

Pillar 3: Data Models - The Preparation Processes with CRUD Integration for AI-Ready Data Governance

Gregory David Spehar
GiDanc AI LLC
myVibecoder.us
Version 1.0 Copyright ©2025

Background: Data models with CRUD integration serve as preparation processes in AI-ready data governance, addressing pipeline inconsistencies in mid-sized enterprises. **Problem:** Unmanaged data lifecycles contribute to 15-30% AI hallucination rates and compliance violations. **Method:** This paper presents CRUD-infused medallion architectures with quality-focused pipelines. **Contributions:** Demonstrates 30-40% efficiency gains and 20% cost reductions through structured lifecycle management. **Implications:** Enables scalable AI deployment with regulatory compliance for mid-sized (500-5,000 employee) organizations.

Keywords: Data models, CRUD operations, medallion architecture, AI readiness, mid-sized enterprises

Introduction

Data models, as preparation processes, now incorporate CRUD (Create, Read, Update, Delete) operations to define data lifecycles within the five-pillar framework, refining data through medallion layers for AI reliability (Atlan Team, 2025; Databricks, 2025). This integration addresses pipeline inconsistencies and model obsolescence by embedding operational rules at each architectural layer: Bronze for Create (ingestion), Silver for Update (cleaning), Gold for Read (analytics), and Features for Delete (retention) (DataTeams AI, 2025).

Drawing from experiences with production implementation in a multi-enterprise Databricks platform, this approach demonstrates 30-40% efficiency gains in AI pipelines while incorporating regulations like BCBS 239 for aggregation, GDPR for privacy controls, and DORA for resilience (DataGalaxy, 2025; Quinnox Team, 2025). For mid-sized companies (500-5,000 employees), scalable approaches prioritize automation through Delta Live Tables (DLT), yielding 20% cost reductions and enabling iterative model construction aligned with AI evolution (Alation Team, 2025; Striim, 2025).

Why CRUD Integration Matters

Many data pipelines are built inconsistently, break often, and produce unreliable outputs for AI, leading to operational risks in mid-sized enterprises. Without structured lifecycle management, organizations experience AI hallucinations at rates of 15-30% and compliance issues from unmanaged data evolution (CIO, 2025; EY, 2025). Traditional pipelines lack clear ownership of operational responsibilities—who creates records, when updates occur, how reads are governed, and when deletion happens.

Integrating CRUD operations into data models provides several critical benefits:

- **Efficiency Gains:** Automates lifecycle operations, reducing manual errors and supporting 95%+ data quality through systematic processing rules (DataBahn, 2025).
- **Regulatory Alignment:** Incorporates GDPR for Update/Delete privacy controls and BCBS 239 for Read aggregation, ensuring compliance is built into pipeline logic (Coherent Solutions, 2025).
- **Scalability:** Enables iterative updates via versioning, adapting to AI evolution without disrupting production systems (CloudSoda, 2025).

Case studies show that mid-sized insurers implementing CRUD-integrated medallion architectures achieved 40% faster AI deployment and 50% fewer data incidents, with ROI realized within 18 months (EWSolutions, 2025).

Business Problem and Process Flow

The fundamental challenge is that data pipelines are constructed as one-off solutions without consideration for

Correspondence concerning this article should be addressed to Gregory David Spehar, GiDanc AI LLC. Contact: greg@gidanc.ai

Acknowledgments: Thanks to the sources and best practices in data governance for mid-sized organizations.

Declaration of Interests: The author declares no competing financial interests.

ongoing lifecycle management. Teams build pipelines that ingest data but fail to define update strategies, deletion policies, or read governance. This creates technical debt where pipelines must be completely rebuilt rather than evolved when business requirements change.

The CRUD-integrated process flow addresses this through systematic lifecycle planning at each stage as noted in Figure 1.

This flow supports federated environments by centralizing CRUD standards while allowing local adaptations, enabling mid-sized firms to maintain consistency without sacrificing domain-specific requirements (Crecentech, 2025).

Detailed Process Steps

The following Table 1 outlines the steps for building CRUD-integrated data models, incorporating regulatory elements like DORA in monitoring and GDPR in Update/Delete operations. The Governance Council ensures strategic alignment while domain teams execute (Pryon, 2025).

