Data Engineer Interview Questions: Part I

Big Data Interview Questions Spark, SQL, Python, Data Modeling, and Data Warehouse, Data Structure & Algorithm

Aman Ranjan verma 3 days ago · 5 min read ★

I am a data engineering with 2.4 Years of experience. During the course of the last 4 months, I have attended 75 interview sessions for the role of data engineering with 26 different companies.

By Aman Ranjan Verma

Amazon, **ANZ**, Apisero, Aviyel, Amagi, Busigence, **BCG**, BitClass, couture.ai, **Fractal**, **Flipkart**, **Indeed**, Healthplix, Lead School, Lumiq, Moveworks, **Nagarro**, Novo Nordisk, **PayPal**, Pharmeasy, Recko, Tredence**,** **Uber**, Vahan, Vimana, Xpressbees.

In most of the interviews, the questions were based on my past experiences and the skill set that I hold. I have attached my resume for you to get an understanding of my background.

It will be a series of blogs on data engineering interview questions that I have been asked in different companies. In each blog, I will put two questions on each topics Python, Spark, and SQL.

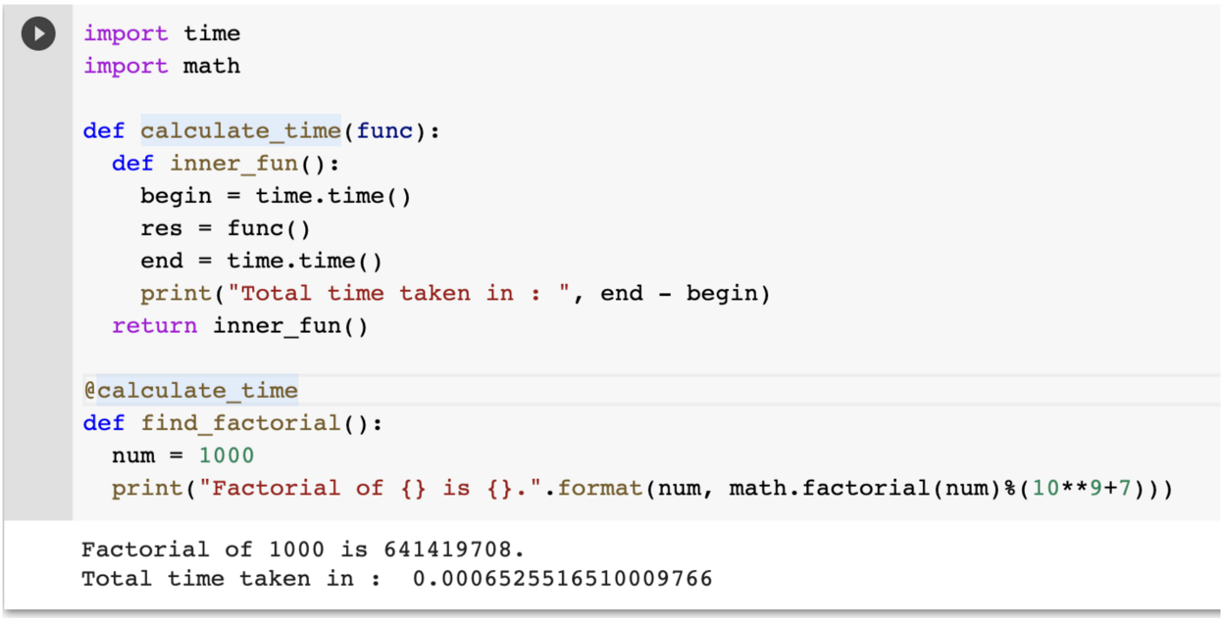
. . .

Python

Q. What is Decorator in Python?

Decorators allow us to extend the behavior of a function by wrapping it into another function, without permanently modifying it.

