Unit 2: Database System Development Life Cycle (DSDLC)

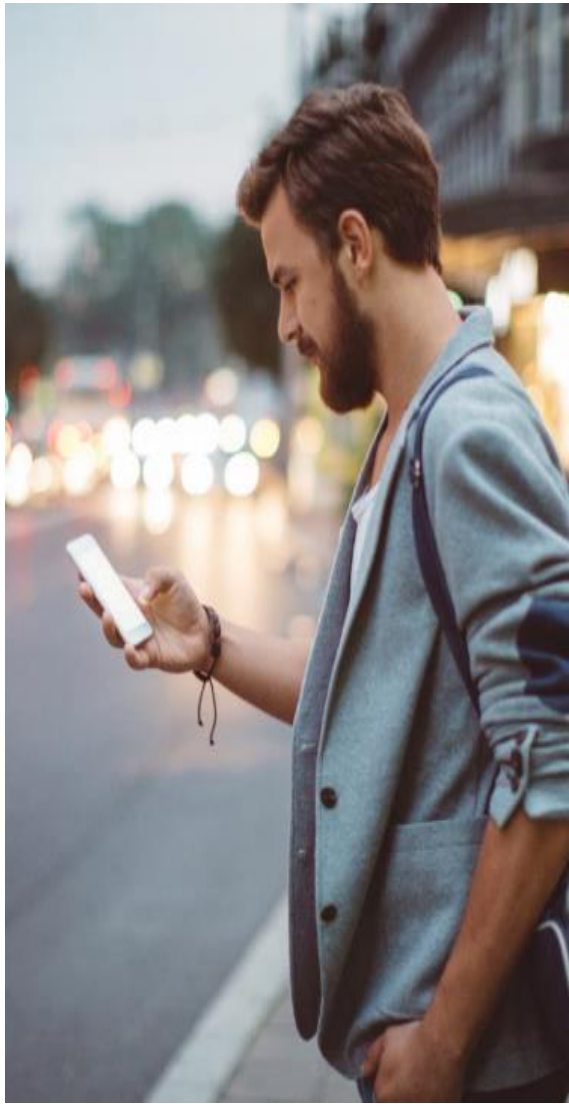
CS352-ADVANCED DATABASES

DR. JOHN CONKLIN

Agenda

Today's Topics:

1. Overview of the Database System Development Life Cycle (DSDLC)
2. Stage 1: Requirements Collection and Analysis
3. Stage 2: Conceptual Design (Enhanced ERD)
4. Stage 3: Logical Design & Normalization
5. Stage 4: Physical Design & SQL Implementation
6. Stage 5–7: Testing, Deployment, and Maintenance
7. Tying the DSDLC to CS352 Project Milestones
8. Individual Project



Overview of DSDLC

Definition: A structured process used for planning, creating, testing, and deploying a database system.

Importance: Ensures accuracy, efficiency, scalability, and future-proofing.

Aligns with ANSI-SPARC architecture for layered design.

Overview

What is the DSDLC?

Why is it Important?

- Ensures the database supports business processes
- Reduces redundancy and increases data integrity
- Allows for thoughtful planning
- Minimizes costly rework

Overview

Core Characteristics:

- Iterative and incremental
- Heavily reliant on communication
- Closely aligned with the 3-level ANSI-SPARC Architecture
- Focuses on translating real-world data

End Goal:
To create a reliable, normalized, secure, and performant database system that meets both current and future information needs.

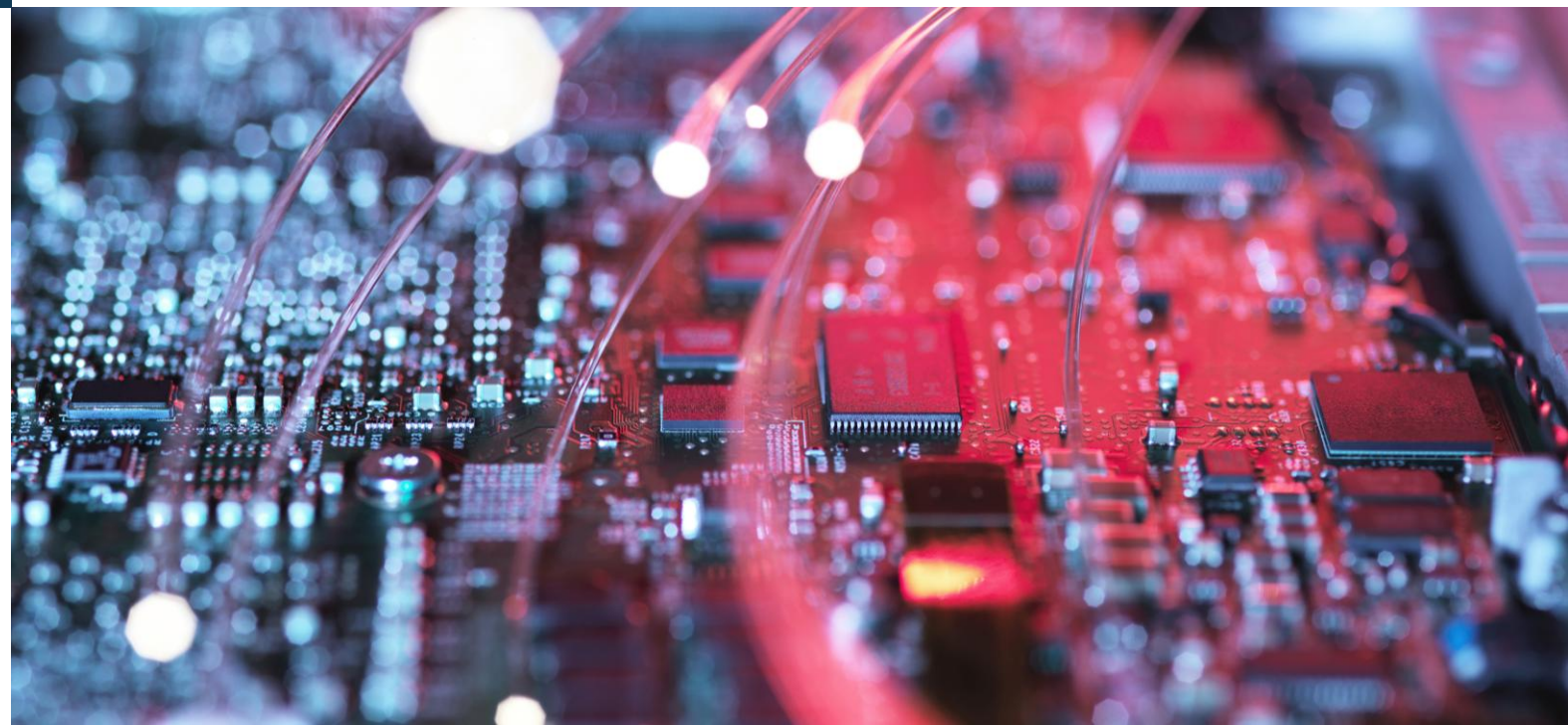
Stages of the DSDLC

1. Requirements Collection and Analysis
2. Conceptual Design
3. Logical Design
4. Physical Design
5. Implementation
6. Testing & Evaluation
7. Operation & Maintenance

