Nama : Muhamad Shidiq Maulana

Tugas MySQL

Soal :

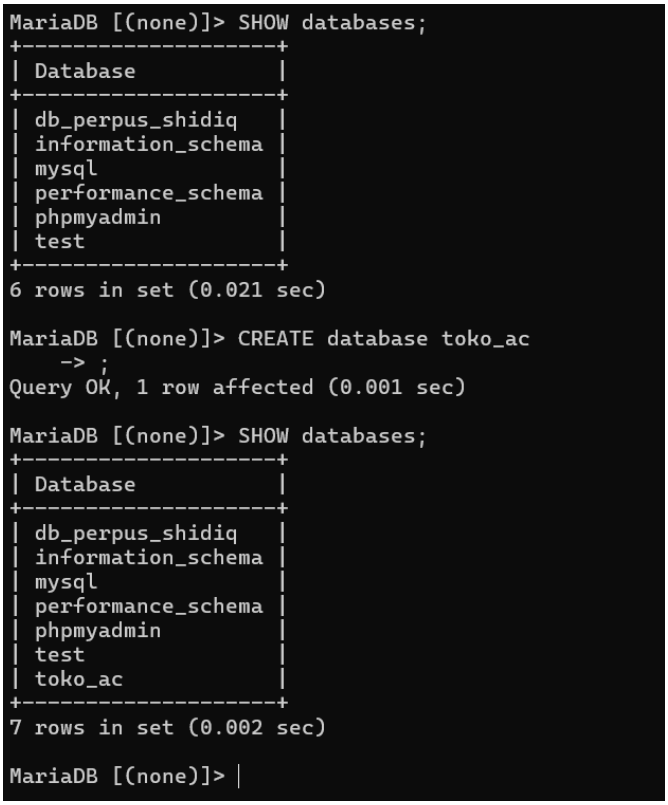
Buatlah query berikut berdasarkan spreadsheet **disini** :

1. Buatlah tabel nya menggunakan query sql
2. Tampilkan jumlah perbaikan yang sudah dibayar
3. Tampilkan tahun beserta jumlah perbaikan yang dilakukan pada tahun tersebut
4. Tampilkan data client yang tidak pernah melakukan perbaikan ac
5. Tampilkan data client beserta jumlah perbaikan yang dilakukan, urutkan berdasarkan perbaikan terbanyak
6. Tampilkan data technician beserta jumlah uang yang diperoleh, urutkan berdasarkan perolehan terbanyak

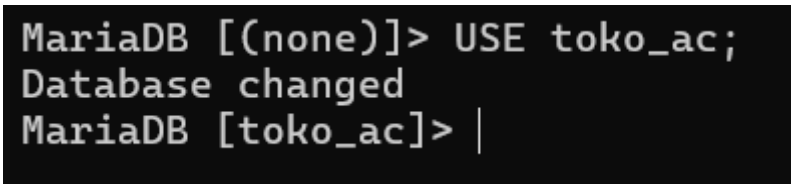
Kumpulkan **query** dan **sceenshot hasil query** dalam bentuk **pdf** , soal dan jawaban harus **urut** dari nomor 1 hingga 6 !

Jawaban :

1. **Membuat tabel menggunakan query sql**



Pertama-tama saya mengecek terlebih dahulu isi dari database, menggunakan command **SHOW databases;** untuk melihat ada apa saja database yang sudah ada. Selanjutnya saya membuat database baru yang bernama toko_ac dengan menggunkan commmad **CREATE database toko_ac;**



Untuk mengakses database yang sudah dibuat yaitu toko_ac, kita gunakan commad **USE toko_ac;**

Selanjutnya membuat table, sebagaimana dalam file spreadsheet yang telah dicantumkan. Kita harus membuat 4 TABLE (users, roles, ac, service).