Example:



```
import time
import math

def calculate_time(func):
  def inner_fun():
    begin = time.time()
    res = func()
    end = time.time()
    print("Total time taken in : ", end - begin)
  return inner_fun()

@calculate_time
def find_factorial():
  num = 1000
  print("Factorial of {} is {}.".format(num, math.factorial(num)%(10**9+7)))
```

```
Factorial of 1000 is 641419708.
Total time taken in :  0.0006525516510009766
```

By Aman Ranjan Verma

Here in this example, there is a factorial function whose functionality is extended by wrapping it into the calculate_time function. The other thing to note here are:

- calculate_time function is accepting and returning another function as a parameter
- calculate_time has an inner function which is calling the wrapped function.

In most of the decorated examples, you will find a similar structure where you will have:

- A function whose functionality is to be extended(find_factorial).
- Another function that is responsible to extend the functionality and returns the inner function. It is called the

wrapper function(calculate_time).

- The inner function, that contains the logic to extend the functionality and which calls the wrapped function(inner_func).

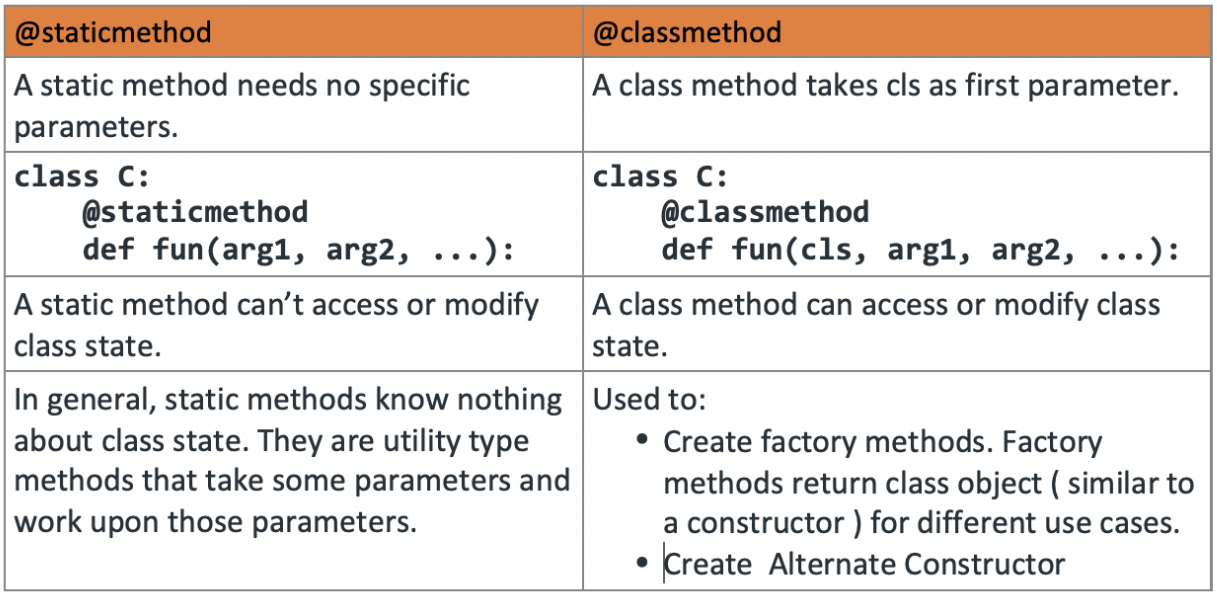
For detailed information:

Decorators in Python - GeeksforGeeks

Decorators are a very powerful and useful tool in Python since it allows programmers to modify the...

www.geeksforgeeks.org

Q. What is the difference between @staticmethod and the @classmethod?



@staticmethod	@classmethod
A static method needs no specific parameters.	A class method takes cls as first parameter.

