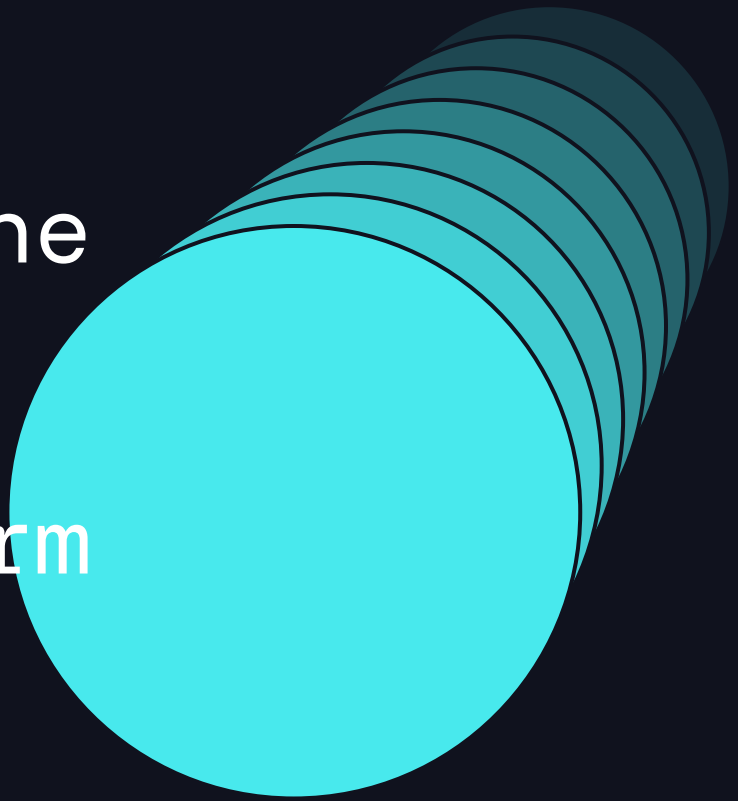
DATA+AI SUMMIT BY databricks

Lessons Learned from Migrating the Largest Immunization Analytics Platform



©2024 Databricks Inc. — All rights reserved

PRESENTERS

Michael Pisarsky – Solution Architect – Mosaic Data Solutions

Rex Phillips – Strategy Senior Principal – Accenture

©2024 Databricks Inc. — All rights reserved

OVERVIEW

- What is an Immunization Registry (IIS)?
- The decision to move to Databricks
- Insights
- The Solution

©2024 Databricks Inc. — All rights reserved

IN THE BEGINNING THERE WAS THE IIS…

And it was good.

©2024 Databricks Inc. — All rights reserved

IMMUNIZATION REGISTRIES / IIS

- The official term is: Immunization Information System (IIS)
- The purpose of an IIS is to:
  - Confidentially and securely store sensitive data
  - Maintain all immunization-related data for individuals
  - Track immunizations administered
  - Assess vaccination coverage
  - Make recommendations for additional vaccinations

©2024 Databricks Inc. — All rights reserved

IMMUNIZATION REGISTRIES / IIS

- The history of IIS's
  - Multiple IIS's used across the state
  - Everything consolidated on one IIS as part of COVID efforts
- The need for the Analytics Platform
  - Its birth
  - Versions 1-3

©2024 Databricks Inc. — All rights reserved

MOVING TO DATABRICKS

©2024 Databricks Inc. — All rights reserved

THE DECISION TO MOVE TO DATABRICKS

Why we wanted to migrate from the previous system to Databricks:

- Better Integration with our Cloud Architecture
- Superior Security Capabilities
- Detailed Monitoring and Tuning of Resources
- Integrated Data Governance
- Streamlined ETL Capabilities
- Ability to Integrate with Other Systems
- Unified Data Platform
- Continuous Innovations
- Advanced Machine Learning and AI Capabilities

©2024 Databricks Inc. — All rights reserved

REQUIREMENTS

The new solution must:

