TUGAS PERTEMUAN 5

MUHAMMAD ILHAM

Link Github : https://github.com/ilhaaam24/tugas-database-nfa

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.

   SQL :

```
151     DELIMITER $$
152
153     CREATE PROCEDURE pro_naikan_harga(
154         IN p_jenis_produk_id INT,
155         IN p_persentase_kenaikan DECIMAL(5,2)
156     )
157     BEGIN
158         UPDATE produk
159         SET harga_jual = harga_jual + (harga_jual * p_persentase_kenaikan / 100)
160         WHERE jenis_produk_id = p_jenis_produk_id;
161     END $$
162
163     DELIMITER ;
164
165     CALL pro_naikan_harga(1, 4);
```

   Hasil:

   Sebelum manggil procedure:

id	kode	nama	harga_beli	harga_jual	stok	min_stok	jenis_produk_id
1	P001	Laptop Asus	7000000	9193600	10	2	1
2	P002	Kemeja Batik	120000	175000	50	5	2
3	P003	Nasi Kotak	20000	30000	100	10	3
4	P004	Air Mineral 600ml	2000	5000	200	20	4
5	P005	Jam Tangan Casio	250000	400000	15	3	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

   Sesudah manggil procedure:

id	kode	nama	harga_beli	harga_jual	stok	min_stok	jenis_produk_id
1	P001	Laptop Asus	7000000	9561344	10	2	1
2	P002	Kemeja Batik	120000	175000	50	5	2
3	P003	Nasi Kotak	20000	30000	100	10	3
4	P004	Air Mineral 600ml	2000	5000	200	20	4
5	P005	Jam Tangan Casio	250000	400000	15	3	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. 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.

   SQL :

```
169     DELIMITER $$
170
171     CREATE FUNCTION umur(tanggal_lahir DATE)
172     RETURNS INT
173     DETERMINISTIC
174     BEGIN
175         DECLARE hasil INT;
176         SET hasil = TIMESTAMPDIFF(YEAR, tanggal_lahir, CURDATE());
177
178         RETURN hasil;
179     END$$
180
181     DELIMITER ;
182
183     SELECT umur('2003-05-10') AS usia;
```

   HASIL :

usia
22

3. 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

   SQL :

```
DELIMITER $$

CREATE FUNCTION kategori_harga(harga DOUBLE)
RETURNS VARCHAR(20)
DETERMINISTIC
BEGIN
    DECLARE hasil VARCHAR(20);

    IF harga >= 0 AND harga <= 500000 THEN
        SET hasil = 'Murah';
    ELSEIF harga > 500000 AND harga <= 3000000 THEN
        SET hasil = 'Sedang';
    ELSEIF harga > 3000000 AND harga <= 10000000 THEN
        SET hasil = 'Mahal';
    ELSEIF harga > 10000000 THEN
        SET hasil = 'Sangat Mahal';
    ELSE
        SET hasil = 'Tidak Valid'; -- jika harga negatif
    END IF;

    RETURN hasil;
END$$

DELIMITER ;
```

HASIL :

```
SELECT kategori_harga(15000000) AS kategori;  -- Sangat Maha
```

kategori
Sangat Mahal

SOAL 6.2

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;
ALTER TABLE pembayaran ADD status_pembayaran varchar(25);
```

   2) Dilanjutkan dengan proses pembayaran di table pembayaran

```
DELIMITER $$
CREATE TRIGGER cek_pembayaran BEFORE INSERT ON pembayaran
FOR EACH ROW
BEGIN
DECLARE total_bayar DECIMAL(10, 2);
DECLARE total_pesanan DECIMAL(10, 2);
SELECT SUM(jumlah) INTO total_bayar FROM pembayaran WHERE pesanan_id = NEW.pesanan_id;
SELECT total INTO total_pesanan FROM pesanan WHERE id = NEW.pesanan_id;
IF total_bayar + NEW.jumlah >= total_pesanan THEN
SET NEW.status_pembayaran = 'Lunas';
END IF;
END $$
DELIMITER ;

select * from pembayaran;
INSERT INTO pembayaran (nokuitansi, tanggal, jumlah, ke, pesanan_id)
VALUES ('K006', '2023-03-03', 200000, 1, 1);
```

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

id	nokuitansi	tanggal	jumlah	ke	pesanan_id	status_pembayaran
3	K003	2025-09-03	300000	1	3	NULL
4	K004	2025-09-04	600000	1	4	NULL
5	K005	2025-09-05	450000	1	5	NULL
6	K006	2023-03-03	200000	1	1	Lunas
NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Buatlah Stored Procedure dengan nama **kurangi_stok** untuk mengurangi stok produk. Stok berkurang sesuai dengan jumlah pesanan produk.

```
DELIMITER $$

CREATE PROCEDURE kurangi_stok(
    IN p_produk_id INT,
    IN p_jumlah INT
)
BEGIN
    UPDATE produk
    SET stok = stok - p_jumlah
    WHERE id = p_produk_id;

    IF (SELECT stok FROM produk WHERE id = p_produk_id) < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Stok produk tidak mencukupi!';
    END IF;
END$$
DELIMITER ;
```

3. 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 2).

   Trigger ini aktif setelah trigger **after_pesanan_items_insert** (trigger pada contoh 3)

```
DELIMITER $$

CREATE TRIGGER trig_kurangi_stok
AFTER INSERT ON pesanan_items
FOR EACH ROW
BEGIN
    -- Memanggil stored procedure kurangi_stok
    CALL kurangi_stok(NEW.produk_id, NEW.qty);
END$$

DELIMITER ;
```