SQL Roadmap

Basics of SQL
1. What is SQL?
2. Installing a Database Management System (MySQL, PostgreSQL, etc.)
3. Basic Database Concepts: Tables, Rows, Columns, Schemas
4. Writing your First SQL Query
Data Retrieval
5. SELECT Statement
6. WHERE Clause (Filtering Data)
7. DISTINCT Keyword
8. ORDER BY Clause (Sorting Data)
Data Manipulation
9. INSERT INTO Statement
10. UPDATE Statement
11. DELETE Statement
Working with Functions
12. Aggregate Functions: COUNT, AVG, SUM, MIN, MAX
13. String Functions: CONCAT, LENGTH, SUBSTRING
14. Date Functions: NOW, DATE, YEAR, MONTH
Filtering and Grouping
15. LIKE, IN, and BETWEEN Operators
16. IS NULL and IS NOT NULL
17. GROUP BY Clause
18. HAVING Clause

Joins
19. INNER JOIN
20. LEFT JOIN
21. RIGHT JOIN
22. FULL OUTER JOIN
23. Self Joins
Advanced Queries
24. Subqueries
25. EXISTS and NOT EXISTS
26. UNION and UNION ALL
27. CASE Statement
Database Design
28. Primary Keys and Foreign Keys
29. Data Types in SQL
30. Normalization and Denormalization
31. Indexes
Advanced Topics
32. Views
33. Stored Procedures
34. Triggers
35. Transactions and ACID Properties
Practice and Optimization
36. Practice SQL Problems (Basic to Advanced)
37. Writing Complex Queries

38. Query Optimization Techniques
39. Database Backup and Restore
Final Steps
40. Real-World Projects (e.g., building a library or e-commerce database)
41. Interview Questions and Problem Solving
42. Continuous Learning (Advanced topics, certifications, etc.)

SQL Practice

Creating Tables

Student

```
CREATE TABLE Students (
    StudentID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    DateOfBirth DATE NOT NULL,
    Gender CHAR(1) CHECK (Gender IN ('M', 'F')),
    Phone VARCHAR(15) UNIQUE,
    Email VARCHAR(100) UNIQUE NOT NULL,
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);
```


Departments

```
CREATE TABLE Departments (
    DepartmentID INT PRIMARY KEY AUTO_INCREMENT,
    DepartmentName VARCHAR(100) NOT NULL UNIQUE,
    HeadOfDepartment VARCHAR(100)
);
```

Courses

```
CREATE TABLE Courses (
    CourseID INT PRIMARY KEY AUTO_INCREMENT,
    CourseName VARCHAR(100) NOT NULL,
    Credits INT NOT NULL CHECK (Credits > 0),
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);
```

Instructors

```
CREATE TABLE Instructors (
    InstructorID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Phone VARCHAR(15) UNIQUE,
    Email VARCHAR(100) UNIQUE NOT NULL,
    HireDate DATE NOT NULL,
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);
```

Enrollments

```
CREATE TABLE Enrollments (
    EnrollmentID INT PRIMARY KEY AUTO_INCREMENT,
    StudentID INT,
    CourseID INT,
    InstructorID INT,
    EnrollmentDate DATE NOT NULL,
    Grade CHAR(2) CHECK (Grade IN ('A', 'B', 'C', 'D', 'F', 'P', 'I')),
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
    FOREIGN KEY (CourseID) REFERENCES Courses(CourseID),
    FOREIGN KEY (InstructorID) REFERENCES Instructors(InstructorID)
);
```


Library

```
CREATE TABLE Library (
    BookID INT PRIMARY KEY AUTO_INCREMENT,
    Title VARCHAR(100) NOT NULL,
    Author VARCHAR(100) NOT NULL,
    ISBN VARCHAR(20) UNIQUE,
```

```
    DepartmentID INT,
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)
);
```

BookLoans

```
CREATE TABLE BookLoans (
    LoanID INT PRIMARY KEY AUTO_INCREMENT,
    BookID INT,
    StudentID INT,
    LoanDate DATE NOT NULL,
    ReturnDate DATE,
    FOREIGN KEY (BookID) REFERENCES Library(BookID),
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
);
```

