# **Final Project: Healthineer**

# Sravan Kumar Maheshwaram (1186204)

# Giri Prasad Rao Surineni (1184458)

# SainathReddy Bhaskari (1183301)

## **Saint Leo University**

**Database Systems COM-566-MSC04**

**Professor Joshua Adams**

**May 2, 2022**

Text, letter

Description automatically generatedStudent Signature: [Type Full Name Here]

# Sravan Kumar Maheshwaram

# Giri Prasad Rao Surineni

# Sai nath reddy

# Group Members Signatures: Sravan kumar Maheswaram, \_Giri Prasad Rao Surineni, \_SainathReddy Bhaskari

Contents

[Introduction 3](#_Toc102396659)

[Background 4](#_Toc102396660)

[Requirements 5](#_Toc102396661)

[ER Model 6](#_Toc102396662)

[Relational Schema: 7](#_Toc102396663)

[Primary Key and Foreign Key Relation 8](#_Toc102396664)

[Database Tuples in Patients Table 8](#_Toc102396665)

[SQL Queries 9](#_Toc102396666)

[1)Aggregate Functions 9](#_Toc102396667)

[2)Dates 9](#_Toc102396668)

[3)SQL Logical Operators 10](#_Toc102396669)

[4)Regular Expressions 10](#_Toc102396670)

[5)Formula Expressions 10](#_Toc102396671)

[6)Subqueries 11](#_Toc102396672)

[Infrastructure Diagram 11](#_Toc102396673)

[DBMS Query 13](#_Toc102396674)

[Database definition Queries 13](#_Toc102396675)

[Database Query Language 13](#_Toc102396676)

[Database manipulation Queries 14](#_Toc102396677)

[Data Control Queries 14](#_Toc102396678)

[Diagrams 15](#_Toc102396679)

[1)Database 15](#_Toc102396680)

[2)Infrastructure 15](#_Toc102396681)

[3)Relational Datatype 16](#_Toc102396682)

[4)Schema 17](#_Toc102396683)

[5)Sequence 17](#_Toc102396684)

[6)Pseudo code 18](#_Toc102396685)

[7)Table with all necessary data type (relational) 18](#_Toc102396686)

[8) diagram / figure for nonsql/ MongoDB/ Unstructured 19](#_Toc102396687)

[Concurrency Control 19](#_Toc102396688)

[Conclusion: 20](#_Toc102396689)

[References 20](#_Toc102396690)

Introduction:

Data has been adding value to every industry and organization to make critical business decisions based on analysis and strategies in the market. Sensitive information needs to be secured with necessary protocols in place, as falling into the wrong hands might compromise an individual's identity or cause a loss of value to an organization. Transmission of sensitive information securely within the robust mechanism helps any organization stay ahead and competitive in the industry. 1

Healthcare is one such industry where every piece of data has a value that allows evaluation of the patient with necessary procedures. The database of healthcare can be vast and increasing every day. Managing large amounts of data can be challenging without the required systems and software. Handling robust software systems within the healthcare industry can improve the overall health system within a specific region or country.

This Healthineer project focuses on developing a robust mechanism of transmission of data with secured operating procedures that help the healthcare organizations and patients with ease in the direction of the proper diagnosis and analysis of patients' health conditions. We aim to develop and integrate the complex areas of healthcare that helps the patients schedule their appointments and tests that store health records which enable the doctors to evaluate the patient's health condition and history at any given point.

This project is classified into the integration of four major areas: Patient Management Systems, Electronic Health Records, Laboratory Information Systems, and financial systems.

* The Patient Management System has the demographic details of the patient and can schedule or alter the appointments that will alert accordingly by an Email or SMS
* Electronic health record has the events of the patient to the healthcare provider and requests the tests before or after the appointment.
* A laboratory information system records all the tests and data of the patients that a doctor can evaluate the condition.
* Finance systems integrates expenses of the patient at healthcare to connect with the right insurance provider provided by them after verification.

Integrating these significant areas helps patients schedule an appointment through the Patient Management system with the necessary doctor at their available time, making the healthcare process feasible for patients, doctors, and management staff. Using the Laboratory and health records, patients can attend to their tests that help evaluate their current condition by the respective doctor. After this, the Finance system verifies the patient details to the necessary healthcare provider that will clear the financial process.

