Alfi Futuhi

Assignment 3

8 April 2021

1. Membuat Tabel Transaction

--Membuat SP Insert Data ke Tabel Transaction

CREATE PROCEDURE SP\_InsertTB\_T\_Transaction

@OrderDate date

AS

BEGIN

INSERT INTO TB\_T\_Transaction(OrderDate)

VALUES (@OrderDate)

END;

--Membuat Tabel Transaction

CREATE TABLE TB\_T\_Transaction(

Id int IDENTITY(101,1) PRIMARY KEY,

OrderDate date

)

--Memasukkan Tabel Transaction dengan SP

EXECUTE SP\_InsertTB\_T\_Transaction '2012-10-05';

EXECUTE SP\_InsertTB\_T\_Transaction '2012-10-15';

EXECUTE SP\_InsertTB\_T\_Transaction '2012-11-05';

EXECUTE SP\_InsertTB\_T\_Transaction '2012-11-20';

EXECUTE SP\_InsertTB\_T\_Transaction '2012-11-30';

--Menampilkan Tabel Transaction dengan SP

CREATE PROC SP\_SelectTB\_T\_Transaction

AS

BEGIN

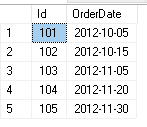
SELECT \* from TB\_T\_Transaction

END

GO

EXEC SP\_SelectTB\_T\_Transaction

OUTPUT:



1. Membuat Tabel Supplier

--Membuat Tabel

CREATE TABLE TB\_S\_Supplier

(

ID varchar(50) PRIMARY KEY,

Name varchar(50),

JoinDate date

);

--Membuat Sequence agar bisa auto increment

CREATE SEQUENCE SQ\_SupplierOrderNumber

AS INT

START WITH 1

INCREMENT BY 1;

--DROP SEQUENCE SQ\_OrderNumber;

GO

--Membuat SP untuk Insert data ke Tabel Supplier

CREATE PROCEDURE SP\_InsertTB\_S\_Supplier

AS

BEGIN

INSERT INTO TB\_S\_Supplier(ID, [Name], JoinDate)

VALUES

(NEXT VALUE FOR SQ\_SupplierOrderNumber, 'PT. Synnex Metrodata Indonesia', '2018-01-15'),

(NEXT VALUE FOR SQ\_SupplierOrderNumber, 'PT. Metrodata Indonesia', '2018-01-16'),

(NEXT VALUE FOR SQ\_SupplierOrderNumber, 'PT. Mitra Informatika Indonesia', '2018-01-16'),

(NEXT VALUE FOR SQ\_SupplierOrderNumber, 'PT. Metro Mobile', '2018-01-15');

END;

GO

--Membuat Trigger saat setelah insert data ke tabel Supplier, agar auto increment bisa dipasangkan dengan

--kata 'SP0'

CREATE TRIGGER TR\_InsertTB\_S\_Supplier

ON TB\_S\_Supplier

AFTER INSERT

AS

BEGIN

UPDATE TB\_S\_Supplier

SET

ID = 'SP' + RIGHT('0' + CAST(i.ID AS varchar),2)

FROM

inserted i

WHERE

i.ID = TB\_S\_Supplier.ID

END;

GO

EXEC SP\_InsertTB\_S\_Supplier;

GO

--Membuat SP untuk Menampilkan Tabel Supplier

CREATE PROC SP\_SelectTB\_S\_Supplier

AS

BEGIN

SELECT \* FROM TB\_S\_Supplier

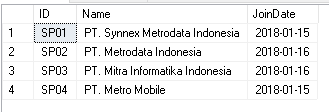
END;

GO

EXEC SP\_SelectTB\_S\_Supplier;

GO

OUTPUT:



1. Membuat Tabel Item

--Membuat tabel Item

CREATE TABLE TB\_I\_Item

(

ID varchar(50) NOT NULL PRIMARY KEY,

Name varchar(50),

Price int,

Stock int,

Supplier\_ID varchar(50) FOREIGN KEY REFERENCES TB\_S\_Supplier(ID)

