Transaction, Stored Procedures dan Stored Functions

Worksheet 6

Nama : Andries Nauvalentin Roestam

NIM : 3012210008

SOAL 6.1

1. Buatlah Procedure untuk mengupdate harga\_jual berdasarkan jenis produk tertentu (jenis\_produk\_id), beri nama procedure **pro\_naikan\_harga** memiliki parameter yang akan menerima argumen: Jenis Produk ID dan Persentase kenaikan harga.

*CREATE PROCEDURE ...*

*DELIMITER $$*

*MariaDB [dbpos1]> CREATE PROCEDURE pro\_naik(*

*-> IN jenis\_produk INT,*

*-> IN persentasi\_kenaikan INT )*

*-> BEGIN*

*-> UPDATE produk SET harga\_jual = harga\_jual + (harga\_jual \* persentasi\_kenaikan / 100)*

*-> WHERE jenis\_produk\_id = jenis\_produk;*

*-> END $$*

*DELIMITER ;*

*MariaDB [dbpos1]> CALL pro\_naik(1,4); 4 disini adalah 4%*

*CREATE PROCEDURE pro\_naikan\_harga*

1. Buat fungsi **umur** dengan parameter yang menerima inputan argumen tipe data date dan mengembalikan hasil perhitungan umur (tahun sekarang dikurang tahun inputan) dengan tipe data bilangan bulat (integer) positif.

*CREATE FUNCTION ...*

MariaDB [dbpos1]> CREATE FUNCTION umur(tgl\_lahir DATE)

-> RETURNS INT

-> BEGIN

-> DECLARE umur INT;

-> SET umur = YEAR(CURDATE()) - YEAR(tgl\_lahir);

-> RETURN umur;

-> END $$

Query OK, 0 rows affected (0.038 sec)

MariaDB [dbpos1]> SELECT nama, umur(tgl\_lahir) AS umur FROM pelanggan;

1. Buat fungsi **kategori\_harga** dengan parameter yang menerima inputan argument tipe data double dan mengembalikan tipe data string kategori harga berdasarkan:

* 0 – 500rb : murah
* 500rb – 3 juta : sedang
* 3jt – 10 juta : mahal
* > 10 juta : sangat mahal

*CREATE FUNCTION ...*

**DELIMITER $$**

**CREATE** **FUNCTION** kategori\_harga(harga **DOUBLE**)

**RETURNS** **VARCHAR**(20)

**BEGIN**

**DECLARE** kategori **VARCHAR**(20);

**IF** harga <= 500000 **THEN**

**SET** kategori = **'murah'**;

**ELSEIF** harga > 500000 **AND** harga <= 3000000 **THEN**

**SET** kategori = **'sedang'**;

**ELSEIF** harga > 3000000 **AND** harga <= 10000000 **THEN**

**SET** kategori = **'mahal'**;

**ELSE**

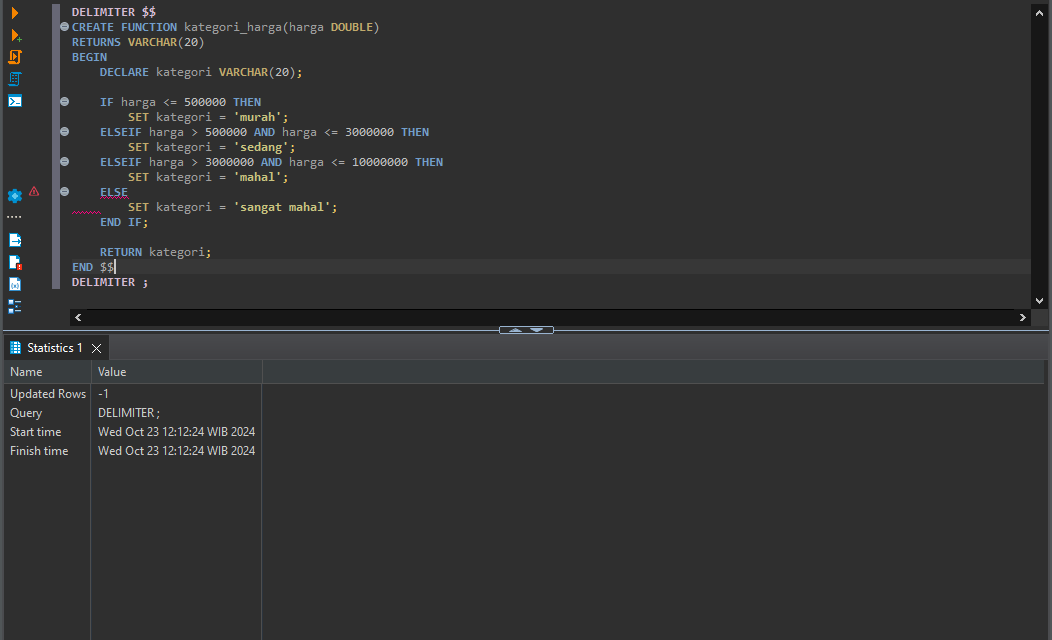
**SET** kategori = **'sangat mahal'**;

**END** **IF**;

**RETURN** kategori;

**END** $$

**DELIMITER ;**



SOAL 5.2

1. Buatlah sebuah transaction dengan skenario-skenario statement sebagai berikut:

* Mulai transaction
* Insert data produk sebanyak 3 record
* Update data stok salah satu produk
* Buat savepoint
* Hapus salah satu data pembayaran
* Kembali ke savepoint
* Update data iuran salah satu kartu
* Akhiri transaction dengan commit

*START TRANSACTION ...*

**START** **TRANSACTION**;

-- Insert data produk

**INSERT** **INTO** produk (nama\_produk, harga\_jual, stok, jenis\_produk\_id) **VALUES** (**'Produk A'**, 1000000, 10, 1);

**INSERT** **INTO** produk (nama\_produk, harga\_jual, stok, jenis\_produk\_id) **VALUES** (**'Produk B'**, 2000000, 15, 2);

**INSERT** **INTO** produk (nama\_produk, harga\_jual, stok, jenis\_produk\_id) **VALUES** (**'Produk C'**, 5000000, 20, 3);

-- Update data stok salah satu produk

**UPDATE** produk **SET** stok = stok - 5 **WHERE** nama\_produk = **'Produk A'**;

-- Buat savepoint

**SAVEPOINT** sebelum\_hapus;

-- Hapus salah satu data pembayaran (misalnya berdasarkan ID pembayaran)

**DELETE** **FROM** pembayaran **WHERE** id\_pembayaran = 1;

-- Kembali ke savepoint

**ROLLBACK** **TO** **SAVEPOINT** sebelum\_hapus;

-- Update data iuran salah satu kartu

**UPDATE** kartu **SET** iuran = iuran + 10000 **WHERE** id\_kartu = 1;

-- Akhiri transaction dengan commit

**COMMIT**;

A screenshot of a computer

Description automatically generated