1  
    while(i * i<= n):  
  
        # If (i * i = n)  
        if ((n % i == 0) and (n / i == i)):  
            return True  
        i = i + 1  
    return False  
  
# Driver code  
if __name__ == "__main__" :  
  
    n = 36  
    if (isPerfectSquare(n)):  
        print("Yes, it is a perfect square.")  
    else :  
        print("No, it is not a perfect square.")  
  
=====
```

import math

CLOUDGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 6

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

```
# Taking the input from user  
number = int(input("Enter the Number"))  
  
root = math.sqrt(number)  
if int(root + 0.5) ** 2 == number:  
    print(number, "is a perfect square")  
else:  
    print(number, "is not a perfect square")
```

5. Write the code to print out the nth Fibonacci number python (7-10 mins)

#recursive approach

```
def Fibonacci(n):  
    # Check if input is 0 then it will  
    # print incorrect input  
    if n < 0:  
        print("Incorrect input")  
    # Check if n is 0  
    # then it will return 0  
    elif n == 0:  
        return 0  
    # Check if n is 1,2
```

```
# it will return 1
elif n == 1 or n == 2:
    return 1
else:
    return Fibonacci(n-1) + Fibonacci(n-2)
# Driver Program
print(Fibonacci(9))
```

CLOUDGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 7

Cloudgress Inc. Technical Interview Questions



CLOUDGRESS

RESTAPI Advanced questions

1. What is the concept of statelessness in REST?

The REST architecture is called stateless as it does not store any state related to the client session on the server. Essentially session states are entirely kept on the client side. This is effectively called Statelessness.

2. What are payloads in RESTful web services?

Payload refers to the request data present in the body part of every HTTP request message. However, the payload is not the same as request parameters and can only be passed through the POST method.

3. How RESTAPI authentication works?

The RESTAPI authentication is based on the transfer of credentials from the client and the server. The basic authentication is based on the HTTP header. The HTTP header is used for providing the username and password that is encoded in base64.

4. How can API call be made faster in Python (performance issues)

For each symbol. If we have five symbols, we are “waiting” five times. That’s five times we are just doing nothing! If we have 100 API calls, we are doing nothing 100 times! Instead of doing this, it would be better to kick off one API call, and instead of waiting, kick off the other API calls and then handle the responses later.

So we now know that when we run code asynchronously, we don’t always have to wait right away for it to finish. We can use **asyncio** and **aiohttp** to do exactly this.

5. Error Handling in REST API? Network related issues?

Keep trying the last request until it succeeds?

Ignore the failure?

Somehow roll back all the previous, successful requests?

HTTP Code

When a client makes a request to an HTTP server — and the server successfully receives the request — **the server must notify the client if the request was successfully handled or not.**

HTTP accomplishes this with five categories of status codes:

- 100-level (Informational) – server acknowledges a request
- 200-level (Success) – server completed the request as expected
- 300-level (Redirection) – client needs to perform further actions to complete the request
- 400-level (Client error) – client sent an invalid request
- 500-level (Server error) – server failed to fulfill a valid request due to an error with server

CLOUDGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 8

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

Based on the response code, a client can surmise the result of a particular request.

Basic Responses

The simplest way we handle errors is to **respond with an appropriate status code**. Here are some common response codes:

400 Bad Request – client sent an invalid request, such as lacking required request body or parameter

- 401 Unauthorized – client failed to authenticate with the server
- 403 Forbidden – client authenticated but does not have permission to access the requested resource
- 404 Not Found – the requested resource does not exist
- 412 Precondition Failed – one or more conditions in the request header fields evaluated to false
- 500 Internal Server Error – a generic error occurred on the server
- 503 Service Unavailable – the requested service is not available

While basic, these codes allow a client to understand the broad nature of the error that occurred. We know that if we receive a 403 error, for example, we lack permissions to access

the resource we requested. In many cases, though, we need to provide supplemental details in our responses.

500 errors signal that some issues or exceptions occurred on the server while handling a request. Generally, this internal error is not our client's business.

6. Can you tell the disadvantages of RESTful web services?

The disadvantages are:

- As the services follow the idea of statelessness, it is not possible to maintain sessions. (Session simulation responsibility lies on the client-side to pass the session id)
- REST does not impose security restrictions inherently. It inherits the security measures of the protocols implementing it. Hence, care must be chosen to implement security measures like integrating SSL/TLS based authentications, etc.

CLOUDGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 9

Cloudgress Inc. Technical Interview Questions



CLOUDGRESS

7. What are the best practices to develop RESTful web services?

