



CS660 – Database systems

Dr. John Conklin

# SQL FUNDAMENTALS

A Comprehensive Guide to Structured Query Language

---

# AGENDA

---

1

## SQL Language Overview

Understanding SQL's role in databases

2

## Data Definition (DDL)

Creating and modifying database structures

3

## Data Manipulation (DML)

Adding, updating, and deleting data

4

## Determining Table Rows

Querying and filtering data

5

## Formulating SQL Statements

Best practices and syntax

# SQL LANGUAGE

---

## What is SQL?

SQL (Structured Query Language) is a standard programming language for managing relational databases.

### Key Characteristics:

- Declarative language - you specify what you want, not how to get it
- Standardized across database systems
- Case-insensitive keywords
- Used for querying and managing data

## SQL Categories

**DDL** Data Definition Language  
*CREATE, ALTER, DROP*

**DML** Data Manipulation Language  
*SELECT, INSERT, UPDATE, DELETE*

**DCL** Data Control Language  
*GRANT, REVOKE*

**TCL** Transaction Control Language  
*COMMIT, ROLLBACK*

# DATA DEFINITION (DDL)

---

DDL commands define and modify database structure

## CREATE TABLE - Define a new table

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    EnrollmentDate DATE  
);
```

## DROP TABLE - Delete a table

```
-- Remove table completely  
DROP TABLE Students;  
  
-- Remove if exists (safe)  
DROP TABLE IF EXISTS Students;
```

## ALTER TABLE - Modify existing table

```
-- Add a new column  
ALTER TABLE Students  
ADD Email VARCHAR(100);  
  
-- Modify column type  
ALTER TABLE Students  
MODIFY LastName VARCHAR(75);
```

## Key Points

- DDL commands auto-commit
- Structure changes affect all data
- Always backup before ALTER/DROP

# DATA MANIPULATION (DML) - INSERT

---

## Adding data to tables

### INSERT - Single Row

```
INSERT INTO Students
  (StudentID, FirstName,
   LastName, EnrollmentDate)
VALUES
  (1, 'John', 'Doe',
   '2024-09-01');
```

### INSERT - Multiple Rows

```
INSERT INTO Students VALUES
  (2, 'Jane', 'Smith', '2024-09-01'),
  (3, 'Bob', 'Johnson', '2024-09-02'),
  (4, 'Alice', 'Williams', '2024-09-02');
```

### INSERT - From Another Table

```
-- Copy students who enrolled before a certain date
INSERT INTO ArchivedStudents (StudentID, FirstName, LastName)
SELECT StudentID, FirstName, LastName
FROM Students
WHERE EnrollmentDate < '2024-01-01';
```

# DATA MANIPULATION (DML)

## UPDATE & DELETE

---

### UPDATE - Modify existing data

```
-- Update single column
UPDATE Students
SET Email = 'john@email.com'
WHERE StudentID = 1;

-- Update multiple columns
UPDATE Students
SET FirstName = 'Jonathan',
    Email = 'jonathan@email.com'
WHERE StudentID = 1;
```

### DELETE - Remove rows

```
-- Delete specific rows
DELETE FROM Students
WHERE StudentID = 4;

-- Delete with condition
DELETE FROM Students
WHERE EnrollmentDate < '2020-01-01';

-- Delete all rows (use carefully!)
DELETE FROM Students;
```



### CRITICAL SAFETY TIPS

- **Always** use WHERE clause with UPDATE and DELETE to avoid modifying/deleting all rows!
- Test your WHERE clause with a SELECT statement first
- Use transactions (BEGIN, COMMIT, ROLLBACK) for critical changes
- Keep backups before running UPDATE/DELETE on production data

# DETERMINING TABLE ROWS

## SELECT

---

Retrieving data from tables

### Basic SELECT

```
-- Select all columns
SELECT * FROM Students;
```

```
-- Select specific columns
SELECT FirstName, LastName
FROM Students;
```

### ORDER BY - Sort results

