1. What are cache and persist in Spark, and how do they work internally?
2. What are the different file formats in Spark? Compare CSV, Parquet, Avro, ORC, and JSON. What are different compression formats used in spark.
3. How does Parquet work internally?
4. Why does Spark read Parquet files faster due to its columnar format?
5. Can a Parquet file contain a header?
6. What are actions and transformations in Spark?
7. What are stages and tasks in Spark?
8. Explain how Spark works internally from the spark-submit command till the end.
9. How are DAGs created in Spark?
10. Given a table that includes user\_id, session\_id, start\_time, and end\_time, write a query to find each user's average and maximum time spent.
11. Given the table Employees (columns: EmployeeID, Name, Department, Salary), write a query to rank employees within each department by their salary in descending order.
12. Given the table StockPrices (columns: StockID, PriceDate, Price), calculate the 3-day moving average of stock prices.
13. **Problem:** N children are standing in a line. Each child is assigned a rating value. You are giving candies to these children subject to the following requirements:

* Each child must have at least one candy.
* Children with a higher rating get more candies than their neighbors.
* What is the minimum number of candies you must give?

1. You have two tables:
   * employees(emp\_id, emp\_name, department\_id, salary, hire\_date)
   * departments(department\_id, department\_name)

Write a query to calculate the difference between each employee’s salary and the highest salary within their department. Then, label employees who have a salary within 1000 of the department's highest salary as 'Near Top Earner'.

1. Given the userProfile table with fields:
   * userId
   * ctc
   * total\_exper

And the communicationSend table with fields:

* + userId
  + date
  + communicationType

Write a SQL query to find the distribution of users who have an entry in the communicationSend table with "notification" communication type in the last 30 days. Group users into buckets:

* + **Send count bucket:** 0-5, 5-10, 10-15, >15
  + **CTC bucket:** 0-5, 5-10, 10-15, >15
  + **Experience bucket:** 0-5, 5-10, 10-15, >15

1. Given a text file or Parquet file, write a Spark program using RDDs/DataFrames to count the frequency of each word in the text file. Exclude stop words like "the", "is", and "and" from the count.
2. Suppose we have a driver with 5GB memory, 5 executors with 5GB each, and each executor has 5 cores. Explain how this Spark program works, how many executors should be used, and how partitions are created.
3. Given the table adrequest with columns user\_id, pageurl, and timestamp, find the winner page (the page on which the maximum time was spent) for each user. Return the winner pageurl.
4. You have two tables with one column each:
   * table1(id)
   * table2(id)
   * Write queries to count the number of rows after performing inner, left, outer, and right joins.
5. For the following code, highlight which part will be executed on the master and which on the worker nodes. Identify the costly operations.

val formatter: DateTimeFormatter = DateTimeFormatter.ofPattern("yyyy/MM")

def getEventCountOnWeekdaysPerMonth(data: RDD[(LocalDateTime, Long)]): Array[(String, Long)] = {

val result = data

.filter(e => e.\_1.getDayOfWeek.getValue < DayOfWeek.SATURDAY.getValue)

.map(mapDateTime2Date)

.reduceByKey(\_ + \_)

.collect()

result

.map(e => (e.\_1.format(formatter), e.\_2))

}

private def mapDateTime2Date(v: (LocalDateTime, Long)): (LocalDate, Long) = {

(v.\_1.toLocalDate.withDayOfMonth(1), v.\_2)

}

1. For the following Spark program:

case class User(userId: Long, userName: String)

case class UserActivity(userId: Long, activityTypeId: Int, timestampEpochSec: Long)

val LoginActivityTypeId = 0

val LogoutActivityTypeId = 1

private def readUserData(sparkSession: SparkSession): RDD[User] = {

sparkSession.sparkContext.parallelize(

Array(

User(1, "Doe, John"),

User(2, "Doe, Jane"),

User(3, "X, Mr."))

)

}

private def readUserActivityData(sparkSession: SparkSession): RDD[UserActivity] = {

sparkSession.sparkContext.parallelize(

Array(

UserActivity(1, LoginActivityTypeId, 1514764800L),

UserActivity(2, LoginActivityTypeId, 1514808000L),

UserActivity(1, LogoutActivityTypeId, 1514829600L),

UserActivity(1, LoginActivityTypeId, 1514894400L))

)

}

def calculate(sparkSession: SparkSession): Unit = {

val userRdd: RDD[(Long, User)] = readUserData(sparkSession).map(e => (e.userId, e))

val userActivityRdd: RDD[(Long, UserActivity)] = readUserActivityData(sparkSession).map(e => (e.userId, e))

val result = userRdd

.leftOuterJoin(userActivityRdd)

.filter(e => e.\_2.\_2.isDefined && e.\_2.\_2.get.activityTypeId == LoginActivityTypeId)

.map(e => (e.\_2.\_1.userName, e.\_2.\_2.get.timestampEpochSec))

.reduceByKey((a, b) => if (a < b) a else b)

result.foreach(e => println(s"${e.\_1}: ${e.\_2}"))

}

What will be the output of this code?

