Healthcare Revenue Cycle Management (RCM)

By

KANDIMALLA MAHENDRA

Microsoft Fabrics Domain

Table of Contents:

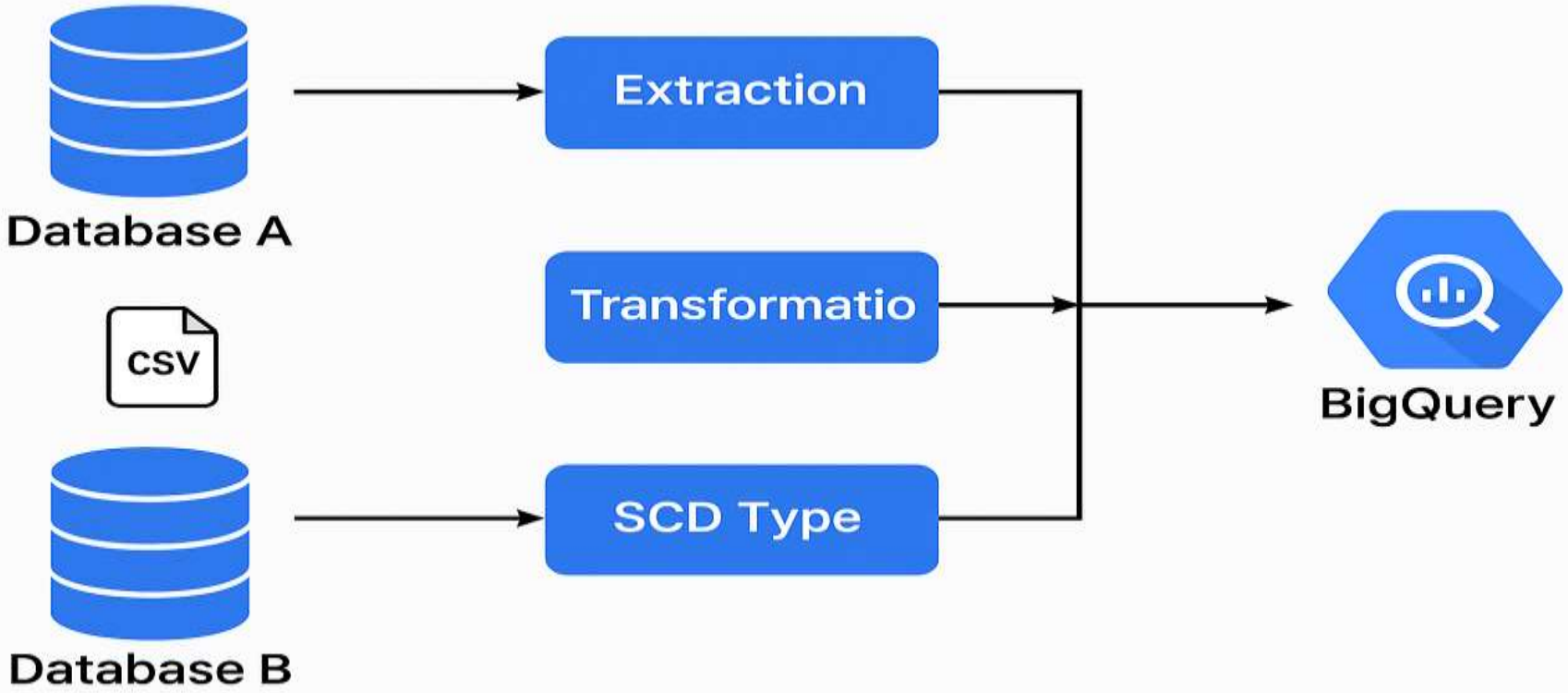
INTRODUCTION

- This project focuses on building a scalable Data Engineering pipeline for Healthcare Revenue Cycle Management (RCM).
- It unifies and transforms patient, transaction, and claims data from multiple hospital sources.
- The pipeline applies dimensional modeling with SCD Type 2 to track historical changes.
- Processed data is incrementally loaded into BigQuery for analytics and reporting.