Commad untuk membuat TABLE :

a. **TABLE users**

CREATE TABLE IF NOT EXISTS users (

id INT AUTO_INCREMENT PRIMARY KEY,

```
    name VARCHAR(255),
    email VARCHAR(255),
    password VARCHAR(255),
    gender CHAR(1),
    photo VARCHAR(255),
    address VARCHAR(255),
    role INT
);
```

b. TABLE roles

```
CREATE TABLE IF NOT EXISTS roles (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255)
);
```

c. TABLE ac

```
CREATE TABLE IF NOT EXISTS ac (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    brand VARCHAR(255),
    pk FLOAT,
    price INT
);
```

d. TABLE service

```
CREATE TABLE IF NOT EXISTS service (
    id INT AUTO_INCREMENT PRIMARY KEY,
    technician_id INT,
    client_id INT,
    ac_id INT,
    date DATE,
    status VARCHAR(20),
    FOREIGN KEY (technician_id) REFERENCES users(id),
    FOREIGN KEY (client_id) REFERENCES users(id),
    FOREIGN KEY (ac_id) REFERENCES ac(id)
);
```

```
MariaDB [toko_ac]> CREATE TABLE IF NOT EXISTS users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255),
    ->     email VARCHAR(255),
    ->     password VARCHAR(255),
    ->     gender CHAR(1),
    ->     photo VARCHAR(255),
    ->     address VARCHAR(255),
    ->     role INT
    -> );
Query OK, 0 rows affected (0.007 sec)

MariaDB [toko_ac]> SHOW TABLES;
+-------------------+
| Tables_in_toko_ac |
+-------------------+
| users             |
+-------------------+
1 row in set (0.001 sec)

MariaDB [toko_ac]> DESCRIBE users;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| name     | varchar(255) | YES  |     | NULL    |                |
| email    | varchar(255) | YES  |     | NULL    |                |
| password | varchar(255) | YES  |     | NULL    |                |
| gender   | char(1)      | YES  |     | NULL    |                |
| photo    | varchar(255) | YES  |     | NULL    |                |
| address  | varchar(255) | YES  |     | NULL    |                |
| role     | int(11)      | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
8 rows in set (0.020 sec)
```

Disini saya coba untuk menggunakan commadnya dimulai dari pembuatan TABLE users. Commad **SHOW TABLES;** adalah commad untuk melihat TABLE yang sudah dibuat. Sedangkan **DESCRIBE users;** untuk melihat penjelasan dari TABLE users.

```
MariaDB [toko_ac]> CREATE TABLE IF NOT EXISTS roles (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.007 sec)

MariaDB [toko_ac]>
```

Diatas proses penginputan commad untuk mebuat TABLE roles.

```
MariaDB [toko_ac]> CREATE TABLE IF NOT EXISTS ac (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(255),
    ->     brand VARCHAR(255),
    ->     pk FLOAT,
    ->     price INT
    -> );
Query OK, 0 rows affected (0.008 sec)

MariaDB [toko_ac]>
```

Diatas proses penginputan commad untuk mebuat TABLE ac.

```
MariaDB [toko_ac]> CREATE TABLE IF NOT EXISTS service (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     technician_id INT,
    ->     client_id INT,
    ->     ac_id INT,
    ->     date DATE,
    ->     status VARCHAR(20),
    ->     FOREIGN KEY (technician_id) REFERENCES users(id),
    ->     FOREIGN KEY (client_id) REFERENCES users(id),
    ->     FOREIGN KEY (ac_id) REFERENCES ac(id)
    -> );
Query OK, 0 rows affected (0.030 sec)

MariaDB [toko_ac]>
```

Diatas proses penginputan commad untuk mebuat TABLE service.

```
MariaDB [toko_ac]> SHOW TABLES;
+------------------+
| Tables_in_toko_ac |
+------------------+
| ac               |
| roles            |
| service          |
| users            |
+------------------+
4 rows in set (0.001 sec)

MariaDB [toko_ac]>
```

Dan command SHOW TABLES; untuk mengecek kembali TABLE apa saja yang sudah dibuat.

Selanjutnya adalah proses untuk mengiputan data untuk setiap TABLEnya.

Berikut commad untuk memsukkan data untuk setiap TABLE :

a. **TABLE users**

INSERT INTO users (name, email, password, gender, photo, address, role) VALUES

('Fulan', 'fulan@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 1),

('Fulanah', 'fulanah@gmail.com', '******', 'P', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 2),

('Ardi', 'ardi@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 2),

('Samsudin', 'samsudin@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 2),

('Eko', 'eko@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 2),

('Sugeng', 'sugeng@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Alif', 'alif@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Siti', 'siti@gmail.com', '******', 'P', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Juminten', 'juminten@gmail.com', '******', 'P', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Paijo', 'paijo@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Saifuddin', 'saifuddin@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Daffa', 'daffa@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Akbar', 'akbar@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Rafli', 'rafli@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),

