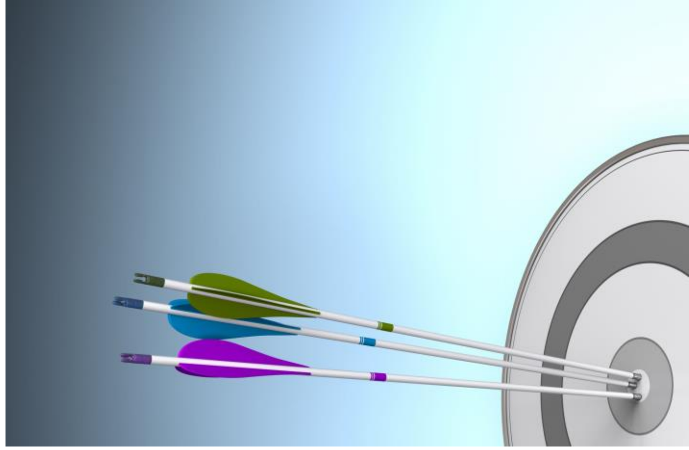
Corndel.

Workshop 9

Advanced Stream Processing Pipelines

Workshop Objectives



Learn how to work with:

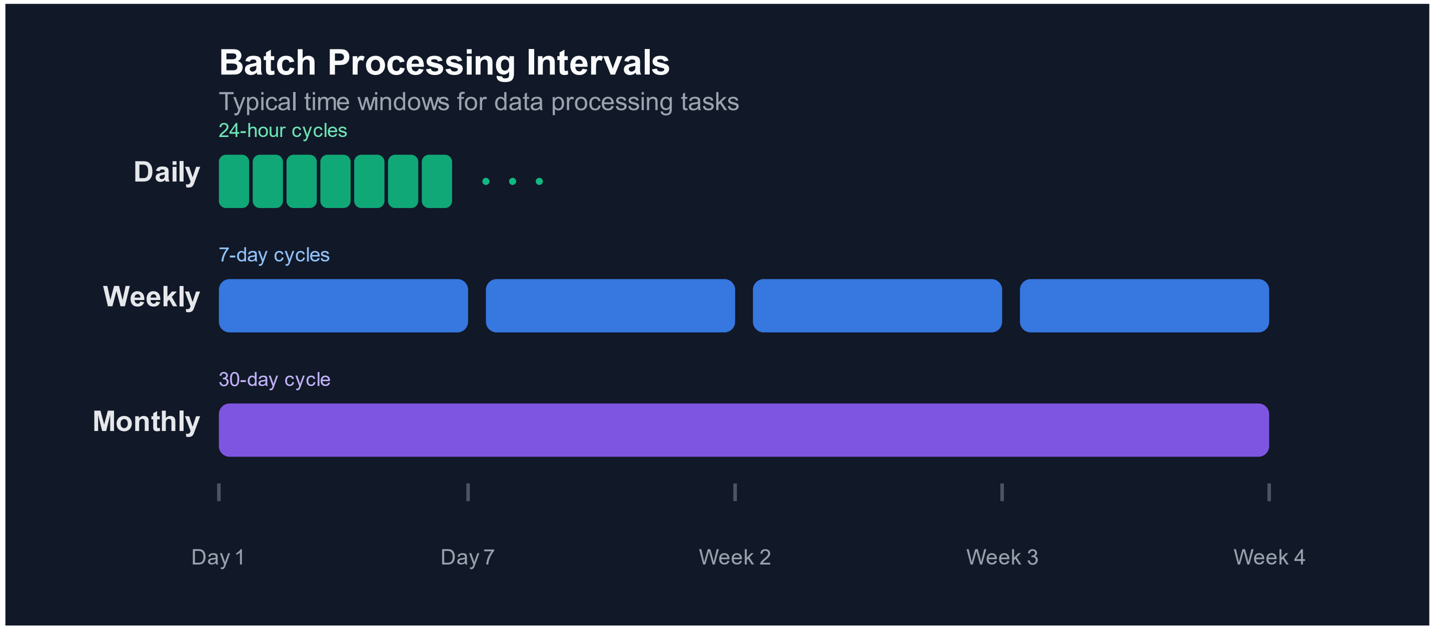
1. **Batch processing:** Create automated batch processing pipelines to keep dimensional models current as source data changes (OLTP to OLAP updates)
2. **Data staging:** Implement reliable data movement using staging tables to transfer data between source and target systems safely
3. **Monitoring:** Implement audit logging tables to track successful loads and provide operational monitoring
4. **Dimension updates:** Apply Type 1 Slowly Changing Dimension (SCD) updates to overwrite old customer data with new values, maintaining only current information

Why Batch Processing?

