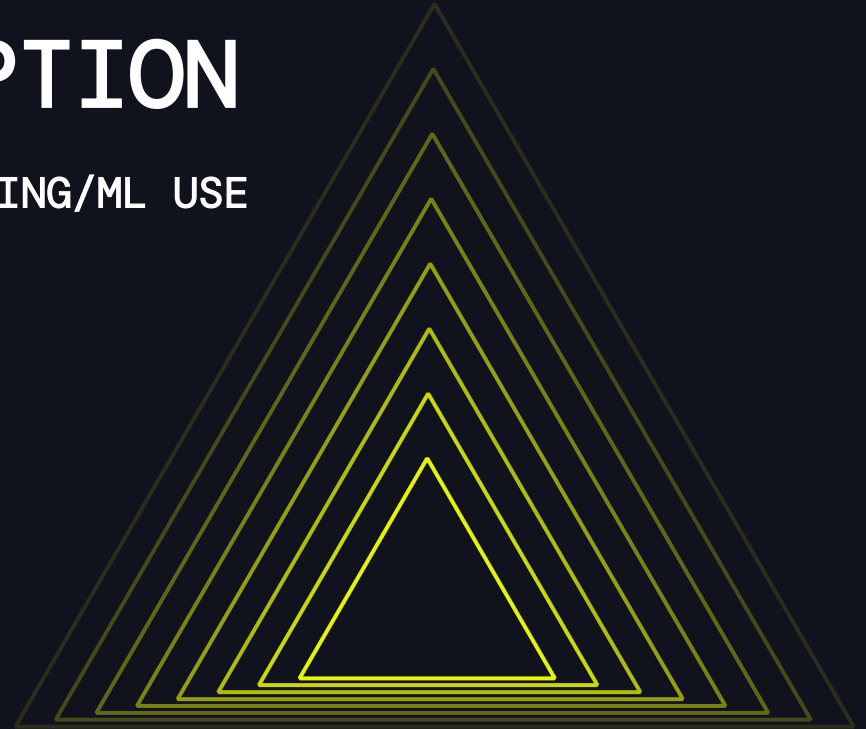


ACCELERATING ADOPTION OF DATALAKE FOR STREAMING/ML USE CASES



June 2024

DATA+AI SUMMIT

SESSION SPEAKERS



Harsha Reddy

/ Engineering Manager
Doordash



Aydar Akhmetzyanov

/ Software Engineer
Doordash

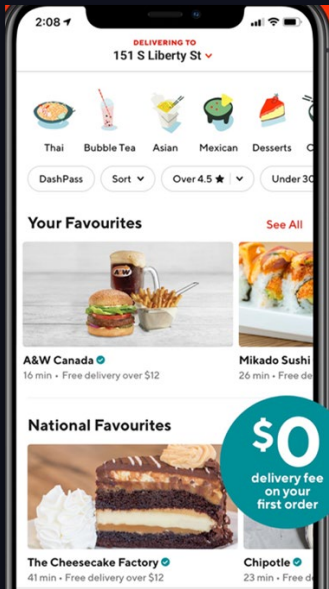
Mission

*Our mission is to grow and empower
local economies*

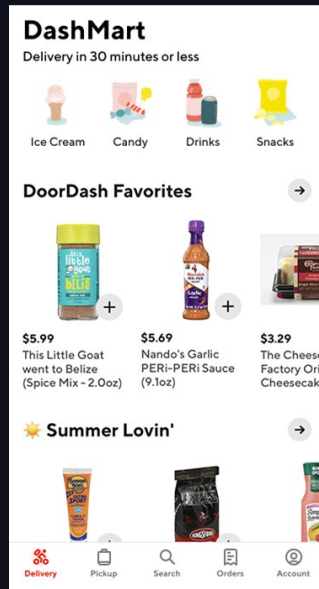


Doordash Marketplace

Restaurant Delivery & Pickup



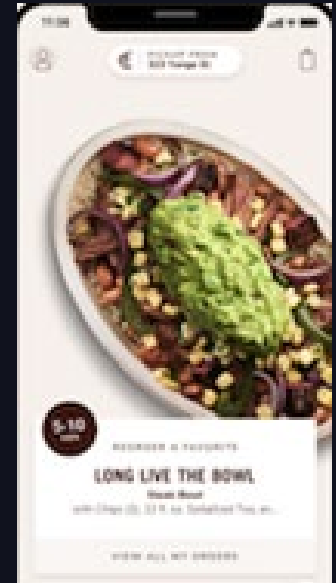
Convenience & Grocery



DashPass Membership

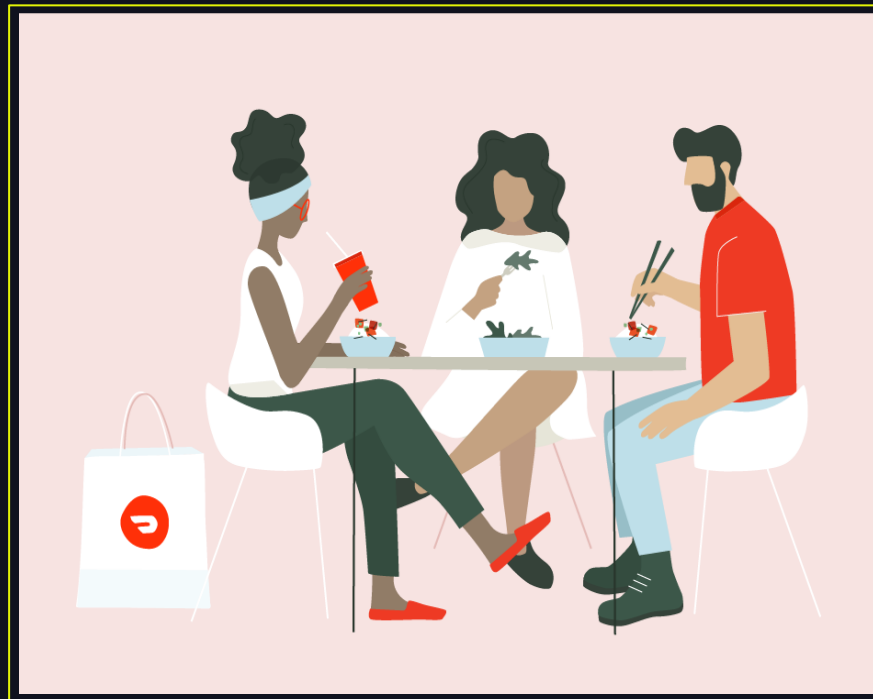


B2B Fulfillment

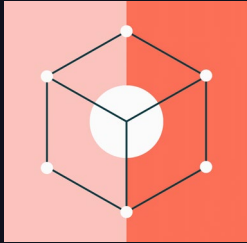


Agenda

- Data warehouse to Datalake Journey
