databricks

Databricks 101

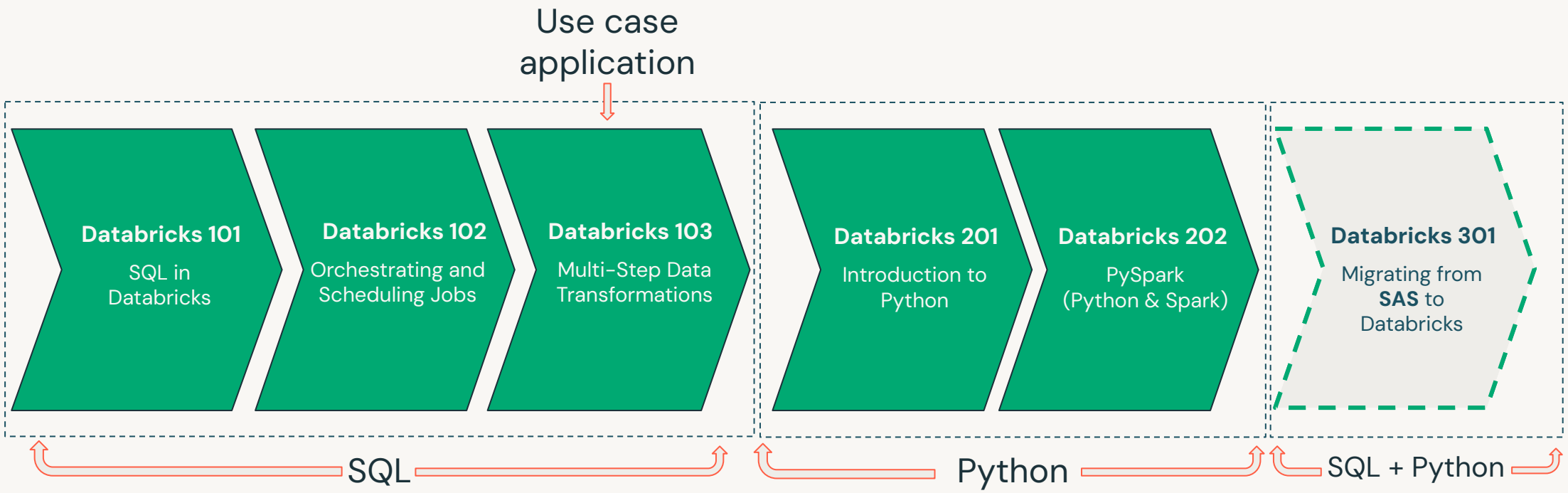
SQL in Databricks

Dan Davis - Databricks

Agenda

- SQL in Databricks Intro
- Exploring Databricks
- Using the SQL Editor
- Q&A / Exploration
- Using SQL in Notebooks
- Q&A / Exploration
- Next Steps

SAS Enablement Pathway



Databricks Resources

Start here

Start here today: Analyst Enablement Workspace

Start here for resources: D&A Reference Hub

Other Resources

- D&A Databricks Development Team Resources
- **Community**
  - Quick questions and community resources- Databricks Viva Engage page
  - Weekly Q&A with Databricks SMEs - DatabricksUserForum@KP
  - Databricks Community
  - FAQ
- **Training**
  - Recommended New AE Training - Training Link
  - Self-Directed Learning

Benefits of Databricks

Benefits of Databricks

Data Sharing and Discovery

- Improve accessibility with a unified, cloud-based data environment.
- Easy data availability (e.g. Foundation Data - Clarity)

Scalability & Performance

- Easily scalable cloud compute power and storage
- Fast, big data processing power
- No server overhead or resource management

Modern Development Environment

- Leverage open-source technology and non-proprietary languages (e.g. python , spark, etc.)
  - Can Install python packages
- Single workspace for all personas

Advanced Analytics & Machine Learning

- Built in support for all types of workloads (SQL, data engineering, machine learning, and AI)
- Ready for GenAI and LLM use cases