Requirements Collection and Analysis

- Objectives of This Phase
- Techniques Used
- Deliverables

Requirements Collection and Analysis is the **first and most critical stage** in the Database System Development Life Cycle (DSDLC). This phase lays the foundation for all design decisions by capturing what the system *must* do from a functional and business standpoint.

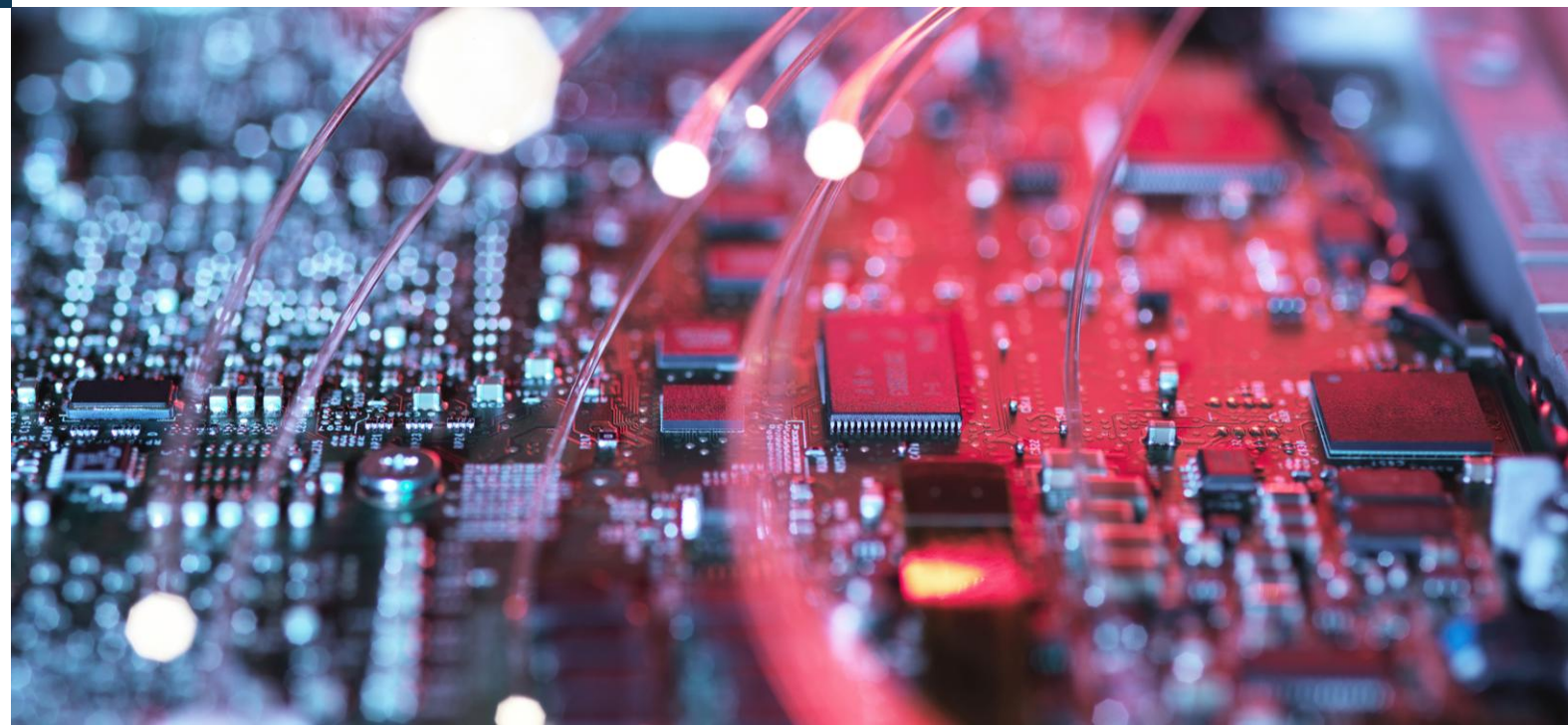


Conceptual Design

- Objectives of This Phase
- Key Features of EERDs
- Deliverables

This phase ensures a **DBMS-neutral blueprint** is in place before normalization or implementation begins, fostering clarity and alignment among stakeholders.

The Conceptual Design phase transforms user and business requirements into a **visual and semantic data model**—specifically, an Enhanced Entity-Relationship Diagram (EERD). This model is **independent of any DBMS** and focuses on the **what** rather than the **how** of data storage.