('Rini', 'rini@gmail.com', '******', 'P', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3);

b. TABLE roles

INSERT INTO roles (name) VALUES

('admin'),

('technician'),

('client');

c. TABLE ac

INSERT INTO ac (name, brand, pk, price) VALUES

('LG-1', 'LG', 0.5, 50000),

('Sharp-2', 'Sharp', 1, 60000),

('Panasonic-3', 'Panasonic', 2, 70000),

('Samsung-4', 'Samsung', 0.5, 80000),

('Daikin-5', 'Daikin', 1, 90000),

('Gree-6', 'Gree', 2, 100000),

('Polytron-7', 'Polytron', 0.5, 110000),

('Electrolux-8', 'Electrolux', 1, 120000),

('Aqua-9', 'Aqua', 2, 130000),

('Midea-10', 'Midea', 0.5, 140000),

('LG-11', 'LG', 1, 200000),

('Sharp-12', 'Sharp', 2, 210000),

('Panasonic-13', 'Panasonic', 0.5, 220000),

('Samsung-14', 'Samsung', 1, 230000),

('Daikin-15', 'Daikin', 2, 240000),

('Gree-16', 'Gree', 0.5, 250000),

('Polytron-17', 'Polytron', 1, 260000),

('Electrolux-18', 'Electrolux', 2, 270000),

('Aqua-19', 'Aqua', 0.5, 280000),

('Midea-20', 'Midea', 1, 290000);

d. TABLE service

INSERT INTO service (technician_id, client_id, ac_id, date, status) VALUES

(2, 6, 1, '2020-06-01', 'finish'),

(3, 7, 2, '2020-05-01', 'finish'),

(4, 8, 3, '2020-06-02', 'finish'),

(5, 9, 4, '2021-03-03', 'finish'),

(2, 6, 5, '2021-12-05', 'finish'),

(3, 7, 6, '2021-12-25', 'finish'),

(4, 10, 7, '2022-01-01', 'finish'),

(5, 11, 8, '2022-02-02', 'finish'),

(2, 6, 9, '2022-04-04', 'finish'),

(3, 7, 10, '2023-05-05', 'on repair'),

(4, 12, 11, '2023-06-06', 'on repair'),

(5, 13, 12, '2023-07-07', 'on repair'),

(2, 6, 13, '2023-08-08', 'paid'),

(3, 7, 14, '2023-09-09', 'paid'),

(4, 14, 15, '2023-10-10', 'unpaid');

Selanjutnya adalah tinggal menginputkan command-commad yang ada diatas, disini saya coba dahulu untuk menginputkan data untuk **TABLE USERS**.

```
MariaDB [toko_ac]> INSERT INTO users (name, email, password, gender, photo, address, role) VALUES
    -> ('Fulan', 'fulan@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 1),
    -> ('Fulanah', 'fulanah@gmail.com', '******', 'P', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 2),
    -> ('Ardi', 'ardi@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 2),
    -> ('Samsudin', 'samsudin@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 2),
    -> ('Eko', 'eko@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 2),
    -> ('Sugeng', 'sugeng@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Alif', 'alif@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Siti', 'siti@gmail.com', '******', 'P', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Juminten', 'juminten@gmail.com', '******', 'P', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Paijo', 'paijo@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Saifuddin', 'saifuddin@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Daffa', 'daffa@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Akbar', 'akbar@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Rafli', 'rafli@gmail.com', '******', 'L', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3),
    -> ('Rini', 'rini@gmail.com', '******', 'P', 'https://lorem.ipsum/dolor.png', 'Jl. Cisitu Indah VI no 6', 3);
Query OK, 15 rows affected (0.049 sec)
Records: 15  Duplicates: 0  Warnings: 0
```