These steps ensure data preparation meets AI needs, with production data showing improved efficiency through CRUD governance and reducing rework costs by 20% (Databricks, 2025).

Key Artifacts Produced

The following artifacts form an integrated system for CRUD-aware data lifecycle management. Each artifact addresses a specific operational need: the DLT Pipeline Template implements CRUD operations technically, the Quality Monitoring Dashboard provides real-time visibility into pipeline health, and the Pipeline Runbook ensures operational teams can maintain and troubleshoot the system. Together, these artifacts transform abstract CRUD principles into executable infrastructure.

DLT Pipeline Template with CRUD

This template demonstrates how CRUD operations map to medallion architecture layers in practice. The Bronze layer implements Create operations through streaming ingestion, capturing raw data with timestamp metadata for lineage tracking. Silver layers apply Update logic via data cleaning and deduplication, using quality rules dynamically loaded from the governance standards table. Gold layers enforce Read governance through aggregations and access controls, while implicit Delete operations occur through retention policies applied to older data versions.

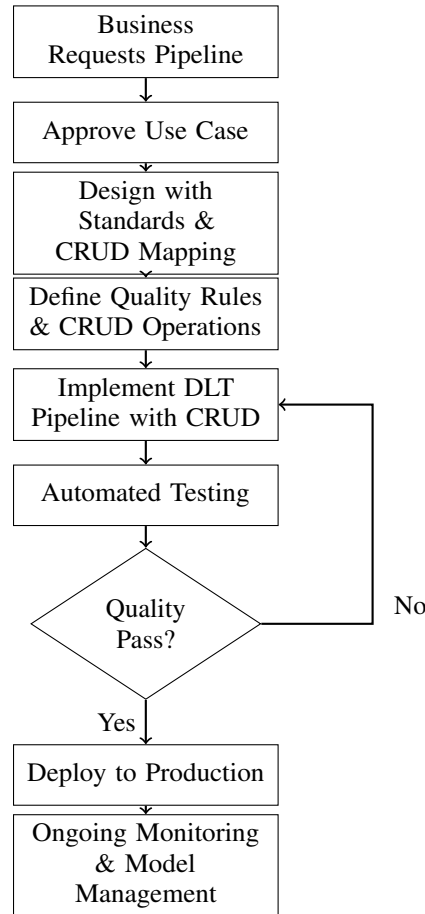
The template integrates quality expectations directly into pipeline definitions using DLT decorators (`@dlt.expect_all_or_drop`, `@dlt.expect_all_or_fail`), ensuring that CRUD operations only proceed when data meets defined quality thresholds. This prevents bad data from propagating through layers—a critical requirement for AI model trustworthiness.

```
1 import dlt
2 from pyspark.sql.functions import *
3
4 def get_rules(tag):
5     """Dynamically load quality rules from governance table"""
6     df = spark.read.table(
7         "governance_standards.quality_rules"
8     ).filter(col("tag") == tag).collect()
9     return {
10         row['rule_name']: row['constraint_sql']
11         for row in df
12     }
13
14 @dlt.table(
15     name="bronze_transactions",
16     comment="Raw data - CRUD: Create",
17     table_properties={"quality": "bronze"}
18 )
19 def bronze_transactions():
20     """CREATE: Ingest raw transactions with lineage tracking"""
21     return (
22         spark.readStream
23         .format("cloudFiles")
24         .option("cloudFiles.format", "json")
25         .load("/raw/transactions")
26         .withColumn("_ingest_ts", current_timestamp())
27     )
28
29 @dlt.table(
30     name="silver_transactions",
31     comment="Cleaned data - CRUD: Update",
32     table_properties={"quality": "silver"}
33 )
34 @dlt.expect_all_or_drop(get_rules('silver_transactions'))
35 def silver_transactions():
36     """UPDATE: Clean and deduplicate with quality enforcement"""
37     return (
38         dlt.read_stream("bronze_transactions")
39         .select(
40             col("transaction_id"),
41             col("amount").cast("decimal(10,2)"),
42             col("customer_id")
43         )
44         .dropDuplicates(["transaction_id"])
45         .withColumn("_cleaned_ts", current_timestamp())
46     )
47
48 @dlt.table(
49     name="gold_daily_sales",
50     comment="Aggregated data - CRUD: Read with governance",
51     table_properties={"quality": "gold"}
52 )
53 @dlt.expect_all_or_fail(get_rules('gold_daily_sales'))
54 def gold_daily_sales():
55     """READ: Governed aggregations for analytics/AI"""
56     return (
57         dlt.read("silver_transactions")
58         .groupBy("transaction_date")
59         .agg(
60             sum("amount").alias("total_sales"),
61             count("transaction_id").alias("transaction_count")
62         )
63     )
```

