**Week5 Data Dictionary for University Library Management System**

Hang Yang

SYS 5013: System Engineering Analysis

Prof. Hitam Zidan

Sep 20, 2024

**Introduction**

In this week we will create data dictionaries of library management system for the Metropolitan University platform upgrade plan. The data dictionary is a collection of domain-specific data attributes that is shared across an organization to enhance the understanding of business operations. It also serves as a valuable reference and design tool for system analysts during the analysis and design phases of new systems. To create a comprehensive data dictionary, a typical workflow would involve documenting all relevant data elements from the Data Flow Diagrams (DFDs), starting from Diagram 0 down to its derived diagrams. In this exercise, however, we will focus on providing a detailed description of fundamental and core data elements used in the library management system, such as students, the faculty, library staffs, vendors, assets, transactions, and acquisitions. These elements have been populated based on data flows, structures, and stores. For simplicity, we will omit the detailed implementation of the DFDs in this exercise.

**Data Dictionaries**

As outlined in the assignment, the library management system will cater to multiple user types, including students, faculty, and library staff. By analyzing the distinct needs of each user type, we can identify certain customized attributes specific to each group, while also recognizing shared fields such as Address (see Table 1). The Address field will be composed of multiple components: street, apartment, city, state, and zip Code. To efficiently manage this data, we will create a common table dedicated to storing address information, which can be referenced by all user types. In addition to the common address table, we will establish three separate tables to store the unique attributes for each user type: students (see Table2), faculty (see Table3), and library staff (see Table4). Each table will contain a unique ID as the primary key to distinguish between users. Other essential fields common to all users, such as firstName, lastName, and email, will also be included. To better organize this information, we will create separate data dictionaries for each user type, as illustrated in Tables 1, 2, 3, and 4. These tables will outline the attributes for students, faculty, library staff, and the common address structure.

**Table 1**

***Data Dictionary of Address***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Field Name | Description | Data Type | Length | Data Range | Default Values |
| AddressID | Primary key, a unique identifier | Numeric | 10 | 100000000 - 999999999 | Auto-increment, cannot be null |
| Street | The street number and street name, as well as apartment/suite number | Varchar | 200 | Alphanumeric | None |
| City | The city name | Varchar | 50 | Alphanumeric | None |
| State | The state name | Varchar | 50 | Alphanumeric | None |
| ZipCode | A digit number that identifies a specific geographic area | Varchar | 12 | Alphanumeric | None |
| Longitude | Longitude of the address | Numeric | (9, 6) | Alphanumeric | None |
| Latitude | Latitude of the address | Numeric | (9, 6) | Alphanumeric | None |

Explanation:

* **AddressID**: This is a unique identifier for each address and serves as a critical reference point for linking user profiles to their contact details. It is auto incremented and cannot be null.
* **Longitude**: Represents the geographical longitude of the address. This data is typically obtained from a mapping service, such as Google Maps, to ensure consistency and accuracy.
* **Latitude**: Represents the geographical latitude of the address. Like the longitude, this is also sourced from a mapping service (e.g., Google Maps) to maintain uniform standards for location-based information.

**Table 2**

***Data Dictionary of Student***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Field Name | Description | Data Type | Length | Data Range | Default Values |
| StudentID | Primary key, a unique identifier | Numeric | 10 | 100000000 - 999999999 | Auto-increment, cannot be null |
| FirstName | Students’ first name | Varchar | 30 | Alphanumeric | None |
| LastName | Students’ last/family name | Varchar | 30 | Alphanumeric | None |
| Email | Students’ university email | Varchar | 100 | Alphanumeric ending with @xx.edu | None |
| HomePhone | The home phone of the student | Varchar | 11 | Alphanumeric starting with country code “1” | None |
| MobilePhone | The mobile phone of the student | Varchar | 11 | Alphanumeric starting with country code “1” | None |
| AddressID | Unique identifier of the address | Numeric | 10 | 1 - 999999999 | None |
| Department | The department name which the student belongs to | Varchar | 100 | Alphanumeric | None |
| Courses | List of courses taken by the student | Varchar | 500 | Alphanumeric | None |
| AdmissionDate | The admission date | Date | N/A | MM-DD-YYYY | None |

