BASIS DATA
Pertemuan 12

Kelompok 4
Nama Anggota:

- Muhammad Luthfi Taufikurrahman (2022071053)
- Arellia Agustin (2022071060)
- Muhammad Faris Hafizh (2022071068)

1. Database optimization involves optimizing only the data retrieval statements that are executed against a database.
a. True
b. False

2. Which of the following guidelines should you follow when optimizing SQL select statements? Select all that apply.
Use OUTER JOIN instead of INNER JOIN wherever possible.
Avoid using leading wildcards in predicates.
Avoid using trailing wildcards in predicates.
Avoid using unnecessary columns in the SELECT clause.
Use INNER JOIN instead of OUTER JOIN wherever possible.

3. Identify the key characteristics of a Primary Index. Select all that apply.
1 point
A Primary Index is created using the CREATE INDEX syntax.
A Primary Key is also known as a Non-Clustered index.
A Primary Key is also known as a Clustered index.
A Primary Index is created automatically when a table is created.

4. The following query is executed on a table named Clients that contains the fields ClientID, FullName and ContactNumber.

```
SELECT ContactNumber
FROM Clients
WHERE FullName='Client Name here';
```

On which field within the table should a Secondary Index be created to optimize this query?

a. **FullName**
b. ContactNumber
c. ClientID

5. What keyword must be added before the following query to view its MySQL query execution plan?

```
_____ SELECT *
FROM Orders
WHERE ClientID='Cl1';
```

Answer: the answer is **EXPLAIN**

JAWABAN PRAKTIKUM

```
CREATE DATABASE Lucky_Shrub;
USE Lucky_Shrub;

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)
);

CREATE TABLE Employees
(
EmployeeID INT DEFAULT NULL,
FullName VARCHAR(100) DEFAULT NULL,
`Role` VARCHAR(50) DEFAULT NULL,
Department VARCHAR(255) DEFAULT NULL
);

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");

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*/
SELECT * FROM Orders;
SELECT OrderID, ProductID, Quantity, `Date` FROM Orders;

/* Task 2*/
SELECT * FROM Orders WHERE ClientID ='Cl1';
CREATE INDEX IdxClientID ON Orders(ClientID);
EXPLAIN SELECT * FROM Orders WHERE ClientID ='Cl1';

/* Task 3*/
SELECT * FROM Employees WHERE FullName LIKE '%Tolo';
ALTER TABLE Employees
ADD COLUMN ReverseFullName VARCHAR(100);

UPDATE Employees
SET ReverseFullName = CONCAT(SUBSTRING_INDEX(FullName, ' ', -1), ' ',
SUBSTRING_INDEX(FullName, ' ', 1));

CREATE INDEX IdxReverseFullName ON Employees (ReverseFullName);
```