. Key MDM Concepts

Understanding these core pillars helps you speak the same language as business stakeholders:

- **The Golden Record (Single Source of Truth):** The core "master" version of an entity (e.g., a specific Taxi Zone) that has been de-duplicated, cleansed, and validated.
- **Data Domains:** Logical groupings of master data. In your case: **Location**, **Partner/Vendor**, and **Product/Tariff**.
- **Data Stewardship:** The "human" side of MDM. While you build the algorithm, a **Data Steward** (e.g., a member of the TLC Operations team) is the one who manually resolves "suspect" matches that the algorithm flags.
- **Survivorship:** The automated ruleset that decides which source system "wins" when data conflicts. (e.g., "If two systems provide a Zone Name, always trust the GIS Department's version first.")

---

2. Strategic Business Alignment

MDM is expensive to build, so it must be justified by business value. For the NYC Taxi dataset, the alignment looks like this:

Business Objective	MDM Contribution	ROI Impact
Revenue Integrity	Ensuring **Rate Codes** are standard across all vendors.	Prevents fare leakage and avoids lawsuits over illegal surcharges.
Operational Reporting	Standardized **Taxi Zones** for all 10M+ monthly trips.	Allows planners to identify "dead zones" accurately to optimize driver distribution.
Vendor Accountability	Unified **Vendor Master** with contact and contract data.	Reduces time to resolve data outages from days to minutes.

Export to Sheets

---

3. The MDM Maturity Framework (Crawl, Walk, Run)

You don't need a million-dollar software suite to start. Most organizations follow this path:

1. **Registry Style (Crawl):** You keep data in original systems but use a central "Index" (your RDS table) to link them together.
2. **Consolidated Style (Walk):** You pull data from various systems into your central RDS to create the Golden Record for reporting. **(This is where your current project sits.)**
3. **Centralized/Transactional Style (Run):** Systems must check with the MDM RDS *before* they can create a new record. This prevents duplicates from ever entering the ecosystem.

---

4. Implementation Checklist for Business Buy-In

To move from "concepts" to "production," you need three non-technical components:

- **The Data Governance Council:** A monthly meeting with the Business Owners you identified earlier to approve changes to Rate Codes or Zones.
- **SLA (Service Level Agreement):** A promise to the business on data quality (e.g., "99.9% of all trips will map to a valid Master Zone ID").
- **Change Management:** A process to notify downstream analysts when a Master record changes (e.g., if a Zone ID is retired).

What is MDM and why it matters to the business?

What is MDM?

Master Data Management is a business-led program that ensures an organization's shared data (like customers, products, or taxi zones) is accurate, consistent, and available across the entire enterprise.

It is the process of taking data from different "silos" (different apps or departments), cleaning it, and creating one **Golden Record** that everyone agrees is the truth.

The "Before and After" MDM

- **Before MDM:** You ask three different departments "Which taxi zone earned the most revenue?" and you get **three different answers** because each department uses a different version of the "Zone List."
- **After MDM:** Every department pulls from the same **Golden Record** table in RDS. There is only one answer.

Why It Matters to the Business

Businesses don't invest in MDM just to have "clean tables"—they do it because "dirty data" has a high financial and operational cost.

1. The "One Version of the Truth" (Decision Making)

If the **Rate Code** master data is inconsistent, a CEO might see a report saying airport trips are declining, while the Finance team sees them increasing. MDM eliminates these "data brawls" in meetings.

- **Business Impact:** Faster, more confident strategic decisions.

2. Operational Efficiency (Cost Savings)

Imagine if a **Vendor** changes their name or contact info. Without MDM, an admin might have to update that info in 10 different systems manually. With MDM, you update it once, and it "pushes" to everywhere else.

- **Business Impact:** Reduced manual labor and "data repair" costs.

3. Regulatory Compliance & Risk

In the NYC Taxi world, the TLC is a regulator. If the **Taxi Zone** data is wrong, the city might collect the wrong amount of "Congestion Surcharge" tax. This leads to audits, fines, and bad PR.