- Common Use Cases
  - Daily sales reports and dashboards
  - Weekly performance summaries
  - Monthly analytics refreshes

What regular reporting or analytics tasks in your organisation currently use or might benefit from batch processing?

How frequently do these need to update?

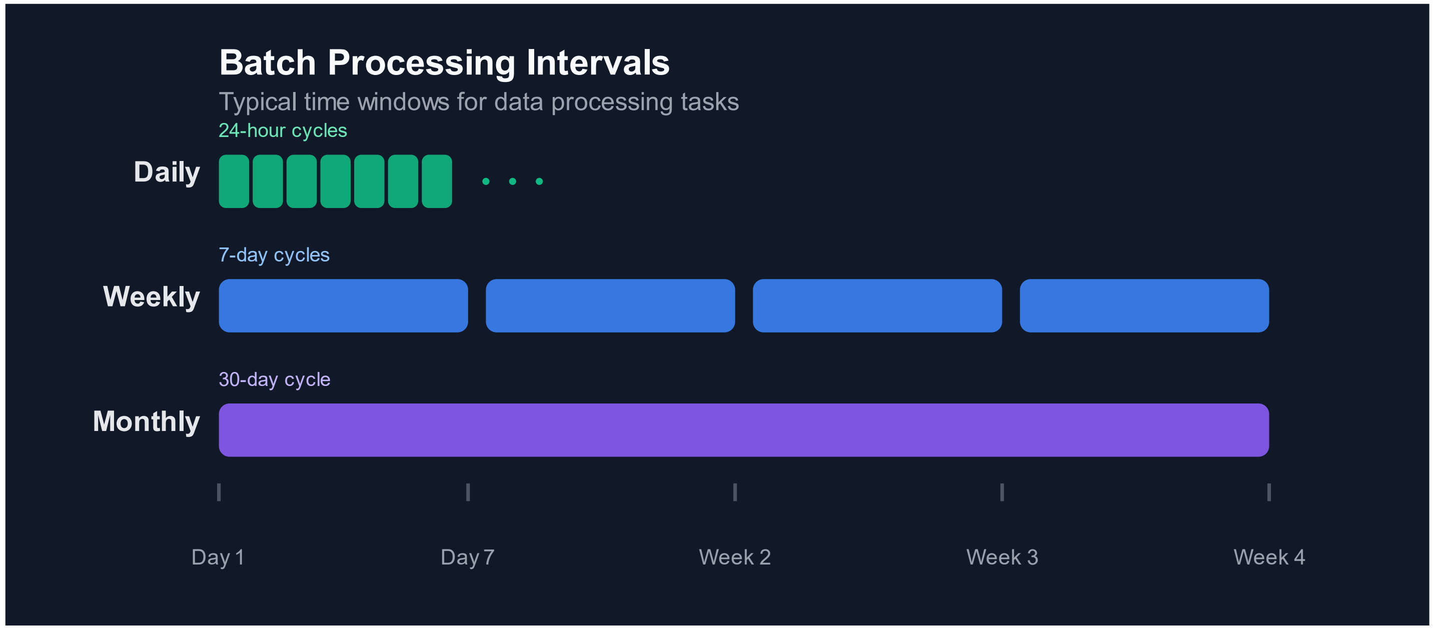


Why Batch Processing?

- Key Benefits
  - Efficient processing of large data volumes
  - Predictable resource usage
  - Simple scheduling and monitoring

Looking at these benefits, where in your organisation have you seen these work well or struggle?

For example, has scheduling regular loads ever been challenging?