Databricks Environments and Workspaces

Databricks Workspaces

Databricks Environments and Workspaces

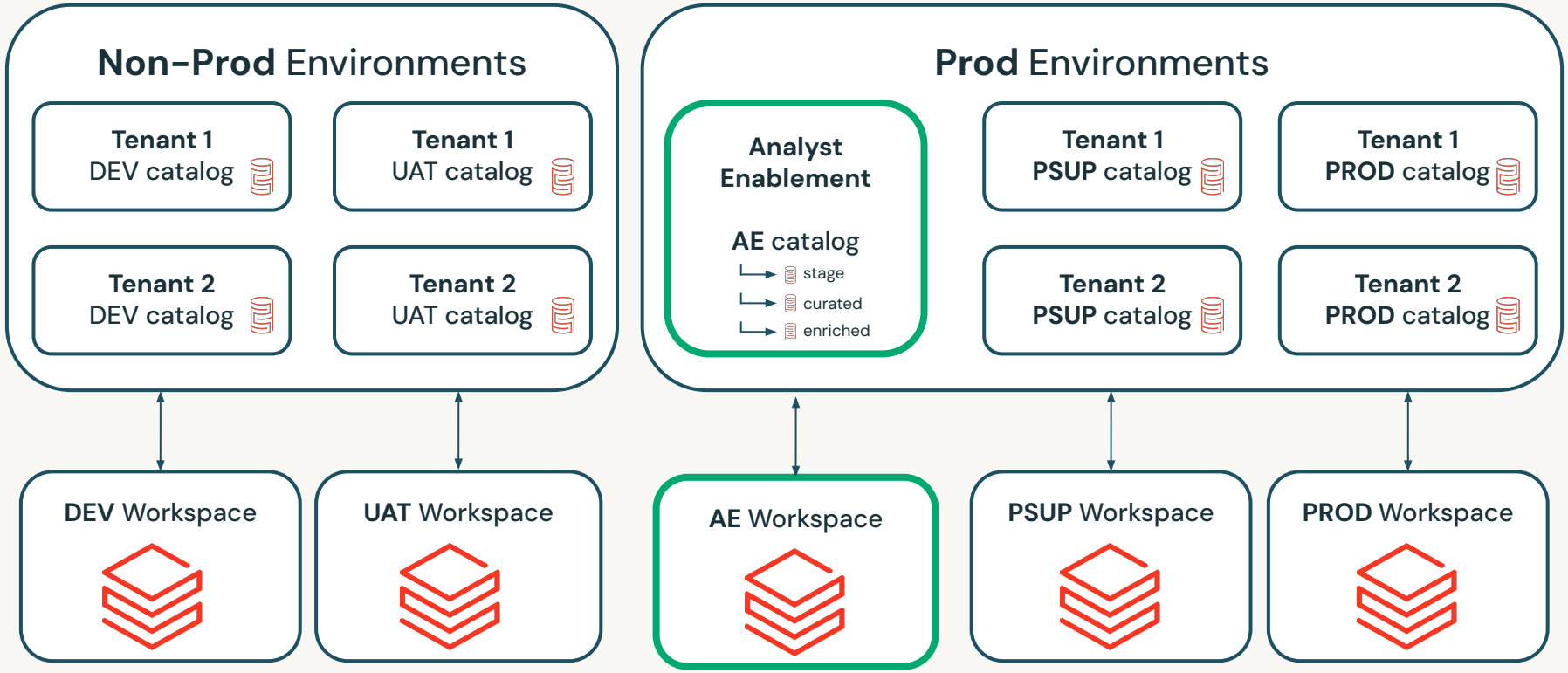
For **most users** in AE, you will only login to the AE workspace

- **AE** has a **single environment**, which is production.

But…

Some AE users also belong to a **tenancy.**

- **Tenancies** have **multiple environments**.
- Environments are separated by different Databricks Workspaces that users log in to.
  - DEV
  - UAT
  - PSUP
  - PROD



SQL in Databricks

PROCS SQL vs. Spark SQL (SQL in Databricks)

SAS **`PROC SQL`** is *largely* ANSI SQL-compliant but **not fully** ANSI SQL. Here's a breakdown:

✅ What aligns with ANSI SQL:

- Basic SQL syntax (e.g., **`SELECT`**, **`FROM`**, **`WHERE`**, **`GROUP BY`**, **`ORDER BY`**)
- Inner and outer joins
- Subqueries
- Math and string functions (**`FLOOR, ROUND, LENGTH, SUBSTR)`**
- Aggregate functions (**`SUM`**, **`AVG`**, etc.)
- **`CASE`** expressions

⚠️ Where it differs:

- **Data types**: SAS uses its own data types (e.g., no **`VARCHAR`**, **`INTEGER`**; everything is either numeric or character).
- **Functions**: Many functions (e.g., string or date functions) are SAS-specific and don't follow ANSI standards.
- **Implicit behavior**: SAS allows non-standard behaviors like:
  - Using **`GROUP BY`** without `SELECT` including all group-by columns
  - Implicit casting and tolerance for type mismatches
- **Extensions**: `PROC SQL` supports SAS-specific extensions like `INTO :macrovar` and the ability to create or manipulate SAS datasets directly.

Write SQL using the **SQL Editor**

What is it?