Tech Stack

- **Languages:** Python, SQL
- **Databases:** MySQL (local), Google BigQuery (cloud)
- **Libraries:** pandas, sqlalchemy, google-cloud-bigquery
- **Development Tools:** Vs code , Git

Architecture



Data Source

- Sample Patient Data

	PatientID	FirstName	LastName	MiddleName	SSN	PhoneNumber	Gender	DOB	Address
▶	HOSP1-000001	Rick	Russo	Z	188-23-9828	+1-630-829-7585x0769	Female	1937-06-04	Unit 0915 Box 7064, DI
	HOSP1-000002	Gregory	Graham	B	730-45-8217	456.746.7289x69233	Female	1937-06-10	9864 Gibson Islands, D
	HOSP1-000003	Mary	Ryan	H	348-14-7947	522-501-5461	Female	1926-08-09	6194 Joseph Turnpike,
	HOSP1-000004	Daniel	Brown	D	013-38-1645	+1-345-608-9409	Male	1971-10-23	780 Conrad Isle, Priceb
	HOSP1-000005	Brad	Carroll	M	461-53-6290	963.994.2969x6232	Male	1927-10-18	3167 Hall Burg, Tanner
	HOSP1-000006	Melissa	Lester	Y	359-88-6883	511-940-3097x95946	Female	1994-03-22	542 Rogers Divide, Sou
	HOSP1-000007	Stephanie	Griffin	Q	879-32-6527	313.753.8992x0867	Male	1957-03-10	3952 Le Highway, Frer
	HOSP1-000008	Sandra	Holt	L	148-25-8251	+1-335-229-5080	Female	2024-07-31	5102 Thompson Locks
	HOSP1-000009	Austin	Cline	H	219-06-7294	001-899-655-3369x486	Male	2001-11-16	465 Flores Forest, Port
	HOSP1-000010	Ryan	Hall	A	425-22-5742	(962)707-1206x023	Female	1983-05-06	915 Joseph Walks Suite
	HOSP1-000011	Catherine	Campbell	S	168-04-0477	6725501528	Female	1957-05-15	241 Elizabeth Causewa

Implementation

1. Data Extraction

- Connect to two MySQL source databases using config-driven Python scripts.
- Execute SQL queries to extract patient and transaction data.
- Load results into Pandas DataFrames for further processing.
- Ensure connection management with proper logging and error handling.

2. Data Cleansing and Transformation

- Clean raw data by removing duplicates, handling nulls, and standardizing formats.
- Merge patient and transaction datasets based on primary key
- Derive new fields like age, total paid amount, and calculated metrics.
- Standardize column names, formats, and data types for consistency.
- Prepare clean, enriched data ready for dimensional modeling.

3. Dimensional Modeling and SCD Type 2

- From the transformed data, created dimension and fact tables using a star schema approach.
- Apply SCD Type 2 logic to track historical changes in dimension data.
- Compare incoming data with existing BigQuery table to detect changes.
- Perform incremental loads by inserting only new or changed records with effective and expiry dates.

4.Load to Big Query

- Create partitioned and clustered BigQuery tables for facts and dimensions to optimize performance.
- Apply incremental load by inserting only new or changed records based on SCD Type 2 logic.
- Use Python BigQuery client or pandas load transformed DataFrames into BigQuery.
- Maintain data accuracy and consistency with proper schema validation, logging, and idempotent job execution.

