**Data Cleaning and Normalization Report for Railway Dataset**

**Project Overview:**  
This report outlines the cleaning, transformation, and normalization processes applied to the railway dataset. The goal was to improve data quality, reduce redundancy, and set up a robust model for in-depth analysis using Power BI and Power Query.

**1. Dataset Overview**

The original dataset consisted of a single table with the following columns:

* **Transaction ID:** Unique identifier for each train ticket purchase.
* **Date of Purchase & Time of Purchase:** When the ticket was purchased.
* **Purchase Type:** Online or Station Counter.
* **Payment Method:** e.g., Contactless, Credit Card, Debit Card.
* **Railcard:** Indicates if a passenger is a National Railcard holder (Adult, Senior, Disabled) or not (None).
* **Ticket Class:** Standard or First Class.
* **Ticket Type:** Advance, Off-Peak, or Anytime (with corresponding discount rules).
* **Price:** Final cost after discounts.
* **Departure Station & Arrival Destination:** Station names (12 and 32 categories, respectively).
* **Date of Journey, Departure Time, Scheduled Arrival Time & Actual Arrival Time:** Journey schedule details.
* **Journey Status:** On Time, Delayed, or Cancelled.
* **Reason for Delay:** Technical issues, weather, etc.
* **Refund Request:** Yes/No indicating if a refund was requested.

**2. Data Cleaning Steps**

**A. Data Type Conversion and Consistency**

* **Dates & Times:** Converted date and time fields (e.g., Date of Purchase, Date of Journey, Departure/Arrival Times) to the appropriate data types.
* **Categorical Fields:** Ensured text fields (e.g., Payment Method, Purchase Type, Railcard) were consistently formatted and free of typos.

**B. Handling Duplicates and Missing Data**

* **Duplicates:** Applied “Remove Duplicates” on key columns for lookup tables (e.g., Payment Method, Railcard) to extract unique values.
* **Missing Values:** Reviewed columns such as Delay Reason and Refund Request to handle blanks or inconsistencies as per business rules.

**3. Normalization & Transformation Process**

The normalization process was carried out in Power Query using the following steps:

**A. Creating a Master Query and References**

* **Master Query:** Loaded the full dataset into Power Query.
* **Reference Queries:** Created multiple reference queries from the master query for different dimensions (e.g., Payment Method, Railcard, Journey).

**B. Building Dimension (Lookup) Tables**

1. **Purchase\_Type Table:**
   * Extracted unique values from the “Purchase Type” column.
   * Removed duplicates.
   * Added an index column to create Purchase\_Type\_ID.
2. **Payment\_Method Table:**
   * Extracted unique values from the “Payment Method” column.
   * Added an index column to create Payment\_Method\_ID.
3. **Railcard Table:**
   * Extracted unique values from the “Railcard” column.
   * Added an index column to create Railcard\_ID.
   * **Additional Column:** Added “Railcard Discount” to record the discount rate (e.g., 33% for holders and 0% for None).
4. **Ticket\_Class Table:**
   * Created a lookup from “Ticket Class” with its own index (Ticket\_Class\_ID).
5. **Ticket\_Type Table:**
   * Extracted distinct ticket types (Advance, Off-Peak, Anytime) and included discount information.
   * Added an index column for Ticket\_Type\_ID.
6. **Stations Table:**
   * Combined data from “Departure Station” and “Arrival Destination.”
   * Removed duplicates.
   * Added an index column to create Station\_ID.
7. **Journey\_Status Table & Delay\_Reason Table:**
   * Extracted unique values from “Journey Status” and “Reason for Delay” fields.
   * Added respective index columns.

**C. Creating the Fact Tables**

1. **Transactions Table (Fact Table)**
   * Kept all purchase-specific columns.
   * Removed the original textual values for dimensions.
   * **Merge Queries:** For each dimension (e.g., Payment Method, Railcard), merged with the corresponding lookup table to bring in the foreign key (e.g., Payment\_Method\_ID, Railcard\_ID).
   * **Refund Request:** Kept in the Transactions Table since it is directly linked to the ticket purchase.
2. **Journey Table**
   * Contains journey-specific details (Date of Journey, Departure/Arrival Stations, Scheduled and Actual Times, Journey Status, Delay Reason).
   * Added a **Journey\_ID** as an index column.
   * **Merge Queries:** In the Transactions Table, replaced detailed journey columns with a single Journey\_ID foreign key pointing to this Journey Table.

**D. Ensuring Consistency Across Tables**

* **Indexing:** Both the Journey and Transaction tables were sorted by a common field (e.g., Transaction ID) before adding the index, ensuring that the Journey\_ID aligns correctly between them.
* **Relationships:** Planned model relationships in Power BI Model View:
  + Transactions[Journey\_ID] linked to Journey[Journey\_ID].
  + Other dimension relationships set similarly (e.g., Payment\_Method\_ID, Railcard\_ID).

**4. Final Schema Overview**

**Transactions Table**

* **Columns:** Transaction\_ID, Date of Purchase, Time of Purchase, Purchase\_Type\_ID, Payment\_Method\_ID, Railcard\_ID, Ticket\_Class\_ID, Ticket\_Type\_ID, Price, Journey\_ID, Refund\_Request

**Journey Table**

* **Columns:** Journey\_ID, Date of Journey, Departure\_Station\_ID, Arrival\_Station\_ID, Scheduled Departure Time, Scheduled Arrival Time, Actual Arrival Time, Journey\_Status\_ID, Delay\_Reason\_ID

**Dimension Tables**

* **Purchase\_Type Table:** Purchase\_Type\_ID, Purchase\_Type
* **Payment\_Method Table:** Payment\_Method\_ID, Payment\_Method
* **Railcard Table:** Railcard\_ID, Railcard\_Type, Railcard\_Discount
* **Ticket\_Class Table:** Ticket\_Class\_ID, Ticket\_Class
* **Ticket\_Type Table:** Ticket\_Type\_ID, Ticket\_Type, Discount
* **Stations Table:** Station\_ID, Station\_Name
* **Journey\_Status Table:** Journey\_Status\_ID, Status
* **Delay\_Reason Table:** Delay\_Reason\_ID, Reason

**5. Summary and Recommendations**

* **Data Integrity:** The normalization process has minimized redundancy and improved data consistency across the dataset.
* **Scalability:** By separating dimensions and fact tables, the model is better positioned to handle growth and more complex queries.
* **Analysis Flexibility:** The clear relationships between tables (e.g., linking Refund Requests in Transactions to Delay Reasons in Journey) allow for more detailed and accurate reporting.
* **Future Enhancements:** If further details (e.g., refund amounts or additional refund attributes) are needed, a dedicated Refunds Table can be created.