

**Referral Intake Management System**

ECE 9017 – Group Project

Group Number: 03

Group Members:

Vishal Bhardwaj (251117897)

Mandeep Singh (251122474)

Rohitashwa Ray Chowdhury (251081900)

**Abstract:**

This project involves a referral intake management system dataset that is used to track patient admission and recovery time through different stages. In each stage, the status of the patient, the time taken to reach that status along with the user id of the nurse involved in updating the record is saved. This dataset involves a total of 27 attributes encompassing of all the unique statuses of the patient’s walkthrough before finally getting discharged from the hospital. The features present in the data set are first modeled into the relational database in the third normal form which further feeds the data into the data warehouse for the analysis purpose. Owing to the multi-dimensionality nature of the data, OLAP hypercube is used for an analytical understanding of the data.

**Objective:**

The main purpose of modeling the data into data warehouse revolves around the following business requirements: -

1. To fetch total time spent in minutes by a patient under-diagnosis in that hospital.
2. To find the average time taken by patients from the admitting date till discharged date in the hospital.
3. To get the total number of days a patient under treatment has spent in the hospital until discharged.
4. To fetch the count of the number of patients a nurse has handled per department per location.

These business requirements will be fulfilled using a data warehousing technique wherein dimension and fact tables will be leveraged to get useful information about the data.

**Dataset Description (Pre-normalized form)**:

The pre-normalized dataset contained in flat file consists of the following attributes.

1. Client ID**:** This is a unique identifier used for the identification of each patient admitted to the hospital. The data type of this attribute is INT.
2. CreatedDateTime**:** This attribute signifies the date at which the patient was added to the system.
3. FYear\_Name**:** The financial year during which the patient entry was added to the system.
4. Work\_OutsideDay: This attribute informs us about the hours during which the patient record was created into the system. This is in the NVARCHAR (30).
5. StatNew: This attribute informs if the patient is new to this hospital or not. This is in the NVARCHAR (10) format.
6. TimeTo\_NewStat: The time taken by each patient for all the statuses which he has undergone. The data type for this attribute is INT.
7. FirstStatChange**:** This attribute displays the first status of the patient after being admitted. This is in the format NVARCHAR (30).
8. TimeTo\_FirstStatChange**:** This displays the time to the first status change of the patient. It is in the INT format.
9. TimeTo\_PSA\_Review: This displays the time corresponding to the PSA review for the patient. It is saved in the INT format.
10. TimeTo\_PSA\_Accepted: The time taken by the patient to reach PSA Accepted status is recorded under this. It is specified in the INT format.
11. TimeTo\_Triaged: This is the time taken to triage the patient to a specific department. This is in the format INT.
12. TimeTo\_CC\_Accepted: This is the time taken for the patient admission confirmation. It is in the format INT.
13. TimeTo\_Awaiting\_CallBack: This is the time taken by the nurse or hospital to resume the patient’s workflow. This data type is in the format INT.
14. StatCompleted: This data type shows the current status of the patient. It is in the format NVARCHAR (20).
15. CompletedDateTime: This gives the date and the time when the patient was discharged. It is in the DATE format.
16. TimeTo\_Completed: This is a calculated value column that shows the amount of time taken by the patient to get discharged from the start or the first stat change. This is in the format INT.
17. Patient\_type: This data type tells the type of the patient which can be either NEW or EXISTING. It is in the format NVARCHAR (20).
18. Priority: This attribute tells the priority of the case that is being handled by the nurse. It is in the format NVARCHAR (20).
19. Referral Category: This corresponds to the department the patient was referred to by the nurse. It is in the format NVARCHAR (30).
20. Location: This is the nurse's location from where he or she is handling the patients. This is in the format NVARCHAR (20).
21. Completed UserID: This attribute tells about the nurse who marked a patient’s case complete. This is in the format NVARCHAR (20).
22. User Group Name: This is the name of the group the nurse belongs to. This is in the format NVARCHAR (20).
23. Completed\_Work\_OutsideDay: This attribute tells if the patient was marked complete or discharged during the workday or outside the workday. This is in the format NVARCHAR (20).
24. 1stStat\_To\_Triage\_Hour: This is the time taken for the patient from his or her first status change to triaging him or her to a department. This is in the format INT.
25. 1stStat\_To\_Completed\_Hour: Time is taken by the patient from his or her first status change until he or she is marked complete by a nurse. This is in the format INT.
26. TimeTo\_Completed\_Day: This is a calculated column which gives the total time taken by the patient from 1st stat change to complete. This is in the format FLOAT.
27. LastStatChange: This is the last recorded status change of the patient. This is in the format NVARCHAR (30).