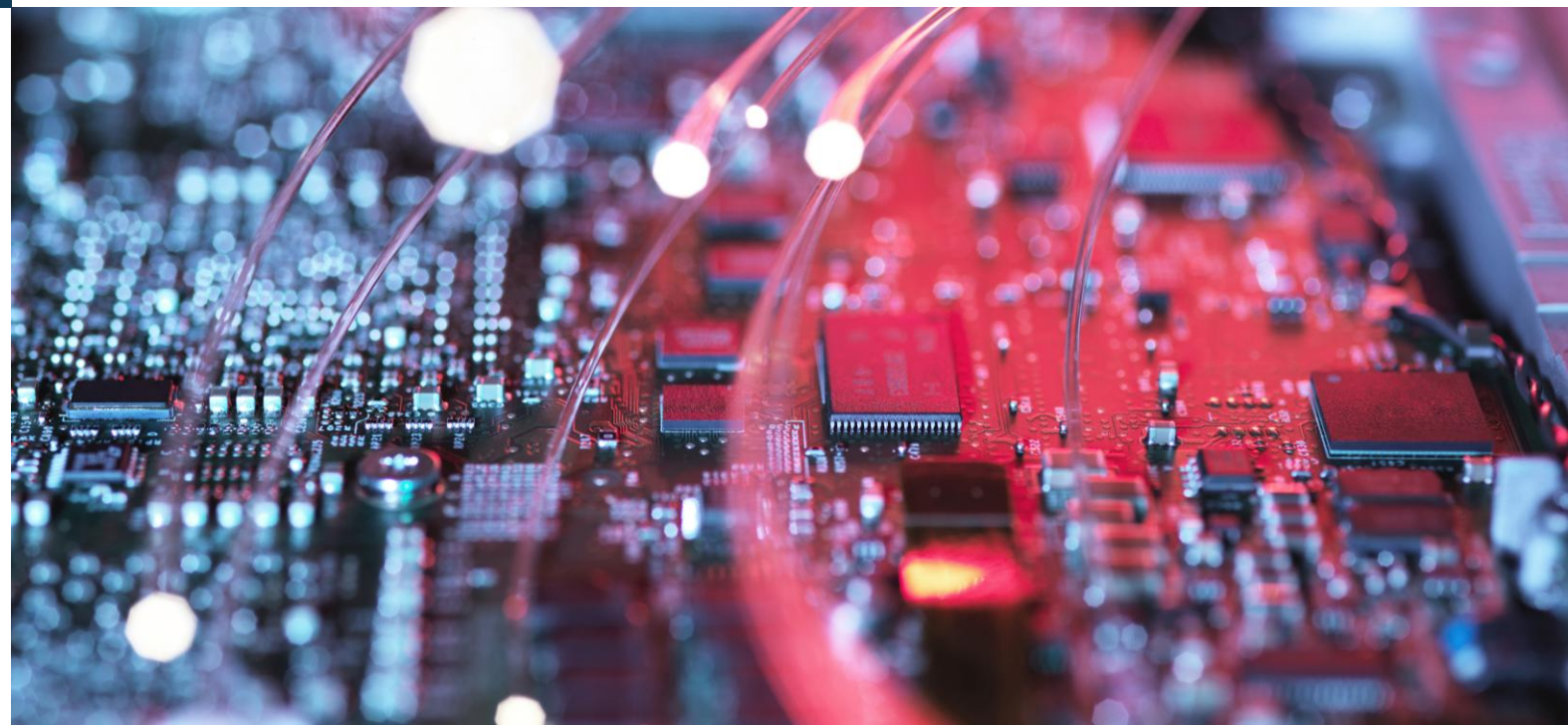


Logical Design

- Objectives of This Phase
- Normalization Process
- Deliverables

This phase bridges the gap between business modeling and physical implementation, ensuring the structure is logically sound and ready for SQL-based deployment.

Logical Design is the phase where the **conceptual model (EERD)** is translated into a **normalized relational schema**. Unlike conceptual design, logical design introduces **formal structure**, such as primary keys, foreign keys, and table relationships, but it still remains **independent of any specific DBMS**.

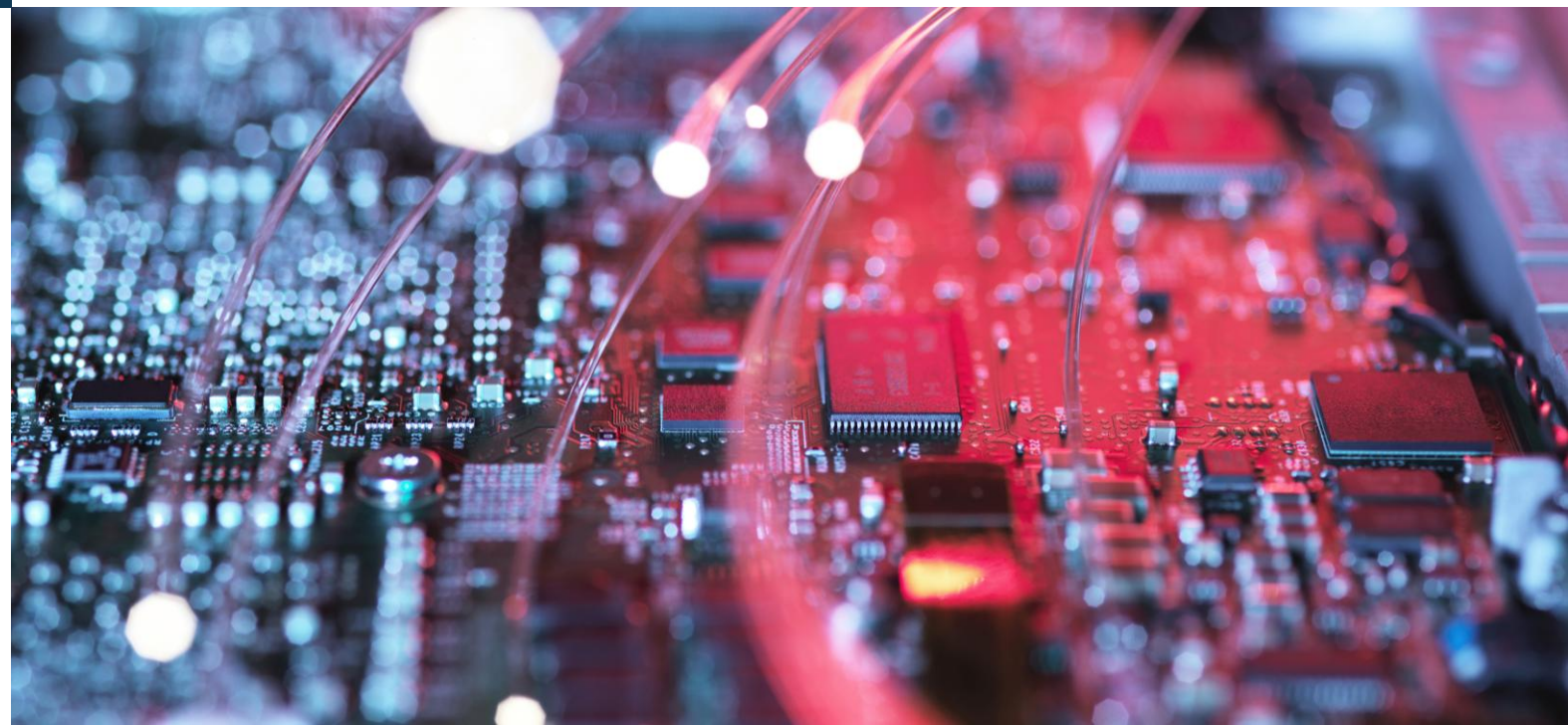


Physical Design

- Objectives of This Phase
- Core Tasks
- Deliverables

This phase implements the **internal level** of the ANSI-SPARC architecture, ensuring the system is ready for testing and deployment in a live environment.

