Below is the **complete, end-to-end blueprint** for CSC 176—“Introduction to Relational Databases with SQL”—capturing every design decision, structure, module, lab, tool choice, and pedagogical strategy we’ve discussed. Treat this as the **single source of truth**: if you (the original instructor) were ever unavailable, this document should enable any educator to deliver exactly the intended experience.

**CSC 176: Introduction to Relational Databases with SQL**

**Semester:** Aug 18 – Dec 5, 2025  
**Meeting Time:** Thursdays 5:00 – 7:00 pm (no class Nov 27)  
**Final Project (“Bullseye”):** “Campus Tech Bazaar” E-Commerce Store

**Course Philosophy:**

* **Project-based & Inquiry-driven:** Students spend class “doing”—building features that culminate in the final GUI.
* **Student agency & safe failure:** Encourage experimentation, celebrate mistakes as learning.
* **Transparency:** Reveal the final product Day 1; no secret exams.
* **Low-anxiety grading:** Weekly “labwork” and mandatory Reflection Forms, plus a midterm demo and final presentation.
* **Inclusive, multi-modal:** In-class labs, interactive quizzes, live demos, optional extra-credit tracks.

**1. Core Competencies & Learning Objectives**

| **#** | **Competency** | **How We Address It** |
| --- | --- | --- |
| 1 | Fundamental Querying: SELECT, WHERE, ORDER BY, LIMIT, DISTINCT | Module A labs, Day 1 “SELECT \*” exercise |
| 2 | Aggregations & Grouping: COUNT, SUM, AVG, MIN, MAX, GROUP BY, HAVING | Module B labs: sales & genre trends |
| 3 | Multi-Table Operations: INNER/LEFT JOIN, subqueries, CTEs | Module C labs: cart, order history |
| 4 | Schema Design Principles: “one fact per column, one theme per table” | Flat→split schema puzzle; PK/FK constraints |
| 5 | Data Integrity & Transactions: BEGIN/COMMIT/ROLLBACK, error handling | Module D labs: batch imports, safe rollback |
| 6 | Performance & Indexing: EXPLAIN, CREATE INDEX | Module D performance lab |
| 7 | Tools & Environments: SQLiteOnline, DBeaver, GitHub basics | Week 1–2 setup; in-class demos |
| 8 | Professional Skills: Reflection, collaboration, Git workflows | Weekly Reflection Forms; structured pair roles; Git fork in Week 2 |

**2. Module & Week-by-Week Outline**

We break the 14 weeks into **four core modules** (A–D) plus an **optional stretch module** (E). Each module spans 2–3 weeks, with clear labs and deliverables.

| **Module** | **Weeks** | **Topics & Labs** | **Deliverables & Checks** |
| --- | --- | --- | --- |
| **A. Foundations & Ingestion** | 1 – 3 | • **Day 1:** Final GUI demo, survey, click-tour, SQLiteOnline/DBeaver setup, first SELECT lab • **Week 2:** SELECT, WHERE, ORDER BY, LIMIT, DISTINCT (“Mad Libs” queries) • **Week 3:** INSERT/UPDATE/DELETE; CSV import into flat orders\_flat | • Lab A1: “View All Products” • Lab A2: CRUD on customers & products • Pre-lab readiness quiz in LMS |
| **B. Aggregation & Grouping** | 4 – 5 | • COUNT, SUM, AVG, MIN, MAX • GROUP BY, HAVING, NULL handling • Lab: “Monthly sales” dashboard | • Lab B1: Revenue by month • LMS quiz + Reflection Form |
| **C. Relationships & Joins** | 6 – 8 | • Flat→split schema puzzle (no NF jargon) • FKs & constraints (SQLite PRAGMA foreign\_keys=ON) • INNER & LEFT JOINs across orders, customers, products • Lab: Cart & Checkout; Order History queries | • Lab C1: Refactor orders\_flat → normalized schema • Lab C2: Queries driving “My Orders” page |
| **D. Transactions & Performance** | 9 – 10 | • Explicit transactions: BEGIN/COMMIT/ROLLBACK; error demo • Batch import safety lab • EXPLAIN plans & CREATE INDEX • Lab: Optimize slow report query | • Lab D1: Safe CSV import with rollback • Lab D2: Index & plan comparison |
| **E. Advanced Queries (Stretch)** | 11 – 14 | • CTEs (WITH …) • Window Functions (ROW\_NUMBER, RANK, SUM OVER) • Subqueries: correlated vs. non • Midterm Demos (Week 11) • Project work, peer reviews, final rehearsals (Weeks 12–14) | • Optional Lab E1: Running totals in dashboard • Peer-review report |

**Key Dates & Assessments**

* **Midterm Checkpoint (Demo):** Week 11
* **Final Project Presentation:** Dec 5 (last class)
* **Weekly Reflection Form:** Due by next class (soft deadline)
* **Grades Due:** Midterm 10/13; Final 12/15

**3. Final Project & GUI Feature Roadmap**

**3.1 “Campus Tech Bazaar” Static GUI**

A static HTML/CSS/JS storefront hosted on GitHub Pages. UI shows all features Day 1; most are stubs. Students clone the repo, open index.html, and immediately see the “bullseye” target.