- Challenges with Adoption
- Accelerator strategy
- Accelerator Tools Deep-dive
- AI Accelerator Deep-dive
- Conclusion

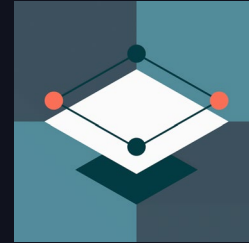


Journey Data Warehouse to Data Lake



Data Warehouse

Common Industry trend is to scale data based decision making while reducing costs



Data Lake

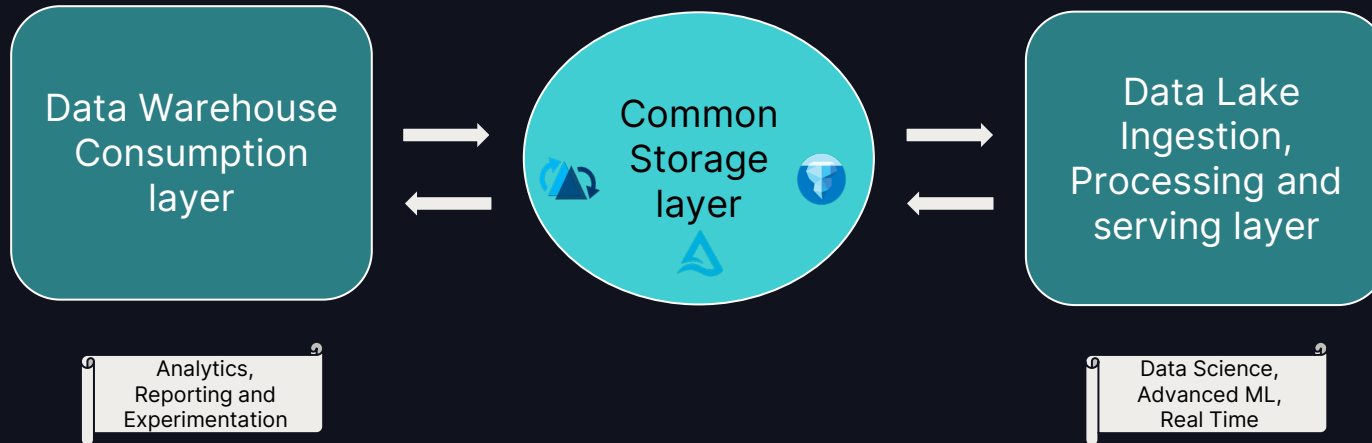
- Comparatively very easy to use
- Primarily used in Analytics/Reporting/Experimentation
- Fully Managed solution end to end
- Great interactive query performance
- Exposes various underlying technologies
- Evolution started with ML/DS and Real-time usecases
- Decoupled Storage and Compute
- Great data processing capabilities on large amounts of data

