03 Healthcare Revenue Cycle Management (RCM)

🏥 Healthcare Revenue Cycle Management (RCM) - Data Engineering Challenge

📋 Overview

Welcome to the **Healthcare Revenue Cycle Management Data Engineering Challenge**! This comprehensive project simulates a real-world scenario where you'll work as a data engineer for a healthcare analytics company.

Scenario: You've been hired by HealthTech Analytics, a company that provides data solutions to healthcare providers. Your client is a regional healthcare network operating two hospitals that need to optimize their revenue cycle management process. Currently, each hospital maintains separate systems, making it difficult to get unified insights into their financial performance.

Your Role: As the lead data engineer, you'll design and implement a complete data pipeline that consolidates data from multiple sources, ensures data quality, tracks historical changes, and provides actionable insights to healthcare administrators.

Technical Stack:

- **Languages:** Python, SQL
- **Databases:** MySQL (local), Google BigQuery (cloud)
- **Libraries:** pandas, sqlalchemy, google-cloud-bigquery
- **Complexity:** Intermediate
- **Duration:** 3-4 hours

🎯 Problem Statement

Business Challenge

The regional healthcare network faces critical challenges in their revenue cycle management:

1. **Data Silos:** Each hospital uses separate databases, making cross-hospital analysis impossible
2. **Manual Processes:** Claims processing insights are extracted manually, leading to delays
3. **Historical Tracking:** No system to track changes in patient information over time
4. **Revenue Leakage:** Inability to identify patterns in claim denials and payment delays
5. **Compliance Risks:** Difficulty in maintaining audit trails for patient data changes

Technical Requirements

You must build a data pipeline that addresses these challenges by:

Data Integration:

- Extract patient, provider, and transaction data from two separate hospital MySQL databases
- Process monthly insurance claims data from CSV files
- Combine data sources into a unified data model

Data Quality & Governance:

- Implement comprehensive data validation and cleansing procedures
- Create standardized patient identifiers across hospitals
- Establish data quality scoring and monitoring

Historical Data Management:

- Implement Slowly Changing Dimension (SCD) Type 2 to track patient information changes
- Maintain audit trails for compliance and analytical purposes
- Preserve historical context for trend analysis

Analytics Infrastructure:

- Load processed data into Google BigQuery for scalable analytics

- Create dimensional model with fact and dimension tables
- Enable self-service analytics for business stakeholders

Performance Metrics:

- Calculate key RCM performance indicators
- Identify revenue optimization opportunities
- Provide insights into operational efficiency

Success Criteria

Your solution will be considered successful if it:

- ✅ Processes 10,000+ patient records with 99%+ data quality
- ✅ Successfully implements SCD Type 2 with version tracking
- ✅ Loads clean, validated data into BigQuery within acceptable timeframes
- ✅ Provides accurate RCM metrics matching business requirements
- ✅ Demonstrates proper error handling and logging throughout the pipeline
- ✅ Creates documentation suitable for production deployment

🎓 Learning Objectives

Primary Learning Outcomes

Upon successful completion of this challenge, you will have demonstrated mastery of:

1. Database Integration & Connectivity

- **MySQL Integration:** Connect to multiple MySQL databases using Python
- **Connection Management:** Implement connection pooling and error handling
- **Cross-Database Queries:** Extract and combine data from multiple sources
- **Data Source Documentation:** Create comprehensive data dictionaries

2. ETL Pipeline Development

- **Extraction Techniques:** Build robust data extraction processes with error handling
- **Transformation Logic:** Implement complex business rules and data standardization
- **Loading Strategies:** Design efficient data loading processes for cloud warehouses
- **Pipeline Orchestration:** Structure ETL workflows for maintainability and monitoring

3. Data Quality Management

- **Validation Rules:** Implement comprehensive data validation frameworks
- **Data Cleansing:** Apply standardization, deduplication, and format correction
- **Quality Scoring:** Create data quality metrics and monitoring dashboards
- **Error Handling:** Design robust exception handling and recovery mechanisms

4. Dimensional Data Modeling

- **Star Schema Design:** Create efficient dimensional models for analytics
- **Fact Table Construction:** Build transaction-level fact tables with proper grain
- **Dimension Management:** Design and populate customer, product, and time dimensions
- **Surrogate Key Strategy:** Implement proper key management across the data warehouse

5. Slowly Changing Dimensions (SCD) Implementation

- **SCD Type 2 Logic:** Track historical changes with effective and expiry dates
- **Change Detection:** Implement algorithms to identify data changes
- **Version Control:** Maintain version history for audit and compliance
- **Performance Optimization:** Design SCD processes for large-scale data processing

6. Cloud Data Warehousing

- **BigQuery Architecture:** Understand columnar storage and distributed processing

- **Schema Design:** Optimize table structures for analytical workloads
- **Partitioning & Clustering:** Implement performance optimization techniques
- **Cost Management:** Design queries and structures for cost-effective operations