**Data Normalization: -**

Since, this dataset is not structured in accordance with the relational database, therefore we restructured it into a relational database to reduce data redundancy and improve data integrity. In other words, the features were organized into database tables and columns to logically store the structure in the database. This process has been done on the imported CSV flat-file table which is the primary data source in our case. The main objective of normalization was to minimize the data redundancy and eliminating insert, update and delete anomalies by dividing the larger tables into smaller chunks and linking them using relationships. Following entities can be extracted from the Referral Intake Management System: -

1. Patient
2. Nurse
3. Status Type
4. Time\_to\_Statuses

Furthermore, the relationship between these entities can be referred from the below logical diagram.

A close up of a map

Description automatically generated

Figure 1 – ER diagram of Referral Intake Management System

**Methodology Adopted for Normalization:**

**First Normal Form:**

The primary flat file encompassing of a single table with several attributes could potentially cause many updates, delete and insert anomalies. For instance, there exist multiple statuses of patients under one column which results in breaking the first normal form principle of atomicity rule. After converting into the first normal form, columns are checked for partial dependency to form the second normal form. Furthermore, transitive dependencies are eliminated, and multiple candidate keys are removed from a single table to get all the tables into the third normal form. Therefore, after separating the inter-dependent relationships from all the tables, the following entities are created to get to the third normal form.

1. Department – This table will contain the department details.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| DepartmentID | INT (Sequence (100)) | Primary Key | Not Null |
| DepartmentName | Nvarchar (50) | Unique Constraint | Not Null |

1. Locations– All the locations' names are persisted into this table.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| LocationID | INT (Sequence (100)) | Primary Key | Not Null |
| LocationName | Nvarchar (10) | Unique Constraint | Not Null |

1. DepartmentLocation– Since, there is a many-to-many relationship between departments and locations, therefore, a junction table between departments and locations is formed.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| DepartmentLocationID | INT (Sequence (100)) | Primary Key | Not Null |
| DepartmentID | INT |  | Not Null |
| LocationID | INT |  | Not Null |

1. Nurse– This table contains the nurse username and the group to which she belongs.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| NurseID | INT (Sequence (100)) | Primary Key | Not Null |
| NurseUserName | Nvarchar (20) |  | Not Null |
| Nurse GroupName | Nvarchar (50) |  |  |

1. PriorityType – This table comprises of all the priorities assigned to the patient in the hospital.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| PriorityTypeID | INT (Sequence (100)) | Primary Key | Not Null |
| PriorityTypeName | Nvarchar (30) | Unique Constraint | Not Null |

1. PatientStatus – This table encompasses all the patient statuses that are assigned to the patient under consideration.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| StatusID | INT (Sequence (10)) | Primary Key | Not Null |
| StatusName | Nvarchar (40) | Unique Constraint | Not Null |

1. PatientType – This table contains the type of the patient.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| PatientTypeID | INT (Sequence (1000)) | Primary Key | Not Null |
| PatientType | Nvarchar (50) | Unique Constraint | Not Null |

1. Patient – This is the main table that contains all the information related to the patients. The following table illustrates the columns present in this table.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| PatientID | BigInt (Sequence (100)) | Primary Key | Not Null |
| PatientTypeID | Nvarchar (50) |  | Not Null |
| PatientReferralCategoryID | Int |  | Not Null |
| LocationID | Int |  | Not Null |
| PriorityTypeID | Int |  | Not Null |
| PatientCurrentStatusID | Int |  | Not Null |
| CreatedAtWorkingHours | Nvarchar (20) |  | Not Null |
| CreatedAtDateTime | Date |  | Null |
| CreatedAtFY\_Name | Nvarchar (10) |  | Null |
| DischargedWorkingHours | Nvarchar (20) |  | Not Null |

1. Patient\_PatientStatus – Owing to the many-to-many relationship which exists between patient and patient status table, a junction table with the following columns is created.