Journey Current state

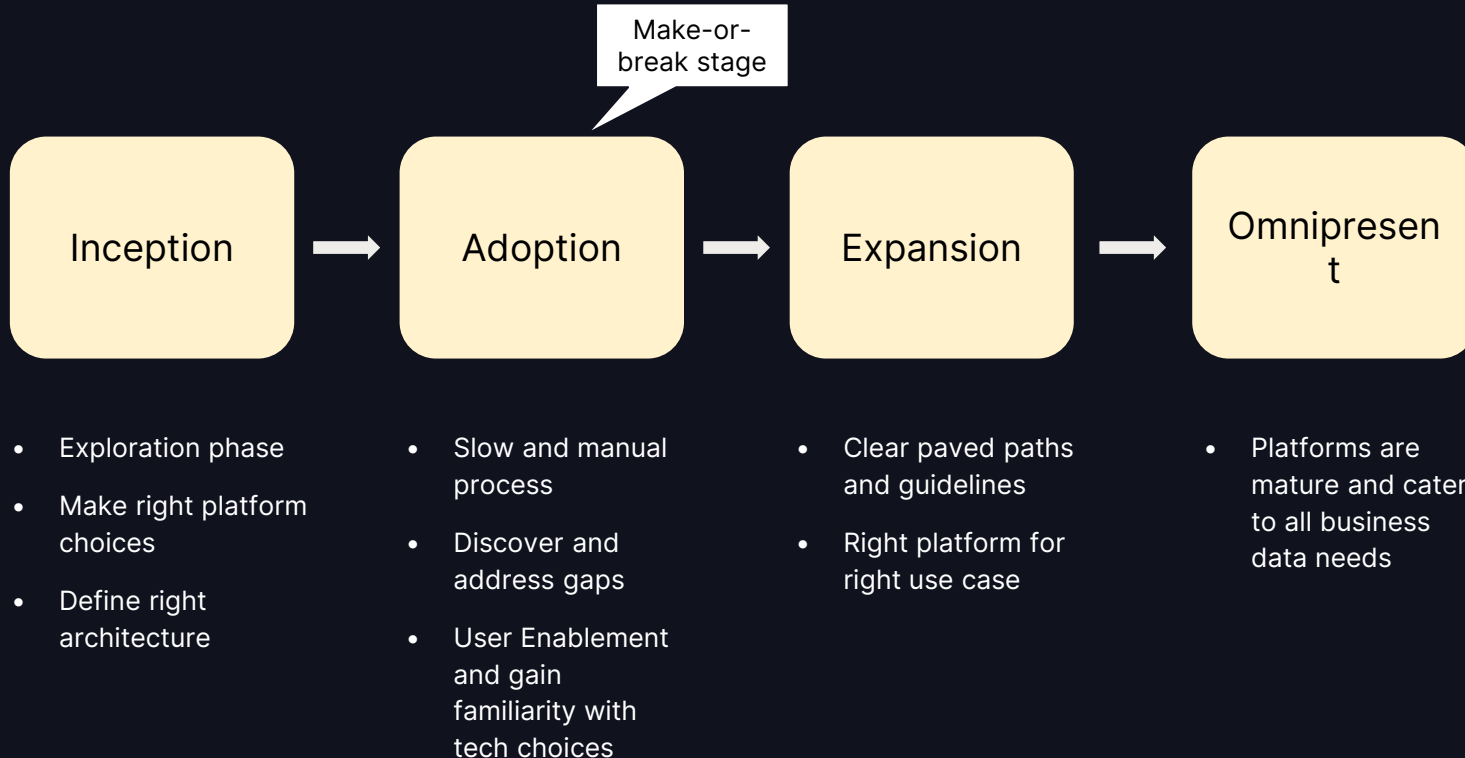


Journey next stop-Data Lakehouse

Not Either-Or, But Both-And.



Stages of DataLake Evolution



Challenges in Adoption Phase

*Change is hard at first, **messy in the middle** and gorgeous at the end*



- Data readiness for real time event streams and ML/DS data
- Time consuming and resource intensive
- Leads to ETLs, ML Models and Storage migrations in some cases
- Downstream impact scope creep

Accelerator Strategy

Do More with Less

Goal: Reduce Time to adopt for new usecases or migrate relevant usecases by 3X

- Automation of manual tasks through series of tools
- AI assistance in Acceleration
- Data readiness tools
- Self-serve



Role of AI in Accelerator Strategy

Inhouse AskDataAI Platform capabilities



ML/Data Model Changes

- Not smart Search and replace
- Understand the semantics and auto apply model level changes
- Auto generate GIT PRs with changes



Data Discovery and Exploration

- Make the existing data catalog available to conversational AI agent
- Explore data insights and trends using results sourced from the right platform



SQL Co-pilot

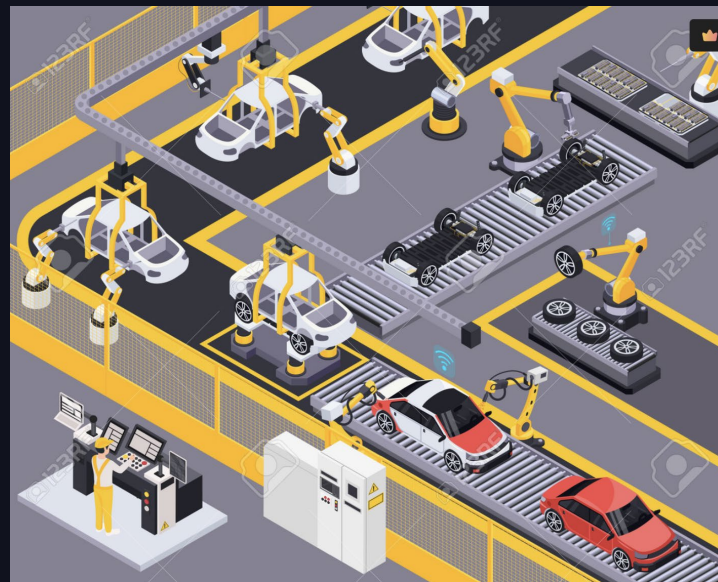
- Generate SQL in the appropriate SQL dialect
- Auto-optimize the SQL patterns
- Queries are generated only on blessed and certified datasets

Accelerator Strategy-Tools

Migration of Existing use cases (if applicable):

Similar to Car manufacturing , Series of automations on an assembly line working together to migrate from one platform to another.

