كلية العلوم

قســــــــم الانظمة الطبية الذكية

Lec 3 & 4

Implementation, Components, Functions, and EHR Data Types

Subject: Electronic Health Records

Level: fourth

Lecturer: Asst. Lecturer Qusai AL-Durrah

INTRODUCTION

EHR Systems: Beyond Software

Critical Understanding

EHR systems represent more than just technology—they are integrated ecosystems combining clinical workflow, data architecture, and governance frameworks.

Clinical Workflow

How providers interact with patient data during care delivery

Data Architecture

Structured design determining how information is stored and retrieved

Governance

Policies ensuring security, privacy, and quality standards

Real EHR success depends on implementation planning and correct data structure. If we misunderstand the data types, we will build the wrong database in SSMS.

Lecture Objectives

01

Implementation Lifecycle

Explain the EHR implementation lifecycle from needs assessment through optimization

02

System Components

Identify the four pillars of EHR system components: people, process, technology, and data

03

Core Functions

Describe essential EHR functions including registration, documentation, orders, and reporting

04

Data Classification

Classify EHR data types: structured vs semi-structured vs unstructured

05

Database Translation

Translate EHR concepts into database entities for SSMS implementation

Why EHR Implementation is Hard

EHR implementation represents one of the most complex socio-technical projects in healthcare, requiring careful orchestration of multiple interdependent factors. Understanding these challenges is essential for successful deployment.

Socio-Technical Complexity

Success requires alignment of people, processes, technology, and data. Each dimension influences the others, creating intricate dependencies that must be managed simultaneously.

Workflow Disruption Risk

New systems fundamentally alter established clinical workflows. Providers must adapt longstanding practices, which can temporarily reduce efficiency and increase cognitive load during transition periods.

