#### Practical no. 4

Aim: Write and execute SQL queries- Operators (and, or, not, like, between, in)

SQL Logical Operators are essential tools used to test the truth of conditions in SQL queries. They return boolean values such as TRUE, FALSE, or NULL, making them invaluable for filtering, retrieving, or manipulating data. These operators allow developers to build complex queries by combining, negating, or comparing conditions effectively.

## SQL Between Operator

The SCCL BETWEEN operator is used to test whether a value falls within a given range of values (inclusive). The values can be text, date, or numbers. It can be used in a SELECT, INSERT, UPDATE or DELETE statement. The SCCL BETWEEN Condition will return the records where the expression is within the range of value1 and vafue2.

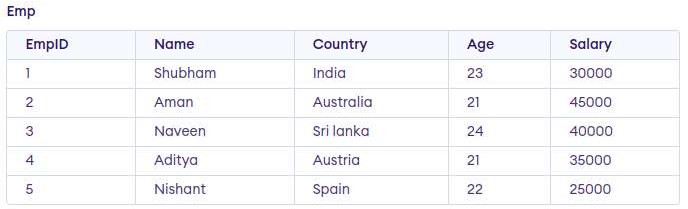
Syntax

*SELECT column name(s) FROM* tabte name

*WHERE* column name *BETWEEN value1 AND value2;*

Key Features:

* Inclusive of both boundary values (va fuel and va1ue2).
* Simplifies queries when working with continuous ranges.



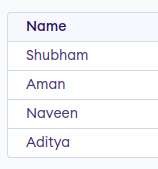
C}uery:

SELECT Name FROM Emp

WHERE Salary

BETWEEN 30000 AND 45000;

Output



# S@L NOT Operator

The SQL NOT Operator is a Logical operator used to negate or reverse the result of a condition in SQL queries. It is commonly used with the WHERE clause to filter records that do not meet a specified condition, helping you exclude certain values from your results.

Syntax:

SELECT column1, colomn2, .:

FROM table name WHERE NOT condition;

Customer ID Customer Name City Posta£Code Country

1 Joh n Wick New York 1248

Around the Horn London WA1 1D.

Ro ha n New Delhi 100084

Example 1: Using SQL NOT to Exclude a Specific VaLue

USA

UK

India

The fofLowinp SOL statement selects all fields from Customers taole where the country is nat UK.

Query:

5ELECT ”

F RON Cu sI omer's

WHERE NOT Country = 'UK';

Output:

Customer ID Customer Name City Posta fCode Country

1 Joh n Wick

3 Rohan

how York 1248

New Delhi 100084

USA

India

Example 2: Using SQL NOT with IN Operator

The NOT aperator ca n also be used with the IN condition to exclude multip ie values fro m the res ult set. C,}uery:

S ELECT ”

F ROu Cu st ofrer s

NHE RE NOT Countny I M ( ' USA ' , ' UK ' ) ;

Output:

Customer ID Customer Name City Posta\Code Country

3 Ro han Ncw Dclhi 100084 India

Example 3: Using SQL NOT with LIKE Operator

We can also combine NOT with the LIKE operator to exclude records that match a certain pattern.

Query:



CustomerlD Customer Name City PostatCode Country

1. Jahn Wick New York 1 24 8
2. Around the Horn London WA1 1DP

USA

UK

Example 4: Using SQL NOT with NULL Values

To exclude records where a column lfas a NU LL va due, combine NOT with the IS NULL cond itioi Query:

SELECT \*

PROM Customers

WHERE NDT Postal[ode I8 NULL;

Output:

CustomerlD CustomerName City PostalCode Country

Jahn Wick N ew Yo rk

12J8

USA

A round the Horn Londo n WA 1 10 P

1. Ro hon N ew Defhi 100081

UK

Indin

Th is query excfuues customers wh a leave a NU LL va lue for PostalCoue.

Example 5: Using NOT with AND Operator

We can cots bine NOT with the AND operator to create more campfex conclition s. T his qu ery retrie'ves cu stom ers wh a are n ot from the USA and are also not from the UK.

Query:



Output:

CustomerlD Customer Name City PostafCode Country

3 Ro nan Ncvv Dcfhi 100084 Ind ia

# Key TakeAways About NOT Operator

* NOT operator *returns* opposite results or *negative* results. /t negates the t›ootean condition *in* the *WHERE* clause.
* /t is *used* to *exclude* specffiC data *from* the result set.

