NAMA : KEREN SANDRA SUDARTA
NIM   : 2024071031

BASIS DATA P12

Quiz Knowledge (File 6)
1. b. False
2. Avoid using leading wildcards in predicates, Avoid using unnecessary columns in the SELECT clause, Use INNER JOIN instead of OUTER JOIN wherever possible
3. A Primary Key is also known as a Clustered index, A Primary Index is created automatically when a table is created
4. FullName
5. EXPLAIN

Lab 1 (File 7)

```
--  Optimizing SELECT statements in MySQL
-- 1: Create the database:
CREATE DATABASE Lucky_Shrub;

-- 2: Use the database:
USE Lucky_Shrub;

-- 3: Create the Orders table:
CREATE TABLE Orders(OrderID INT NOT NULL, ClientID VARCHAR(10) DEFAULT NULL,
ProductID VARCHAR(10) DEFAULT NULL, Quantity INT DEFAULT NULL, Cost
DECIMAL(6,2) DEFAULT NULL,
Date DATE DEFAULT NULL, PRIMARY KEY (OrderID));

-- 4: Create the Employees table:
CREATE TABLE Employees(EmployeeID INT DEFAULT NULL, FullName VARCHAR(100)
DEFAULT NULL,
Role VARCHAR(50) DEFAULT NULL, Department VARCHAR(255) DEFAULT NULL);

-- 5: Insert data into the Orders table:
INSERT INTO Orders (OrderID, ClientID, ProductID , Quantity, Cost, Date)
VALUES
(1, "Cl1", "P1", 10, 500, "2020-09-01"),
(2, "Cl2", "P2", 5, 100, "2020-09-05"),
(3, "Cl3", "P3", 20, 800, "2020-09-03"),
(4, "Cl4", "P4", 15, 150, "2020-09-07"),
(5, "Cl3", "P3", 10, 450, "2020-09-08"),
(6, "Cl2", "P2", 5, 800, "2020-09-09"),
(7, "Cl1", "P4", 22, 1200, "2020-09-10"),
(8, "Cl3", "P1", 15, 150, "2020-09-10"),
(9, "Cl1", "P1", 10, 500, "2020-09-12"),
```

```
(10, "Cl2", "P2", 5, 100, "2020-09-13"),
(11, "Cl1", "P2", 15, 80, "2020-09-12"),
(12, "Cl1", "P1", 10, 500, "2022-09-01"),
(13, "Cl2", "P2", 5, 100, "2022-09-05"),
(14, "Cl3", "P3", 20, 800, "2022-09-03"),
(15, "Cl4", "P4", 15, 150, "2022-09-07"),
(16, "Cl3", "P3", 10, 450, "2022-09-08"),
(17, "Cl2", "P2", 5, 800, "2022-09-09"),
(18, "Cl1", "P4", 22, 1200, "2022-09-10"),
(19, "Cl3", "P1", 15, 150, "2022-09-10"),
(20, "Cl1", "P1", 10, 500, "2022-09-12"),
(21, "Cl2", "P2", 5, 100, "2022-09-13"),
(22, "Cl2", "P1", 10, 500, "2021-09-01"),
(23, "Cl2", "P2", 5, 100, "2021-09-05"),
(24, "Cl3", "P3", 20, 800, "2021-09-03"),
(25, "Cl4", "P4", 15, 150, "2021-09-07"),
(26, "Cl1", "P3", 10, 450, "2021-09-08"),
(27, "Cl2", "P1", 20, 1000, "2022-09-01"),
(28, "Cl2", "P2", 10, 200, "2022-09-05"),
(29, "Cl3", "P3", 20, 800, "2021-09-03");

-- 5: Insert data into the Employees table:
INSERT INTO Employees (EmployeeID, FullName,  Role, Department)
VALUES
(1, "Seamus Hogan", "Manager", "Management"),
(2, "Thomas Eriksson", "Assistant ", "Sales"),
(3, "Simon Tolo", "Executive", "Management"),
(4, "Francesca Soffia", "Assistant  ", "Human Resources"),
(5, "Emily Sierra", "Accountant", "Finance"),
(6, "Greta Galkina", "Accountant", "Finance"),
(7, "Maria Carter", "Executive", "Human Resources"),
(8, "Rick Griffin", "Manager", "Marketing");

-- Task 1
-- Lucky Shrub need data on client orders. They have written the following SELECT query to retrieve
-- all data from the Orders table:

SELECT * FROM Orders;

-- Task 2
-- Lucky Shrub need to find the order placed by the client Cl1.
-- They have written the following query to complete this task:

SELECT * FROM Orders WHERE ClientID ='Cl1';

-- However, this query's execution plan shows that it does not use an index to perform this search,
-- as indicated by the NULL values in possible_keys and keys columns.
```

```
-- Task 3
-- Lucky Shrub have written the following SELECT query to find the details of the employee whose
-- last name is 'Tolo':

SELECT * FROM Employees WHERE FullName LIKE '%Tolo';
```

