**SQL SELECT Statement**

1. Create a database with a table named Books with columns BookID, Title, Author, Genre, and PublicationYear. Populate the table with sample data. Write SQL queries to perform the following tasks:

* Retrieve all columns for books published in the year 2020.
* Find the distinct genres available in the Books table.
* Alias the column Author as BookAuthor in a query result.

1. **-- Create the Books table**

CREATE TABLE Books (

BookID INT PRIMARY KEY,

Title VARCHAR(255),

Author VARCHAR(255),

Genre VARCHAR(50),

PublicationYear INT

);

1. **-- Populate the table with sample data**

INSERT INTO Books (BookID, Title, Author, Genre, PublicationYear)

VALUES

(1, 'Book1', 'Author1', 'Fantasy', 2020),

(2, 'Book2', 'Author2', 'Science Fiction', 2020),

(3, 'Book3', 'Author3', 'Mystery', 2019),

(4, 'Book4', 'Author4', 'Thriller', 2021),

(5, 'Book5', 'Author5', 'Romance', 2020);

1. **-- Retrieve all columns for books published in the year 2020**

SELECT \* FROM Books WHERE PublicationYear = 2020;

1. **-- Find the distinct genres available in the Books table**

SELECT DISTINCT Genre FROM Books;

1. **-- Alias the column Author as BookAuthor in a query result**

SELECT BookID, Title, Author AS BookAuthor, Genre, PublicationYear FROM Books;

**SQL ORDER BY**

Design a database query that demonstrates the use of the SQL ORDER BY clause. Apply the concept to a relevant dataset and create a meaningful sorting scenario based on the data.

1. Choose a dataset or create a sample dataset with at least three columns.
2. Write a SQL query that utilizes the ORDER BY clause to sort the data in a specific way.
3. Document the sorting logic and the purpose of the sorting in your query.
4. Include comments in your SQL code to explain each step of the query.
5. Test your query to ensure it produces the desired result.

**a-- Selecting employee information and sorting based on salary in descending order**

SELECT EmployeeID, FirstName, Salary

FROM Employees

ORDER BY Salary DESC;

SELECT EmployeeID, FirstName, Salary: This part of the query selects the columns we want to retrieve from the table Employees, namely EmployeeID, FirstName, and Salary.

FROM Employees: Specifies the table in this case, the Employees table.

ORDER BY Salary DESC: This part of the query sorts the result set based on the Salary column in descending order (DESC), meaning the highest salaries will appear first.

By sorting the data in descending order of salary, we can easily identify the employees who earn the highest salaries.