OAKRIDGE
GENERAL
HOSPITAL

DATABASE DESIGN & ARCHITECTURE

Presented by:
AURORA GONZALEZ

2025

Requirements & Scope

The OakRidge General Hospital database was designed to fulfill critical clinical and administrative needs for both inpatient and outpatient care workflows. The system architecture supports:

Inpatient and Outpatient Support: The data model accommodates patient admissions, appointments, medical histories, prescriptions, lab results, and bed assignments. Inpatients are linked to bed tracking and admissions data, while outpatients are associated with appointment scheduling, diagnostics, and physician workflows.

Multi-role Access Model: A core requirement was to grant secure, role-specific access to a wide range of hospital personnel including physicians, nurses, administrative coordinators, and external auditors. Access permissions are configured based on least privilege principles, with row-level security (RLS) ensuring users only see data relevant to their responsibilities. Examples include doctors viewing only their assigned patients, nurses updating vital signs, and billing clerks accessing insurance data.

I**nteroperability & System Integration:** The schema was structured to be interoperable with electronic health record (EHR) systems, medical billing platforms, laboratory devices, and external analytics tools such as Power BI. Tables and fields are normalized and include integration keys (e.g., patient external IDs, insurance references) to facilitate data flow through ETL pipelines, APIs, and external connectors.

Regulatory & Compliance Alignment: All aspects of the design comply with major data protection and healthcare regulations such as HIPAA (USA), PHIPA (Canada), and GDPR (Europe). This includes the implementation of:

- Field-level encryption (e.g., Always Encrypted columns for diagnoses and medications).
- Dynamic Data Masking for personally identifiable information (PII) such as email addresses and phone numbers.
- SQL Audit for access tracking and traceability.
- Secure access configurations to protect data at rest and in transit, consistent with Azure security standards.

This foundational layer ensures that the hospital's information system is compliant, scalable, and equipped to evolve alongside emerging clinical technologies and governance requirements.

Data Modeling (ERD)

The data model at the core of the OakRidge General Hospital system was designed to accurately reflect real-world clinical relationships while maintaining scalability, performance, and referential integrity. The foundation of the model is the **centralized Patient table**, which anchors all patient-related data through consistent and enforced foreign key relationships.

Relational Core Design:

- The **Patient** entity serves as the hub, with direct relationships to **Diagnoses**, **Prescriptions, LabResults, and VitalSigns**. Each record in these tables is tied to a specific patient via a foreign key, enabling precise tracking of medical history over time.
- The **Appointments** table links patients to healthcare providers and departments, including timestamps, status codes (Scheduled, Completed, Cancelled), and billing information.
- **BedAssignments** and **Admissions** are used to monitor room occupancy, patient flow, and availability of hospital resources in real-time.

Modular & Scalable Structure:

- Each entity was designed as a **modular component** with minimal coupling to other entities, supporting easy updates or additions. For instance, the **MedicalEquipment** table could be extended to include IoT integration in future phases.
- Tables like **ImagingRecords, InsuranceClaims,** and **ClinicalNotes** are optional modules that can be plugged into the system without disrupting core functionality.

Referential Integrity & Governance:

- **Foreign key constraints** are rigorously applied across all primary relationships to ensure referential integrity.
- Cascading rules (ON DELETE/ON UPDATE) are selectively applied to prevent data orphaning while maintaining auditability.
- **Composite keys** are used where appropriate to handle many-to-many relationships (e.g., Doctor-Patient assignments, Medication-Treatment mappings).

Future-readiness:

- The **ERD** was designed with extensibility in mind, allowing for schema evolution and integration with external clinical, administrative, or regulatory data domains.
- Metadata tables (e.g., RecordStatus, DepartmentTypes) support localization, versioning, and cross-system mapping.

This modeling approach ensures that the database structure remains robust, navigable, and adaptable to the hospital's evolving operational and technological needs.