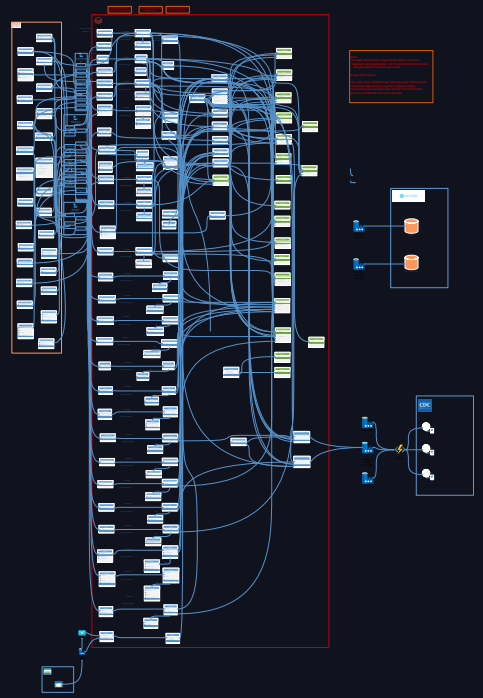
- Have, at most, the same duration as the current version (<45 minutes)
- Cost about the same
- Implement the new Master Person Index
- Have easily maintained code
- Maintain historical data
- Apply new data governance rules
- Automate CI/CD and Unit Testing

©2024 Databricks Inc. — All rights reserved

CODE MAINTENANCE ISSUES

The Old Pipeline was difficult to maintain

- Multiple intermediary steps and levels
- Hundreds of tables and views
- Only one developer knew how it all worked



©2024 Databricks Inc. — All rights reserved

PLANNED SOLUTION

For the solution, we planned on using:

- Azure Data Factory to orchestrate the solution
- Change Data Capture from the on-prem Oracle DB
- Delta Live Tables to ingest the new data
- SCD Type 2 to maintain the history
- Iceberg compatible tables to integrate with other systems

©2024 Databricks Inc. — All rights reserved

LESSONS LEARNED

1. Know your data
2. Understand your data structure
3. Have a cluster strategy
4. Decouple workflows

©2024 Databricks Inc. — All rights reserved

LESSON 1: KNOW YOUR DATA

The Assumption

- Change Data Capture to the Oracle DB was done using MVLogs
  - So far so good
- The processed data was saved to a blob container
  - No issues
- Autoloader picked up the data
  - Totally smooth
- A few partitions would get updated
  - This is where things went wrong

©2024 Databricks Inc. — All rights reserved

LESSON 1: KNOW YOUR DATA

The Reality

- We noticed that what should have been a quick update took 30+ minutes
- Logs indicated that almost all of the data was being updated
- Nothing wrong with the CDC
- We then found out what was happening with the source data…
  - We had minimal control of the upstream data
  - The source system was constantly updating most of the tables, across the entire data set, every 30 minutes

©2024 Databricks Inc. — All rights reserved

LESSON 1: KNOW YOUR DATA

The Impact

- There are 650m+ rows and we get a few minutes to do the Bronze/Silver update. Updating 90% of partitions is resource intensive
- Gut response: Throw more workers at it
  - However, one of the requirements is to keep the cost down
- The resulting change was that we needed more frequent CDC updates
- And… we needed to seriously consider the impact of using SCD Type 2

©2024 Databricks Inc. — All rights reserved

LESSON 1: KNOW YOUR DATA

Misunderstandings about incoming data can have significant impacts

- **Assumption:** Most changed data will come in as new records with a few old records updated. One partition would always be updated, with a small percentage of others needing an update
- **Reality:** The underlying IIS is making constant rolling updates across all records. 90% of partitions require updating
- **Impact:** There are 650m+ rows and we get a few minutes to do the Bronze/Silver update. Updating 90% of partitions is resource intensive

©2024 Databricks Inc. — All rights reserved

LESSON 2: UNDERSTAND DATA STRUCTURE

The Assumption

- Using SCD Type 2 we could preserve the historical data
- The resulting dataset would be large, but manageable as old records rarely changed
- Due to few partitions being updated, time would not be overly affected

