Lumenn AI Guides

# Data Dictionary

## Understanding Data Dictionary and creating one for your data on Lumenn AI

Version 1.0

# Table of Contents

Introduction ........................................................................................... 3

Why Data Dictionaries are Essential ............................................................ 3

Sample Data Dictionaries ........................................................................... 4

Real World Data Scenarios-Based Sample Dictionaries ..................................... 5

For a Hospital’s Patient & RCM Data ........................................................ 5

Loans & Repayments Data for Detecting Fraud ........................................... 9

For an Energy Company Data ................................................................. 14

About Lumenn ............................................................................................ 17

# Introduction

Lumenn AI is a generative BI platform where users can perform BI analysis and generate reports and data visualizations from their enterprise data using natural language prompts.

A data dictionary defines all data elements or contents of a database and the relationship between them. The metadata included in the data dictionary for the connected data assists our AI Models to understand the nature of the data and its relationships better and reduce hallucinations, biases and incorrect data analysis.

## Why Data Dictionaries are Essential

### Context is Key

AI models, even sophisticated ones, need context to understand the data they're analyzing.  A Data Dictionary provides this crucial context by:

- **Defining Terms:** Clearly explaining what each column, field, or variable represents.
- **Describing Relationships:** Outlining how different data points connect (e.g., one-to-one, one-to-many).
- **Specifying Data Types:** Indicating whether a field is numerical, categorical, textual, etc. Although not absolutely necessary as modern AI Models can autodetect and understand data types.
- **Providing Business Meaning:** Explaining the significance of each data point within the broader business context.

## Improved Accuracy

With a better understanding of the data, the AI can:

- **Generate more accurate visualizations:** Choosing the right chart types, appropriate scales, and meaningful labels.
- **Provide more insightful analysis:** Identify trends, patterns, and anomalies more effectively.
- **Reduce errors:** Minimize the risk of misinterpreting data or drawing incorrect conclusions.

## Enhanced User Experience

A well-defined Data Dictionary can:

- **Empower users:** Enable them to understand the data better and ask more informed questions.
- **Facilitate collaboration:** Improve communication and understanding among team members.
- **Streamline data exploration:** Make it easier for users to navigate and interact with the data.

On this guide we provide samples for Data Dictionaries to help you understand how to create one for your business data that you add to **Lumenn AI**

# Sample Data Dictionaries

## Real World Data Scenarios-Based Sample Dictionaries

### Sample#1

## For a Hospital’s Patient & RCM Data

Let’s hypothetically take an example where a Hosipital/Hospice Business stores its Patient Data, Encounters, Procedures, Diagnoses, Charges, Payments, Claims and Hospital AGED AR Data into multiple database tables. The dictionary for it may look like this.

**Patient Table**

| Column Name | Description |
|---|---|
| PatientID | Unique identifier for each patient. |
| FirstName | First name of the patient. |
| LastName | Last name of the patient. |
| DateOfBirth | Date of birth of the patient. |
| Gender | Gender of the patient (e.g., Male, Female, Other). |
| Address | Complete address of the patient. |
| PhoneNumber | Primary phone number of the patient. |
| Email | Email address of the patient. |
| InsuranceID | Unique identifier of the patient's insurance policy. |
| InsuranceProvider | Name of the insurance provider. |
| InsuranceGroup | Insurance group or plan name. |
| GuarantorName | Name of the guarantor (if different from patient). |
| GuarantorRelationship | Relationship of guarantor to patient (e.g., Spouse, Parent). |

## Encounters Table

| Column Name | Description |
|---|---|
| EncounterID | Unique identifier for each patient encounter. |
| PatientID | Foreign key referencing the PatientID in the Patient Demographics table. |
| EncounterDate | Date of the patient encounter. |
| EncounterType | Type of encounter (e.g., Inpatient, Outpatient, Emergency). |
| ReferringPhysician | Name of the referring physician. |
| AttendingPhysician | Name of the attending physician. |
| Location | Location of the encounter (e.g., Clinic, Hospital, Home Health). |
| Episode | Episode of care identifier (if applicable). |