7. Healthcare Domain Expertise

- **RCM Process Understanding:** Learn healthcare billing and claims processing workflows
- **Healthcare Data Standards:** Apply HIPAA compliance and healthcare data best practices
- **Clinical Terminology:** Work with medical procedure codes, diagnosis codes, and provider identifiers
- **Healthcare Analytics:** Calculate industry-standard healthcare financial metrics

Secondary Learning Outcomes

You will also develop skills in:

8. Production-Ready Development

- **Code Organization:** Structure code for maintainability and team collaboration
- **Documentation Standards:** Create technical documentation for production deployment
- **Testing Strategies:** Implement data validation and pipeline testing
- **Monitoring & Logging:** Build observability into data pipelines

9. Business Intelligence & Analytics

- **KPI Development:** Translate business requirements into measurable metrics
- **Analytical Thinking:** Identify trends, patterns, and insights in healthcare data
- **Stakeholder Communication:** Present technical solutions to business audiences
- **Performance Analysis:** Evaluate and optimize pipeline performance

10. Professional Skills

- **Problem-Solving:** Break down complex requirements into manageable tasks
- **Project Management:** Plan and execute multi-phase technical projects
- **Quality Assurance:** Implement comprehensive testing and validation procedures
- **Continuous Learning:** Adapt to new technologies and healthcare industry changes

Career Readiness

This project prepares you for roles such as:

- **Data Engineer** - Healthcare technology companies
- **ETL Developer** - Hospital systems and health insurance companies
- **Healthcare Data Analyst** - Regional health systems and ACOs
- **Business Intelligence Developer** - Healthcare consulting firms
- **Data Architect** - Electronic health record (EHR) vendors

Industry Relevance

The skills demonstrated in this project directly apply to:

- **Healthcare Data Integration** projects at hospital systems
- **Claims Processing** optimization at insurance companies
- **Population Health** analytics for accountable care organizations
- **Revenue Cycle** optimization consulting engagements
- **Healthcare Compliance** and audit support systems

🚀 Challenge Tasks

Phase 1: Environment Setup (30 minutes)

Task 1.1: Database Setup

- [ ] Install MySQL locally and create two databases: `hospital_a_db` and `hospital_b_db`
- [ ] Load the provided sample data using the setup scripts

- [ ] Verify data integrity and explore table structures

Task 1.2: BigQuery Setup

- [ ] Create a Google Cloud project and enable BigQuery API
- [ ] Set up service account authentication
- [ ] Create a dataset named `healthcare_rcm`

Task 1.3: Python Environment

- [ ] Install required packages: `pandas` , `mysql-connector-python` , `google-cloud-bigquery` , `sqlalchemy`
- [ ] Test database connections to both hospitals
- [ ] Verify BigQuery connectivity

Phase 2: Data Extraction (45 minutes)

Task 2.1: Hospital Data Extraction

Create a `DataExtractor` class that can:

- [ ] Connect to both hospital MySQL databases
- [ ] Extract patient data with proper error handling
- [ ] Extract transaction data for specified date ranges
- [ ] Log extraction statistics (record counts, timing)

Task 2.2: Claims Data Extraction

- [ ] Read monthly insurance claims CSV files
- [ ] Handle different file formats and schemas
- [ ] Validate required columns are present

Task 2.3: Data Source Integration

- [ ] Combine data from both hospitals
- [ ] Add source identifiers to track data origin
- [ ] Create unified patient ID across hospitals

Expected Output: Raw DataFrames with ~10,000 patients and ~50,000 transactions

---

Phase 3: Data Transformation (60 minutes)

Task 3.1: Data Quality & Cleansing

Implement comprehensive data cleaning:

- [ ] Remove duplicate records
- [ ] Standardize name formats (proper case)
- [ ] Clean and format phone numbers
- [ ] Validate email addresses using regex
- [ ] Flag records with data quality issues

Task 3.2: Business Logic Implementation

- [ ] Calculate patient age from date of birth
- [ ] Compute insurance coverage percentages
- [ ] Categorize payment statuses (Paid, Partial, Pending, Denied)
- [ ] Add time dimensions (year, month, quarter, day of week)

Task 3.3: Common Data Model (CDM)

Create standardized schemas:

- [ ] Normalize patient data across hospitals
- [ ] Standardize procedure codes and descriptions

- [ ] Create consistent transaction structures
- [ ] Generate surrogate keys where needed

Expected Output: Clean, standardized DataFrames ready for dimensional modeling

---

Phase 4: Dimensional Modeling (45 minutes)

Task 4.1: Dimension Tables

Create the following dimension tables:

- [ ] `dim_patients` - Patient demographics and insurance info
- [ ] `dim_providers` - Healthcare provider details
- [ ] `dim_procedures` - Medical procedure codes and descriptions
- [ ] `dim_date` - Date dimension with business calendar