Adoption for New use cases : Series of automations that can help data readiness for upstream dependencies and translation + impact assessment for downstream artifacts like ML models, data models, reports, etc



Accelerators

- Inventory Tool: Migration Diagnostics tool
- Transaxle: SQL Translator -Supports multiple dialects like Trino, Spark SQL and Snowflake SQL
- Assembly Tool: Airflow Dag generator
- Inspection Tool: Data Validation
- AskDataAI: AI assistance in Acceleration

DATALAKE ACCELERATION TOOLS DEEPLIVE



INVENTORY TOOL

Pipeline analysis at scale for Datalake adoption problem



Effort related to pipeline analysis

- Manual SQL Review
- Dependency analysis for each DAG/task/table
- Downstream impact analysis
- Datalake table search
- Parity inspection

INVENTORY TOOL

Assistance UI and automated pipeline analysis

- Dependency and downstream parsing from SQL logs
- Downstream impact analysis
- SQLdialect translation
- Real-time validation / Inspection
- DWH/DL mirrors search
- Code generation

DAG: feedonomics_v6_daily

Legend: snowflake table (click for downstream analysis and curator job assembly) related datalake table

🔍 ☰ ⌵ ⬇️

task_id	task_type	manifest	sqls	serialized_code
proddb.public.fact_feedonomics_v6_zone_mapping_zips_optimization	DdRunetOperator	<p>Downstreams:</p> <ul style="list-style-type: none">proddb.public.fact_feedonomics_v6_zone_mapping_zips_optimization Create Curator job <p>Upstreams:</p> <ul style="list-style-type: none">proddb.public.fact_feedonomics_v6_zone_mapping_zips_optimization Create Curator jobgeo_intelligence.public.fact_feedonomics_v6_zone_mapping_zips_optimization (curator) Tables are healthyproddb.public.fact_feedonomics_v6_zone_mapping_zips_optimization Create Curator jobdoordash_merchant.public.fact_feedonomics_v6_zone_mapping_zips_optimization (maindb) doordash_merchant.fact_feedonomics_v6_zone_mapping_zips_optimization (curator) Tables are healthyedw_merchant.di.fact_feedonomics_v6_zone_mapping_zips_optimization (curator) Run inspectionedw.finance.fact_feedonomics_v6_zone_mapping_zips_optimization Create Curator job	Show sqls	Show serialized_code

INVENTORY TOOL

Real-time health inspection capabilities

- Low-latency querying across different platforms using Trino
- Schema comparison
- Data volume comparison (row count)

DAG: feedonomics_v6_daily

Legend: snowflake table (click for downstream analysis and curator job assembly) related datalake table

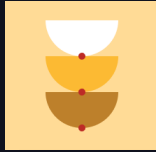
⌵ ⌵ ⌵

task_id	task_type	manifest	sqls	serialized_code
proddb.public.fact_feedonomics_v6_zone_mapping_zips_optimization	DdRunetOperator	<p>Downstreams:</p> <ul style="list-style-type: none">proddb.public.fact_feedonomics_v6_zone_mapping_zips_optimization Create Curator job <p>Upstreams:</p> <ul style="list-style-type: none">proddb.public.fact_feedonomics_v6_zone_mapping_zips_optimization Create Curator jobgeo_intelligence.public.fact_feedonomics_v6_zone_mapping_zips_optimization (curator) Tables are healthyproddb.public.fact_feedonomics_v6_zone_mapping_zips_optimization Create Curator jobdoordash_merchant.public.fact_feedonomics_v6_zone_mapping_zips_optimization (maindb) doordash_merchant.fact_feedonomics_v6_zone_mapping_zips_optimization (curator) Tables are healthyedw.merchant.dl.fact_feedonomics_v6_zone_mapping_zips_optimization (curator) Run inspectionedw.finance.fact_feedonomics_v6_zone_mapping_zips_optimization Create Curator job	Show sqls	Show serialized_code



INVENTORY TOOL

Implementation details- Data sources



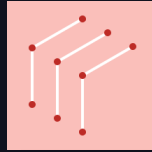
Airflow DB

- Dag list
- Task list
- Task operators
- Log links



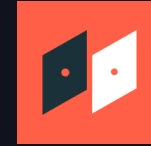
Source code

- Operator arguments
- Source SQL templates



Task logs

- Executed SQLs



DL mirror service

- Datalake table mapping
- Table origin
- Additional metadata

INVENTORY TOOL

Implementation details - Evolving airflow operators

PYTHON

```
db_campaign_test_orders = DdDWHTargetLoadOperatorV2(  
    dag=dag,  
    task_id="xx.db.campaign_test_orders",  
    table_fq_name="foobar.campaign_test_orders",  
    column_definitions=COLUMN_DEFINITIONS,  
    table_type=TableType.TRANSACTIONAL.name,  
    query=SQL_QUERIES["campaign_test_orders"],  
    load_type="INSERT",  
    pre_delete_sql="DELETE FROM {TargetDatabase}.{TargetSchema}.campaign_test_orders",  
    sla=datetime.timedelta(minutes=10),  
)
```