Physical Design is the stage where the **logical schema is transformed into actual SQL structures** tailored to the chosen Database Management System (DBMS). It addresses performance, storage, and system-specific features while ensuring data integrity and consistency.

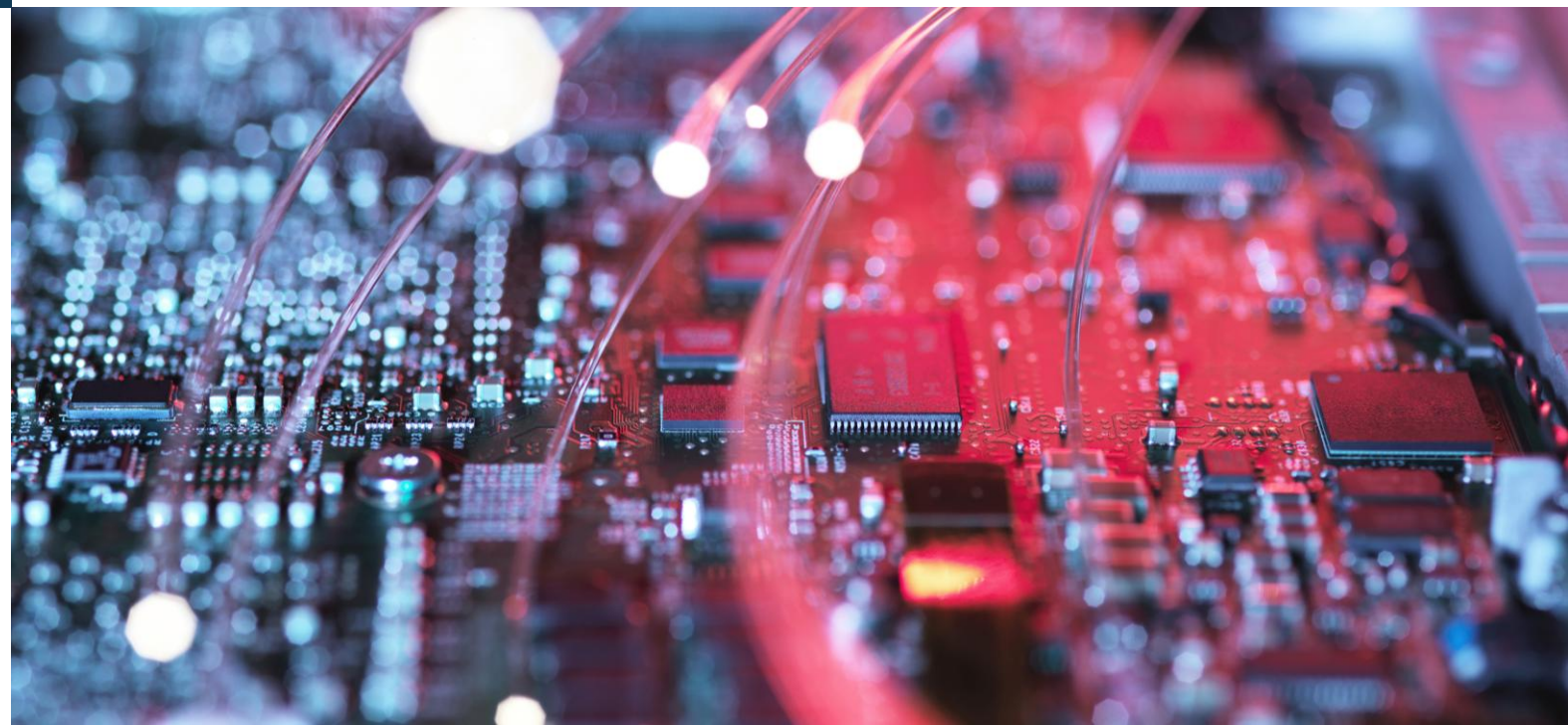


Implementation

- Objectives of This Phase
- Key Implementation Steps
- Deliverables

Implementation confirms that the **design translates into a functional, accessible, and accurate system**, and sets the stage for testing, optimization, and maintenance.

Implementation is the phase where the physical database is **installed, configured, and populated with real data**. It marks the transition from design to operational deployment and validates that the structure works as intended through real-world use cases.