- **Business Impact:** Lower legal risk and guaranteed compliance with city laws.

4. Improved Customer/Passenger Experience

If the master data for **Payment Types** is messy, a passenger might be charged twice or have a "Dispute" recorded incorrectly.

- **Business Impact:** Higher trust in the service and fewer support tickets.

The "Business Justification" Summary

If you were pitching MDM to the TLC Commissioner, you would say:

"MDM is the foundation that ensures every dollar we track, every zone we analyze, and every vendor we regulate is based on the same high-quality information. **We aren't just managing data; we are managing the integrity of the city's transportation policy.**"

Master data domains: Customer, Product, Location, Vendor

For each domain: Who cares? What decisions depend on it? What breaks if it's wrong?

1. Customer Domain

In the NYC Taxi context, this usually refers to "Credit Card Hash/ID" or "Corporate Accounts."

- **Who cares?** Marketing, Customer Loyalty teams, and the Finance Department.
- **Decisions depending on it:** Whom do we offer discounts to? How do we identify "Power Users"? What is the Customer Lifetime Value (CLV)?
- **What breaks if it's wrong?**
    - **Privacy Violations:** You accidentally link Trip A to the wrong Person B.
    - **Customer Friction:** A regular rider doesn't get their "10th ride free" perk because their ID was duplicated as two different people.

---

2. Product Domain

*In the Taxi dataset, the "Product" is the **Service Type** (Yellow Taxi, Green Taxi, For-Hire-Vehicle) or the **Rate Code** (Standard, Airport, Negotiation).*

- **Who cares?** Product Managers, Policy Makers, and Pricing Analysts.
- **Decisions depending on it:** Do we need to increase the price of JFK trips? Is the "Green Taxi" program failing to serve the outer boroughs?
- **What breaks if it's wrong?**
    - **Financial Leakage:** You undercharge for a premium service because the "Product" was mapped incorrectly to a cheaper rate.
    - **Strategy Failure:** The city invests in Green Taxis based on data that was actually mixed with Yellow Taxi data.

---

3. Location Domain

*This is your **Taxi Zones** (Boroughs, Neighborhoods, Service Zones).*

- **Who cares?** City Planners, Dispatchers, and Drivers.
- **Decisions depending on it:** Where should we build new taxi stands? Where is the "Congestion Fee" applied? Which areas are "underserved"?
- **What breaks if it's wrong?**

    - **Legal/Tax Risk:** A driver is charged a "Manhattan Congestion Fee" for a drop-off that actually happened in Brooklyn.
    - **Operational Chaos:** Drivers are sent to the wrong "Hot Zone" because the heatmap logic is based on faulty location master data.

---

4. Vendor Domain

*These are the **Technology Providers** (Verifone, CMT) and **Fleet Owners**.*

- **Who cares?** Procurement, IT Operations, and Legal/Licensing.
- **Decisions depending on it:** Which vendor has the most technical downtime? Which contract should we renew? Which vendor is failing to report data accurately?
- **What breaks if it's wrong?**
    - **Audit Failure:** You cannot prove to a regulator which company was responsible for a data gap.
    - **Payment Delays:** The city sends a payout to "Vendor A" when it should have gone to "Vendor B" due to a name change that wasn't updated in the master list.

Summary Table for Stakeholders

Domain	Primary Stakeholder	Worst Case Scenario
Customer	Marketing/CRM	Legal Lawsuits & Bad Brand Rep
Product	Finance/Policy	Revenue Loss & Wrong Pricing
Location	City Planning	Traffic Gridlock & Unfair Taxing
Vendor	Procurement/IT	Contract Breaches & Data Gaps

MDM implementation styles with real-world scenarios:

Here are the four standard MDM implementation styles applied to the NYC Taxi (TLC) ecosystem:

---

1. Registry Style

Concept: The "Index" approach. You don't move the data from the silos. Instead, you create a central list of IDs that links records together.