There are 50 different operators that generate SQLs

INVENTORY TOOL

Solution - Airflow logs parsing with regex

Potentially CRF ML Model for sequence labeling can be trained

Airflow log

```
[2023-12-04T23:20:50.977+0000] {subprocess.py:93} INFO - 2023-12-04 23:20:50,977,977 INFO [snow.py:142] Query time(s): 7.61
[2023-12-04T23:20:50.977+0000] {subprocess.py:93} INFO - 2023-12-04 23:20:50,977,977 INFO [s3_raw_base.py:391] Created compacted clone of "R
[2023-12-04T23:20:50.978+0000] {subprocess.py:93} INFO - 2023-12-04 23:20:50,977,977 INFO [snow.py:114] Executing SQL: MERGE INTO "INCOME_P
[2023-12-04T23:20:50.978+0000] {subprocess.py:93} INFO - (SELECT "ROW_ID","DATA",
[2023-12-04T23:20:50.978+0000] {subprocess.py:93} INFO - WHEN MATCHED AND t."UPDATED_AT" > f."UPDATED_AT" THEN
[2023-12-04T23:20:50.978+0000] {subprocess.py:93} INFO - UPDATE SET f."ROW_ID" = t."ROW_ID", f."DATA" = t."DATA", f."CHANGE_REF
[2023-12-04T23:20:50.978+0000] {subprocess.py:93} INFO - WHEN NOT MATCHED THEN
[2023-12-04T23:20:50.978+0000] {subprocess.py:93} INFO - INSERT ("ROW_ID","DATA","CHANGE_REF"
[2023-12-04T23:20:50.978+0000] {subprocess.py:93} INFO - 2023-12-04 23:20:50,977,977 INFO [cursor.py:738] query: [MERGE INTO "INCOME_PLATFOR
[2023-12-04T23:20:54.961+0000] {subprocess.py:93} INFO - 2023-12-04 23:20:54,961,961 INFO [cursor.py:751] query execution done
```

INVENTORY TOOL



SQL Decomposition into downstream and dependencies with open source SQLGlott lib

PYTHON

```
from sqlglot import parse_one, exp

ast = parse_one(sql, read='source system')
tables = ast.find_all(exp.Table)
tables = filter(lambda x: x.db, tables)
tables = list(map(lambda x: f"{x.catalog}.{x.db}.{x.name}".lower(), tables))
downstream = ''
if ast.key != 'select' and ast.key != 'union':
    if len(tables) > 0:
        if ast.key == 'altertable' and len(tables) >= 2 and 'rename ' in sql.lower():
            downstream = tables.pop(1)
        else:
            downstream = tables.pop(0)
```

TRANSAXLE - SQL TRANSLATION TOOL

SQL migration problem



End-to-end migration to the DataLake requires making SQL translations. This can take a lot of time and manual effort.

TRANSAXLE - SQL TRANSLATION TOOL

SQL migration solution

Central hub to serve SQL translation needs

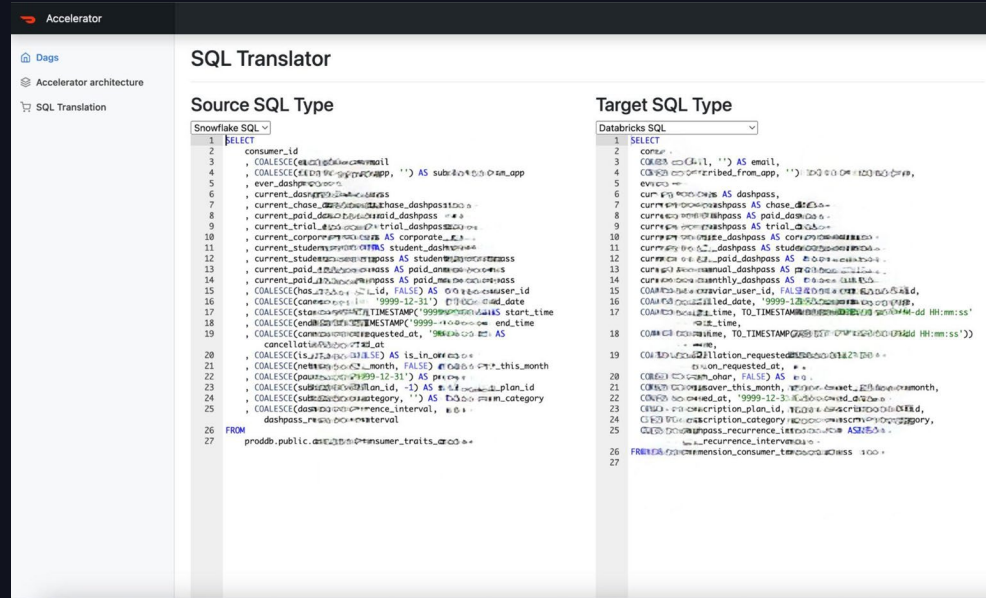
Reduce query translation and syntax validation time from days to minutes

