Problem Statement:

Imagine you are a data engineer working for AdvertiseX, a digital advertising technology company. AdvertiseX specializes in programmatic advertising and manages multiple online advertising campaigns for its clients. The company handles vast amounts of data generated by ad impressions, clicks, conversions, and more. Your role as a data engineer is to address the following challenges:

Data Sources and Formats:

Ad Impressions:

Data Source: AdvertiseX serves digital ads to various online platforms and websites.

Data Format: Ad impressions data is generated in JSON format, containing information such as ad creative ID, user ID, timestamp, and the website where the ad was displayed.

Clicks and Conversions:

Data Source: AdvertiseX tracks user interactions with ads, including clicks and conversions (e.g., sign-ups, purchases).

Data Format: Click and conversion data is logged in CSV format and includes event timestamps, user IDs, ad campaign IDs, and conversion type.

Bid Requests:

Data Source: AdvertiseX participates in real-time bidding (RTB) auctions to serve ads to users.

Data Format: Bid request data is received in a semi-structured format, mostly in Avro, and includes user information, auction details, and ad targeting criteria.

Case Study Requirements:

Data Ingestion:

Implement a scalable data ingestion system capable of collecting and processing ad impressions (JSON), clicks/conversions (CSV), and bid requests (Avro) data.

Ensure that the ingestion system can handle high data volumes generated in real-time and batch modes.

Data Processing:

Develop data transformation processes to standardize and enrich the data. Handle data validation, filtering, and deduplication.

Implement logic to correlate ad impressions with clicks and conversions to provide meaningful insights.

Data Storage and Query Performance:

Select an appropriate data storage solution for storing processed data efficiently, enabling fast querying for campaign performance analysis.

Optimize the storage system for analytical queries and aggregations of ad campaign data.

Error Handling and Monitoring:

Create an error handling and monitoring system to detect data anomalies, discrepancies, or delays.

Implement alerting mechanisms to address data quality issues in real-time, ensuring that discrepancies are resolved promptly to maintain ad campaign effectiveness.

This Ad Tech case study scenario focuses on the challenges and data formats commonly encountered in the digital advertising industry. Candidates can use this information to design a data engineering solution that addresses the specific data processing and analysis needs of AdvertiseX.

Solution:

Data Sources and Formats:

1. **Ad Impressions:**

   Data Source: AdvertiseX serves digital ads to various online platforms and websites.

   Data Format: JSON

   Example :

   ```
   {
       "ad_creative_id": "12345",
       "user_id": "abc123",
        "timestamp": "2024-06-12T12:34:56Z",
       "website": "example.com"
       }
   ```

2. **Clicks and Conversions:**

   Data Source: AdvertiseX tracks user interactions with ads, including clicks and conversions (e.g., sign-ups, purchases).

   Data Format: CSV

   Example:

   event_timestamp, user_id, ad_campaign_id, conversion_type

   2024-06-12 12:34:56,abc123,98765,click

3. **Bid Requests:**

   Data Source: AdvertiseX participates in real-time bidding (RTB) auctions to serve ads to users.

   Data Format: Avro (semi-structured)

   Example Schema:

   ```
   {
    "type": "record",
    "name": "BidRequest",
    "fields": [
      {"name": "user_id", "type": "string"},
      {"name": "auction_id", "type": "string"},
      {"name": "ad_targeting_criteria", "type": "string"}
    ]
   }
   ```

Detailed Steps to Perform the Task

I. Data Ingestion

Objective: Implement a scalable data ingestion system capable of collecting and processing ad impressions (JSON), clicks/conversions (CSV), and bid requests (Avro) data. Ensure the ingestion system can handle high data volumes in real-time and batch modes.

Steps:

1. **Set Up Real-time Data Streaming:**

- Tool: Apache Kafka
- Implementation:
  - Create Kafka topics: **ad_impressions**, **clicks_conversions**, and **bid_requests**.
  - Configure Kafka Connectors to ingest data from sources:
    - **Ad Impressions (JSON):** Configure a connector to read JSON data from online platforms/websites.
    - **Clicks/Conversions (CSV):** Configure a connector to read CSV data.
    - **Bid Requests (Avro):** Configure a connector to read Avro data.\
- Example Commands:

  kafka-topics.sh --create --topic ad_impressions --bootstrap-server localhost:9092 --partitions 10 --replication-factor 3

  kafka-topics.sh --create --topic clicks_conversions --bootstrap-server localhost:9092 --partitions 10 --replication-factor 3

  kafka-topics.sh --create --topic bid_requests --bootstrap-server localhost:9092 --partitions 10 --replication-factor 3

2. Batch Data Ingestion:

- Tool: AWS S3 for storage, AWS Glue for ETL.
- Implementation:
  - Store CSV files in Amazon S3.
  - Schedule AWS Glue jobs to process batch data.
- Example Setup:

  Yaml File->

  S3Bucket: advertiseX-data
  S3Folder: clicks_conversions
  GlueJobName: process_clicks_conversions

II. Data Processing

Objective: Develop data transformation processes to standardize and enrich the data. Handle data validation, filtering, and deduplication. Implement logic to correlate ad impressions with clicks and conversions to provide meaningful insights.

Steps:

1 **Develop ETL Pipelines:**
- **Tool:** Apache Spark or AWS Glue
- **Implementation:**
  - Read data from Kafka topics and S3 buckets.

- Standardize formats (e.g., convert timestamps to a standard format).
- Enrich data (e.g., add geolocation based on IP).
- Validate data (e.g., check for missing or malformed fields).
- Filter out invalid or irrelevant data.
- Deduplicate records.
- Correlate data to link ad impressions with clicks and conversions.

III. Data Storage and Query Performance

Objective: Select an appropriate data storage solution for storing processed data efficiently, enabling fast querying for campaign performance analysis. Optimize the storage system for analytical queries and aggregations.

Steps:

1. **Select Data Storage Solution:**
   - **Tool:** Amazon Redshift for data warehousing, Amazon S3 for data lake.
   - **Implementation:**
     - Store processed data in Amazon Redshift.
     - Store raw and processed data in Amazon S3.
1. **Optimize Query Performance:**

- **Implementation:**
  - Partition tables by date and campaign ID.
  - Create indexes on frequently queried columns.
  - Use columnar storage and compression.

IV. Error Handling and Monitoring

Objective: Create an error handling and monitoring system to detect data anomalies, discrepancies, or delays. Implement alerting mechanisms to address data quality issues in real-time.

Steps:

1. **Set Up Error Handling:**
   - **Tool:** Apache Airflow
   - **Implementation:**
     - Define workflows with error handling logic.
     - Log errors to a centralized system (e.g., AWS CloudWatch).

2 Monitoring and Alerting:

- **Tool:** Amazon CloudWatch, Great Expectations
- **Implementation:**

  - Set up CloudWatch alarms for monitoring ETL job statuses and data anomalies.
  - Use Great Expectations for data quality checks.
  - Configure alerting to notify stakeholders of issues.

Conclusion

This step-by-step plan ensures that AdvertiseX can efficiently manage, process, and analyze large volumes of data generated from ad impressions, clicks/conversions, and bid requests. The proposed solution covers real-time and batch data ingestion, robust data processing, optimized data storage for fast querying, and comprehensive error handling and monitoring mechanisms. This will enable AdvertiseX to derive meaningful insights and maintain high data quality, ultimately supporting effective ad campaign management.