Workshop Alignment with IFATE Pass Descriptors

Star Schemas and Data Warehousing (K15)	Building and maintaining a star schema with Type 1 SCD updates in Azure Synapse
Data Engineering Tools (K20)	Practical experience with Azure SQL, Azure Synapse, and Dedicated SQL Pools
Pipeline Deployment (K8)	End-to-end deployment of automated batch update pipeline
Data Store Monitoring (K1, S7)	Implementing audit logging and pipeline monitoring for reliable updates

Data engineer / Institute for Apprenticeships and Technical Education

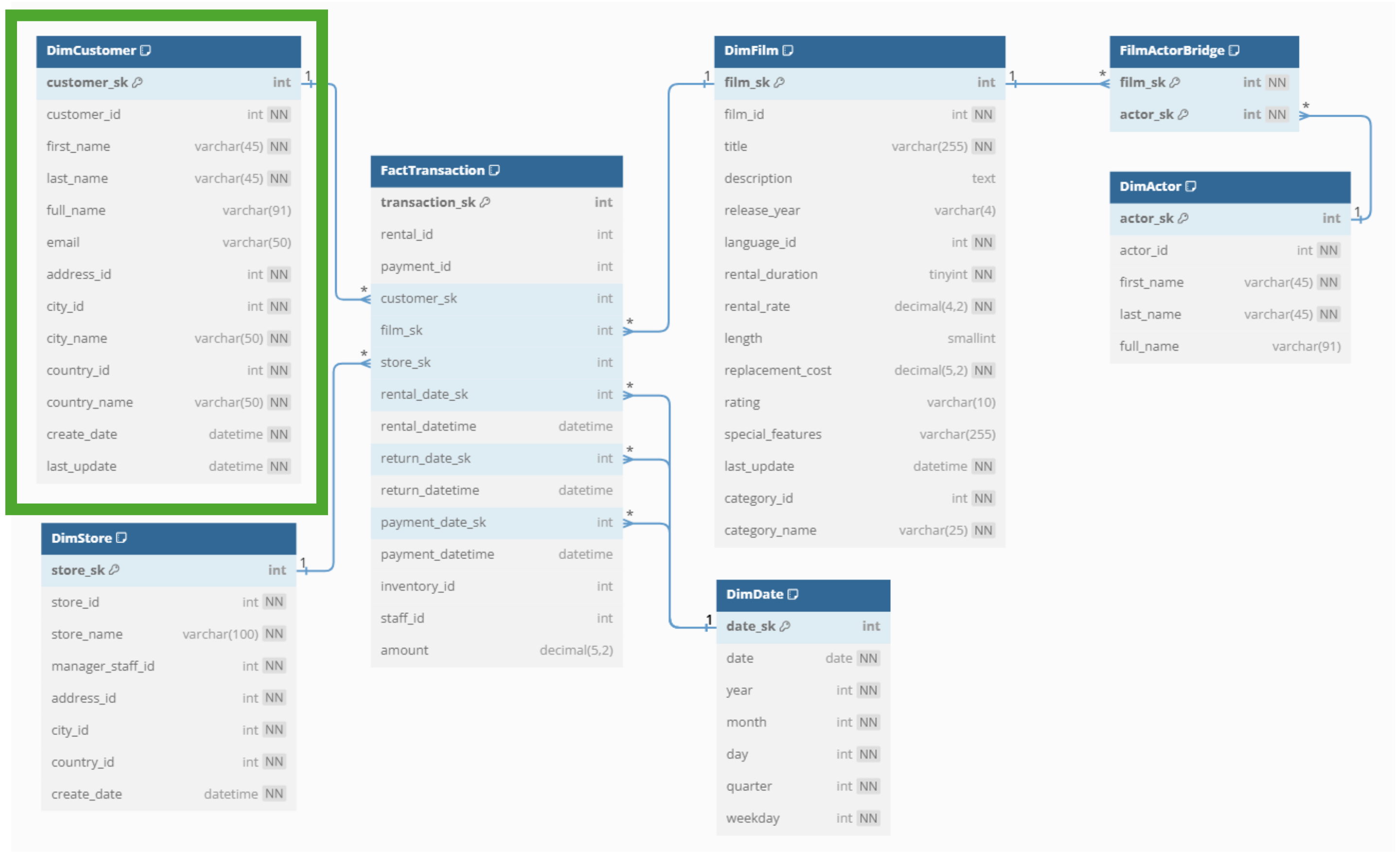
Morning activity

Building the pipeline

What We'll Build Today

- Automated data warehouse updates using Azure Synapse
- Focus on Customer dimension maintenance
- Type 1 SCD implementation with audit logging