Provides vendor agnostic SQL

Table name mapping

Query validation with Live Spark Cluster

Integrates with code generation tools

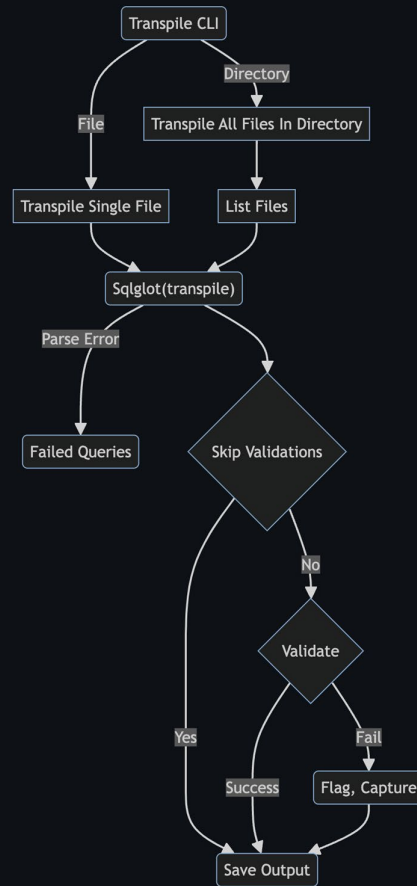


TRANSAXLE - SQL TRANSLATION TOOL

Exploits Databricks labs Remorph library



SQL code converter and data reconciliation tool for accelerating data onboarding to Databricks from EDW, CDW and other ETL sources.



ASSEMBLY TOOL – DAG GENERATION

Code generation problem



Migrating a DAG code to Datalake version involves multiple steps that takes few days from code analysis to testing.

ASSEMBLY TOOL – DAG GENERATION

Code generation solution

Generation steps:

1. SQL extraction and translation
2. Operators replacement with parity
3. Pull-request assembly
4. Integration and unit testing
5. Migration report generation

Assembly tool automates the majority of steps and in some cases is capable to generate the final end-to-end solution.

AI Accelerator - Problem Statement

Agent deployment problem for adoption and data exploration



- Data exploration across different platforms
- Semantic search
- Data interpretation
- SQL Query generation
- Accessing complex DWH structures like metric cubes
- Searching internal documentation in google drive/confluence

AskDataAI - Solution

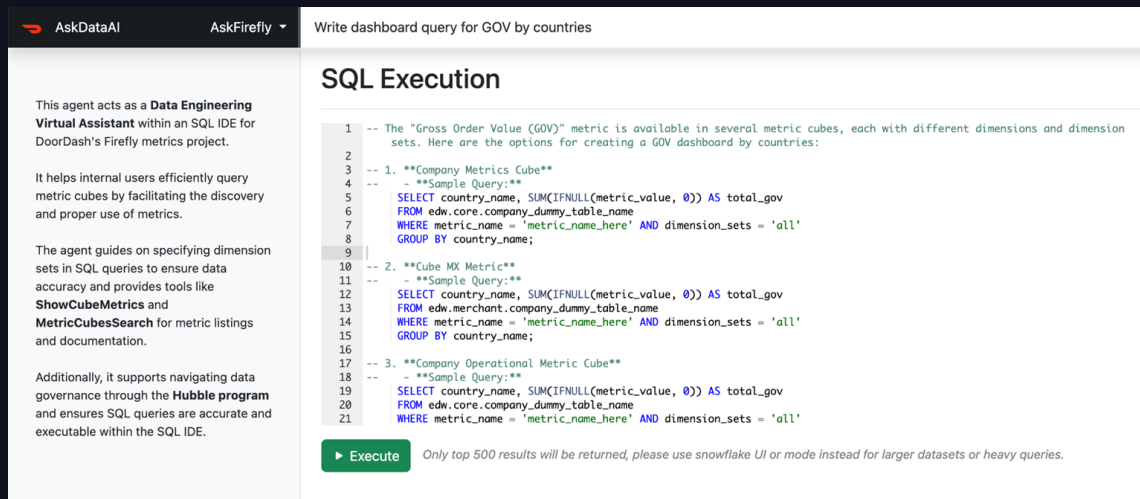
AskDataAI as the internal “GPTs” platform for data exploration solution

AskDataAI Platform:

- User interface
- VectorDB and semantic search engine
- Integration with slack and other communication channels
- API endpoint
- Loader/Worker templates (google drive, etc.)

Client AskData-X Apps:

- Custom prompt
- VectorDB collections
- Data loaders
- Functions/tools
- API endpoints for internal services



The screenshot displays the AskDataAI interface. The top navigation bar includes 'AskDataAI' and 'AskFirefly'. The main content area is titled 'Write dashboard query for GOV by countries' and is divided into two panels. The left panel contains instructional text: 'This agent acts as a **Data Engineering Virtual Assistant** within an SQL IDE for DoorDash's Firefly metrics project. It helps internal users efficiently query metric cubes by facilitating the discovery and proper use of metrics. The agent guides on specifying dimension sets in SQL queries to ensure data accuracy and provides tools like **ShowCubeMetrics** and **MetricCubesSearch** for metric listings and documentation. Additionally, it supports navigating data governance through the **Hubble program** and ensures SQL queries are accurate and executable within the SQL IDE.' The right panel, titled 'SQL Execution', shows a SQL query with three numbered sections: 1. 'Company Metrics Cube', 2. 'Cube MX Metric', and 3. 'Company Operational Metric Cube'. Each section includes a sample query using a SUM function on 'metric_value' grouped by 'country_name'. Below the code is an 'Execute' button and a note: 'Only top 500 results will be returned, please use snowflake UI or mode instead for larger datasets or heavy queries.'