Saya coba cek isi dari TABLE USERS dengan command **SELECT * from users;**

```
MariaDB [toko_ac]> SELECT * from users;
+----+-----------+--------------------+----------+--------+------------------------------+-------------------------+------+
| id | name      | email              | password | gender | photo                        | address                 | role |
+----+-----------+--------------------+----------+--------+------------------------------+-------------------------+------+
|  1 | Fulan     | fulan@gmail.com    | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    1 |
|  2 | Fulanah   | fulanah@gmail.com  | ******   | P      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    2 |
|  3 | Ardi      | ardi@gmail.com     | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    2 |
|  4 | Samsudin  | samsudin@gmail.com | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    2 |
|  5 | Eko       | eko@gmail.com      | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    2 |
|  6 | Sugeng    | sugeng@gmail.com   | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
|  7 | Alif      | alif@gmail.com     | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
|  8 | Siti      | siti@gmail.com     | ******   | P      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
|  9 | Juminten  | juminten@gmail.com | ******   | P      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
| 10 | Paijo     | paijo@gmail.com    | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
| 11 | Saifuddin | saifuddin@gmail.com | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
| 12 | Daffa     | daffa@gmail.com    | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
| 13 | Akbar     | akbar@gmail.com    | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
| 14 | Rafli     | rafli@gmail.com    | ******   | L      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
| 15 | Rini      | rini@gmail.com     | ******   | P      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
+----+-----------+--------------------+----------+--------+------------------------------+-------------------------+------+
15 rows in set (0.000 sec)
```

Ternyata berhasil tinggal isi data TABLE yang belum diisi.

```
MariaDB [toko_ac]> INSERT INTO roles (name) VALUES
    -> ('admin'),
    -> ('technician'),
    -> ('client');
Query OK, 3 rows affected (0.003 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [toko_ac]> SELECT * from roles;
+----+------------+
| id | name       |
+----+------------+
|  1 | admin      |
|  2 | technician |
|  3 | client     |
+----+------------+
3 rows in set (0.000 sec)
```

Ini untuk penginputan dan pengecekan **TABLE roles**.

```
MariaDB [toko_ac]> INSERT INTO ac (name, brand, pk, price) VALUES
    -> ('LG-1', 'LG', 0.5, 50000),
    -> ('Sharp-2', 'Sharp', 1, 60000),
    -> ('Panasonic-3', 'Panasonic', 2, 70000),
    -> ('Samsung-4', 'Samsung', 0.5, 80000),
    -> ('Daikin-5', 'Daikin', 1, 90000),
    -> ('Gree-6', 'Gree', 2, 100000),
    -> ('Polytron-7', 'Polytron', 0.5, 110000),
    -> ('Electrolux-8', 'Electrolux', 1, 120000),
    -> ('Aqua-9', 'Aqua', 2, 130000),
    -> ('Midea-10', 'Midea', 0.5, 140000),
    -> ('LG-11', 'LG', 1, 200000),
    -> ('Sharp-12', 'Sharp', 2, 210000),
    -> ('Panasonic-13', 'Panasonic', 0.5, 220000),
    -> ('Samsung-14', 'Samsung', 1, 230000),
    -> ('Daikin-15', 'Daikin', 2, 240000),
    -> ('Gree-16', 'Gree', 0.5, 250000),
    -> ('Polytron-17', 'Polytron', 1, 260000),
    -> ('Electrolux-18', 'Electrolux', 2, 270000),
    -> ('Aqua-19', 'Aqua', 0.5, 280000),
    -> ('Midea-20', 'Midea', 1, 290000);
Query OK, 20 rows affected (0.003 sec)
Records: 20  Duplicates: 0  Warnings: 0

MariaDB [toko_ac]> SELECT * from ac;
+----+--------------+-----------+------+--------+
| id | name         | brand     | pk   | price  |
+----+--------------+-----------+------+--------+
|  1 | LG-1         | LG        |  0.5 |  50000 |
|  2 | Sharp-2      | Sharp     |    1 |  60000 |
|  3 | Panasonic-3  | Panasonic |    2 |  70000 |
|  4 | Samsung-4    | Samsung   |  0.5 |  80000 |
|  5 | Daikin-5     | Daikin    |    1 |  90000 |
|  6 | Gree-6       | Gree      |    2 | 100000 |
|  7 | Polytron-7   | Polytron  |  0.5 | 110000 |
|  8 | Electrolux-8 | Electrolux|    1 | 120000 |
|  9 | Aqua-9       | Aqua      |    2 | 130000 |
| 10 | Midea-10     | Midea     |  0.5 | 140000 |
| 11 | LG-11        | LG        |    1 | 200000 |
| 12 | Sharp-12     | Sharp     |    2 | 210000 |
| 13 | Panasonic-13 | Panasonic |  0.5 | 220000 |
| 14 | Samsung-14   | Samsung   |    1 | 230000 |
| 15 | Daikin-15    | Daikin    |    2 | 240000 |
| 16 | Gree-16      | Gree      |  0.5 | 250000 |
| 17 | Polytron-17  | Polytron  |    1 | 260000 |
| 18 | Electrolux-18| Electrolux|    2 | 270000 |
| 19 | Aqua-19      | Aqua      |  0.5 | 280000 |
| 20 | Midea-20     | Midea     |    1 | 290000 |
+----+--------------+-----------+------+--------+
20 rows in set (0.000 sec)
```