);

--Membuat sequence untuk auto increment

CREATE SEQUENCE SQ\_ItemOrderNumber

AS INT

START WITH 1

INCREMENT BY 1;

GO

--Membuat SP untuk Insert data ke tabel Item

CREATE PROCEDURE SP\_InsertTB\_I\_Item

AS

BEGIN

INSERT INTO TB\_I\_Item (ID, Name, Price, Stock, Supplier\_ID) VALUES

(NEXT VALUE FOR SQ\_ItemOrderNumber, 'Developer .Net', 6000000, 11, 'SP03'),

(NEXT VALUE FOR SQ\_ItemOrderNumber, 'Developer Java', 6000000, 10, 'SP02'),

(NEXT VALUE FOR SQ\_ItemOrderNumber, 'Developer C#', 5500000, 10, 'SP04'),

(NEXT VALUE FOR SQ\_ItemOrderNumber, 'Developer Python', 5500000, 10, 'SP02');

END;

GO

--Membuat Trigger untuk membuat auto increment dipasangkan dengan kata 'B0'

CREATE TRIGGER TR\_UpdateTB\_I\_Item

ON TB\_I\_Item

AFTER INSERT

AS

BEGIN

UPDATE TB\_I\_Item

SET

ID = 'B' + RIGHT('0' + CAST(i.ID AS varchar),2)

FROM

inserted i --virtual temporary table dalam trigger

WHERE

i.ID = TB\_I\_Item.ID

END;

GO

--Membuat SP untuk menampilkan Tabel Item

CREATE PROC SP\_SelectTB\_I\_Item

AS

BEGIN

SELECT \* FROM TB\_I\_Item

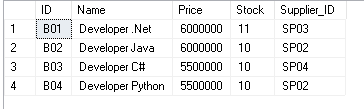
END;

GO

EXEC SP\_SelectTB\_I\_Item;

GO

OUTPUT:



1. Membuat Tabel TransactionItem

--Membuat Tabel TransactionItem

CREATE TABLE TB\_TI\_TransactionItem

(

ID int IDENTITY(1,1) NOT NULL PRIMARY KEY,

Quantity int,

Transaction\_ID int FOREIGN KEY REFERENCES TB\_T\_Transaction(ID),

Item\_ID varchar(50) FOREIGN KEY REFERENCES TB\_I\_Item(ID)

);

GO

--membuat SP untuk memasukkan Data ke Tabel TransactionItem

CREATE PROCEDURE SP\_InsertTB\_TI\_TransactionItem (

@Quantity AS DECIMAL,

@Transaction\_ID AS DECIMAL,

@Item\_ID AS VARCHAR(50)

)

AS

BEGIN

--Kondisi dimana data bisa dimasukkan HANYA KALAU quantity minimal 1

IF @Quantity > 0

BEGIN

INSERT INTO TB\_TI\_TransactionItem (Quantity, Transaction\_ID, Item\_ID) VALUES

(@Quantity, @Transaction\_ID, @Item\_ID);

END

ELSE

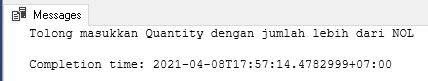
BEGIN

--Kalau quantity tidak bernilai minimal 1, akan muncul tulisan ini

PRINT 'Tolong masukkan Quantity dengan jumlah lebih dari NOL';

END

END;



GO

--Memasukkan Data

EXECUTE SP\_InsertTB\_TI\_TransactionItem 10,'101','B01';

EXECUTE SP\_InsertTB\_TI\_TransactionItem 5,'101','B02';

EXECUTE SP\_InsertTB\_TI\_TransactionItem 4,'102','B04';

EXECUTE SP\_InsertTB\_TI\_TransactionItem 5,'103','B03';