Explanation:

* **StudentID**: This is the unique identifier for each student, serving as the primary key. It is essential for logging into the library management system and other system processes, ensuring each student has a distinct identity.
* **Email**: The student's university email address, which is often used as alternative approach of the login credentials for the library management system, like login and reset password. Typically, this email ends with the university’s domain, such as @xx.edu.
* **Courses**: Represents the list of courses a student is currently enrolled in or has completed. These course identifiers can be stored as comma-separated values (e.g., “5103, 5101, 5300”) and are useful for generating transcripts or academic reports.
* **AddressID**: A unique identifier that links the student to their corresponding address details in the Address Information table.
* **AdmissionDate**: The date when the student officially begins their program at the university. This information is critical for tracking academic timelines and generating reports.

**Table 3**

***Data Dictionary of Faculty***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Field Name | Description | Data Type | Length | Data Range | Default Values |
| FacultyID | Primary key, a unique identifier | Numeric | 10 | 100000000 - 999999999 | Auto-increment, cannot be null |
| FirstName | Faculty’s first name | Varchar | 30 | Alphanumeric | None |
| LastName | Faculty’s Last name | Varchar | 30 | Alphanumeric | None |
| Email | Faculty’s university email | Varchar | 100 | Alphanumeric ending with @xx.edu | None |
| OfficePhone | The office phone of the faculty | Varchar | 11 | Alphanumeric starting with country code “1” | None |
| MobilePhone | The mobile phone of the faculty | Varchar | 11 | Alphanumeric starting with country code “1” | None |
| AddressID | Unique identifier of the address | Numeric | 10 | 1 - 999999999 | None |
| Department | The department name where the faculty teaches | Varchar | 100 | Alphanumeric | None |
| Role | The academic role rank of the faculty | Varchar | 30 | Instructor, Associate Professor, and Professor | None |
| StartDate | The first day the faculty starts working for the university | Date | N/A | MM-DD-YYYY | None |

Explanation:

* **FacultyID**: A unique identifier assigned to each faculty member, serving as the primary key. It ensures that each faculty member has a distinct ID within the system.
* **Email**: The faculty’s email address, often used for communication and system login. Like student emails, it typically ends with the university’s domain, such as @xx.edu.
* **AddressID**: A unique identifier that links the faculty member to their address details in the Address Information table.
* **Role**: Defines the academic rank of the faculty member. The standard ranks include Instructor, Assistant Professor, Associate Professor, and Professor. Faculty members can select from these predefined roles or leave it blank if necessary.
* **StartDate**: The date when the faculty members officially begin their employment at the university. This is used to track tenure and employment history.

**Table 4**

***Data Dictionary of Library Staff***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Field Name | Description | Data Type | Length | Data Range | Default Values |
| LibraryStaffID | Primary key, a unique identifier | Numeric | 10 | 100000000 - 999999999 | Auto-increment, cannot be null |
| FirstName | Library Staff’s first name | Varchar | 30 | Alphanumeric | None |
| LastName | Library Staff’s last name | Varchar | 30 | Alphanumeric | None |
| Email | Library Staff’s university email | Varchar | 100 | Alphanumeric ending with @xx.edu | None |
| OfficePhone | The office phone of the Library Staff | Varchar | 11 | Alphanumeric starting with country code “1” | None |
| MobilePhone | The mobile phone of the Library Staff | Varchar | 11 | Alphanumeric starting with country code “1” | None |
| AddressID | Unique identifier of the address | Numeric | 10 | 1 - 999999999 | None |
| StartDate | The first day the Library Staff starts working for the university library | Date | N/A | MM-DD-YYYY | None |

Explanation:

* **LibraryStaffID**: The primary key for library staff member is unique within the system.
* **Email**: The email address of the library staff member, typically ending with the university’s domain, such as @xx.edu. This email should also be unique.
* **AddressID**: A unique identifier linking the library staff members to their address details stored in the Address Information table.

