

CS240 • Databases & Data Modeling

School of Computational Arts & Sciences • Undergraduate • 6 ECTS

Overview

Model real domains in relational databases: schema design, normalization, SQL querying, indexing, and integrity constraints. Includes a mini project with audits and migrations, emphasizing data quality, reproducibility, and reasoning about consistency.

LOGISTICS

Credits: 6 ECTS

Level: Undergraduate

School: School of Computational Arts & Sciences

Prerequisites: None listed

Tags: databases, sql, data-modeling

Meeting time: Weekly lecture + SQL studio

Instruction mode: Model-first: schema design, constraints, and query reasoning

LEARNING OUTCOMES

You will be able to:

- Design relational schemas with constraints that protect integrity
- Write non-trivial SQL queries and explain their execution implications
- Evaluate indexing and normalization trade-offs
- Design relational schemas with constraints that protect integrity
- Write non-trivial SQL queries and explain their execution implications
- Evaluate indexing and normalization trade-offs

ASSESSMENT

Components

- Coursework: 60%
- Final project: 40%

Submissions include schema diagrams and an audit note describing how constraints prevent data corruption. Projects are graded on correctness, integrity, and clarity of the data model.

WEEKLY PLAN

Schedule

Week 1: Week 1

- Relational modeling: entities, relationships, keys

Week 2: Week 2

- SQL essentials: joins, aggregates, and subqueries

Week 3: Week 3

- Integrity: constraints, transactions, and concurrency basics

Week 4: Week 4

- Indexing: when it helps, when it hurts

Extended outline

- Relational modeling: entities, relationships, keys
- SQL essentials: joins, aggregates, and subqueries
- Integrity: constraints, transactions, and concurrency basics
- Indexing: when it helps, when it hurts
- Normalization vs denormalization: performance vs clarity
- Mini-project: design + implement a small database-backed app

POLICIES & RESOURCES

-
- Data safety: use synthetic datasets unless otherwise instructed.
 - Documentation: every schema change should have a reason.
 - Academic integrity: cite any schema patterns or references you used.

Suggested resources

- ER diagram template + checklist
- SQL query review rubric (correctness, clarity, performance hints)
- Transactions notes: isolation levels and common anomalies