`class C:`
`    @staticmethod`
`    def fun(arg1, arg2, ...):`
	`class C:`
	`    @classmethod`
	`    def fun(cls, arg1, arg2, ...):`

A static method can't access or modify class state.	A class method can access or modify class state.
In general, static methods know nothing about class state. They are utility type methods that take some parameters and work upon those parameters.	Used to: • Create factory methods. Factory methods return class object (similar to a constructor) for different use cases. • Create Alternate Constructor

By Aman Ranjan Verma

Python’s Instance, Class, and Static Methods

This article uncovers the class methods, static methods, and regular instance methods.

medium.com

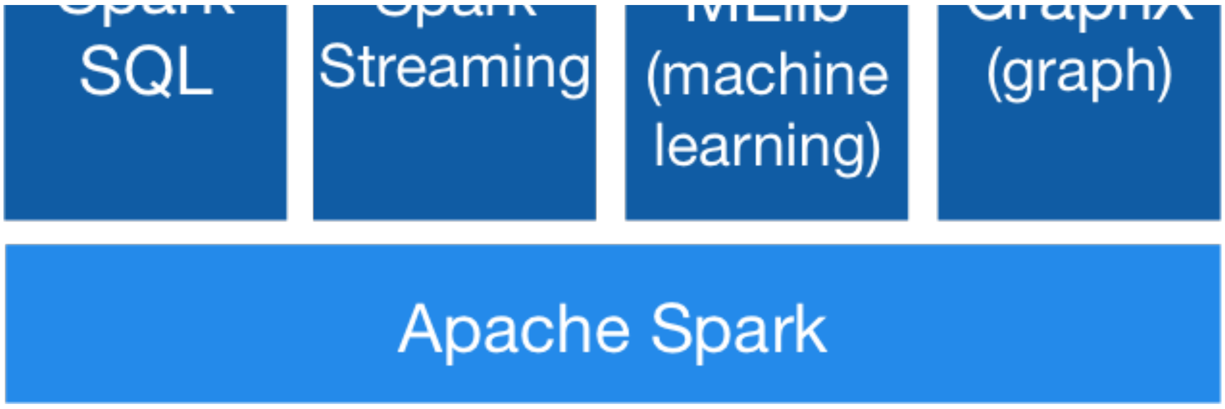
Spark

Q. What are the components of Spark?

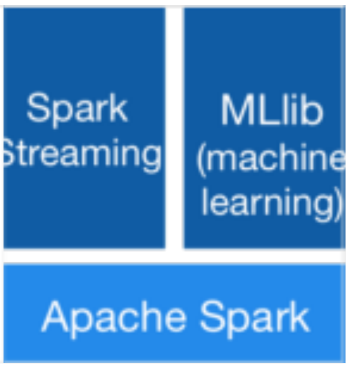
Spark has several components for different types of processing. All components are built on top of Spark Core(RDD layer abstraction).

- Spark Streaming: For processing streaming data in real-time
- GraphX: Performs processing on graphs. Solves problems using graph theory.
- SparkSQL: Provided data frame and dataset API to process data. One can also run SQL queries on top of the relational representation of the data.
- MLlib: Provides machine learning library, with different algorithms for several activities like collaborative filtering, classification, clustering, and regression.

Spark
Spark
MLlib
GraphX



Reference: https://spark.apache.org/



Start Your Journey with Apache Spark — Part 1

Understanding Apache Spark and RDD (Resilient Distributed Datasets)

medium.com

Q. How do you optimize spark jobs for optimum performance?

Spark programs can be bottlenecked by any of these resources in the cluster:

- CPU
- Network bandwidth
- Memory

Optimization technique:

- Do not use `collect()` on a dataset that is too large to fit into the driver memory. Instead, use `take()` to get only a certain number of elements.

- Do not use `count()` when you do not need to return the exact number of rows. Instead, you can check if it is empty with a simple `if(take(1).length == 0)`
- Use `coalesce` function instead of `repartition` if you decrease the number of partitions of the RDD
- Use broadcast variable Joining a large(fact table) and a small dataset (dim table)

Aman Ranjan verma

Engineer who loves forest, mountains, and general science.

https://www.linkedin.com/in/ar-verma/

Follow

How to set up spark configuration for optimum performance?