- **Real-World Scenario:** The TLC keeps a central registry that says "Vendor ID 1" in the Finance system is the same as "Vendor ID 1" in the GPS tracking system.

- **Best for:** Large organizations where departments refuse to change their local systems.
- **The "Golden Record":** It doesn't actually exist in one place; it is "assembled" on the fly when needed.

2. Consolidated Style

Concept: The "Reporting Hub." Data is pulled from various silos into a central RDS (like your project) to create a single Golden Record for analytics.[1]

- **Real-World Scenario:** You pull "Taxi Zone" names from the GIS team and "Zone Surcharge Rates" from the Legal team into one Master Table. Analysts use this table for their monthly reports instead of the raw silo data.
- **Best for:** Business Intelligence (BI) and reporting.
- **The "Golden Record":** Lives in your central RDS, but changes made there **do not** flow back to the source systems.

---

3. Coexistence Style

Concept: The "Bi-Directional" approach. The Golden Record is created in the central hub, but any updates made to the Golden Record are pushed back to the original silos.[2]

- **Real-World Scenario:** A **Vendor** changes their official business address. You update it in the MDM Hub. The Hub then automatically updates the Licensing Department's database AND the Payment Department's database.
- **Best for:** Maintaining consistency across the whole company while letting departments keep their own software.
- **The "Golden Record":** Lives in the hub and acts as the "Master Copy" that synchronizes everyone else.

---

4. Centralized (Transaction) Style

Concept: The "Dictatorship" approach. Data is created, stored, and managed **only** in the MDM hub. If a department wants to add a new "Rate Code," they must do it directly in the MDM system.

- **Real-World Scenario:** The TLC creates a new **Rate Code 7** (e.g., "Holiday Surcharge"). No vendor is allowed to use it until it is first defined and published in the Central Master Table. The vendors then "subscribe" to this table to get the update.
- **Best for:** High-security and highly regulated data where mistakes are unacceptable.
- **The "Golden Record":** The **only** record that exists.

Comparison of Styles

Style	Difficulty	Data Consistency	Best Use Case
Registry	Low	Low (Lookup only)	Quick wins with limited budget.
Consolidated	Medium	Medium (Reporting)	**Your current NYC Taxi project.**
Coexistence	High	High (Synchronized)	Enterprise-wide digital transformation.
Centralized	Very High	Total (Single Source)	Critical financial or legal definitions.

MDM Governance Model:

Governance defines the rules, but the **People** make it work. Here is how those roles and responsibilities break down for your specific taxi domains.

---

1. The MDM Governance Matrix

Role	Responsibility	For the NYC Taxi Project...
Data Owner	**Accountable.** Defines the business rules and takes the "blame" if data is legally wrong.	**TLC Commissioner / Head of Policy.** (e.g., Decides that the JFK rate is exactly $70).
Data Steward	**Operational.** Manages the "Golden Record." Resolves duplicates and approves new Rate Codes.	**TLC Business Analyst.** (e.g., Manually checks if "Zone 264" is valid before you add it to RDS).
Data Custodian	**Technical.** You! You build the pipelines, the RDS tables, and the matching algorithms.	**Data Engineer.** (e.g., Writes the SQL and Python that moves data from silos to the hub).
Data Consumer	**Usage.** Queries the Golden Records to build insights or train models.	**Data Scientists / Traffic Planners.** (e.g., Person building the "Taxi Demand Prediction" model).

---

2. Responsibilities in Action: The "New Zone" Scenario

Imagine a new taxi zone is created. Here is how your roles interact:

1. **Data Owner:** Signs a legal document authorizing a new neighborhood zone for congestion pricing.

2. **Data Steward:** Enters the new Zone Name and Borough into a "Staging" table and double-checks for typos.
3. **Data Custodian (You):** Runs the ETL script that promotes that record into the **Golden Record** table and assigns it a version_id. You ensure the database index is updated for performance.