Task 4.2: Fact Tables

- [ ] `fact_transactions` - Billing transactions (grain: one row per transaction)
- [ ] `fact_claims` - Insurance claims processing (grain: one row per claim)

Task 4.3: Data Validation

- [ ] Verify referential integrity between fact and dimension tables
- [ ] Check for orphaned records
- [ ] Validate business rules (e.g., amounts > 0, valid dates)

Expected Output: Star schema with fact and dimension tables

Phase 5: SCD Type 2 Implementation (60 minutes)

Task 5.1: SCD Logic Development

Implement Slowly Changing Dimension Type 2 for patient data:

- [ ] Identify which patient attributes should be tracked historically
- [ ] Add SCD columns: `effective_date`, `expiry_date`, `is_current`, `version`
- [ ] Create logic to detect changes in tracked attributes

Task 5.2: Change Detection

- [ ] Compare new patient data against existing records
- [ ] Identify patients with changed information (address, phone, insurance)
- [ ] Flag new patients vs. existing patients with updates

Task 5.3: Historical Tracking

- [ ] Close expired records by updating `expiry_date` and `is_current`
- [ ] Insert new records for changed patients
- [ ] Maintain version numbers for audit trails

Challenge: Handle edge cases like patients switching between hospitals

Expected Output: Patient dimension with full historical tracking

---

Phase 6: BigQuery Integration (45 minutes)

Task 6.1: Schema Creation

- [ ] Create BigQuery tables with appropriate schemas
- [ ] Set up partitioning on date columns for performance
- [ ] Configure clustering on frequently queried columns

Task 6.2: Data Loading

- [ ] Load dimension tables to BigQuery

- [ ] Load fact tables with proper data types
- [ ] Implement incremental loading for new data

Task 6.3: Data Validation in BigQuery

- [ ] Run row count validations
- [ ] Check for data type consistency
- [ ] Verify foreign key relationships

Expected Output: Fully populated BigQuery dataset ready for analytics

---

Phase 7: RCM Analytics (45 minutes)

Task 7.1: Key Performance Indicators

Write SQL queries to calculate:

- [ ] **Revenue Metrics:** Total revenue, revenue by hospital, monthly trends
- [ ] **Claims Performance:** Approval rates, denial rates, average processing time
- [ ] **Patient Metrics:** Patient volume, demographics analysis, insurance mix
- [ ] **Operational Efficiency:** Days in A/R, collection rates, write-off amounts

Task 7.2: Advanced Analytics

- [ ] Patient lifetime value analysis
- [ ] Procedure profitability analysis
- [ ] Seasonal trends in patient volume
- [ ] Insurance company performance comparison

Task 7.3: Data Quality Reports

- [ ] Create dashboards showing data completeness
- [ ] Track SCD version distributions
- [ ] Monitor ETL pipeline performance metrics

Expected Output: Business-ready analytics queries and insights

---

📋 Deliverables

Submit the following:

1. Code Repository

- [ ] Python ETL scripts with proper documentation
- [ ] SQL DDL scripts for BigQuery table creation
- [ ] Configuration files for database connections
- [ ] Requirements.txt with all dependencies

2. Data Pipeline Documentation

- [ ] Architecture diagram showing data flow
- [ ] Data dictionary for all tables and columns
- [ ] ETL process documentation with error handling
- [ ] SCD Type 2 implementation explanation

3. Analytics Report

- [ ] Executive summary of key findings
- [ ] RCM performance metrics and insights
- [ ] Data quality assessment results
- [ ] Recommendations for process improvements

4. BigQuery Artifacts

- [ ] All dimension and fact tables properly loaded

- [ ] Sample analytical queries demonstrating RCM insights
- [ ] Data validation results and quality checks

🏆 Evaluation Criteria

Your solution will be evaluated on:

Technical Excellence (40%)

- Code quality, modularity, and documentation
- Proper error handling and logging
- Efficient SQL queries and data processing
- Correct implementation of SCD Type 2

Data Quality (25%)

- Comprehensive data validation and cleansing
- Proper handling of missing/invalid data
- Consistent data standards across sources
- Referential integrity maintenance

Business Understanding (20%)

- Understanding of RCM domain concepts
- Relevant KPI calculations and insights
- Practical recommendations for healthcare operations
- Clear communication of technical concepts

Architecture & Scalability (15%)

- Well-designed ETL pipeline architecture
- Proper use of BigQuery features (partitioning, clustering)
- Consideration for production deployment
- Performance optimization techniques

💡 Bonus Challenges

For extra credit, implement:

- [ ] **Real-time Processing:** Stream processing for real-time claims updates
- [ ] **Data Lineage:** Track data lineage from source to analytics
- [ ] **Automated Testing:** Unit tests for ETL functions and data validation
- [ ] **Monitoring & Alerting:** Pipeline monitoring with email notifications
- [ ] **API Integration:** Pull reference data from external APIs (CPT codes, ICD codes)