©2024 Databricks Inc. — All rights reserved

LESSON 2: UNDERSTAND DATA STRUCTURE

The Reality

The difference between SCD Type 1 and Type 2

- Example CDC Input:

userId	name	city	operation	sequenceNum
124	Raul	Oaxaca	INSERT	1
123	Isabel	Monterrey	INSERT	1
125	Mercedes	Tijuana	INSERT	2
126	Lily	Cancun	INSERT	2
123	null	null	DELETE	6
125	Mercedes	Guadalajara	UPDATE	6
125	Mercedes	Mexicali	UPDATE	5
123	Isabel	Chihuahua	UPDATE	5

©2024 Databricks Inc. — All rights reserved

LESSON 2: UNDERSTAND DATA STRUCTURE

The Reality

SCD Type 1 Example:

userId	name	city
124	Raul	Oaxaca
125	Mercedes	Guadalajara
126	Lily	Cancun

SCD Type 2 Example:

userId	name	city	__START_AT	__END_AT
123	Isabel	Monterrey	1	5
123	Isabel	Chihuahua	5	6
124	Raul	Oaxaca	1	null
125	Mercedes	Tijuana	2	5
125	Mercedes	Mexicali	5	6
125	Mercedes	Guadalajara	6	null
126	Lily	Cancun	2	null

©2024 Databricks Inc. — All rights reserved

LESSON 2: UNDERSTAND DATA STRUCTURE

The Reality

- Given that our understanding of the number of changing partitions was wrong, the difference between using Type 1 and Type 2 became significant
- The table size significantly increased over time

©2024 Databricks Inc. — All rights reserved

LESSON 2: UNDERSTAND DATA STRUCTURE

The Impact

- We still had a requirement to track changes
- It was clear that we could not sustain the overhead of Type 2
- Decision: Abandon using Type 2 for the Bronze layer. Instead create a nightly history that didn’t affect the regular job

©2024 Databricks Inc. — All rights reserved

LESSON 2: UNDERSTAND DATA STRUCTURE

Choosing between SCD Type 1 and Type 2 greatly affects

- **Assumption:** Assuming a small number of partitions being changed, SCD Type 2 would allow us to preserve the change history
- **Reality:** With 90% of partitions being updated, SCD Type 2 would require significant overhead
- **Impact:** SCD Type 2 requires multiple lookups and joins across every partition, with a complete rewrite of each one. This became resource prohibitive

©2024 Databricks Inc. — All rights reserved

LESSON 3: CLUSTER STRATEGY

The Assumption

- The organization’s existing policies would be sufficient
- We would have the ability to configure the workers as needed
- The Databricks admin team would be responsive to our requests

©2024 Databricks Inc. — All rights reserved

LESSON 3: CLUSTER STRATEGY

The Reality

- The organization was new to administering Databricks
- Their policies were based on ad hoc data analyst and data science work
- Initially, we were limited to very inefficient/unstable cluster policies. For example, up until that point they required all clusters to be Interactive Clusters

©2024 Databricks Inc. — All rights reserved

LESSON 3: CLUSTER STRATEGY

The Impact

- Change management and additional training were needed
- The immediate plan made use of existing policies
- Phased approach was used to create and migrate to the new policies

©2024 Databricks Inc. — All rights reserved

LESSON 3: CLUSTER STRATEGY

Choose an effective cluster strategy

- **Assumption:** Organizational default profiles would allow sufficient time to start the clusters and run all processing
- **Reality:** The organization lacked sufficient maturity to accommodate our needs
- **Impact:** The organization needed to realign expectations of the types of jobs that would be run. Pools, instance types, and cluster types all needed to be reconsidered

©2024 Databricks Inc. — All rights reserved

LESSON 4: DECOUPLE WORKFLOWS

The Assumption

- The business needs both the Silver and Gold zones to be available by 45 minutes after the hour
- Minor adjustments to the cluster would give enough of a performance increase

