Stored Procedures, Stored Functions dan Trigger

Worksheet 6

Nama : Dewa Gede Andika Andara Putra

https://github.com/ChromaXz/tugas-mysql.git

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*

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

*MariaDB [dbpos\_sib6]> CREATE PROCEDURE pro\_naikan\_harga(*

*-> 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 $$*

*Query OK, 0 rows affected (0.045 sec)*

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;

MariaDB [dbpos\_sib6]> SELECT nama, umur1(tgl\_lahir) AS umur FROM

pelanggan;

+--------------------+------+

| nama | umur |

+--------------------+------+

| Agung Sedayu Group | 14 |

| Pandan Wangi | 74 |

| Sekar Mirah | 41 |

| Swandaru Geni | 43 |

| Pradabashu | 39 |

| Gayatri Dwi | 37 |

| Dewi Gyat | 36 |

| Andre Haru | 34 |

| Ahmad Hasan | 32 |

| Cassanndra | 34 |

| Andi Wijaya | 1 |

| Ira | 1 |

+--------------------+------+

12 rows in set (0.014 sec)

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

DELIMITER $$

CREATE FUNCTION kategori\_harga(harga DOUBLE)

RETURNS VARCHAR(20)

BEGIN

DECLARE kategori VARCHAR(20);

IF harga BETWEEN 0 AND 500000 THEN

SET kategori = 'murah';

ELSEIF harga BETWEEN 500001 AND 3000000 THEN

SET kategori = 'sedang';

ELSEIF harga BETWEEN 3000001 AND 10000000 THEN

SET kategori = 'mahal';

ELSE

SET kategori = 'sangat mahal';

END IF;

RETURN kategori;

END$$

DELIMITER ;

MariaDB [dbpos\_sib6]> SELECT nama, harga\_jual

, kategori\_harga(harga\_jual) AS kategori\_harg

a FROM produk;

+-------------------+------------+----------------+

| nama | harga\_jual | kategori\_harga |

+-------------------+------------+----------------+

| Televisi 21 inchs | 50500000 | sangat mahal |

| Televisi 40 inch | 7440000 | mahal |

| Kulkas 2 pintu | 4680000 | mahal |

| Meja Makan | 600000 | sedang |

| Teh Kotak | 3850 | murah |

| PC Desktop HP | 9984000 | mahal |

| Teh Botol | 2750 | murah |

| Notebook Acer S | 11232000 | sangat mahal |

| Notebook Lenovo | 12480000 | sangat mahal |

| Laptop Lenovo | 16000000 | sangat mahal |

| Kopi | 33000 | murah |

| Teh Sosro 2 | 15000 | murah |

| Laptop Asus | 5000000 | mahal |

| Televisi 22 inc` | 50500000 | sangat mahal |

| Televisi 23 inc | 50500000 | sangat mahal |

| Televisi 24 inc | 50500000 | sangat mahal |

| Televisi 25 inc | 50500000 | sangat mahal |

| Televisi 27 inc | 50500000 | sangat mahal |

| Televisi 28 inc | 50500000 | sangat mahal |

| Televisi 29 inc | 50500000 | sangat mahal |

| Teh Pucuk | 5500 | murah |

| Teh Pucuk2 | 5500 | murah |

+-------------------+------------+----------------+

22 rows in set (0.020 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;

MariaDB [dbpos\_sib6]> SELECT \* FROM pesanan;

+----+------------+---------+--------------+

| id | tanggal | total | pelanggan\_id |

+----+------------+---------+--------------+

| 1 | 2015-11-04 | 9720000 | 1 |

| 2 | 2015-11-04 | 17500 | 3 |

| 3 | 2015-11-04 | 0 | 6 |

| 4 | 2015-11-04 | 0 | 7 |

| 5 | 2015-11-04 | 0 | 10 |

| 6 | 2015-11-04 | 0 | 2 |

| 7 | 2015-11-04 | 0 | 5 |

| 8 | 2015-11-04 | 0 | 4 |

| 9 | 2015-11-04 | 0 | 8 |

| 10 | 2015-11-04 | 0 | 9 |

| 11 | 2015-11-04 | 30000 | 9 |

| 12 | 2020-11-04 | 40000 | 9 |

+----+------------+---------+--------------+

12 rows in set (0.001 sec)

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`*