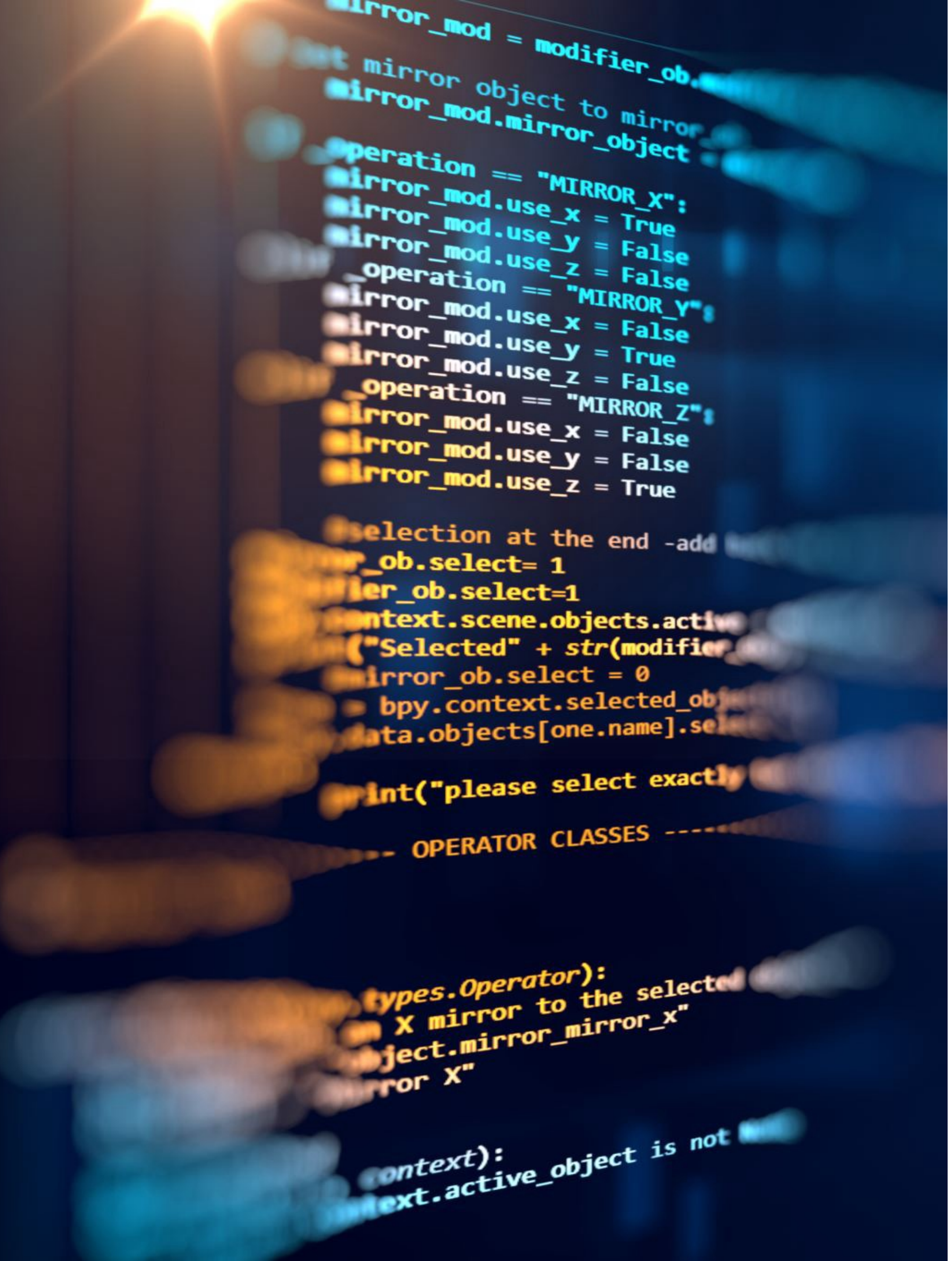
Building a Data Pipeline with dbt

A Comprehensive Guide for Aspiring Data Engineers

Presented by: Tajudeen Abdulazeez

What is Data Pipeline?

- **Definition:** A data pipeline is a series of processes that move data from a source to a destination, transforming it along the way.
- **Importance:** Critical for data analytics, machine learning, and business intelligence.
- **Components:**
  - Data Ingestion
  - Data Storage
  - Data Transformation
  - Data Load



What is dbt?