Data Consumer: Refreshes their PowerBI dashboard to show the new zone's trip volume.

MDM Operating Model:

1. How are Golden Records Created?

In the NYC Taxi project, you use a **Hybrid Approach** combining automation with human oversight.

- **Step A: Automated Match (The First Filter):** Your Custodian (Data Engineer) runs a script in the RDS. It uses "Fuzzy Matching" (like Levenshtein Distance) to find records that look similar (e.g., "CMT" and "Creative Mobile Tech").
- **Step B: The Stewardship Queue:** Any records with a similarity score between **0.80 and 0.95** are sent to a "Task List" for the Data Steward.
- **Step C: Manual Enrichment:** For new records that don't exist yet (like a brand new taxi vendor), an **Application Form** is used. The Vendor must submit their official legal name, address, and license number to the Steward before they are added to the MDM.

---

2. Who Approves Changes?

Approval is tiered based on the **impact** of the change:

Type of Change	Approval Authority	Logic
Simple Correction	**Data Steward**	Fixishing a typo (e.g., "Queens" instead of "Queeens").
New Master Record	**Data Steward + Rules**	Adding a new Taxi Zone that passes all automated validation checks.
Policy/Financial Change	**Data Owner (VP)**	Changing the price of a **Rate Code** or retiring a **Vendor**.
Systemic Rule Change	**Governance Council**	Deciding to change the survivorship rule for an entire domain.

---

3. What's the Escalation Path for Conflicts?

Conflicts usually happen when two "Silos" disagree and the Data Steward isn't sure which one to trust.

1. **Level 1 (The Steward):** Attempts to resolve using documentation (e.g., checking the official TLC website).
2. **Level 2 (The Custodian):** If the conflict is technical (e.g., data is corrupt in the source), the Custodian investigates the ETL pipeline.
3. **Level 3 (The Data Owner):** If it's a "Business Stand-off" (e.g., Finance says a zone is in Brooklyn, but GIS says it's in Queens), the **Data Owner** makes the final executive decision. Their word becomes the new "Golden Rule."

---

4. How are Exceptions Handled?

Exceptions are records that "break the rules" (e.g., a trip record arrives with `RateCodeID = 99`, which doesn't exist in your Master Table).

- **The "Holding Pen" (Staging):** The record is blocked from entering the Golden Record table. It stays in a **Staging Area** so it doesn't "pollute" your clean data.
- **Notification:** An automated alert (email/Slack) is sent to the **Data Steward**.
- **Resolution Options:**
    - **Override:** The Steward corrects the data.
    - **Ignore:** The Steward allows the record but flags it as "Unverified."
    - **Update Master:** If the exception is actually a new valid code, the Steward creates a new Golden Record to support it.

Summary of the Operating Model

The goal of this model is to ensure that you, the **Data Custodian**, never have to guess. If the data is messy, you have a process to follow, a Steward to ask, and an Owner to back you up.

Match and merge strategies with governance implications

In MDM, **Match and Merge** is the "engine" that builds the Golden Record. However, it isn't just a technical task; it requires a **Governance Model** to decide what happens when the computer is unsure.

Here is how the technical strategies tie into the human responsibilities.

---

1. The Matching Strategies

Matching is the process of identifying that two records from different silos represent the same thing (e.g., the same Taxi Vendor).

- **Deterministic Matching (Exact Match):**
  - **Strategy:** The system looks for an exact match on a unique identifier, like a **License Number** or **Tax ID**.
  - **Governance Implication:** Low effort. The **Data Owner** sets a rule: "If the License IDs match, they are the same." You (the **Custodian**) automate this.
- **Probabilistic Matching (Fuzzy Match):**
  - **Strategy:** The system compares names and addresses (e.g., "Verifone Inc" vs. "Verifone NY") and calculates a similarity score (e.g., 85%).
  - **Governance Implication:** High effort. The **Data Steward** must define the "Threshold."
    - Above 95% = Auto-merge.
    - 70% to 95% = Send to Steward for manual review.
    - Below 70% = Keep separate.