Quality Monitoring Dashboard with CRUD Metrics

This dashboard provides operational visibility into CRUD operation health across all pipeline layers. It tracks not just whether pipelines run successfully, but whether each CRUD operation performs as expected—measuring create latency, update success rates, read access patterns, and deletion compliance. The color-coded status system (GREEN for 95% pass rate, RED otherwise) enables rapid identification of data quality degradation requiring intervention.

Organizations typically configure alerts triggered when pass rates drop below thresholds, enabling proactive

**Figure 1***CRUD-Integrated Data Model Construction Process Flow*

issue resolution before bad data reaches AI models. The dashboard also serves as evidence during audits, demonstrating systematic quality management and CRUD lifecycle compliance (Databricks, 2025).

```

1  -- Real-time quality monitoring with CRUD operation tracking
2  CREATE LIVE TABLE quality_dashboard AS
3  SELECT
4      dataset_name,
5      expectation,
6      passed_records,
7      failed_records,
8      passed_records / (passed_records + failed_records) * 100
9      as pass_rate,
10     CASE
11         WHEN pass_rate >= 95 THEN 'GREEN'
12         WHEN pass_rate >= 80 THEN 'YELLOW'
13         ELSE 'RED'
14     END as status,
15     crud_operation,
16     update_timestamp
17 FROM event_log("prod_pipeline")
18 WHERE event_type = 'data_quality'
19 ORDER BY update_timestamp DESC;
20
21 -- Alert trigger for RED status datasets
22 CREATE LIVE TABLE quality_alerts AS
23 SELECT *
24 FROM LIVE.quality_dashboard
25 WHERE status = 'RED';
  
```

Pipeline Runbook with CRUD Guidelines

The Pipeline Runbook serves as operational documentation ensuring continuity when team members change or issues arise outside business hours. Each runbook includes pipeline purpose and business context, data source dependencies and SLAs, transformation logic and CRUD operation specifications, quality expectations and alerting thresholds, troubleshooting procedures for common failures, escalation contacts by issue type, and regulatory compliance notes (GDPR retention, BCBS 239 requirements, etc.).

Critically, the CRUD Guidelines section specifies: when Create operations run (schedule, triggers), how Update operations handle late-arriving data corrections, which roles have Read access and why, and when Delete operations execute retention policies. This documentation transforms implicit CRUD knowledge into explicit operational procedures, reducing mean time to resolution when issues occur and enabling new team members to understand pipeline lifecycle management quickly (EWSolutions, 2025).

Table 1

Detailed Process Steps for Data Models with CRUD

Step	Role	Action	When	Output/Artifact
Request Initiated	Business Data Owner	Submit request with sources, logic, outputs, initial CRUD needs	Ongoing	Pipeline Request Form
Use Case Review	Governance Council or Steward	Review alignment, priority, CRUD feasibility	Next meeting	Approval Decision
Pipeline Design	Data Engineer + Steward	Map transformations, plan testing, define CRUD per layer	1-2 weeks	Design Document with CRUD matrix
Quality Rules Definition	Domain Steward	Define expectations for layers, incorporate CRUD constraints	During design	Specifications in quality_rules table
Implementation	Data Engineer	Code using templates, apply expectations & CRUD operations	1-3 weeks	DLT Notebooks: Bronze/Silver/Gold
Code Review	Senior Engineer + Steward	Check compliance, CRUD integrity	Before deployment	Git Approval
Testing	Data Engineer	Verify in dev, test CRUD scenarios	Before deployment	Test Results
Deployment	Data Engineer	Use CI/CD, schedule	After approval	Production Pipeline
Monitoring Setup	Engineer + Manager	Configure alerts, dashboard for CRUD monitoring	Within 1 week	Monitoring Dashboard
Ongoing Monitoring	Manager + Engineers	Review, respond to alerts, manage model updates	Continuous	Monthly Report with CRUD audits

