SQL Operators: Bookstore Inventory database

We will first of all create a bookstore table and fill it with dummy data and utilize that to learn about different types of operators.

Creating Tables & Inserting required data into it:

```
CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title VARCHAR(100),
    Author VARCHAR(100),
    Genre VARCHAR(50),
    Price DECIMAL(5, 2),
    Quantity INT,
    PublicationYear INT
);
```

- `CREATE TABLE Books` starts the command to create a new table named 'Books'.
- Inside the parentheses, we define the columns and their data types:
    - `BookID INT PRIMARY KEY`: An integer column for the book ID, set as the primary key.
    - `Title VARCHAR(100)`: A variable character string column for the book title, with a maximum length of 100 characters.
    - `Author VARCHAR(100)`: A column for the author's name, with a maximum length of 100 characters.
    - `Genre VARCHAR(50)`: A string column for the genre, with a maximum length of 50 characters.
    - `Price DECIMAL(5, 2)`: A decimal column for the price, allowing up to 5 digits with 2 decimal places.
    - `Quantity INT`: An integer column for the quantity of books in stock.
    - `PublicationYear INT`: An integer column for the publication year.

Inserting data into books table:

```
INSERT INTO Books (BookID, Title, Author, Genre, Price, Quantity,
PublicationYear) VALUES

(1, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction', 18.00, 5,
1925),

(2, 'To Kill a Mockingbird', 'Harper Lee', 'Fiction', 15.00, 7, 1960),

(3, '1984', 'George Orwell', 'Sci-Fi', 12.00, 10, 1949),
```

```
(4, 'The Catcher in the Rye', 'J.D. Salinger', 'Fiction', 20.00, 3,
1951),

(5, 'Pride and Prejudice', 'Jane Austen', 'Romance', 14.00, 6, 1813);
```

Operators

1. Arithmetic Operators

- Arithmetic operators are used for performing mathematical operations.

Example (Addition '+'):

```
SELECT Title, Price + 5 AS IncreasedPrice FROM Books;
```

- This query increases the price of each book by 5 and displays the new price alongside the title.

Example (Multiplication '*'):

```
SELECT Title, Price * Quantity AS StockValue FROM Books;
```

- This calculates the total stock value of each book (price multiplied by quantity).

2. Wildcard Operators

- Wildcard operators are used with the LIKE operator for pattern matching in strings.

Example (Percent '%'):

```
SELECT * FROM Books WHERE Title LIKE '%and%';
```

- Finds all books with 'and' in their title, such as "Pride and Prejudice".

Example (Underscore '_'):

```
SELECT * FROM Books WHERE Title LIKE 'The _reat Gatsby';
```

- Matches titles like "The Great Gatsby" (where the underscore represents any single character).

3. Logical Operators

- Logical operators are used to combine conditions.

Example (AND):

```
SELECT * FROM Books WHERE Genre = 'Fiction' AND Price < 20;
```

Selects all fiction books priced below $20.

Example (OR):

```
SELECT * FROM Books WHERE Genre = 'Fiction' OR Genre = 'Sci-Fi';
```

- Retrieves all books that are either in the 'Fiction' or 'Sci-Fi' genre.

Example (NOT):

```
SELECT * FROM Books WHERE NOT Genre = 'Fiction';
```

- Finds all books that are not in the 'Fiction' genre.

4. BETWEEN & IN Operators

- These operators are used for range checking and matching a list of values, respectively.

Example (BETWEEN):

```
SELECT * FROM Books WHERE PublicationYear BETWEEN 1900 AND 2000;
```

- Selects books published between the years 1900 and 2000.

Example (IN):

```
SELECT * FROM Books WHERE Genre IN ('Fiction', 'Romance');
```

- Retrieves all books that are in either the 'Fiction' or 'Romance' genres.