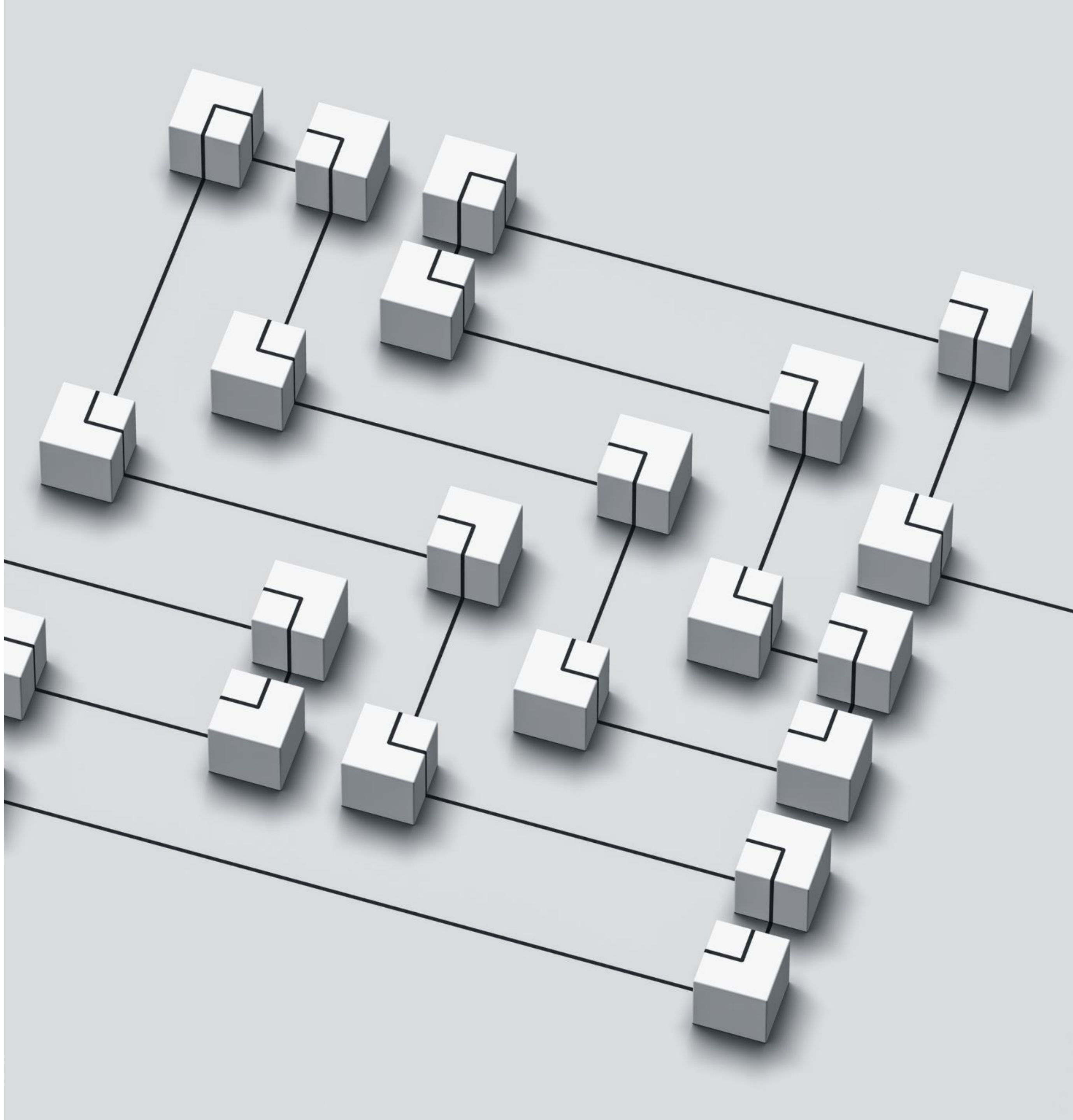
- **Full Name:** Data Build Tool
- **Purpose:** dbt is a command-line tool that enables data analysts and engineers to transform data in their warehouse more effectively.
- **Features:**
  - SQL-based transformation
  - Version control
  - Modular approach
  - Automated documentation
  - Testing capabilities

The dbt Workflow

- **Development:** Write SQL transformations and tests.
- **Testing:** Validate transformations with tests.
- **Documentation:** Automatically generate documentation.
- **Deployment:** Run transformations in production.