```
-- Ascending order (default)
SELECT * FROM Students
ORDER BY LastName ASC;
```

```
-- Descending order
SELECT * FROM Students
ORDER BY EnrollmentDate DESC;
```

### WHERE - Filter rows

```
SELECT FirstName, LastName
FROM Students
WHERE EnrollmentDate >= '2024-01-01';
```

```
-- Multiple conditions
SELECT * FROM Students
WHERE LastName = 'Smith'
  AND EnrollmentDate > '2024-01-01';
```

### LIMIT - Restrict row count

```
-- Get first 10 rows
SELECT * FROM Students
LIMIT 10;
```

```
-- Skip first 5, get next 10
SELECT * FROM Students
ORDER BY StudentID
LIMIT 10 OFFSET 5;
```

# ADVANCED FILTERING

---

## LIKE - Pattern matching

```
-- Starts with 'J'  
SELECT * FROM Students  
WHERE FirstName LIKE 'J%';  
  
-- Contains 'son'  
SELECT * FROM Students  
WHERE LastName LIKE '%son%';  
  
-- Exactly 4 characters  
WHERE FirstName LIKE '____';
```

## IS NULL / IS NOT NULL

```
-- Find rows with no email  
SELECT * FROM Students  
WHERE Email IS NULL;  
  
-- Find rows with email  
SELECT * FROM Students  
WHERE Email IS NOT NULL;
```

## IN & BETWEEN

```
-- IN - match list of values  
SELECT * FROM Students  
WHERE StudentID IN (1, 3, 5);  
  
-- BETWEEN - range of values  
SELECT * FROM Students  
WHERE EnrollmentDate  
    BETWEEN '2024-01-01' AND '2024-12-31';
```

## DISTINCT - Unique values

```
-- Get unique last names  
SELECT DISTINCT LastName  
FROM Students;  
  
-- Unique combinations  
SELECT DISTINCT LastName, City  
FROM Students;
```

# FORMULATING SQL STATEMENTS

---

Best practices for writing effective SQL

## Syntax Best Practices

- Use consistent capitalization
  - Keywords in UPPERCASE
  - Tables/columns in lowercase
- Indent for readability
- Use meaningful aliases
  - Students AS s
  - Enrollments AS e
- Always specify column names in INSERT
- Use comments to explain complex logic
  - Single line: -- comment
  - Multi-line: /\* comment \*/

## Performance Tips

- Select only needed columns
  - Avoid SELECT \*
- Use indexes on frequently queried columns
- Filter early with WHERE
  - Apply most restrictive filters first
- Avoid functions in WHERE clause on indexed columns
- Use LIMIT to test queries
- Use JOIN instead of subqueries when possible

# SQL STATEMENT EXECUTION ORDER

---

Understanding how SQL processes your query

- 1 FROM** Identify source tables
- 2 WHERE** Filter rows
- 3 GROUP BY** Group filtered rows
- 4 HAVING** Filter groups
- 5 SELECT** Choose columns
- 6 ORDER BY** Sort results
- 7 LIMIT** Restrict row count

```
SELECT FirstName, LastName, EnrollmentDate
FROM Students
WHERE EnrollmentDate >= '2024-01-01'
ORDER BY LastName
LIMIT 10;
```

# COMMON SQL FUNCTIONS

---

## Aggregate Functions

```
SELECT COUNT(*) AS TotalStudents,  
       AVG(Age) AS AverageAge,  
       MIN(EnrollmentDate) AS FirstEnrollment,  
       MAX(EnrollmentDate) AS LastEnrollment,  
       SUM(Credits) AS TotalCredits  
FROM Students;
```

## Date Functions

```
SELECT  
  CURRENT_DATE AS Today,  
  YEAR(EnrollmentDate) AS EnrollYear,  
  MONTH(EnrollmentDate) AS EnrollMonth,  
  DATEDIFF(CURRENT_DATE, EnrollmentDate) AS  
  DaysEnrolled,  
  DATE_ADD(EnrollmentDate, INTERVAL 1 YEAR) AS  
  Anniversary  
FROM Students;
```

## String Functions