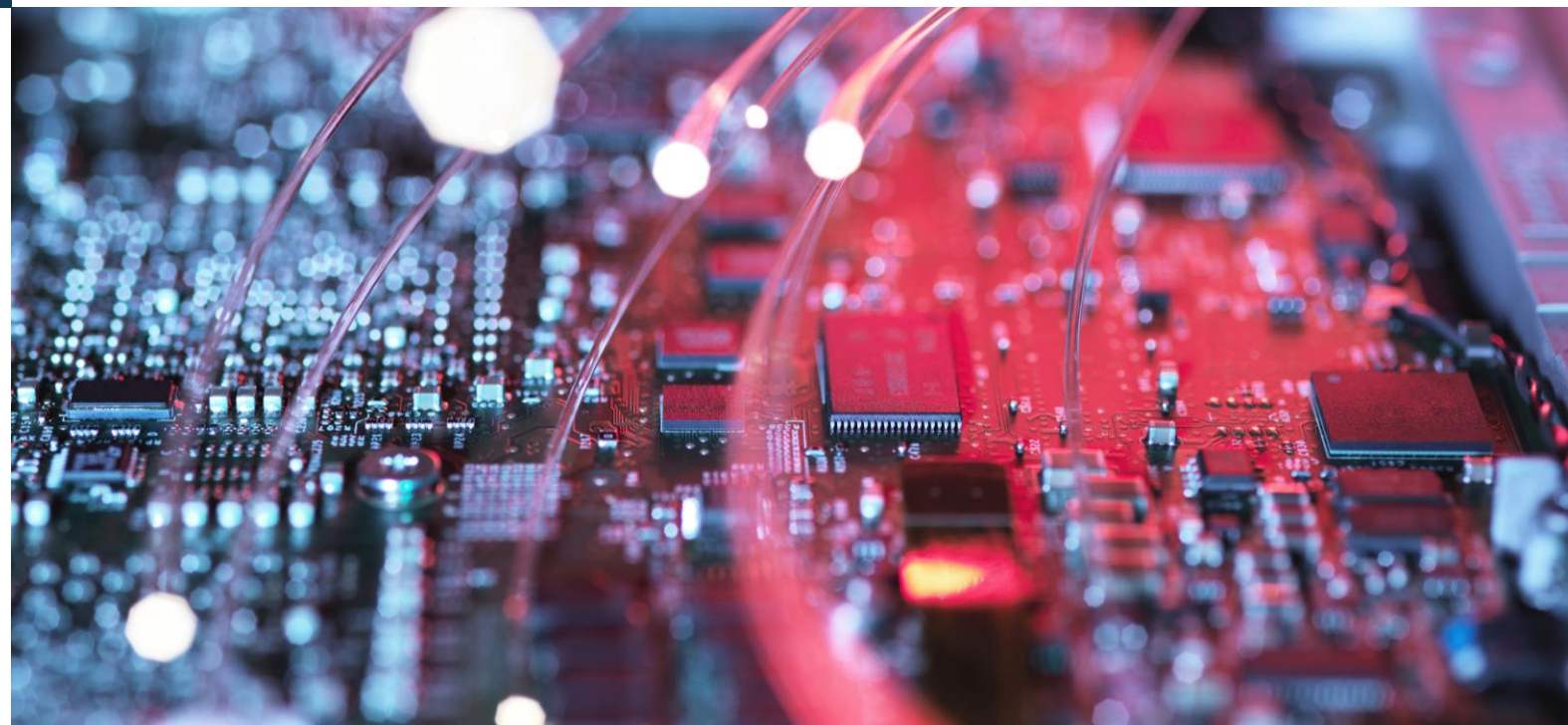


Testing & Evaluation

- Objectives of This Phase
- Testing Areas
- Deliverables

This phase assures that the database is ready for deployment and confirms it aligns with the **conceptual and logical models** developed earlier.

The Testing and Evaluation phase ensures that the **implemented database functions correctly, efficiently, and securely**. It verifies that the system meets the original requirements and is free of logical, performance, and data integrity issues.



Operation & Maintenance

- Objectives of This Phase
- Key Tasks
- Deliverables

This phase is crucial for maintaining the system's security, efficiency, and alignment with future data demands, encompassing the full life cycle from analysis to long-term support.

Operation and Maintenance is the **final phase** of the Database System Development Life Cycle (DSDLC). Once deployed, the database must be **actively monitored, maintained, and updated** to ensure long-term performance, accuracy, and reliability.