Ini untuk penginputan dan pengecekan **TABLE ac**.

```
MariaDB [toko_ac]> INSERT INTO service (technician_id, client_id, ac_id, date, status) VALUES
    -> (2, 6, 1, '2020-06-01', 'finish'),
    -> (3, 7, 2, '2020-05-01', 'finish'),
    -> (4, 8, 3, '2020-06-02', 'finish'),
    -> (5, 9, 4, '2021-03-03', 'finish'),
    -> (2, 6, 5, '2021-12-05', 'finish'),
    -> (3, 7, 6, '2021-12-25', 'finish'),
    -> (4, 10, 7, '2022-01-01', 'finish'),
    -> (5, 11, 8, '2022-02-02', 'finish'),
    -> (2, 6, 9, '2022-04-04', 'finish'),
    -> (3, 7, 10, '2023-05-05', 'on repair'),
    -> (4, 12, 11, '2023-06-06', 'on repair'),
    -> (5, 13, 12, '2023-07-07', 'on repair'),
    -> (2, 6, 13, '2023-08-08', 'paid'),
    -> (3, 7, 14, '2023-09-09', 'paid'),
    -> (4, 14, 15, '2023-10-10', 'unpaid');
Query OK, 15 rows affected (0.016 sec)
Records: 15  Duplicates: 0  Warnings: 0

MariaDB [toko_ac]> SELECT * from service;
+----+---------------+-----------+-------+------------+-----------+
| id | technician_id | client_id | ac_id | date       | status    |
+----+---------------+-----------+-------+------------+-----------+
|  1 |             2 |         6 |     1 | 2020-06-01 | finish    |
|  2 |             3 |         7 |     2 | 2020-05-01 | finish    |
|  3 |             4 |         8 |     3 | 2020-06-02 | finish    |
|  4 |             5 |         9 |     4 | 2021-03-03 | finish    |
|  5 |             2 |         6 |     5 | 2021-12-05 | finish    |
|  6 |             3 |         7 |     6 | 2021-12-25 | finish    |
|  7 |             4 |        10 |     7 | 2022-01-01 | finish    |
|  8 |             5 |        11 |     8 | 2022-02-02 | finish    |
|  9 |             2 |         6 |     9 | 2022-04-04 | finish    |
| 10 |             3 |         7 |    10 | 2023-05-05 | on repair |
| 11 |             4 |        12 |    11 | 2023-06-06 | on repair |
| 12 |             5 |        13 |    12 | 2023-07-07 | on repair |
| 13 |             2 |         6 |    13 | 2023-08-08 | paid      |
| 14 |             3 |         7 |    14 | 2023-09-09 | paid      |
| 15 |             4 |        14 |    15 | 2023-10-10 | unpaid    |
+----+---------------+-----------+-------+------------+-----------+
15 rows in set (0.000 sec)

MariaDB [toko_ac]>
```

Ini untuk penginputan dan pengecekan **TABLE service**.

2. **Menampilkan jumlah perbaikan yang sudah dibayar**

```
MariaDB [toko_ac]> SELECT COUNT(*) AS jumlah_perbaikan_dibayar
    -> FROM service
    -> WHERE status = 'paid';
+--------------------------+
| jumlah_perbaikan_dibayar |
+--------------------------+
|                        2 |
+--------------------------+
1 row in set (0.001 sec)
```

SELECT COUNT(*) AS jumlah_perbaikan_dibayar

FROM service

WHERE status = 'paid';

Perintah SQL di atas digunakan untuk menghitung jumlah baris (atau jumlah perbaikan) di dalam tabel `service` yang memiliki status `'paid'`.

1. `SELECT COUNT(*) AS jumlah_perbaikan_dibayar`: Ini adalah bagian dari perintah yang bertanggung jawab untuk memilih data dari tabel. Dalam hal ini, kita menggunakan fungsi agregat `COUNT(*)` untuk menghitung jumlah baris dalam tabel. `COUNT(*)` menghitung jumlah baris di dalam hasil pemilihan. `AS jumlah_perbaikan_dibayar` memberikan alias (nama alternatif) untuk hasil hitungan, sehingga hasilnya akan ditampilkan dengan nama tersebut.