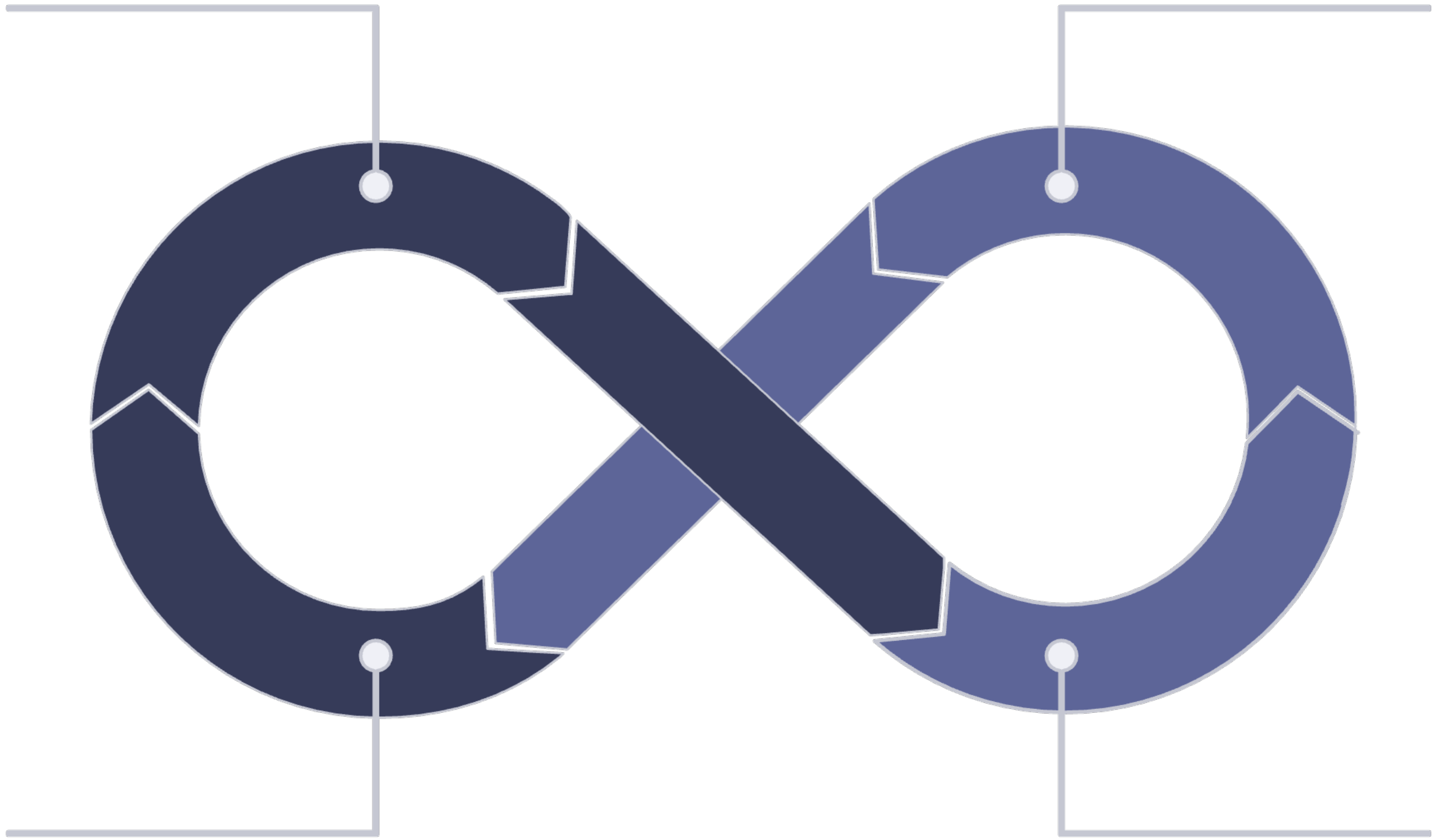
Data Quality & Patient Safety

Poor data design directly impacts patient outcomes. Incorrect database structures lead to wrong clinical decisions, medication errors, and patient harm. Bad design equals dangerous care.

IMPLEMENTATION

EHR Implementation Lifecycle

The implementation lifecycle represents a continuous improvement loop, not a linear one-time project. Organizations must plan for ongoing optimization and evaluation to maintain system effectiveness and adapt to evolving clinical needs.



Needs Assessment & Scope

Requirements Definition

Data Migration

Selection & Build

Each phase builds upon the previous one, and skipping or rushing any phase significantly increases implementation failure risk. The optimization phase feeds back into needs assessment, creating a cycle of continuous improvement.

Phase 1: Needs Assessment & Scope

The needs assessment phase establishes the foundation for all subsequent implementation decisions. Scope determination directly influences which database tables will be built first and how resources are allocated throughout the project lifecycle.

Care Setting Definition

Identify whether the EHR will serve outpatient clinics, inpatient hospitals, specialty practices, or multi-site networks. Each setting has unique workflow requirements.

Key Workflows

Determine priority workflows to support: appointment scheduling, clinical documentation, medication management, lab ordering, billing, or reporting.

Resource Constraints

Assess realistic constraints including budget limitations, IT staff expertise, network infrastructure capacity, and organizational change readiness.

Phase 2: Requirements Definition

Requirements must be clearly distinguished from user interface design. Stating "we need a screen to enter patient data" is not a requirement—it's a design assumption. True requirements describe *what* the system must do, not *how* it looks.

Functional Requirements

What the system **does** to support clinical operations:

- Register new patients and update demographics
- Document clinical encounters with structured and narrative data
- Order laboratory tests and retrieve results
- Prescribe medications with dosage and frequency
- Generate clinical reports and summaries
- Support communication between providers

Non-Functional Requirements

How the system **performs** and meets quality attributes:

- Security and privacy compliance (HIPAA, authentication)
- Auditability and comprehensive logging
- Performance under concurrent user load
- System reliability and uptime guarantees
- Interoperability with external systems
- Disaster recovery and backup procedures

Phase 3: Selection & Build

Buy from Vendor

Pros: Faster deployment, established support, regular updates, proven reliability

Cons: Less customization, vendor lock-in, licensing costs, dependency on vendor roadmap

Build Custom

Pros: Complete control, tailored workflows, no licensing fees, intellectual property ownership

Cons: High development cost, longer timeline, ongoing maintenance burden, requires specialized staff

Deployment Models

On-Premises

- Full data control and physical security
- High upfront infrastructure costs
- Requires in-house IT expertise

Cloud-Based

- Lower initial investment, subscription model
- Automatic updates and maintenance
- Scalability and disaster recovery built-in

Integration capability matters regardless of deployment choice. Ensure the solution supports healthcare interoperability standards, which we'll cover in future lectures.

