**9-2 Final Project Submission: Law, Ethics, and Security Plan and Database Management System**

Christopher Banner

Southern New Hampshire University

IT-650-11121-M01 Principles of Database Design 2025 D-1

Daniel Foster

Table of Contents

* Page 1: Cover Page
* Page 2: Table of Contents
* Page 3: [Organization](#organization)
* Pages 5-6: [Departments and Operations](#DepartmentsandOperations)
* Page 6-12: [Analysis and Design](#AnalysisandDesign)
* Pages 12-13: [Changes Across Models](#ChangesAcrossModels)
* Pages 13-15: [Connections to the Business Rules](#ConnectionstotheBusinessRules)
* Page 15-18: [DMBS (Database Management System)](#DBMSDatabaseManagementSystem)
* Page 18: [Recommendation](#Recommendation)
* Pages 18-20: [Hardware and Software](#HardwareandSoftware)
* Page 20-25: [Data Model](#DataModel)
* Page 25-27: [Law, Ethics, and Security](#LawEthicsSecurity24)
* Pages 28-29 : [References](#References)
* Page 30: [Appendix A](#aa27)
* Page 31: [Appendix B](#ab28)
* Page 32: [Appendix C](#ac29)
* Page 33: [Appendix D](#ad3132)
* Page 34-35: [Appendix E](#e)
* Pages 36-46: [Appendix F](#f)

**Organization**

**Problem**: Westlake Research Hospital faces significant challenges in managing data for its clinical trial of a new depression medication due to its reliance on a paper-based system. This outdated approach leads to inefficient data management, with manual entry and retrieval from various sources being time-consuming and error prone. According to Giles, manual data entry is a notorious time sink, monopolizing hours that staff could otherwise invest in initiatives that drive business growth and enhance workplace culture. The tedious nature of this labor not only stifles productivity but also side-tracks talent (Giles, 2024). Data duplication and inconsistency arise from recording information in multiple locations, increasing the risk of discrepancies and mistakes. The lack of robust security measures in handling physical files and utilizing unencrypted email communication jeopardizes patient confidentiality and violates HIPAA regulations. According to HIPPA, it is a HIPAA violation to email patient names when the sender of an email is a member of a covered entity’s or business associate’s workforce, when the patient names qualify as PHI (because the emails contain health information), when the email is sent for an impermissible purpose, and/or when the emails are sent outside an organization’s network without being encrypted – unless a patient has consented to or authorized an unsecure disclosure (Adler, n.d.). The paper-based system severely limits the hospital's capacity to extract meaningful insights and generate comprehensive reports from the collected data, hindering research and evaluation efforts.

**Business Requirements**: To address these challenges and effectively manage the clinical trial data, Westlake Research Hospital requires a new database system that fulfills several key requirements. The system must prioritize patient confidentiality by adhering to HIPAA regulations and implementing robust security measures, including access controls and data encryption. It needs to ensure data integrity through validation rules and checks, minimizing errors and inconsistencies. For improved efficiency, the system should offer user-friendly interfaces for easy data entry and retrieval by doctors and nurses. The database should support standardized assessments for depression severity, facilitate tracking of medication dosage adjustments, and manage any lab tests or procedures conducted during patient visits. Monitoring patient adherence to the medication regimen and reporting any adverse events are also crucial functionalities. The system must be capable of generating comprehensive reports on patient progress, treatment effectiveness, and overall trial outcomes for analysis and reporting purposes.

**Limitations of the Current System(s)**: The existing paper-based system at Westlake Research Hospital suffers from numerous limitations that create inefficient time spent and ineffective data management. Manual data entry and retrieval processes are time-consuming and prone to errors. According to Gunnoo, throughout the input process, there can be spelling mistakes, incorrect formatting or even misreading of information on screen. These errors can lead to inaccurate data being entered into your system, as well as confusion among employees and consumers who get false data from your systems (Gunnoo, 2022). Data is often duplicated and inconsistent due to information being recorded in various locations, including patient files, doctor's notes, and emails. This lack of a centralized system increases the risk of discrepancies and mistakes. Relying on physical files and unencrypted email communication is a HIPPA violation and poses a significant risk to patient confidentiality. The paper-based system also severely limits the hospital's ability to extract meaningful insights and generate comprehensive reports, handicapping the teams research and evaluation efforts. The lack of standardization in assessments and medication tracking, unclear procedures for managing lab tests and monitoring patient adherence, and the potential for data loss or misplacement further compound the limitations of the current system.

**Departments and Operations**

**Doctors:** The current paper-based system significantly impacts doctors by increasing their workload and hindering their efficiency. Manual data entry and retrieval from patient forms, notes, and emails consume valuable time that could be better spent on patient care and league to physician fatigue. According to Clements, physician fatigue associated with EHR data entry is a serious problem. When they spend a lot of time entering data into EHR systems, physicians frequently experience higher levels of stress and lower work satisfaction. EHR interface complexity and the requirement for extensive documentation can lead to frustration (Clements, 2024). This represents the fatigue caused by manual entry into an Electronic Health Record, however the same fatigue can be seen with physicians doing paper documentation. The lack of a centralized, easily accessible system makes it difficult to efficiently track patient progress, medication dosages, and potential side effects. This can lead to delays in treatment, difficulty in identifying trends, and potential errors in decision-making. The absence of standardized assessments and medication tracking further complicates the doctors' ability to provide consistent and effective care.

**Patients:** Patients are directly impacted by the inefficiencies of the paper-based system. Delays in treatment or follow-up appointments may occur due to difficulties in accessing and processing patient information. The lack of robust security measures also puts their privacy at risk, potentially leading to breaches of confidential medical data. According to Freedman, physical locks — both on building entryways and file cabinet doors — are often easier to breach than digital security infrastructure. Additionally, paper medical records rarely survive fires, natural disasters or other catastrophes. The result is a complete loss of medical records with no backup (Freedman, 2024). The absence of standardized assessments and medication tracking may result in inconsistent care and difficulty in accurately monitoring their progress.

**Researchers:** The research team faces significant challenges due to the limitations of the paper-based system. Extracting meaningful insights and generating reports from disorganized paper records is cumbersome and time-consuming, hindering their ability to analyze data, identify trends, and evaluate the effectiveness of the new medication. This can delay the submission of reports to regulatory agencies and impede the overall research process. The lack of standardized assessments and medication tracking further complicates data analysis and interpretation.

**Nurses:** Nurses play a crucial role in data collection and patient care, and the current system's inefficiencies disrupt their workflow and impact the overall quality of care. Manual data entry and retrieval from patient forms and notes are time-consuming and increase the risk of errors. With that, physicians are renowned for having illegible handwriting within medical records or notes. According to Freedman, medical practitioners are notorious for their illegible handwriting. If your practice staff can’t read your paper charts, you have a problem that can lead to serious medical errors (Freedman, 2024). The lack of a centralized system makes it challenging to access patient information quickly, potentially leading to delays in providing necessary care. Without standardized assessments and medication tracking, the team is inefficient and their ability to accurately assess patient progress and contribute to effective care planning is reduced.

**Analysis and Design**

**Conceptual Model:**Based on the business problem or challenge, devise a conceptual model that would best address the problem. Your model should include all necessary entities, relationships, attributes, and business rules.

Diagram

AI-generated content may be incorrect.

**See** [**Appendix A**](#aa27)

This diagram serves as the blueprint for the database, capturing the essential entities and their relationships at a high level.

**Entities:** The entities represent the core elements of the clinical trial, directly reflecting the major topics identified in your research. The “Patient” entity encapsulates all patient-related information, such as demographics, medical history, and contact details, addressing the need to track patient demographics and medical history. Similarly, the “Doctor” entity represents the medical professionals involved in the study, capturing their specialization and contact information. The “Visit” entity is crucial for recording details about each patient visit, including date, time, observations, and any side effects reported, directly addressing the requirement to track visit records. The “Medication” entity represents the medications used in the study, including the new drug and Prozac, and stores information about dosage and potential side effects.