Lab 2 (File 8)

```
-- C4M2L2 Item 8 Lab: MySQL optimization techniques exercise

-- 1.Create the database:
CREATE DATABASE IF NOT EXISTS Lucky;

-- 2.Use database:
USE Lucky;

-- 3.Create the tables:
CREATE TABLE  Orders (OrderID INT NOT NULL PRIMARY KEY, ClientID VARCHAR(10),
ProductID VARCHAR(10), Quantity INT, Cost DECIMAL(6,2), Date DATE);

CREATE TABLE Products (
    ProductID VARCHAR(10),
    ProductName VARCHAR(100),
    BuyPrice DECIMAL(6,2),
    SellPrice DECIMAL(6,2),
    NumberOfItems INT
);

CREATE TABLE Activity (
    ActivityID INT PRIMARY KEY,
    Properties JSON
);


-- 4.Populate the tables with data:
INSERT INTO Orders (OrderID, ClientID, ProductID , Quantity, Cost, Date) VALUES
(1, "Cl1", "P1", 10, 500, "2020-09-01" ),
(2, "Cl2", "P2", 5, 100, "2020-09-05"),
(3, "Cl3", "P3", 20, 800, "2020-09-03"),
(4, "Cl4", "P4", 15, 150, "2020-09-07"),
(5, "Cl3", "P3", 10, 450, "2020-09-08"),
(6, "Cl2", "P2", 5, 800, "2020-09-09"),
(7, "Cl1", "P4", 22, 1200, "2020-09-10"),
(8, "Cl3", "P1", 15, 150, "2020-09-10"),
(9, "Cl1", "P1", 10, 500, "2020-09-12"),
(10, "Cl2", "P2", 5, 100, "2020-09-13"),
```

```
(11, "Cl4", "P5", 5, 100, "2020-09-15"),
(12, "Cl1", "P1", 10, 500, "2022-09-01"),
(13, "Cl2", "P2", 5, 100, "2022-09-05"),
(14, "Cl3", "P3", 20, 800, "2022-09-03"),
(15, "Cl4", "P4", 15, 150, "2022-09-07"),
(16, "Cl3", "P3", 10, 450, "2022-09-08"),
(17, "Cl2", "P2", 5, 800, "2022-09-09"),
(18, "Cl1", "P4", 22, 1200, "2022-09-10"),
(19, "Cl3", "P1", 15, 150, "2022-09-10"),
(20, "Cl1", "P1", 10, 500, "2022-09-12"),
(21, "Cl2", "P2", 5, 100, "2022-09-13"),
(22, "Cl2", "P1", 10, 500, "2021-09-01"),
(23, "Cl2", "P2", 5, 100, "2021-09-05"),
(24, "Cl3", "P3", 20, 800, "2021-09-03"),
(25, "Cl4", "P4", 15, 150, "2021-09-07"),
(26, "Cl1", "P3", 10, 450, "2021-09-08"),
(27, "Cl2", "P1", 20, 1000, "2022-09-01"),
(28, "Cl2", "P2", 10, 200, "2022-09-05"),
(29, "Cl3", "P3", 20, 800, "2021-09-03"),
(30, "Cl1", "P1", 10, 500, "2022-09-01");

INSERT INTO Products (ProductID, ProductName, BuyPrice, SellPrice, NumberOfItems)
VALUES
("P1", "Artificial grass bags ", 40, 50, 100),
("P2", "Wood panels", 15, 20, 250),
("P3", "Patio slates",  35, 40, 60),
("P4", "Sycamore trees ", 7, 10, 50),
("P5", "Trees and Shrubs", 35, 50, 75),
("P6", "Water fountain", 65, 80, 15);

INSERT INTO Activity(ActivityID, Properties) VALUES
(1, '{ "ClientID": "Cl1", "ProductID": "P1", "Order": "True" }' ),
(2, '{ "ClientID": "Cl2", "ProductID": "P4", "Order": "False" }' ),
(3, '{ "ClientID": "Cl5", "ProductID": "P5", "Order": "True" }' );

-- Task 1
-- Lucky Shrub need to find out how many orders were placed by clients with the following
Client IDs in 2022; Cl1, Cl2 and Cl3. They have created the following query to extract this
information.

SELECT
    CONCAT(ClientID, ": ", COUNT(OrderID), " orders") AS Result
FROM Orders
WHERE YEAR(Date) = 2022
  AND ClientID IN ("Cl1", "Cl2", "Cl3")
GROUP BY ClientID;


-- Task 2
PREPARE GetOrderDetail FROM
```

```
'SELECT OrderID, Quantity, OrderCost, Date
 FROM Orders
 WHERE ClientID = ?
 AND YEAR(Date) = ?';
SET @Client = 'Cl1';
SET @Yr = 2020;

EXECUTE GetOrderDetail USING @Client, @Yr;

-- Task 3
SELECT
   p.ProductID,
   p.ProductName,
   p.BuyPrice,
   p.SellPrice
FROM Activity a
JOIN Products p
   ON p.ProductID = a.Properties ->> '$.ProductID'
WHERE a.Properties ->> '$.Order' = 'True';
```

Practice Quiz (File 8)
1. d. You rewrote the query to avoid the use of
2. a. ClientID column
3. a. True
4. b. With a trailing wild card, MySQL can make use of an index created on the column to which the wildcard is assigned.