Stored Procedures, Stored Functions dan Trigger

Worksheet 6

Nama : Zahra Talita Putri Rumalia

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

*MariaDB [dbpos3]> DELIMITER $$*

*MariaDB [dbpos3]> CREATE FUNCTION kategori\_harga(harga DOUBLE)*

*-> RETURNS VARCHAR(20)*

*-> BEGIN*

*-> DECLARE keterangan VARCHAR(20);*

*-> CASE*

*-> WHEN harga BETWEEN 0 AND 500000 then*

*-> SET keterangan = 'Murah';*

*-> WHEN harga BETWEEN 500000 AND 3000000 THEN*

*-> SET keterangan = 'SEDANG';*

*-> WHEN harga BETWEEN 3000000 AND 10000000 THEN*

*-> SET keterangan = 'MAHAL';*

*-> ELSE*

*-> SET keterangan = 'Sangat Mahal';*

*-> END CASE;*

*-> RETURN keterangan;*

*-> END $$*

*Query OK, 0 rows affected (0.068 sec)*

*MariaDB [dbpos3]> select kategori\_harga("9000000");*

*+---------------------------+*

*| kategori\_harga("9000000") |*

*+---------------------------+*

*| MAHAL |*

*+---------------------------+*

*1 row in set (0.001 sec)*

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.

MariaDB [dbpos3]> delimiter $$

MariaDB [dbpos3]> create procedure kurangi\_stok(

-> in produk\_id INT,

-> in jmlh\_pesan int)

-> begin

-> declare stok\_baru int;

-> select stok into stok\_baru from produk where id = produk\_id;

-> set stok\_baru = stok\_baru - jmlh\_pesan;

-> update produk set stok = stok\_baru where id = produk\_id;

-> end $$

Query OK, 0 rows affected (0.013 sec)

MariaDB [dbpos3]> 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).

*MariaDB [dbpos3]> delimiter $$*

*MariaDB [dbpos3]> create trigger trig\_kurangi\_stok*

*-> after insert on pesanan\_items for each row*

*-> begin*

*-> DECLARE produk\_id int;*

*-> DECLARE jmlh\_pesan int;*

*-> select new.produk\_id, new.qty into produk\_id, jmlh\_pesan;*

*-> call kurangi\_stok(produk\_id, jmlh\_pesan);*

*-> END $$*

*Query OK, 0 rows affected (0.025 sec)*

*MariaDB [dbpos3]> DELIMITER ;*

*MariaDB [dbpos3]> select \* from pesanan\_items;*

*+----+-----------+------------+------+---------+*

*| id | produk\_id | pesanan\_id | qty | harga |*

*+----+-----------+------------+------+---------+*

*| 1 | 1 | 1 | 1 | 5040000 |*

*| 2 | 3 | 1 | 1 | 4680000 |*

*| 3 | 5 | 2 | 5 | 3500 |*

*| 6 | 5 | 3 | 10 | 3500 |*

*| 7 | 1 | 3 | 1 | 5040000 |*

*| 9 | 5 | 5 | 10 | 3500 |*

*| 10 | 5 | 6 | 20 | 3500 |*

*+----+-----------+------------+------+---------+*

*MariaDB [dbpos3]> select \* from produk;*

*+----+------+------------------+------------+------------+------+----------+-----------------+*

*| id | kode | nama | harga\_beli | harga\_jual | stok | min\_stok | jenis\_produk\_id |*

*+----+------+------------------+------------+------------+------+----------+-----------------+*

*| 1 | TV01 | Televisi 21 inch | 3500000 | 5040000 | 2 | 2 | 1 |*

*| 2 | TV02 | Televisi 40 inch | 5500000 | 7440000 | 4 | 2 | 1 |*

*| 3 | K001 | Kulkas 2 pintu | 3500000 | 4680000 | 6 | 2 | 1 |*

*| 4 | M001 | Meja Makan | 500000 | 600000 | 4 | 3 | 2 |*

*| 5 | TK01 | Teh Kotak | 3000 | 3500 | 6 | 10 | 4 |*

*| 6 | PC01 | PC Desktop HP | 7000000 | 9984000 | 9 | 2 | 5 |*

*| 7 | TB01 | Teh Botol | 2000 | 2500 | 53 | 10 | 4 |*

*| 8 | AC01 | Notebook Acer | 8000000 | 11232000 | 7 | 2 | 5 |*

*| 9 | LN01 | Notebook Lenovo | 9000000 | 12480000 | 9 | 2 | 5 |*

*| 10 | L004 | Laptop HP | 12000000 | 13520000 | 20 | 5 | 5 |*

*+----+------+------------------+------------+------------+------+----------+-----------------+*

*10 rows in set (0.001 sec)*

*MariaDB [dbpos3]> insert into pesanan\_items (produk\_id, pesanan\_id, qty, harga) values*

*-> (5,5,8,70000);*

*Query OK, 1 row affected (0.012 sec)*

*MariaDB [dbpos3]> select \* from produk;*

*+----+------+------------------+------------+------------+------+----------+-----------------+*

*| id | kode | nama | harga\_beli | harga\_jual | stok | min\_stok | jenis\_produk\_id |*

*+----+------+------------------+------------+------------+------+----------+-----------------+*

*| 1 | TV01 | Televisi 21 inch | 3500000 | 5040000 | 2 | 2 | 1 |*

*| 2 | TV02 | Televisi 40 inch | 5500000 | 7440000 | 4 | 2 | 1 |*

*| 3 | K001 | Kulkas 2 pintu | 3500000 | 4680000 | 6 | 2 | 1 |*

*| 4 | M001 | Meja Makan | 500000 | 600000 | 4 | 3 | 2 |*

*| 5 | TK01 | Teh Kotak | 3000 | 3500 | -2 | 10 | 4 |*

*| 6 | PC01 | PC Desktop HP | 7000000 | 9984000 | 9 | 2 | 5 |*

*| 7 | TB01 | Teh Botol | 2000 | 2500 | 53 | 10 | 4 |*

*| 8 | AC01 | Notebook Acer | 8000000 | 11232000 | 7 | 2 | 5 |*

*| 9 | LN01 | Notebook Lenovo | 9000000 | 12480000 | 9 | 2 | 5 |*

*| 10 | L004 | Laptop HP | 12000000 | 13520000 | 20 | 5 | 5 |*

*+----+------+------------------+------------+------------+------+----------+-----------------+*

*10 rows in set (0.000 sec)*