---

2. The Merging Strategies (Survivorship)

Once you know two records are the same, you must decide which data "survives" to become the Golden Record.

- **Source Reliability (Trust Scores):**
  - **Strategy:** You rank your silos. For example: "Trust the **GIS Silo** for Zone Names, but trust the **Finance Silo** for Rate Prices."
  - **Governance Implication:** The **Data Owner** must sign off on these rankings. If the GIS data is wrong, the GIS team is accountable.
- **Most Recent (Recency):**
  - **Strategy:** The system always picks the record with the latest `Last_Updated_Timestamp`.
  - **Governance Implication:** Dangerous for master data. A newer "messy" record might overwrite an older "clean" record. The **Data Steward** must monitor for "data decay."
- **Completeness:**
  - **Strategy:** The system picks the record that has the fewest `NULL` values.

---

3. Governance Roles in the Match/Merge Flow

Stage	Responsibility	Role Involved
Defining Rules	Deciding that a 90% score is "good enough" for an auto-merge.	**Data Owner**
Configuring Rules	Writing the SQL/Python logic for Levenshtein distance or Trust scores.	**Data Custodian (You)**
Resolving "Suspects"	Manually reviewing the 80% matches that the system flagged.	**Data Steward**
Audit & Quality	Checking the Golden Table to ensure no "false matches" occurred.	**Data Consumer / Steward**

---

4. Handling "False Positives" (The Escalation Path)

What happens if your ETL merges two different vendors because they have similar names?

1. **Identification:** A **Data Consumer** (Analyst) notices a weird jump in revenue for one vendor.
2. **Reporting:** They alert the **Data Steward**.
3. **Correction:** The Steward uses an "Unmerge" tool (or asks you to run a script) to split the records.
4. **Prevention:** The **Data Custodian** adjusts the matching threshold or adds a new "Negative Match" rule to prevent it from happening again during the next ETL run.

Summary for your Project

In your NYC Taxi project, you will likely use **Deterministic matching** for IDs and **Source Reliability survivorship** (trusting the TLC's official lookup tables over the trip record headers).

Data quality dimensions:

In Master Data Management (MDM), these six dimensions are the "Health Metrics" of your data. As the **Data Custodian**, you use these to prove to the **Data Steward** that the **Golden Records** you've built are actually high-quality.

Here is how each dimension applies to your **NYC Taxi Project**:

---

1. Accuracy (Is it "Correct"?)

Accuracy measures how closely the data represents the real-world event.

- **The Taxi Test:** Does the trip_distance in your database match the actual miles the taxi drove?
- **The Problem:** A GPS glitch says a trip was 500 miles, but it only took 10 minutes. This is **inaccurate**.

2. Completeness (Is anything "Missing"?)

Completeness checks if all required data points are present.

- **The Taxi Test:** Does every single record have a pickup_location and a dropoff_location?
- **The Problem:** A trip record is missing the passenger_count. You can't perform total-volume analysis because the data is **incomplete**.

3. Consistency (Does it "Match" elsewhere?)

Consistency ensures that data across different systems (or even within the same database) does not contradict itself.

- **The Taxi Test:** If the Total_Amount is $20, do the Fare, Tip, and Tolls columns add up to exactly $20?
- **The Problem:** One table says the Vendor is "Verifone" and another says "VTS." This is **inconsistent**.

---

4. Timeliness (Is it "Up to Date"?)

Timeliness measures how "fresh" the data is for its intended use.

- **The Taxi Test:** Are yesterday's trip records available for analysis this morning?
- **The Problem:** If you are trying to manage traffic congestion *right now*, but your data is 3 days old, it is no longer **timely**.

5. Validity (Does it follow the "Rules"?)

Validity checks if the data follows the specific format, type, or range defined by the business.