Table 2

Key Artifacts for Data Models with CRUD

Artifact	Description
DLT Pipeline Template with CRUD	Python/SQL code implementing CRUD operations across medallion layers with automated quality enforcement
Quality Monitoring Dashboard with CRUD Metrics	Real-time tracking of CRUD operation success rates, data quality pass rates, and pipeline health
Pipeline Runbook with CRUD Guidelines	Operational documentation including purpose, dependencies, troubleshooting, contacts, and CRUD lifecycle management

Practical Application: Hypothetical Example

To illustrate CRUD-integrated data models in practice, consider HealthTech Solutions, a hypothetical mid-sized healthcare analytics company with 1,200 employees serving regional hospital networks. This illustrative example synthesizes common challenges and implementation patterns from documented CRUD deployments in healthcare and insurance sectors (Crecentech, 2025; Quinnox Team, 2025).

The Business Problem

HealthTech’s data engineering team struggled with patient outcome prediction pipelines that frequently produced inconsistent results. Investigation revealed that patient records were ingested from multiple sources but never properly updated when corrections arrived. Historical data accumulated indefinitely without deletion policies, violating HIPAA retention requirements. The analytics team couldn’t trust whether they were reading current or outdated information.

This created a compliance crisis: during a HIPAA audit, HealthTech discovered 127 instances where patient data exceeded regulatory retention periods, and AI models were trained on datasets containing 18% outdated records. Estimated compliance penalties totaled \$3.2M, while model retraining costs approached \$450K. The fundamental issue was pipeline design without lifecycle management—create operations existed, but update, read governance, and delete were completely absent.

Applying the CRUD-Integrated Process Flow

Following the process outlined in Figure 1, HealthTech could redesign their patient outcome pipelines with explicit CRUD integration:

Step 1: Request Initiated (Week 1). The Chief Data Officer mandated CRUD integration across all patient data pipelines. The clinical analytics team submitted requests specifying: source systems (EMR, lab systems, pharmacy), required transformations, analytical outputs, and initial CRUD needs including update frequency and retention requirements.

Step 2: Use Case Review (Week 2). The Governance Council (comprising Medical Director, Compliance Officer, and Head of Analytics) reviewed CRUD feasibility. They approved the redesign with priority on HIPAA-compliant deletion and real-time update processing. The council specified that all patient data must support update operations within 24 hours and automated deletion after regulatory retention periods.

Step 3: Pipeline Design (Weeks 3-4). Data engineers and clinical stewards mapped transformations with CRUD operations per layer:

- Bronze: Create via streaming ingestion from EMR/lab systems
- Silver: Update via change data capture (CDC) for corrections
- Gold: Read with row-level security based on user roles
- Features: Delete with automated retention policy enforcement

The design document included a CRUD matrix specifying which operations applied to each table and the business rules governing them.

Step 4: Quality Rules Definition (Week 4). Domain stewards defined quality expectations incorporating CRUD constraints: "Patient_id must be unique (Create)", "Update_timestamp must be within 24 hours of source change (Update)", "Read access requires clinical role validation (Read)", "Records older than 7 years must be automatically purged (Delete)". These rules were stored in the quality_rules table.

