Stored Procedures, Stored Functions dan Trigger

Worksheet 6

Nama : Toriqul Firdaus Tear Pangeran

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 [dbpos1]> 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 $$*

*Query OK, 0 rows affected (0.006 sec)*

*MariaDB [dbpos1]> DELIMITER ;*

*MariaDB [dbpos1]> SELECT nama,harga\_jual,jenis\_produk\_id from produk;*

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

*| nama | harga\_jual | jenis\_produk\_id |*

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

*| Televisi 21 inch | 5040000 | 1 |*

*| Televisi 40 inch | 7440000 | 1 |*

*| Kulkas 2 pintu | 4680000 | 1 |*

*| Meja Makan | 600000 | 2 |*

*| Teh Kotak | 3500 | 4 |*

*| PC Desktop HP | 9600000 | 5 |*

*| Teh Botol | 2500 | 4 |*

*| Notebook Acer | 10800000 | 5 |*

*| Notebook Lenovo | 12000000 | 5 |*

*| Laptop HP | 13000000 | 5 |*

*| Asus Rog | 5000000 | 2 |*

*| Asus Rog Strix | 10000000 | 2 |*

*| Asus Rog zephyrus | 15000000 | 1 |*

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

*13 rows in set (0.002 sec)*

*MariaDB [dbpos1]> CALL pro\_naik(5, 4);*

*Query OK, 4 rows affected (0.004 sec)*

*MariaDB [dbpos1]> SELECT nama,harga\_jual,jenis\_produk\_id from produk;*

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

*| nama | harga\_jual | jenis\_produk\_id |*

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

*| Televisi 21 inch | 5040000 | 1 |*

*| Televisi 40 inch | 7440000 | 1 |*

*| Kulkas 2 pintu | 4680000 | 1 |*

*| Meja Makan | 600000 | 2 |*

*| Teh Kotak | 3500 | 4 |*

*| PC Desktop HP | 9984000 | 5 |*

*| Teh Botol | 2500 | 4 |*

*| Notebook Acer | 11232000 | 5 |*

*| Notebook Lenovo | 12480000 | 5 |*

*| Laptop HP | 13520000 | 5 |*

*| Asus Rog | 5000000 | 2 |*

*| Asus Rog Strix | 10000000 | 2 |*

*| Asus Rog zephyrus | 15000000 | 1 |*

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

*13 rows in set (0.005 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 [dbpos1]> DELIMITER $$

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 (0.019 sec)

MariaDB [dbpos1]> DELIMITER ;

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

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

| kode | nama | jk | umur |

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

| C001 | Agung Sedayu | L | 13 |

| C002 | Pandan Wangi | P | 19 |

| C003 | Sekar Mirah | P | 40 |

| C004 | Swandaru Geni | L | 42 |

| C005 | Pradabashu | L | 38 |

| C006 | Gayatri Dwi | P | 36 |

| C007 | Dewi Gyat | P | 25 |

| C008 | Andre Haru | L | 24 |

| C009 | Ahmad Hasan | L | 31 |

| C010 | Cassanndra | P | 33 |

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

10 rows in set (0.001 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

*CREATE FUNCTION ...*

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 [dbpos1]> 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 | 2015-11-14 | 30000 | 9 |

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

11 rows in set (0.002 sec)

Didalam table pembayaran tambahkan kolom status\_pembayaran

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

MariaDB [dbpos1]> ALTER TABLE pembayaran ADD status\_pembayaran varchar(25);

Query OK, 0 rows affected (0.045 sec)

Records: 0 Duplicates: 0 Warnings: 0

2. Dilanjutkan dengan proses pembayaran di table pembayaran

MariaDB [dbpos1]> 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;

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

SET NEW.status\_pembayaran = 'Lunas';

END IF;

END $$

Query OK, 0 rows affected (0.022 sec)

MariaDB [dbpos1]> DELIMITER ;

// Menambahkan data pada tabel pembayaran

MariaDB [dbpos1]> INSERT INTO pembayaran (nokuitansi, tanggal, jumlah, ke, pesanan\_id, status\_pembayaran) VALUES

-> ('KWI002','2023-04-04', 200000, 2, 2, '');

MariaDB [dbpos1]> select \* from pembayaran;