RESTful web services use REST API as means of implementation using the HTTP protocol. REST API is nothing but an application programming interface that follows REST architectural constraints such as statelessness, cacheability, maintainability, and scalability. It has become very popular among the developer community due to its simplicity. Hence, it is very important to develop safe and secure REST APIs that follow good conventions. Below are some best practices for developing REST APIs:

- Since REST supports multiple data formats, it is however good practice to develop REST APIs that accept and responds with JSON data format whenever possible. This is because a majority of the client and server technologies have inbuilt support to read and parse JSON objects with ease, thereby making JSON the standard object notation.
 - To ensure that the application responds using JSON data format, the response header should have Content-Type set to as application/JSON, this is because certain HTTP clients look at the value of this response header to parse the objects

appropriately.

- To ensure that the request sends the data in JSON format, again the Content-Type must be set to application/JSON on the request header.
- While naming the resource endpoints, ensure to use plural nouns and not verbs. The API endpoints should be clear, brief, easy to understand, and informative. Using verbs in the resource name doesn't contribute much information because an HTTP request already has what the request is doing in its HTTP method/verb. An appropriate HTTP verb should be used to represent the task of the API endpoint.

Below are the most commonly used HTTP methods to define the verb:

- GET - indicates get/retrieve the resource data
- POST - indicates create new resource data
- PUT - indicates update the existing resource data
- DELETE - indicates remove the resource data
- To represent the hierarchy of resources, use the nesting in the naming convention of the endpoints. In case, you want to retrieve data of one object residing in another object, the endpoint should reflect this to communicate what is happening. For example, to get the address of an author, we can use the GET method for the URI /authors/:id/address'
- Please ensure there are no more than 2 or 3 levels of nesting as the name of the URI can become too long and unwieldy.
- Error Handling should be done gracefully by returning appropriate error codes the application has encountered. REST has defined standard HTTP Status codes that can be sent along with the response based on the scenario.

CLOUDGRESS INTERNAL DOCUMENT – STRICTLY CONFIDENTIAL 10

Cloudgress Inc. Technical Interview Questions CLOUDGRESS

- Error codes should also be accompanied by appropriate error messages that can help the developers to take corrective actions. However, the message should not be too elaborate as well which can help the hacker to hack your application.
- Common status codes are:
 - 400 - Bad Request – client-side error - failed input validation.
 - 401 - Unauthorized – The user is not authenticated and hence does not have authority to access the resource.
 - 403 - Forbidden – User is authenticated but is not authorized to access the resource.
 - 404 - Not Found – The resource is not found.
 - 500 - Internal server error – This is a very generic server-side error that is thrown when the server goes down. This shouldn't be returned by the programmer explicitly.
 - 502 - Bad Gateway – Server did not receive a valid response from the upstream server.
 - 503 - Service Unavailable – Some unexpected things happened on the server

such as system failure, overload, etc.

- While retrieving huge resource data, it is advisable to include filtering and pagination of the resources. This is because returning huge data all at once can slow down the system and reduce the application performance. Hence, filter some items reduces the data to some extent. Pagination of data is done to ensure only some results are sent at a time. Doing this can increase the server performance and reduce the burden of the server resources.
- Good security practices are a must while developing REST APIs. The client-server communication must be private due to the nature of data sensitivity. Hence, incorporating SSL/TLS becomes the most important step while developing APIs as they facilitate establishing secure communication. SSL certificates are easier to get and load on the server.
 - Apart from the secure channels, we need to ensure that not everyone should be able to access the resource. For example, normal users should not access the data of admins or another user. Hence, role-based access controls should be in place to make sure only the right set of users can access the right set of data.
- Since REST supports the feature of caching, we can use this feature to cache the data in order to improve the application performance. Caching is done to avoid querying the database for a request repeated times. Caching makes data retrieval fast. However, care must be taken to ensure that the cache has updated data and not outdated ones. Frequent cache update measures need to be incorporated. There are many cache providers like Redis that can assist in caching.
- API Versioning: Versioning needs to be done in case we are planning to make any changes with the existing endpoints. We do not want to break communication between our application and the apps that consume our application while we are working on the API release. The transition has to be seamless. Semantic versioning can be followed. For example, 3.0.1 represents 3rd major version with the first patch. Usually, in the API endpoints, we define /v1,/v2, etc at the beginning of the API path.