## Procedures Table

| Column Name | Description |
|---|---|
| ProcedureID | Unique identifier for each procedure performed. |
| EncounterID | Foreign key referencing the EncounterID in the Encounters table. |
| CPTCode | CPT code for the procedure. |
| ProcedureDescription | Description of the procedure. |
| Units | Number of units for the procedure. |
| Modifier | Procedure modifier (if applicable). |

## Diagnosis Table

| Column Name | Description |
|---|---|
| DiagnosisID | Unique identifier for each diagnosis. |
| EncounterID | Foreign key referencing the EncounterID in the Encounters table. |
| ICD10Code | ICD-10 code for the diagnosis. |
| DiagnosisDescription | Description of the diagnosis. |

## Charges Table

| Column Name | Description |
|---|---|
| ChargeID | Unique identifier for each charge. |
| EncounterID | Foreign key referencing the EncounterID in the Encounters table. |
| ProcedureID | Foreign key referencing the ProcedureID in the Procedures table. |
| ChargeAmount | Amount charged for the procedure. |
| ChargeDate | Date the charge was generated. |
| PatientResponsibility | Amount of the charge the patient is responsible for. |
| **ChargeDueDate** | Date the charge is due for payment. |

## Payments Table

| Column Name | Description |
|---|---|
| PaymentID | Unique identifier for each payment received. |
| EncounterID | Foreign key referencing the EncounterID in the Encounters table. |
| Payer | Name of the payer (e.g., Insurance company, Patient). |
| PaymentAmount | Amount of the payment received. |
| PaymentDate | Date the payment was received. |
| PaymentType | Type of payment (e.g., Check, Credit Card, Cash). |

## Claims Table

| Column Name | Description |
|---|---|
| ClaimID | Unique identifier for each claim submitted. |
| EncounterID | Foreign key referencing the EncounterID in the Encounters table. |
| ClaimStatus | Status of the claim (e.g., Pending, Submitted, Paid, Denied). |
| SubmissionDate | Date the claim was submitted. |
| ClaimAmount | Total amount of the claim. |
| PayerResponseDate | Date of the payer's response. |
| PayerResponse | Description of the payer's response. |

| | |
|---|---|
| Eligibility | Eligibility verification status (e.g., Verified, Pending, Not Verified). |

## Aged AR table

| Column Name | Description |
|---|---|
| AR_Bucket | Age of the outstanding account (e.g., 0-30 days, 31-60 days, 61-90 days, 90+ days, 120+ days). |
| Total_Amount | Total amount of outstanding accounts within the age bucket. |

## Relationships

### One-to-many:

- Encounters to Patients
- Procedures to Encounters
- Diagnoses to Encounters
- Charges to Encounters
- Charges to Procedures
- Payments to Encounters
- Claims to Encounters

## Calculating AR and Aged AR

### Accounts Receivable (AR):

- Total Charges: Sum of all charges for a specific period (e.g., month, quarter, year).
- Total Payments: Sum of all payments received within the same period.
- AR = Total Charges - Total Payments

### Aged AR:

### Categorize Charges by Age:

- Current: Charges with a due date within the current period (e.g., within the last 30 days).
- 0-30 Days: Charges past due by 0-30 days.
- 31-60 Days: Charges past due by 31-60 days.
- 61-90 Days: Charges past due by 61-90 days.
- 90+ Days: Charges past due by more than 90 days.
- Calculate Total Amount for Each Age Bucket: Sum the outstanding balance of charges within each age category.

**Notes:**

- **Charge Due Date:** The **ChargeDueDate** field is crucial for accurate AR aging calculations.
- **Episode:** Tracks related healthcare services delivered across time, improving care coordination and billing accuracy.
- **Patient Responsibility:** Captures the out-of-pocket costs the patient is liable for, aiding in patient billing and collections.
- **Eligibility:** Monitors the verification status of patient insurance coverage, reducing claim denials.
- **Hospital Aged AR:** Tracks the aging of outstanding accounts receivable, enabling proactive revenue cycle management and identifying potential bottlenecks.