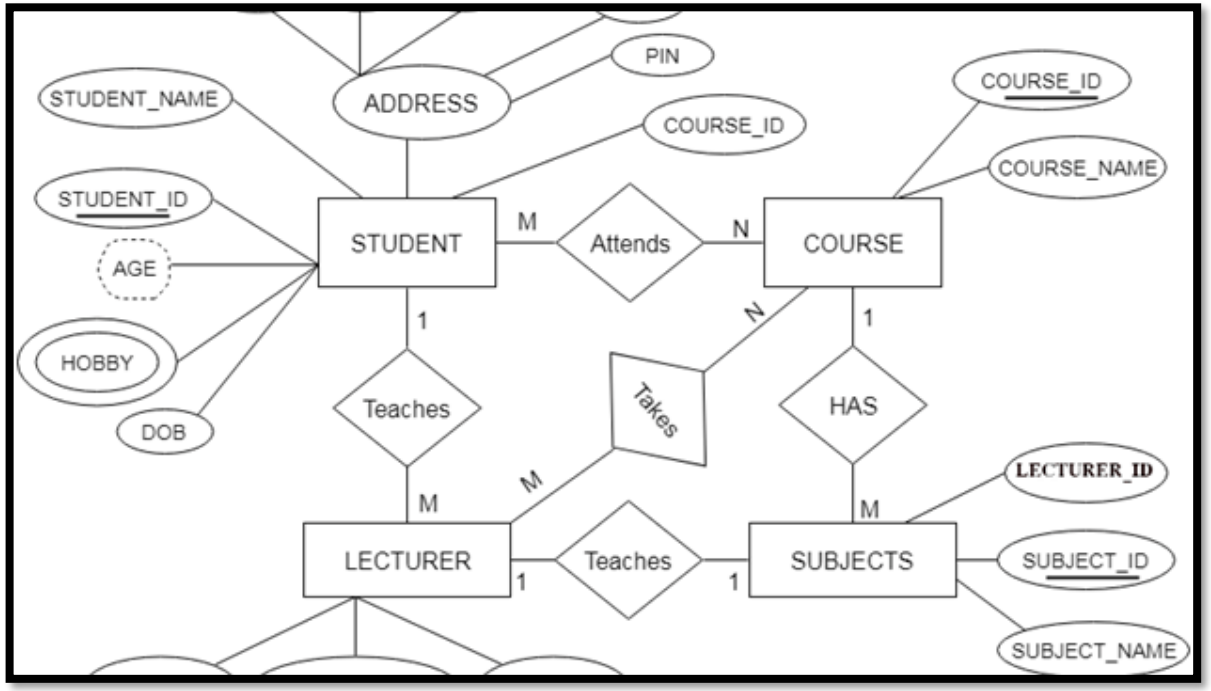
Connection to Student Projects

Unit 2 IP: ERD and relationship multiplicity

Unit 3 IP: Normalization to BCNF

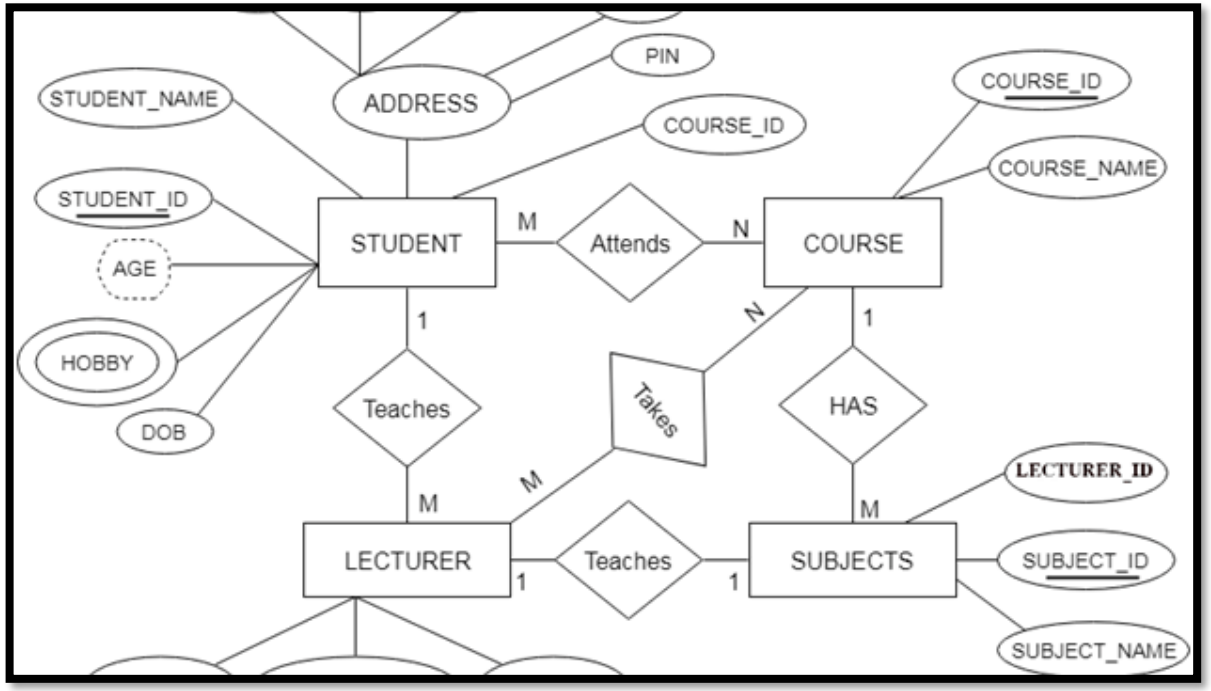
Unit 4 IP: DDL and DML implementation

Unit 5 IP: Star schema and ETL process



DSDLC Phase → Project Connection

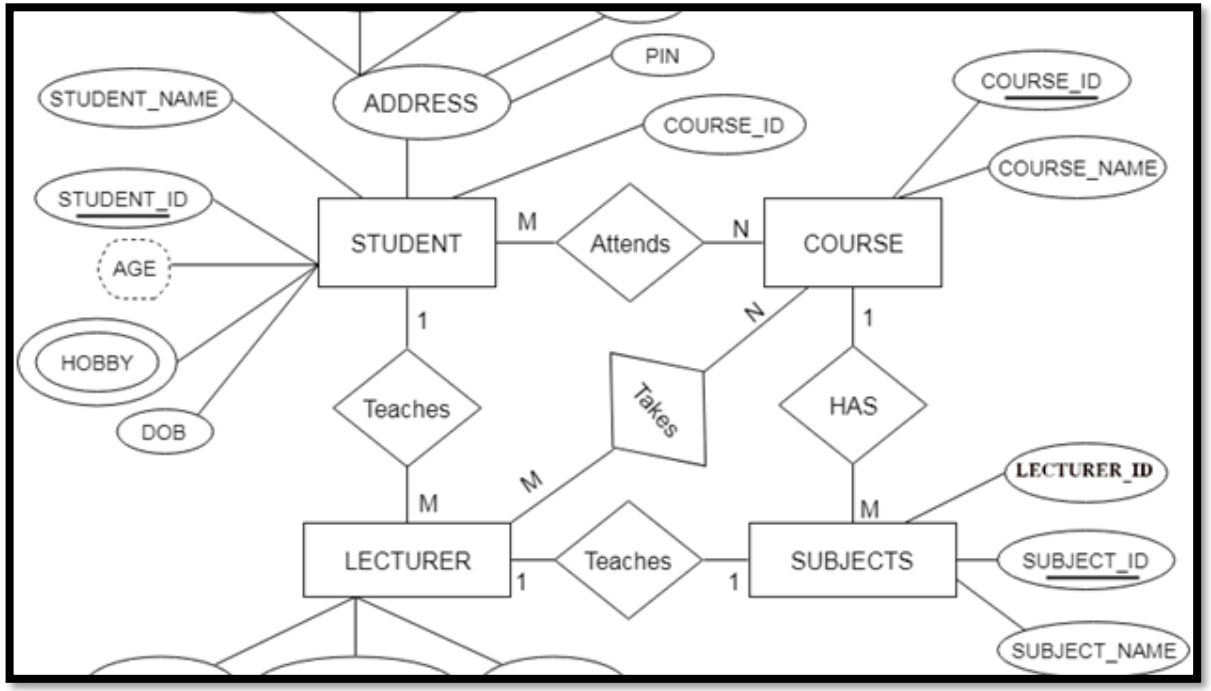
Each phase of the DSDLC directly aligns with your CS352 Individual Projects (IP1–IP5). Understanding this mapping will help you structure your work, meet expectations, and connect academic theory to practical database development.