##### Using the NOT Operator with BETWEEN

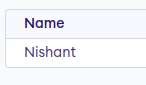
C}uery:

SELECT Name FROM Emp

WHERE Salary

NOT BETWEEN 30000 AND 45000;

Output



## SQL IN Operator

IN operator allows us to easily test if the expression matches any value in the fist of values. It is used to remove the need for multiple OR conditions in SELECT, INSERT, UPDATE, or DELETE. We can also use NOT IN to exclude

the rows in our fist. We should note that any kind of duplicate entry will be retained.

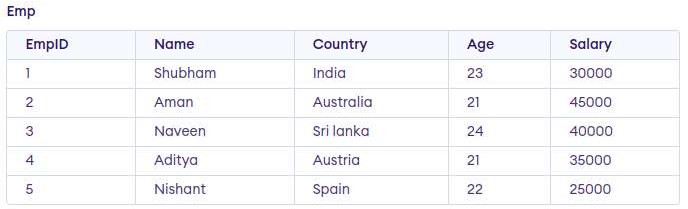
Syntax

*SELECT column name(s) FROM* table name

*WHERE column name* IN {list of vatues/;

Key Features:

* Ideal for filtering non-sequential values.
* Handles duplicates in the list of values.



##### Example 1: Using IN Operator

Query:

SELECT Name FROM Emp

WHERE Salary IN (30000, 40000, 25000):

Output

Name Shuühom Noveen Nishont

Example 2: Using the NOT Operator with IN

Query:

SELECT Name

FROM Emp

WHERE Salary NOT IN (25000, 30000);

Output

Nome AmOr. Nnveen Aditya

### S§L AND Operator

The AND operator allows you to fitter data based on multiple conditions, all of which must be true for the record to be included in the result set.

Syntax:

The syntax to use the AND operator in SQL is:

*!SKk ICT " FROM* tabte\_name *WI-IERE* conditionl *AND* condition2 *AND*

...conditionN;

Here,

* table name: name of the table
* condition1,2,..N: first condition, second condition, and so on.

### S§L OR Operator

The OR Operator in S€}L displays the records where any one condition is true, i.e. either condition 1 or condition2 is True.

Syntax:

The syntax to use the OR operator in SQL is:

*SELECT FROM table name WHERE condition 1 OR* condition2 *OR...*

conditions;

* table name: name of the table
* condition1,2,..N: first condition, second condition, and so on

### SQL AND and OR Operator Examp\es

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ROLL | NO | NAME | ADDREsS | PHONE | Age |
| 1 |  | Ram  RAMESH | Delhi  GURGADN |  | 1B  1B |
|  |  | SUJIT | ROHTAK |  | 20 |
|  |  | SURESH | Delhi |  | 16 |
|  |  | SUJIT | ROHTAK |  | 20 |
|  |  | RAMESH | GURGADN |  | 18 |

Example 1: SQL AND Operator

If su ppose we want to fetch all the records from the Student tab Ie where Ag e is 18 and ADDRESS is Delhi.

Query:

SE LECT \* FROPt S-Lu dens

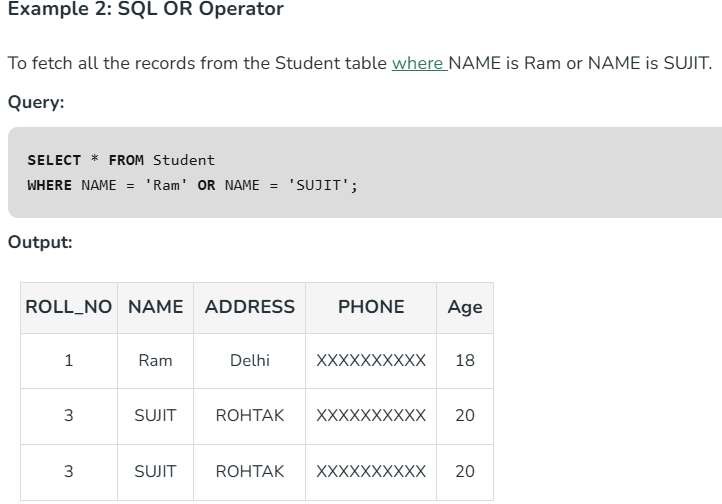
WHERE Age = 18 AND ADDRESS = 'Delhi' ;