EXECUTE SP\_InsertTB\_TI\_TransactionItem 10,'104','B04';

EXECUTE SP\_InsertTB\_TI\_TransactionItem 3,'105','B02';

--Pengecekan kalau input dibawah 1

EXECUTE SP\_InsertTB\_TI\_TransactionItem 0,103,'B02';

--Membuat SP untuk menampilkan Data TransactionItem

CREATE PROC SP\_SelectTB\_TI\_TransactionItem

AS

BEGIN

SELECT \* FROM TB\_TI\_TransactionItem

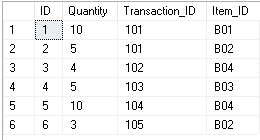
END;

GO

EXEC SP\_SELECTTB\_TI\_TransactionItem;

GO

OUTPUT:



PROBLEM SOAL

SOAL 1A

------------------1a--------------------------

--Membuat View

CREATE VIEW View\_TB\_I\_ItemTB\_S\_Supplier

AS

SELECT

i.ID AS 'Kode Barang',

i.[Name] AS 'Nama Barang',

s.Name AS 'Supplier'

FROM

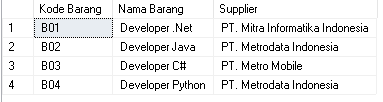
TB\_I\_Item i

INNER JOIN TB\_S\_Supplier s ON i.Supplier\_ID = s.ID;

GO

SELECT \* FROM View\_TB\_I\_ITemTB\_S\_Supplier

OUTPUT:



SOAL 1B

---------------1b----------------------

CREATE VIEW View\_TB\_I\_ItemTB\_S\_SupplierTB\_TI\_TransactionItem

AS

SELECT

TI.Transaction\_ID AS 'Nomor Faktur',

s.Name AS 'Supplier', i.Name AS 'Nama Barang',

i.Price AS 'Price'

FROM

TB\_I\_Item i

INNER JOIN TB\_S\_Supplier s ON i.Supplier\_ID = s.ID

INNER JOIN TB\_TI\_TransactionItem ti ON i.ID = ti.Item\_ID

WHERE

TI.Transaction\_ID = 101;

GO

SELECT \* FROM View\_TB\_I\_ItemTB\_S\_SupplierTB\_TI\_TransactionItem

OUTPUT:



SOAL 1C

--------------------1c------------------------------

CREATE VIEW View\_TB\_I\_ItemTB\_TI\_TransactionItem

AS

SELECT

ti.Transaction\_ID AS 'Nomor Faktur',

i.[Name] AS 'Nama Barang',

ti.Quantity,

FORMAT(i.Price, 'N') AS 'Harga',

FORMAT((Ti.Quantity \* i.Price),'N') as 'Total Harga'

FROM

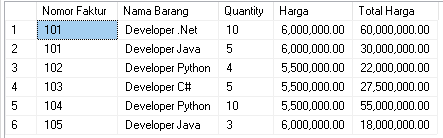
TB\_I\_Item i

INNER JOIN TB\_TI\_TransactionItem ti ON i.ID = ti.Item\_ID;

GO

SELECT \* FROM View\_TB\_I\_ItemTB\_TI\_TransactionItem;

OUTPUT:



SOAL 2

CREATE VIEW View\_TB\_TI\_TransactionItemTB\_T\_TransactionTB\_I\_Item

AS

SELECT

ti.Transaction\_ID AS 'Nomor Faktur',

FORMAT(SUM(ti.Quantity \* i.Price),'N') AS 'Total Faktur',

t.OrderDate AS 'Tanggal Faktur',

DATEADD(DAY, 14, T.OrderDate) AS 'Jatuh Tempo'

FROM

TB\_TI\_TransactionItem ti

INNER JOIN TB\_T\_Transaction t ON TI.Transaction\_ID = t.ID

INNER JOIN TB\_I\_Item i ON TI.Item\_ID = i.ID

GROUP BY

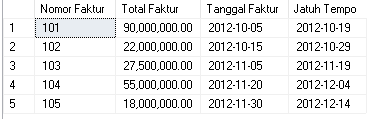
ti.Transaction\_ID,t.OrderDate;