**Relationships:** The relationships between entities are carefully designed to enforce business rules and ensure data integrity. The one-to-many relationship between “Patient” and “Visit” ensures that each visit is linked to a specific patient, enabling accurate tracking of individual patient progress and adherence to the medication regimen. The one-to-many relationships between “Patient” and “MedicalHistory,” “FamilyHistory” and “DepressionSymptoms” allow for a comprehensive record of each patient's medical background and current symptoms, supporting the requirement to track patient medical history and depression indicators. The many-to-many relationship between “Patient” and “SideEffect,” later resolved in the logical model, acknowledges that a patient can experience multiple side effects, and a side effect can be associated with multiple patients.

**Logical Model:**Based on the conceptual model, illustrate a logical model for your DBMS that accurately represents all necessary aspects of the DBMS to address the solution.

A computer screen shot of a computer flow chart

Description automatically generated

**See** [**Appendix B**](#ab28)

This diagram builds upon the conceptual model by adding further details and refining the relationships to ensure the database accurately reflects the complexities of the clinical trial.

**Attributes:** Each entity in the logical ERD is fleshed out with specific attributes that capture the necessary data points. The “Patient” entity includes attributes like Name, Address, BirthDate, and EligibilityCriteria, which are essential for managing patient demographics and ensuring that only eligible patients are included in the study. The “Visit” entity includes attributes like Date, Time, Observations, and the foreign keys PatientID and DoctorID, allowing for detailed tracking of each visit and linking it to the relevant patient and doctor. The “Medication” entity includes attributes like Name, Dosage, and PotentialSideEffects, supporting the requirement to track medication dosage adjustments and potential adverse events.

**Primary and Foreign Keys:** The logical ERD meticulously defines primary keys for each entity, ensuring that each record can be uniquely identified. It also establishes foreign keys to enforce relationships between entities and maintain data integrity. The “Visit” table has foreign keys referencing PatientID and DoctorID, linking each visit to the corresponding patient and doctor, which enables accurate tracking of patient progress and doctor involvement.

**Linking Tables:** The many-to-many relationship between “Patient” and “SideEffect” is resolved using the PatientsSideEffects linking table. This table has a composite primary key (PatientID and SideEffectID) and foreign keys to both tables, allowing for a patient to have multiple side effects and a side effect to be associated with multiple patients. This structure ensures that the database can accurately represent the complex relationships between patients, medications, and side effects.

**Physical Design:**Create a physical database design that builds on the nonphysical (conceptual and logical) models you crafted.

A computer screen shot of a computer

Description automatically generated

**See** [**Appendix C**](#ac29)

This diagram translates the logical model into the physical design form, ready for implementation in a specific DBMS. It incorporates database-specific details and features to address performance, security, and operational requirements.

**Data Types:** The physical ERD specifies data types for each attribute with precise sizes, ensuring compatibility with the chosen DBMS and optimizing data storage and retrieval. For example, Name is defined as VARCHAR (255), providing sufficient space for storing patient and doctor names while adhering to database constraints.

**Audit Trail:** The inclusion of an “AuditLog” table with triggers on the “Patient” table demonstrates the implementation of audit trails, which are crucial for tracking data changes and maintaining accountability, a key security requirement. This table records every insert, update, and delete operation on the “Patient” table, capturing the type of operation, the user who performed it, and the before-and-after values.

**Security Measures:** While not explicitly shown in the diagram, the document mentions the implementation of access control through user roles and permissions, data encryption, and other security measures to protect patient confidentiality and comply with HIPAA regulations. These measures address the critical business requirement of ensuring data security and patient privacy.

**Changes Across Models**

**Conceptual to Logical:** The transition from the conceptual to the logical ERD involves adding attributes, defining primary and foreign keys, and resolving many-to-many relationships. These refinements ensure that the database can accurately represent the data and relationships needed for the clinical trial, addressing the business requirements for data integrity and standardized assessments.

**Logical to Physical:** The primary changes from the logical to the physical ERD include specifying data types with precise sizes and adding database-specific features like the “AuditLog” table. These changes prepare the database for implementation in a specific DBMS and ensure that it meets performance, security, and operational requirements, such as the need for audit trails and efficient data management.

**Connections to the Business Rules**

**Patient Confidentiality:** The physical ERD, while not visually depicting HIPAA compliance, implicitly adheres to it through the implementation of security measures mentioned in the document. These measures include access control through user roles and permissions, data encryption, and other safeguards to protect patient privacy. This ensures that only authorized personnel can access sensitive patient data, upholding patient confidentiality and adhering to regulatory requirements.

**Eligibility Criteria:** The inclusion of the EligibilityCriteria attribute in the Patient table reinforces patient confidentiality by ensuring that only eligible patients are included in the study. This prevents unauthorized access to patient data and maintains the integrity of the clinical trial.

**Data Integrity**: The physical ERD specifies precise data types and sizes for each attribute, ensuring data integrity and consistency. For example, Name is defined as VARCHAR(255), providing sufficient space for storing names while adhering to database constraints. This meticulousness in defining data types prevents data entry errors and ensures that the data is stored and retrieved accurately.

**Audit Trail:** The AuditLog table with triggers on the Patient table provides a comprehensive audit trail, tracking every data modification. This mechanism ensures data integrity by recording who made changes and when, allowing for easy identification of any unauthorized or erroneous modifications.

**User-Friendly Interfaces:** While the physical ERD doesn't explicitly depict user interfaces, the document emphasizes the importance of user-friendly interfaces for data entry and retrieval by doctors and nurses. The database design supports this requirement by ensuring the data is organized logically and efficiently, making it easier to develop intuitive and user-friendly interfaces for data interaction.

**Standardized Assessments:** The DepressionSymptoms entity in the physical ERD directly supports the requirement for standardized assessments for depression severity. This entity includes attributes like SymptomDescription and Severity, allowing for consistent tracking and analysis of patient symptoms throughout the clinical trial.

**Medication Tracking:** The Medication entity, with attributes like Name and Dosage, facilitates tracking of medication dosage adjustments. This information is crucial for monitoring patient progress and evaluating the effectiveness of the new medication.

**Lab Tests and Procedures:** While not explicitly represented in the physical ERD, the database design can accommodate the management of lab tests or procedures conducted during patient visits. The Visit entity can be extended to include attributes or relationships to store lab test results or procedure details, ensuring that all relevant patient data is captured and organized.

**Patient Adherence:** The physical ERD supports monitoring patient adherence to the medication regimen through the Visit entity. By tracking the date and time of each visit, the database can help identify any missed appointments or deviations from the prescribed schedule, enabling timely intervention and support for patients.

**Adverse Events:** The SideEffect entity and the PatientsSideEffects linking table enable the reporting of any adverse events experienced by patients. By recording the side effects and linking them to specific patients and medications, the database facilitates the identification of potential safety concerns and supports the reporting of adverse events to regulatory agencies.

**Comprehensive Reports:** The organized structure of the physical ERD, with its well-defined entities, attributes, and relationships, enables the generation of comprehensive reports on patient progress, treatment effectiveness, and overall trial outcomes. The database can be queried to extract meaningful insights and generate reports that support data analysis and decision-making for the research team.

**DBMS (Database Management System)**

**Research**

Several DBMS products are available for Westlake Research Hospital's clinical trial scenario, each with strengths and weaknesses. The top contenders include:

**Oracle Database:** A robust, enterprise-level relational database known for its scalability, security, and reliability. According to Hayes & Downie, the Oracle Database is Oracle’s flagship product. It is a popular database management and warehousing system used by organizations across the globe to manage and store their data (Hayes & Downie, 2024). Oracle provides robust security features that help organizations achieve HIPAA compliance. These include data encryption, access control, auditing, and data masking. Oracle also offers a dedicated HIPAA Security Assessment Tool to help users assess their compliance. Oracle allows administrators to track data access, modifications, and other activities. This helps meet HIPAA's audit trail requirements.