Phase 4: Data Migration

Data migration represents one of the highest-risk phases of EHR implementation. Poor migration can result in duplicate patient records, lost clinical history, and dangerous care gaps. The process requires meticulous planning and validation.

Map Old to New

Analyze legacy data structure and map fields to new EHR schema. Address format differences, terminology changes, and structural mismatches.

Clean & Deduplicate

Identify and merge duplicate patient records using probabilistic matching algorithms. Resolve identity conflicts and standardize data formats.

Validate After Import

Run comprehensive validation queries to ensure data integrity. Verify patient counts, clinical completeness, and referential integrity across tables.

Critical Safety Issue

Duplicate patient records create unsafe care conditions. When a patient exists multiple times in the system, providers cannot see the complete clinical picture.

This leads to:

- Missed allergy alerts
- Duplicate medication orders
- Repeated diagnostic tests
- Fragmented medical history

Result: Patient harm and compromised care quality.

COMPONENTS & FUNCTIONS

The Four Pillars of EHR Systems

Every EHR system is built on four interdependent pillars. Understanding these components helps us design better databases, because each pillar generates or consumes data in specific ways that must be reflected in our schema design.

People (Roles)

Physicians, nurses, administrators, patients—each with specific permissions and responsibilities. Roles later map to database permissions and audit trails.

Process (Workflow)

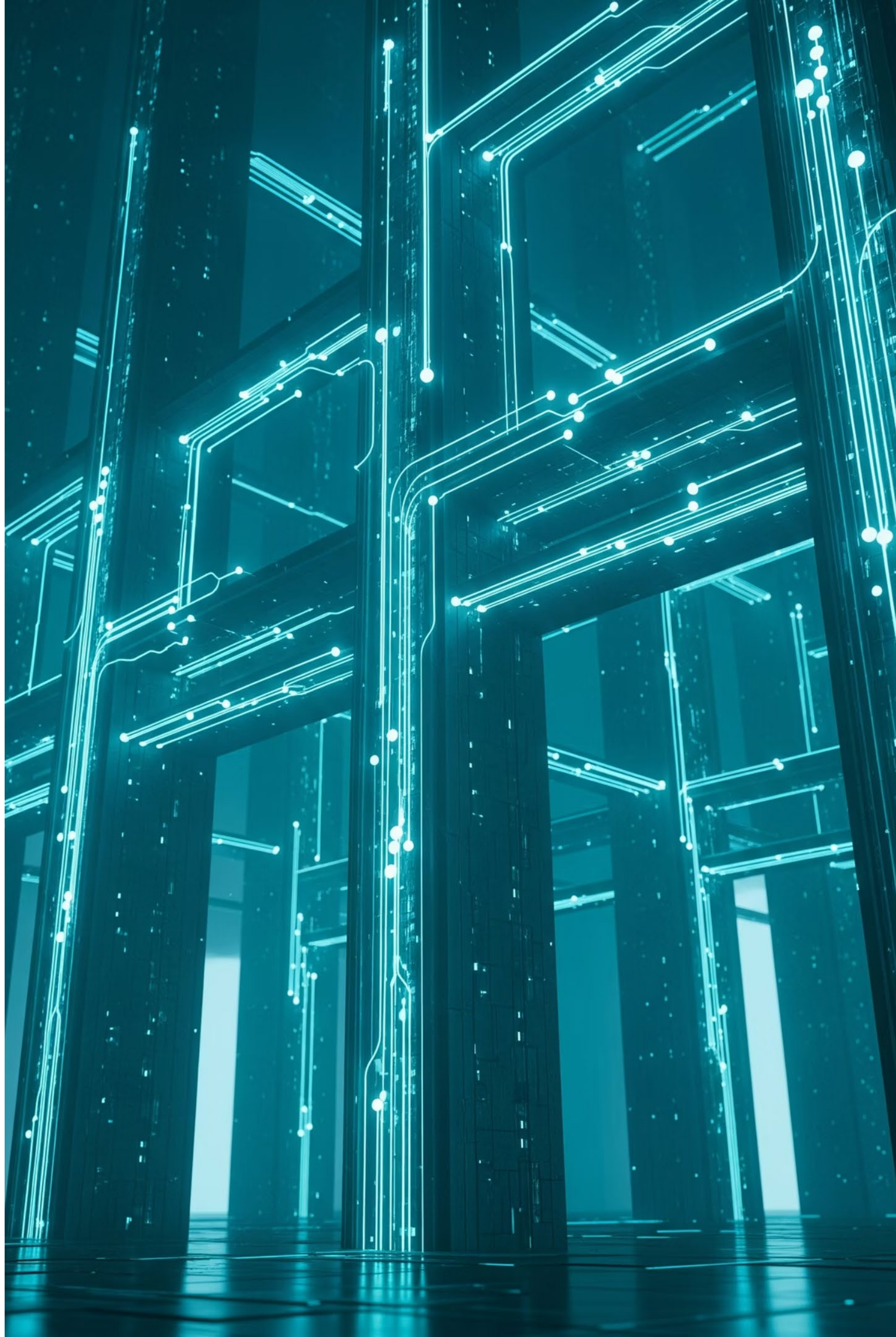
Standardized clinical workflows define how care is delivered. Processes determine what data is captured, when, and by whom—directly shaping database transactions.