- **The Taxi Test:** Is the RateCodeID a number between 1 and 6?
- **The Problem:** A record comes in with a RateCodeID of "99" or "Airport." Since these aren't in the official TLC list, the record is **invalid**.

6. Uniqueness (Are there "Duplicates"?)

Uniqueness ensures that no piece of data is recorded more than once.

- **The Taxi Test:** Does each trip have a unique Trip_ID?
- **The Problem:** You see the exact same trip (same time, same driver, same car) listed twice. If you sum the revenue, you will double-count the money because the data is not **unique**.

Comparison for MDM

Dimension	Primary Role Responsible	MDM Goal
Accuracy	Data Steward	Ensure the "Golden Record" is the truth.
Completeness	Data Custodian (You)	Ensure ETL captured all required fields.
Consistency	Data Steward	Resolve conflicts between Silos.
Timeliness	Data Custodian (You)	Optimize the ETL pipeline speed.
Validity	Data Custodian (You)	Use "Check Constraints" in RDS.
Uniqueness	Data Custodian (You)	Run Deduplication/Matching scripts.

Summary

In your **Consolidation** project, you will use these dimensions to create a **Data Quality Scorecard**. Before you allow data to move from the "Staging" table to the "Master" table, it must pass these six tests.

MDM components with governance lens:

When we look at **MDM components** through a **governance lens**, we stop seeing them as just "software parts" and start seeing them as the "legal and operational guardrails" of the company.

Here is how these five components function to keep your NYC Taxi data under control:

---

1. Repository (The "Safe")

This is the physical storage (like your **Amazon RDS**) where the Master Data lives.

- **Governance Lens:** It's not just a database; it's the **Authoritative Source**.
- **Responsibility:** The **Data Custodian** ensures it is encrypted and backed up. The **Data Owner** decides who has the "key" (access permissions) to see the data inside.
- **Taxi Example:** The table that stores the "Golden" list of all licensed Taxi Vendors.

2. Integration Layer (The "Messenger")

This is the plumbing that moves data in and out using **Publish/Subscribe** (Pub/Sub) or APIs.

- **Governance Lens:** It ensures **Consistency**. When a change happens in the MDM, the integration layer "publishes" that change so all other systems "subscribe" to the update.
- **Taxi Example:** As soon as a Vendor's license is revoked in the MDM, the Integration Layer sends an alert to the Payment System to stop processing their transactions.

---

3. Quality Engine (The "Validator")

This is where your **Data Quality Dimensions** (Accuracy, Validity, etc.) are turned into code.

- **Governance Lens:** These rules are **Governance Artifacts**. They are the digital version of the laws set by the Data Owner.
- **Taxi Example:** A rule that says: *"If a trip lasts 1 minute but costs $500, flag it as 'Suspect' and do not add it to the Golden Record."*

4. Workflow Engine (The "Process")

This manages the human steps, like **Approvals**.

- **Governance Lens:** It enforces **Accountability**. It ensures that no single person can change a "Golden Record" without someone else checking it.
- **Taxi Example:** If you (the Custodian) try to add a new Taxi Zone, the Workflow Engine automatically emails the **Data Steward** to click "Approve" before the change goes live.

---

5. Audit & Lineage (The "History Book")

Audit tracks *who* changed what. **Lineage** tracks *where* the data came from.

- **Governance Lens:** This is for **Regulatory Compliance**. If a regulator (like the NYC Auditor) asks, "Why was this Vendor paid $1M?", you can show exactly which silo provided the data and who approved it.
- **Taxi Example:** A report showing that Vendor "Verifone" had its name changed by "User_John" on Jan 5th, based on data imported from "Licensing_Silo_v2."

Summary Table: Component vs. Governance Goal

Component	Governance Goal	What it prevents
Repository	Security	Unauthorised access to sensitive driver data.
Integration	Synchronization	Different departments using different versions of the truth.
Quality Engine	Integrity	"Garbage In, Garbage Out."
Workflow	Control	Rogue changes to Master Data without business approval.
Audit/Lineage	Transparency	Not knowing how a mistake got into the system.