Below is the illustration of the integration

Diagram

Description automatically generated

Fig 1. Illustration on major areas integration of Healthineer Project

Background:

Healthcare systems are prone to vulnerabilities due to emerging technology and sophisticated methods from attackers. 94% of healthcare organizations have experienced security breaches as the systems are not updated, which causes vulnerabilities that allow an attacker to get into the system. A security breach in the healthcare industry results in manipulation that can be fatal. Authentication and authorization to healthcare databases are secured at every stage of the development process from time to time. Systems need to be updated accordingly to avoid the breaches and vulnerabilities that make the database management systems more robust. (Kumar, 2020)

Attackers manipulated the required dosages of insulin at Johnson and Johnson in 2015. Data breaches have been increasing enormously, according to recent reports. Russian attackers compromised the data of the world anti-doping agency by altering and publishing health information on athletes. According to Federal Trade Commission, it has been cited that there are 19 attacks mentioned every minute, which is due to the weak operating systems, which are not updated with the technology movement. There has been three times increase in the compromise of patient records from 2017 to 2018. From the well-known health organizations Quest diagnostics and LabCorp, attackers compromised the data of almost 19 million patients. By 2025, the security market for the healthcare industry can exceed $27bn. Maintaining data integrity and securing sensitive information through connected management systems and operational procedures can be vital, especially in healthcare.

Requirements:

Healthcare data has never-ending addition of data either on a patient, processes, and new technologies and parameters that help evaluate a patient and provide proper care based on the condition. When multiple users try to access and alter data, there might be data duplication issues. This will affect the data storage with complications and affect the evaluation of the condition. Data and applications need to accommodate the increasing amounts of data and future applications. Hence, there will be a need for two servers, one that hosts the patient's data and the other for the healthcare management that doctors, management, and clinical staff use. API will be used to interact between patients, healthcare staff, and database servers.

ER Model:

Timeline

Description automatically generated with low confidence

Fig 2. Entity Relationship Diagram

The above Entity Relationship Diagram represents the relation between the entities used in database systems. Usually, the relation between the two tables represents the database process flow that includes the relation between primary keys and foreign keys.

## Relational Schema:

Diagram

Description automatically generated

Fig 3. Relational Schema

The Relational Schema represents the relation between attributes, fields, and columns associated with the same entity.

For Example, the above doctors' table represents the relation of other entity attributes like schedule ID, schedule type and Doctor ID, Doctor type, Doctor name.

### Primary Key and Foreign Key Relation

Table

Description automatically generated

### Database Tuples in Patients Table

Table

Description automatically generated

## SQL Queries

### 1)Aggregate Functions

Graphical user interface, text, application

Description automatically generated

### 2)Dates

A picture containing text

Description automatically generated

### 3)SQL Logical Operators

A screenshot of a computer

Description automatically generated with medium confidence

### 4)Regular Expressions

Graphical user interface, text, application

Description automatically generated

### 5)Formula Expressions

Graphical user interface, application

Description automatically generated

### 6)Subqueries

A screenshot of a computer

Description automatically generated

## Infrastructure Diagram

Diagram

Description automatically generated

The above infrastructure illustrates the structure flow on Azure services. The on-premises system needs to be moved on to the cloud for better availability of the servers. Azure services provide 99.99% availability, high scalability, and low latency. Staff and patients use the web application, and this application will interact with the cloud database, which is hosted on Azure. On Azure, besides the cloud solution, webservers that host virtual machines are also located. The Virtual Machines holds to Azure SQL, My SQL database, and webservers. Virtual networks are used to connect web servers and Database systems. Azure Active Directory maintains the application users with the authentication process and holds the blob data in the cloud environment with restricted access.