2. `FROM service`: Ini menunjukkan tabel yang akan digunakan dalam operasi pemilihan, dalam hal ini, tabel yang digunakan adalah `service`.

3. `WHERE status = 'paid'`: Ini adalah klausa WHERE yang membatasi baris yang akan dihitung berdasarkan kondisi yang ditentukan. Kondisi di sini adalah `status = 'paid'`, yang berarti hanya baris dengan nilai kolom `status` yang sama dengan `'paid'` yang akan dihitung.

Jadi, keseluruhan perintah akan menghasilkan satu baris dengan satu kolom yang berisi jumlah perbaikan yang telah dibayar dalam tabel `service`. Alias `jumlah_perbaikan_dibayar` digunakan untuk memberi nama pada kolom hasil hitungan.

3. **Menamampilkan tahun beserta jumlah perbaikan yang dilakukan pada tahun tersebut**

```
MariaDB [toko_ac]> SELECT YEAR(date) AS tahun, COUNT(*) AS jumlah_perbaikan
    -> FROM service
    -> GROUP BY YEAR(date);
+-------+-----------------+
| tahun | jumlah_perbaikan |
+-------+-----------------+
|  2020 |               3 |
|  2021 |               3 |
|  2022 |               3 |
|  2023 |               6 |
+-------+-----------------+
4 rows in set (0.001 sec)
```

SELECT YEAR(date) AS tahun, COUNT(*) AS jumlah_perbaikan

FROM service

GROUP BY YEAR(date);

Perintah SQL di atas digunakan untuk menghitung jumlah perbaikan yang dilakukan pada setiap tahun, dengan mengelompokkannya berdasarkan tahun.

1. `SELECT YEAR(date) AS tahun, COUNT(*) AS jumlah_perbaikan`: Ini adalah bagian dari perintah yang bertanggung jawab untuk memilih data dari tabel. Dalam hal ini, kita menggunakan fungsi `YEAR(date)` untuk mengekstrak tahun dari kolom `date`, dan memberikan alias `tahun` pada hasilnya. Selain itu, kita menggunakan fungsi agregat `COUNT(*)` untuk menghitung jumlah baris dalam setiap kelompok tahun. `AS jumlah_perbaikan` memberikan alias pada hasil hitungan, sehingga hasilnya akan ditampilkan dengan nama tersebut.

2. `FROM service`: Ini menunjukkan tabel yang akan digunakan dalam operasi pemilihan, dalam hal ini, tabel yang digunakan adalah `service`.

3. `GROUP BY YEAR(date)`: Ini adalah klausa `GROUP BY` yang digunakan untuk mengelompokkan baris-baris berdasarkan tahun dari kolom `date`. Dengan kata lain, data akan dikelompokkan menjadi kelompok-kelompok yang sesuai dengan tahunnya.

Jadi, keseluruhan perintah akan menghasilkan satu baris untuk setiap tahun yang ada dalam tabel `service`. Setiap baris akan berisi tahun tersebut (dengan nama kolom `tahun`) dan jumlah perbaikan yang dilakukan pada tahun tersebut (dengan nama kolom `jumlah_perbaikan`).

4. **Menampilkan data client yang tidak pernah melakukan perbaikan AC**

```
MariaDB [toko_ac]> SELECT *
    -> FROM users
    -> WHERE role = 3
    -> AND id NOT IN (
    ->     SELECT DISTINCT client_id
    ->     FROM service
    -> );
+----+------+----------------+----------+--------+-----------------------------+-------------------------+------+
| id | name | email          | password | gender | photo                       | address                 | role |
+----+------+----------------+----------+--------+-----------------------------+-------------------------+------+
| 15 | Rini | rini@gmail.com | ******   | P      | https://lorem.ipsum/dolor.png | Jl. Cisitu Indah VI no 6 |    3 |
+----+------+----------------+----------+--------+-----------------------------+-------------------------+------+
1 row in set (0.001 sec)
```

SELECT *

FROM users

WHERE role = 3

AND id NOT IN (

SELECT DISTINCT client_id

FROM service

);

Perintah SQL di atas digunakan untuk memilih semua data dari tabel `users` di mana peran (role) setiap pengguna adalah 3 (mengindikasikan bahwa mereka adalah klien) dan di mana ID pengguna tidak termasuk dalam hasil subquery yang memilih ID klien yang terlibat dalam layanan.