In a cluster with 10 nodes with each node(16 cores and 64GB RAM)

- Assign 5 core per executors , — executor-cores = 5 (for good HDFS throughput)
- Leave 1 core per node for Hadoop/Yarn daemons, Num cores available per node = 16–1 = 15 So, Total available of cores in cluster = 15 x 10 = 150
- Number of available executors = (total cores/num-cores-per-executor) = 150/5 = 30
- Leaving 1 executor for Application Manager, — num-executors = 29

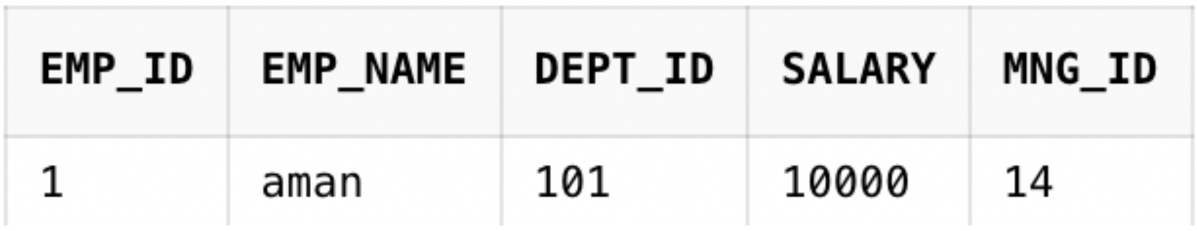
- Number of executors per node = 30/10 = 3
- Memory per executor = 64GB/3 = 21GB
- Counting off heap overhead = 7% of 21GB = 3GB. So, actual — executor-memory = 21–3 = 18GB

SQL

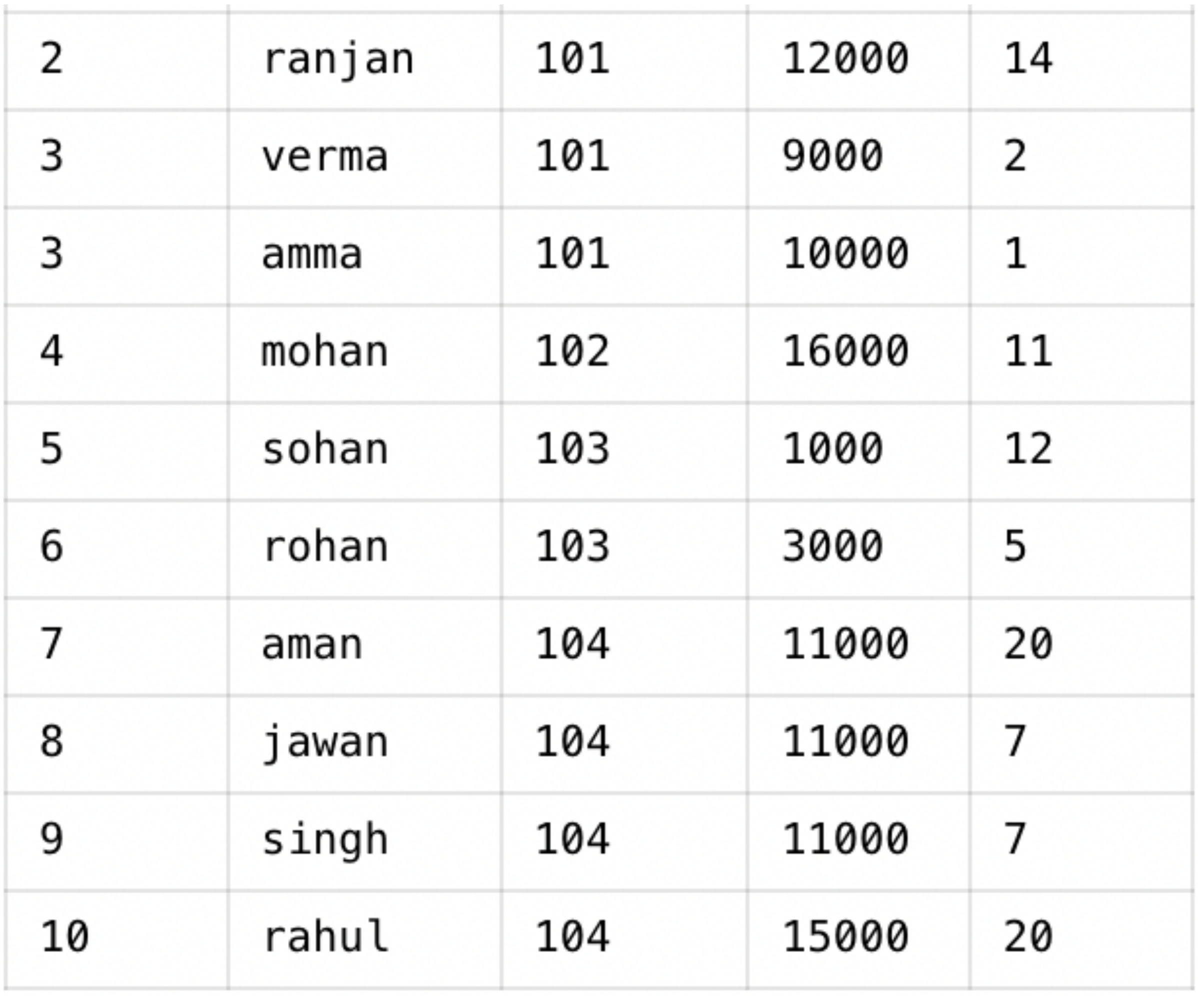
Q. What are different keys in a table?