Setting Up dbt

- **Installation:** Install dbt using pip.
- **Initialization:** Initialize a new dbt project.
- **Configuration:** Configure profiles and database connections.
- **Folder Structure:** Overview of the dbt folder structure.



Creating Models in dbt

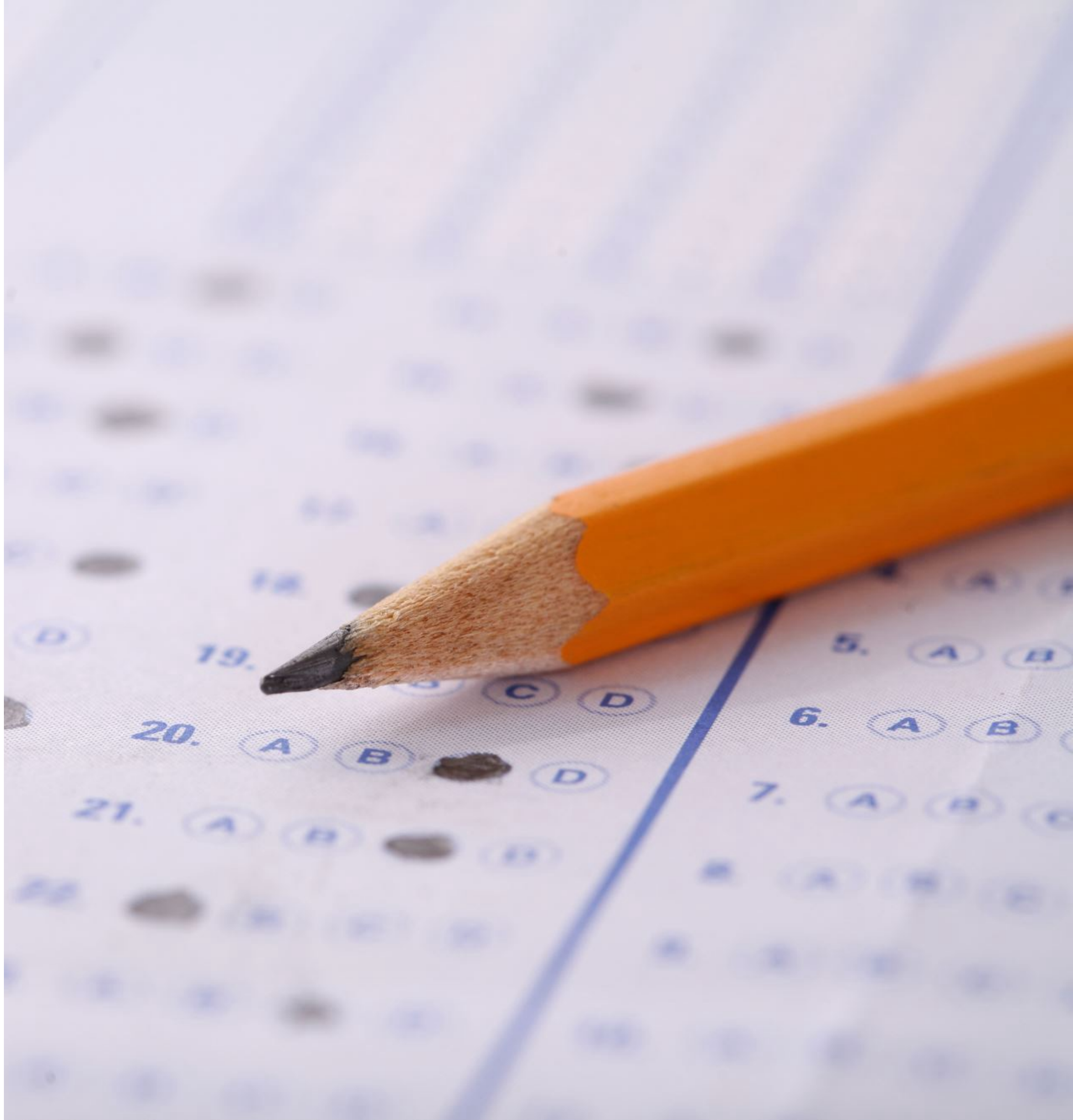
SQL Files: Write SQL files for transformations.

Model Types: Staging, intermediate, and final models.

Materializations: View, table, ephemeral.

