International Journal of Business Quantitative Economics and Applied Management Research
Volume-7, Issue-10, 2024 ISSN No: 2349-5677

BUILDING ENTERPRISE-LEVEL DATA MANAGEMENT TOOLS FOR SECURE FINANCIAL DATA

Sandeep Rachapudi
Sandeep.Rachapudi@gmail.com

Abstract

Managing data securely with proper authentication and authorization mechanisms is key to the success of financial systems. The need for accessing production data and performing (Create, Read, Update, Delete) CRUD operations securely based on business needs has led to the development of various processes and tools for financial enterprises. This paper aims to provide strategies for developing robust data management tools for financial enterprises.

Keywords: Data manipulation, Data tools, Data editor, Data management

I. INTRODUCTION

Every financial institution deals with data in various forms. An analyst needs to access the data to perform data analysis [1]. Data engineers work to design, construct, and manage systems that collect, store, and process data. Data interactions are based on the role of the person using the tool. Traditionally, data manipulation in production environment required engineering teams to create change requests. Support teams would further process the request to perform the required actions in production environment on behalf of the requested team.

II. PROBLEM

The process of working with the application support teams, obtaining approvals from management, (Database administrator) DBA and Dev Ops teams is a time-consuming process. Even a minor change in scripts leads to another approval process, making it more inefficient to perform data fixes or perform (data definition language) DDL statements.

These inefficiencies involved in performing data manipulation can be addressed by developing data management tools based on organizational needs.

III. SOLUTION

Developing data management tools for financial systems involves several key steps, focusing on ensuring accuracy, security, and efficiency.
Engineering teams must define data model before implementing the tool. There are several models to define for the development of a data management tool, such as:

- **Conceptual Model:** Outline high-level data entities and their relationships. This helps in understanding what data needs to be captured and how it interrelates.
- **Logical Model:** Detail the structure of your data, including tables, fields, and relationships. This is more specific than the conceptual model and includes data types and constraints.
- **Physical Model:** Define how data will be physically stored in the database. This includes indexing, partitioning, and storage considerations.

The tool must have several modules to enable the user to define the data editor with business rules and user access management. A data management tool should have the following modules:

Module 1 - Database Setup and Integration: The setup and integration of database management tools involves identifying the tables that need data manipulation by business users or engineering teams. Not all the tables in a database need integration with data management tools. Tables containing personally identifiable information (PII) may need an additional approval process. Similarly, integrating system configuration tables is at the discretion of engineering teams. Business analysts need to access and manipulate data warehouse tables for analysis and business-related data updates.

This module should determine where data will come from, such as transactional systems, external financial data providers, or internal databases. Different types of source data should be configurable (e.g., databases, flat files, APIs).

Database setup should define the data storage mechanism. Based on the development framework used by various organizations, intermediate tables are used to perform CRUD operations before final action can be taken on the target table. Hence, data storage should have the following functionalities:

1.1 Select a Database: Choose a database system (e.g., relational databases like SQL Server, Oracle, or NoSQL databases like MongoDB).

1.2 Schema Design: Create tables, define primary and foreign keys, and set up relationships.

1.3 Data Types and Constraints: Define data types for each field and set constraints (e.g., unique, not null).

Module 2 -Data Editor: Data editor provides functionality to configure and create a rule-based data editor based on business needs.This is the most important technical feature of a data management tool. Users should be able to select the database schema, tables and apply rules using the dynamic rule generators. Rule-based SQL queries generated by the tool help define the functionality of the editor.

It's best to create data editors in a wizard style that can guide the users to apply the rules, constraints, data operators, and joins in a logical order.

The SQL query generated will be applied on the production database system in the form of a script that's executed during a nighty batch cycle or as an immediate update to the database.
If there are DDL(data definition language) statements in the generated query by the data editor module, the query goes through nightly batch as it's not advisable to create database objects in runtime.

If there are DML (data manipulation language) statements in the generated query, it can be executed immediately.

All the commands are executed based on the rules applied while creating data editors for the specific needs of the organization. Hence, building the editor forms the basis for a successful interface with production data environments without going through the review and approval process for data manipulation and data management.

Components and Column Options provide an additional set of tools that can be used to maximize ease of use and efficiency when users are interacting with the Online Editor tool. These components allow users to edit data using the online editor without needing to add the exact format required for each field type. Using components like (Date Picker, Drop Downs, Number Editor) you can ensure that data is easy to change and always remains in the correct format.