AskDataAI - Solution

Data exploration with LLM, VectorDB, and AI Agents

Do we have any metrics to track protective equipment masks?

AI Agent actions:

1. MetricsSearch('protective equipment')
2. TablesSearch('protective equipment')
3. GetTableDescription('edw.core.finance_metrics')
4. FinalResponse: ('{SQL}')

Write dashboard query for GOV by countries

SQL Execution

```
1 -- The "Gross Order Value (GOV)" metric is available in several metric cubes, each with different dimensions and
2   dimension sets. Here are the options for creating a GOV dashboard by countries:
3 -- 1. **Company Metrics Cube**
4 --   - **Sample Query:**
5   SELECT country_name, SUM(IFNULL(metric_value, 0)) AS total_gov
6   FROM edw.core.company_dummy_table_name
7   WHERE metric_name = 'metric_name_here' AND dimension_sets = 'all'
8   GROUP BY country_name;
9 |
10 -- 2. **Cube MX Metric**
11 --   - **Sample Query:**
12   SELECT country_name, SUM(IFNULL(metric_value, 0)) AS total_gov
13   FROM edw.merchant.company_dummy_table_name
14   WHERE metric_name = 'metric_name_here' AND dimension_sets = 'all'
15   GROUP BY country_name;
16 |
17 -- 3. **Company Operational Metric Cube**
18 --   - **Sample Query:**
19   SELECT country_name, SUM(IFNULL(metric_value, 0)) AS total_gov
20   FROM edw.core.company_dummy_table_name
21   WHERE metric_name = 'metric_name_here' AND dimension_sets = 'all'
```

▶ Execute

Only top 500 results will be returned, please use snowflake UI or mode instead for larger datasets or heavy queries.

AskDataAI - Architecture

Implementing an agent with LangChain



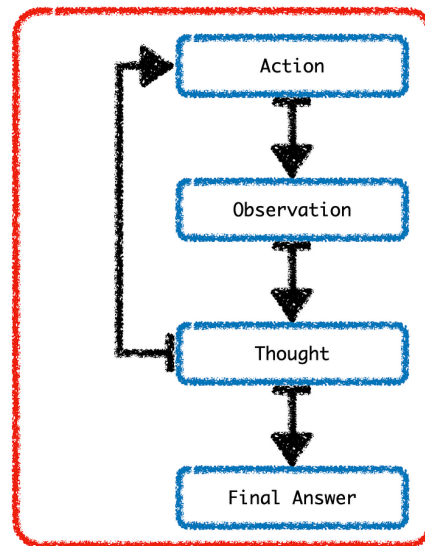
Langchain is used to define tools

LLM decides on what to do next in agentic workflow
(generates thoughts, actions, and the final response)

Tool/function examples:

- GoogleDriveSearch
- ShowCubeMetrics
- DescribeTable
- ShowTablesLikeInHubble
- ShowTablesLike
- CallAPI

LangChain Agent - Sequence Of Events



ASKDATAAI - Data Accuracy

How to ensure reliability with AI-graded tests

- Test-driven development
- Regression testing
- Tools/Semantic search testability