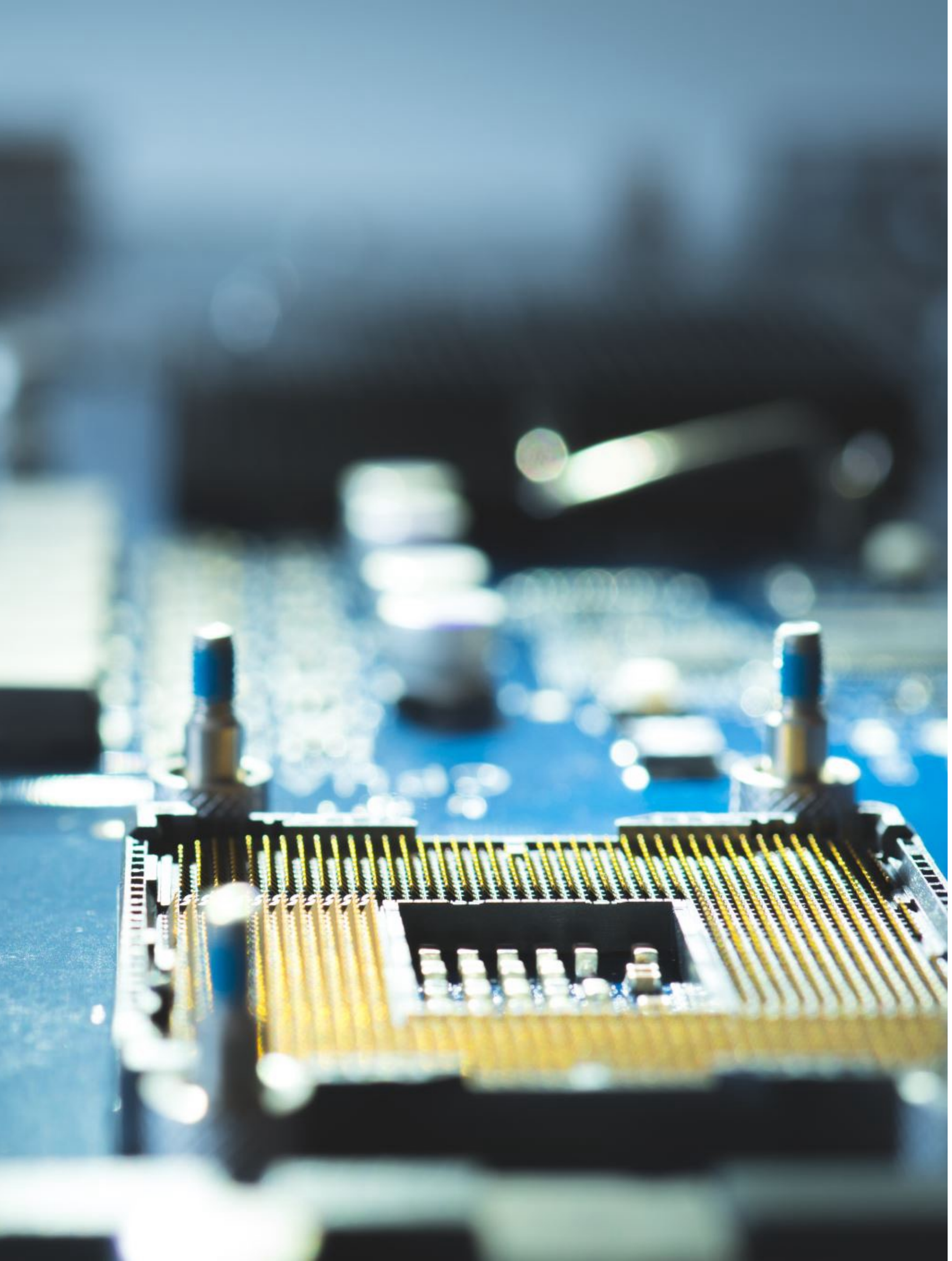
Writing Tests in dbt

- **Built-in Tests:** Unique, not null, accepted values.
- **Custom Tests:** Create your own tests using SQL.
- **Running Tests:** Use the 'dbt test' command.



Generating Documentation with dbt

- **Auto-Generation:** dbt automatically generates documentation for your models.
- **Data Lineage:** Visualize how data flows through your pipeline.
- **Serving Docs:** Use the 'dbt docs serve' command to serve documentation locally.

