Transaction, Stored Procedures dan Stored Functions

Worksheet 6

Nama : Muhamad Akmal Maulana

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*

*IF harga <= 500000 THEN*

*RETURN 'Murah';*

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

*RETURN 'Sedang';*

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

*RETURN 'Mahal';*

*ELSE*

*RETURN 'Sangat Mahal';*

*END IF;*

*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;*

*-- 1. Insert data produk sebanyak 3 record*

*INSERT INTO produk (kode, nama, stok, harga\_beli, harga\_jual, min\_stok)*

*VALUES*

*('P001', 'Produk A', 10, 100000, 120000, 5),*

*('P002', 'Produk B', 15, 200000, 250000, 10),*

*('P003', 'Produk C', 8, 50000, 75000, 3);*

*-- 2. Update data stok salah satu produk*

*UPDATE produk SET stok = stok + 5 WHERE kode = 'P001';*

*-- 3. Buat savepoint*

*SAVEPOINT sebelum\_hapus;*

*-- 4. Hapus salah satu data pembayaran*

*DELETE FROM pembayaran WHERE id = 1;*

*-- 5. Kembali ke savepoint*

*ROLLBACK TO sebelum\_hapus;*

*-- 6. Update data iuran salah satu kartu*

*UPDATE kartu SET iuran\_tahunan = iuran\_tahunan + 5000 WHERE id = 2;*

*-- 7. Akhiri transaction dengan commit*

*COMMIT;*