Expert Dossier: Group 2

Topic: Data Quality, Metadata & Lineage

1. Deconstructing "Quality": The Theoretical Framework

In casual conversation, "Data Quality" (DQ) is often treated as a synonym for "cleanliness." In a Data Engineering context, this definition is insufficient. Data that is perfectly clean (syntactically correct) can still be useless if it arrives too late or describes the wrong time period.

We define Data Quality using the **Juran Principle**: *"Fitness for Use."* Data is of high quality if it satisfies the requirements of its intended consumers (Decision Makers, Data Scientists, Operational Processes).

The Cost of Poor Data Quality

The impact of bad data is often illustrated by the **1-10-100 Rule**:

- **$1:** The cost to verify a record as it is entered (Prevention).
- **$10:** The cost to cleanse the record later in an ETL pipeline (Correction).
- **$100:** The cost of making a bad business decision based on that record (Failure).

The Wang & Strong Framework (1996)

To engineer quality, we must categorize it. Richard Wang and Diane Strong (MIT) developed the standard hierarchical framework for DQ:

1. **Intrinsic DQ:** Does the data have quality in its own right? (Accuracy, Objectivity).
2. **Contextual DQ:** Is the data applicable to the task? (Timeliness, Completeness).
3. **Representational DQ:** Is the data clearly described? (Interpretability, Consistency).
4. **Accessibility DQ:** Can the user retrieve it? (Security, Access).

2. The 6 Dimensions of Data Quality

To operationalize these theories, Data Engineers measure data against **six specific dimensions**. These are the standard metrics used to define "Dirty Data."

1. Accuracy

Definition: The degree to which data correctly describes the "real world" object or event.

- *Example:* A customer’s address in the database is "Berlin," but they actually live in "Munich."
- *Detection:* Hard to detect technically. Requires comparison with a "Source of Truth" (e.g., verifying against a Postal Service API).

2. Completeness

Definition: The proportion of stored data against the potential of "100% complete."

- *Metric:* Null Rate (Percentage of missing values).
- *Nuance:* Not all missing data is bad. A "Middle Name" field might be legitimately empty. Engineers must distinguish between "Missing but Optional" and "Missing and Critical."

3. Consistency

Definition: The absence of difference, when comparing two or more representations of a thing against a definition.

- *Cross-System Consistency:* The "Date of Birth" for Customer X is the same in the CRM and the Billing System.
- *Internal Consistency:* The sum of `Line_Item_Totals` must equal `Order_Total`.

4. Timeliness

Definition: The degree to which data represents reality from the required point in time.

- *Metric:* **Data Latency**. If a dashboard is updated nightly, the latency is 24 hours. For a Fraud Detection algorithm, 24-hour latency renders the data useless (Low Quality).

5. Uniqueness

Definition: No entity exists more than once within the dataset.

- *Failure Mode:* **Duplication**. Customer "J. Smith" and "John Smith" are the same person but created as two rows.
- *Impact:* Counting these rows leads to inflated metrics (e.g., overstating the customer base).

6. Validity

Definition: Data conforms to the syntax (format, type, range) of its definition.

- *Format:* Dates must be `YYYY-MM-DD`.
- *Range:* `Age` cannot be negative. `Temperature` (Kelvin) cannot be below 0.
- *Set:* `Country_Code` must be in the ISO-3166 list (e.g., "DE" is valid, "GER" is invalid).

3. The Engineering Process: Profiling & Cleansing

Quality is not an accident; it is an engineered outcome. The lifecycle of Data Quality consists of three phases: **Profiling** (Diagnosis), **Cleansing** (Treatment), and **Monitoring** (Prevention).

Phase A: Data Profiling (The "MRI Scan")

Before building an ETL pipeline, engineers must understand the raw data. **Data Profiling** is the statistical analysis of a dataset to detect quality issues.

- **Column Statistics:** Min, Max, Mean, Null Count. (e.g., "Why is the Max Age 199?")
- **Frequency Distributions:** Histograms showing the most common values. (e.g., "Why do 40% of our customers live in 'Test City'?")
- **Pattern Matching:** Using Regular Expressions (Regex) to validate formats (e.g., "Do all phone numbers start with +?")

Advanced Profiling: Benford's Law

For financial data, engineers often test against **Benford's Law**, which predicts the frequency of leading digits in naturally occurring numerical datasets (e.g., the number 1 appears as the leading digit ~30% of the time). Deviations from this distribution often indicate **Fraud** or **Artificial Data Generation**.

Phase B: Data Cleansing

Once issues are identified, we implement automated cleansing logic in the pipeline.

1. **Parsing:** Decomposing unstructured text.
   - Input: `"Mr. John Smith III"` → Output: `{Title: Mr, First: John, Last: Smith, Suffix: III}`.
2. **Standardization:** Mapping synonyms to a canonical value.
   - Input: `"Deutschland", "Germany", "BRD"` → Output: `"DE"`.
3. **Deduplication (Entity Resolution):** Identifying non-identical duplicates.
   - Technique: **Fuzzy Matching**. We calculate the **Levenshtein Distance** (the number of edits required to change one string into another). If "Jon Smith" and "John Smith" have a distance of 1, we merge them.