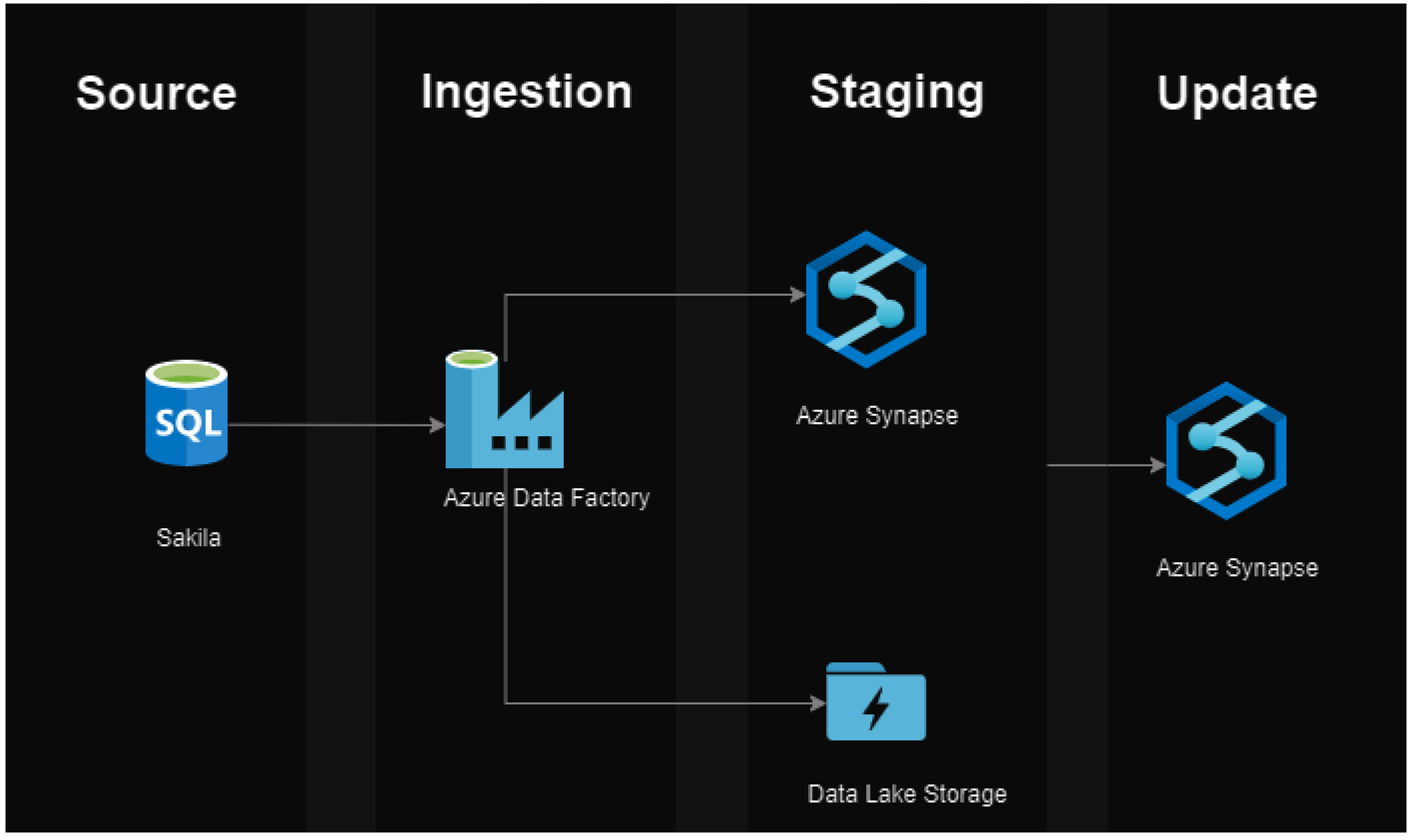
Building on Workshop 3's static star schema and Workshop 6's design work



Our Architecture

- Key Components
  - **Source:** Sakila OLTP database in Azure SQL (OLTP)
  - **Ingestion:** Azure Data Factory pipeline
  - **Staging:** Choice of:
    - Azure Synapse staging tables
    - Data Lake Storage
  - **Update:** Azure Synapse dimension table (OLAP)

- Pipeline Operations.
  1. Extract the latest customer data
  2. Stage with audit columns
  3. Process Type 1 SCD updates