Technology (Infrastructure)

Applications, servers, networks, and security systems. Technology choices affect database architecture, performance requirements, and integration patterns.

Data (All Types)

Structured data in tables, semi-structured documents, unstructured text and images, plus metadata describing it all. Data is what we'll focus on for SSMS implementation.



Components vs Functions

It's essential to distinguish between what an EHR system **is made of** (components) and what it **does** (functions). This distinction helps us understand how system architecture supports clinical operations.

Components

What the system IS

People: Roles and users

Process: Clinical workflows

Technology: Hardware and software

Data: Information assets

Functions

What the system DOES

- Patient registration and identity management
- Clinical documentation of encounters
- Orders and results management
- Medication prescribing and tracking
- Provider-to-provider communication
- Reporting and analytics generation

Mapping Example: The "Medication Management" function requires the People component (prescribing provider), Process component (medication ordering workflow), Technology component (drug interaction checking algorithms), and Data component (medication tables with drug codes, dosages, and patient allergies).

Core Functions of EHR Systems

These six core functions represent the essential capabilities that every EHR must provide to support comprehensive patient care. Each function generates specific data types that must be properly modeled in our database design.

Patient Registration & Identity Management

Create and maintain unique patient identifiers, demographic information, contact details, and insurance data. Ensures one patient = one record across the system.

Clinical Documentation

Record encounter notes, chief complaints, history of present illness, physical exam findings, assessment, and treatment plans using structured and free-text formats.

Orders & Results Management

Place orders for labs, imaging, and procedures; receive and display results; track order status; alert providers to critical values requiring immediate attention.

Medication & Allergy Management

Prescribe medications electronically, check drug interactions and allergies, maintain medication lists, track adherence, and manage refills and renewals.

Communication Tools

Enable secure provider-to-provider messaging, care team coordination, patient portal access for viewing records and lab results, and appointment scheduling.

Reporting & Analytics

Generate quality measure reports, population health analytics, financial summaries, regulatory compliance reports, and clinical decision support insights.

EHR DATA TYPES

What Data Lives Inside an EHR?

Understanding EHR data categories is fundamental to database design. Each data type represents a potential entity or table in our SSMS schema. These are not text fields—they are structured entities with relationships, constraints, and business rules.