Inserting Data

Departments

INSERT INTO Departments (DepartmentName, HeadOfDepartment) VALUES

('Computer Science', 'Dr. Alice Johnson'),

('Mechanical Engineering', 'Dr. Bob Smith'),

('Electrical Engineering', 'Dr. Carol Lee'),

('Civil Engineering', 'Dr. David Brown'),

('Mathematics', 'Dr. Emma Wilson'),

('Physics', 'Dr. Frank Thomas'),

('Chemistry', 'Dr. Grace Miller'),

('Biology', 'Dr. Henry Adams'),

('Business Administration', 'Dr. Ivy Moore'),

('Economics', 'Dr. Jack White');

Student

INSERT INTO Students (FirstName, LastName, DateOfBirth, Gender, Phone, Email, DepartmentID) VALUES

('John', 'Doe', '2002-05-15', 'M', '9876543210', 'john.doe@example.com', 1),

('Jane', 'Smith', '2003-11-20', 'F', '9876543211', 'jane.smith@example.com', 2),

('Alice', 'Brown', '2001-03-10', 'F', '9876543212', 'alice.brown@example.com', 3),

('Bob', 'Taylor', '2000-12-05', 'M', '9876543213', 'bob.taylor@example.com', 4),

('Emma', 'Wilson', '2002-08-22', 'F', '9876543214', 'emma.wilson@example.com', 5),

('Frank', 'Thomas', '2003-06-18', 'M', '9876543215', 'frank.thomas@example.com', 6),

('Grace', 'Miller', '2001-01-12', 'F', '9876543216', 'grace.miller@example.com', 7),

('Henry', 'Adams', '2002-09-07', 'M', '9876543217', 'henry.adams@example.com', 8),

('Ivy', 'Moore', '2001-04-03', 'F', '9876543218', 'ivy.moore@example.com', 9),

('Jack', 'White', '2000-07-25', 'M', '9876543219', 'jack.white@example.com', 10);

Courses

INSERT INTO Courses (CourseName, Credits, DepartmentID) VALUES

('Data Structures', 4, 1),

('Thermodynamics', 3, 2),

('Circuit Analysis', 3, 3),

('Structural Engineering', 4, 4),

('Linear Algebra', 3, 5),

('Quantum Mechanics', 4, 6),

('Organic Chemistry', 4, 7),

('Genetics', 3, 8),

('Marketing Principles', 3, 9),

('Microeconomics', 3, 10);

Instructors

INSERT INTO Instructors (FirstName, LastName, Phone, Email, HireDate, DepartmentID) VALUES

('Dr. Alan', 'Turing', '9876543220', 'alan.turing@example.com', '2010-09-01', 1),

('Dr. Barbara', 'Liskov', '9876543221', 'barbara.liskov@example.com', '2008-01-15', 2),

('Dr. Charles', 'Babbage', '9876543222', 'charles.babbage@example.com', '2012-05-10', 3),

('Dr. Dorothy', 'Vaughan', '9876543223', 'dorothy.vaughan@example.com', '2005-11-20', 4),

('Dr. Edwin', 'Hubble', '9876543224', 'edwin.hubble@example.com', '2007-03-25', 5),

('Dr. Florence', 'Nightingale', '9876543225', 'florence.nightingale@example.com', '2003-07-30', 6),

('Dr. Gregor', 'Mendel', '9876543226', 'gregor.mendel@example.com', '2000-12-05', 7),

('Dr. Hedy', 'Lamarr', '9876543227', 'hedy.lamarr@example.com', '2009-08-15', 8),

('Dr. Isaac', 'Newton', '9876543228', 'isaac.newton@example.com', '1998-02-28', 9),

('Dr. James', 'Watson', '9876543229', 'james.watson@example.com', '2015-06-12', 10);

Enrollments

INSERT INTO Enrollments (StudentID, CourseID, InstructorID, EnrollmentDate, Grade) VALUES