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | Constraint (If Any) | Null Constraint |
| Patient\_PatientStatusID | INT (Sequence (100)) | Primary Key | Not Null |
| PatientID | Nvarchar (50) | Unique Constraint | Not Null |
| StatusID | Int |  | Not Null |
| Time\_To\_Status | Int |  | Not Null |
| MarkedCompletedByNurse | Int |  | Null |
| MarkedCompletedOnDate | Date |  | Null |
| MarkedCompletedOnTime | Time |  | Null |

Below physical diagram elucidates blueprint of the actual relations between the entities in the database.

A screenshot of a cell phone

Description automatically generated

Figure 2 – Physical Diagram of Relational Database in third normal form

Post creating the relational database structure, tables were populated using the SQL Server Integration Services or SSIS package. This package is contained in the RIMS\_Integration\_Services\_Project solution. Herein, the flat file was first loaded into the staging table in the database using the data flow feature of the SSIS. Post loading the flat file into our operational database, a stored procedure was triggered by the execute SQL task function in the SSIS package for populating the tables. This task function along with the flat file connected was used to populate the entire operational database.

**Data Warehouse Modelling: -**

In order to process the technical requirements of the project and draw the useful business insights of the data, we need to create a central repository of a system that will store all the related data. Since, we are calculating measures related to both Patients and Nurse, two fact tables are created using the data present in the data warehouse. Therefore, the constellation schema was chosen for the multidimensional modeling of our data warehouse. The following data marts (dimensions and fact tables) are used in data warehousing. The following diagram illustrates the architecture of the data warehouse of the application.

A picture containing cup, mug, food, glass

Description automatically generated

Figure 3 – Process involved for fetching requested technical requirements

**E.T.L. Process With S.S.I.S: -**

**Introduction:**

E.T.L. stands for Extract, Transform and Load. In this process, the data is extracted from different source systems which further is cleaned, merged through the transformation process to load it into the data warehouse system. This data collected from multiple sources is stored using staging tables. In the transform step, the data previously collected is transformed and checked against various predefined rules. The final step involves loading this transformed data into the warehouse.

**Methodology Adopted:**

1. **Extraction:** In order to store the data extracted from the relational database, staging tables dbo.patient\_staging, dbo.department\_staging are leveraged. These tables extract data from the operational database as per their attribute requirement. Stored procedures for extracting the data to further feed into staging tables have been written in SQL Server which has been triggered in the SSIS package using the Execute SQL Task functions. In total two execute SQL task functions patient extract and department extract have been used in this step to extract the data into the two staging tables.
2. **Transform:** The transformation part is completely dependent on the staging tables created in the extraction phase. In this step, the data is processed and converted according to the proper structure/storage format of the dimension tables created in the data warehouse. The two staging tables feed the data into four transformation tables. The data transformation and conversion function available in the SSIS package has been used in combination with the slowly changing dimension function also available in SSIS to transform the data in the tables according to their SCD levels. The patient and nurse dimension were created as an SCD two-level in order to store the patients and nurse history based on the new insertion, deletion or updates. The data of the transformed stage tables are stored into the preload tables which further loads the data into the dimension tables. The execute SQL task functions are again used to trigger the pre-written SQL server stored procedures to automatically loads the data into the preload tables. The four stored procedure used to populate transformed data are Patient\_preload, Nurse\_preload, Department\_preload and Fact\_patient\_preload is Patient\_transform, Nurse\_transform, Department\_transform and Fact\_patient\_transform. The next step is to load the data from these preload tables into the previously created dimension and fact tables.
3. **Loading:** In this step, the transformed data was loaded into the dimension tables (Patient dimension, Nurse dimension and Department dimension) and fact tables (Patient Fact and Nurse Fact). Like the extraction and loading phase, the data has been loaded using the SSIS package and then execute SQL task functions. These data marts and fact tables will help in fulfilling our technical requirements and solving the purpose of analyzing the data through SQL Server Analysis Services (S.S.A.S.). based on which the cube will be constructed and formed. The following diagram illustrates the whole process of Extraction, Transformation and loading into the RIMS\_DW data warehouse.