Administrative & Patient Services (APS)

Patient demographics, insurance, contact information, emergency contacts, preferred language, marital status

Medical History

Past diagnoses, surgical history, hospitalizations, family history, social history (smoking, alcohol use)

Current Diagnoses

Active problem lists with ICD-10 codes, onset dates, severity, status (active/resolved)

Medications

Current prescriptions, dosages, routes, frequencies, start/stop dates, prescribing providers

Vaccination Schedules

Immunization history, vaccine types, administration dates, lot numbers, due dates for next doses

Allergies & Reactions

Allergen substance, reaction type, severity level, onset date, verification status

Imaging & Radiographs

X-ray, MRI, CT images with DICOM metadata, radiology reports, ordering providers, findings

Laboratory Tests & Results

Test orders, specimen types, result values, units, reference ranges, abnormal flags, performing lab

Database Design Principle: Each category above becomes entities (tables) with primary keys, foreign keys, and relationships—NOT comma-separated text values stored in a single column.

Structured Data: Easy to Query

Structured data represents information organized into predefined fields with specific data types, making it easily searchable, sortable, and analyzable. This is the foundation of relational database design and enables powerful clinical queries and decision support.

Structured Data Examples

- **Patient Demographics:** ID, date of birth, sex, address, phone number
- **Encounters:** Visit date, visit type, clinic location, attending provider
- **Diagnoses:** ICD-10 code, description text, diagnosis date, status
- **Medications:** Drug name, dose, route, frequency, start date, end date
- **Allergies:** Substance name, reaction type, severity level, verification status
- **Vital Signs:** Blood pressure, heart rate, weight, temperature, measurement timestamp
- **Lab Results:** Test name, numeric result value, units, reference range, abnormal flag

Database Implementation

Structured data translates to **tables with proper primary keys (PK) and foreign keys (FK)**—never as comma-separated text strings.

Correct approach:

```
Patient (PatientID, FirstName, LastName, DOB)
Medication (MedicationID, PatientID, DrugCode, Dose)
Allergy (AllergyID, PatientID, Substance, Severity)
```

Wrong approach:

```
Patient (PatientID, Allergies VARCHAR(500))
```

The wrong approach prevents querying "all patients allergic to penicillin" and violates normalization principles.

Semi-Structured Data: Mixed Format

Semi-structured data contains organizational markup or tags but lacks the rigid schema of fully structured data. It often appears in healthcare interoperability messages, device data streams, and flexible clinical forms.

Variable Forms

Clinical assessment forms where not all fields are required or applicable to every patient. Fields present depend on patient condition or workflow context.

HL7 & FHIR Messages

Healthcare interoperability standards transmit data using JSON or XML formats. These contain structured elements but flexible hierarchies and optional fields.

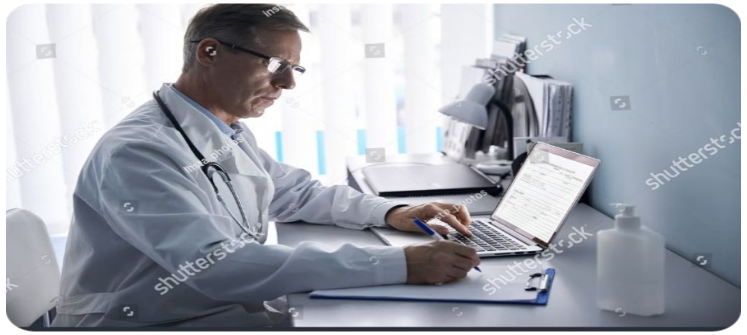
Device Data Streams

Continuous monitoring devices generate timestamped measurements in various formats: vital signs monitors, infusion pumps, telemetry systems.

Database Strategy: Store semi-structured data using tables with JSON/XML columns where schema flexibility is needed, but **always maintain PatientID, EncounterID, and timestamps** as structured columns for querying and joining. Even flexible JSON documents need relationships to core entities.

