SQL Clauses and Operators

I. SQL Clauses

A. **DISTINCT Clause**
   - Used to return only distinct (unique) values in the result set
   - Eliminates duplicate rows from the query result
   - Syntax: **SELECT DISTINCT column1, column2, ... FROM table_name;**
   - Example:
     SELECT DISTINCT department FROM employees;

B. **AS Clause (Alias)**
   - Used to give a table or a column a temporary name
   - Makes column names more readable
   - Syntax:
   - **SELECT column_name AS alias_name FROM table_name;**
   - **SELECT column_name FROM table_name AS alias_name;**
   - Example:
     SELECT first_name AS fname, last_name AS lname FROM employees;
     SELECT e.first_name, e.last_name FROM employees AS e;

C. **WHERE Clause**
   - Used to filter records based on a specified condition
   - Syntax: **SELECT column1, column2, ... FROM table_name WHERE condition;**
   - Example:
     SELECT * FROM employees WHERE department = 'Sales';

II. Operators

A. Arithmetic Operators
   - Used to perform mathematical operations
   - Operators:
   - + (Addition)
   - - (Subtraction)
   - * (Multiplication)
   - / (Division)
   - % (Modulus)

B. Comparison Operators
- Used to compare two values
- Operators:
- = (Equal to)
- <> or != (Not equal to)
- > (Greater than)
- < (Less than)
- >= (Greater than or equal to)
- <= (Less than or equal to)

C. Logical Operators
- Used to combine multiple conditions
- Operators:
- AND (True if all conditions are true)
- OR (True if any condition is true)
- NOT (Reverses the result of the condition)

III. Combining Clauses and Operators

- Clauses and operators can be combined to create complex queries
- Example:

```
SELECT DISTINCT e.department, AVG(e.salary) AS avg_salary
FROM employees AS e
WHERE e.hire_date >= '2020-01-01'
AND e.department IN ('Sales', 'Marketing')
GROUP BY e.department
HAVING AVG(e.salary) > 50000;
```

This query:
1. Uses DISTINCT to get unique departments
2. Uses AS to alias the average salary column
3. Uses WHERE with comparison and logical operators to filter employees
4. Uses special operators (>= for date comparison, IN for multiple values)