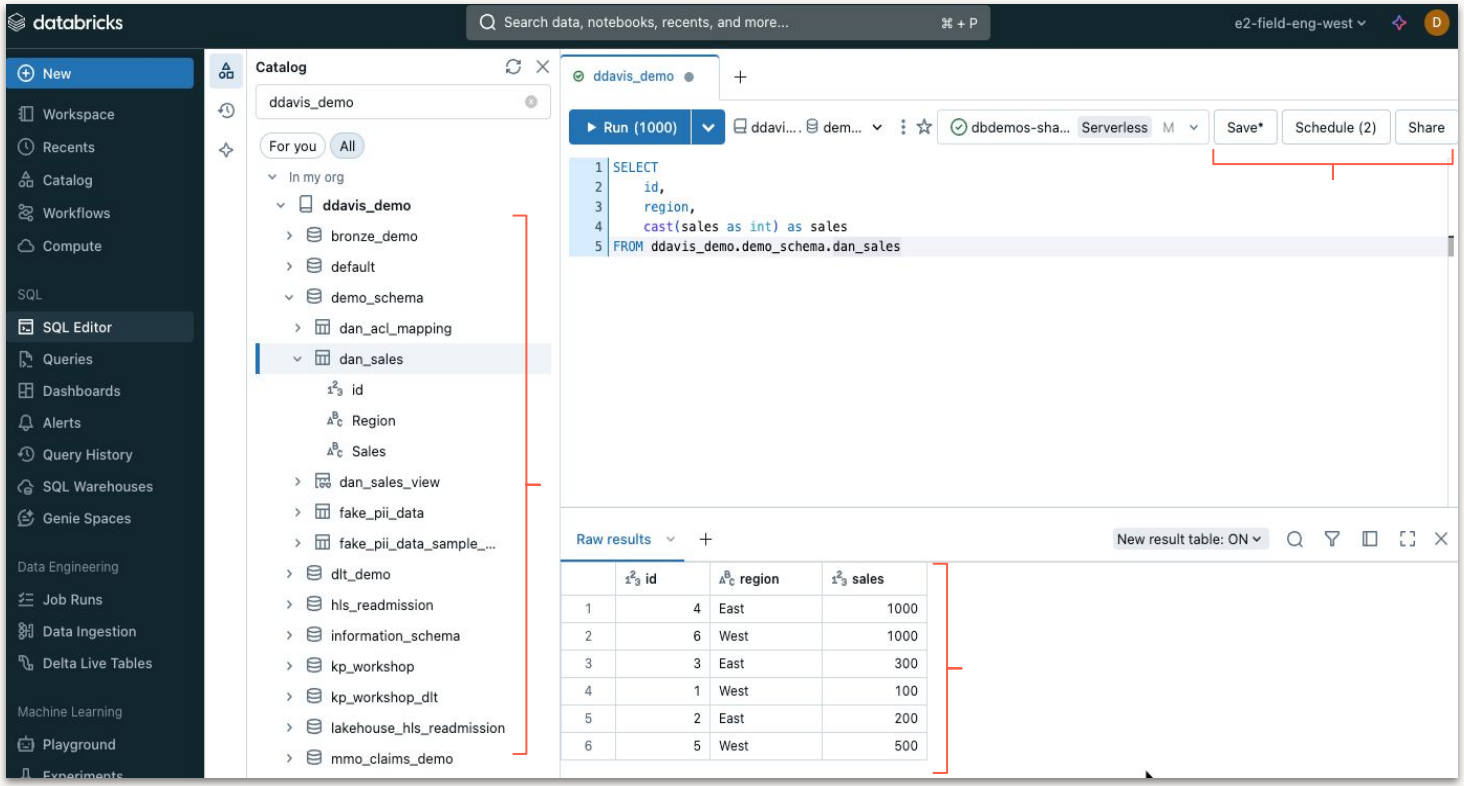
- Tool to **author**, **save** and **share** queries
- **Browse** available data
- **Create** visualizations

Purpose

- Exploratory analysis
- Ad-hoc queries

Similar to

- Oracle SQL Developer
- SQL Server Management Studio



Write SQL using a **notebook**

What is it?

- Primary tool to **author**, **save** and **share** programs
- **Develop** and **debug** code
- **Browse** available data
- **Create** visualizations and **profile** data
- **Write** in multiple languages (SQL, python, r, etc.)
- **Install** libraries
- **Run programs** interactively and **schedule** regular jobs
- **Add** text (markdown) and documentation

Purpose

- Construct a program with multiple steps
- Use for a sequence of code that performs multiple tasks
- Exploratory Analysis

Similar to

- SAS Program
  - Contains a series of DATA Steps or PROC Steps