Syllabus: Practical SQL for Data Analysis and Applications

Define the meaning of each row, write queries across related tables, and verify results before using SQL in an application.

Course design and notes: Sunwoo Kim · Introductory · no CS background required

Course purpose

Define a data question and the meaning of each result row, write a query, and check the output against manually verified examples. Progress from filtering and aggregation to joins, staged queries, and window functions, then examine data changes, query plans, and application use.

The course emphasizes matching the grains of related tables and checking intermediate results. Query correctness, duplicate or missing records, execution cost, and transaction behavior are evaluated separately.

Sunwoo Kim designed this eight-week, self-paced course for learners without a computer science background. Original explanations and worked readings present the schema, example data, query, and expected result together, alongside official documentation.

Audience and prerequisites

The course is for researchers, engineers, and analysts who need to retrieve and summarize table-based data. Familiarity with spreadsheet rows and columns is expected. Schema, keys, and relationships are introduced before multi-table queries. Prior programming and database coursework are not required; the Python examples used to connect SQL to an application are explained when introduced.

Learning outcomes

By the end of the course, learners should be able to:

1. Read a schema and identify each table's grain, keys, and relationships.
2. Write filtering, aggregation, and conditional-summary queries with a defined treatment of missing values.
3. Join tables at compatible grains and check row counts, totals, and unmatched records.
4. Structure composite conditions with subqueries, CTEs, and existence checks; inspect the result of each stage.
5. Use window functions with explicit partitions, event order, tie handling, and relevant frame boundaries.
6. Check data changes and rollback behavior, compare query plans, and call parameterized SQL from a program.

Eight-week sequence

Week	Topic and scope	Required output
1	**Tables, grain, and basic queries.** Schema, keys, SELECT, WHERE, ORDER BY, DISTINCT, LIMIT, and NULL.	A basic query set with an explicit meaning for each output row.
2	**Aggregation and conditional summaries.** GROUP BY, aggregates, HAVING, CASE WHEN, and the denominator of an average.	A grouped report and manually checked counts, totals, and averages.
3	**Joins and matching grains.** Primary/foreign keys, 1:N relationships, INNER/LEFT JOIN, and aggregation before joining.	A multi-table query with row-count and total checks before and after each join.
4	**Subqueries, CTEs, and existence checks.** Subqueries, WITH, EXISTS/NOT EXISTS, staged aggregation, and composite conditions.	A multi-stage query with the grain, key, and expected rows written for every CTE.
5	**Window functions and event order.** ROW_NUMBER, RANK, LAG, SUM OVER, partitions, and explicit window frames.	Latest-record, change-from-previous, and cumulative-total queries.
6	**Schema constraints and controlled writes.** CREATE TABLE, types, NOT NULL/UNIQUE/CHECK, INSERT/UPDATE/DELETE, and transactions.	A small schema and a write exercise that checks affected rows and rollback behavior.
7	**Query plans and application use.** Indexes, EXPLAIN QUERY PLAN, parameter binding, permissions, and concurrent-write concepts.	A plan comparison and a parameterized query called from a small program.
8	**An integrated SQL workflow.** Combine earlier examples into a checked report or monitoring data layer; review dialect and pipeline boundaries.	Schema, queries, expected results, and a reproducible verification record.

Core sources

- Carter Zenke and David J. Malan, **CS50's Introduction to Databases with SQL**, Harvard University.