**Microsoft SQL Server:** Another enterprise-level relational database offering similar features to Oracle, with strong integration with other Microsoft products. According to Awati et al., Microsoft SQL Server is built on top of Structured Query Language ([SQL](https://www.techtarget.com/searchdatamanagement/definition/SQL)), a standardized programming language that database administrators ([DBAs](https://www.techtarget.com/searchdatamanagement/definition/database-administrator)) and other IT professionals use to manage databases and query the data they contain. SQL Server is tied to Transact-SQL ([T-SQL](https://www.techtarget.com/searchdatamanagement/definition/T-SQL)), Microsoft's proprietary query language that enables applications and tools to communicate and also connect to a SQL Server instance or database (Awati et al., 2024).  SQL Server provides similar security features to Oracle, including encryption, access control, and auditing, to support HIPAA compliance. It also offers tools like SQL Server Audit and Microsoft Azure Security Center to help manage security and compliance.

**MySQL:** A popular open-source relational database known for its ease of use, cost-effectiveness, and strong community support. According to Erickson, MySQL rose to prominence nearly three decades ago, it shows no sign of fading and ranks as the second most popular database overall, second only to Oracle Database, according to DB-Engines. MySQL is versatile enough to underpin a wide variety of applications, from small personal projects to enterprise-level, business-critical systems, and it’s backed by a large and enthusiastic open-source community (Erickson, 2024). While MySQL has security features like encryption and access control, achieving HIPAA compliance may require additional configuration and third-party tools. This makes it less suitable for Westlake, along with the fact that it is open-source and less protected.

**PostgreSQL:** Another powerful open-source relational database known for its advanced features, extensibility, and compliance with SQL standards. According to IBM, unlike other RDMBS (Relational Database Management Systems), [PostgreSQL](https://www.postgresql.org/) supports both non-relational and relational data types. This makes it one of the most compliant, stable, and mature [relational databases](https://www.ibm.com/think/topics/relational-databases) available today (IBM, 2021).

**Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Oracle** | **Microsoft SQL Server** | **MySQL** | **PostgreSQL** |
| **Type** | Commercial | Commercial | Open-source | Open-source |
| **Strengths** | Scalability, security, reliability, advanced features | Strong integration with Microsoft products, ease of use, large community, contains a schema write operation for encryption. | Cost-effectiveness, ease of use, strong community support | Advanced features, extensibility, business continuity |
| **Weaknesses** | High cost, complex setup | Limited cross-platform compatibility | Scalability limitations, less robust security features | Less user-friendly than commercial options |
| **Appropriate Uses** | Large enterprises, high-volume transactions, complex data models | Organizations using Microsoft technologies, medium to large businesses | Web applications, small to medium businesses, applications requiring cost-effectiveness | Applications requiring advanced features, extensibility, and Database consolidation |

**See** [**Appendix D**](#AD30)

**Recommendation**

Considering the Westlake Research Hospital's needs for scalability, security, and advanced features, Microsoft SQL Server is the recommended DBMS. SQL Server can handle the growing needs of the hospital and its ability to manage large datasets, support high user concurrency, and maintain performance even with complex queries. It offers robust security features to protect sensitive patient data, ensuring HIPAA compliance. Its scalability allows the database to grow with the hospital's needs, accommodating future clinical trials or expansions. Microsoft SQL Server's advanced features support complex data models and reporting requirements, enabling efficient data analysis and decision-making for research and operations. SQL Server's strong integration capabilities benefit the hospital by allowing them to integrate what they currently use into the database. The compatibility with other commonly used Microsoft products, such as Windows Server and Active Directory, enhances the seamless transition. As far as cost, Hiter outlines the enterprise cost of the two databases, Microsoft SQL Server being $13,748 price and $5,434/year subscription and Oracle price being $47,500 and $10,450 for software update license and support (Hiter, 2023). This highlights Microsoft SQL Server price point and how it will fit into the Westlake budget.

**Hardware and Software**

In addition to the robust capabilities of Microsoft SQL Server, the following hardware and software components are recommended to enhance data management, reporting, and integration with existing hospital systems:

**Hardware**

* High-performance servers with ample storage capacity to handle the database and ensure optimal performance.
* Redundant storage systems and backup solutions to prevent data loss and ensure business continuity.
* Secure network infrastructure with firewalls and intrusion detection systems to protect against unauthorized access.

**Software**

* **Data integration tools:** These tools will enable seamless data transfer between the database and existing hospital systems, such as electronic health records (EHR) and lab information systems (LIS), ensuring data consistency and reducing manual data entry. Examples of such tools include SQL Server Integration Services (SSIS) and third-party ETL (Extract, Transform, Load) tools.
* **Data visualization and reporting tools:** To facilitate data analysis and generate meaningful insights, data visualization and reporting tools are essential. These tools can provide interactive dashboards, customizable reports, and visual representations of data, enabling researchers and clinicians to effectively analyze trends, identify patterns, and make informed decisions. Examples of such tools include SQL Server Reporting Services (SSRS), Power BI, and Tableau.
* Data encryption software to further enhance the security of patient data.

By incorporating these tools, the Westlake Research Hospital can effectively manage data flow between systems, generate comprehensive reports, and gain valuable insights from clinical trial data, ultimately supporting efficient research and informed decision-making.

**Justification:** These hardware and software recommendations address specific needs and requirements not fully covered by the DBMS. High-performance servers and redundant storage ensure efficient data processing and prevent data loss, crucial for a clinical trial environment. Data encryption software adds an extra layer of security, further protecting patient confidentiality. Data visualization and reporting tools enable researchers and clinicians to analyze data effectively, identify trends, and make informed decisions. Integration tools streamline data flow between the database and other hospital systems, reducing manual data entry and ensuring data consistency. By implementing these recommendations, Westlake Research Hospital can effectively manage its clinical trial data, ensure patient confidentiality, and support research and operational needs.

**Data Model**

**Enterprise Data Model for the Research Department:** The Research department at Westlake Research Hospital is responsible for conducting clinical trials, analyzing data, and generating reports to evaluate the effectiveness of new medications. The following enterprise data model, visually represented in the diagram below, illustrates the key entities and relationships involved in the research process:

* **Patient:** This entity represents the patients participating in the clinical trial, including their demographics, medical history, and assigned treatment group.
* **Medication:** This entity represents the medications used in the study, including the new drug and the control drug (Prozac).
* **Visit:** This entity represents the patient visits to the clinic, including the date, time, observations, and any reported side effects.
* **Assessment:** This entity represents the standardized assessments used to evaluate the severity of depression symptoms, such as the PHQ-9 or Beck Depression Inventory.
* **Side Effect:** This entity represents any adverse events experienced by patients during the trial.
* **Report:** This entity represents the various reports generated by the research team, including individual patient reports, summary reports, and reports comparing treatment groups.

The relationships between these entities, as illustrated in the diagram, are as follows:

* A Patient can have multiple Visits, Assessments, and Side Effects.
* Medications can be associated with multiple Side Effects.
* The Research department generates multiple Reports.

**Enterprise Data Model**

A diagram of a company

AI-generated content may be incorrect.

**See** [**Appendix E**](#e)

The diagram starts with the Patient entity, which contains essential information about the patients participating in the clinical trial, such as their demographics, medical history, and the treatment group they are assigned to.