Workshop Flow

Morning

- Building the Pipeline
    1. Environment setup
    2. Schema and table creation
    3. Pipeline development
    4. Testing with simulated changes
- Going Further (time permitting)
    - Data Lake integration
    - Data Flow transformations
    - PySpark notebook processing

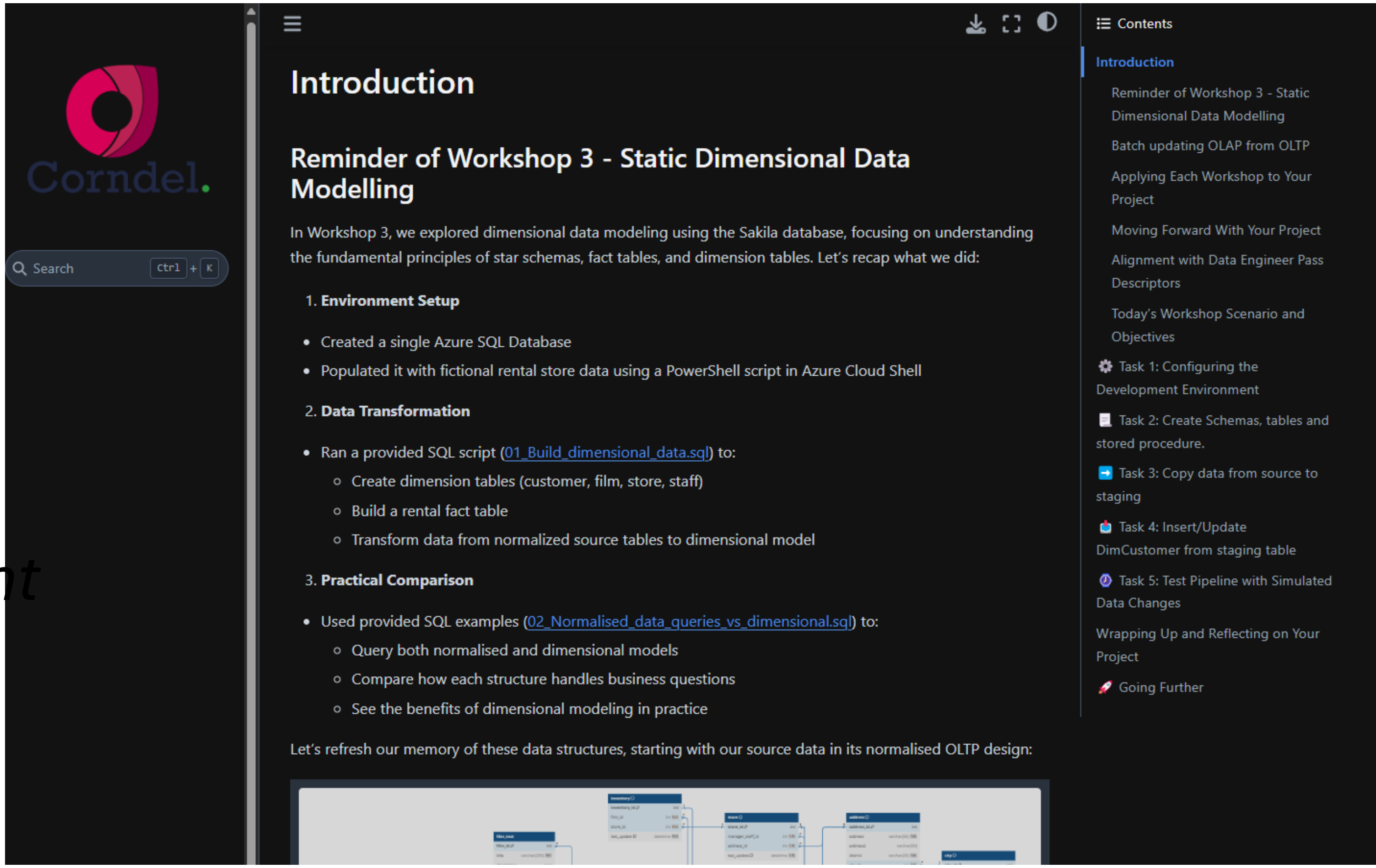
Afternoon

- Workshop Reflection
    - What went well?
    - Technical challenges?
    - Learning points and insights?
- Project Application
    - Project data sources
    - Update patterns and data models
    - Implementation strategies