### Sample#2

## Loans & Repayments Data for Detecting Fraud

Let’s hypothetically take an example where there are 2 CSV files:

1. Loan.csv
2. Payments.csv

They are used for Loan & Payment relationship for customers of a Bank/Financial Institution. You can describe the data dictionary on a text document as below:

This dataset contains loan application, loans disbursed and payments data. Loan status that indicates the outcome and is categorized into two groups: normal loans (value 0) and fraudulent loans (value 1).

Normal loans include statuses like Paid Off Loan, Charged Off Paid Off, and Settlement Paid Off. Fraudulent loans include Rejected, Internal Collection, and Charged Off. Other statuses are excluded from the classification process.

In loans.csv there are 18 columns:

1. loanId: This is a unique loan identifier. Use this for joins with the payment.csv file
2. anon_ssn: This is a hash based on a client's SSN (Anonymous ssn). You can use this as
if it is a SSN to compare if a loan belongs to a previous customer.
3. payFrequency: This column represents repayment frequency of the loan:

- B is biweekly payments
- I is irregular
- M is monthly
- S is semi monthly
- W is weekly

4. apr: Annual Percentage Rate of the loan (%)
5. applicationDate: Date of application (start date)
6. originated: Indicates if the loan has been initiated (underwriting process started).
7. originatedDate: Date of origination, day the loan was originated
8. nPaidOff: Number of MoneyLion loans previously paid off by the client.
9. approved: Indicates if the loan has been approved (final step of underwriting).
10. isFunded: Whether or not a loan is ultimately funded. a loan can be voided by a
customer shortly after it is approved, so not all approved loans are ultimately funded.
11. loanStatus: Current loan status (this column is used for prediction). Most are self-explanatory. Below are the statuses which need clarification:

- Withdrawn Application: The applicant has withdrawn their loan application
before it was approved or funded.• Paid Off Loan: The loan has been fully paid off by the borrower according to the
repayment terms.
- Rejected: The loan application was rejected, typically due to failure to meet
underwriting criteria.
- New Loan: A newly approved loan that has not yet been funded.
- Internal Collection: The loan is being managed and collected internally by
MoneyLion due to missed payments or delinquency.
- CSR Voided New Loan: A new loan application was voided by a customer
service representative (CSR) before funding.
- External Collection: The loan has been transferred to an external collection

agency for management and collection.

- Returned Item: A payment on the loan has been returned due to insufficient funds in the borrower's account.
- Customer Voided New Loan: The borrower voided a new loan application before funding.
- Credit Return Void: The loan was voided due to a credit return, typically related to a refunded transaction.
- Pending Paid Off: The loan is in the process of being paid off, but the process is pending completion.
- Charged Off Paid Off: The loan has been charged off as a loss by MoneyLion but has also been paid off by the borrower.
- Settled Bankruptcy: The loan has been settled as part of a bankruptcy proceeding.
- Settlement Paid Off: The loan has been paid off through a settlement agreement.
- Charged Off: The loan has been charged off as a loss by MoneyLion due to non-payment.
- Pending Rescind: The loan is pending rescission, meaning it may be canceled or reversed.
- Customver Voided New Loan: Typo: Likely should be "Customer Voided New Loan". Similar to "Customer Voided New Loan", indicating the borrower voided a new loan application before funding.
- Pending Application: The loan application is pending review and approval.
- Voided New Loan: The loan application was voided before funding.• Pending Application Fee: The loan application is pending due to the application fee not being paid.
- Settlement Pending Paid Off: The loan is pending being paid off through a settlement agreement.