(1, 1, 1, '2023-09-01', 'A'),

(2, 2, 2, '2023-09-01', 'B'),

(3, 3, 3, '2023-09-01', 'C'),

(4, 4, 4, '2023-09-01', 'B'),

(5, 5, 5, '2023-09-01', 'A'),

(6, 6, 6, '2023-09-01', 'C'),

(7, 7, 7, '2023-09-01', 'B'),

(8, 8, 8, '2023-09-01', 'A'),

(9, 9, 9, '2023-09-01', 'C'),

(10, 10, 10, '2023-09-01', 'B');

Library

INSERT INTO Library (Title, Author, ISBN, DepartmentID) VALUES

('Introduction to Algorithms', 'Thomas H. Cormen', '9780262033848', 1),

('Thermodynamics: An Engineering Approach', 'Yunus Çengel', '9780073398174', 2),

('Electric Circuits', 'James W. Nilsson', '9780134746968', 3),

('Structural Analysis', 'R. C. Hibbeler', '9780134605144', 4),

('Linear Algebra Done Right', 'Sheldon Axler', '9783319110790', 5),

('Principles of Quantum Mechanics', 'R. Shankar', '9781475705773', 6),

('Organic Chemistry', 'Paula Yurkanis Bruice', '9780134042282', 7),

('Genetics: A Conceptual Approach', 'Benjamin A. Pierce', '9781319050962', 8),

('Principles of Marketing', 'Philip Kotler', '9780134492513', 9),

('Microeconomics', 'Paul Krugman', '9781464188279', 10);

BookLoans

INSERT INTO BookLoans (BookID, StudentID, LoanDate, ReturnDate) VALUES

(1, 1, '2023-10-01', '2023-10-15'),

(2, 2, '2023-10-03', '2023-10-18'),

(3, 3, '2023-10-05', NULL),

(4, 4, '2023-10-07', '2023-10-22'),

(5, 5, '2023-10-10', NULL),

(6, 6, '2023-10-12', '2023-10-27'),

(7, 7, '2023-10-15', '2023-10-30'),

(8, 8, '2023-10-18', NULL),

(9, 9, '2023-10-20', '2023-11-05'),

(10, 10, '2023-10-25', NULL);

Basic Queries

1. Data Retrieval (SELECT Queries)

1. Retrieve all columns and rows from the Students table.
2. Display the first and last names of all students.
3. List all courses offered by the Computer Science department.
4. Find the details of students born after January 1, 2002.
5. Retrieve distinct department names from the Departments table.
6. Display the names of students enrolled in courses with 4 credits.
7. Find all books authored by "Paula Yurkanis Bruice."
8. Show the names and grades of students who received a grade of 'A.'
9. Retrieve the first and last names of instructors hired before 2010.
10. List all students and their email addresses.

2. Filtering with WHERE

11. Find courses with more than 3 credits.
12. List all male students in the Physics department.
13. Retrieve all instructors from the Business Administration department.
14. Find students whose phone number starts with '9876.'
15. Retrieve all books loaned out but not yet returned (ReturnDate IS NULL).
16. Show all enrollments where the grade is either 'A' or 'B.'
17. Find all books with the word "Principles" in their title.
18. Retrieve all students whose email domain is example.com.
19. List all departments with names starting with "C."
20. Find students born between 2001-01-01 and 2002-12-31.

3. Sorting (ORDER BY)

21. List all courses sorted alphabetically by course name.
22. Display students sorted by their last name in descending order.
23. Retrieve instructors sorted by their hire date (earliest first).
24. List books sorted by their title.
25. Retrieve the latest 5 enrollments based on enrollment date.

4. Aggregate Functions

26. Count the total number of students in the database.
27. Find the average number of credits for all courses.
28. Count the total number of books in the Library table.
29. Find the earliest hire date among all instructors.
30. Calculate the number of students enrolled in each course.

5. Grouping (GROUP BY)

31. Group students by their department and count the number of students in each.
32. Find the number of courses offered by each department.
33. Retrieve the total number of male and female students.
34. Count the number of books authored by each author.
35. List each grade along with the number of students who received it.