Workshop Instructions: Your Step-by-Step Guide

- Key concepts and terminology
- Task breakdown and dependencies
- Optional "Going Further" sections
- How to ask for help

We will now walk through the document structure and highlight how each section builds on the previous one to create a complete pipeline



Introduction

Reminder of Workshop 3 - Static Dimensional Data Modelling

In Workshop 3, we explored dimensional data modeling using the Sakila database, focusing on understanding the fundamental principles of star schemas, fact tables, and dimension tables. Let's recap what we did:

1. **Environment Setup**

- Created a single Azure SQL Database
- Populated it with fictional rental store data using a PowerShell script in Azure Cloud Shell

2. **Data Transformation**

- Ran a provided SQL script (01_Build_dimensional_data.sql) to:
  - Create dimension tables (customer, film, store, staff)
  - Build a rental fact table
  - Transform data from normalized source tables to dimensional model

3. **Practical Comparison**

- Used provided SQL examples (02_Normalised_data_queries_vs_dimensional.sql) to:
  - Query both normalised and dimensional models
  - Compare how each structure handles business questions
  - See the benefits of dimensional modeling in practice

Let's refresh our memory of these data structures, starting with our source data in its normalised OLTP design:

Afternoon activity

Reflecting on the morning activity and how it applies to your project

Workshop reflection

⚙️ Technical Implementation

- 1 ✨ Which parts of the pipeline came together smoothly?
- 2 🚧 Where did you encounter challenges?
- 3 🔎 What debugging approaches worked best?

🛣️ Learning Journey

- 1 💡 What key concepts clicked today?
- 2 🤔 Which concepts need more exploration?
- 3 ⚖️ How does this compare to any previous pipeline work?

Build on Your Learning from E-Learning Module 06.3

- The next questions in the following two slides distil key ideas from **06.3 Batch Processing: OLTP Data Source Integration**, which you may have completed prior to this workshop.
- If you haven't yet completed it, we encourage you to do so. It provides:
  - A foundation in batch processing concepts and their practical applications.
  - Key reflective prompts to align these ideas with your project and role.
- Use questions in 6.3 to deepen your understanding of:
  - The batch processing patterns most relevant to your project.
  - Robust pipeline design, update patterns, and optimisation techniques.
  - How these align with user and business needs
- These reflections aren't just for today:
  - Use them as a starting point for future discussions with your PDE in 121s.
  - Identify areas for improvement and questions to explore further with your PDE.

Project Application Discussion

📊 Project data sources

- 1 What type(s) of data sources does your project handle? 🗄 OLTP databases? 📄 Files? 🌐 APIs?
- 2 What challenges do the formats present? 🚧
- 3 What access methods are needed for each source type, and how are they configured securely? 🔐
- 4 Does the origin of your data (e.g., normalised tables, JSON files) impact your pipeline design? 📝
- 3 What security and access patterns do you need to consider? 🔒

🔄 Update patterns

- 1 What update pattern best fits your use case? 🔄 Full refresh 🔀 incremental updates 📝 CDC?
- 2 🤔 What drives this choice (data volume, change frequency, etc)?
- 3 How can you track and validate changes? ✅
- 4 What could be your strategy for handling failed updates or reprocessing? 🚨

Project Application Discussion

🧩 Data Modelling & Requirements

- 1 Have you defined clear user and business requirements yet? 📝
- 2 Does your chosen data model support these effectively? 🎯
- 3 How do you handle historical data needs? Is Type 1 SCD sufficient or do you need Type 2? 🔁
- 4 How do you validate data quality? ✅

🚀 Implementation and Optimisation

- 1 What tools and frameworks are you using/considering? ☁️ Cloud-native services? 🖥️ Local or on-prem tools? 🎵 Orchestration tools?
- 2 🤔 Why are these the right choices for your needs?
- 3 How are you optimising performance? 📊 Partitioning strategy? 🏷️ Indexing approach? 📂 Compression techniques? 🔍 Query optimisation? 🖥️ Hardware/resource allocation? ⚙️ Parallel processing?
- 4 What monitoring and logging is appropriate? 📈
- 5 Are your pipelines idempotent, and how do you handle retries?