PAYNET

Please explain in detail the following and elaborate on your Data Engineering knowledge, it will good if there are specific used cases you want to use for example.

1. **What is your Data platform stack, ETL / ELT and Visualisation?**
    - Cloud Platform: AWS, GCP, Snowflake, Databricks
    - Programming: Python, SQL, R
    - Tools: Airflow, DBT, Docker
    - Distributed Framework & Big Data : Spark, Hadoop, Hive
    - Visualisations: Tableau, PowerBI, DOMO, Python (Matplotlib, seaborn)
2. **What is the entire lifecycle to operate datalake end-to-end?**
    - Ingestion: Collect structured/semi-structured data from various sources (databases, APIs, logs, csv, parquet, json). Use spark to ingest (schema inference, file reformatting json→ parquet), partitioning, don't do full cleansing) data into warehouse, GCS or S3
    - Raw Zone (Landing): Store incoming data in its original format (e.g., GCS or S3). Can use AWS Athena or equivalent to do some ad hoc analysis, to see what kind of data is needed to load into Redshift for transformation. Use AWS Glue crawlers or equivalents (e.g., Hive metastore, Unity Catalog) to auto-discover and catalog schemas, enabling data discovery, schema evolution tracking, and lineage analysis.
    - Staging/Cleansing: Apply delta loads, deduplication, format normalization via SQL with DBT.
    - Transformation: Use dbt to build layered models (staging → intermediate → marts).

      Staging: Light cleanup and standardization (e.g. rename columns, cast types)

      Intermediate: Joins, window functions, calculated fields

      Marts: Aggregate metrics and reporting tables (e.g. DAU, LTV, ARPPU)
    - Storage: Persist transformed data in Redshift, Snowflake, BigQuery, or Hive.
    - Governance & Quality: Implement validation with dbt tests and automated row/sum checks (e.g., in Tencent & Colgate).

- Serving Layer: Visualizations in Tableau (Hong Leong), dashboards for business teams after building XGBoost time series forecasting model & Clustering Model (HDBSCAN + GMM)
- Monitoring & Logging: Use Grafana, Airflow logs, and alerting for pipeline health.
- IAM Policies: Implement fine-grained access control using IAM policies, table- or column-level masking, and audit logs to ensure secure and compliant data access.

3. What did you work on across the lifecycle of the data and what are the tools that you used?

- Extraction: GCP, AWS, Snowflake. ClickHouse, PostgreSQL, API calls (Tencent, Colgate, Hong Leong).
- Transformation: Snowflake with DBT (Colgate), GCP & AWS with DBT (Tencent), Postgresql & Python & Spark (Hong Leong)
- Orchestration: Airflow, GCP Composer, Step functions (AWS)
- Validation: Custom-built row count checks, sum comparisons, random sampling, and dbt testing.
- Visualization: Tableau dashboards for machine learning models metrics (forecasting, clustering in Hong Leong Bank), DOMO dashboards for product performance in Colgate

4. What is a rather complex workflow you've worked on and its targeted use case?

Utilized a 5-layer data modelling structure to compute essential game KPIs including DAU, LTV, ARPPU, and more at Tencent.

- Extended existing pipeline to support a new game with different schema
- Identified missing fields (e.g., no country) and enriched data by joining with dimension/fact tables
- Customized dbt models to compute new metrics (e.g., revenue by country)
- Ensured consistency with original pipeline while adapting to new data structures

- During dbt v1.5 → v1.7 upgrade, ran automated data validation across all games final reporting tables (total 100++ tables)
- Used row counts, sum checks, and schema validation to catch silent regressions
- Outcome: scalable, consistent cross-game analytics with validated outputs

5. **Data Processing knowledge (data warehousing)**

   For example AWS, I ingest raw data into S3, then use Athena for ad hoc analysis to decide what needs loading into Redshift—saving storage and compute costs. dbt handles transformation in Redshift or Snowflake using layered models. To manage cost further, I apply lifecycle rules: older data moves to S3 Infrequent Access or Glacier for long-term storage.

6. **Data orchestration tool (airflow is what the team uses)**

   Airflow, GCP Composer (Airflow), Step Functions (AWS)

7. **Data Platform knowledge (DevOps experience)**

   I'm familiar with using Docker and Kubernetes in data engineering workflows. While the Kubernetes environments were set up by the platform team, I've deployed and scheduled pipelines within those clusters. On the Docker side, I've built custom images to package dependencies and libraries for Python-based workflows, ensuring portability across dev and prod environments. I also version control all code and dbt projects via Git.

8. **Large data and dynamic data handling (thinking beyond and considering data size and performance**

   At Tencent, I processed over 15 million rows of PUBG game data per hour. To optimize performance, I applied partitioning and caching, and addressed data skew by implementing key salting for heavily skewed columns. At Hong Leong Bank, I handled dynamic customer data using delta load logic in PostgreSQL to efficiently refresh downstream tables with minimal data movement.

PAYNET

Visualisations

https://public.tableau.com/app/profile/seong.wei.chin/viz/paynet_assessment/Dashboard1?publish=yes

https://public.tableau.com/app/profile/seong.wei.chin/viz/paynet_assessment/Dashboard2?publish=yes