1. What is the difference between a data warehouse and a data lake?
2. What are star and snowflake schemas? Why are they used?
3. How can Airflow be scaled to run 1000 jobs in parallel?
4. Why is Elasticsearch fast? How does its indexing work?
5. How spark.sql.adaptive.enabled works internally?
6. What are stored procedures in snowflake?
7. What are SCD1 and SCD2.
8. Diff between normalised and denormalised data in star and snowflake schema?
9. Diff between rank(), dense\_rank(), row\_number()
10. Diff between drop, delete and truncate
11. Given an array arr[] of n elements that contains elements from 0 to n-1, with any of these numbers appearing any number of times. The task is to find the repeating numbers.
12. What are constructors, decorators in python? implement them?
13. What are access modifiers?
14. Remove duplicates from an array using python.
15. What is deterministic matching? Explain me with an example.
16. Write pyspark code for word count from a file using df and rdd? How does reducemap work in rdd?
17. Difference between rdd, df and dataset in pyspark?
18. What are window functions? explain lead and lag.
19. Use some window function and write sql query for it?
20. I have a table sales\_id, product\_name, sales\_date. We have to give only those rows with latest data based on sales\_date for each product name .
21. # Write a function that, given a string, returns its longest palindromic substring.

#   A palindrome is defined as a string that's written the same forward and backward. Note that single-character strings

#   are palindromes.

# You can assume that there will only be one longest palindromic substring.

# Sample Input

# string = "abaxyzzyxf"

# Sample Output

# "xyzzyx"

1. How to pass output from one operator to another operator as input in Airflow?
2. What is xcom in airflow?
3. Have you used a custom operator in Airflow?
4. Sort the array which contains only 0 and 1
5. Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.An input string is valid if:

Open brackets must be closed by the same type of brackets.

Open brackets must be closed in the correct order.

Every close bracket has a corresponding open bracket of the same type.

1. What is diff between partition and bucketing in spark
2. What is diff between snappy and gzip and when to use them
3. how to resolve data skewness in spark
4. what are the diff data optimization in spark
5. Define arctitecture of airflow and its components.
6. difference between run length encoding and dictionary encoding, delta encoding, bit-packing
7. decorater example

def my\_decoratror(func)

def wrapper():

print('before function call')

func()

print('after function call')

@my\_decorator

def greet():

print('hello world')

greet()

1. Constructor example

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def display(self):

print(f"Name: {self.name}, Age: {self.age}")

p1 = Person('abc',36)

p1.display()

1. Sorting Algo (merge sort and quick sort)
2. How SQL query works and how we can optimize SQL
3. Where we have to use avro and where we have to use parquet
4. what is SCDs and scd0, scd1 and scd2 examples and how we maintain them with example
5. Where we have to use partitioning and where we have to use bucketing
6. what are the diff schema types you have heard example star or snowflake schema
7. what are narrow and wide transformations and its example
8. how spark works define its architecture
9. A company maintains information about its order on orders table. Write a query to print details of the earliest five orders(sorted by order\_date ascending) that have not been delivered(i.e status is not delivered). If there are more than 5 orders to chose from select the ones with the lowest order id. Sort the output in the increasing order of the order\_id. the output should contains id, order\_date, status, customer\_id.
10. Given a table users in which we have columns as user\_id, deptartment, salary and we have to find the diff between 2nd and 3rd highest salary of each dept.
11. You are given a table user\_sales with the following columns:

user\_id (unique identifier for each user),

quarter (values: 'Q1', 'Q2', 'Q3', 'Q4'),

sales (sales amount for that user in the respective quarter).

Write a SQL query to pivot this data such that each row represents a user and their sales in each quarter, with the output columns:

user\_id, Q1\_sales, Q2\_sales, Q3\_sales, Q4\_sales.

1. You are given three tables:

users(user\_id, user\_name, manager\_id) — contains information about employees and their respective managers.

managers(manager\_id, manager\_name) — contains information about managers.

orders(order\_id, user\_id, order\_date) — contains information about orders placed by users.

Task:

Write a SQL query to find the names of employees (user\_name) and their corresponding manager\_name for all employees who have not placed any orders.

1. You are given a DataFrame with the following columns:

user\_id (integer): Unique identifier for each student

name (string): Name of the student

marks (array of integers): A list containing the marks obtained by the student in 5 different subjects

You are required to calculate the average of the top 3 marks for each student.

student\_date = [(1, 'Janit', [80,70,81,54,90]),

(2, 'Ankit', [81,72,81,54,91]),

(3, 'Syed', [82,74,81,56,92]),

(4, 'Lokesh', [83,76,81,58,93]),

(5, 'Nishant', [84,70,81,60,94])]

1. "Given a table named users with the following columns: user\_id, product\_id, and sales, write an SQL query to fetch the top two sales (highest sales amounts) for each user."
2. "You are given a 1 TB file that needs to be sorted based on one or more columns. However, the machine available has only 1 GB of RAM. How would you efficiently perform this sort operation given the memory constraint?"
3. You are working as a receptionist and you receive a list of all the meetings scheduled for the next day. Each meeting has a start and end time, and there is only one meeting room available. Your task is to schedule the maximum number of non-overlapping meetings in that room. How would you approach this problem?"
4. "In large-scale data processing (e.g., using PySpark or SQL), GROUP BY operations can be computationally expensive due to shuffling and data movement. If you want to avoid using GROUP BY, what alternative approaches can you use to achieve similar aggregation or grouping results?"
5. what are sensors in airflow
6. what are difference between map, flatmap, reducebykey?
7. What will happen in spark if one of the executors got killed due to OOM or because of any other error? How it affects the running job?
8. What are shell scripts?
9. Minimum Platforms Required for Given Arrival and Departure Times DSA question
10. Flatten an array (we can use generator to escape from memory overflow (using recursion))
11. What is micro partitioning in snowflake?
12. What is Fail-safe in Snowflake?
13. What is diff between zero copy clone and full copy in snowflake?
14. What is Snowpipe and how we can use it for realtime data ingestion?
15. What is diff between pandas and spark dataframe and which one is immutable and why it is immutable?
16. You are given two tables:

employee(emp\_id, emp\_name, emp\_salary, dep\_name)

department(dept\_id, dept\_name)

Find the emp\_name and the corresponding dept\_id of employees whose emp\_salary is greater than the average salary of their department.

1. How lambda, glue, EMR works with example for data engineering?
2. What is delta lake and Apache iceberg and Apache Hudi?
3. What is diff between data lake, data mart, Lakehouse?
4. You are given a table employees with column emp\_id, salary, dept\_id and we have to give those rows with 4th highest salary in each dept. Write code in SQL and pyspark.
5. What is diff between partitioning and bucketing.
6. What are the different optimisation techniques used in spark.
7. Parquet files are immutable. If a new file lands in S3 and some data needs to be updated or merged with existing Parquet files, how would you handle this upsert operation?
8. When a new Parquet file lands in S3, how can you detect that the data has changed and trigger an upsert job?

**Part 1: SQL**

**Q1.** Write a query to fetch each customer’s name, total number of claims, and total approved claim amount.

**Input Tables**:

Customer Table:

|  |  |  |  |
| --- | --- | --- | --- |
| **customer\_id** | **name** | **age** | **gender** |
| 101 | Rahul Sharma | 35 | M |
| 102 | Priya Verma | 29 | F |
| 103 | Amit Patel | 42 | M |
| 104 | Neha Singh | 31 | F |
| 105 | Sunil Mehra | 48 | M |
| 106 | Ritu Gupta | 27 | F |

Claims Table:

|  |  |  |  |
| --- | --- | --- | --- |
| **claim\_id** | **customer\_id** | **claim\_amount** | **claim\_status** |
| C001 | 101 | 10000 | Approved |
| C002 | 102 | 25000 | Rejected |
| C003 | 103 | 18000 | Approved |
| C004 | 104 | 32000 | Pending |
| C005 | 101 | 15000 | Approved |
| C006 | 106 | 21000 | Approved |

**Expected Output**:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **name** | **total\_claims** | **approved\_claim\_amount** | | Rahul Sharma | 2 | 25000 | | Priya Verma | 1 | 0 | | Amit Patel | 1 | 18000 | | Neha Singh | 1 | 0 | | Sunil Mehra | 0 | 0 | | Ritu Gupta | 1 | 21000 | |  |  |

**Q2.** Customer Premium Behaviour Analysis

**Input Tables**:

Customer Table:

|  |  |  |  |
| --- | --- | --- | --- |
| **customer\_id** | **name** | **age** | **gender** |
| 101 | Rahul Sharma | 35 | M |
| 102 | Priya Verma | 29 | F |
| 103 | Amit Patel | 42 | M |
| 104 | Neha Singh | 31 | F |
| 105 | Sunil Mehra | 48 | M |

Policy Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **policy\_id** | **customer\_id** | **premium\_amount** | **start\_date** | **policy\_type** |
| 201 | 101 | 10000 | 10-01-2023 | Health |
| 202 | 101 | 15000 | 15-06-2023 | Health |
| 203 | 102 | 12000 | 20-03-2023 | Health |
| 204 | 103 | 20000 | 05-02-2023 | Critical Illness |
| 205 | 104 | 18000 | 01-08-2023 | Health |
| 206 | 105 | 22000 | 18-07-2023 | Health |

**Problem Statement** – For each customer, find total premium paid, number of policies purchased, average premium paid, and most recent policy start date. Also, categorize the customers based on total premium paid:

* ‘High Value’ if total premium > 25000
* ‘Medium Value’ if between 15000 and 25000
* ‘Low Value’ otherwise

**Expected Output:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **customer\_name** | **num\_policies** | **total\_premium** | **avg\_premium** | **latest\_policy\_start\_date** | **premium\_category** |
| Rahul Sharma | 2 | 25000 | 12500 | 15-06-2023 | Medium Value |
| Sunil Mehra | 1 | 22000 | 22000 | 18-07-2023 | Medium Value |
| Neha Singh | 1 | 18000 | 18000 | 01-08-2023 | Medium Value |
| Amit Patel | 1 | 20000 | 20000 | 05-02-2023 | Medium Value |
| Priya Verma | 1 | 12000 | 12000 | 20-03-2023 | Low Value |

**PART 2: PySpark**

**Q1.** Write a PySpark script that:

* Reads both files from S3.
* Joins customer and policy data.
* Filters only Active policies.
* Calculates total premium amount paid by each customer.
* Returns the result as a DataFrame.

**Files:**

s3://insurance-data/customers.csv

s3://insurance-data/policies.csv

**Schemas:**

**customers.csv:**

customer\_id,name,city

C001,Rahul,Delhi

C002,Anita,Mumbai

C003,Ajay,Bangalore

**policies.csv:**

policy\_id,customer\_id,premium\_amount,status

P1001,C001,1200,Active

P1002,C002,800,Inactive

P1003,C001,500,Active

P1004,C003,900,Active

**Q2.** Write a PySpark script that:

* Ranks transactions of each customer by amount (highest first).
* Returns top 2 transactions per customer.

**File:**

s3://insurance-data/customer\_transactions.csv

**Schema of file:**

customer\_id,transaction\_id,amount,transaction\_date

C001,T1001,500,2024-01-01

C001,T1002,1200,2024-01-15

C002,T1003,800,2024-02-01

C001,T1004,400,2024-03-10

C002,T1005,1000,2024-03-15

**Q3.** Write a PySpark script that detect Gaps in Customer Transactions

**File:**

s3://lumiq-data/insurance/customer\_transactions.csv

**Schema:**

| customer\_id | transaction\_date |
| --- | --- |
| C001 | 2023-01-10 |
| C001 | 2023-02-12 |
| C001 | 2023-04-15 |
| C002 | 2023-01-05 |
| C002 | 2023-03-07 |

**Task:**

* Find missing months in customer payments.
* Flag if the gap between transactions is more than 30 days.

#### **Output Format:**

| **customer\_id** | **transaction\_date** | **prev\_date** | **days\_diff** | **missing\_month\_flag** |
| --- | --- | --- | --- | --- |
| C001 | 2023-01-10 | null | null | false |
| C001 | 2023-02-12 | 2023-01-10 | 33 | true |
| C001 | 2023-04-15 | 2023-02-12 | 63 | true |
| C002 | 2023-01-05 | null | null | false |
| C002 | 2023-03-07 | 2023-01-05 | 61 | true |

**PART 3: Python**

**Q1.** You are given a list of customer transactions in JSON format. Each record contains customer\_id, transaction\_id, and amount. Return the total transaction amount per customer.

Sample Input:

transactions = [

{"customer\_id": "C001", "transaction\_id": "T1001", "amount": 1200},

{"customer\_id": "C002", "transaction\_id": "T1002", "amount": 850},

{"customer\_id": "C001", "transaction\_id": "T1003", "amount": 500},

{"customer\_id": "C003", "transaction\_id": "T1004", "amount": 700},

{"customer\_id": "C002", "transaction\_id": "T1005", "amount": 150}

]

Expected Output:

{

"C001": 1700,

"C002": 1000,

"C003": 700

}

**Q2.** Given a list with duplicate integers, return a list with duplicates removed but maintain the original order

Input:

numbers = [4, 5, 2, 4, 5, 6, 7, 2]

Expected Output:

[4, 5, 2, 6, 7]

**Q3.** You are given a stream of events where each event has a user ID and timestamp.  
Return only the latest event per user

Input List:

events = [

{"user\_id": "U1", "timestamp": "2025-07-23T10:01:00", "action": "login"},

{"user\_id": "U2", "timestamp": "2025-07-23T10:03:00", "action": "view"},

{"user\_id": "U1", "timestamp": "2025-07-23T10:05:00", "action": "logout"},

]

Expected Output:

{

"U1": {"timestamp": "2025-07-23T10:05:00", "action": "logout"},

"U2": {"timestamp": "2025-07-23T10:03:00", "action": "view"},

}

**PART 4: Data Engineering (MCQs)**

**Q1.** In a Slowly Changing Dimension (SCD) Type 2 implementation, what is typically done to track history?

A. Overwrite old data  
B. Update existing rows with new values  
C. Create new rows for each change with effective date range  
D. Ignore changes and store only the latest record

**Q2.** What is a primary benefit of using Apache Hudi or Delta Lake over plain Parquet files in a data lake?

A. Data compression  
B. Support for schema-on-write  
C. ACID transactions and record-level updates  
D. Better visualization

**Q3.** What causes a Spark job to create multiple stages?

A. Use of wide transformations that require shuffling  
B. Caching data  
C. Running in local mode  
D. Use of UDFs

**Q4.** What is the purpose of XComs in Airflow?

A. To handle retries  
B. To move files across systems  
C. To exchange messages or data between tasks  
D. To define external triggers

**Q5.** How can you handle task failures in Airflow to retry a task automatically?

A. Use depends\_on\_past=True  
B. Set retries and retry\_delay in task definition  
C. Enable dagrun\_timeout  
D. Set catchup=False

**Q6.** When designing a hospital dimension table, which of the following would not be ideal as a dimension attribute?

A. Hospital Name  
B. City  
C. Accreditation Status  
D. Number of Claims Filed

**Q7.** You have a DAG that fetches agent incentive data and sends it to an API. How can you ensure that the API task runs only if the data fetching task succeeds?

A. Use depends\_on\_past=True  
B. Use trigger\_rule="all\_failed"  
C. Set task dependency using >>  
D. Set retries=0

**Q8.** Which of the following SQL statements returns all customers and their orders, even if they haven’t placed one?  
A. SELECT \* FROM Customers INNER JOIN Orders ON Customers.id = Orders.customer\_id;  
B. SELECT \* FROM Customers RIGHT JOIN Orders ON Customers.id = Orders.customer\_id;  
C. SELECT \* FROM Customers LEFT JOIN Orders ON Customers.id = Orders.customer\_id;  
D. SELECT \* FROM Customers CROSS JOIN Orders;

**Q9.** Which transformation is considered a wide transformation in Spark?

A. map()

B. filter()

C. groupByKey()

D. flatMap()

**Q10.** What is the difference between coalesce() and repartition() in Spark?

* 1. coalesce() increases partitions  
     B. repartition() is lazy  
     C. coalesce() avoids full shuffle  
     D. repartition() merges small files

Q93: You are given an array of strings strs.  
Group the strings that are **anagrams** of each other.

Input: strs = ["eat","tea","tan","ate","nat","bat"]

Output: [["bat"],["nat","tan"],["ate","eat","tea"]]

Q: 94 **Question: Top K Frequent Elements**

Given an integer array nums and an integer k, return the k most frequent elements.

* You may return the answer in **any order**.
* If multiple elements have the same frequency, any valid order is acceptable.

Input: nums = [1,1,1,2,2,3,3], k = 2

Output: [1,2]

Q 95: Table: Transactions

Columns: id, country, state, amount, trans\_date

id is the primary key of this table.

The table has information about incoming transactions.

The state column is an enum of type ["approved", "declined"].

Write an SQL query to find for each month and country, the number of transactions and their total amount, the number of approved transactions and their total amount.

Q 96: count of rows after left, right, inner, left\_anti joins and if we are joining two tables on left\_anti with 2 or more columns then what are the rows we will get.

Q: 97 How we resolve data skweness in data and for example if we have data like three columns a, b, c and a is highly skewed then how to resolve this skweness if we have to group by on three columns and find count.

Q:98 If I have two very large tables and I have to join(by default spark uses suffle\_hash join) them and our job is failing again and again so how to resolve this. Hint bucketing on primary key columns