1. `SELECT * FROM users`: Ini adalah bagian dari perintah yang bertanggung jawab untuk memilih semua kolom dari tabel `users`.

2. `WHERE role = 3`: Ini adalah klausa `WHERE` yang membatasi baris yang akan dipilih berdasarkan nilai kolom `role`. Dalam hal ini, hanya baris-baris di mana nilai kolom `role` adalah 3 yang akan dipilih, yang berarti hanya pengguna dengan peran klien yang akan dipilih.

3. `AND id NOT IN (SELECT DISTINCT client_id FROM service)`: Ini adalah klausa `AND` yang memperluas kondisi `WHERE`. Kondisi ini mengecek bahwa ID pengguna tidak termasuk dalam hasil subquery. Subquery tersebut memilih ID klien yang telah terlibat dalam layanan. Dengan kata lain, perintah ini memilih pengguna yang memiliki peran klien tetapi tidak memiliki catatan layanan terkait dalam tabel `service`.

Jadi, keseluruhan perintah akan menghasilkan semua data pengguna yang memiliki peran klien tetapi belum pernah melakukan perbaikan AC.

5. **Menampilkan data client beserta jumlah perbaikan yang dilakukan, urutkan berdasarkan perbaikan terbanyak**

```
MariaDB [toko_ac]> SELECT u.id, u.name, COUNT(s.id) AS jumlah_perbaikan
    -> FROM users u
    -> LEFT JOIN service s ON u.id = s.client_id
    -> WHERE u.role = 3
    -> GROUP BY u.id
    -> ORDER BY jumlah_perbaikan DESC;
+----+----------+-----------------+
| id | name     | jumlah_perbaikan |
+----+----------+-----------------+
|  6 | Sugeng   |               4 |
|  7 | Alif     |               4 |
|  8 | Siti     |               1 |
|  9 | Juminten |               1 |
| 10 | Paijo    |               1 |
| 11 | Saifuddin |              1 |
| 12 | Daffa    |               1 |
| 13 | Akbar    |               1 |
| 14 | Rafli    |               1 |
| 15 | Rini     |               0 |
+----+----------+-----------------+
10 rows in set (0.000 sec)
```

SELECT u.id, u.name, COUNT(s.id) AS jumlah_perbaikan

FROM users u

LEFT JOIN service s ON u.id = s.client_id

WHERE u.role = 3

GROUP BY u.id

ORDER BY jumlah_perbaikan DESC;

Perintah SQL di atas digunakan untuk menghitung jumlah perbaikan yang dilakukan oleh setiap klien, dan mengurutkannya berdasarkan jumlah perbaikan dari yang terbanyak ke yang terendah.

1. `SELECT u.id, u.name, COUNT(s.id) AS jumlah_perbaikan`: Ini adalah bagian dari perintah yang bertanggung jawab untuk memilih ID dan nama dari tabel `users`, serta menghitung jumlah perbaikan yang dilakukan oleh setiap klien. Kita menggunakan fungsi agregat `COUNT(s.id)` untuk menghitung jumlah baris dalam setiap kelompok klien (berdasarkan ID pengguna), dan memberikan alias `jumlah_perbaikan` pada hasil hitungan.

2. `FROM users u`: Ini menunjukkan tabel utama yang digunakan dalam operasi pemilihan, dalam hal ini, tabel `users` diberi alias `u`.

3. `LEFT JOIN service s ON u.id = s.client_id`: Ini adalah jenis join yang digunakan untuk menggabungkan tabel `users` dengan tabel `service` berdasarkan kolom `id` dari `users` dan kolom `client_id` dari `service`. Dengan menggunakan `LEFT JOIN`, kita memastikan bahwa semua baris dari tabel `users` akan tetap dipertahankan, bahkan jika tidak ada kecocokan dalam tabel `service`.

4. `WHERE u.role = 3`: Ini adalah klausa `WHERE` yang membatasi baris yang akan dipilih dari tabel `users`. Dalam hal ini, hanya baris-baris di mana nilai kolom `role` adalah 3 (klien) yang akan dipilih.

5. `GROUP BY u.id`: Ini adalah klausa `GROUP BY` yang digunakan untuk mengelompokkan hasil berdasarkan ID pengguna. Dengan ini, kita menghitung jumlah perbaikan untuk setiap klien secara terpisah.