Output:

ROLL NO NAME ADDRESS PHONE Ag e

1 Ram Dclhi XXXXXXXXXX 18

4 SURESH Dclhi XXXX XX”X” XX”X”\*” ” 1' 8”



## Combining AND and OR Operators in SQL

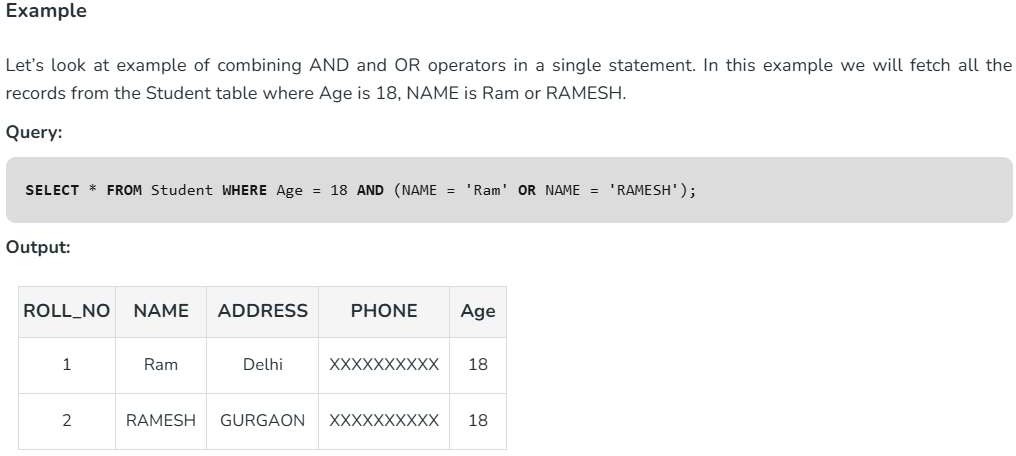
Combining AND and OR Operators in SCCL allows the creation of complex conditions in queries. This helps in filtering data on multiple conditions.

Syntax:

Syntax to use AND and OR operator in one statement in SQL is:

*SELECT\* FROM* tabte *name*

*WHERE* conditionl *AND* {condition2 *OR* condition3};



*Important Points About* S§1 *AND and OR* Operators

* *The* SQL AND *operator is used* to *combine* multiple *conditions, where all*

the *conditions* must *be* true *for the* row to *be included in* the resutt set.

* *The OR* operator *is used* to comtiine *multiple conditions,* where at least *one of* the conditions must be true *for the row* to *be included in* the resutt set.
* *Any kind of condition, including* equality, *inequality, comparison, and*

*logical* operators, can die utftfZ£?d with the *AND and OR* operators.

* *The AND* operator *is more important than* the 0/ *operator. In* other *words,* when *both* are *used in* the *same* SQL statement, the *AND operator will be executed* first. *To change* the *order of* evaluation, *parentheses* can due *used.*
* *You can employ* the *AND and OR* operators *inside of* other conditions

t›ecause they can *both* die nested.

*SQL LIKE* Operafor

The SQL LIKE operator is used for performing pattern-based searches in a database. It is used in combination with the WHERE clause to filter records based on specified patterns, making it essential for any database-driven application that requires flexible search functionality. LIKE operator is case-insensitive by default in most database systems. This means that if you search for "apple" using the LIKE operator, it will return results that include "Apple", "APPLE", "aPpLe", and so on.

Syntax:

*SELECT column1, column2, ... FROM* table name

###### WHERE column name LINE pattern;

* + column name: The column to be searched.
  + pattern: The pattern to search for, which can include wildcard characters.

For making the LIKE operator case-sensitive, you can use the "BINARY" keyword in MySC}L or the "COLLATE" keyword in other database systems.