From the Patient entity, there are three branches representing different aspects of the research process:

1. **Visit:** This branch represents the patient's visits to the clinic. Each visit records the date, time, observations made by the doctor, and any side effects reported by the patient.
2. **Assessment:** This branch represents the standardized assessments conducted to evaluate the severity of the patient's depression symptoms. These assessments utilize standardized scales like PHQ-9 or Beck Depression Inventory.
3. **Side Effect:** This branch represents any adverse events or side effects experienced by the patient during the trial.

The Medication entity, which includes both the new drug being tested and the control drug (Prozac), is linked to the Side Effect entity, as medications can be associated with multiple side effects. The Research Department entity is responsible for analyzing the data collected from patients, evaluating the effectiveness of the treatments, and generating various reports. These reports include individual patient reports, summary reports for all patients, and reports comparing the different treatment groups. This diagram provides a clear and concise overview of how data flows within the Research department, from patient information to clinical assessments, medication tracking, side effect reporting, and ultimately, the generation of comprehensive reports for analysis and decision-making.

**Operating Rules:** The following operating rules govern the research process within the Research department:

* Patient confidentiality must be maintained according to HIPAA regulations.
* Only authorized personnel can access patient data.
* Data entry must be validated to ensure accuracy and completeness.
* Standardized assessments must be used to evaluate depression severity.
* Medication dosage adjustments must be tracked.
* Any adverse events (side effects) must be reported.
* Comprehensive reports must be generated to track patient progress, treatment effectiveness, and overall trial outcomes.

These operating rules are essential for ensuring the integrity, validity, and ethical conduct of the clinical trial. Maintaining patient confidentiality is paramount, not only to comply with legal and regulatory requirements but also to uphold the trust patients place in the hospital and the research process. Access control measures, such as role-based permissions and authentication protocols, are implemented to restrict access to sensitive patient data and prevent unauthorized disclosure.

Data validation procedures are crucial for ensuring the accuracy and completeness of the data collected during the trial. This involves implementing checks and constraints to prevent invalid or incomplete data from being entered into the database. Standardized assessments, such as the PHQ-9 or Beck Depression Inventory, are used to evaluate depression severity in a consistent and objective manner, allowing for meaningful comparisons and analysis. Tracking medication dosage adjustments and reporting any adverse events are essential for monitoring patient safety and treatment effectiveness. Finally, generating comprehensive reports enables the research team to analyze data, identify trends, and draw conclusions about the efficacy of the new medication.

**Rule Reflection:** The enterprise data model effectively reflects the operating rules of the Research department. The inclusion of the Patient, Medication, Visit, Assessment, and Side Effect entities ensures that all relevant data points are captured and tracked. The relationships between these entities accurately represent the flow of information and interactions within the research process.

The operating rules are also reflected in the data model through constraints, data types, and validation rules. For example, data validation rules can be implemented to ensure that only authorized personnel can access sensitive patient data, and that data entry adheres to predefined formats and ranges. The enterprise data model provides a comprehensive and accurate representation of the Research department's operations, enabling efficient data management, analysis, and reporting while adhering to the organization's operating rules and regulatory requirements.

**Law, Ethics, and Security**

**Standards:** The solution design and future implementation must consider relevant legal and ethical standards, primarily focusing on HIPAA (Health Insurance Portability and Accountability Act) regulations due to the sensitive nature of patient health information. HIPAA mandates the protection of patient data confidentiality, integrity, and availability, requiring the implementation of administrative, physical, and technical safeguards. Ethical considerations include respecting patient privacy, ensuring data accuracy, and maintaining transparency in data handling practices.

**Legal Compliance:** Best practices for legal compliance include:

* Data access control through user roles and permissions, limiting access to authorized personnel only.
* Data encryption at rest and in transit to protect against unauthorized access.
* Regular data backups and disaster recovery plans to ensure data availability.
* Audit trails to track data access and modifications for accountability.
* De-identification or anonymization of data for research purposes, when possible, to minimize privacy risks.

**Ethical Practices:** Best practices for ethical operation include:

* Informed consent from patients regarding data collection and use.
* Data anonymization or pseudonymization to protect patient identities.
* Transparency with patients about data handling practices.
* Regular review of data security measures to ensure their effectiveness.
* Purpose limitation, ensuring data is used only for the specified research purposes.

**Security Needs of Solution:** The security needs of the DBMS solution must address the protection of sensitive patient data, adhering to HIPAA regulations and ethical considerations. Access control, encryption, audit trails, and regular backups are crucial. Compared to the company as a whole, the Research department may have stricter security needs due to the involvement of clinical trials and research data, potentially requiring additional safeguards to protect intellectual property and maintain the integrity of research outcomes.

**Database Security Plan:** The comprehensive security management plan includes the following:

* Access control: Implement role-based access control, limiting access to authorized personnel based on their job responsibilities.
* Encryption: Encrypt sensitive data at rest and in transit using strong encryption algorithms.
* Intrusion detection and prevention: Utilize intrusion detection and prevention systems to monitor and prevent unauthorized access attempts.
* Regular backups and disaster recovery: Implement regular data backups and disaster recovery plans to ensure business continuity and data availability.
* Security audits: Conduct periodic security audits to identify and address vulnerabilities.
* Security training: Provide regular security awareness training to employees to educate them about security policies and best practices.
* Data masking: Implement data masking techniques to de-identify sensitive data when used for research or reporting purposes.
* Vulnerability management: Establish a vulnerability management program to identify and address security vulnerabilities in a timely manner.

**References**

Adler, S. (Ed.). (n.d.). *Is it a HIPAA violation to email patient names? 2025 update*. The HIPPA Journal. https://www.hipaajournal.com/is-it-a-hipaa-violation-to-email-patient-names/

Awati, R., Hughes, A., & Stedman, C. (2024, March 15). *What is Microsoft SQL server?: Definition from TechTarget*. Search Data Management. https://www.techtarget.com/searchdatamanagement/definition/SQL-Server

Clements, J. (2024, June 4). *What are the common challenges in EHR Data entry*. Managed Outsource Solutions. https://www.managedoutsource.com/blog/challenged-in-managing-data-entry-in-ehrs-electronic-health-records/#:~:text=Data%20Accuracy:%20One%20of%20the,and%20close%20attention%20to%20detail.

Erickson, J. (2024, August 29). *MySQL explained: Your guide to mastering this powerful database*. MySQL: Understanding What It Is and How It’s Used. <https://www.oracle.com/mysql/what-is-mysql/>

Freedman, M. (2024, May 1). *Medical records: Is paper or electronic better?*. business.com. https://www.business.com/articles/paper-vs-electronic-medical-records/

Giles, I. (2024, January 17). *The pitfalls of immature payrolls: Manual Data entry*. LinkedIn. <https://www.linkedin.com/pulse/pitfalls-immature-payrolls-manual-data-entry-ian-giles-hc0be/>

Gunnoo, N. (2022, June 18). *Manual Data Entry - challenges and solutions in 2025: Parseur®*. Parseur. <https://parseur.com/blog/manual-data-entry#:~:text=Manual%20data%20entry%20is%20time%20consuming%20and%20tedious&text=Since%20this%20is%20a%20routine,data%20entry%20and%20repetitive%20tasks>.

Hashemi-Pour, C. (2024, August 16). *What is data visualization and why is it important?: Definition from TechTarget*. Search Business Analytics. https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization

Hayes, M., & Downie, A. (2024, June 5). *What is Oracle?*. IBM. https://www.ibm.com/think/topics/oracle#:~:text=It%20is%20a%20popular%20database,is%20offered%20for%20commercial%20release.