**3.2 Feature-to-Module Mapping**

| **Module** | **UI Feature** | **Stub Behavior Day 1** | **SQL Behind the Scenes** |
| --- | --- | --- | --- |
| **A** | View All Products | Works; shows products | SELECT \* FROM products; |
| **A** | Customer Signup | Form present; clicking shows “Not Implemented” | INSERT INTO customers(name,email)… |
| **B** | Monthly Sales Dashboard | Blank chart area | SELECT strftime('%Y-%m',order\_date) AS m,… SUM(quantity\*unit\_price) |
| **C** | Add to Cart & Checkout | Buttons do nothing | Two-phase transaction: insert into orders + order\_items |
| **C** | Order History | Empty list | SELECT … FROM orders o JOIN products p … |
| **D** | Upload Orders CSV | File-picker stub | Wrap INSERT statements in BEGIN…COMMIT |
| **D** | Query Plan Viewer | Shows placeholder text | EXPLAIN QUERY PLAN SELECT … |
| **E** | Top 10 Products Chart | Empty widget | Window function RANK() OVER (…) |

**Note:** Modules A–D cover all “must-have” features. Module E features are **extra credit**.

**4. Schema Puzzle Exercise (Module C)**

**Part 1: Flat Table**

* Load orders\_flat.csv into SQLiteOnline as orders\_flat.

**Flat-table labs:**

sql

CopyEdit

SELECT SUM(quantity\*unit\_price) AS revenue FROM orders\_flat;

SELECT DISTINCT customer\_name, customer\_email FROM orders\_flat;

SELECT customer\_name, SUM(quantity\*unit\_price) AS spent

FROM orders\_flat

GROUP BY customer\_name

ORDER BY spent DESC

LIMIT 1;

**Discussion:**

* How many rows for “Alex Kim”?
* If Alex’s email changes, how many updates?

**Part 2: Split Schema**

1. Create customers(id,name,email), products(id,name,unit\_price).
2. Populate via INSERT … SELECT DISTINCT ….
3. Create orders(order\_id,customer\_id,product\_id,quantity,order\_date) with FKs.
4. Compare flat vs. split queries: length, safety, duplication.

**5. Pedagogical & Engagement Strategies**

1. **Transparent Project Reveal** (Day 1).
2. **Structured Pairing** each lab: roles **Driver** (writes code) & **Navigator** (questions logic, takes notes), swap weekly.
3. **Multi-Modal Labs**:
   * **In-class:** Guided labs.
   * **Online Quizzes:** 3-4 readiness questions before each module.
   * **Interactive Tutorials/Videos:** Links to Mode Analytics, freeCodeCamp, W3Schools.
4. **Weekly Reflection Form** (Participation points):
   * Sentence stems:
     1. *“Today I learned …”*
     2. *“I’m still unclear about …”*
     3. *“If I had more time, I would explore …”*
   * Completed in class or as “homework” by next session.
5. **Peer Review & Mini-Workshops:**
   * GitHub fork setup in **Week 2** (no early friction).
   * Rotate triads every three weeks: one on SQL, one on UI stub testing, one on reflection synthesis.
6. **Low-Overhead Checks:**
   * Auto-graded Cobra LMS quizzes pre-module.
   * In-lab “muddiest point” polling.

**6. Tools & Tech Stack**

* **SQL Editor & Environment:** SQLiteOnline.com (single URL hosting starter.db).
* **Static GUI:** HTML/CSS/JS + [sql.js](https://github.com/sql-js/sql.js) for optional local querying. Hosted via **GitHub Pages**.
* **Version Control:** GitHub forks in Week 2; push lab solutions.
* **LMS Integration:** Cobra (Desire2Learn derivative) for quizzes, Reflection Forms, gradebook.

**7. Day 1 Detailed Plan**

| **Time** | **Activity** |
| --- | --- |
| 5:00 pm (20 min) | **Welcome & Theme Reveal** – Demo hosted storefront GUI (no code). – Explain semester journey. |
| 5:20 pm (10 min) | **Survey & Icebreaker** – “Why are you here?” group share. |
| 5:30 pm (15 min) | **Click Tour** – Students explore each button, note live vs. stub features. |
| 5:45 pm (15 min) | **Environment Setup** – Visit SQLiteOnline link; connect in DBeaver CLI. |
| 6:00 pm (30 min) | **Mini Lab: First SQL Query** – In pairs, SELECT \* FROM products; – View results; download CSV. |
| 6:30 pm (10 min) | **Debrief & Reflection Form Intro** – Hand out sentence-stem worksheet. – Explain soft deadline (next class). |
| 6:40 pm (20 min) | **GitHub Teaser** – Show repo, forking workflow (practice in Week 2). |
| 7:00 pm | **Adjourn** |

**Prep:**

* Host starter.db on SQLiteOnline with empty orders\_flat & products tables.
* Publish static GUI URL.
* Create Cobra quiz & Reflection Form template.

**8. Reflection Form Template**

**Weekly SQL Reflection** (due by next class; can be done in lab)

1. **“Today I learned…”**
2. **“I’m still unclear about…”**
3. **“If I had more time, I would explore…”**

**Grading:** 1 point each for completeness, honesty, and timeliness (max 3 points/week).

**9. Collaboration & Accessibility**

* **Pair Roles & Rotation** ensure novices aren’t overshadowed.
* **Triad Reviews** every 3 weeks for deeper peer feedback.
* **Accessibility Checks:**
  + High-contrast colors in GUI.
  + All images & buttons labelled for screen readers.
  + Transcripts for any embedded videos.

**10. Stretch Goals & Extra Credit**

* **Module E Labs:** CTEs, window functions, advanced analytics.
* **GitHub Badges (Optional):** “SELECT Master,” “Join Hero,” issued via Badgr if time permits.
* **No-Code Dashboard:** Metabase demo (instructor-only) for inspiration.

**This document encapsulates every facet** of CSC 176 as envisioned: the scaffolding, modules, labs, tools, and pedagogical strategies. Use it to onboard instructors, train TAs, or restore your meta-ideals if ever needed. Good luck, and may every student leave confident in their SQL mastery!