Data Health Governor

Executive Summary

The Data Health Governor is an automated data-governance framework built on n8n and LLM-powered agents to continuously monitor, approve, and remediate data-quality issues in AtliQ's PostgreSQL environment.

It replaces manual weekly audits with an automated, controlled, and fully documented system that detects anomalies, routes them for human approval, applies validated SQL fixes, and generates audit-ready reports.

Key Deliverables

- 17 total issues detected across 5 core tables
- 15 issues automatically resolved (88% automation rate)
- Health Score improved from 71 → 88
- Manual effort reduced from 8 hours → 12 minutes
- Zero unauthorized SQL changes (100% approval compliance)

Business Problem

Modern organizations depend on accurate and reliable data to drive analytics, reporting, compliance, and daily operations. As data volumes grow and systems become more complex, data quality often deteriorates due to inconsistent entries, missing values, incorrect formats, and broken relationships. Many teams still rely on manual database checks, which are time-consuming, prone to errors, and inconsistent. This leads to delayed issue detection, potential mistakes in remediation, and poor communication with stakeholders.

Key challenges include:

- Lack of continuous visibility into overall database health
- Risk of human error when applying manual SQL fixes
- Inadequate processes for classifying and prioritizing data issues

Without an automated governance system that can scan the entire database, classify issues by fixability, enforce human approvals, safely execute fixes, and generate audit-ready reports, organizations face risks to analytics accuracy, regulatory compliance, and operational efficiency. Implementing such a system ensures long-term data quality, reduces manual burden, improves accountability, and supports trustworthy business decision-making.

Project Objectives

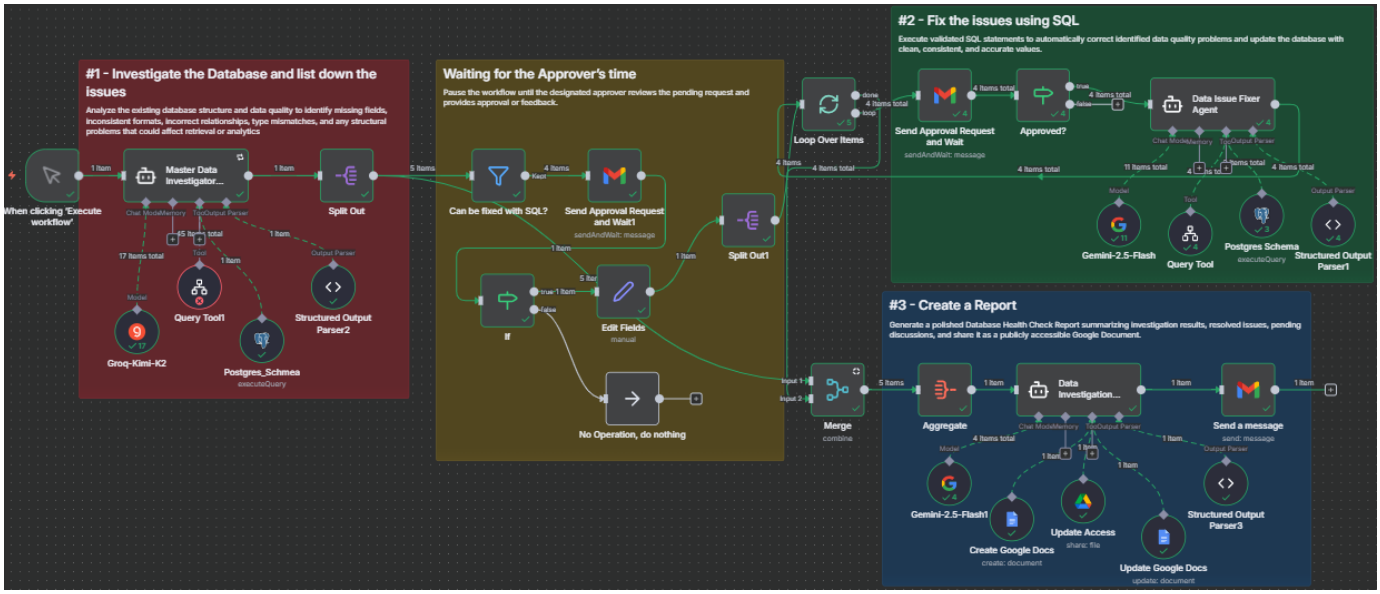
The Data Health Governor aims to transform database maintenance into an intelligent, automated system with the following goals:

