

LATIHAN

Buatlah tabel seperti berikut:

BUAT TABEL Barang Pesanan

```
CREATE TABLE OrderItems
(
  OrderID INT NOT NULL,
  Item VARCHAR(20) NOT NULL,
  Quantity SMALLINT NOT NULL,
  PRIMARY KEY(OrderID, Item)
);
```

```
INSERT INTO OrderItems (OrderID, Item, Quantity)
VALUES
(1, 'M8 Bolt', 100),
(2, 'M8 Nut', 100),
(3, 'M8 Washer', 200);
```

Deklarasikan kursor untuk mengulangi semua OrderItems dan menghitung jumlah total untuk setiap pesanan.

```
DECLARE OrderItemCursor CURSOR FAST_FORWARD
FOR
SELECT OrderID,
       SUM(Quantity) AS NumItems
FROM OrderItems
GROUP BY OrderID
ORDER BY OrderID;

DECLARE @OrderID INT, @NumItems INT;

-- Instantiate the cursor and loop through all orders.
OPEN OrderItemCursor;

FETCH NEXT FROM OrderItemCursor
INTO @OrderID, @NumItems

WHILE @@Fetch_Status = 0
BEGIN;
  IF @NumItems > 100
    PRINT 'EXECUTING LogLargeOrder - '
      + CAST(@OrderID AS VARCHAR(5))
      + ' ' + CAST(@NumItems AS VARCHAR(5));
```

```
ELSE
    PRINT 'EXECUTING LogSmallOrder - '
    + CAST(@OrderID AS VARCHAR(5))
    + ' ' + CAST(@NumItems AS VARCHAR(5));

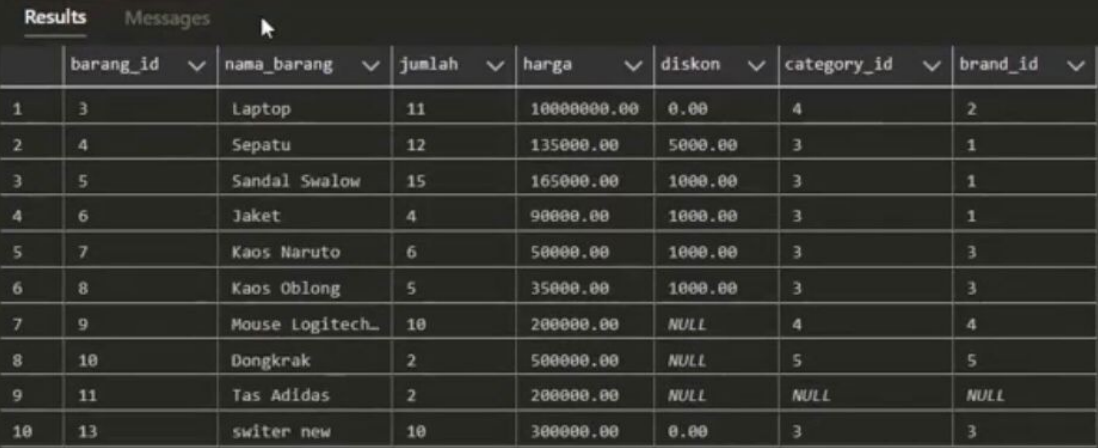
FETCH NEXT FROM OrderItemCursor
INTO @OrderID, @NumItems;
END;

-- Close and deallocate the cursor.
CLOSE OrderItemCursor;
DEALLOCATE OrderItemCursor;
```

Untuk contoh sebelumnya, hasilnya terlihat seperti yang ditunjukkan berikut ini.

```
EXECUTING LogSmallOrder - 1 100
EXECUTING LogSmallOrder - 2 100
EXECUTING LogLargeOrder - 3 200
```