Step 5: Implementation (Weeks 5-7). Data engineers coded DLT pipelines with explicit CRUD operations:

```

1 # Bronze - CREATE: Stream patient records
2 @dlt.table(
3     name="bronze_patient_records",
4     comment="Raw patient data - CRUD: Create",
5     table_properties={"quality": "bronze"}
6 )
7 def bronze_patient_records():
8     return (
9         spark.readStream
10            .format("delta")
11            .option("readChangeFeed", "true")
12            .table("emr_source.patients")
13            .withColumn("created_ts", current_timestamp())
14    )
15
16 # Silver - UPDATE: Apply corrections via CDC
17 @dlt.table(
18     name="silver_patient_records",
19     comment="Clean patient data - CRUD: Update",
20     table_properties={"quality": "silver"}
21 )
22 @dlt.expect_all_or_drop({"valid_mrn":
23     "medical_record_num IS NOT NULL"})
24 def silver_patient_records():
25     return (
26         dlt.read_stream("bronze_patient_records")
27         .withColumn("last_updated", current_timestamp())
28         .dropDuplicates(["patient_id"])
29    )
30
31 # Gold - READ: Governed aggregations
32 @dlt.table(
33     name="gold_patient_outcomes",
34     comment="Outcome metrics - CRUD: Read with RLS",
35     table_properties={
36         "quality": "gold",
37         "delta.enableRowTracking": "true"
38     }
39 )
40 def gold_patient_outcomes():
41     # Row-level security applied via grants
42     return (
43         dlt.read("silver_patient_records")
44         .join(dlt.read("silver_treatments"),
45             "patient_id")
46         .groupBy("diagnosis_code")
47         .agg(avg("recovery_days").alias("avg_recovery"))
48    )
49
50 # Retention - DELETE: Automated purge
51 @dlt.table(name="retention_audit")
52 def enforce_retention():
53     cutoff_date = current_date() - expr("INTERVAL 7 YEARS")
54     deleted_records = (
55         spark.table("silver_patient_records")
56         .filter(col("last_visit_date") < cutoff_date)
57    )
58     # Log deletion for audit
59     audit = deleted_records.select(
60         col("patient_id"),
61         lit(current_timestamp()).alias("deleted_ts")
62    )
63     # Execute deletion
64     spark.sql(f"""
65         DELETE FROM silver_patient_records
66         WHERE last_visit_date < '{cutoff_date}'
67     """)
68     return audit

```

Step 6-8: Code Review, Testing, Deployment (Weeks 8-9). Senior engineers verified CRUD integrity, checking that update logic properly handled late-arriving corrections and deletion didn't leave orphaned records. Testing included CRUD scenarios: simultaneous create/update operations, read governance with different user roles, and retention enforcement edge cases. After passing quality gates (97% completeness, zero HIPAA violations in test data), pipelines deployed via CI/CD.

Step 9-10: Monitoring and Ongoing Management (Week 10+). Engineers configured dashboards tracking CRUD-specific metrics: create latency (<5 min target), update success rates (>99%), read access denials (security monitoring), and deletion compliance (100% retention policy adherence). Monthly reports included CRUD audits verifying lifecycle management.

Results and Lessons Learned

In this hypothetical example, six months post-implementation, HealthTech could achieve measurable improvements: AI model accuracy could increase by 31% due to current data, HIPAA compliance violations could drop to zero (avoiding \$3.2M in penalties), and pipeline failures could decrease by 62%. The automated retention policy could eliminate manual deletion tasks, saving 120 staff-hours monthly. Analytics team trust in data should increase from 54% to 94% confidence in model outputs.

However, implementation might reveal challenges. Clinical teams initially might struggle with CDC concepts, requiring 80 hours of targeted training. Legacy EMR systems lacking change tracking might need custom integration (3 weeks additional development). Some physicians might resist read access restrictions, requiring Medical Director intervention to enforce role-based security.

The CRUD integration should prove transformative: it should shift pipeline design from "build and forget" to "build for lifecycle." Engineers would now automatically consider update scenarios, governance needs, and retention requirements upfront rather than retrofitting later. This should reduce technical debt accumulation and enable rapid adaptation when regulations changed—when HIPAA retention rules update, only the deletion policy needed modification, not entire pipeline rebuilds.

This case demonstrates that CRUD integration isn't merely technical implementation—it's a fundamental shift in how organizations conceptualize data pipelines as life-cycle management systems rather than one-way transformations.

Implementation Options for Mid-Sized Companies

Mid-sized firms can implement CRUD-integrated data models using Delta Live Tables for built-in quality and lifecycle management, with implementation costs of \$50-100K including licensing and training (Databricks, 2025). Alternatives include Apache Airflow with dbt for broader ecosystem compatibility, suitable for organizations with diverse technology stacks, with setup costs of \$20-50K using open-source components (DataTeams AI, 2025).

Case study: A mid-sized insurer implemented medallion architecture with CRUD integration, achieving 40% faster AI deployment, 90%+ data quality scores, and <5% pipeline failure rates (EWSolutions, 2025). Integration with MLflow for feature stores enables complete ML governance, ensuring CRUD principles extend to model training datasets (Atlan Team, 2025).