The system needs to alter and adjust the type of data, whether structured, Semi-structured, or unstructured, to the kinds of databases. Databases can be OLTP (On-Line Transaction Processing), read-only databases used to store the data, and OLAP (On-Line Analytical Processing), which is used for editing, altering, and deleting according to the requirement. Relational SQL and no SQL or MongoDB are used in database servers for structured and unstructured data, respectively. Using this SQL can further develop and induce concepts like DDL and DML, which help the database queries with ease. Data Definition Language defines the data within the database structure, and Data Manipulation Language performs, edits, and deletes the requirement to the database.

Table

Description automatically generated

The above table is the understanding that this project depends on Relational Database Management System which provides secured mechanism and tables that are relational and interlinked for better availability.

## DBMS Query

Database definition Queries

These are used to create the entities in the database

Graphical user interface, text, application

Description automatically generated

Database Query Language

It is used to query the stored data

Text

Description automatically generated

### Database manipulation Queries

It is used to alter the data which is existing in database

Text

Description automatically generated

### Data Control Queries

It will control the database operations like data insertion, modification, and removal process.

Text

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

## Diagrams

### 1)Database

Graphical user interface

Description automatically generated with medium confidence

The above diagram illustrates the database process from customer proxy initiation to database engine node access. This diagram shows the connection between IDE to SQL engine.

### 2)Infrastructure

Diagram

Description automatically generated

The above diagram depicts the structural flow of Azure services on cloud.

### 3)Relational Datatype

Diagram

Description automatically generated

The above tables are connected in relational between two entity datatypes

### 4)Schema

Table

Description automatically generated

The above table shows the schema of a structured entity.

### 5)Sequence

Diagram, timeline

Description automatically generated

The above process flow describes the sequence of operations in appointment scheduling.

### 6)Pseudo code

Timeline

Description automatically generated with low confidence

The above commands provides the structural idea for the exceptional handling.

### 7)Table with all necessary data type (relational)

Table

Description automatically generated

The above tabular data shows the various datatypes used in the structured SQL tables.

### 8) diagram / figure for nonsql/ MongoDB/ Unstructured

Text

Description automatically generated

The above code snippet show the non-structured data storage using MongoDB collections.

Concurrency Control

In healthcare systems, the data is vast, and increasing usage level. To provide the availability with low latency, SQL or any database provides multi-user access to the database operations. This will cause concurrency errors like lost updates, access to uncommitted data, non-repeatable reads, and phantom read phenomenon. To overcome these errors database is provided with concurrency control features including in the database configuration. Explicitly database server supports concurrency controls in the query language. (Bansal, 2015)

In this project, Concurrency control is used in scheduling appointments with the doctors by multiple users at a time without any conflicts. For example, Users can create appointments with their respective doctors; however, when a user fixes the appointment schedule, it cannot be accessible by other users.

## Conclusion:

Healthineer project aims to provide a robust database management system to the healthcare organization that concentrates on a cloud solution for high scalability, up-time, and low latency. The database management system has SQL for structured and MongoDB for unstructured. The main goal is to ensure providing better systems to the healthcare systems as everyone understand the healthcare industry plays a significant role in every individual and the government. The systems need to update in time for security concerns, and audits help the organizations provide better facilities to their patients with uptime.

References:

1. Bansal, V., Bhatia, A. B. (2015). Database Management System. United Kingdom: Alpha Science International, Limited.
2. Kumar, R., Agrawal, A., & Khan, R. A. (2020). A wake-up call for data integrity invulnerability. *Computer Fraud & Security*, *2020*(4), 14-19.
3. Kumar, R., Agrawal, A., & Khan, R. A. (2020). A wake-up call for data integrity invulnerability. *Computer Fraud & Security*, *2020*(4), 14-19.
4. ‘Data breach at major healthcare firms’ Computer Fraud & Security, 2019 (6) (Jun 2019)

[www.sciencedirect.com/science/article/pii/S1361372319300594](http://www.sciencedirect.com/science/article/pii/S1361372319300594),