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

| id | nokuitansi | tanggal | jumlah | ke | pesanan\_id | status\_pembayaran |

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

| 7 | MD004 | 2023-10-10 | 15000 | 1 | 2 | |

| 9 | KWI001 | 2023-03-03 | 200000 | 1 | 1 | |

| 10 | KWI002 | 2023-04-04 | 200000 | 2 | 2 | Lunas |

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

3 rows in set (0.001 sec)

MariaDB [dbpos1]>

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

MariaDB [dbpos1]> CREATE PROCEDURE kurangi\_stok(IN produk\_id INT, IN jumlah\_pesanan INT)

-> BEGIN

-> DECLARE stok\_produk INT;

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

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

-> IF stok\_produk < 0 THEN

-> SET stok\_produk = 0;

-> END IF;

-> UPDATE produk SET stok = stok\_produk WHERE id = produk\_id;

-> END $$

Query OK, 0 rows affected (0.011 sec)

MariaDB [dbpos1]> select \* from produk;

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

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

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

| 1 | TV01 | Televisi 21 inch | 3500000 | 5040000 | 10 | 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 |

| 11 | A001 | Asus Rog | 4000000 | 5000000 | 15 | 2 | 2 |

| 12 | A002 | Asus Rog Strix | 8000000 | 10000000 | 4 | 2 | 2 |

| 13 | A003 | Asus Rog zephyrus | 10000000 | 15000000 | 3 | 1 | 1 |

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

13 rows in set (0.001 sec)

MariaDB [dbpos1]> call kurangi\_stok(1, 2);

Query OK, 2 rows affected (0.004 sec)

MariaDB [dbpos1]> select \* from produk;

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

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

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

| 1 | TV01 | Televisi 21 inch | 3500000 | 5040000 | 8 | 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 |

| 11 | A001 | Asus Rog | 4000000 | 5000000 | 15 | 2 | 2 |

| 12 | A002 | Asus Rog Strix | 8000000 | 10000000 | 4 | 2 | 2 |

| 13 | A003 | Asus Rog zephyrus | 10000000 | 15000000 | 3 | 1 | 1 |

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

13 rows in set (0.001 sec)

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 ...*

MariaDB [dbpos1]> DELIMITER $$

MariaDB [dbpos1]> CREATE TRIGGER trig\_kurangi\_stok

-> AFTER INSERT ON pesanan\_items FOR EACH ROW

-> BEGIN

-> DECLARE produk\_id INT;

-> DECLARE jumlah\_pesanan INT;

-> SELECT NEW.produk\_id, NEW.qty INTO produk\_id, jumlah\_pesanan;

-> CALL kurangi\_stok(produk\_id, jumlah\_pesanan);

-> END $$

Query OK, 0 rows affected (0.027 sec)

MariaDB [dbpos1]> DELIMITER ;

MariaDB [dbpos1]> select \* from produk;

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

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

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

| 1 | TV01 | Televisi 21 inch | 3500000 | 5040000 | 8 | 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 |

| 11 | A001 | Asus Rog | 4000000 | 5000000 | 15 | 2 | 2 |

| 12 | A002 | Asus Rog Strix | 8000000 | 10000000 | 4 | 2 | 2 |

| 13 | A003 | Asus Rog zephyrus | 10000000 | 15000000 | 3 | 1 | 1 |

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

13 rows in set (0.001 sec)

MariaDB [dbpos1]> 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 |

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

7 rows in set (0.009 sec)

MariaDB [dbpos1]> INSERT INTO pesanan\_items (produk\_id, pesanan\_id, qty, harga) VALUES

-> (1, 1, 2, 300000);

Query OK, 1 row affected (0.007 sec)

MariaDB [dbpos1]> 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 |

| 11 | 1 | 1 | 2 | 300000 |

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

8 rows in set (0.001 sec)

MariaDB [dbpos1]> select \* from produk;

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

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

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

| 1 | TV01 | Televisi 21 inch | 3500000 | 5040000 | 6 | 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 |

| 11 | A001 | Asus Rog | 4000000 | 5000000 | 15 | 2 | 2 |

| 12 | A002 | Asus Rog Strix | 8000000 | 10000000 | 4 | 2 | 2 |

| 13 | A003 | Asus Rog zephyrus | 10000000 | 15000000 | 3 | 1 | 1 |

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

13 rows in set (0.001 sec)