# Module: Software Design

# Lesson: Data Modeling

## Overview

This 3-hour programming fundamentals class introduces students to data modeling concepts with a focus on database schemas and data flows. Students will learn the foundational principles of organizing and structuring data for software applications, using Python-friendly approaches.

## Section 1 (90 minutes)

### Topic 1: Introduction to Data Modeling (20 minutes)

* What is data modeling and why is it important in software design?
* Key principles of data organization and structure
* The relationship between data models and program functionality
* Common data modeling approaches in Python applications
* Real-world examples of data models in familiar applications

### Topic 2: Database Schema Basics (25 minutes)

* Understanding database schemas and their purpose
* Types of databases (relational vs. non-relational)
* Key components: tables, fields, records, relationships
* Primary and foreign keys
* Normalization concepts (1NF, 2NF, 3NF) simplified for beginners
* SQL vs. NoSQL approaches to data organization

### Topic 3: Object-Relational Mapping (ORM) Concepts (25 minutes)

* Bridging Python objects and database structures
* Introduction to ORM concepts (without specific frameworks)
* How Python classes can represent database tables
* Benefits of using ORM in application development
* Data typing in Python vs. database systems
* Common patterns for data access and manipulation

### Breakout Room 1: Designing a Simple Database Schema (25 minutes)

Students will work in small groups to:

1. Design a basic schema for a given scenario (e.g., online bookstore, student registration system)
2. Identify entities, attributes, and relationships
3. Sketch the schema using entity-relationship diagrams
4. Discuss potential challenges and trade-offs in their design
5. Consider how their schema would be represented in Python objects

## Section 2 (90 minutes)

### Topic 4: Data Flows in Software Applications (30 minutes)

* Understanding how data moves through an application
* Input validation and data cleaning
* Data transformation pipelines
* Managing state and persistence
* Handling data across different application layers
* Error handling and exception management for data operations

### Topic 5: Practical Data Modeling Patterns (35 minutes)

* Repository pattern for data access
* Data Transfer Objects (DTOs)
* Active Record vs. Data Mapper patterns
* Best practices for serialization and deserialization
* Caching strategies for optimized data access
* Testing data models and flows effectively

### Breakout Room 2: Implementing Data Flows (25 minutes)

Students will work in groups to:

1. Define a data flow diagram for a specified application feature
2. Identify key validation and transformation requirements
3. Plan error handling strategies at critical points
4. Discuss how their data model from the first breakout room would be used in this flow
5. Present their solution and receive feedback from peers

## Lesson Summary and Next Steps (10 minutes)

* Review of key data modeling concepts covered
* Common pitfalls in data modeling and how to avoid them
* Best practices for evolving data models as applications grow
* Preview of advanced topics (migrations, versioning, distributed data)
* Assignment of a mini-project to design and implement a simple data model in Python

## Additional Resources

* SQLite documentation: https://www.sqlite.org/docs.html
* Python's built-in sqlite3 module: https://docs.python.org/3/library/sqlite3.html
* "Database Design for Mere Mortals" by Michael J. Hernandez
* Online tool for creating ER diagrams: https://app.diagrams.net/
* "Clean Architecture" by Robert C. Martin (chapter on data modeling)

## Assessment Criteria

* Understanding of basic data modeling terminology and concepts
* Ability to design appropriate database schemas for simple applications
* Recognition of common data flow patterns and when to apply them
* Critical thinking about data integrity and validation requirements
* Collaboration skills demonstrated during breakout sessions