12. loanAmount: Principal amount of the loan ($) (for non-funded loans this will be the principal in the loan application)
13. originallyScheduledPaymentAmount: This is the Initialy scheduled repayment amount ($) (if a customer pays off all his scheduled payments, this is the amount we should receive)
14. state: State of the client
15. Lead type: The lead type determines the underwriting rules for a lead.

- bvMandatory: leads that are bought from the ping tree – required to perform bank verification before loan approval
- lead: very similar to bvMandatory, except bank verification is optional for loan approval

• california: similar to lead, but optimized for California lending rules
• organic: customers that came through the MoneyLion website
• rc_returning: customers who have at least 1 paid off loan in another loan portfolio.
(The first paid off loan is not in this data set).
• prescreen: preselected customers who have been offered a loan through direct
mail campaigns
• express: promotional “express” loans
• repeat: promotional loans offered through sms
• instant-offer: promotional “instant-offer” loans
16. Lead cost: Cost associated with acquiring the lead ($)
17. fpStatus: Result of the first payment of the loan:
• Checked: payment is successful
• Rejected: payment is unsuccessful
• Cancelled: payment is cancelled
• No Payments/No Schedule: loan is not funded
• Pending: ACH attempt has been submitted to clearing house but no response yet
• Skipped: payment has been skipped
• None: No ACH attempt has been made yet – usually because the payment is
scheduled for the future18. clarityFraudId: unique underwriting id
Every row represents an accepted loan application/ successfully funded loan.
Missing values can exist. Some fields are only implemented after the loan application was made.

In payments.csv there are 9 columns:

1. loanId: This is a unique loan identifier. Use this for joins with the loan.csv file

2. isCollection: A loan can have a custom-made collection plan if the customer has trouble making repayments as per the original schedule. TRUE means the payment is from a custom-made collection plan.

3. installmentIndex: This counts the nth payment for the loan. First payment is 1, 2nd payment is 2 and so on. This index resets for collection payment plans. So some loans can have 2 payments with the same installmentIndex. One from the regular plan and one from the collection plan.

4. Paymentdate: Effective of payment

5. Principal: principal component of the payment

*6. Fees: Fee/ interest amount of the payment*

*7. paymentAmount: Total amount of the payment. Usually equals to fees + principal*

*8. paymentStatus:*

- *Checked: payment is successful*
- *Rejected: payment is unsuccessful*
- *Cancelled: payment is cancelled*
- *Pending: ACH attempt has been submitted to clearing house but no response yet*
- *Skipped: payment has been skipped*
- *None : No ACH attempt has been made yet – usually because the payment is scheduled for the future*
- *Rejected awaiting retry: retrying a failed ACH attempt.*

*9. paymentReturnCode: these are ACH error codes to explain why the payment failed. You can find more information about this at the end of this document, or visit the following link: https://www.vericheck.com/ach-return-codes/*

*Each row in this file represents an ACH attempt (either scheduled for the future or has elapsed in the past) associated to the loan.*

