Sophia Learning

CS1011: Introduction to Relational Databases (3 semester credits)

COURSE DESCRIPTION

Sophia's Introduction to Relational Databases course will build mastery of foundational commands in SQL, with a focus on Postgresql. Students will actively practice writing SQL queries in a modifiable database. Their hands-on work will build their understanding of the key aspects of operating a database. Students will also get an introduction to challenges in administering and designing databases.

COURSE EFFECTIVE DATES: March 2021 - Present

PREREQUISITES: No prerequisites

LENGTH OF COURSE: This is a self-paced course. Students may use as much or as little time as needed to complete the course.

ACE CREDIT® RECOMMENDATION: In the lower-division baccalaureate/associate degree category, 3 semester hours in relational databases (3/21).

GRADING: This is a pass/fail course. Students must complete 19 Challenges (formative assessments) and 6 Milestones (summative assessments) with an overall score of 70% or better.

Challenges	Points Possible
Challenge 1.1: Single Table Query Foundations	10
Challenge 1.2: Single Table Construction	9
Challenge 1.3: Aggregate Function Queries	10
Challenge 1.4: Updating a Single Table	7
Challenge 2.1: Querying Multiple Tables	9
Challenge 2.2: Constructing Related Tables	5
Challenge 2.3: Specialized Queries	5
Challenge 2.4: Views	6
Challenge 3.1: Reliability	7

Challenge 3.2: Security	6
Challenge 3.3: Enhancement	4
Challenge 3.4: Management	4
Challenge 4.1: Utility	6
Challenge 4.2: Common Database Models	8
Challenge 4.3: Commercial Variety	4
Challenge 4.4: Constraints	5
Challenge 5.1: Entity-Relationship Diagram	6
Challenge 5.2: Normalization	7
Challenge 5.3: Applied Normalization	4
Total	**122**

Milestones	Points Possible
Milestone 1	108
Milestone 2	75
Milestone 3	63
Milestone 4	66
Milestone 5	51
Final Milestone	75
Total	**438**
Grand Total	**560**

For more general information on assessments, please visit the Student Guide located on your course dashboard.

LEARNING OUTCOMES

Upon completion of the course, the student will be able to:

1. Use Select-From-Where queries to collect information from a single table database.
2. Form a single table database using Create and Insert commands.
3. Use Group By, Order By, and aggregate functions.
4. Alter and update the database table by adding, deleting, or modifying columns.
5. Use Select-From-Where queries to combine information from multiple tables in a database using Joins.
6. Form a database with multiple tables using primary keys and foreign keys.
7. Demonstrate awareness of sub-queries, views, indices, and unions.

8. Summarize the importance of backing up data.
9. Evaluate multiple approaches to securing relational data.
10. Explore other methods of enhancing relational table design.
11. Explain the purpose and utility of a database.
12. Identify the most common database models used today.
13. Distinguish between several commercial relational DBMS.
14. Explain the purpose of an ERD.
15. Explain the purpose of normalization with respect to the relational data model.
16. Given an ERD depicting non-normalized data, normalize the data model.
17. Adapt RDMS use to business considerations.

OUTLINE OF MAJOR CONTENT AREAS

- SQL Clauses
- SELECT to Display Data
- ORDER BY to Sort Data
- WHERE to Filter Data
- LIKE to Search Data
- LIKE Wildcards
- Filter by Date
- Multiple Filters
- IN to Filter Data
- BETWEEN to Filter Data
- CREATE TABLE Syntax
- Table Constraints
- Primary Key and Auto-increment
- CHECK to Validate Data
- UNIQUE to Validate Data
- ALTER TABLE to Change Columns: Add/Drop
- ALTER TABLE to Change Columns: Data Type
- ALTER TABLE to Change Columns: Data Characteristics
- DROP TABLE to Remove Tables
- Aggregate Functions
- LIMIT and OFFSET to Cap Results
- MAX & MIN to Find Extremes
- COUNT to Count Records
- SUM to Add Values
- AVG to Average Values
- ROUND to Round Numbers
- GROUP BY to Combine Data

- HAVING to Filter On Aggregates
- Filters to Specify Data
- INSERT INTO to Add Row
- INSERT to Add Data
- INSERT INTO to Add Multiple Rows
- INSERT to Add Queried Data
- UPDATE to Edit Row
- UPDATE to Edit Multiple Rows
- DELETE FROM to Remove Row
- Joins
- Natural Joins
- JOIN USING to Link By Column
- JOIN ON to Link Tables
- AS/ALIAS to Rename Tables and Columns
- Outer Joins
- Left Joins
- Right Joins
- Cross Joins
- Foreign Keys & Creating Tables
- Foreign Keys & Altering Tables
- Foreign and Primary Keys
- Foreign Keys & Referential Data
- Foreign Key Errors
- Subqueries
- Subquery Performance
- Find Duplicate Rows
- Display Partial List
- UNION to Combine Results
- ANY and ALL Operators
- Calculations in SELECT Statements
- VIEW to Provide a Subset
- VIEW to Simplify Queries
- VIEW & Complex Queries
- CREATE OR REPLACE VIEW to Update Views
- DROP VIEW to Remove Views
- Transactions
- ACID Properties
- Atomicity
- Consistency
- Isolation

- Durability
- COMMIT and ROLLBACK to Manage Changes
- CREATE USER/ROLE to Add Users
- CREATE ROLE to Create Groups
- GRANT to Assign Users
- GRANT to Assign Privileges
- Application Security
- Superusers
- Index Overview
- B-Tree Index
- Hash Index
- DROP INDEX to Remove Indexes
- Create a Backup
- Restore from Backup
- Backups: Command Line vs. GUI
- Backup Methods
- Database Purpose
- Databases in the Real World
- Databases vs. Flat Files
- Parts of a Database System
- Non-relational Databases
- Data Warehouse vs. Transactional Databases
- Data Model Innovations
- Hierarchical and Network Data Models
- Relational Models
- Entity-Relationship Model
- Object and Relational Models
- Conceptual Design
- Logical Design
- Physical Design
- Commercial Databases
- Using ANSI SQL
- Using SQLite
- Using MySQL and MariaDB
- Considering Business Rules
- Translating Business Rules
- Storage Limitations
- Query Processing Bottlenecks
- Migrating Databases in an Organization
- Tables as Entities

- Adding Attributes
- Relationships and Cardinality
- Weak Relationships and Entities
- Associative Entities
- Reference Tables
- Normalization Overview
- Normal Forms
- First Normal Form
- Second Normal Form
- Third Normal Form
- Higher Normalization
- Denormalization
- ERD Connection Traps
- ERD Example: eCommerce
- ERD Example: Complexity
- ERD Example: Movie Ratings