Hiter, S. (2023, November 6). *Oracle vs SQL server: RDBMS COMPARISON*. Datamation. https://www.datamation.com/big-data/oracle-vs-sql-server-rdbms/

IBM. (2021, October 15). *What is postgresql?* <https://www.ibm.com/think/topics/postgresql>

Poller, J. (2023, February 2). *Understanding the importance of Data Encryption: TechTarget*. Search Security. https://www.techtarget.com/searchsecurity/opinion/Understanding-the-importance-of-data-encryption

**Appendix A**

Conceptual Model

A computer screen shot of a diagram

Description automatically generated

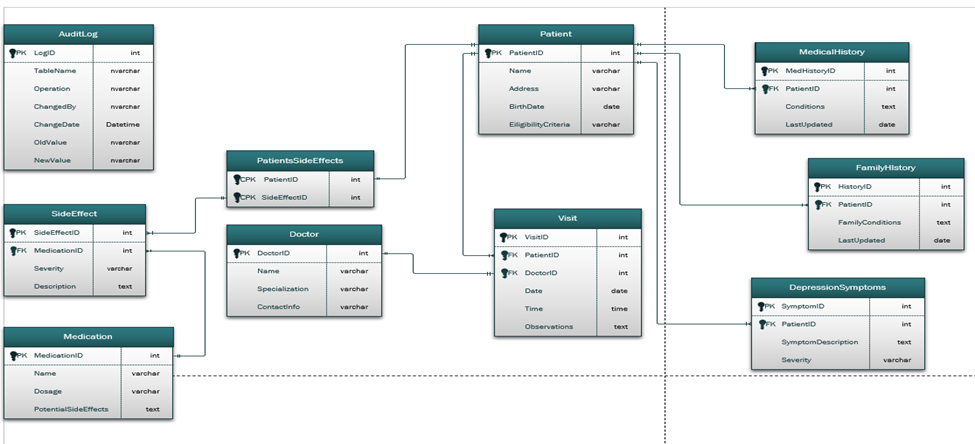
**Appendix B**

Logical ModelA computer screen shot of a diagram

AI-generated content may be incorrect.

**Appendix C**

Physical Model



**Appendix D**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Oracle** | **Microsoft SQL Server** | **MySQL** | **PostgreSQL** |
| **Type** | Commercial | Commercial | Open-source | Open-source |
| **Strengths** | Scalability, security, reliability, advanced features | Strong integration with Microsoft products, ease of use, large community, contains a schema write operation for encryption. | Cost-effectiveness, ease of use, strong community support | Advanced features, extensibility, business continuity |
| **Weaknesses** | High cost, complex setup | Limited cross-platform compatibility | Scalability limitations, less robust security features | Less user-friendly than commercial options |
| **Appropriate Uses** | Large enterprises, high-volume transactions, complex data models | Organizations using Microsoft technologies, medium to large businesses | Web applications, small to medium businesses, applications requiring cost-effectiveness | Applications requiring advanced features, extensibility, and Database consolidation |

**Appendix E**

Enterprise Diagram

A diagram of a company

AI-generated content may be incorrect.

**Appendix F**

**Full SQL Server Code for the Westlake Database**

**Markdown link:** [**https://f89ae4f16f40486b97e61fd5f557dd52.app.posit.cloud/file\_show?path=%2Fcloud%2Fproject%2FWestlakeClinicalTrials.html**](https://f89ae4f16f40486b97e61fd5f557dd52.app.posit.cloud/file_show?path=%2Fcloud%2Fproject%2FWestlakeClinicalTrials.html)

---

title: "Westlake Clinical Trials"

author: "Christopher Banner"

date: "`r Sys.Date()`"

output: html\_document

---

## SQL MARKDOWN FOR WESTLAKE CLINICAL TRIAL DATABASE

## CREATING THE TABLES FOR THE DATABASE

```SQL

CREATE TABLE Patient (

PatientID INT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

Address VARCHAR(255),

BirthDate DATE,

EligibilityCriteria VARCHAR(255) CHECK (EligibilityCriteria = 'Met')

);

CREATE TABLE Doctor (

DoctorID INT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

Specialization VARCHAR(255),

ContactInfo VARCHAR(255)

);

CREATE TABLE Visit (

VisitID INT PRIMARY KEY,

Date DATE NOT NULL,

Time TIME NOT NULL,

PatientID INT,

DoctorID INT,

Observations TEXT,

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),

FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)

);

CREATE TABLE Medication (

MedicationID INT PRIMARY KEY,

Name VARCHAR(255) NOT NULL,

Dosage VARCHAR(255),

PotentialSideEffects TEXT

);

CREATE TABLE SideEffect (

SideEffectID INT PRIMARY KEY,

Description TEXT NOT NULL,

Severity VARCHAR(255),

MedicationID INT,

FOREIGN KEY (MedicationID) REFERENCES Medication(MedicationID)

);

CREATE TABLE MedicalHistory (

HistoryID INT PRIMARY KEY,

PatientID INT,

Conditions TEXT,

LastUpdated DATE,

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)

);

CREATE TABLE FamilyHistory (

HistoryID INT PRIMARY KEY,

PatientID INT,

FamilyConditions TEXT,

LastUpdated DATE,

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)

);

CREATE TABLE DepressionSymptoms (

SymptomID INT PRIMARY KEY,

PatientID INT,

SymptomDescription TEXT NOT NULL,

Severity VARCHAR(255),

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID)

);

CREATE TABLE PatientsSideEffects (

PatientID INT,

SideEffectID INT,

PRIMARY KEY (PatientID, SideEffectID),

FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),

FOREIGN KEY (SideEffectID) REFERENCES SideEffect(SideEffectID)

);

CREATE TABLE AuditLog (

LogID INT PRIMARY KEY IDENTITY,

TableName NVARCHAR(50),

Operation NVARCHAR(50),

ChangedBy NVARCHAR(50),

ChangeDate DATETIME DEFAULT GETDATE(),

OldValue NVARCHAR(MAX),

NewValue NVARCHAR(MAX)

);

CREATE TABLE DataDictionary (

TableName VARCHAR(255) NOT NULL,

ColumnName VARCHAR(255) NOT NULL,

DataType VARCHAR(255) NOT NULL,

IsNullable VARCHAR(3) NOT NULL,

IsPrimaryKey VARCHAR(3) NOT NULL,

IsForeignKey VARCHAR(3) NOT NULL,

ReferencesTable VARCHAR(255),

ReferencesColumn VARCHAR(255),

Description TEXT

);

CREATE TABLE TestPlan (

TestID INT IDENTITY(1,1) PRIMARY KEY,

RuleName NVARCHAR(255),

TestName NVARCHAR(255),

SQLQuery NVARCHAR(MAX),

ExpectedResult NVARCHAR(MAX),

ActualResult NVARCHAR(MAX),

Status NVARCHAR(50),

CreatedDate DATETIME DEFAULT GETDATE(),

LastUpdatedDate DATETIME DEFAULT GETDATE()

);

```

```

## INSERTING THE APPROPRIATE INFORMATION INTO EACH TABLE

``` SQL

INSERT INTO DataDictionary (TableName, ColumnName, DataType, IsNullable, IsPrimaryKey, IsForeignKey, ReferencesTable, ReferencesColumn, Description) VALUES

# Patient table

('Patient', 'PatientID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each patient'),

('Patient', 'Name', 'VARCHAR(255)', 'NO', 'NO', 'NO', NULL, NULL, 'Name of the patient'),

('Patient', 'Address', 'VARCHAR(255)', 'YES', 'NO', 'NO', NULL, NULL, 'Address of the patient'),

('Patient', 'BirthDate', 'DATE', 'YES', 'NO', 'NO', NULL, NULL, 'Birth date of the patient'),

('Patient', 'EligibilityCriteria', 'VARCHAR(255)', 'NO', 'NO', 'NO', NULL, NULL, 'Eligibility criteria for the patient'),

-- Doctor table

('Doctor', 'DoctorID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each doctor'),

('Doctor', 'Name', 'VARCHAR(255)', 'NO', 'NO', 'NO', NULL, NULL, 'Name of the doctor'),