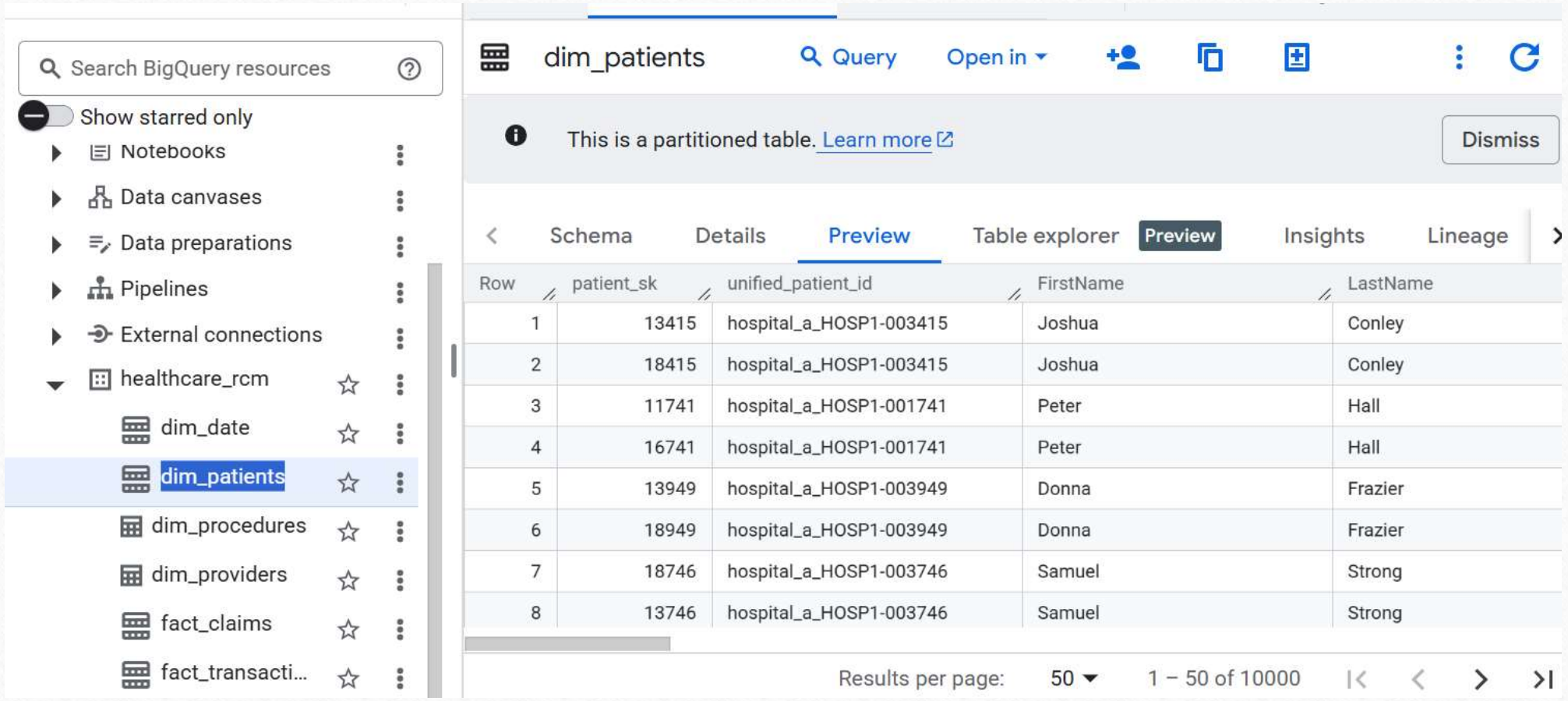
Sample code

run_pipeline.py | ReadME.md | load.py | gcp_key.json | scdtype2.py | datacle

run_pipeline.py > ...

```
153   def main():
205       fact_claims = create_fact_claims(claims_df, updated_dim_patients, dim_date)
206
207       validate_referential_integrity(fact_transactions, updated_dim_patients, dim_providers, dim_procedu
208       validate_referential_integrity(fact_claims, updated_dim_patients, None, None, dim_date)
209
210       print("\n🚀 Phase 6: Loading to BigQuery")
211       print("==============================")
212
213       load_to_bigquery(updated_dim_patients, "dim_patients", partition_field="effective_date", cluster_f
214       load_to_bigquery(dim_providers, "dim_providers")
215       load_to_bigquery(dim_procedures, "dim_procedures")
216       load_to_bigquery(dim_date, "dim_date", partition_field="date")
217       load_to_bigquery(fact_transactions, "fact_transactions", partition_field="ServiceDate", cluster_fi
218       load_to_bigquery(fact_claims, "fact_claims", partition_field="ServiceDate", cluster_fields=["Claim
219
220       print("\n✅ Pipeline completed successfully!")
221           You, 5 hours ago • extraction
222   if __name__ == "__main__":
223       main()
224
```

Big Query Tables Image



Conclusion

This project successfully built a scalable and modular data pipeline for healthcare revenue cycle management. Starting from raw data extraction, we systematically cleansed, transformed, and modeled the data into meaningful structures. By implementing historical tracking using SCD Type 2 and enabling incremental loading into BigQuery, the solution ensures both data accuracy and performance. The final output supports efficient reporting and analytics, laying the foundation for deeper business insights and decision-making in the healthcare domain.

Thank You …….