1. Buatlah database seperti output berikut!



	barang_id	nama_barang	jumlah	harga	diskon	category_id	brand_id
1	3	Laptop	11	10000000.00	0.00	4	2
2	4	Sepatu	12	135000.00	5000.00	3	1
3	5	Sandal Swallow	15	165000.00	1000.00	3	1
4	6	Jaket	4	90000.00	1000.00	3	1
5	7	Kaos Naruto	6	50000.00	1000.00	3	3
6	8	Kaos Oblong	5	35000.00	1000.00	3	3
7	9	Mouse Logitech	10	200000.00	NULL	4	4
8	10	Dongkrak	2	500000.00	NULL	5	5
9	11	Tas Adidas	2	200000.00	NULL	NULL	NULL
10	13	switer new	10	300000.00	0.00	3	3

2. Buatlah flow control menggunakan begin-end yang mana jika ada harga barang yang melebihi 5.000.000 maka akan keluar output 'terdapat barang bernama'+@nama_barang, namun jika rowcount = 0, maka print 'tidak ada produk yang dipilih!'
3. Buatlah if statement yang mengeluarkan jumlah harga yang di join pada tahu 2022 dan jika penjualan lebih besar dari 300.000 maka akan mnegeluarkan output "Penjualan lebih dari 300.000"
4. Buatlah looping menggunakan while dengan mengeluarkan output nama barang dengan id 5 sampai 10!

TRIGGER

1. Create Table BARANG

```
create table
BARANG (
BAR_ID int not null,
BAR_NAMA varchar(255) not null,
BAR_STOCK
int null default
0,
constraint PK_BARANG primary key (BAR_ID)
)
```

```
create table
PEMBELIAN (
PEM_ID int not
null, BAR_ID
int null ,
PEM_JUMLAH
int null ,
constraint PK_PEMBELIAN primary key (PEM_ID)
)
```

```
alter table PEMBELIAN
add constraint FK_PEMBELIA_RELATION_BARANG foreign key
(BAR_ID) references BARANG (BAR_ID)
```

```
INSERT INTO BARANG (BAR_ID,BAR_NAMA) VALUES
(1,'AQUA'); INSERT INTO BARANG (BAR_ID,BAR_NAMA)
VALUES (2,'TOTAL'); INSERT INTO BARANG
(BAR_ID,BAR_NAMA) VALUES (3,'AQUADES');
```

Kemudian saatnya kita buat trigger sehingga ketika kita menambahkan data di tabel pembelian dengan jumlah pembelian barang tertentu, maka stock di tabel barang akan bertambah sesuai dengan barang yg dibeli

syntac trigger tersebut adalah sebagai berikut :

```
Create trigger tambahStockbarang          on
PEMBELIAN after insert as

update barang set bar_stock = bar_stock + i.PEM_JUMLAH from BARANG join
inserted i on i.BAR_ID = barang.BAR_ID
```

perintah diatas adalah membuat trigger dengan nama **tambahStockBarang** dimana trigger tersebut akan terpicu jika ada perubahan di tabel pembelian

kita coba masukkan kode berikut:

```
INSERT INTO PEMBELIAN (PEM_ID, BAR_ID, PEM_JUMLAH)
VALUES (1,1,4); INSERT INTO PEMBELIAN (PEM_ID, BAR_ID,
PEM_JUMLAH) VALUES (2,3,2); INSERT INTO PEMBELIAN
(PEM_ID, BAR_ID, PEM_JUMLAH) VALUES (3,1,1); INSERT
INTO PEMBELIAN (PEM_ID, BAR_ID, PEM_JUMLAH)
VALUES (4,1,2);
```

TUGAS

1. Buatlah Tabel Penjualan dengan field sebagai berikut : ID_PENJUALAN, BAR_ID, QTY_JUAL
2. Buatlah trigger pada Tabel Penjualan jika ada barang yang terjual maka stock di tabel barang akan berkurang secara otomatis
3. Buatlah insert data di tabel pembelian dengan menggunakan stored procedure
4. **Buatlah insert data di tabel penjualan dengan menggunakan stored procedure**