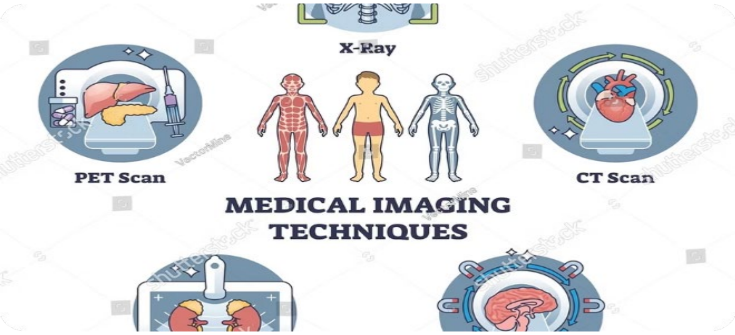
Unstructured Data: Human-Readable Content

Unstructured data lacks predefined organization and cannot be easily stored in traditional database rows and columns. It requires different storage strategies and metadata management to remain accessible and useful within the EHR ecosystem.



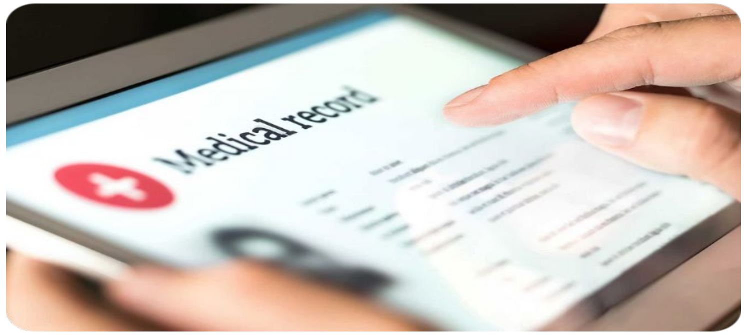
Clinical Notes (Free Text)

Narrative documentation including history of present illness, physical examination findings, clinical impressions, and treatment plans written in natural language by providers.



Radiology Images & Reports

Large DICOM image files from CT, MRI, X-ray equipment, accompanied by radiologist interpretation reports describing findings and recommendations.



Scanned Documents & PDFs

Historical paper records converted to digital format, external consultation reports, patient-provided documents, consent forms, and insurance correspondence.

Database Implementation

Store **metadata + file references**, not entire files as BLOB columns (though architecture-dependent).

Metadata examples:

- PatientID (FK to Patient table)
- EncounterID (FK if encounter-specific)
- FileType (e.g., "PDF", "DICOM", "TXT")
- FilePath or URI to file storage location
- CreatedDate and Author/CreatedBy
- DocumentCategory (e.g., "Lab Report", "Consent Form")

Why This Approach?

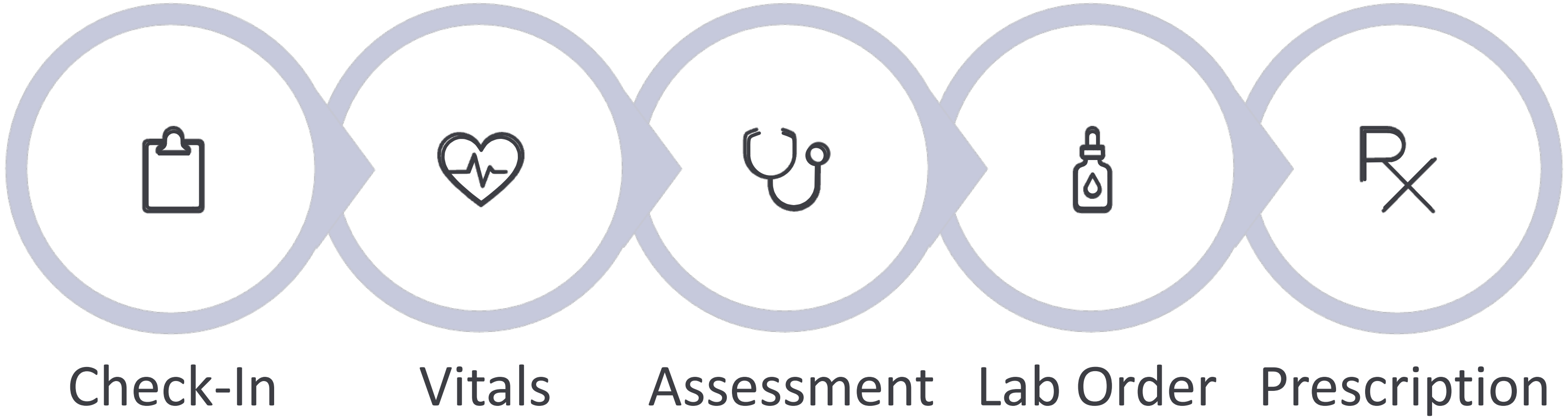
Storing large binary files directly in database tables creates performance bottlenecks, increases backup complexity, and limits scalability.