- Proactive Issue Detection: Use AI-driven schema and data analysis to identify quality problems early.
- Issue Classification: Differentiate between SQL-fixable issues and those needing manual review.
- Human-in-the-Loop Approval: Ensure all changes are explicitly authorized before execution.
- Safe Automation: Apply validated SQL fixes with transactional safety and rollback options.
- Comprehensive Reporting: Produce audit-ready documentation of issues, approvals, and fixes.
- Transparent Communication: Keep stakeholders informed with automated notifications at each workflow stage.

System Architecture

The Data Health Governor uses a four-stage automated pipeline orchestrated with n8n:

- **Investigation:** AI-powered agents analyze the database schema and data to detect anomalies, constraint violations, and integrity issues.
- **Approval Workflow:** A two-tier human-in-the-loop process ensures batch and individual issue approvals via email before any changes.
- **Remediation:** Approved SQL fixes are executed safely with transactional controls, validation, and rollback support.
- **Reporting:** Audit-ready documentation is generated in Google Docs and shared with stakeholders via automated email notifications.



Workflow Description

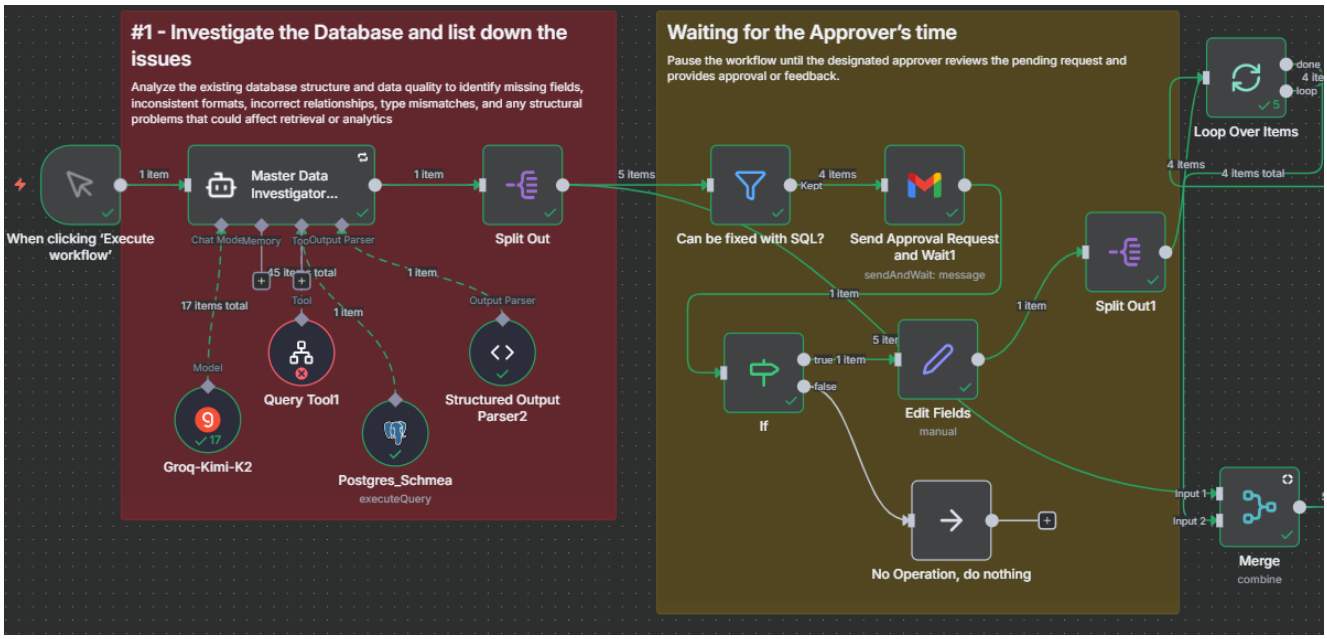
Investigation Process

- Schema Metadata Extraction:
  The Query Tool connects to PostgreSQL to retrieve comprehensive schema details, including table structures, column definitions, constraints (PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE), data types, and relationships.
- LLM-Powered Analysis:
  The Master Data Investigator agent analyzes the schema and data to identify patterns indicating quality issues, such as referential integrity breaks, constraint gaps, inconsistent data types, and anomalies.
- Issue Categorization:
  Findings are converted into standardized JSON format by the Structured Output Parser, including issue ID, description, affected table/column, issue type, a flag for SQL fixability, suggested remediation query, and estimated affected rows.
- Classification & Routing:
  Issues are split and routed—SQL-fixable items move to the approval workflow, while complex issues are directed to manual review.