**Table 5**

***Data Dictionary of Library Asset***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Field Name | Description | Data Type | Length | Data Range | Default Values |
| AssetID | Primary key, a unique identifier | Numeric | 10 | 100000000 - 999999999 | Auto-increment, cannot be null |
| Title | The Title/Name of the asset | Varchar | 255 | Alphanumeric | None |
| Author | Name (List) of the authors | Varchar | 255 | Alphanumeric | None |
| AssetType | Category of the asset | Enum | N/A | Book,  Journal,  E-book,  Video | Books |
| Category | Category of the asset | Varchar | 100 | Free Text | None |
| ISBN | ISBN for books | Varchar | 13 | Alphanumeric | None |
| PublicationDate | Date of the asset was published | Date | N/A | MM-DD-YYYY | None |
| Status | Status of the asset | Enum | N/A | Available,  Checked Out,  Lost,  Repair | Available |

Explanation:

* **AssetID**: A unique identifier for each asset, serving as the primary key. This ensures that every asset in the system is uniquely identifiable.
* **Title**: The title or name of the asset, such as the name of a book, journal, or video.
* **Author**: The name of the author(s) who created the asset. For multi-author works, this field can include multiple names separated by commas.
* **AssetType**: Defines the type of asset, which can be one of the following: Books, Journals, E-books, or Videos. The default category is set to "Books."
* **Category**: Defines the genre or category of asset. Its categories is defined as free text input, but can also be Enum if any standard exists, like ‘Fiction’, ‘Science’, etc.
* **Status**: Indicates the current availability of the asset. The default asset status can be Available (ready for borrowing).
* **PublicationDate**: The date when the asset was published or released. It follows the standard MM-DD-YYYY format.

**Table 6**

***Data Dictionary of Library Transaction***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Field Name | Description | Data Type | Length | Data Range | Default Values |
| TransactionID | Primary key, a unique identifier | Numeric | 15 | 100000000000000 – 999999999999999 | Auto-increment, cannot be null |
| BorrowerID | Unique identifier for the borrower | Numeric | 10 | 1000000000-9999999999 | None |
| BorrowerType | The user type of borrower | Varchar | 15 | Student,  Faculty,  Library Staff | None |
| AssetID | Unique identifier for the borrowed asset | Numeric | 10 | 1000000000-9999999999 | None |
| IssueDate | The date the asset is borrowed out | Date | N/A | MM-DD-YYYY | None |
| DueDate | The due date of the asset that should be returned | Date | N/A | MM-DD-YYYY | IssueDate + 30 days |
| ReturnDate | The actual return date of the asset | Date | N/A | MM-DD-YYYY | None |
| TransactionStatus | Status of transaction | Varchar | 10 | Open  Closed |  |
| IssuedBy | The ID of the Library Staff who helps process the transaction | Numeric | 10 | 1000000000-9999999999 | None |
| ReturnedBy | The ID of the Library Staff who helps process the transaction | Numeric | 10 | 1000000000-9999999999 | None |

Explanation:

* **TransactionID**: A unique identifier for each borrow/return transaction. The table will be the most important when auditing due to its characteristics.
* **BorrowerID**: This field will reference the ID from the Student, Faculty, or Library Staff dictionary. It links to the user who performed the transaction.
* **BorrowerType**: Indicates the type of borrower making the transaction. Combined with the BorrowerID, the system can accommodate different user roles without creating separate transaction records for each type of user. This simplifies the transaction structure while ensuring that all users can engage in the same types of activities (borrowing/returning assets).
* **AssetID**: This field will reference the ID from the asset information dictionary.
* **TransactionStatus**: when the asset is borrowed out, the transaction status is “Open” and keep open until the asset is returned, the transaction status then changes to “Closed”. When the transaction is closed, the asset would show as “Available” status in the system.
* **IssueDate**: the date when the user went to borrow the asset.
* **DueDate**: The default due date when the borrower should return the borrowed asset. The default value can set to “IssueDate” + 30 days.
* **ReturnDate**: The actual date when the asset was returned.
* **IssuedBy**: A foreign key reference to the library staff ID, identifying the member who issued the asset.
* **ReturnedBy**: A foreign key reference to the library staff ID, identifying the member who returned the asset.

