# Section 5: SQL Queries

## Summary

### 5.1 Introduction

Welcome to SkillSprint 5! In this section, we'll explore the core of SQL—writing queries to retrieve data. This is where you'll begin to interact with your database, extracting the information you need using SQL queries. These queries allow you to specify what data you want to retrieve and where to find it in the database.

### 5.2 Introduction to SQL Queries

SQL queries are the fundamental building blocks of interacting with databases. These queries are commands written in SQL (Structured Query Language) that instruct the database management system to perform specific actions, such as retrieving, updating, or deleting data from the database.

In this section, we will dive into the world of SQL queries, focusing on data retrieval. You will learn how to construct queries that specify which columns of data you want to retrieve from a table. SQL queries form the backbone of database interaction, and understanding how to write effective queries is essential for working with databases.

## 5.3 Using the SELECT Clause

The SELECT clause is a pivotal component of SQL queries. It allows you to define precisely which columns of data you want to extract from a table. With the SELECT clause, you have the flexibility to choose specific columns, retrieve all columns, or even perform calculations on the data before retrieval.

In this section, we will explore the intricacies of the SELECT clause. You will gain a deep understanding of how to craft queries that cater to your specific data retrieval needs. Whether you need to fetch basic data or perform advanced calculations, the SELECT clause is your tool of choice.

## 5.4 Understanding the FROM Clause

The FROM clause plays a vital role in SQL queries as it determines the source of the data you want to retrieve. It specifies the table or tables from which you wish to fetch data. Understanding how to navigate your database's structure using the FROM clause is crucial, as it ensures that you are pulling data from the right place.

In this section, we will delve into the FROM clause and learn how to effectively identify and select the tables that contain the data you need. You will gain the skills to navigate complex database structures confidently, ensuring that your queries retrieve the correct information.

## 5.5 Combining SELECT and FROM

To create a complete and functional SQL query, you need to combine the SELECT and FROM clauses. This combination allows you to target specific data within a table and retrieve it based on your defined criteria. Understanding how to integrate these clauses is key to constructing powerful and precise queries.

In this section, we will explore the process of combining SELECT and FROM to create complete queries. You will learn how to specify the columns you want to retrieve and the tables from which to fetch the

data. This skill is essential for crafting queries that deliver the exact information you seek.

## 5.6 Refining SELECT Queries

Once you have mastered the basics of SQL queries, it's essential to refine your skills to become a proficient data retriever. You will learn advanced techniques to filter, sort, and manipulate data using SQL queries. This allows you to tailor your query results to meet specific requirements.

In this section, we will delve into the art of refining SELECT queries. You will explore how to apply conditions, use aliases to rename columns, and perform calculations on the retrieved data. These advanced techniques empower you to extract the precise information you need from your database.

## 5.7 Best Practices in Writing SQL Queries

To become a proficient SQL query writer, it's crucial to follow best practices that ensure your queries are efficient, maintainable, and secure. In this section, we will share valuable guidelines and practices for writing high-quality SQL queries.

You will learn about optimization techniques to enhance query performance, how to write queries that are easy to understand and maintain, and the importance of security when dealing with databases. Following these best practices will elevate your SQL querying skills and make you a proficient database user.

# Code Breakdown

In this section, we will break down the SQL code used to construct various types of queries. Understanding the code behind SQL queries is crucial for becoming proficient in database interaction. Let's explore the code associated with each aspect of SQL querying:

## Basic SELECT Query

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

- `SELECT`: This keyword is used to specify the columns you want to retrieve data from.
- `column1, column2`: Replace these with the actual column names you want to fetch.
- `FROM`: Indicates the source table from which you are retrieving data.
- `table_name`: Replace this with the name of the table you are querying.

## Using WHERE Clause for Filtering

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

- `WHERE`: This clause allows you to apply conditions to filter the retrieved data based on specific criteria.
- `condition`: Replace this with the condition you want to apply. For example, `column1 = 'value'`.
- Sorting Results with `ORDER BY`.

```
SELECT column1, column2
FROM table_name
ORDER BY column1 ASC;
```

- `ORDER BY`: This clause is used to sort the results in ascending (ASC) or descending (DESC) order based on a specific column.

- `column1`: Replace this with the column by which you want to sort the results.

## Using Aliases for Column Names

```
SELECT column1 AS alias1, column2 AS alias2
FROM table_name;
```

- `AS`: Allows you to provide an alias for column names, making the output more readable.
- `alias1, alias2`: Replace these with the desired aliases for the columns.

## Aggregating Data with Functions

```
SELECT COUNT(column1), AVG(column2)
FROM table_name
GROUP BY column3;
```

- `COUNT()`, `AVG()`: These are aggregate functions used to perform calculations on columns.
- `GROUP BY`: Groups the results based on a specific column.
- `column3`: Replace this with the column you want to group by.

