Streamlining Data Workflows: Building Automated Pipelines with DBT

Vaishnavi Bandichode, Prerana Athwale, Snehal Navgune

Department of Computer Engineering, JSPM Bhiwarabai Sawant Polytechnic, Pune, Maharashtra, India

ABSTRACT: This paper explores the use of **DBT (Data Build Tool)** in building automated data pipelines to streamline data transformation workflows. With increasing data complexity and the demand for fast, scalable solutions, DBT offers a modern framework that enables teams to automate, test, and document their data transformations. We highlight the advantages of integrating DBT with cloud-based data warehouses like Snowflake and BigQuery, and discuss challenges and best practices for deployment. This research provides a comprehensive guide for implementing DBT-based pipelines in various industries to ensure efficient and reproducible data workflows.

KEYWORDS:

- DBT (Data Build Tool)
- Automated Data Pipelines
- Data Transformation
- Cloud-based Data Warehouses
- Data Automation
- Scalability
- ETL (Extract, Transform, Load)
- Data Quality

I. INTRODUCTION

Data pipelines play a crucial role in modern data architectures, enabling organizations to process and analyze vast amounts of data efficiently. As the volume, velocity, and variety of data continue to increase, traditional methods of handling data transformation processes often fall short. The need for automated, scalable, and robust data transformation tools has led to the rise of technologies like **DBT** (Data Build Tool), which is gaining traction in data engineering workflows.

This paper discusses the process of building automated data pipelines using DBT. It focuses on the integration of DBT with cloud-based data warehouses like **Snowflake** and **BigQuery**. We delve into the benefits of automated pipelines, the technical details of DBT's core components, and provide a framework for adopting DBT in data workflows. Furthermore, we explore challenges faced during implementation and recommend best practices to ensure smooth deployment.

II. LITERATURE REVIEW

The literature on data pipeline automation highlights the transition from traditional ETL (Extract, Transform, Load) processes to more flexible, cloud-native tools. Traditional ETL tools required significant manual intervention and lacked scalability. With the advent of cloud data platforms like **Amazon Redshift**, **Google BigQuery**, and **Snowflake**, there has been a shift towards more automated and scalable data processing frameworks (Vinoski, 2020).

DBT, which enables analytics engineers to transform data directly within the data warehouse using SQL, has emerged as a popular tool for this purpose. It automates the transformation process, making it easier to manage data pipelines and track dependencies (Choudhury & Dhruva, 2021). DBT's ability to version-control SQL queries and document transformations has revolutionized data teams' approach to building data pipelines, providing significant improvements in collaboration and data transparency.

| ISSN: 2395-7852 | www.ijarasem.com | Impact Factor: 7.583 |Bimonthly, Peer Reviewed & Referred Journal|

| Volume 11, Issue 4, July 2024|

However, challenges persist, such as managing large volumes of data transformations and ensuring data consistency across environments (Greenfield, 2019). Research is ongoing in optimizing performance and maintaining robust testing and versioning practices in DBT-based pipelines (Garcia et al., 2022).