Super Key: Set of columns that help in identifying a unique record in a table.
Candidate key: All those minimal sets of columns which are a subset of super keys that help in identifying a unique record in a table.
Primary Key: A candidate key that is chosen to act as PK
Example: student_id and phone number both are candidate keys but it makes sense to keep student_id as PK because it can be used as FK for in another table.
Foreign Key: Foreign keys are the column of the table which is used to point to the primary key of another table.

Q. SELECT 1st 2 employees from each department who joined first



EMP_ID	EMP_NAME	DEPT_ID	SALARY	MNG_ID
1	aman	101	10000	14



2	ranjan	101	12000	14
3	verma	101	9000	2
3	amma	101	10000	1
4	mohan	102	16000	11
5	sohan	103	1000	12
6	rohan	103	3000	5
7	aman	104	11000	20
8	jawan	104	11000	7
9	singh	104	11000	7
10	rahul	104	15000	20

Given Table, By Aman Ranjan Verma

```sql
CREATE TABLE EMPLOYEE (
    emp_id INT,
    emp_name VARCHAR(15),
    dept_id INT,
    salary INT,
    mng_id INT
);

INSERT INTO EMPLOYEE VALUES(1, 'aman', 101, 10000, 14);
INSERT INTO EMPLOYEE VALUES(2, 'ranjan', 101, 12000, 14);
INSERT INTO EMPLOYEE VALUES(3, 'verma', 101, 9000, 2);
INSERT INTO EMPLOYEE VALUES(3, 'amma', 101, 10000, 1);
INSERT INTO EMPLOYEE VALUES(4, 'mohan', 102, 16000, 11);
INSERT INTO EMPLOYEE VALUES(5, 'sohan', 103, 1000, 12);
INSERT INTO EMPLOYEE VALUES(6, 'rohan', 103, 3000, 5);
INSERT INTO EMPLOYEE VALUES(7, 'aman', 104, 11000, 20);
INSERT INTO EMPLOYEE VALUES(8, 'jawan', 104, 11000, 7);
INSERT INTO EMPLOYEE VALUES(8, 'jawan', 104, 11000, 7);
INSERT INTO EMPLOYEE VALUES(9, 'singh', 104, 11000, 7);
INSERT INTO EMPLOYEE VALUES(10, 'rahul', 104, 15000, 20);


SELECT * FROM EMPLOYEE;

-- SELECT 1st 2 employees from each department who joined first
```

Solution, By Aman Ranjan Verma

. . .

I hope that you found this article useful. The next blog of the series is live.

Data Engineer Interview Questions: Part II

Big Data Interview Questions Spark, SQL, Python, Data Modeling, and Data Warehouse, Data Structu...

medium.com

If you want company-specific interview questions kindly let me know in the ✍️comment section.

All the best for your next interview! 😅

Data Engineering Interview Spark Sql Python