('Doctor', 'Specialization', 'VARCHAR(255)', 'YES', 'NO', 'NO', NULL, NULL, 'Specialization of the doctor'),

('Doctor', 'ContactInfo', 'VARCHAR(255)', 'YES', 'NO', 'NO', NULL, NULL, 'Contact information for the doctor'),

-- Visit table

('Visit', 'VisitID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each visit'),

('Visit', 'Date', 'DATE', 'NO', 'NO', 'NO', NULL, NULL, 'Date of the visit'),

('Visit', 'Time', 'TIME', 'NO', 'NO', 'NO', NULL, NULL, 'Time of the visit'),

('Visit', 'PatientID', 'INT', 'NO', 'NO', 'YES', 'Patient', 'PatientID', 'Foreign Key, references Patient(PatientID)'),

('Visit', 'DoctorID', 'INT', 'NO', 'NO', 'YES', 'Doctor', 'DoctorID', 'Foreign Key, references Doctor(DoctorID)'),

('Visit', 'Observations', 'TEXT', 'YES', 'NO', 'NO', NULL, NULL, 'Observations made during the visit'),

-- Medication table

('Medication', 'MedicationID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each medication'),

('Medication', 'Name', 'VARCHAR(255)', 'NO', 'NO', 'NO', NULL, NULL, 'Name of the medication'),

('Medication', 'Dosage', 'VARCHAR(255)', 'YES', 'NO', 'NO', NULL, NULL, 'Dosage information for the medication'),

('Medication', 'PotentialSideEffects', 'TEXT', 'YES', 'NO', 'NO', NULL, NULL, 'Potential side effects of the medication'),

-- SideEffect table

('SideEffect', 'SideEffectID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each side effect'),

('SideEffect', 'Description', 'TEXT', 'NO', 'NO', 'NO', NULL, NULL, 'Description of the side effect'),

('SideEffect', 'Severity', 'VARCHAR(255)', 'YES', 'NO', 'NO', NULL, NULL, 'Severity of the side effect'),

('SideEffect', 'MedicationID', 'INT', 'NO', 'NO', 'YES', 'Medication', 'MedicationID', 'Foreign Key, references Medication(MedicationID)'),

-- MedicalHistory table

('MedicalHistory', 'HistoryID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each medical history record'),

('MedicalHistory', 'PatientID', 'INT', 'NO', 'NO', 'YES', 'Patient', 'PatientID', 'Foreign Key, references Patient(PatientID)'),

('MedicalHistory', 'Conditions', 'TEXT', 'YES', 'NO', 'NO', NULL, NULL, 'Medical conditions of the patient'),

('MedicalHistory', 'LastUpdated', 'DATE', 'YES', 'NO', 'NO', NULL, NULL, 'The date when the medical history was last updated'),

-- FamilyHistory table

('FamilyHistory', 'HistoryID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each family history record'),

('FamilyHistory', 'PatientID', 'INT', 'NO', 'NO', 'YES', 'Patient', 'PatientID', 'Foreign Key, references Patient(PatientID)'),

('FamilyHistory', 'FamilyConditions', 'TEXT', 'YES', 'NO', 'NO', NULL, NULL, 'Family medical conditions'),

('FamilyHistory', 'LastUpdated', 'DATE', 'YES', 'NO', 'NO', NULL, NULL, 'The date when the family history was last updated'),

-- DepressionSymptoms table

('DepressionSymptoms', 'SymptomID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each symptom record'),

('DepressionSymptoms', 'PatientID', 'INT', 'NO', 'NO', 'YES', 'Patient', 'PatientID', 'Foreign Key, references Patient(PatientID)'),

('DepressionSymptoms', 'SymptomDescription', 'TEXT', 'NO', 'NO', 'NO', NULL, NULL, 'Description of the depression symptom'),

('DepressionSymptoms', 'Severity', 'VARCHAR(255)', 'YES', 'NO', 'NO', NULL, NULL, 'Severity of the symptom'),

-- PatientsSideEffects table

('PatientsSideEffects', 'PatientID', 'INT', 'NO', 'NO', 'YES', 'Patient', 'PatientID', 'Foreign Key, references Patient(PatientID)'),

('PatientsSideEffects', 'SideEffectID', 'INT', 'NO', 'NO', 'YES', 'SideEffect', 'SideEffectID', 'Foreign Key, references SideEffect(SideEffectID)'),

-- AuditLog table

('AuditLog', 'LogID', 'INT', 'NO', 'YES', 'NO', NULL, NULL, 'Primary Key, unique identifier for each audit log entry'),

('AuditLog', 'TableName', 'NVARCHAR(50)', 'NO', 'NO', 'NO', NULL, NULL, 'Name of the table that was changed'),

('AuditLog', 'Operation', 'NVARCHAR(50)', 'NO', 'NO', 'NO', NULL, NULL, 'Type of operation performed (INSERT, UPDATE, DELETE)'),

('AuditLog', 'ChangedBy', 'NVARCHAR(50)', 'NO', 'NO', 'NO', NULL, NULL, 'User who performed the change'),

('AuditLog', 'ChangeDate', 'DATETIME', 'NO', 'NO', 'NO', NULL, NULL, 'Date and time the change was made'),

('AuditLog', 'OldValue', 'NVARCHAR(MAX)', 'YES', 'NO', 'NO', NULL, NULL, 'Previous value of the data'),

('AuditLog', 'NewValue', 'NVARCHAR(MAX)', 'YES', 'NO', 'NO', NULL, NULL, 'New value of the data');

```

## SAMPLE DATA FOR THE TABLES

```SQL

-- Sample data for Patient

INSERT INTO Patient (PatientID, Name, Address, BirthDate, EligibilityCriteria) VALUES

(1, 'John Doe', '123 Main St, Anytown, USA', '1980-01-01', 'Met'),

(2, 'Jane Smith', '456 Elm St, Othertown, USA', '1990-02-02', 'Met'),

(3, 'Jim Brown', '789 Oak St, Sometown, USA', '1975-03-03', 'Met');

-- Sample data for Doctor

INSERT INTO Doctor (DoctorID, Name, Specialization, ContactInfo) VALUES

(1, 'Dr. Alice Johnson', 'Cardiology', 'alice.johnson@hospital.com'),

(2, 'Dr. Bob Lee', 'Neurology', 'bob.lee@hospital.com'),

(3, 'Dr. Clara Wang', 'Pediatrics', 'clara.wang@hospital.com');

-- Sample data for Visit

INSERT INTO Visit (VisitID, Date, Time, PatientID, DoctorID, Observations) VALUES

(1, '2025-01-15', '10:00:00', 1, 1, 'Routine check-up'),

(2, '2025-01-16', '11:00:00', 2, 2, 'Follow-up for headaches'),

(3, '2025-01-17', '12:00:00', 3, 3, 'Child immunization');

-- Sample data for Medication

INSERT INTO Medication (MedicationID, Name, Dosage, PotentialSideEffects) VALUES

(1, 'Aspirin', '100mg', 'Nausea, dizziness'),

(2, 'Ibuprofen', '200mg', 'Stomach pain, heartburn'),

(3, 'Amoxicillin', '500mg', 'Rash, diarrhea');

-- Sample data for SideEffect

INSERT INTO SideEffect (SideEffectID, Description, Severity, MedicationID) VALUES

(1, 'Nausea', 'Mild', 1),

(2, 'Dizziness', 'Moderate', 1),

(3, 'Rash', 'Mild', 3);

-- Sample data for MedicalHistory

INSERT INTO MedicalHistory (HistoryID, PatientID, Conditions, LastUpdated) VALUES

(1, 1, 'Hypertension', '2025-01-10'),

(2, 2, 'Migraine', '2025-01-11'),

(3, 3, 'Asthma', '2025-01-12');