Data editors must be developed with the option of performing data updates online or offline. This means the editor should be able to make updates by reading data from files. Offline data updates are a desired feature since most business teams would like to take their time to think and analyze data before applying changes to production.

Data updates can happen in a couple of ways as defined below.

- **Load to Target System:** Implement ETL (Extract, Transform, Load) processes to move transformed data into the target system.
- **Incremental Loads:** Set up processes for incremental data loads if only changes need to be updated rather than the entire dataset.
- **Real-Time Integration:** If required, set up real-time data synchronization to keep systems updated with the latest information.
- **Batch Processing:** For non-critical data, batch processing may be used to update systems at scheduled intervals.

There are several tools that can be used for data integration [3]

- **ETL Tools:** Utilize ETL tools like Apache NiFi, Talend, Informatica, or Microsoft SSIS to facilitate data integration.
- **Data Integration Platforms:** Consider platforms like Apache Kafka for real-time data streaming and integration.

As the data management tools are developed and evolve over a period, additional features can be incorporated. These are optional functionalities, but they greatly enhance the usage of the system.

- **Data Cleansing:** Address inconsistencies, duplicates, and errors in the data.
- **Data Mapping:** Map data from source systems to target systems, ensuring that data fields align correctly.
- **Data Aggregation:** Combine data from multiple sources if needed, summarizing or aggregating it to meet reporting needs.

Module 3 - Business Validation: Business validation rules are essentially created using the Data Editor with Admin role. Not every user in an organization should have access to define the rules that allow seamless access to production databases. Hence, user access and management are critical to define the functionality of Data editors.
As the scope for data management grows, one should have the ability to create, modify and delete business rules to keep them relevant for the organization.
Data management tool should allow the definition of multiple Data Editors, which are essentially rules integrated to the user interface specific to a team.
Module 4 - User access provision:
Once the table is fully configured and business rules applied for data processing, the final step is to provide access to the users who will help manage the setup of the table.Users can perform data edits on the data using the online data editor or the offline data upload process.
Data security and governance come into play during user provision. Here are some of the important elements to consider:

- **Access Controls:** Implement user roles and permissions to control who can access or modify data.
- **Data Encryption:** Ensure data is encrypted both at rest and in transit to protect sensitive information.
- **Data Retention Policies:** Establish guidelines for how long data is kept and when it should be archived or deleted.

IV. VALIDATION AND TESTING

Any tool developed needs to have verification and validation mechanisms. Data management tools must perform testing and validation for:

- **Data Accuracy:** Verify that the data in the target system matches the source data and meets quality standards.

- **Performance Testing:** Ensure that data integration processes do not negatively impact system performance.
- **Error Handling:** Implement robust error handling and logging mechanisms to track and resolve integration issues.

V. ONGOING MAINTENANCE

While developing a tool is the first step towards successful data management, as the data grows and new database integrations are required, we must maintain tools relevance and efficiency by performing the following activities:

a. Monitoring:

- Performance Monitoring: Continuously monitor data integration processes for performance and reliability.
- Data Quality Monitoring: Set up mechanisms to regularly check data quality and consistency.

b. Updates and Enhancements:[2]

- Adapt to Changes: Modify integration processes as source systems, data models, or business requirements change.
- Tool Upgrades: Update integration tools and systems to leverage new features and maintain compatibility.

c. Documentation:

- Integration Maps: Document data flows, transformation rules, and integration points.
- Procedures and Protocols: Maintain detailed documentation for data integration processes, error handling, and recovery procedures.

VIII. CONCLUSION

By carefully developing and setting up data management tools, you can ensure that your financial systems operate efficiently, providing accurate and timely information for decision-making. This foundational work supports reliable data management, reporting and analysis, which is crucial for financial management and compliance. Removing the manual approval process and avoiding time consuming release management process, we can save time and be more efficient in our day to day data management activities.

REFERENCES

1. "Data ManagementWhat It Is and Why It Matters" https://www.sas.com/en_us/insights/data-management/data-management.html (accessed Jan 25, 2024)
2. "What Is Data Management?" https://www.oracle.com/database/what-is-data-management/ (accessed Jan 15, 2024)

3. "What Is Data Management? Benefits, Types and Best Practices" https://www.netsuite.com/portal/resource/articles/erp/data-management-benefits.shtml (accessed Jan 20, 2024)