1. Seber, A., Lins, M. M., Hazarbassanov, N. Q., & Lee, M. L. D. M. (2021). Pediatric Acute Lymphoblastic Leukemia: Database Analysis of Patients and Treatments in a National Public Healthcare System. *Blood*, *138*, 4105.
2. Emborg, H. D., Kahlert, J., Braeye, T., Bauwens, J., Bollaerts, K., Danieli, G., ... & Sturkenboom, M. (2020). Advance system testing: can coverage of pertussis vaccination be estimated in European countries using electronic healthcare databases: an example. *Vaccine*, *38*, B22-B30.
3. Pricing Calculator | Microsoft Azure. (2019). Microsoft.com; Microsoft Azure. <https://azure.microsoft.com/en-us/pricing/calculator/>
4. Microsoft. (2017, May 31). Active Directory Domain Services Overview. Microsoft.com. <https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
5. Azure Benefits and incentives | Microsoft Azure. (n.d.). Azure.microsoft.com. <https://azure.microsoft.com/en-us/pricing/offers/#cloud>‌
6. Directory of azure cloud services: Microsoft Azure. Directory of Azure Cloud Services | Microsoft Azure. (n.d.). Retrieved March 21, 2022, from <https://azure.microsoft.com/en-us/services/>
7. *SQL Server when to use Rowlock, Updlock etc*. SQL Server When to use RowLOCK, UPDLOCK etc - Microsoft Q&A. (n.d.). Retrieved March 31, 2022, from https://docs.microsoft.com/en-us/answers/questions/150070/sql-server-when-to-use-rowlock-updlock-etc.html
8. David-Engel. (n.d.). *Structured query language (SQL) - ODBC API reference*. ODBC API Reference | Microsoft Docs. Retrieved May 2, 2022, from https://docs.microsoft.com/en-us/sql/odbc/reference/structured-query-language-sql?view=sql-server-ver15
9. Jboeshart. (n.d.). *Azure SQL Fundamentals - Learn*. Learn | Microsoft Docs. Retrieved May 2, 2022, from https://docs.microsoft.com/en-us/learn/paths/azure-sql-fundamentals/
10. Nanda, A. (2006). Hands-On Microsoft SQL ServerTM 2005 Integration Services. Portugal: McGraw-Hill Education.
11. Markingmyname. (n.d.). *Create relationships between tables on a diagram - visual database tools*. Create Relationships Between Tables on a Diagram - Visual Database Tools | Microsoft Docs. Retrieved May 2, 2022, from https://docs.microsoft.com/en-us/sql/ssms/visual-db-tools/create-relationships-between-tables-on-a-diagram-visual-database-tools?view=sql-server-ver15
12. Köhler, H., Leck, U., Link, S., & Zhou, X. (2016). Possible and certain keys for SQL. *The VLDB Journal*, *25*(4), 571-596.
13. Wu, C., Luo, G., Guo, C., Ren, Y., Zheng, A., & Yang, C. (2020). An attention-based multi-task model for named entity recognition and intent analysis of Chinese online medical questions. *Journal of Biomedical Informatics*, *108*, 103511.
14. Kline, K., Kline, D., Hunt, B. (2004). SQL in a Nutshell: A Desktop Quick Reference. United States: O'Reilly Media.
15. SQLite Database System Design and Implementation (Second Edition, Version 1): (See other editions and books at https://books.google.com/books/?id=zSbxCwAAQBAJ and decide one). (2015). (n.p.): Sibsankar Haldar.
16. Minewiskan. (n.d.). *Comparison Operators - SQL Server*. SQL Server | Microsoft Docs. Retrieved May 2, 2022, from <https://docs.microsoft.com/en-us/sql/mdx/comparison-operators?view=sql-server-ver15>
17. Minewiskan. (n.d.). *Expressions (MDX) - SQL server*. SQL Server | Microsoft Docs. Retrieved May 2, 2022, from <https://docs.microsoft.com/en-us/sql/mdx/expressions-mdx?view=sql-server-ver15>
18. Minewiskan. (n.d.). *Comments (MDX syntax) - SQL server*. SQL Server | Microsoft Docs. Retrieved May 2, 2022, from <https://docs.microsoft.com/en-us/sql/mdx/comments-mdx-syntax?view=sql-server-ver15>