Requirements Collection & Analysis

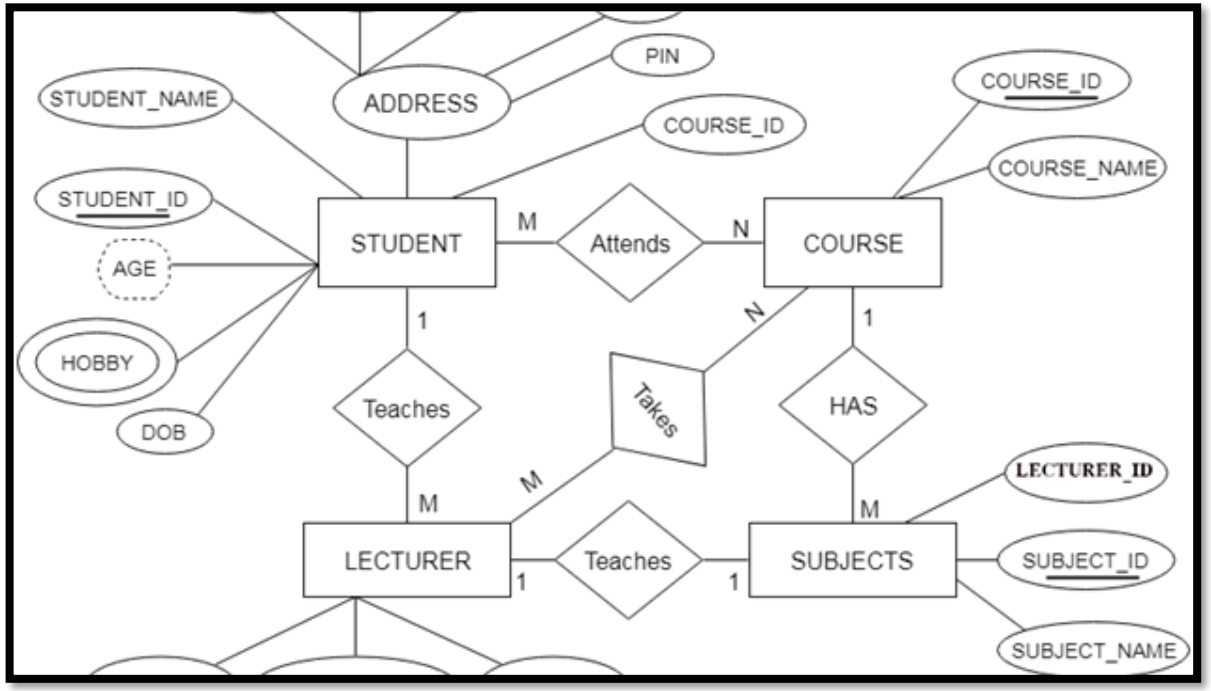
→ IP1 Discussion & Planning

- You reviewed the current fragmented system and discussed the need for a centralized database and data warehouse.
- You identified business needs and roles like Data Administrator (DA) and Database Administrator (DBA), setting the foundation for the rest of the project.



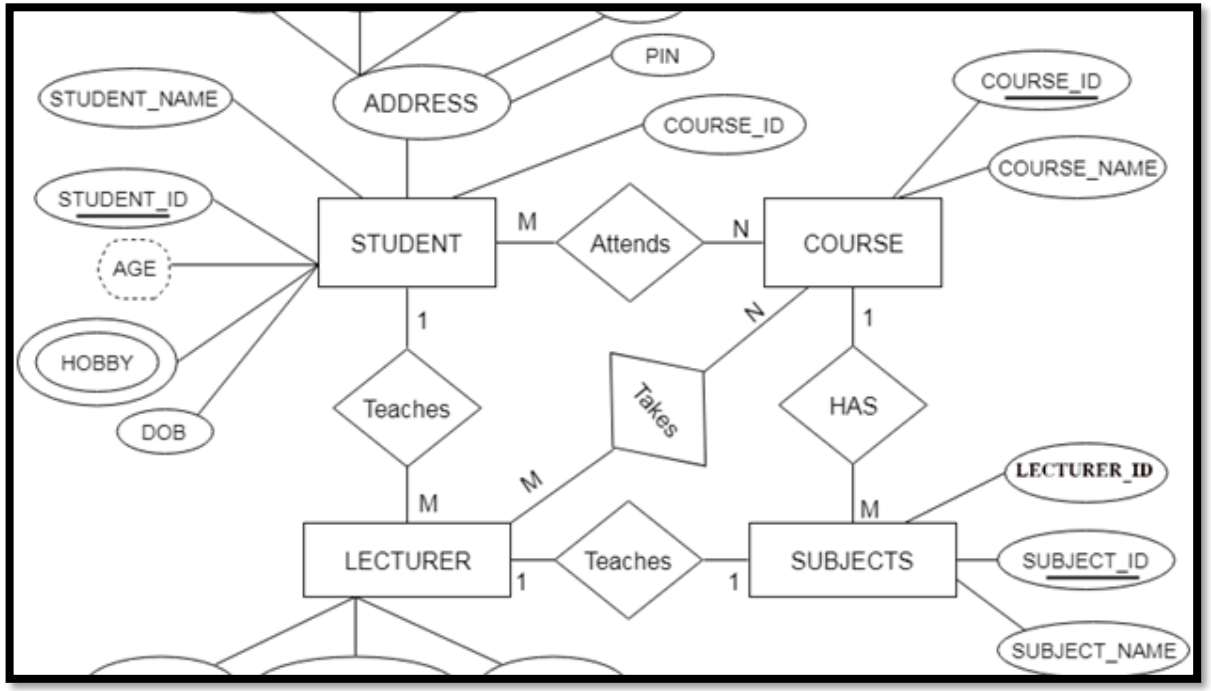
Conceptual Design → *IP2: Enhanced ERD*

- You will build an Enhanced Entity-Relationship Diagram (EERD) with disjoint subclasses (e.g., internal vs. customer-facing employees).
- You will explain multiplicity and relationship design decisions, such as how Orders resolve M:N customer-product relationships.



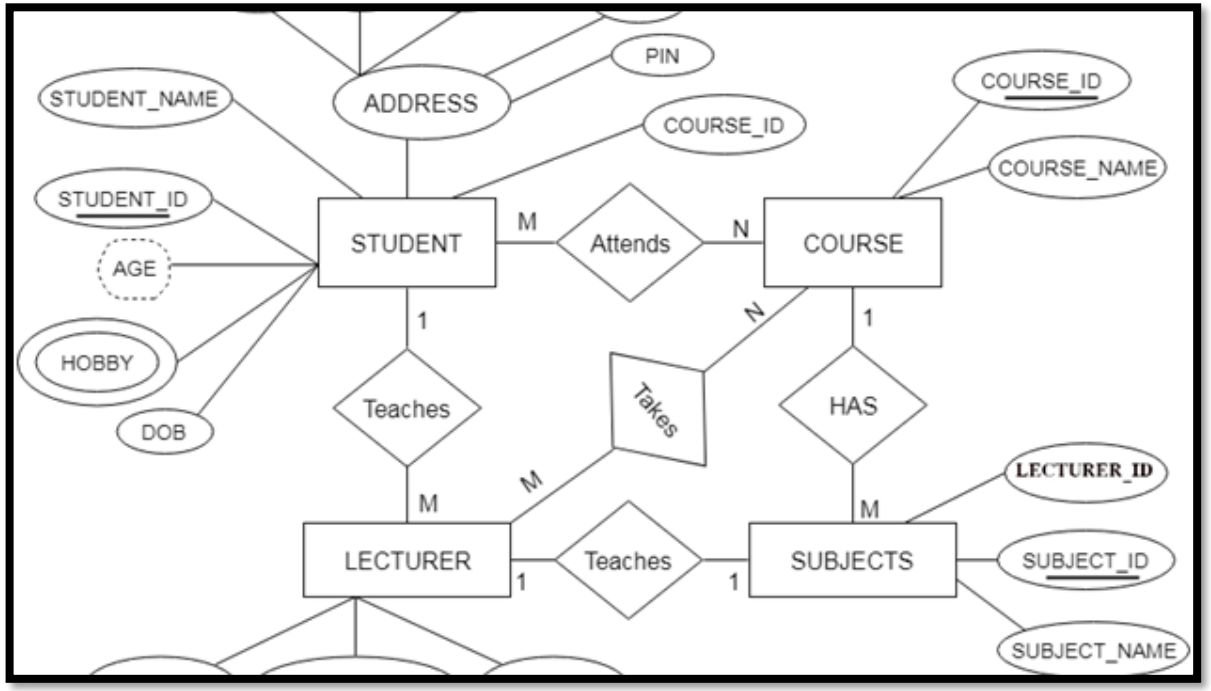
Logical Design → *IP3: Normalization and Logical Schema*