Approval Mechanism

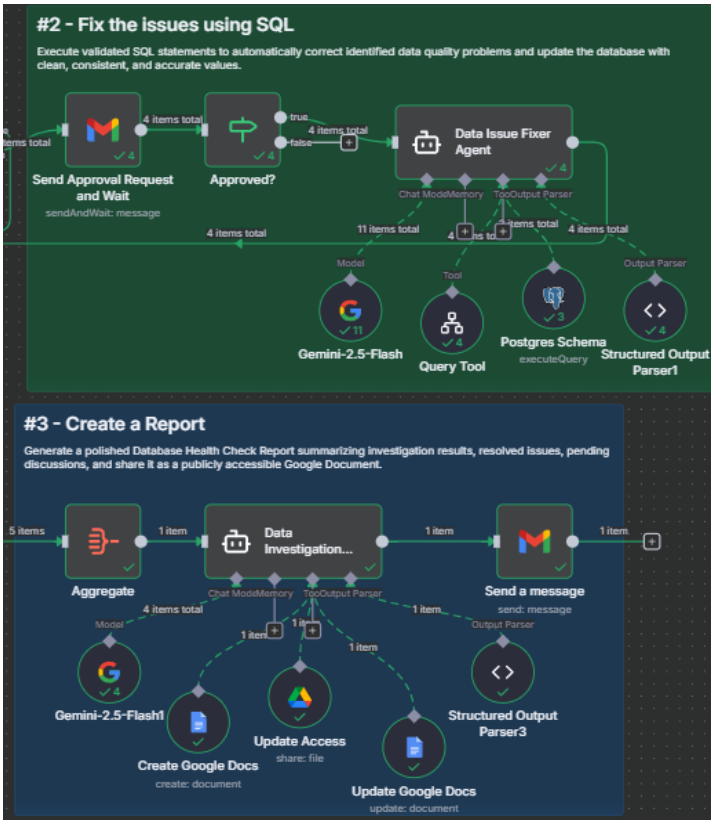
- Tier 1: Initial Batch Approval
  The designated approver receives an email summary of all detected issues and the proposed remediation approach. Approval proceeds only if the approver confirms via YES; otherwise, the process halts.

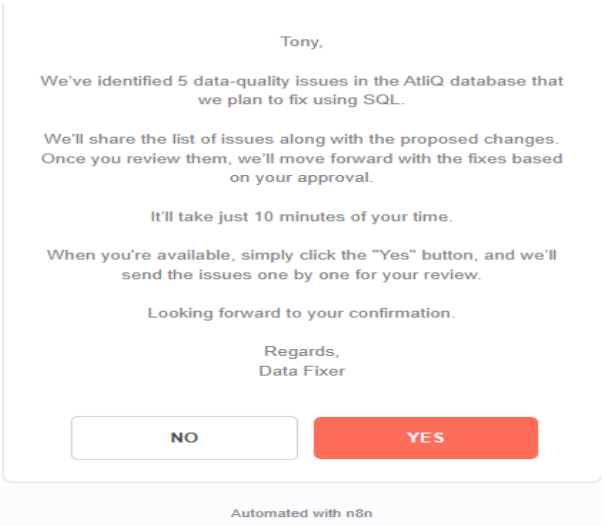
- Tier 2: Individual Issue Authorization
  For each SQL-fixable issue, a detailed email is sent with the issue ID, context, suggested SQL fix, estimated impact, and Approve/Decline options. All decisions are logged with timestamps, approver identity, original/modified SQL (if edited), and rationale.



Remediation Execution

- Sequential Processing:
  Approved issues are processed in sequence, ensuring correct order for dependent fixes.
- Transactional Safety:
  Each fix executes within its own PostgreSQL transaction, using BEGIN/COMMIT for successful updates or ROLLBACK on errors to prevent partial or corrupt changes.
- Validation & Verification:
  Execution outcomes are validated, affected row counts recorded, and errors or warnings captured to ensure results match expectations.