For example:

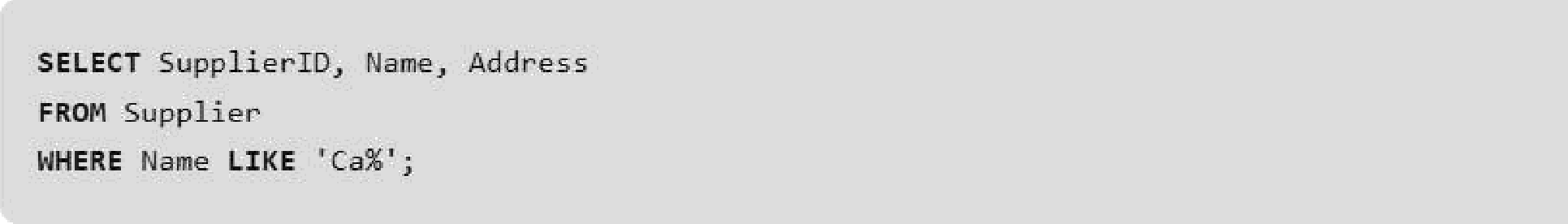
SELECT \* FROM products WHERE name LIKE BINARY 'apple%'

This following query will only return products whose name starts with “apple” and is spelled exactly like that, without capital letters.

|  |  |  |
| --- | --- | --- |
| !2uppLierJO | Name | Address |
| S1 | Paragon Suppliers | 21-3, Okhla, Deli |
| S2 | Mango Nation | 21, Faridabad, Haryana |
| 53 | Canadian Biz | 6/7, Okhfa Phase II, Delhi |
| S4 | Caravan Traders | 2—A, Pitampura, Delhi |
| S5 | Harish and Sons | Gurgaon. NCR |
| 56 | Om Suppliers | 2/1, Faridabad, Haryana |

Example 1 : Match Names Sta rting with ‘Ca’

Retrieve Supplier1D. Name, and Audress from suppliers table, where supplier name starts form k.



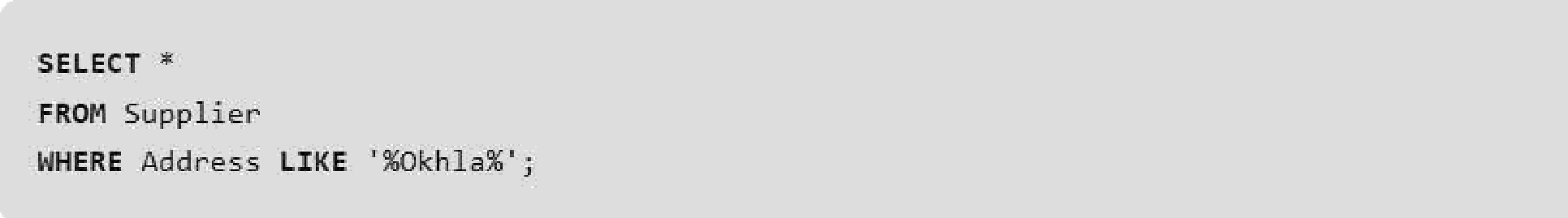
Output:

SP Can a dian Biz 6/7, Okhfa Phase II, Delhi

SP Car avan Trade rs 2 -A, Pitam pura, De lhi

Example 2: Match Addresses Containing ‘Okh \a’

Retrieve entire table, where address contains OKHLA.



Output:

S 1 Parag an Su ppliers 2 1-3. Okhla, Delhi

S 3 Can adian Biz 6/7, Ok hfa Phase II, Delhi

Example 3: Match Names Where ‘ango’ Appears in the Second Position

Retrieve Supplier1D, Name and Address of supp lier whose name conta ins “ango" in second sribstring.

SELECT SupplierID, Name, Address FQOM Supplier



Output:

S2 M ango Natio n 21, Farid aba d, IN aryan a

Example 4: Using LIKE with AND for Complex Conditions

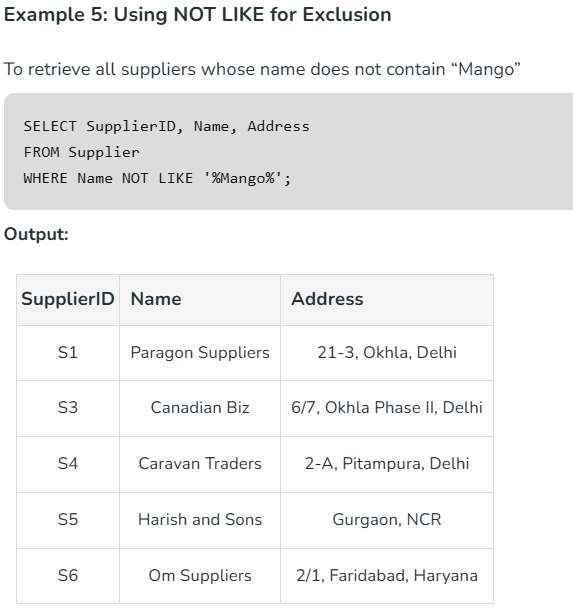
Retrieve suppliers from Defhi with names starting with "C“:



WHERE Address LIKE '%Delhi%' AND Name LIKE 'O%';

Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SupplierlD |  | Name |  | Address |
| S3 |  | Canadian Biz |  | 6/7, Okhla Ph a se II. Delhi |
| S4 |  | Ca rev an Traders |  | 2-A. Pitamp urea. Delhi |



# S§L LIKE Application

The LIKE operator is extremely resourceful in situations such as address filtering wherein we know only a segment or a portion of the entire address (such as locality or city) and would like to retrieve results based on that. The wildcards can be resourcefully exploited to yield even better and more filtered tuples based on the requirement.

Key Takeaways About LIKE Operator

* *LIKE* operator is used to search for specific patterns in a cotomn.
* /t is mostty *used* with *WHERE clause for ñnding or Altering* specific

#### data.

* *Like Operator is* case-insensitive *by default,* to *make* it case

###### sensitive, we can use BINARY keyword.

* *LIKE operator has 4 wild cards, which we* can *use* with *LIKE* operator to *specify* the fitter. *The wild cards* are: *%, ,[) and -.*

Queries for Practice

-- Customer Table

CREATE TABLE Customer (

customer id NUMBER PRIMARY KEY, name VARCHAR2(100),

email VARCHAR2(100) UNIQUE, phone VARCHAR2(15),

address VARCHAR2(255)

) •

-- Product Table

##### CREATE TABLE Product (

product id NUMBER PRIMARY KEY, name VARCHAR2(100),

category VARCHAR2(50),

price NUMBER(10,2),

stock quantity NUMBER

)•

-- Orders Table

CREATE TABLE Order Details (

order id NUMBER PRIMARY KEY,

customer id NUMBER,

order date DATE,

total amount NUMBER(10,2),

FOREIGN KEY (customer id) REFERENCES Customer(customer id)

) '

-- Order Items Table CREATE TABLE Order ltem (

order id NUMBER, product id NUMBER, quantity NUMBER, subtotal NUMBER(10,2),

PRIMARY KEY (order id, product id),

FOREIGN KEY (order id) REFERENCES Order Details(order id), FOREIGN KEY (product id) REFERENCES Product(product id)

) •

-- Employee Table

CREATE TABLE Employee 1 (

employee id NUMBER PRIMARY KEY, name VARCHAR2(100),

role VARCHAR2(50),

salary NUMBER(10,2),

hire date DATE

)•

-- Insert Customers

INSERT INTO Customer (customer id,name, email, phone, address) VALUES (1,'Alice Johnson’, ['alice@gmaiL.com',](mailto:%27alice@gmaiL.com) '9876543210', 'New York’);

INSERT INTO Customer (customer id,name, email, phone, address) VALUES (2, 'Bob Smith', ['bob@yahoo.com',](mailto:%27bob@yahoo.com) '9123456789', 'Los Angeles’);

INSERT INTO Customer (customer id,name, email, phone, address) VALUES (3, 'Charlie Brown', ['charlie@outlook.com',](mailto:%27charlie@outlook.com) '9998887776', 'Chicago’); INSERT INTO Customer (customer id,name, email, phone, address) VALUES (4, 'David Miller', ['david@gmail.com',](mailto:%27david@gmail.com) '8765432109', 'Miami');

INSERT INTO Customer (customer id,name, email, phone, address) VALUES (5, 'Emily Davis', ['emily@hotmaif.com',](mailto:%27emily@hotmaif.com) '7654321098', 'New York’);

-- Insert Products

INSERT INTO Product ( product id, name, category, price, stock quantity)

VALUES (1, 'Mifk', 'Dairy', 2.50, 50);

INSERT INTO Product (product id, name, category, price, stock quantity)

VALUES

(2, 'Bread', 'Bakery', 1.80, 30);

INSERT INTO Product (product id, name, category, price, stock quantity) VALUES

(3, 'Eggs', 'Dairy', 3.20, 40);

INSERT INTO Product (product id, name, category, price, stock quantity)

VALUES

(4, 'Chicken', 'Meat', 7.50, 20);

INSERT INTO Product (product id, name, category, price, stock quantity) VALUES

(5, 'Apples', 'Fruit', 1.20, 60);

