Understanding Data Contracts

Julekha Khatun

Affiliation not available

June 07, 2024

Abstract

In today’s data-driven world, seamless data exchange between different systems, teams, and organizations is crucial for operational efficiency and informed decision-making. Data contracts have emerged as a vital tool to ensure that data is exchanged reliably, consistently, and with high quality. This article thoroughly examines data contracts - formal agreements that specify how data is structured, formatted, and shared between different systems or parties. Data contracts define the schema, semantics, quality, and terms of use for data exchange, ensuring a common understanding between data providers and consumers.

Figure 1: Overall Data flow

So, let’s understand what a Data Contract is

A data contract is a formal agreement that specifies how data are structured, formatted, and shared among different systems or parties. It defines the schema, semantics, quality, and terms used for data exchange, ensuring that all parties understand the shared data. Data contracts are essential for maintaining data consistency, interoperability, and reliability across diverse systems and organizations.

To implement data contracts with access control and data quality checks, you can use a YAML file to define the schema, constraints, and access rules. Below is a sample YAML file that demonstrates how to set up a data contract with these features:

yaml
table_name: customer_bookings
version: 1.1
owner: jack_dawson
schema:
  - column_name: tx_date
type: timestamp
constraints:
  not_null: true
  no_future_dates: true
- column_name: customer_email
type: string
constraints:
  not_null: true
  check_pii: true
- column_name: sales_amt
type: decimal
constraints:
  not_negative: true
- column_name: revenue_amt
type: decimal
constraints:
  not_negative: true
- column_name: booking_type
type: string
access_control:
roles:
  - role: data_analyst
    permissions:
      - read
  - role: data_engineer
    permissions:
      - read
      - write
  - role: data_scientist
    permissions:
      - read
      - analyze
data_quality_checks:
  - check: duplicate_check
description: Ensure no duplicate records
- check: freshness
description: Table to be updated every 30 mins
- check: email_format
description: Ensure email format is correct
- check: pii_masking
description: Mask customer address and email

Explanation
1. Table Metadata:
   - 'table_name': The name of the table.
   - 'version': The version of the data contract.
   - 'owner': The owner of the data contract.

2. Schema Definition:
   - 'column_name': The name of the column.
   - 'type': The data type of the column.
   - 'constraints': Constraints applied to the column, such as 'not_null', 'no_future_dates', 'checkpii', and 'not_negative'.

3. Access Control:
   - 'roles': Defines different roles and their permissions.
     - 'role': The name of the role.
     - 'permissions': The permissions granted to the role, such as 'read', 'write', and 'analyze'.

4. Data Quality Checks:
   - 'check': The type of data quality check.
     - 'description': A brief description of the check.

Let's look at the Implementation Steps
1. Define the Data Contract: Create a YAML file with the schema, access control, and data quality checks as shown above.
2. Enforce the Data Contract: Use a CI/CD pipeline to enforce the data contract. For example, you can use GitHub Actions to validate the data contract against the data before merging any changes.
3. Monitor and Maintain: Continuously monitor the data quality and access control to ensure compliance with the data contract. Use tools like dbt, Great Expectations, or custom scripts to automate these checks.
By following these steps and using the provided YAML template, you can implement data contracts that ensure data quality and proper access control within your data architecture.

Let’s review some real-life examples of Data Contracts

Example 1: An E-commerce Platform

Consider an e-commerce platform that integrates multiple third-party vendors for inventory management, payment processing and shipping. A data contract between the e-commerce platform and payment processor specifies the structure and format of transaction data, including fields such as transaction ID, amount, currency, and timestamp. This contract maintains that both parties understand the data being exchanged and can process it accurately without errors.

Example 2: A Healthcare Data Exchange

Data contracts facilitate patient information exchange among hospitals, laboratories, and insurance companies in the healthcare industry. A data contract may define the structure of patient records, including fields for patient ID, name, diagnosis, treatment, and insurance details. This ensures that all parties can accurately interpret and use the data correctly, improve patient care, and reduce administrative errors.

Now, let’s understand some of the challenges in Implementing Data Contracts

One of the most common challenges in implementing data contracts is the resistance to change. Teams may be accustomed to existing processes and reluctant to adopt new standards and practices. Such resistance can hinder the successful implementation of data contracts.

Figure 2: How Data Contracts Work
Effective data contracts require collaboration and trust between teams and organizations. A lack of trust and poor communication can lead to misunderstandings and conflicts, complicating the establishment and enforcement of data contracts.

Data requirements often evolve over time, necessitating updates to the data contracts. Managing these changes, while ensuring backward compatibility and minimizing disruptions can be challenging.

So, how do we overcome these challenges? Let us have a look:

To overcome resistance to change, all relevant stakeholders should be involved in the design of the data contracts. This collaborative design approach ensures that the contract meets everyone’s expectations and reduces misunderstanding. Regular workshops and discussions can help build consensus and foster a collaborative culture.

The clear documentation of data contracts is essential for effective communication. Provide examples to illustrate various contract components and use standardized terminology. Regular updates and reviews of contracts can help maintain trust and ensure that all parties are on the same page.

Implementation of versioning strategy to manage changes in data requirements. Each version of the data contract should detail the modifications made to ensure backward compatibility and smooth transition for stakeholders. This approach allows flexibility while maintaining the integrity of the data exchange process.

![Figure 3: How we can solve Data Contracts challenges](image)

Figure 3: How we can solve Data Contracts challenges
Now that we have understood the challenges and steps to overcome the implementation of data contracts, let's discuss how Data Contracts Help with Data Quality

Data contracts play a crucial role in ensuring data quality by formalizing the expectations and requirements for data exchange. They help them in several ways

By Defining Data Structure and Format- Data contracts specify the schema and format of the data, ensuring that data are consistently structured and formatted across different systems. This reduces the risk of data errors and discrepancies.

By Setting Quality Standards- Data contracts include data quality standards and validation rules, ensuring that incoming data meet the predefined criteria. This helps to detect and address data quality issues early in the data processing pipeline.

By Facilitating Data Governance - Data contracts promote effective data governance by defining the roles, responsibilities, and accountability for data quality. This ensures that data are appropriately managed across the organization.

By enhancing collaboration and communication, providing a common understanding of data expectations and contracts improves the communication and cooperation between data producers and consumers. This leads to more effective data analysis and decision making.

Keywords: Data, DataContracts, DataQuality, DataGovernance, AI

Figure 4: how data contracts look

Conclusion

Data contracts ensure reliable, consistent, and high-quality data exchanges in today's interconnected world. Data contracts provide a common understanding between data providers and consumers by defining data structure, format, semantics, and quality. Although challenges such as resistance to change, lack of trust, and evolving data requirements exist, they can be overcome through collaborative design, clear documentation, and versioning strategies.

By implementing data contracts effectively, organizations can improve data governance, build trust between teams, and ensure that data are used efficiently and accurately, ultimately driving better business outcomes.

References