# Vocabulary

**SQL Queries:** Commands used to interact with a database, including retrieving, updating, or deleting data.

**SELECT Clause:** Part of an SQL query that specifies which columns of data to retrieve from a table.

**FROM Clause:** Part of an SQL query that specifies the table or tables from which to fetch data.

**Database Retrieval:** The process of extracting data from a database using SQL queries.

# Practice

Now that you've learned the fundamentals of SQL querying using the SELECT and FROM clauses, it's time to put your knowledge into practice. These exercises will help reinforce your understanding and build your confidence in writing SQL queries.

You may download a database to complete these exercises at https://drive.google.com/file/d/1fBY69nkSullfNoaSxebKkVBldetOT4O8/view?usp=sharing.

## Exercise 1: Basic SELECT Queries

Write SQL queries to retrieve specific information from a sample database. Use the following scenarios as a guide:

1. Retrieve the names of all employees from the "employees" table.
2. Get the titles of all books from the "books" table.
3. Fetch the order IDs and order dates from the "orders" table.
4. Find the customer names and their corresponding email addresses from the "customers" table.
5. Select the product names and prices from the "products" table.

## Exercise 2: Filtering with WHERE

Practice filtering data using the WHERE clause. Write SQL queries to achieve the following:

1. Retrieve the names of employees who earn a salary greater than $50,000.
2. Get the titles of books published after the year 2000.
3. Fetch order details (order ID, order date) for orders placed by the customer with the name "John Smith."

4. Find the product names and prices for products with a price less than or equal to $20.
5. Select customer names and phone numbers for customers located in the city of "New York."

## Exercise 3: Sorting with ORDER BY

Practice sorting query results using the ORDER BY clause. Write SQL queries to:

1. Retrieve employee names and salaries, sorted in descending order of salary.
2. Get the book titles and publication years, sorted in ascending order of publication year.
3. Fetch product names and prices, sorted in ascending order of product names.
4. Find customer names and their total order amounts, sorted in descending order of order amounts.
5. Select order IDs and order dates for orders placed by the customer with the name "Mary Johnson," sorted in ascending order of order dates.

## Exercise 4: Using Aliases

Explore the use of aliases for column names. Write SQL queries to:

1. Retrieve the first and last names of employees, but use "First Name" and "Last Name" as aliases for the columns.
2. Get the book titles and their respective authors' names, using "Title" as an alias for book titles and "Author" as an alias for author names.
3. Fetch product names and their prices, using "Product Name" and "Price" as aliases.
4. Find customer names and their email addresses, using "Name" as an alias for customer names and "Email" as an alias for email addresses.
5. Select order IDs and order dates, using "Order ID" and "Order Date" as aliases.

## Exercise 5: Aggregating Data

Practice using aggregate functions. Write SQL queries to:

1. Calculate the total number of employees in the "employees" table.
2. Find the average price of books in the "books" table.
3. Calculate the total order amount for each customer in the "orders" table.
4. Find the highest salary among employees.
5. Calculate the total number of products in the "products" table.

# Exam Q & A

**Q1: What is the primary purpose of the SELECT clause in SQL?**
a) To specify the tables to retrieve data from
b) To filter data based on specific conditions
c) To specify the columns to retrieve from a table
d) To order the query results
**Answer: c) To specify the columns to retrieve from a table.**
*Explanation: The SELECT clause in SQL is used to indicate which columns should be included in the query result. It specifies what data you want to retrieve from a table.*

**Q2: In SQL, which clause is used to specify the source of data for a query?**
a) WHERE
b) FROM
c) SELECT
d) ORDER BY
**Answer: b) FROM.**
*Explanation: The FROM clause in SQL is used to specify the source table or tables from which data should be retrieved in a query.*

**Q3: What does the WHERE clause in SQL allow you to do?**
a) Specify the columns to retrieve
b) Define the order of the query results
c) Filter rows based on specific conditions

d) Perform aggregate calculations

**Answer: c) Filter rows based on specific conditions.**

*Explanation: The WHERE clause in SQL is used to filter rows from the source table based on specified conditions.*

**Q4: Which SQL clause is used to sort query results in a specific order?**

a) SELECT

b) FROM

c) WHERE

d) ORDER BY

**Answer: d) ORDER BY.**

*Explanation: The ORDER BY clause in SQL is used to sort the query results in ascending or descending order based on one or more columns.*

**Q5: When using aliases in SQL, what is their primary purpose?**

a) To change the data type of a column

b) To provide alternate names for columns or tables

c) To filter rows in a query

d) To perform aggregate calculations

**Answer: b) To provide alternate names for columns or tables.**

*Explanation: Aliases in SQL are used to provide alternate names for columns or tables, making the query results more readable and concise.*