A picture containing black

Description automatically generated

Figure 4 – ETL process involved in Data Warehousing

**Multidimensional Cube: -**

Since, our dataset contains multiple dimensions, therefore, the hypercube is used to fetch all the technical requirements and gather useful insights from our data. SQL Server Analysis Services is used to build and deploy the data cube for further processing. The following figure illustrates the data source view of the data warehouse in S.S.A.S.

A close up of text on a white background

Description automatically generated

Figure 5 – Data source view while making multi-dimensional cube

Furthermore, the following attribute hierarchies are created based on the columns of the dimension tables. These hierarchies are logically created for further analysis of the data.

1. Nurse Hierarchy
2. Department Hierarchy
3. Calendar Hierarchy

Measures used while creating a cube are as follows: -

1. Total Patients Handled by the nurse. (Total Patients Handled)
2. Average Minutes Patient has spent in the hospital. (Average Mins)
3. Total Minutes spent by the patient in the hospital. (Total Mins)
4. Total admitted days of the patient until discharge. (Total Admitted Days)

These measures are calculated against patient cube and nurse cube to satisfy the technical and business requirements mentioned above.

![A screenshot of a cell phone

Description automatically generated]()

Figure 6 – Nurse Cube depicting number of patients handled by a nurse in particular department and location

![A screenshot of a cell phone

Description automatically generated]()

Figure 7 – Patient Cube depicting average time, days, total time, total days spent by a patient in the hospital

**Please note that following MDX query was used to extract the patient cube and nurse cube respectively.**

Nurse Cube MDX query-

(SELECT NON EMPTY { [Measures].[Patient Key], [Measures].[Total Admitted Days], [Measures].[Average Mins], [Measures].[Total Mins] } ON COLUMNS, NON EMPTY { ([Dim Nurse].[Nurse Key].[Nurse Key].ALLMEMBERS \* [Dim Nurse].[Nurse User Name].[Nurse User Name].ALLMEMBERS \* [Dim Department].[Dept Key].[Dept Key].ALLMEMBERS \* [Dim Department].[Location Name].[Location Name].ALLMEMBERS ) } DIMENSION PROPERTIES MEMBER\_CAPTION, MEMBER\_UNIQUE\_NAME ON ROWS FROM [RIMS DW] CELL PROPERTIES VALUE, BACK\_COLOR, FORE\_COLOR, FORMATTED\_VALUE, FORMAT\_STRING, FONT\_NAME, FONT\_SIZE, FONT\_FLAGS).

Patient Cube MDX query-

SELECT NON EMPTY { [Measures].[Total Patients Handled] } ON COLUMNS, NON EMPTY { ([Dim Department].[Department Name].[Department Name].ALLMEMBERS \* [Dim Department].[Location Name].[Location Name].ALLMEMBERS \* [Dim Nurse].[NurseHierarchy].[Nurse User Name].ALLMEMBERS ) } DIMENSION PROPERTIES MEMBER\_CAPTION, MEMBER\_UNIQUE\_NAME ON ROWS FROM [RIMS DW] CELL PROPERTIES VALUE, BACK\_COLOR, FORE\_COLOR, FORMATTED\_VALUE, FORMAT\_STRING, FONT\_NAME, FONT\_SIZE, FONT\_FLAGS