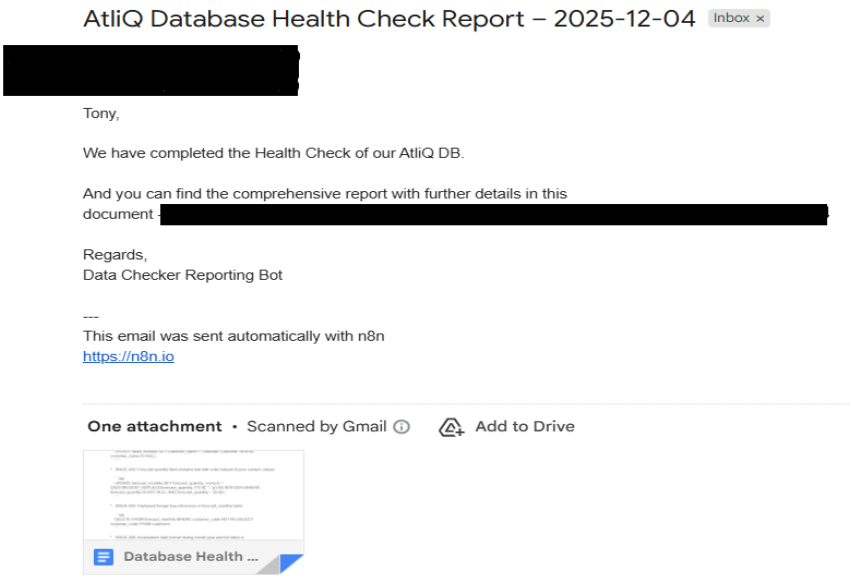




- Results Aggregation:
  The system compiles statistics on successful fixes, failed attempts, rows affected per issue, and execution time.

Reporting & Documentation

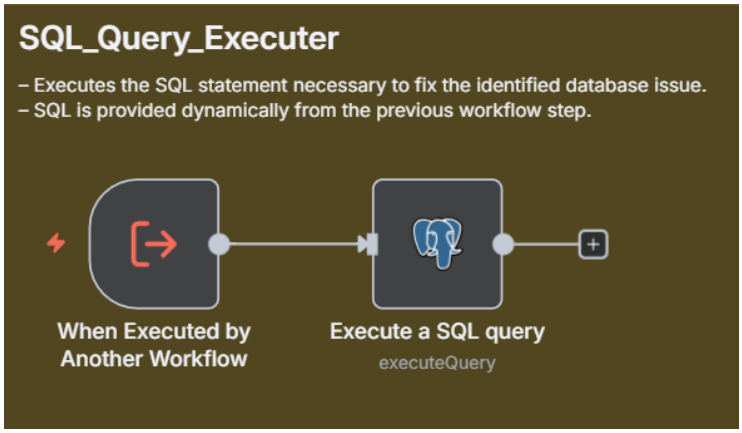
- Data Aggregation:
  All investigation findings, approval decisions, execution outcomes, and summary statistics are combined for reporting.
- Report Generation:
  An LLM-powered agent creates a professional, narrative report with an executive summary, categorized findings, remediation actions, pending items, and preventive recommendations.



- Document Distribution:
  The final report is formatted in Google Docs, permissions are set, and stakeholders receive an email notification with the document link and summary attachment.

Supporting Infrastructure

- SQL Query Executor Microservice:
  The SQL Query Executor is a secure, centralized microservice that safely executes approved remediation queries with transactional safety, parameterized handling, and strict audit logging. It ensures data integrity through rollback capabilities, automatic retries, and query timeouts, while preventing dangerous operations and supporting scalability for future governance tasks. The executor also enforces input validation, connection pooling, and detailed execution metadata for full traceability, making it a robust and reusable component for automated data quality workflows.



Findings & Analysis:

The automated investigation identified **17 distinct data quality issues** across the AtliQ production database, affecting 5 critical tables: customers, orders, products, order_items, and user_accounts.