©2024 Databricks Inc. — All rights reserved

LESSON 4: DECOUPLE WORKFLOWS

The Reality

- Once again, the misconception about the changed data and partitioning posed a problem
- The new MPI required significantly more overhead
- Migrating the existing pipeline logic led to inefficiencies and uncertainty around which data was available throughout the process

©2024 Databricks Inc. — All rights reserved

LESSON 4: DECOUPLE WORKFLOWS

The Impact

- We worked with the business to break this down to two different requirements:
  - The Silver layer needed to be *available within* 45 minutes of an update
  - The post-MPI Gold layer needed to be *accessible and stable* every 45 minutes
- This meant that we could treat them as two different problems
- The CDC data was processed through DLT every 15 minutes
- The MPI job was run every hour

©2024 Databricks Inc. — All rights reserved

LESSON 4: DECOUPLE WORKFLOWS

Decoupling unrelated workflows and using an asynchronous approach can yield additional gains

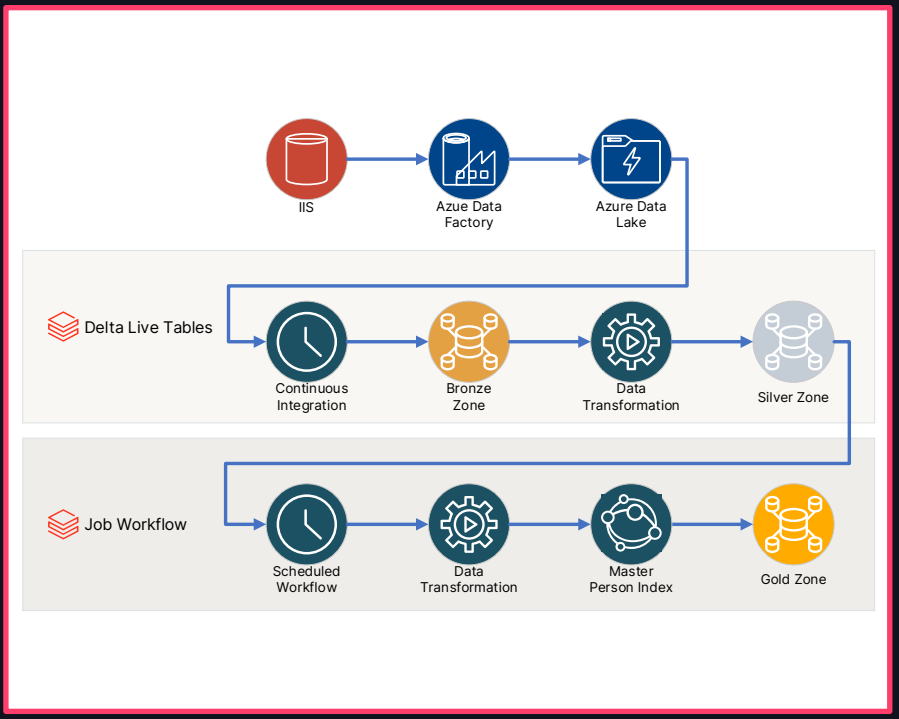
- **Assumption:** The existing sequential approach to loading and processing the data would ensure data consistency
- **Reality:** Our consumers had different requirements for accessing the Gold layer than they did for the Silver layer
- **Impact:** By more frequently updating the Bronze and Silver layer using a separate workflow, resources were freed up to perform more complex operations to generate our Gold layer, allowing us to meet our required timelines

©2024 Databricks Inc. — All rights reserved

THE SOLUTION

©2024 Databricks Inc. — All rights reserved

THE SOLUTION



- Overall Orchestration is still driven by Azure Data Factory
- Change data capture is handled using Oracle MVLogs
- DLT ingests the changed data (Bronze and Silver)
- Job Workflows engage the Master Person Index (MPI) and creates the Gold zone

©2024 Databricks Inc. — All rights reserved