4. Metadata Management: The "Map"

If Data is the "Asset," Metadata is the "Label" that tells you what the asset is. Without metadata, a Data Lake is just a **Data Swamp**—a dumping ground of files that no one dares to use.

We categorize metadata into three buckets:

1. Technical Metadata (For the Engineer)

Describes the structure and storage.

- Table Names, Column Names, Data Types (`INT`, `VARCHAR`).
- Partition Keys, Indexes, Compression Formats.
- Source System location (e.g., `s3://bucket/raw/logs/`).

2. Business Metadata (For the User)

Describes the meaning and context.

- **Definitions:** "Net Revenue = (Gross Sales - Returns - Tax)."
- **Ownership:** "Data Steward: Sarah Jones."
- **Quality Score:** "This table is Certified Gold (99% Accurate)."

3. Operational Metadata (For the Ops Team)

Describes the history of the process.

- "Last Updated: Today at 03:00 AM."
- "Row Count: 1,000,000 (+5% since yesterday)."
- "Job Status: Success."

The Data Catalog

In modern enterprises, metadata is managed in a **Data Catalog** (e.g., Alation, Collibra, Databricks Unity Catalog). The Catalog crawls the database, extracts Technical Metadata, and provides a UI for Stewards to add Business Metadata. It serves as the "Google Search" for the company's data.

5. Data Lineage: The Trust Graph

The single most common question executives ask is: *"Where did this number come from?"* **Data Lineage** is the visual representation of the data's journey through the pipeline. It maps the dependencies between Sources, Transformations, and Dashboards.

Technique: Modern tools parse the SQL logs to automatically generate a **Directed Acyclic Graph (DAG)**.

- `Node A (Source Table)` $\rightarrow$ `Edge (SQL Join)` $\rightarrow$ `Node B (Target Table)`.

Use Case 1: Root Cause Analysis (Upstream)

- **Incident:** The CEO's dashboard shows $0 Sales for yesterday.
- **Action:** The engineer looks at the Lineage Graph for the "Dashboard Widget." They trace the line backwards (Upstream) to find the broken dependency.
- **Result:** "The error originated in the `currency_conversion` table, which failed to update."

Use Case 2: Impact Analysis (Downstream)

- **Scenario:** The ERP team plans to rename the column `customer_id` to `cust_uuid`.
- **Action:** They check the Downstream Lineage.
- **Result:** "Stop! If you rename this, you will break 15 Marketing Reports and the Churn Prediction Model."

6. The Wagner Case Study (Group 2)

The Quality & Trust Crisis: You are the **Data Quality Task Force**. You have been called in to solve two specific problems that are paralyzing Wagner Technologies.

Conflict A: The "Dirty" Excel Files (Quality Audit)

The Situation: The Production Department refuses to use the ERP. They record Quality Control (QC) results in manual Excel files. The Data Engineering team needs to ingest this data, but it is a mess.

The Data Sample: *(Rows from `QC_Log_v2.xlsx`)*

ID	Date	Inspector	Result	Comments
P-101	01.12.2025	fischer	OK	
P-102	Dec 1st	SF	OK	
P-103	2025/12/01	S. Fischer	Not OK	Scratch on surface
P-104		Fischer		(Test skipped)
P-101	01.12.2025	fischer	OK	(Duplicate entry)

Task 1: The Audit

1. **Identify Dimensions:** Analyze the rows above. List **4 specific Data Quality violations** and map each one to a **Dimension** (Accuracy, Completeness, Consistency, Timeliness, Uniqueness, Validity).
    - *Example:* "The date format varies" → *Consistency/Validity Violation*.
2. **Design the Clean-Up:**
    - **Standardization:** How do you fix the "Inspector" column? (Input: `fischer`, `SF`, `S. Fischer` → Output: `?`).
    - **Validity:** How do you fix the "Result" column? (Input: `Not OK` → Output: `FALSE`).

Conflict B: The CFO's Blockade (Lineage)

The Situation: Robert (CFO) has refused to sign the Annual Report.

- **The Issue:** The Data Warehouse reports **€15.2M** Net Revenue. His manual Excel calculation shows **€15.4M**.
- **The Demand:** He demands to see the *exact* logic. *"Did you include the returns from the UK branch? Did you use the exchange rate from the transaction day or the end-of-month rate?"*
- **The Barrier:** The logic is buried in a complex SQL script (`JOIN table_a ON... WHERE...`) that Robert cannot read.

Task 2: The Trust Solution

1. **The Metadata Solution:** You need to tag the `Revenue` column in the Data Catalog so Robert trusts it. Define the **Business Metadata** you would write for this column (Definition, Owner, Calculation Rule).
2. **The Lineage Solution:** Sketch a **Data Lineage Graph** that would answer Robert's questions.
    - Start: `Raw Orders (ERP)` & `Exchange Rates (API)`.
    - Middle: `Transformation (Logic)`.
    - End: `Net Revenue Report`.
    - *Show where the "Filter: UK Returns" and "Exchange Rate Join" would appear in this graph.*