6. `ORDER BY jumlah_perbaikan DESC`: Ini adalah klausa `ORDER BY` yang digunakan untuk mengurutkan hasil berdasarkan jumlah perbaikan dari yang terbanyak ke yang terendah (dalam urutan menurun).

Jadi, keseluruhan perintah akan menghasilkan ID, nama, dan jumlah perbaikan yang dilakukan oleh setiap klien, yang diurutkan berdasarkan jumlah perbaikan dari yang terbanyak ke yang terendah.

6. **Menampilkan data technician beserta jumlah uang yang diperoleh, urutkan berdasarkan perolehan terbanyak**

```
MariaDB [toko_ac]> SELECT u.id, u.name, SUM(a.price) AS total_uang_diperoleh
    -> FROM users u
    -> INNER JOIN service s ON u.id = s.technician_id
    -> INNER JOIN ac a ON s.ac_id = a.id
    -> WHERE u.role = 2 AND s.status = 'paid'
    -> GROUP BY u.id
    -> ORDER BY total_uang_diperoleh DESC;
+----+--------+----------------------+
| id | name   | total_uang_diperoleh |
+----+--------+----------------------+
|  3 | Ardi   |               230000 |
|  2 | Fulanah |              220000 |
+----+--------+----------------------+
2 rows in set (0.001 sec)
```

SELECT u.id, u.name, SUM(a.price) AS total_uang_diperoleh

FROM users u

INNER JOIN service s ON u.id = s.technician_id

INNER JOIN ac a ON s.ac_id = a.id

WHERE u.role = 2 AND s.status = 'paid'

GROUP BY u.id

ORDER BY total_uang_diperoleh DESC;

Perintah SQL di atas digunakan untuk menghitung total uang yang diperoleh oleh setiap teknisi dari layanan yang telah dibayar, dan mengurutkannya berdasarkan total uang yang diperoleh dari yang terbanyak ke yang terendah. Mari kita jelaskan setiap bagian dari perintah tersebut:

1. `SELECT u.id, u.name, SUM(a.price) AS total_uang_diperoleh`: Ini adalah bagian dari perintah yang bertanggung jawab untuk memilih ID, nama, dan menghitung total uang yang diperoleh oleh setiap teknisi. Kita menggunakan fungsi agregat `SUM(a.price)` untuk menjumlahkan harga dari semua AC yang diperbaiki oleh setiap teknisi, dan memberikan alias `total_uang_diperoleh` pada hasil penjumlahan.

2. `FROM users u`: Ini menunjukkan tabel utama yang digunakan dalam operasi pemilihan, dalam hal ini, tabel `users` diberi alias `u`.

3. `INNER JOIN service s ON u.id = s.technician_id`: Ini adalah jenis join yang digunakan untuk menggabungkan tabel `users` dengan tabel `service` berdasarkan kolom `id` dari

`users` dan kolom `technician_id` dari `service`. Dengan menggunakan `INNER JOIN`, kita hanya akan memilih baris-baris di mana ada kecocokan antara kedua tabel.

4. `INNER JOIN ac a ON s.ac_id = a.id`: Ini adalah jenis join lain yang digunakan untuk menggabungkan hasil join sebelumnya dengan tabel `ac` berdasarkan kolom `ac_id` dari tabel `service` dan kolom `id` dari tabel `ac`.

5. `WHERE u.role = 2 AND s.status = 'paid'`: Ini adalah klausa `WHERE` yang membatasi baris yang akan dipilih dari tabel. Dalam hal ini, kita memilih hanya baris-baris di mana peran (role) pengguna adalah 2 (teknisi) dan status layanan adalah 'paid'.

6. `GROUP BY u.id`: Ini adalah klausa `GROUP BY` yang digunakan untuk mengelompokkan hasil berdasarkan ID teknisi. Dengan ini, kita menghitung total uang yang diperoleh oleh setiap teknisi secara terpisah.

7. `ORDER BY total_uang_diperoleh DESC`: Ini adalah klausa `ORDER BY` yang digunakan untuk mengurutkan hasil berdasarkan total uang yang diperoleh dari yang terbanyak ke yang terendah (dalam urutan menurun).

Jadi, keseluruhan perintah akan menghasilkan ID, nama, dan total uang yang diperoleh oleh setiap teknisi dari layanan yang telah dibayar, yang diurutkan berdasarkan total uang yang diperoleh dari yang terbanyak ke yang terendah.