`evaluate(query, expected_answer, actual_answer) -> explanation, score`

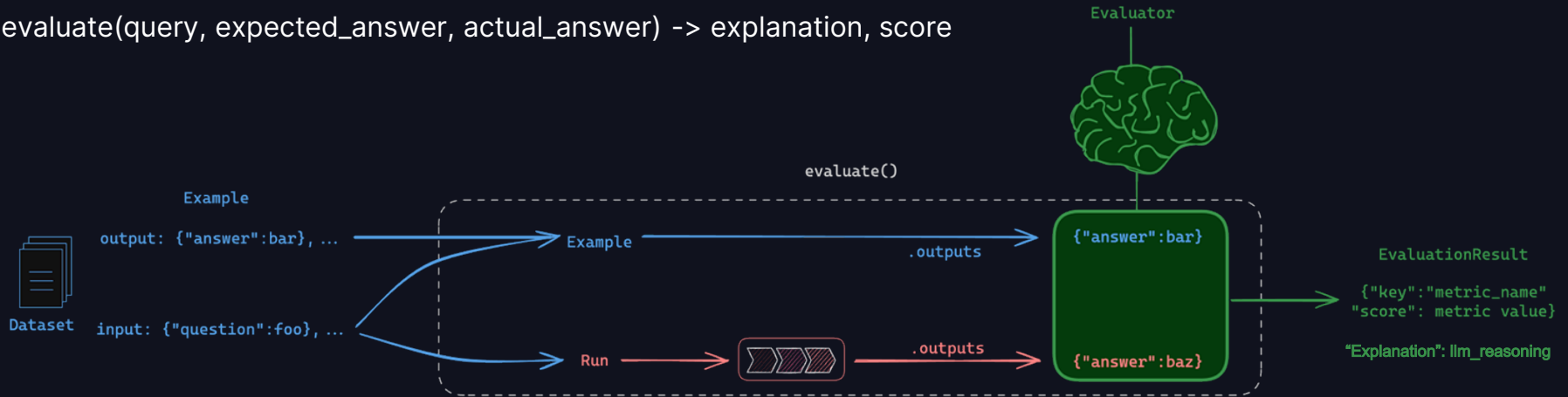


Image source: <https://docs.smith.langchain.com/old/evaluation>

AskDataAI - Semantic Search

Semantic search with FAISS Library and generating documents

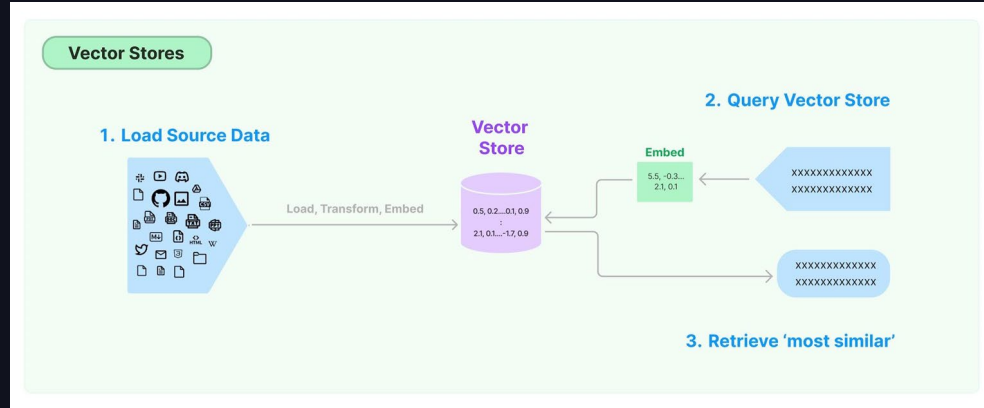
Applying LLM to generate table description based on column names and available documentation

LLM to generate business question and SQL solution using table/metric

Document = metric metadata + LLM description + SQL example + describe table results

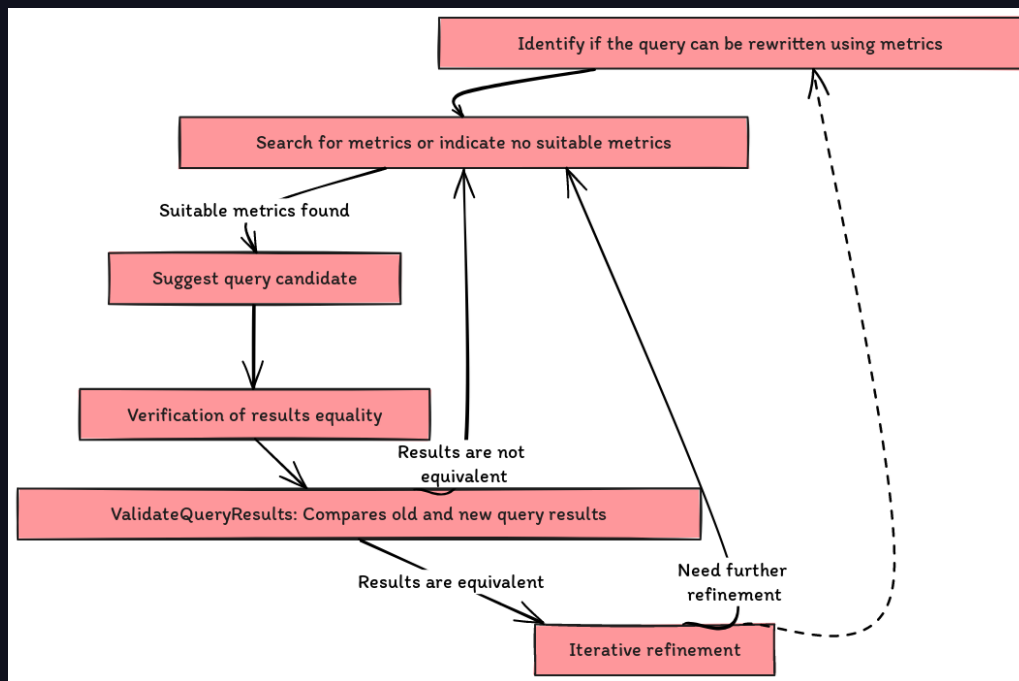
Evaluation using AI generated business questions and cross-validation

Data loaders and workers



ASKDATAAI - Acceleration Example

Schema migration acceleration use case with agentic workflow



Conclusion

- Change in the data space is a constant, Adapt fast.
- Define the right data architecture - There is no one size fits all solution
- Proactively identify adoption/migration bottlenecks very early in the game
- Tools/Frameworks play an important role in technology adoption. Invest in them
- AI based solutions have high potential in data applications beyond mainstream use cases

QUESTIONS?