Database Normalization

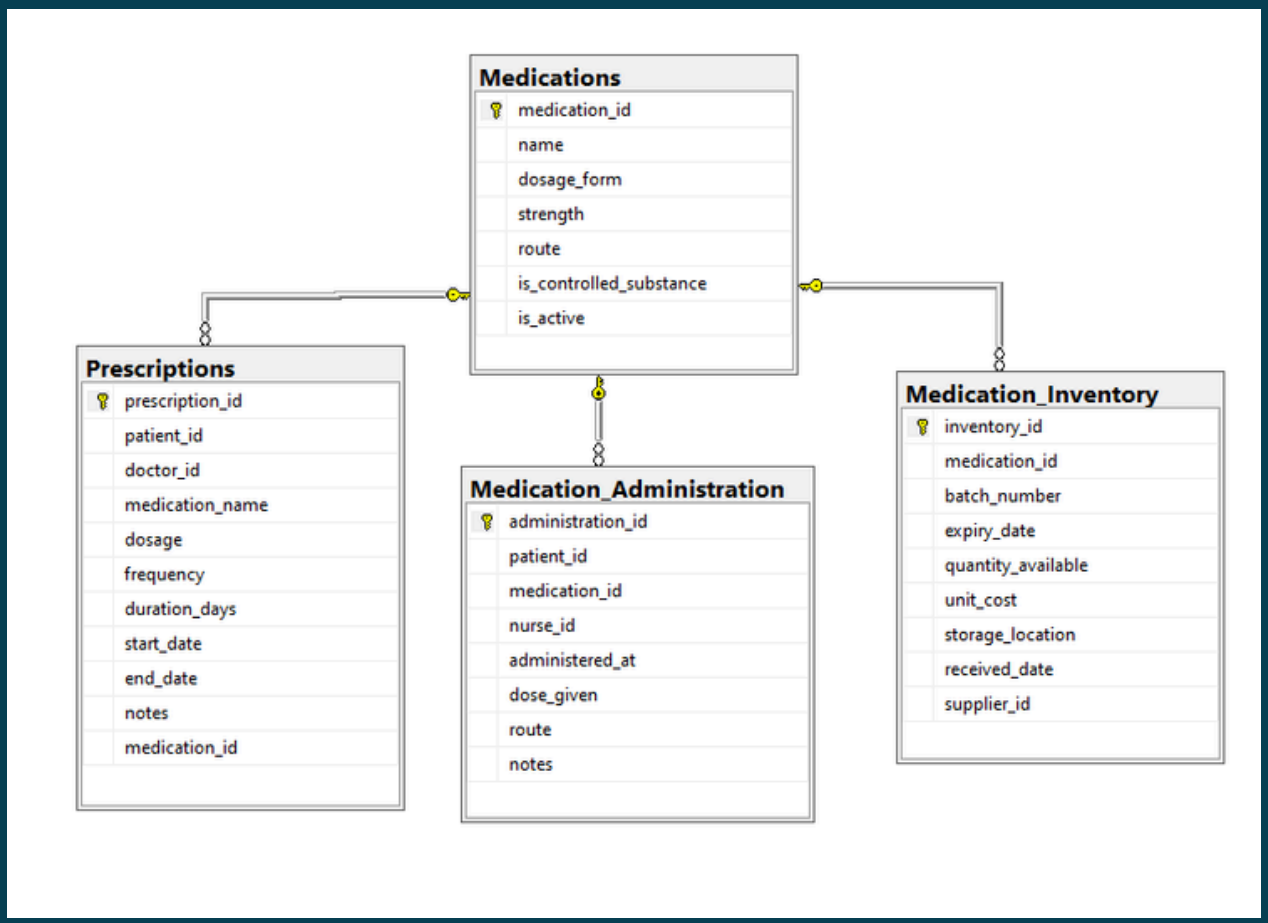
To ensure data consistency, efficiency, and long-term maintainability, the OakRidge General Hospital database has been fully normalized to the **Third Normal Form (3NF)**. This decision eliminates data redundancy, improves relational clarity, and allows for seamless schema evolution as hospital needs grow.

Normalization Objectives:

- Eliminate duplication of data across entities such as **Medications, Doctors, and Departments.**
- Break down complex or repeating fields into individual entities (e.g., multiple contact numbers stored in a separate ContactDetails table).
- Guarantee that every non-key attribute is **fully functionally** dependent on the primary key of its table.

Examples of Normalized Design:

- The **Doctor** entity maintains unique records for each practitioner, while their schedules and department affiliations are stored in separate associative tables.
- **Departments** are managed in a distinct table, linked via foreign keys to Appointments, Admissions, and Medical Staff, ensuring consistency across usage points.
- **Medications** are cataloged with unique codes and standard dosage units, which are referenced in Prescriptions but never duplicated.



Benefits Realized:

- Improved query performance for reporting tools like Power BI by minimizing unnecessary joins on duplicate data.
- Reduced storage overhead by avoiding repetition of large text fields or foreign key chains.
- Easier enforcement of business rules and data validation through a logically consistent schema.

