PHASE 0 – Environment & Context Setup

Setup Tasks

- Open existing Databricks Data Engineering project
- Open VS Code with GitHub Copilot enabled
- Confirm access to:
    - Python ETL code
    - Databricks notebooks
    - Delta tables
    - Power BI model

Context Alignment

- Review existing DE pipeline (Bronze → Silver → Gold)
- Identify repetitive / boilerplate tasks suitable for LLM help
- Identify logic-heavy areas requiring human judgment

PHASE 1 – Prompt Engineering Fundamentals (Applied)

Goal

Teach participants to **structure prompts** instead of typing random comments.

Prompt Structure to Practice

Participants must use:

```
Role + Task + Context + Constraints + Output format
```

Tasks

- Write a prompt defining Copilot's role as "Senior Data Engineer"
- Specify Spark / Delta / Databricks context
- Add constraints (incremental load, schema enforcement, etc.)

Example Prompt (Comment)

```
# You are a senior data engineer.
# Write a PySpark function for incremental ETL ingestion
# using Delta Lake MERGE with watermark logic.
# Ensure idempotency and schema enforcement.
```

PHASE 2 – Prompt Engineering for Python ETL

Goal

Use prompts to generate **ETL code**, then refine it.

2.1 ETL Scaffolding

Tasks

- Write a high-level ETL description as a prompt

- Allow Copilot to generate ETL function structure
- Review generated code line by line

Prompt Example

```
# Generate an ETL pipeline in PySpark that:
# - Reads sales data
# - Cleans and validates records
# - Applies incremental logic
# - Writes to Delta Lake
```

2.2 Prompt Refinement

Tasks

- Identify missing logging
- Add prompt asking for error handling
- Add prompt asking for modular functions

Refinement Prompt

```
# Improve the above ETL code by adding:
# - Logging
# - Exception handling
# - Parameterized paths
```

2.3 Validation Task

- Identify at least **3 incorrect assumptions** in Copilot code
- Fix logic manually
- Explain why Copilot failed

PHASE 3 – Prompt Engineering for Spark SQL & Delta Lake

Goal

Use prompts to generate **Spark SQL and Delta MERGE logic**.

3.1 Spark SQL Aggregations

Tasks

- Write prompt for business aggregation
- Generate SQL using Copilot
- Validate grouping & joins

Prompt Example

```
-- As a data engineer, write Spark SQL to calculate
-- daily revenue by region using Delta tables
```

3.2 Delta Lake MERGE Prompts

Tasks

- Prompt Copilot to write MERGE logic
- Verify primary keys
- Add schema evolution manually

Prompt Example

```
# Write PySpark code to merge incremental sales data
# into a Delta table using transaction_id as key
```

3.3 Prompt Anti-Patterns

- Identify hallucinated columns
- Identify inefficient joins
- Identify missing partition logic

PHASE 4 – Prompt Engineering for Power BI DAX

Goal

Use LLMs to accelerate **DAX creation**, not replace BI modeling.

4.1 DAX Prompting

Tasks

- Use Copilot to generate simple measures
- Use Copilot to generate time intelligence logic

Prompt Example

```
-- Create a DAX measure for month-over-month revenue growth
```

4.2 Refinement

Tasks

- Fix incorrect filter context
- Adjust for missing dates
- Validate results against SQL output

PHASE 5 – Prompt Evaluation & Documentation (CRITICAL)

Goal

Develop **responsible LLM usage skills**.

5.1 Prompt Engineering Log (Mandatory)

Participants must maintain a **Prompt Log**:

Use Case	Prompt	Copilot Output	Issues Found	Final Fix

5.2 Reflection Questions

Participants must answer:

- Where did prompt clarity improve output?
- Where Copilot failed despite good prompts?
- What should NEVER be delegated to an LLM?

FINAL DELIVERABLES

Mandatory Submissions

- Updated ETL code (with Copilot assistance)
- Spark SQL & Delta MERGE scripts
- Power BI measures
- Prompt Engineering Log
- Reflection document