```
SELECT  
  CONCAT(FirstName, ' ', LastName) AS FullName,  
  UPPER(LastName) AS UpperLast,  
  LOWER(Email) AS LowerEmail,  
  LENGTH(FirstName) AS NameLength,  
  SUBSTRING(Email, 1, 5) AS EmailPrefix  
FROM Students;
```

## GROUP BY with Aggregates

```
-- Count students per enrollment year  
SELECT YEAR(EnrollmentDate) AS EnrollYear,  
       COUNT(*) AS StudentCount  
FROM Students  
GROUP BY YEAR(EnrollmentDate)  
HAVING COUNT(*) > 10  
ORDER BY EnrollYear DESC;
```

# KEY TAKEAWAYS

---

- SQL is the standard language for relational databases
- DDL defines structure (CREATE, ALTER, DROP)
- DML manipulates data (INSERT, UPDATE, DELETE)
- SELECT retrieves data with powerful filtering options
- Always use the WHERE clause carefully to avoid data loss
- Practice writing clear, well-formatted SQL statements

*Happy Querying!* 

# INDIVIDUAL PROJECT



# Individual Project

The case study retail store has provided a list of reports and data manipulation tasks that are needed in the processing of orders for their customers. Answer the following:

- What structured query language (SQL) statement scripts are needed to create the database schema for the relational database system and manipulate the data in the solution that you are proposing to the company?
  - How does each of these scripts specifically support the goals and objectives of the company?
-

The project deliverables are as follows:

- Data Manipulation Tasks
  - Insert 20 records into each table for testing purposes.
  - Delete an entire order by using the unique identifier for that order.
  - Update the price of a product by using the unique identifier for that product.
  - Add a minimum of 3 of your own data manipulation language (DML) scripts based on the needs and specifications of your retail store.
- Report List
  - Total revenue (sales) per month, grouped by customer
  - Total revenue (sales) per month, grouped by product
  - Total count of products, grouped by category
  - Add minimum of 3 of your own report scripts based on the needs and specifications of your retail store (one must be a CROSSTAB)
- SQL (4–5 pages)
  - Include the database definition language (DDL) scripts to CREATE to database schema as described in the entity–relationship (E–R) diagram (Unit 2).
  - Include the database manipulation scripts (DML) that will be used to INSERT, DELETE, and UPDATE data in the proposed database system.
  - Include the SELECT, CROSSTAB, and AGGREGATE FUNCTION statements that will be used to read data from the proposed database system.
  - Provide your analysis as to how this part of the project fulfills the mission and 1 or more goals of the case study organization.
  - Provide the following attachments (in addition to embedding in document):
    - DDL.sql (including CREATE and INSERT statements so that they execute in the correct order [top-down])
    - DML.sql (including DELETE and UPDATE statements so that they can be executed in any order as selected)
    - REPORT.sql (including SELECT, CROSSTAB, AGGREGATE FUNCTION statements so that they can be executed in any order as selected)
  - **Note:** You will embed each script in the Word document and also provide it as an attachment.
- All sources should be cited both in-text and in References using APA format.
- Name the document "yourname\_CS660\_IP3.doc."

# Contact Information

|               |  |
|---------------|--|
|               |  |
| Email:        | <a href="mailto:Jconklin@coloradotech.edu">Jconklin@coloradotech.edu</a> |
| Phone:        | 602.796.5972   |
| Website:      | <a href="http://drjconklin.com">http://drjconklin.com</a>                |
| Office Hours: | Wednesdays: 6:00 PM – 7:00 PM (CST)                                      |
|               | Saturdays: 11:00 AM – 12:00 PM (CST)                                     |
| Live Chats:   | Wednesdays: 6:00 PM – 7:00 PM (CST)                                      |