Maintenance & Extensibility:

- New features (e.g., multilingual support, audit logs) can be integrated by extending the normalized structure without overhauling core logic.
- Normalization simplifies the process of identifying anomalies, facilitating easier debugging and long-term refactoring.

Overall, normalization plays a critical role in the integrity and operational efficiency of the hospital's information system, enabling developers and analysts to work with clean, accurate, and logically organized data.

Sensitive Data Types

In a healthcare environment, the protection of sensitive data is not only a best practice—it is a legal obligation. The OakRidge General Hospital system enforces strong data protection protocols by categorizing and securing personally identifiable information (PII), protected health information (PHI), and system credentials using industry-grade techniques.

Personally Identifiable Information (PII):

- Fields such as **Social Security Numbers (SSNs)**, email addresses, and **phone numbers** are protected using **Dynamic Data Masking (DDM)**. This ensures that non-privileged users only see partially obscured values, maintaining usability while reducing exposure risk.
- Example: A masked email might appear as j***@example.com to a receptionist, while fully visible to an administrator.

Protected Health Information (PHI):

- Clinical fields such as **diagnoses, treatment plans**, and **medical record content** are encrypted using **Always Encrypted** technology with column-level encryption. Keys are stored using a **Column Master Key (CMK)** and **Column Encryption Key (CEK)** strategy.
- Encryption is applied at rest and in transit, making data unreadable to unauthorized users—even DBAs without access to encryption keys.

System Credentials and Authentication Data:

- All user passwords and API tokens are stored as **hashed varbinary values**, generated using SHA-2 family algorithms with salt.
- No plain-text credentials are stored, and failed authentication attempts are logged for auditing and lockout enforcement.

Compliance Alignment:

- These protections align with **HIPAA, PHIPA, and GDPR** security requirements.
- Data access is governed by role-level controls, and additional layers such as **Row-Level Security (RLS**) ensure that users see only the data they are authorized to access.

Integration with Other Systems

One of the defining strengths of the OakRidge General Hospital database design is its readiness for seamless integration with external systems. Whether interfacing with Electronic Health Records (EHR) platforms, insurance networks, or analytics tools like Power BI, the schema was crafted with interoperability, automation, and exportability at its core.

API-Ready Structure:

- Tables are structured with consistent naming conventions, primary keys, and lookup references to support RESTful API development.
- Patient, appointment, prescription, and billing modules were modeled with standardized timestamps and status fields to facilitate reliable data synchronization.
- Integration endpoints can expose key views or stored procedures without compromising sensitive data, thanks to masking and row-level filtering.

Cross-Platform Export Support:

- The system includes transformation-ready views and export tables designed for structured output into **PostgreSQL, MongoDB, CSV files, and JSON payloads.**
- Mapping tables and staging areas support format conversion for legacy systems or third-party applications.
- Export formats are compatible with downstream consumers such as insurance processors, government reporting systems, and laboratory analysis sof

Automation & Job Scheduling:

- The database is equipped with a job-ready schema for use with **SQL Server Agent, Azure Data Factory, or custom PowerShell scripts.**
- Data pipelines can be orchestrated to perform daily extracts, conditional loads, or scheduled API pushes.
- Logging tables and run history tracking ensure auditability of all automation jobs.

Benefits:

- Reduces manual intervention through repeatable integrations.
- Supports hybrid deployment models by enabling on-prem and cloud data movement.
- Increases operational agility and facilitates data-driven decision-making across the organization.

This integration-first architecture ensures that the hospital's digital ecosystem remains open, scalable, and fully aligned with modern healthcare interoperability standards.

Performance & Optimization

To ensure fast query execution and high availability for critical hospital operations, the OakRidge General Hospital database was designed and continuously optimized with industry-standard performance strategies. Even in early stages, this optimization mindset was embedded into the schema and data access layers.

Indexing Strategy:

- A baseline of clustered and non-clustered indexes was created on high-traffic tables such as Appointments, MedicalRecords, Patients, and LabResults.
- Developers used the system view **sys.dm_db_missing_index_details** and related DMVs to identify missing indexes based on actual query patterns observed in development and test environments.
- Composite indexes were designed where queries regularly filtered on multiple columns (e.g., AppointmentDate, DoctorID, Status).

Query Store & Cardinality Insight:

- **Query Store** was enabled from the start to track query plans, regressions, and runtime statistics.
- Execution plans were reviewed to identify cardinality estimation issues, including unnecessary full table scans or nested loops.
- Problematic queries were restructured using best practices (e.g., avoiding functions in WHERE clauses, simplifying joins).