6. Joins

36. Retrieve the names of students and the departments they belong to.
37. List all courses along with their respective departments.
38. Display the names of instructors and the courses they teach.
39. Retrieve the names of students and the books they have borrowed.
40. List all students and the courses they are enrolled in.

7. Subqueries

41. Find the names of students who are enrolled in the course "Data Structures."
42. Retrieve the department with the maximum number of courses.
43. Find students who have not borrowed any books.
44. List courses that have no enrollments.
45. Find the most recently loaned book.

8. Modifying Data (INSERT, UPDATE, DELETE)

46. Add a new student named "Maya Patel" to the Students table.
47. Update the grade for all students in "Quantum Mechanics" to 'B.'
48. Delete all enrollments where the grade is 'F.'
49. Add a new course called "Machine Learning" to the Computer Science department.

50. Remove all books authored by "James W. Nilsson."

Intermediate Queries

1. Advanced Data Retrieval

1. Retrieve the full names of students with their department names.
2. Find students who are enrolled in multiple courses.
3. List all instructors who teach at least one course with 4 credits.
4. Retrieve the names of all students who have borrowed more than 3 books.
5. Show the details of students who have received the same grade in more than one course.
6. Find the titles of all books borrowed by students in the Computer Science department.
7. List all students who enrolled in a course taught by "Dr. Alan Turing."
8. Retrieve courses that are taught in more than one department.
9. Find books that were loaned out and returned on the same day.
10. List students who have not enrolled in any courses.

2. Filtering with Multiple Conditions

11. Retrieve courses with 3 or 4 credits offered by the Mathematics department.
12. Find students whose last name starts with "W" and who are in the Physics department.
13. List all books where the title contains "Introduction" but excludes "Advanced."
14. Retrieve all enrollments where the grade is not 'A' or 'B.'
15. Find instructors who were hired after 2010 and belong to the Electrical Engineering department.

3. Using Joins Effectively

16. Retrieve the names of students along with the names of courses they are enrolled in.
17. List all books along with the names of the students who borrowed them, even if the book is not currently loaned out.
18. Retrieve all departments along with the number of courses offered by each.
19. List instructors and the total number of students they teach across all courses.
20. Find the names of students who borrowed books from their own department's library.

4. Subqueries

21. Retrieve the names of students who are enrolled in the most popular course (course with the highest enrollment).
22. Find the names of instructors who teach the course "Thermodynamics."
23. Retrieve the title of the book borrowed by the oldest student.
24. Find courses that have more students enrolled than the course "Data Structures."
25. List the departments that do not have any courses.

5. Grouping and Aggregates

26. Find the average number of credits for courses in each department.
27. List each instructor along with the number of courses they teach.
28. Retrieve the total number of books loaned by each student.
29. Show the average grade (A=4, B=3, etc.) for each course.
30. Find the total number of male and female students in each department.

6. Nested and Correlated Subqueries

31. Retrieve the names of students who have borrowed all books authored by "Thomas H. Cormen."
32. Find the names of students who are enrolled in all courses offered by the Computer Science department.
33. List all instructors who teach courses with no enrollments.
34. Find the names of students who have never received a grade of 'F.'
35. Retrieve the most recent hire date of instructors in each department.

7. Window Functions

36. Rank students within each department based on their grades.
37. Retrieve the top 3 courses with the highest number of enrollments.
38. Calculate the cumulative number of books borrowed by students, ordered by loan date.
39. List each student along with their rank in terms of the number of courses they are enrolled in.
40. Find the average grade of students in each course and show how each student compares to the average.

8. Updates and Deletes

41. Update the phone numbers of all instructors in the Physics department to a new format.
42. Increase the credits of all courses in the Mathematics department by 1.
43. Delete all students who have not enrolled in any courses.
44. Remove all enrollments for courses that are no longer offered.
45. Update the grades of students who scored 'C' in "Quantum Mechanics" to 'B.'