***Appendix:***

*ACH return codes:*

*R01 Insufficient Funds*

*R02 Account Closed*

*R03 No Account/Unable to Locate Account*

*R04 Invalid Account Number*

*R05 Unauthorized Debit Entry*

*R06 Returned per ODFI's Request*

*R07 Authorization Revoked by Customer (adjustment entries)*

*R08 Payment Stopped or Stop Payment on Item*

*R09 Uncollected Funds*

*R10*

*Customer Advises Not Authorized; Item Is Ineligible, Notice Not Provided,*

*Signatures Not Genuine, or Item Altered (adjustment entries)*

*R11 Check Truncation Entry Return*

*R12 Branch Sold to Another DFI*

*R13 RDFI not qualified to participate*

*R14 Representative Payee Deceased or Unable to Continue in that Capacity*

*R15*

*Beneficiary or Account Holder (Other Than a Representative Payee)*

*Deceased*

*R16 Account Frozen*

*R17 File Record Edit Criteria (Specify)*

*R20 Non-Transaction Account*

*R21 Invalid Company Identification*

*R22 Invalid Individual ID Number*

*R23 Credit Entry Refused by Receiver*

*R24 Duplicate Entry*

*R29 Corporate Customer Advises Not Authorized*

*R31 Permissible Return Entry (CCD and CTX only)*

*R33 Return of XCK Entry*

### Sample#3

## For an Energy Company Data

Let’s hypothetically take an example of an Energy company that has data under 3 SQL Tables – Customers, Meter Readings, Bills and Outage. For the associated data we can have either 3 CSV or XLS files uploaded to explain the description of the columns and an additional .txt file explaining the relationships between the tables. You can also have all of them in a single DOC, DOCX or PDF File. Even better!

### Table: Customers

| Column Name | Data Type | Description |
|---|---|---|
| CustomerID | INT | Unique identifier for each customer |
| FirstName | VARCHAR(50) | First name of the customer |
| LastName | VARCHAR(50) | Last name of the customer |
| Address | VARCHAR(255) | Customer's address |
| ContactNumber | VARCHAR(20) | Customer's contact number |
| Email | VARCHAR(100) | Customer's email address |
| AccountNumber | VARCHAR(50) | Unique identifier for the customer's account |
| ServiceType | VARCHAR(50) | Type of service provided (e.g., electricity, gas, water) |
| MeterNumber | VARCHAR(50) | Unique identifier for the customer's meter |

### Table: Meter Readings

| Column Name | Data Type | Description |
|---|---|---|
| MeterReadingID | INT | Unique identifier for each meter reading |
| MeterNumber | VARCHAR(50) | Foreign key referencing the MeterNumber in the Customers table |
| ReadingDate | DATE | Date and time of the meter reading |
| Consumption | DECIMAL | Amount of energy consumed |
| Unit | VARCHAR(20) | Unit of measurement for consumption (e.g., kWh, m3) |

### Table: Bills

| Column Name | Data Type | Description |
|---|---|---|

| | | |
|---|---|---|
| BillID | INT | Unique identifier for each bill |
| AccountNumber | VARCHAR(50) | Foreign key referencing the AccountNumber in the Customers table |
| BillDate | DATE | Date the bill was generated |
| DueDate | DATE | Due date for bill payment |
| Amount | DECIMAL | Total amount of the bill |
| PaymentStatus | VARCHAR(50) | Status of the bill payment (e.g., unpaid, paid, overdue) |

**Table: Outage**

| Column Name | Data Type | Description |
|---|---|---|
| OutageID | INT | Unique identifier for each outage |
| OutageStartDate | DATETIME | Start time of the outage |
| OutageEndDate | DATETIME | End time of the outage |
| AffectedArea | VARCHAR(255) | Description of the affected area |
| OutageCause | VARCHAR(255) | Cause of the outage |

**Relationships:**

- **MeterReadings** table has a one-to-many relationship with the Customers table through the **MeterNumber** foreign key.
- **Bills** table has a one-to-one relationship with the Customers table through the **AccountNumber** foreign key.

**PS: You don't necessarily need to define the Datatypes as AI can autodetect that.**

Hope this guide helped you understand the following:

- **What is a Data Dictionary?**
- **Why Lumenn AI needs Data Dictionaries for your connected data?**
- **What are the advantages of a Data Dictionary in terms of Data Analysis and Bias & Hallucination Reduction?**
- **How to create your own Data Dictionary for your connected data on Lummen AI?**

**Feel free to reach out to us on hello@lummen.ai to ask your additional queries on Data Dictionary or if you face a challenge creating one.**

# About Lumenn AI

Breaking down technical barriers, **Lumenn AI** provides a no-code environment for non-technical users to extract and leverage enterprise data analysis. Through natural language instructions, complex data is transformed into accessible visualizations with AI-driven insights. Customizable dashboards empower any user to create, manage, and share BI assets.

To know more about Lumenn AI:

**Visit Lummen AI**

**Follow us on Linkedin**

**Write to us at hello@lumenn.ai**