This notebook performs initial data loading and cleaning for a healthcare fraud detection dataset. Here's a breakdown of the key steps:

1. **Data Loading:**
   * It imports necessary libraries, primarily pandas for data manipulation.
   * It defines a base path to the raw data files stored in Google Drive.
   * It loads several CSV files into pandas DataFrames:
     + Train.csv
     + Test.csv
     + Train\_Beneficiarydata.csv
     + Test\_Beneficiarydata.csv
     + Train\_Inpatientdata.csv
     + Test\_Inpatientdata.csv
     + Train\_Outpatientdata.csv
     + Test\_Outpatientdata.csv
2. **Initial Data Inspection (using column\_sample\_report\_transposed):**
   * A custom function column\_sample\_report\_transposed is defined to display a transposed summary of each DataFrame. This report shows the column names, their data types, and a few sample unique values.
   * This function is then called for the training dataframes (train\_inp, train\_out, train\_bene, train) to get an initial understanding of the data structure and content.
3. **Date Conversion:**
   * It identifies date-related columns in the inpatient, outpatient, and beneficiary DataFrames.
   * It converts these columns to datetime objects using pd.to\_datetime, with errors='coerce' to turn any unparseable dates into NaT (Not a Time).
   * It then prints the data types of these columns to confirm the conversion.
4. **Categorical Encoding:**
   * The PotentialFraud column in the train DataFrame is converted from categorical ('Yes', 'No') to numerical (1, 0) using the .map() function. It then prints the value counts of the new numerical column.
   * The RenalDiseaseIndicator column in the train\_bene DataFrame is converted from categorical ('Y', '0') to numerical (1, 0) using the .map() function. It then prints the value counts, including NaN values if any exist.
5. **Handling Missing Values (Imputation):**
   * **Inpatient Data:**
     + Missing values in DeductibleAmtPaid are filled with 0.
     + Missing values in DiagnosisGroupCode are filled with the string '-1'.
     + Missing values in ClmAdmitDiagnosisCode are filled with the string 'Unknown'.
   * **Outpatient Data:**
     + It checks if DeductibleAmtPaid and ClmAdmitDiagnosisCode exist in the train\_out DataFrame before attempting to fill missing values in those columns.
     + Missing values in DeductibleAmtPaid are filled with 0.
     + Missing values in ClmAdmitDiagnosisCode are filled with the string 'Unknown'.
   * It then prints the count of remaining nulls for the imputed columns in both inpatient and outpatient data to verify the imputation.
6. **Data Cleaning Summary (Markdown):**
   * A markdown cell provides a summary of the cleaning steps performed, including date conversions, categorical encoding, handling of numeric and code nulls, and the decision to keep columns with high missing values for later feature engineering.
7. **Saving Cleaned Data:**
   * The cleaned training DataFrames (train\_inp, train\_out, train\_bene, train) are saved as Parquet files in a 'processed' data directory in Google Drive. Parquet is an efficient columnar storage format.
8. **Production Data Validation Checks:**
   * A markdown cell outlines the types of validation checks typically performed in a production pipeline.
   * The following checks are then implemented in code:
     + **Critical Column Null Check:** It checks for nulls in essential columns (BeneID, ClaimID, Provider, date columns, InscClaimAmtReimbursed, DOB, Gender, Race, PotentialFraud) to ensure no critical information is missing after cleaning.
     + **Date Logic Validation:** It checks if AdmissionDt is before DischargeDt in the inpatient data, identifying any records with invalid date ranges.
     + **Value Range Validation:** It checks for negative reimbursement amounts in both inpatient and outpatient data.
     + **Referential Integrity Checks:**
       - It checks if all Provider IDs in the inpatient and outpatient claims exist in the train (provider labels) DataFrame.
       - It checks if all BeneIDs in the inpatient and outpatient claims exist in the train\_bene (beneficiary) DataFrame.
     + **Row Count Documentation:** It prints the number of rows in each training DataFrame to document that no rows were dropped during this cleaning phase.
9. **Adding HasDied Feature:**
   * A new binary column HasDied is added to the train\_bene DataFrame. This column is 1 if the DOD (Date of Death) is not null and 0 otherwise. It prints the value counts for this new column.
10. **Saving Updated Beneficiary Data:**
    * The updated train\_bene DataFrame, including the HasDied column, is saved back to a Parquet file, overwriting the previous version.
11. **Checking Cleaned Beneficiary Data Head:**
    * It prints the first few rows of the cleaned train\_bene DataFrame to visually inspect the result, specifically mentioning checking the new column.
12. **Transposed Schema Report (using transposed\_schema\_report):**
    * A second custom function transposed\_schema\_report is defined to display the schema (column name and data type) of a DataFrame in a transposed format.
    * This function is applied to all cleaned training DataFrames to provide a clear overview of the final data types after cleaning and transformations.
13. **Production Data Validation Summary (Markdown):**
    * A final markdown cell summarizes the results of the production data validation checks, confirming that critical columns are clean, date logic and value ranges are valid, referential integrity holds, row counts are consistent, the HasDied flag was added, and the final schemas were validated.

### Key Findings Summary

Here is a concise summary of the important findings and statistics from the data loading and cleaning process:

\* \*\*Potential Fraud Distribution (Train Data):\*\*

\* No: 4904  
 \* Yes: 506

\* \*\*Renal Disease Indicator Distribution (Train Beneficiary Data):\*\*

\* 0: 118978

\* 1: 19578

\* \*\*Missing Value Imputation (Train Data):\*\* All critical columns and specifically imputed columns (`DeductibleAmtPaid`, `DiagnosisGroupCode`, `ClmAdmitDiagnosisCode`) show 0 nulls remaining after cleaning.

\* \*\*Invalid Admission/Discharge Dates (Train Inpatient Data):\*\* 0 records found with invalid date ranges.

\* \*\*Negative Reimbursement Amounts (Train Data):\*\* 0 records found with negative claim reimbursement amounts.

\* \*\*Referential Integrity Checks (Train Data):\*\* All Provider and Beneficiary IDs in claim data successfully matched to their respective master tables.

\* \*\*Row Counts (Train Data):\*\* Row counts remain unchanged, indicating no rows were dropped during this cleaning phase:

\* `train\_inp`: 40474 rows

\* `train\_out`: 517737 rows

\* `train\_bene`: 138556 rows

\* `train`: 5410 rows

\* \*\*Has Died Flag Distribution (Train Beneficiary Data):\*\*

\* 0 (Did not die): 137135   
 \* 1 (Died): 1421