9. Advanced Joins

46. Retrieve students who are not enrolled in any course but have borrowed books.
47. List books that have never been borrowed by any student.
48. Find courses that are taught by multiple instructors.
49. Retrieve the names of instructors who have taught all the courses in their department.
50. List all students along with the courses they are enrolled in and the books they have borrowed.

Complex Queries

1. Advanced Joins

1. Find the names of students who borrowed books written by authors whose names start with "T."
2. List the names of instructors who teach courses in departments where they are not assigned.
3. Retrieve the names of students and courses for those who have borrowed books and are enrolled in those courses' related departments.
4. Find all students who borrowed books but have not enrolled in any courses.
5. Retrieve courses that have students enrolled but are not taught by any instructor.

2. Subqueries and Correlated Subqueries

6. Retrieve the name of the student who has borrowed the most books.
7. Find the department with the highest number of students enrolled in its courses.
8. List students who are enrolled in all the courses taught by "Dr. Alan Turing."
9. Retrieve the names of students who borrowed books from authors they are also studying in their courses.

10. Find the department(s) offering the maximum number of courses.

3. Aggregations with HAVING

11. Find all courses where more than 5 students are enrolled.
12. Retrieve instructors teaching more than 3 courses.
13. List departments with fewer than 10 students.
14. Find students who borrowed more than 5 books but returned less than 3.
15. List all courses with an average grade greater than 'B.'

4. Nested Joins and Subqueries

16. List all books borrowed by students enrolled in courses with fewer than 5 students.
17. Retrieve the names of students who have received at least one 'A' in all courses they are enrolled in.
18. Find books authored by instructors who belong to the same department as the students who borrowed them.
19. List the names of students who have taken courses in more than one department.
20. Retrieve the names of instructors who teach courses that no students are enrolled in.

5. Window Functions

21. Rank students within their departments by the number of courses they are enrolled in.
22. Calculate the running total of books borrowed by each student.
23. Find the top 3 students with the highest total grades in the entire college.
24. Retrieve the cumulative average grades for each student in their enrolled courses.
25. Show each instructor's rank within their department based on the number of courses they teach.

6. Recursive CTEs

26. Retrieve a hierarchical list of instructors and the departments they belong to, including the Head of Department.
27. Generate a report showing each student, their department, and a count of all other students in the same department.
28. Find all departments that have offered courses directly or indirectly to students who borrowed books.
29. Show a chain of instructors who teach students who have borrowed books written by other instructors.

30. Build a chain showing students borrowing books linked to their course instructors.

7. Complex Filtering

31. Retrieve all students who borrowed books whose titles contain words from their enrolled courses.
32. List all instructors who have taught every course in their department.
33. Find all books that were borrowed during the same month they were added to the library.
34. Retrieve students who have received at least one grade higher than the average grade of their department.
35. List all students who borrowed books and returned them in a shorter duration than the average return time.

8. Advanced Updates

36. Update the grade for students in courses where the average grade is below 'C' to 'B.'
37. Change the department of all students enrolled in courses taught by "Dr. Barbara Liskov" to Computer Science.
38. Mark all overdue book loans (LoanDate + 30 days < CURRENT_DATE) as returned with today's date.
39. Update the number of credits for all courses to 5 for courses with at least 10 students enrolled.
40. Assign a new Head of Department to departments with no students currently enrolled in their courses.

9. Advanced Deletes

41. Delete all students who have not borrowed books and have not enrolled in any courses.
42. Remove all books authored by authors whose books have never been borrowed.
43. Delete all courses that have not been taught in the past 2 years.
44. Remove all instructors who do not teach any courses.
45. Delete all enrollments where the student has received a grade of 'F' and has no other grades.

10. Complex Reports

46. Generate a report of all students with their total number of courses enrolled and total number of books borrowed.

47. Create a department-wise breakdown showing the number of instructors, courses, and students.
48. List each student along with their average grade and the total number of books they have borrowed.
49. Show each instructor's name, the total number of courses they teach, and the average enrollment per course.
50. Generate a comprehensive report of each department, including the total number of students, courses, instructors, and books.