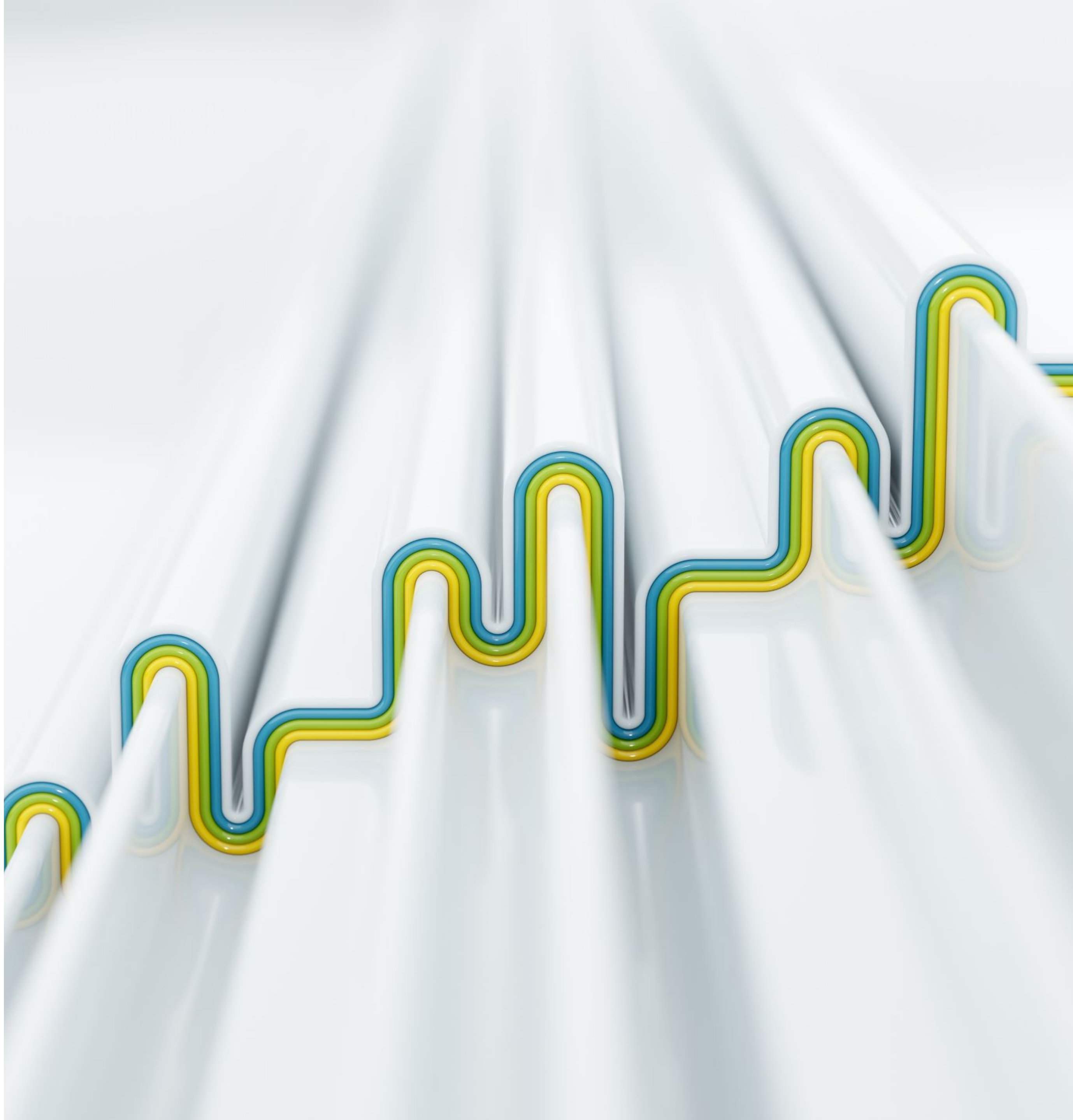


Running and Scheduling dbt Jobs

- **Commands:** 'dbt run' to execute transformations.
- **Schedulers:** Use Airflow, dbt Cloud, or other schedulers.
- **Automation:** Automate your data pipeline with scheduled jobs.