**Table 7**

***Data Dictionary of Library Asset Acquisition***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Field Name | Description | Data Type | Length | Data Range | Default Values |
| AcquisitionID | Primary key, a unique identifier | Numeric | 15 | 100000000000000 – 999999999999999 | Auto-increment, cannot be null |
| AssetID | ID of the newly acquired asset | Numeric | 10 | 1000000000-9999999999 | None |
| VendorID | Unique identifier for the vendor | Numeric | 10 | 1000000000-9999999999 | None |
| AcquisitionDate | Date the asset was acquired | Date | N/A | MM-DD-YYYY | None |
| AcquisitionDetails | Detail notes of the asset acquisition | Varchar | 500 | Free text | None |
| Cost | The cost to acquire the asset | Numeric | (11, 2) | 0.00 - 999999999.99 | None |

Explanations:

* **AcquisitionID**: A unique identifier assigned to each acquisition, allowing for the tracking and management of new additions to the library's inventory.
* **AssetID**: References the unique ID from the Asset Information table, linking the acquisition to the specific asset that was purchased.
* **VendorID**: References the unique ID from the Vendor Information table, identifying the vendor from whom the asset was acquired.
* **AcquisitionDate**: The date on which the asset was officially acquired by the library, recorded in MM-DD-YYYY format.
* **AcquisitionDetails**: Additional notes or comments about the acquisition, such as quantity, quality (used, brand-new, etc.) of the purchased asset or other relevant information about the purchase or vendor.
* **Cost**: The total cost incurred to acquire the asset, with a precision of two decimal places (e.g., $1234.56), useful for budgeting and financial tracking.

**Table 8**

***Data Dictionary of Vendor***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Field Name | Description | Data Type | Length | Data Range | Default Values |
| VendorID | Primary key, a unique identifier | Numeric | 10 | 100000000 - 999999999 | Auto-increment, cannot be null |
| VendorName | the name of the vendor | Varchar | 150 | Alphanumeric | None |
| Description | The description of the vendor | Varchar | 500 | Alphanumeric | None |
| Email | The unique email of the vendor | Varchar | 50 | Alphanumeric | None |
| OfficePhone | The Office Phone of the vendor | Varchar | 11 | Alphanumeric | None |
| AddressID | The address of the vendor | Varchar | 50 | Alphanumeric | None |

Explanation:

* **VendorID**: A unique identifier for each vendor, serving as the primary key. It ensures that each vendor is distinctly recognized in the system.
* **VendorName**: The name of the vendor, which could be a personal name, company name, or publisher name, depending on the source of the asset.
* **Description**: A brief description of the vendor, providing additional context, such as the type of business or services they offer.
* **Email, OfficePhone, and AddressID**: These fields store the vendor’s contact details, including their email address, office phone number, and reference to their address in the Address Information table. This information is essential for auditing, communication, and ensuring accountability for asset quality and maintenance.

**Conclusion**

This data dictionary outlines the core elements essential for a well-structured library management system. It categorizes key entities such as students, faculty, library staff, vendors, addresses, assets, transactions, and acquisitions, ensuring that each is accurately represented with well-defined data types and lengths. The use of unique identifiers (e.g., StudentID, AssetID, VendorID) ensures seamless tracking and management of resources across the system. By maintaining consistency in data structures, particularly through shared elements like address information, the system enhances clarity, operational performance, and accountability. Furthermore, the uniform handling of transaction information across different user types (students, faculty, staff) simplifies the system's design, streamlining reporting and querying processes. This unified approach not only optimizes the user experience but also supports effective auditing and resource management, resulting in a more efficient and reliable library management system.

**References**

Airbyte. (2024, July 18). How to Create a Data Dictionary in 10 Simple Steps. *Airbyte*. <https://airbyte.com/data-engineering-resources/how-to-create-a-data-dictionary>