Hands -on:

- Identify master data domains in NYC taxi dataset (zones, vendors, rate codes)
- For each domain, document:
- Business owner (simulated)
- Business justification (why is this master data?)
- Update frequency and change triggers
- Quality requirements and thresholds
- Approval workflow (who can create/update/retire?)

1. Domain: Location (Taxi Zones)

The geographical boundaries used to calculate fares and track demand.

- **Business Owner (Simulated):** Director of GIS(Geographic Information System) & City Planning.
- **Business Justification:** This is Master Data because it is the **common denominator** for all reporting. Without a "Golden" list of zones, Finance cannot apply congestion surcharges, and Operations cannot identify underserved neighborhoods.
- **Update Frequency & Change Triggers:**
  - **Frequency:** Rare (Annual or bi-annual).
  - **Triggers:** New terminal openings (e.g., JFK expansion), rezoning of city neighborhoods, or changes to the Congestion Pricing Zone.
- **Quality Requirements & Thresholds:**
  - **Validity:** Every Zone ID must map to one of the 5 Boroughs.
  - **Uniqueness:** No overlapping polygons; no duplicate Zone IDs.
  - **Threshold:** 100% accuracy required for IDs to prevent billing errors.
- **Approval Workflow:** * **Create/Update:** GIS Dept submits a new Shapefile $\rightarrow$ **Data Steward** validates names $\rightarrow$ **Data Owner** signs off.
  - **Retire:** Only possible if a zone is legally removed from city maps.

---

2. Domain: Vendor (Technology Providers)

The companies (like Verifone and CMT) authorized to collect and transmit trip data.

- **Business Owner (Simulated):** Chief Technology & Innovation Officer.
- **Business Justification:** This is Master Data because it regulates **Data Lineage**. We must know exactly which legal entity is responsible for a batch of trip records to handle audits or technical downtime.
- **Update Frequency & Change Triggers:**
  - **Frequency:** Low (Ad-hoc).
  - **Triggers:** New vendor contract signed, a vendor name change/merger, or a license revocation.
- **Quality Requirements & Thresholds:**
  - **Completeness:** Must have a valid tax ID, contact email, and active license status.
  - **Threshold:** 100% completeness for the "Active" status field.
- **Approval Workflow:**
  - **Create:** Procurement uploads contract $\rightarrow$ **Data Steward** verifies credentials.
  - **Retire: Data Owner** must move a vendor to "Inactive" if their license expires; the **Data Custodian** (you) then flags the ID in the RDS to block new data.

---

3. Domain: Product (Rate Codes)

The pricing logic applied to trips (Standard, JFK, Newark, etc.).

- **Business Owner (Simulated):** Deputy Commissioner of Policy & Finance.
- **Business Justification:** This is Master Data because it controls **Revenue**. If "Rate Code 2" (JFK) is defined differently in two silos, the city faces massive financial discrepancies and passenger complaints.
- **Update Frequency & Change Triggers:**
  - **Frequency:** Very Low (Every 2–5 years).
  - **Triggers:** Legal changes to fare structures or new "Flat Rate" agreements (e.g., LaGuardia flat rate).
- **Quality Requirements & Thresholds:**
  - **Consistency:** Rate Code definitions must match the public-facing passenger "Bill of Rights."
  - **Threshold:** Zero tolerance for "Duplicate" rate IDs.
- **Approval Workflow:**
  - **Update:** TLC Board votes on fare hike $\rightarrow$ **Data Owner** defines the new rule $\rightarrow$ **Data Steward** updates descriptions $\rightarrow$ **Data Custodian** (you) updates the lookup table in RDS.

Summary Table for your Project Documentation

Domain	Change Trigger	Who Approves?	Failure Impact
Location	City Rezoning	GIS Director	Wrong Tax/Surcharge
Vendor	New Contract	CTO	Audit Failure
Rate Code	Fare Hike	Policy VP	Financial Loss