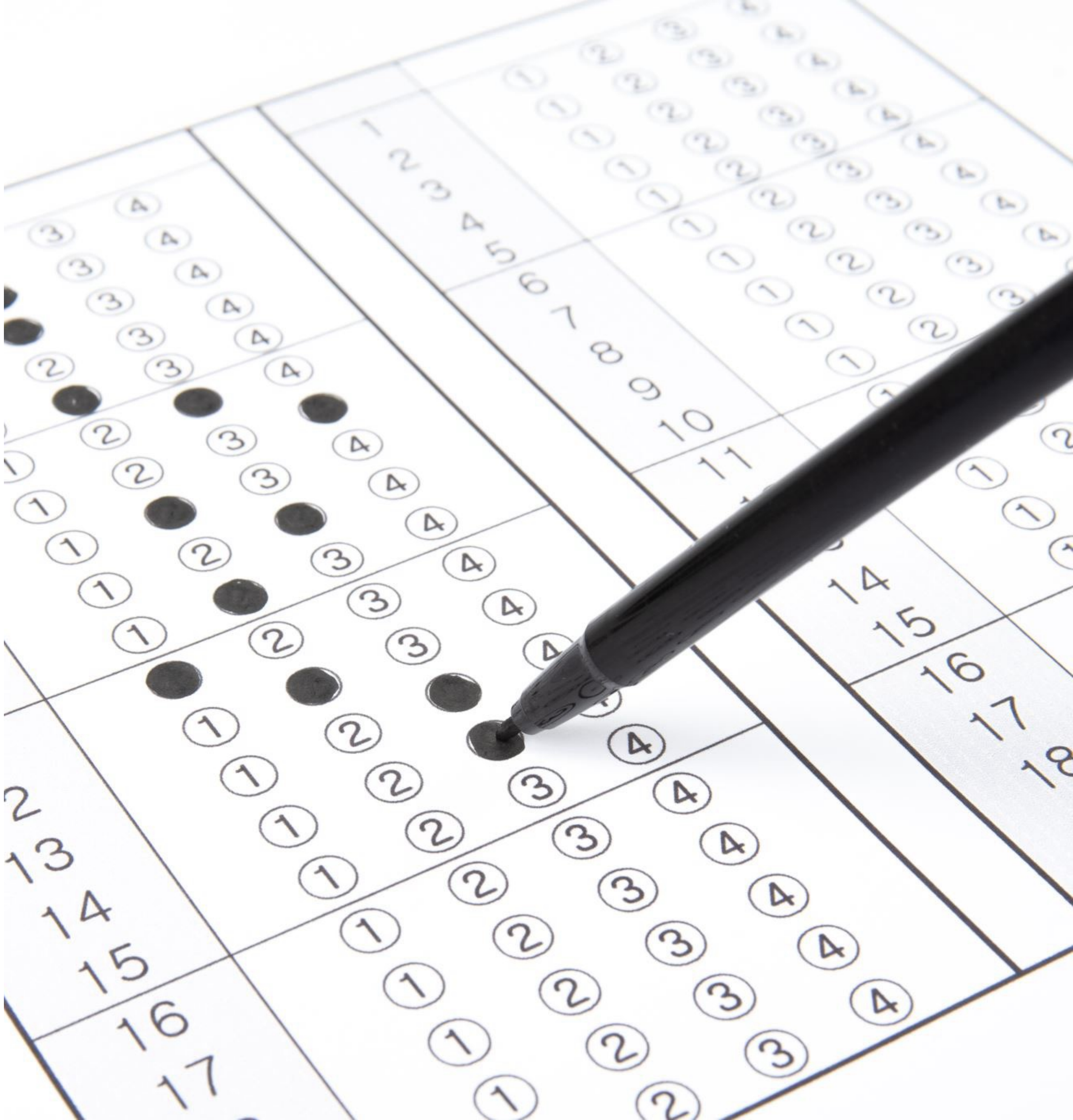
Case Study: Real-World Application of dbt

- **Scenario:** Suppose a retail company wants to analyze sales data from multiple sources
- **Solution:** Using dbt, they can build a data pipeline that ingests data from various systems, transforms it, and loads it into a data warehouse.
- **Outcome:** The outcome is a centralized, reliable data source that provides valuable insights, improving decision-making and business performance.



Best Practices for Using dbt

- **Modular Development:** Keep transformations modular.
- **Version Control:** Use Git for version control.
- **Testing:** Regularly test your transformations.
- **Documentation:** Keep your documentation up to date.



Conclusion

In conclusion, dbt is a powerful tool for building data pipelines, offering features like SQL-based transformations, testing, and automated documentation. By following best practices, you can create robust, scalable data pipelines that drive valuable insights. Now, I'd like to open the floor for any questions you might have

Thank You

Contact Information:

- Name: Tajudeen Abdulazeez
- Linkedin: https://www.linkedin.com/in/tajudeenolarewajuabdulazeez/