Statistics and Index Maintenance

- A lightweight SQL Agent job was implemented to run **index rebuilds or reorganizations** based on fragmentation thresholds.
- **Statistics were updated** regularly using the UPDATE STATISTICS command with FULLSCAN for critical tables during low-traffic windows.
- System tables were monitored for skewed distributions that could affect query plans.

Additional Optimization Techniques:

- Parameter sniffing issues were mitigated with **OPTIMIZE FOR** hints and stored procedure reviews.
- Views and materialized summaries were proposed for reporting dashboards (e.g., daily appointment counts, lab result turnaround times).
- Usage of WITH (NOLOCK) was evaluated cautiously to reduce blocking where appropriate.

This foundation ensures that as data volume grows, the database remains performant and ready to scale with minimal technical debt. Performance tuning is an ongoing process, but the early implementation of these mechanisms places the system in a strong position for long-term stability.

Legal & Compliance Considerations

In healthcare, regulatory compliance is not optional—it is foundational. The OakRidge General Hospital database was built with a deep alignment to legal frameworks including HIPAA (USA), PHIPA (Canada), and GDPR (EU). The design supports patient privacy, auditability, and long-term legal defensibility across all operational layers.

SQL Auditing & Change Tracking:

- Tables such as Change_Log and Clinical_Audits were implemented to capture data modifications, user actions, and sensitive access events.
- Custom triggers and SQL Server Audit objects are used to track **INSERT, UPDATE, DELETE** operations on high-risk tables like MedicalRecords, Prescriptions, and Patients.
- Audit entries include metadata such as UserID, timestamp, action type, and affected fields.

SQData Retention Policies:

- Retention rules were applied to ensure data is preserved for the minimum regulatory periods:
  - Medical records: 10 years (based on jurisdiction).
  - Access logs: 5–7 years depending on risk level.
- Historical records are archived using separate partitioned tables or archived databases.
- Old entries can be flagged as immutable or write-protected post-audit.

Encryption & Secure Storage:

- Data is protected at rest using Transparent Data Encryption (TDE) or Always Encrypted, depending on sensitivity.
- In-transit protection is ensured via encrypted database connections using TLS.
- Backup files are encrypted and stored in Azure with geo-redundancy for disaster recovery compliance.

Compliance Framework Alignment:

- Field-level security aligns with GDPR's "right to access" and "right to be forgotten" through controlled exposure and soft-deletion.
- All audit processes are documented for internal and third-party inspections.
- The system was designed to pass compliance audits with full traceability of access, changes, and incident response.

Together, these measures form a multi-layered compliance architecture that protects both patient data and institutional liability while maintaining operational efficiency in a regulated industry.

Access Control

To ensure data confidentiality and operational integrity, the OakRidge General Hospital database implements a robust access control system rooted in the principles of least privilege and role separation. This enables differentiated access paths for medical staff, administrators, and external auditors while enforcing compliance with legal and ethical data practices.

Role-Based Access Model (RBAC):

- Defined roles include **Administrator, Doctor, Nurse, Receptionist, and Billing Agent.**
- Each role has access only to the operations and data necessary for their duties:
  - **Doctors** can view and update their assigned patients' records and prescriptions.
  - **Nurses** can view vital signs and administer treatments but not modify diagnoses.
  - **Billing Agents** have access to insurance and invoice tables, but not clinical data.
- Role permissions are managed using SQL Server database roles and secured views.

Row-Level Security (RLS):

- RLS is applied to critical patient-related tables, such as MedicalRecords, Appointments, and LabResults.
- Filter predicates ensure that users only see rows where they are authorized, such as:
  - A doctor viewing only patients assigned to their ID.
  - A nurse accessing vitals for patients currently admitted in their unit.

Auditing Access Violations:

- SQL Server Audit policies are configured to detect unauthorized data access attempts.
- Attempts to access restricted patient records or modify protected fields are logged to an **AuditEvents** table.
- Each log captures UserID, IP address (if available), timestamp, action attempted, and success/failure status.

Dynamic Access Enforcement:

- In scenarios where users have dynamic assignments (e.g., covering shifts or emergency access), temporary permissions can be granted and revoked with expiration logic.
- Stored procedures are used to escalate access in emergencies, and all such events are audited.

This multi-layered access control strategy ensures not only operational efficiency and role clarity, but also provides the foundational infrastructure for data protection, accountability, and audit-readiness across all levels of the hospital's digital environment.