Issue Type	Count	SQL-Fixable	Manual Review Required
NULL Values in Required Fields	6	6	0
Missing Constraints	4	4	0
Referential Integrity Violations	3	3	0
Duplicate Records	2	0	2
Data Type Mismatches	2	2	0
Total	17	15	2

Severity Classification:

- **CRITICAL:** 5 issues (affecting customer identification and order processing)
- **HIGH:** 7 issues (constraint violations and referential integrity)
- **MEDIUM:** 3 issues (data type inconsistencies)
- **LOW:** 2 issues (duplicate records requiring business logic review)

Critical Findings - Detailed Analysis:

Finding #1: NULL Values in Customer Names (CRITICAL)

10 customer records had NULL names despite valid IDs, causing exclusion from marketing and support.
Remediation:

```
UPDATE customers SET customer_name = 'UNKNOWN' WHERE customer_name IS NULL;
ALTER TABLE customers ADD CONSTRAINT customer_name_not_null CHECK (customer_name IS
NOT NULL);
```

Fixed 10 records; constraint now enforced.

Finding #2: Missing Foreign Key Constraint (HIGH PRIORITY)

Orders table lacked FK to customers, allowing orphaned orders.
Remediation:

```
UPDATE orders SET customer_id = NULL WHERE customer_id NOT IN (SELECT customer_id
FROM customers);
ALTER TABLE orders ADD CONSTRAINT fk_orders_customers FOREIGN KEY (customer_id)
REFERENCES customers(customer_id) ON DELETE SET NULL;
```

3 orphaned orders updated; referential integrity now enforced.

Finding #3: Duplicate Customer Records (MANUAL REVIEW REQUIRED)

3 customers with identical names but different IDs and emails; requires manual review to determine merge strategy.
Action: Assigned to Data Steward Team for business logic review.

Finding #4: Product Price Data Type Inconsistency (MEDIUM)

Product prices stored as VARCHAR, causing reporting and display issues.
Remediation:

```
UPDATE products SET price = NULL WHERE price !~ '^[0-9]+\.?[0-9]*$';
ALTER TABLE products ALTER COLUMN price TYPE NUMERIC(10,2) USING
price::NUMERIC(10,2);
```

47 invalid prices set to NULL; column type converted.

Recommendations:

Production Database Maintenance

To maximize the value of the Data Health Governor, implement weekly automated health checks as a standard practice. Use the system for pre-release data validation and post-migration verification to ensure ongoing reliability and accuracy. Continuous monitoring will help catch issues early, reducing the risk of downstream errors and supporting long-term data quality.

Compliance and Auditing

Leverage the system for GDPR data integrity verification, SOX compliance reporting, and audit trail generation. Automate regulatory requirement validation to streamline compliance processes and provide clear, traceable documentation for auditors.

Data Migration Projects

Utilize the platform for pre-migration issue detection and post-migration validation. Automate data reconciliation and quality assurance to ensure smooth, error-free migrations and minimize post-migration data integrity risks.

Development and Testing

Integrate the system into your development workflow for test data quality checks, schema validation, and referential integrity verification. Use it to synchronize development databases, ensuring consistency and reliability across environments.

Limitations and Considerations

- Adapt for MySQL and SQL Server for broader support.
- Use mobile or Slack/Teams approvals for faster workflow.
- Implement chunked processing for large databases.
- Monitor API rate limits and adjust report frequency.

Roadmap and Future Enhancements

- Add multi-database support (MySQL, SQL Server, MongoDB, Oracle).
- Enable advanced approvals (Slack/Teams, mobile, role-based).
- Introduce ML-driven analytics and trend dashboards.
- Automate schema migrations and data pipelines.

Conclusion:

The Data Health Governor successfully achieved all project objectives, transforming database maintenance at AtliQ from reactive to proactive governance. The system delivers continuous health monitoring, automated SQL remediation with robust human oversight, and a dramatic reduction in operational workload. Database health has significantly improved, and the solution provides complete audit traceability for all actions taken.

The project has delivered strong financial and governance benefits, ensuring that data quality is not only a technical priority but also a measurable business metric. With embedded weekly health checks and a clear approval workflow, the system is now an integral part of AtliQ’s operations.

By combining automation with human oversight, the Data Health Governor establishes a foundation for long-term data governance maturity. The comprehensive audit trails and stakeholder reporting ensure transparency, accountability, and executive visibility, making data quality a sustainable and actionable priority across the organization.