Stored Procedures, Stored Functions dan Trigger

Worksheet 6

Nama : Moh Farrij Wajdi | Politeknik Negeri Jember | Group 01 | E41210223

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 $$*

*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 ;*

*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 ...*

DELIMITER $$

DROP FUNCTION IF EXISTS umur $$

CREATE FUNCTION umur(tgl\_lahir DATE)

RETURNS INT

BEGIN

DECLARE umur INT;

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

RETURN umur;

END $$

DELIMITER ;

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

*MariaDB [dbpos\_sib6]> DELIMITER $$*

*MariaDB [dbpos\_sib6]>*

*MariaDB [dbpos\_sib6]> 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$$*

*Query OK, 0 rows affected (0.002 sec)*

*MariaDB [dbpos\_sib6]>*

*MariaDB [dbpos\_sib6]> DELIMITER ;*

*MariaDB [dbpos\_sib6]> SELECT nama, harga\_jual, kategori\_harga(harga\_jual) AS kategori FROM produk;*

Soal 6.2

*Trigger*

1. Buatlah bisnis proses pembayaran dengan menggunakan trigers, dengan skenario sebagai berikut :

- pelanggan memesan didalam table pesanan

- dilanjutkan dengan proses pembayaran di table pembayaran

- didalam table pembayaran tambahkan kolom status\_pembayaran

- jika pesanan sudah dibayar maka status pembayaran akan berubah menjadi lunas

1. Pelanggan memesan didalam table pesanan

SELECT \* FROM pesanan;

Didalam table pembayaran tambahkan kolom status\_pembayaran

ALTER TABLE pembayaran ADD status\_pembayaran varchar(25);

2. Dilanjutkan dengan proses pembayaran di table pembayaran

DELIMITER $$

CREATE TRIGGER cek\_pembayaran BEFORE INSERT ON pembayaran

FOR EACH ROW

BEGIN

DECLARE total\_bayar DECIMAL(10, 2);

DECLARE total\_pesanan DECIMAL(10, 2);

SELECT SUM(jumlah) INTO total\_bayar FROM pembayaran WHERE pesanan\_id = NEW.pesanan\_id;

SELECT total INTO total\_pesanan FROM pesanan WHERE id = NEW.pesanan\_id;

4. Jika pesanan sudah dibayar maka status pembayaran akan berubah menjadi lunas

IF total\_bayar + NEW.jumlah >= total\_pesanan THEN

SET NEW.status\_pembayaran = 'Lunas';

END IF;

END $$

DELIMITER ;

// Menambahkan data pada tabel pembayaran

INSERT INTO pembayaran (no\_kuitansi, tanggal, jumlah, ke, pesanan\_id, status\_pembayaran)

VALUES ('KWI001', '2023-03-03', 200000, 1, 1);

1. Buatlah Stored Procedure dengan nama **kurangi\_stok** untuk mengurangi stok produk. Stok berkurang sesuai dengan jumlah pesanan produk.

*DELIMITER $$*

*CREATE PROCEDURE kurangi\_stok(IN produk\_id INT, IN jumlah\_pesanan INT)*

*BEGIN*

*DECLARE stok\_produk INT;*

*-- Dapatkan jumlah stok produk saat ini*

*SELECT stok INTO stok\_produk FROM produk WHERE id = produk\_id;*

*-- Kurangi stok dengan jumlah pesanan*

*SET stok\_produk = stok\_produk - jumlah\_pesanan;*

*-- Pastikan stok tidak negatif*

*IF stok\_produk < 0 THEN*

*SIGNAL SQLSTATE '45000'*

*SET MESSAGE\_TEXT = 'Stok produk tidak mencukupi';*

*END IF;*

*-- Update stok produk yang telah dikurangi*

*UPDATE produk SET stok = stok\_produk WHERE id = produk\_id;*

*END $$*

*DELIMITER ;*

1. Buatlah Trigger dengan nama **trig\_kurangi\_stok** yang akan mengurangi stok produk jika terjadi transaksi pesanan oleh pelanggan (memanggil stored procedure kurangi\_stok soal no 1).

Trigger ini aktif setelah trigger **after\_pesanan\_items\_insert** (trigger pada contoh 3).

*CREATE TRIGGER ...*

*DELIMITER $$*

*CREATE TRIGGER trig\_kurangi\_stok AFTER INSERT ON pesanan\_items*

*FOR EACH ROW*

*BEGIN*

*-- Memanggil stored procedure kurangi\_stok untuk mengurangi stok produk*

*CALL kurangi\_stok(NEW.produk\_id, NEW.qty);*

*END $$*

*DELIMITER ;*

*SELECT \* FROM `produk`*

*INSERT INTO pesanan\_items (produk\_id, pesanan\_id, qty, harga) VALUES*

*(7, 3, 2, 7000)*

*SELECT \* FROM `produk`*