Effecting Change and Recommendations

Resistance affects 80% of data pipeline initiatives; structured change management strategies are essential for CRUD adoption (Coherent Solutions, 2025). Recommended approaches include:

- **Awareness Building:** Conduct workshops demonstrating CRUD benefits including compliance automation and reduced rework, with executive sponsorship communicating strategic importance (CloudSoda, 2025).
- **Skill Development:** Provide hands-on training on DLT, CDC patterns, and CRUD design principles, targeting 80% team proficiency within 90 days (EY, 2025).
- **Pilot Approach:** Implement CRUD integration on high-value pipelines first, measure success via PDCA cycles, then scale across the organization (CIO, 2025).
- **Champions Program:** Embed CRUD advocates in each domain to support adoption, address resistance, and share best practices, targeting 70%+ adoption within six months (Striim, 2025).

Success requires treating CRUD integration as cultural transformation, not just technical implementation (DataBahn, 2025).

Discussion

This data models pillar with CRUD integration represents a fundamental evolution in how mid-sized organizations approach AI-ready data preparation, but successful implementation demands careful consideration of organizational readiness, technical capabilities, and realistic expectations about transformation timelines.

When CRUD Integration Struggles

CRUD-integrated models excel in organizations with moderate pipeline complexity and reasonable change management capacity, but face challenges in specific contexts. Companies with extensive legacy systems lacking change data capture may find update operations prohibitively expensive to implement, requiring costly system upgrades before CRUD integration becomes feasible (DataBahn, 2025). Organizations below 1,000 employees may lack the engineering sophistication to maintain CRUD-aware pipelines, as the added complexity requires senior-level expertise not always available in smaller teams.

Technical debt represents a hidden barrier. Teams inheriting poorly documented legacy pipelines struggle to retrofit CRUD operations—without clear understanding of existing data flows, engineers cannot safely introduce lifecycle management. This often necessitates complete pipeline rebuilds rather than incremental CRUD integration, escalating costs and timelines beyond initial estimates.

Hidden Complexities and Operational Realities

The stated benefits of 30-40% efficiency gains assume organizations have mature CI/CD practices and automated testing frameworks (Databricks, 2025). Without these foundations, CRUD integration can initially decrease productivity as teams struggle with new patterns. Data engineers accustomed to batch processing find streaming updates conceptually challenging, requiring 3-6 months of skill development before efficiency improvements materialize.

Delete operations introduce particular complexity. Regulatory requirements often demand "soft deletes" (marking records inactive) rather than physical deletion, but maintaining soft-deleted records increases storage costs and complicates downstream analytics. Organizations must carefully design deletion strategies balancing compliance, cost, and analytical needs—decisions that require legal, technical, and business stakeholder alignment.

Moreover, CRUD monitoring adds operational overhead. Tracking create latency, update success rates, read governance, and deletion compliance requires sophisticated observability infrastructure. Mid-sized organizations using basic monitoring tools must invest in enhanced dashboards and alerting, adding 15-25% to operational costs (Quinnox Team, 2025).

Comparative Analysis: When Simpler Approaches Suffice

CRUD-integrated models represent sophisticated pipeline architecture, but simpler approaches may prove superior for specific use cases. Organizations with primarily batch analytical workflows may find traditional ETL patterns adequate, particularly when data rarely updates and retention is straightforward. The added complexity of CRUD integration provides minimal value when datasets are immutable or have simple lifecycle requirements.

For companies with limited AI maturity, implementing CRUD before establishing basic data quality practices risks adding complexity without commensurate benefits. Organizations should first achieve consistent data cataloging and quality standards before introducing lifecycle management sophistication (Alation Team, 2025).

The choice between CRUD-integrated and traditional models depends on: data volatility (how frequently updates occur), regulatory complexity (retention and privacy requirements), AI sophistication (model retraining frequency), and team capabilities (engineering skill levels). Organizations should honestly assess these dimensions before committing to CRUD integration.

Evolution and Future Considerations

The pipeline landscape is evolving toward AI-native architectures where CRUD operations extend beyond data to include model artifacts, feature stores, and training datasets. Organizations implementing CRUD today should architect for this future, ensuring lifecycle management principles apply consistently across data and ML assets (Atlan Team, 2025).