Instead, store files in dedicated storage systems (file servers, object storage like S3) and maintain **pointers** to them in database tables. This preserves relationships while optimizing performance.

PRACTICAL EXAMPLE

Mini Case: Diabetes Clinic Workflow

Let's trace a patient journey through a diabetes clinic to see how workflow steps generate specific data entities. This example bridges theoretical concepts to practical database design you'll implement in SSMS.



Workflow Steps

1. **Check-in:** Patient arrives, registration confirms demographics and insurance
2. **Vitals:** Nurse measures blood pressure, weight, glucose level
3. **Assessment:** Provider reviews history, examines patient, documents findings
4. **Labs:** Order HbA1c test, lipid panel for monitoring
5. **Medications:** Prescribe metformin, adjust insulin dosage
6. **Follow-up:** Schedule 3-month return visit, patient education

Minimum Database Entities

Each workflow step produces or consumes data:

PatientDemographics, contact info

EncounterVisit date, type, provider

Provider Physician information

DiagnosisType 2 diabetes with ICD code

MedicationOrderPrescriptions with dosages

LabOrderTests requested

LabResultHbA1c value, lipid results

VitalBP, weight, glucose readings

Bridge to Practical Lab

In our next lab session, we transition from theory to hands-on database implementation. You'll install SQL Server Management Studio and begin building the foundational schema for a functional EHR database.

Next Lab: SSMS Setup

01

Install SSMS

Download and install SQL Server Management Studio on your machine

02

Connect to Server

Establish connection to the course database server using provided credentials

03

Create Database

Initialize new database named `EHR_DB` for your schema development

04

Start Schema Design

Create initial tables: Patient, Encounter, Diagnosis, and Medication with appropriate data types and constraints

Homework Assignment

Objective: Apply data classification concepts to a real clinical scenario

Requirements:

1. Select **one** clinical scenario (e.g., emergency department visit, surgical procedure, chronic disease management)
2. Identify and list **12 specific data elements** captured during that scenario
3. Classify each element as: **structured**, **semi-structured**, or **unstructured**
4. Propose **4–6 table names** for storing the structured elements with brief justification

Submission: Word or PDF document, maximum 1 page

Due: Before next lab session

Key Takeaways

Implementation is Cyclical

EHR implementation follows a structured lifecycle from needs assessment through optimization, with continuous improvement loops rather than one-time deployment.

Four Component Pillars

Every EHR system comprises people, process, technology, and data. Understanding these components informs better database design decisions.

Core Functions Drive Data

Six essential functions—registration, documentation, orders, medications, communication, reporting—determine what data entities we need to model.

Three Data Types

Structured data goes in tables with keys; semi-structured needs flexible formats with core relationships; unstructured requires metadata + file references.

Design Impacts Safety

Poor data modeling leads to wrong clinical decisions and patient harm. Duplicate records, lost relationships, and bad structures have real-world safety consequences.

Theory Meets Practice

Every workflow step generates data entities. Our SSMS implementation will translate these concepts into working tables with proper relationships and constraints.

Thank you

Google Classroom