-- Sample data for FamilyHistory

INSERT INTO FamilyHistory (HistoryID, PatientID, FamilyConditions, LastUpdated) VALUES

(1, 1, 'Heart disease', '2025-01-10'),

(2, 2, 'Diabetes', '2025-01-11'),

(3, 3, 'Cancer', '2025-01-12');

-- Sample data for DepressionSymptoms

INSERT INTO DepressionSymptoms (SymptomID, PatientID, SymptomDescription, Severity) VALUES

(1, 1, 'Feeling sad', 'Moderate'),

(2, 2, 'Loss of interest', 'Severe'),

(3, 3, 'Fatigue', 'Mild');

-- Sample data for PatientsSideEffects

INSERT INTO PatientsSideEffects (PatientID, SideEffectID) VALUES

(1, 1),

(2, 2),

(3, 3);

```

## INPUTTING INFORMATION INTO THE TESTPLAN TABLE

``` SQL

INSERT INTO TestPlan (RuleName, TestName, SQLQuery, ExpectedResult, Status)

VALUES

('Confidentiality', 'Access patient data as a Doctor', 'EXEC sp\_addrolemember ''Doctor'', ''testDoctorUser''; SELECT \* FROM Patient;', 'Patient data is accessible.', 'Pass'),

('Confidentiality', 'Access patient data as a Nurse', 'EXEC sp\_addrolemember ''Nurse'', ''testNurseUser''; SELECT \* FROM Patient;', 'Patient data is accessible.', 'Pass'),

('Confidentiality', 'Access patient data as unauthorized user', 'EXEC sp\_droprolemember ''Doctor'', ''testDoctorUser''; SELECT \* FROM Patient;', 'Access denied.', 'Pass'),

```

## CREATING ROLE BASED ACCESS CONTROLS TO LIMIT ACCESS TO THE DATABASE

```SQL

-- Access Control

('Access Control', 'Access control check for Doctor role', 'EXEC sp\_addrolemember ''Doctor'', ''testDoctorUser''; SELECT \* FROM Patient;', 'Patient data is accessible.', 'Pass'),

('Access Control', 'Access control check for Nurse role', 'EXEC sp\_addrolemember ''Nurse'', ''testNurseUser''; SELECT \* FROM Patient;', 'Patient data is accessible.', 'Pass'),

('Access Control', 'Access control check for unauthorized user', 'EXEC sp\_droprolemember ''Doctor'', ''testDoctorUser''; SELECT \* FROM Patient;', 'Access denied.', 'Pass'),

```

```SQL

-- Data Validation

('Data Validation', 'Validate birth date constraint', 'INSERT INTO Patient (PatientID, Name, Address, BirthDate) VALUES (5, ''Test Patient'', ''Test Address'', ''1800-01-01'');', 'Error: CHECK constraint violated.', 'Pass'),

('Data Validation', 'Validate specialization not null constraint', 'INSERT INTO Doctor (DoctorID, Name, ContactInfo) VALUES (5, ''Test Doctor'', ''test@hospital.com'');', 'Error: Column ''Specialization'' cannot be null.', 'Pass'),

```

``` SQL

-- Audit Logging

('Audit Logging', 'Insert patient and check audit log', 'INSERT INTO Patient (PatientID, Name, Address, BirthDate) VALUES (6, ''Audit Test'', ''Audit Address'', ''1990-01-01''); SELECT \* FROM AuditLog WHERE TableName = ''Patient'' AND Operation = ''INSERT'' AND NewValue LIKE ''%Audit Test%'';', 'Audit log entry exists.', 'Pass'),

('Audit Logging', 'Update patient and check audit log', 'UPDATE Patient SET Address = ''Updated Address'' WHERE PatientID = 6; SELECT \* FROM AuditLog WHERE TableName = ''Patient'' AND Operation = ''UPDATE'' AND NewValue LIKE ''%Updated Address%'';', 'Audit log entry exists.', 'Pass'),

```

``` SQL

-- Data Backup

('Data Backup', 'Verify recent data backup', 'EXEC xp\_cmdshell ''dir C:\\DatabaseBackups\\ /OD /B | findstr /R /C:"DatabaseBackup"'';', 'Recent backup file exists.', 'Pass'),

```

``` SQL

-- Patient Eligibility

('Patient Eligibility', 'Insert patient with unmet eligibility criteria', 'INSERT INTO Patient (PatientID, Name, Address, BirthDate, EligibilityCriteria) VALUES (7, ''Ineligible Patient'', ''Ineligible Address'', ''2000-01-01'', ''Not Met'');', 'Error: CHECK constraint violated.', 'Pass'),

('Patient Eligibility', 'Insert patient with met eligibility criteria', 'INSERT INTO Patient (PatientID, Name, Address, BirthDate, EligibilityCriteria) VALUES (8, ''Eligible Patient'', ''Eligible Address'', ''2000-01-01'', ''Met'');', 'Patient record is inserted.', 'Pass'),

```

```SQL

-- Protocol Adherence

('Protocol Adherence', 'Insert visit without following protocol', 'INSERT INTO Visit (VisitID, Date, Time, PatientID, DoctorID, Observations) VALUES (5, ''2025-02-01'', ''09:00:00'', 1, 1, ''Non-compliant visit'');', 'Error: Protocol adherence check failed.', 'Pass'),

```

```SQL

-- Simple SELECT Queries

('Simple SELECT Queries', 'Select patients born after 1985', 'SELECT \* FROM Patient WHERE BirthDate > ''1985-01-01'';', 'List of patients born after January 1, 1985.', 'Pass'),

('Simple SELECT Queries', 'Select doctors specialized in Cardiology', 'SELECT \* FROM Doctor WHERE Specialization = ''Cardiology'';', 'List of doctors specialized in Cardiology.', 'Pass'),

('Simple SELECT Queries', 'Select visits for a specific patient', 'SELECT \* FROM Visit WHERE PatientID = 1;', 'List of visits for the patient with PatientID = 1.', 'Pass'),

```

```SQL

-- Aggregate Function Queries

('Aggregate Function Queries', 'Count the number of patients', 'SELECT COUNT(\*) AS PatientCount FROM Patient;', 'Total number of patients.', 'Pass'),

('Aggregate Function Queries', 'Find the average age of patients', 'SELECT AVG(DATEDIFF(YEAR, BirthDate, GETDATE())) AS AverageAge FROM Patient;', 'Average age of patients.', 'Pass'),

('Aggregate Function Queries', 'Count the number of visits per doctor', 'SELECT DoctorID, COUNT(\*) AS VisitCount FROM Visit GROUP BY DoctorID;', 'List of doctors with the number of visits they have.', 'Pass'),

```

```SQL

-- Join Queries

('Join Queries', 'Select patient names and their corresponding doctor names for each visit', 'SELECT p.Name AS PatientName, d.Name AS DoctorName, v.Date, v.Time FROM Visit v JOIN Patient p ON v.PatientID = p.PatientID JOIN Doctor d ON v.DoctorID = d.DoctorID;', 'List of visits with patient names and their corresponding doctor names.', 'Pass'),

('Join Queries', 'Select patients and their medical conditions', 'SELECT p.Name AS PatientName, mh.Conditions FROM MedicalHistory mh JOIN Patient p ON mh.PatientID = p.PatientID;', 'List of patients with their medical conditions.', 'Pass'),

```

``` SQL

-- INSERT Statements

('INSERT Statements', 'Insert a new patient', 'INSERT INTO Patient (PatientID, Name, Address, BirthDate, EligibilityCriteria) VALUES (4, ''Alice Blue'', ''101 Maple St, Anytown, USA'', ''1995-04-04'', ''Met'');', 'A new patient record for Alice Blue is inserted.', 'Pass'),