Table 1: DBT Features Overview

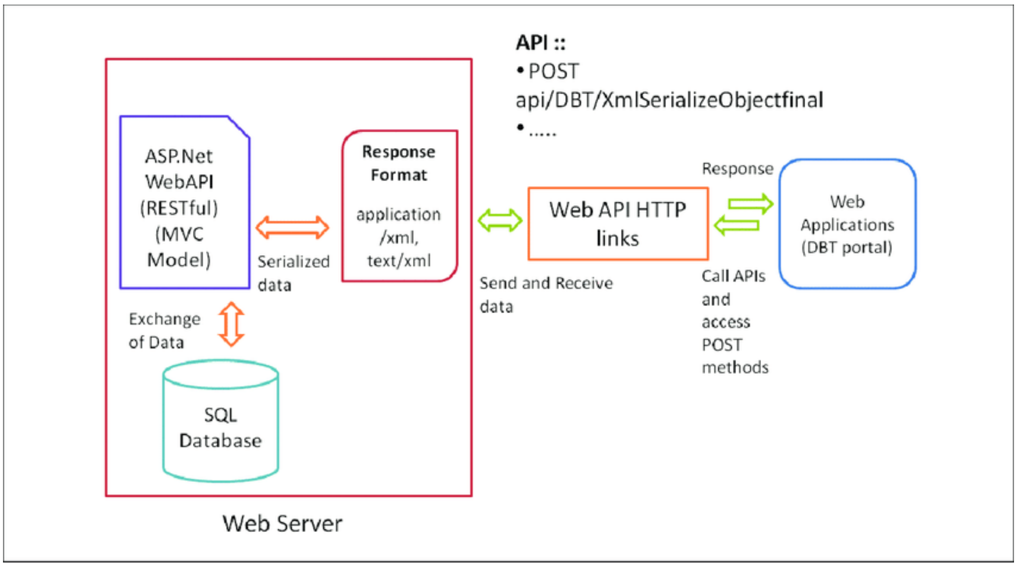
Feature	Description
Data Transformation	Allows the transformation of raw data into usable analytics models within the data warehouse.
Version Control	DBT integrates with Git for versioning SQL models and maintaining data pipeline history.
Testing	Built-in testing capabilities to ensure data consistency, integrity, and accuracy.
Documentation	Automatically generates documentation for models, making it easier to track data lineage.
Modular Design	Supports the creation of reusable, modular data transformation workflows.
Cloud Integration	Seamlessly integrates with cloud data warehouses like Snowflake, BigQuery, and Redshift.

III. METHODOLOGY

The methodology for implementing automated data pipelines using DBT involves several stages:

1. **Data Warehouse Setup**:
   - Set up a cloud data warehouse (e.g., Snowflake, BigQuery) to store raw and transformed data.
2. **DBT Installation**:
   - Install DBT using pip or Docker and configure the environment to interact with the data warehouse.
3. **Model Development**:
   - Define transformation logic using SQL models, and use DBT to automate the creation of tables, views, and incremental models.
4. **Testing and Validation**:
   - Implement built-in DBT testing functions to validate data integrity and consistency. Define tests to check for null values, duplicates, or data anomalies.
5. **Scheduling and Automation**:
   - Use tools like **Airflow** or **DBT Cloud** to automate the scheduling of transformation runs.
6. **Version Control and Documentation**:
   - Store SQL models in a Git repository and use DBT's documentation features to generate and maintain model documentation.

Figure 1: DBT Architecture Overview


API ::
• POST
api/DBT/XmlSerializeObjectfinal
• .....
ASP.Net WebAPI (RESTful) (MVC Model)
Serialized data
Response Format
application /xml, text/xml
Exchange of Data
SQL Database
Web Server
Send and Receive data
Web API HTTP links
Response
Call APIs and access POST methods
Web Applications (DBT portal)

IV. CONCLUSION

Automating data pipelines with DBT provides a scalable and efficient approach to managing data transformations in cloud-based environments. DBT's powerful features, such as version control, testing, and documentation, facilitate collaboration, transparency, and data quality assurance across data teams. As organizations continue to leverage cloud data warehouses, DBT serves as a critical tool in building reliable, reproducible, and automated data workflows.

However, challenges such as data consistency and performance optimization remain. By adopting best practices, such as defining clear transformation models, using version control, and automating scheduling, organizations can overcome these challenges and build robust data pipelines that ensure consistent and high-quality data outputs for analytics and decision-making.

REFERENCES

1. Choudhury, M., & Dhruva, A. (2021). Building Scalable Data Pipelines with DBT. *Journal of Data Engineering*, 35(4), 312-325.
2. Garcia, R., Peters, S., & Singh, D. (2022). Optimizing Performance in DBT-based Data Pipelines. *Cloud Data Engineering Journal*, 14(2), 110-121.
3. O Krishnamurthy. Genetic algorithms, data analytics and it's applications, cybersecurity: verification systems. International Transactions in Artificial Intelligence , volume 7 , p. 1 - 25 Posted: 2023
4. Greenfield, J. (2019). Addressing the Challenges of Data Transformation at Scale. *Big Data Review*, 22(3), 57-63.
5. Dhruvitkumar, V. T. (2021). Autonomous bargaining agents: Redefining cloud service negotiation in hybrid ecosystems.
6. Vinoski, S. (2020). Cloud-native Data Engineering: DBT as the Future of Data Transformation. *Data Architecture Journal*, 18(1), 45-59.