INSERT INTO Product (product id, name, category, price, stock quantity)

VALUES

(6, 'Orange Juice’, ’Beverage', 3.50, 25);

-- Insert Orders

INSERT INTO Order Details (order id, customer id, order date, total amount) VALUES (1, 1, TO DATE(’2024-01-10’, ’YYYY-MM-DD'), 10.50);

INSERT INTO Order Details (order id, customer id, order date, total amount) VALUES (2, 2, TO\_DATE(’2024-01-12’, ’YYYY-MM-DD'), 15.20);

INSERT INTO Order Details (order id, customer id, order date, total amount) VALUES (3, 3, TO\_DATE(’2024-02-01’, ’YYYY-MM-DD'), 20.80);

INSERT INTO Order Details (order id, customer id, order date, total amount) VALUES (4, 4, TO\_DATE(’2024-02-05’, ’YYYY-MM-DD'), 30.00);

INSERT INTO Order Details (order id, customer id, order date, total amount) VALUES (5, 5, TO\_DATE(’2024-02-10’, ’YYYY-MM-DD'), 25.50);

-- Insert Employees

INSERT INTO Employee 1 ( employee id, name, role, salary, hire date)

VALUES

(1, ’Michael Scott', ’Manager’, 75000.00, TO\_DATE(’2020-05-10’,

'YYYY-MM-DD’));

INSERT INTO Employee 1 ( employee id, name, role, salary, hire date)

VALUES

(2, 'Jim Halpert’, ’Cashier’, 30000.00, TO\_DATE(’2021-08-15’, 'YYYY-MM—DD’));

INSERT INTO Employee 1 ( employee id, name, role, salary, hire date)

VALUES

(3, 'Pam Beesly', 'Sales Associate', 28000.00, TO DATE(’2022-02-20’, 'YYYY-MM-DD’));

INSERT INTO Employee 1 ( employee id, name, role, salary, hire date)

VALUES

(4, ’Dwight Schrute', 'Supervisor', 50000.00, TO\_DATE(’2019—11—30',

'YYYY-MM-DD'));

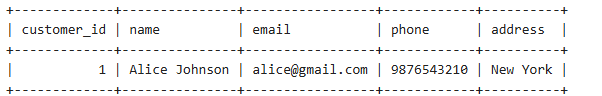
INSERT INTO Employee 1 ( employee id, name, role, salary, hire date)

VALUES

(5, ’Kevin Malone', 'Cashier', 29000.00, TO DATE(’2023-03-10’,

’YYYY-MM-DD'));

LAND Operator

SELECT \* FROM Customer

WHERE address = 'New York' AND email LIKE '%Ogmail.com' ;

SELECT \* FROM Product

WHERE category = 'Dairy' AND stock quantity 20;

2 OR Operator

SELECT FROM Employee

WHERE role = 'Manager' 0R role = 'Superv1sor';

SELECT \* FROM Order Details

WHERE order date = TO DATE('2024-01-10', 'YYYY-MM-DD') OR

order date = TO DATE('2024-02-05', 'YYYY-MM-DD') ;

B NOT Operator

SELECT \* FROM Custome r

WHERE add ress NOT LIKE ' 0âNew Yo rk% ' ;

SELECT \* FROM Employee

WHERE role NOT IN ('Cashier');

BLIKE Operator

SELECT \* FROM Customer

WHERE name LIKE ' A% ' ;

SELECT \* FROM Customer

WHERE email LIKE '%hotmail%' ;

B BETWEEN Operator

SELECT FROM Product

WHERE pr1ce BETWEEN 2 AND 5;

SELECT \* FROM Employee

WHERE hire date BETWEEN TO DATE('2021-07-01', 'YYYY-MM-DD') AND T0„DATE('2023-12-31 ', 'YYYY-MM-DD') ;

SELECT \* FROM Order Details

WHERE order„date BETWEEN TO DATE('2024-02-01',

'YYYY-MM-DD') AND TO DATE('2024-02-28', 'YYYY-MM-DD') ;

lNOperatoF

SELECT \* FROM Customer

WHERE address IN ('New York', 'Los Angeles', 'Miami');

SELECT \* FROM Product

WHERE catego ry IN ( ' Da:I ry ' , Bake ry ' ) ;

SELECT \* FROM Employee

fiHERE role IN ( Cash1e r ' , ' Sales Assoc1ate ' ) ;