19. VanMSFT. (n.d.). *Queries - SQL server*. SQL Server | Microsoft Docs. Retrieved May 2, 2022, from <https://docs.microsoft.com/en-us/sql/t-sql/queries/queries?view=sql-server-ver15>
20. VanMSFT. (n.d.). *With common\_table\_expression (transact-SQL) - SQL server*. SQL Server | Microsoft Docs. Retrieved May 2, 2022, from <https://docs.microsoft.com/en-us/sql/t-sql/queries/with-common-table-expression-transact-sql?view=sql-server-ver15>
21. MikeRayMSFT. (n.d.). *Group by (transact-SQL) - SQL server*. GROUP BY (Transact-SQL) - SQL Server | Microsoft Docs. Retrieved May 2, 2022, from <https://docs.microsoft.com/en-us/sql/t-sql/queries/select-group-by-transact-sql?view=sql-server-ver15>
22. VanMSFT. (n.d.). *Having (transact-SQL) - SQL server*. HAVING (Transact-SQL) - SQL Server | Microsoft Docs. Retrieved May 2, 2022, from <https://docs.microsoft.com/en-us/sql/t-sql/queries/select-having-transact-sql?view=sql-server-ver15>
23. Smith, H. C. (1985). Database design: Composing fully normalized tables from a rigorous dependency diagram. *Communications of the ACM*, *28*(8), 826-838.
24. Tschanz, D. W., Gunderloy, M., Jorden, J. L. (2007). Mastering Microsoft SQL Server 2005. Germany: Wiley.
25. Link, S., & Prade, H. (2019). Relational database schema design for uncertain data. *Information Systems*, *84*, 88-110.
26. Maksimchuck, R. A., Naiburg, E., Naiburg, E. J. (2001). UML for Database Design. United Kingdom: Addison-Wesley.
27. Metwalli, S. A. (2021, June 23). *Pseudocode 101: An introduction to writing good pseudocode*. Medium. Retrieved May 2, 2022, from <https://towardsdatascience.com/pseudocode-101-an-introduction-to-writing-good-pseudocode-1331cb855be7>
28. Harrington, J. L. (2003). SQL Clearly Explained. Netherlands: Elsevier Science.
29. Encyclopedia of Big Data. (2022). United States: Springer International Publishing.
30. Lee, Y. T. (2017). Formalisms on Semi-Structured and Unstructured Data Schema Computations. United States: BiblioBazaar.
31. *Create and format tables*. in Excel. (n.d.). Retrieved May 2, 2022, from <https://support.microsoft.com/en-us/office/create-a-table-in-excel-bf0ce08b-d012-42ec-8ecf-a2259c9faf3f>
32. Cloud: Mongodb cloud. (n.d.). Retrieved May 2, 2022, from <https://account.mongodb.com/account/login?fromURI=https%3A%2F%2Funiversity.mongodb.com%2Foidc%2Fauthenticate>
33. Cloud: Mongodb cloud. (n.d.). Retrieved May 2, 2022, from <https://account.mongodb.com/account/login?fromURI=https%3A%2F%2Funiversity.mongodb.com%2Foidc%2Fauthenticate>
34. Cloud: Mongodb cloud. (n.d.). Retrieved May 2, 2022, from <https://university.mongodb.com/mercury/M001/2022_April_26/chapter/Chapter_1_What_is_MongoDB_/lesson/5f32ceb604e9ffbc5bc69b86/lecture>
35. Cloud: Mongodb cloud. (n.d.). Retrieved May 2, 2022, from <https://university.mongodb.com/mercury/M001/2022_April_26/chapter/Chapter_2_Importing_Exporting_and_Querying_Data/lesson/5f356aeb32339e734c9da867/lecture>
36. Cloud: Mongodb cloud. (n.d.). Retrieved May 2, 2022, from https://university.mongodb.com/mercury/M001/2022\_April\_26/chapter/Chapter\_3\_Creating\_and\_Manipulating\_Documents/lesson/5f36bd3e04e9ff98b3e7eed0/lecture
37. Dzsquared. (n.d.). *Tutorial: Design your first relational database using SSMS - azure SQL database*. Tutorial: Design your first relational database using SSMS - Azure SQL Database | Microsoft Docs. Retrieved May 2, 2022, from <https://docs.microsoft.com/en-us/azure/azure-sql/database/design-first-database-tutorial?view=azuresql>