GO

SELECT \* FROM View\_TB\_TI\_TransactionItemTB\_T\_TransactionTB\_I\_Item

OUTPUT:

SOAL 3A



-----------------3a-------------------------

CREATE VIEW View\_TB\_I\_ItemTB\_TI\_TransactionItem3A

AS

SELECT

i.Name,

SUM(ti.Quantity) AS 'Quantity'

FROM

TB\_I\_Item i

INNER JOIN TB\_TI\_TransactionItem ti ON i.ID = ti.Item\_ID

GROUP BY

i.[Name];

GO

SELECT \* FROM View\_TB\_I\_ItemTB\_TI\_TransactionItem3A

OUTPUT:



SOAL 3B

------------------------3b--------------------

CREATE VIEW View\_TB\_TI\_TransactionItemTB\_I\_ItemTB\_T\_Transaction

AS

SELECT

i.Name,

SUM(ti.Quantity) AS 'Quantity'

FROM

TB\_TI\_TransactionItem ti

INNER JOIN TB\_I\_Item i ON i.ID = ti.Item\_ID

INNER JOIN TB\_T\_Transaction t ON t.ID = ti.Transaction\_ID

WHERE

t.OrderDate LIKE '2012-10%'

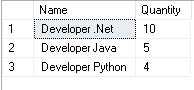
GROUP BY

i.Name;

GO

SELECT \* FROM View\_TB\_TI\_TransactionItemTB\_I\_ItemTB\_T\_Transaction

OUTPUT:



SOAL 4

-----------------4--------------------

CREATE VIEW View\_TB\_TI\_TransactionItemTB\_T\_TransactionTB\_I\_Item4

AS

SELECT

i.[Name],

SUM(CASE WHEN t.OrderDate LIKE '2012-10%' THEN ti.Quantity ELSE 0 END) AS 'October',

SUM(CASE WHEN t.OrderDate LIKE '2012-11%' THEN ti.Quantity ELSE 0 END) AS 'November',

SUM(CASE WHEN t.OrderDate LIKE '2012-12%' THEN ti.Quantity ELSE 0 END) AS 'December'

FROM

TB\_TI\_TransactionItem ti

INNER JOIN TB\_T\_Transaction t ON ti.Transaction\_ID = t.ID

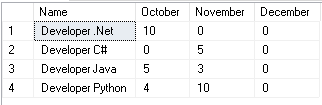
INNER JOIN TB\_I\_Item i ON i.ID = ti.Item\_ID

GROUP BY i.[Name];

GO

SELECT \* FROM View\_TB\_TI\_TransactionItemTB\_T\_TransactionTB\_I\_Item4

OUTPUT:



SOAL 5

CREATE VIEW View\_TB\_TI\_TransactionItemTB\_T\_TransactionTB\_I\_Item5

AS

SELECT

COALESCE(i.[Name], 'TOTAL') AS 'Name',

SUM(CASE WHEN t.OrderDate LIKE '2012-10%' THEN ti.Quantity ELSE 0 END) AS 'October',

SUM(CASE WHEN t.OrderDate LIKE '2012-11%' THEN ti.Quantity ELSE 0 END) AS 'November',

SUM(CASE WHEN t.OrderDate LIKE '2012-12%' THEN ti.Quantity ELSE 0 END) AS 'December'

FROM

TB\_TI\_TransactionItem ti

INNER JOIN TB\_T\_Transaction t ON ti.Transaction\_ID = t.ID

INNER JOIN TB\_I\_Item i ON i.ID = ti.Item\_ID

GROUP BY ROLLUP([Name]);

GO

SELECT \* FROM View\_TB\_TI\_TransactionItemTB\_T\_TransactionTB\_I\_Item5

OUTPUT:

