Dunnhumby Data Engineer Interview Guide – Experienced 2+

Technical Round 1:

1. Explain your projects on which you worked till now and what was your role?
2. Name the tools and technologies you have worked with to date.
3. What is the difference between static and dynamic partitioning in Hive?
4. What is the difference between narrow and wide transformations in Apache Spark? Explain with examples.
5. What is the difference between repartition and coalesce in Apache Spark?
6. What are the different modes in which you can submit Spark jobs? Explain each.
7. When submitting Spark jobs, how does the process work in the backend? Explain.
8. In Spark, what is the difference between cores and executors?
9. What is the difference between external and internal tables in Hive?
10. If manual partitions are created in a Hive data-warehouse table directory, and you query records from those partitions, will you see the data? If not, how can this be fixed?
11. How can you delete partitions from a table in Hive using a command?
12. Write a SQL query to find distinct IDs from a table where the count is more than 1 and greater than 200.
13. Write the Spark command to read a CSV file.
14. What is the difference between Pandas DataFrame and Spark DataFrame? When would you prefer using each?
15. Write the Spark command to add a new column to a DataFrame.
16. Write the Spark command to rename an existing column in a DataFrame.
17. Given a DataFrame with columns id and name, add a new column department with values:
    - If id < 100, assign "HR".
    - If id >= 100 and id < 200, assign "admin".
18. Have you worked on Data Warehousing?

Technical Round 2:

PySpark Scenario Questions

1. **Data Transformation**:
   You have a dataset with user login details. Write a PySpark job that calculates the number of unique users who logged in per day, but exclude any logins from inactive users listed in a separate file.
2. **Data Partitioning**:
   You need to process a large sales dataset partitioned by region and date. Explain how you would use **repartition** or **coalesce** effectively to optimize processing when analyzing data only for a specific region.
3. **Schema Evolution**:
   A JSON file with evolving schema needs to be ingested into a DataFrame. How would you handle new fields dynamically in PySpark without breaking the job for previous structures?
4. **Joins Optimization**:
   Describe how you would optimize a join between two large tables where one is significantly smaller, using **broadcast joins** in PySpark.
5. **Data Quality**:
   Write a PySpark script to check for missing values and duplicate rows in a DataFrame. How would you ensure data quality before saving it to a storage system?

Airflow Scenario Questions

6. **Dependency Management**:
   You need to create a workflow where Task B runs only if Task A is successful, and Task C should always run regardless of Task A or B's status. How would you define this dependency using Airflow?
7. **Dynamic DAG Creation**:
   A data pipeline processes files for different clients stored in separate directories. Explain how you would use dynamic DAG creation to handle client-specific workflows in Airflow.
8. **Retries and Failure Handling**:
   A task intermittently fails due to external API limitations. How would you configure Airflow retries and alerts to manage this situation efficiently?
9. **Data Backfill**:
   You have to rerun a data pipeline for the last three months due to a bug. Describe how you would set up a backfill process using Airflow without disrupting ongoing workflows.
10. **Task Parallelism**:
    Suppose you have a DAG that ingests data from multiple databases. How would you increase task parallelism in Airflow to improve performance without overloading the system?

Kafka Scenario Questions

11. **Consumer Lag Monitoring**:
    Your Kafka consumer shows significant lag during peak hours. What strategies would you employ to reduce lag and ensure timely data processing?
12. **Partitioning Strategy**:
    You need to design a Kafka topic for a logging service. How would you decide the number of partitions and the key for partitioning to balance throughput and ordering requirements?
13. **Handling Message Failures**:
    If a consumer fails to process a message due to data corruption, describe how you would configure Kafka to handle retries and avoid message loss.
14. **Kafka Connect**:
    Explain how you would use Kafka Connect to ingest data from a relational database into Kafka while ensuring minimal latency and exactly-once semantics.
15. **Schema Evolution**:
    Your Kafka producer schema has changed, and the new data includes additional fields. How would you ensure backward compatibility using **Schema Registry** while consuming data from the same topic?

Glassdoor Dunnhumby Review –

https://www.glassdoor.co.in/Reviews/dunnhumby-Reviews-E195922.htm

Dunnhumby Careers –

https://www.dunnhumby.com/careers/

Subscribe to my YouTube Channel for Free Data Engineering Content –

https://www.youtube.com/@shubhamwadekar27

Connect with me here –

https://bento.me/shubhamwadekar

Checkout more Interview Preparation Material on –

https://topmate.io/shubham_wadekar