('INSERT Statements', 'Insert a new doctor', 'INSERT INTO Doctor (DoctorID, Name, Specialization, ContactInfo) VALUES (4, ''Dr. Emily White'', ''Dermatology'', ''emily.white@hospital.com'');', 'A new doctor record for Dr. Emily White is inserted.', 'Pass'),

('INSERT Statements', 'Insert a new visit', 'INSERT INTO Visit (VisitID, Date, Time, PatientID, DoctorID, Observations) VALUES (4, ''2025-02-01'', ''09:00:00'', 4, 4, ''First consultation'');', 'A new visit record is inserted.', 'Pass'),

```

``` SQL

-- UPDATE and DELETE Statements

('UPDATE and DELETE Statements', 'Update patient address', 'UPDATE Patient SET Address = ''202 Oak St, Newtown, USA'' WHERE PatientID = 1;', 'The address of the patient with PatientID = 1 is updated.', 'Pass'),

('UPDATE and DELETE Statements', 'Delete a visit', 'DELETE FROM Visit WHERE VisitID = 1;', 'The visit record with VisitID = 1 is deleted.', 'Pass');

```

## TRIGGER FUNCTION FOR THE AUDITING CHANGES IN THE PATIENT TABLE

``` SQL

-- Trigger for auditing changes in the Patient table

CREATE TRIGGER trg\_Audit\_Patient

ON Patient

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

-- Log insertions

IF EXISTS (SELECT \* FROM inserted)

BEGIN

INSERT INTO AuditLog (TableName, Operation, ChangedBy, OldValue, NewValue)

SELECT 'Patient', 'INSERT', SYSTEM\_USER, NULL,

CONCAT('PatientID=', inserted.PatientID, ', Name=', inserted.Name,

', Address=', inserted.Address, ', BirthDate=', inserted.BirthDate)

FROM inserted;

END

-- Log updates

IF EXISTS (SELECT \* FROM deleted) AND EXISTS (SELECT \* FROM inserted)

BEGIN

INSERT INTO AuditLog (TableName, Operation, ChangedBy, OldValue, NewValue)

SELECT 'Patient', 'UPDATE', SYSTEM\_USER,

CONCAT('PatientID=', deleted.PatientID, ', Name=', deleted.Name,

', Address=', deleted.Address, ', BirthDate=', deleted.BirthDate),

CONCAT('PatientID=', inserted.PatientID, ', Name=', inserted.Name,

', Address=', inserted.Address, ', BirthDate=', inserted.BirthDate)

FROM deleted

JOIN inserted ON deleted.PatientID = inserted.PatientID;

END

-- Log deletions

IF EXISTS (SELECT \* FROM deleted) AND NOT EXISTS (SELECT \* FROM inserted)

BEGIN

INSERT INTO AuditLog (TableName, Operation, ChangedBy, OldValue, NewValue)

SELECT 'Patient', 'DELETE', SYSTEM\_USER,

CONCAT('PatientID=', deleted.PatientID, ', Name=', deleted.Name,

', Address=', deleted.Address, ', BirthDate=', deleted.BirthDate), NULL

FROM deleted;

END

END;

```

``` SQL

-- Select patients born after 1985

SELECT \* FROM Patient WHERE BirthDate > '1985-01-01';

```

``` SQL

-- Select doctors specialized in Cardiology

SELECT \* FROM Doctor WHERE Specialization = 'Cardiology';

```

``` SQL

-- Select visits for a specific patient

SELECT \* FROM Visit WHERE PatientID = 1;

```

``` SQL

-- Count the number of patients

SELECT COUNT(\*) AS PatientCount FROM Patient;

```

``` SQL

-- Find the average age of patients

SELECT AVG(DATEDIFF(YEAR, BirthDate, GETDATE())) AS AverageAge FROM Patient;

```

``` SQL

-- Count the number of visits per doctor

SELECT DoctorID, COUNT(\*) AS VisitCount FROM Visit GROUP BY DoctorID;

```

``` SQL

-- Select patient names and their corresponding doctor names for each visit

SELECT p.Name AS PatientName, d.Name AS DoctorName, v.Date, v.Time

FROM Visit v

JOIN Patient p ON v.PatientID = p.PatientID

JOIN Doctor d ON v.DoctorID = d.DoctorID;

```

``` SQL

-- Select patients and their medical conditions

SELECT p.Name AS PatientName, mh.Conditions

FROM MedicalHistory mh

JOIN Patient p ON mh.PatientID = p.PatientID;

```

``` SQL

-- Insert a new patient

INSERT INTO Patient (PatientID, Name, Address, BirthDate, EligibilityCriteria)

VALUES (4, 'Alice Blue', '101 Maple St, Anytown, USA', '1995-04-04', 'Met');

```

``` SQL

-- Insert a new doctor

INSERT INTO Doctor (DoctorID, Name, Specialization, ContactInfo)

VALUES (4, 'Dr. Emily White', 'Dermatology', 'emily.white@hospital.com');

```

``` SQL

-- Insert a new visit

INSERT INTO Visit (VisitID, Date, Time, PatientID, DoctorID, observations)

VALUES (4, '2025-02-01', '09:00:00', 4, 4, 'First consultation');

```

``` SQL

-- Update patient address

UPDATE Patient SET Address = '202 Oak St, Newtown, USA' WHERE PatientID = 1;

```

``` SQL

-- Delete a visit

DELETE FROM Visit WHERE VisitID = 1;

```

```SQL

USE WestLakeClinicalTrial;

```

## CREATING LOGIN INFORMATION BASED ON THE ROLE BASED ACCESS REQUIREMENTS

``` SQL

-- Create logins

CREATE LOGIN doctorLogin WITH PASSWORD = 'YourPassword1!';

CREATE LOGIN nurseLogin WITH PASSWORD = 'YourPassword2!';

CREATE LOGIN adminLogin WITH PASSWORD = 'YourPassword3!';

-- Create users in the database

CREATE USER doctorUser FOR LOGIN doctorLogin;

CREATE USER nurseUser FOR LOGIN nurseLogin;

CREATE USER adminUser FOR LOGIN adminLogin;

-- Create roles

CREATE ROLE Doctor;

CREATE ROLE Nurse;

CREATE ROLE Admin;

-- Grant explicit permissions to roles

GRANT SELECT, INSERT, UPDATE ON Patient TO Doctor;

GRANT SELECT ON Patient TO Nurse;

GRANT SELECT, INSERT, UPDATE, DELETE ON Patient TO Admin;

GRANT SELECT, INSERT, UPDATE ON Visit TO Doctor;

GRANT SELECT ON Visit TO Nurse;

GRANT SELECT, INSERT, UPDATE, DELETE ON Visit TO Admin;

GRANT SELECT, INSERT, UPDATE ON Doctor TO Admin;

-- Assign roles to users

EXEC sp\_addrolemember 'Doctor', 'doctorUser';

EXEC sp\_addrolemember 'Nurse', 'nurseUser';

EXEC sp\_addrolemember 'Admin', 'adminUser';

-- Add the BirthDate constraint to the Patient table

ALTER TABLE Patient ADD CONSTRAINT CHK\_BirthDate CHECK (BirthDate > '1900-01-01');

-- Add the Specialization constraint to the Doctor table

ALTER TABLE Doctor ADD CONSTRAINT CHK\_Specialization CHECK (Specialization IS NOT NULL);

-- Add the EligibilityCriteria column and constraint to the Patient table

ALTER TABLE Patient ADD EligibilityCriteria VARCHAR(255);

ALTER TABLE Patient ADD CONSTRAINT CHK\_Eligibility CHECK (EligibilityCriteria = 'Met');

ALTER TABLE Patient ADD CONSTRAINT CHK\_Eligibility CHECK (EligibilityCriteria = 'Met');

```