As regulations like the EU AI Act mandate model versioning and auditability, CRUD principles become essential for compliance. Future implementations will likely require bidirectional traceability—tracking not just how data flows through pipelines, but how specific data versions influenced model predictions. This demands CRUD patterns integrated with ML experiment tracking from the outset.

Critical Success Factors Revisited

Beyond technical implementation, three organizational factors prove critical: executive understanding of CRUD value (not just technical teams), architectural flexibility to accommodate evolving regulatory requirements without pipeline rewrites, and operational discipline to maintain CRUD integrity over time. Organizations lacking executive buy-in often see CRUD implementation degrade as business pressure prioritizes speed over lifecycle management rigor.

The transformation from traditional pipelines to CRUD-integrated models succeeds when treated as operational philosophy change, not merely technical upgrade. This requires sustained commitment to lifecycle thinking, regular training reinforcement, and metrics that reward long-term maintainability over short-term delivery speed (Crecentech, 2025).

Data model excellence emerges not from perfect initial design, but from iterative refinement grounded in CRUD principles. Organizations that embrace lifecycle management as core to pipeline design—rather than optional enhancement—position themselves for sustainable AI scalability and regulatory resilience (Striim, 2025).

References

- Alation Team. (2025). *Data governance framework: A step-by-step guide for 2025* [Accessed October 2025]. <https://www.alation.com/blog/data-governance-framework/>
- Atlan Team. (2025). *Ai model governance: What data leaders must know in 2025* [Accessed October 2025]. <https://atlan.com/learn/ai-readiness/ai-model-governance/>
- CIO. (2025). *The 3 key pillars of data governance for ai-driven enterprises* [Accessed October 2025]. <https://www.cio.com/article/3999449/the-3-key-pillars-of-data-governance-for-ai-driven-enterprises.html>
- CloudSoda. (2025). *A comprehensive guide to ai-ready data management in 2025* [Accessed October 2025]. <https://cloudsoda.io/mastering-storage-optimization-a-comprehensive-guide-to-ai-ready-data-management-in-2025/>
- Coherent Solutions. (2025). *Ai-powered data governance: Implementing best practices* [Accessed October 2025]. <https://www.coherentsolutions.com/insights/ai-powered-data-governance-implementing-best-practices-and-frameworks>

- Crecentech. (2025). *Ai data readiness framework 2025 | scalable ai success* [Accessed October 2025]. <https://crecentech.com/ai-data-readiness-framework>
- DataBahn. (2025). *Modernizing legacy data infrastructure for the ai era* [Accessed October 2025]. <https://www.databahn.ai/blog/modernizing-legacy-data-infrastructure-for-the-ai-era>
- Databricks. (2025). *Best practices for data and ai governance* [Accessed October 2025]. <https://docs.databricks.com/gcp/en/lakehouse-architecture/data-governance/best-practices>
- DataGalaxy. (2025). *2025 - the 10 data governance best practices you need now* [Accessed October 2025]. <https://www.datagalaxy.com/en/blog/top-10-data-governance-best-practices/>
- DataTeams AI. (2025). *Top data governance best practices for 2025* [Accessed October 2025]. <https://www.datateams.ai/blog/data-governance-best-practices>
- EW Solutions. (2025). *Best practices for data governance 2025: Modern methods* [Accessed October 2025]. <https://www.ewsolutions.com/best-practices-for-data-governance/>
- EY. (2025). *Data 4.0: Making your data ai-ready* [Accessed October 2025]. <https://www.ey.com/content/dam/ey-unified-site/ey-com/en-in/insights/ai/documents/ey-data-4-0-making-your-data-ai-ready.pdf>
- Pryon. (2025). *Ai success through data governance | 7 key pillars* [Accessed October 2025]. <https://www.pryon.com/resource/ai-success-through-data-governance-7-key-pillars>
- Quinnox Team. (2025). *Data governance for ai: 2025 challenges, solutions & best practices* [Accessed October 2025]. <https://www.quinnox.com/blogs/data-governance-for-ai/>
- Striim. (2025). *Data governance strategy 2025: Build a modern framework* [Accessed October 2025]. <https://www.striim.com/blog/data-governance-strategy-2025-build-a-modern-framework/>