- You will normalize an unstructured table through UNF → 1NF → 2NF → 3NF → BCNF.
- You will convert your conceptual model (EERD) into a logical relational schema with appropriate keys.



Physical Design → *IP4: SQL DDL & DML*

- You will use SQL to implement the schema: creating tables, defining primary/foreign keys, and writing insert, update, and delete statements.
- You will include SELECT queries to demonstrate real usage, including JOINs.



Implementation, Testing, Maintenance → *IP5: Final Adjustments, Star Schema, ETL*

- You will revise and finalize the database.
- You will create a data warehouse star schema (Fact table + 4 Dimension tables) and describe the ETL process to populate it from the OLTP system.

Individual Project

Description

After you have analyzed the existing material used by the company for their day-to-day duties, the current Access database, and the additional requirements that the current system does not meet, the following requirements, entity/attributes have been compiled:

• Customer information is tracked using 'standard' attributes.
• A customer may purchase products or services; if they purchase products or services, the following is tracked:
    - Product purchased
    - Date of purchase
    - Total of purchase
• A supplier may sell products or services; if they sold products or services, the following is tracked:
    - Product sold
    - Date of sale
    - Total of sale
    - Is item purchased available for resale

• Your company wants to keep the number of tables storing address information to a bare minimum (read this as 1).
• Customers can be both a "supplier" and "someone that purchased services," and it is not required that they be either.
• Employee information is tracked using standard attributes.
• An employee will be considered either customer-interfacing or internal support.
• If the employee is customer-facing, the following information is tracked:
    - Customer for interaction (note that a customer will only interface with a single employee)
    - Product specialty
    - Hours of training
    - Commission rate

Individual Project

Description

- If the employee is internal support, then the following is tracked:
    - Salary
    - Support area
- An employee can be either customer-facing or internal support, but not both.
- Your company wants to keep the number of tables storing generic employee information to a bare minimum (read this as 1).
- Finally, the company wants to track products or services offered. This should be a single table with typical attributes that describe inventory.
- Every customer that either makes a purchase or sells goods to the company must be associated with an employee.
- Every transaction that a customer makes with the company is stored/tracked, a customer may buy or sell many products, and a product is sold to more than one customer.

Create an enhanced Entity-Relationship Diagram (ERD) to meet these requirements. Ensure that entities are correctly defined and appropriate attributes are listed for each entity. Also, ensure that all entities are properly related.

Add your ERD as a screenshot to a Word document, and provide supporting discussion about the need for the enhanced diagram tools and the reasoning behind the multiplicity for the relationships.

Add the enhanced ERD and the discussion about the relationship multiplicity to your project template to the section titled "Database System Development Life Cycle."

Name the document CS352_<First and Last Name> IP2.docx, and submit the document for grading.

Resources

Database System Development Life Cycle (DSDLC)

Here are several high-quality resources that cover the **Database System Development Life Cycle (DSDLC)**, which you can use for teaching, student references, or presentation content. These resources span textbooks, academic articles, and trusted websites.

Here are some helpful online resources for students:

Coronel, C. & Morris, S. (2018).
Database Systems: Design, Implementation, & Management.

Although a textbook, Chapter 9 presents the DSDLC in the context of project management and system analysis.

Elmasri, R., & Navathe, S. B. (2016).
Fundamentals of Database Systems (7th ed.). Pearson.

A more theoretical examination of the conceptual, logical, and physical design stages.

IBM Knowledge Center – Database Design Concepts

https://www.ibm.com/docs/en

Practical DSDLC coverage within enterprise systems.

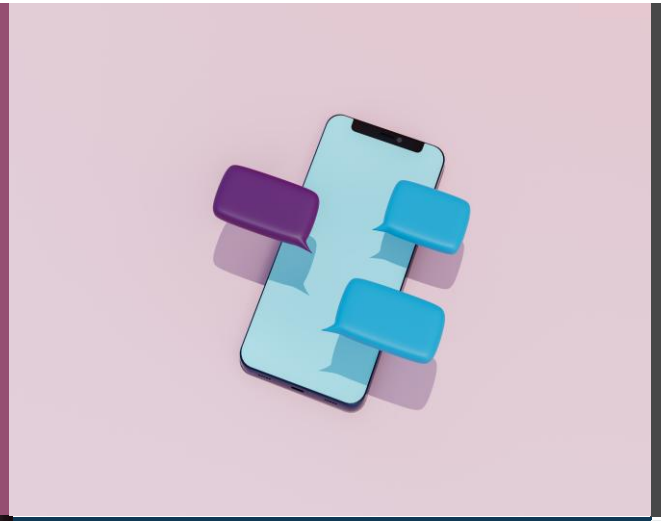
Vertabelo Academy

https://academy.vertabelo.com

Offers interactive courses on data modeling, entity relationship diagrams (ERDs), and database design steps.

INSTRUTOR

DR. JOHN CONKLIN



PHONE

602.796.5972

EMAIL

jconklin@coloradotech.edu

WEBSITE

drjconklin.com