

EE353: SQL Basics

Sravan K Suresh

September 19, 2024

1 Introduction

Structured Query Language (SQL) is a standard programming language used to manage and manipulate relational databases. Here I have covered the basics of SQL focusing on the **SELECT**, **FROM**, **WHERE**, and **JOIN** clauses.

2 SELECT Statement

2.1 Purpose

The **SELECT** statement is used to retrieve data from a database.

2.2 Syntax

```
SELECT column1, column2, ...  
FROM table_name;
```

2.3 Example

```
SELECT first_name, last_name  
FROM employees;
```

3 FROM Clause

3.1 Purpose

The **FROM** clause specifies the table from which to retrieve data.

3.2 Example

```
SELECT *  
FROM customers;
```

4 WHERE Clause

4.1 Purpose

The **WHERE** clause filters records based on specified conditions.

4.2 Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

4.3 Common Operators

- =: Equal
- != or <>: Not equal
- >: Greater than
- <: Less than
- >=: Greater than or equal to
- <=: Less than or equal to
- LIKE: Pattern matching
- IN: Matches any value in a list
- BETWEEN: Within a range

4.4 Example

```
SELECT *  
FROM products  
WHERE price > 50;
```

5 JOIN Clause

5.1 Purpose

The JOIN clause combines rows from two or more tables based on a related column.

5.2 Types of Joins

- **INNER JOIN**: Returns records with matching values in both tables.

```
SELECT a.column1, b.column2  
FROM table1 a  
INNER JOIN table2 b  
ON a.common_column = b.common_column;
```

- **LEFT JOIN (or LEFT OUTER JOIN)**: Returns all records from the left table, and matched records from the right table.

```
SELECT a.column1, b.column2  
FROM table1 a  
LEFT JOIN table2 b  
ON a.common_column = b.common_column;
```

- **RIGHT JOIN (or RIGHT OUTER JOIN)**: Returns all records from the right table, and matched records from the left table.

```
SELECT a.column1, b.column2  
FROM table1 a  
RIGHT JOIN table2 b  
ON a.common_column = b.common_column;
```

- **FULL JOIN (or FULL OUTER JOIN):** Returns all records when there is a match in either table.

```
SELECT a.column1, b.column2
FROM table1 a
FULL OUTER JOIN table2 b
ON a.common_column = b.common_column;
```

5.3 Example Combining Everything

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e
INNER JOIN departments d
ON e.department_id = d.id
WHERE e.salary > 50000;
```

6 Tips for Writing SQL Queries

- Use aliases (e.g., e and d) for better readability.
- Be mindful of case sensitivity; SQL keywords are typically written in uppercase.
- Use comments to annotate your code:

```
-- This query retrieves employee names
-- and their departments with a salary
-- above 50,000
```

7 Practice Exercises

1. Write a query to select all columns from a table named `orders` where the status is 'completed'.
2. Use an `INNER JOIN` to combine data from `customers` and `orders` tables based on `customer_id`.
3. Create a query to find products with prices between 20 and 100.

8 Solutions to Practice Exercises

1. **Solution:** To select all columns from the `orders` table where the status is 'completed':

```
SELECT *
FROM orders
WHERE status = 'completed';
```

2. **Solution:** To combine data from the `customers` and `orders` tables based on `customer_id` using an `INNER JOIN`:

```
SELECT c.*, o.*
FROM customers c
INNER JOIN orders o ON c.customer_id = o.customer_id;
```

3. **Solution:** To find products with prices between 20 and 100:

```
SELECT *
FROM products
WHERE price BETWEEN 20 AND 100;
```