

Cara Singkat Mempelajari

DATABASE dengan SQL



Kata Pengantar

Kursus Database System mempelajari tentang konsep dan terminologi dalam database seperti arsitektur database, model relasional, SQL, desain database, teknologi dan masalah dalam database environment.

Setelah menempuh kursus ini, peserta diharapkan memiliki pengetahuan dasar sistem database seperti arsitektur, desain database, dan SQL. Dengan adanya bekal tersebut, peserta dapat melanjutkan ke database lanjutan seperti database administrator dan menerapkannya dalam integrasi ke programming.

Daftar Isi

Kata Pengantar.....	1
Daftar Isi.....	2
1. Database Environment.....	3
2. Data Definition Language	4
3. DML Insert, Update, Delete, Select Data	7
4. DML Grouping & Aggregate	8
5. DML Order By & Compute By	9
6. DML String and Date Function	10
7. DML Inner Join, Outer Join, Union	12
8. DML Subquery, Alias Subquery	14
9. DML IN & EXISTS	15
10. DML ANY, SOME, ALL.....	16
11. View	17
12. Data Control Language.....	18
13. Store Procedure	19
14. Cursor	20
15. Trigger	21

1. Database Environment

Database atau basis data adalah koleksi data yang saling berhubungan dan terorganisasi secara rapi.

Database Management System (DBMS) adalah software yang digunakan untuk mengelola database. Beberapa contoh DBMS adalah Microsoft SQL Server, Oracle, dan MySQL.

Database relasional adalah database yang dirancang melalui tabel. Setiap tabel memiliki nama yang unik dan terdiri dari kolom dan baris. Kolom berisi informasi tentang jenis data yang tersimpan dalam tabel.

SQL (Structured Query Language) adalah bahasa yang digunakan untuk melakukan query dalam database. Jenis dari SQL statement yaitu sebagai berikut :

1. DDL (Data Definition Language).
DDL berhubungan dengan pembuatan, penghapusan, dan operasi lain definis database dan objek di dalamnya.
2. DML (Data Manipulation Language).
DML berhubungan dengan operasi manipulasi data dalam objek database.
3. DCL (Data Control Language).
DCL berhubungan dengan operasi untuk mengontrol sistem database.

2. Data Definition Language

Create Table

```
CREATE TABLE <<table_name>>
(
    <<field_name1>> <<data_type>> (<<length>>),
    <<field_name2>> <<data_type>> (<<length>>),
    <<field_name3>> <<data_type>> (<<length>>),
    ...
    PRIMARY KEY (<<field_name>>),
    FOREIGN KEY (<<field_name>>) REFERENCES
<<table_name>>,
    CONSTRAINT <<constraint_name>> CHECK
(<<criteria>>)
)
```

Drop Table

```
DROP TABLE <<table_name>>
```

Alter Table

```
ALTER TABLE <<table_name>>
ADD [ <<column_name>> <<data_type>> | CONSTRAINT
<<constraint_name>> CHECK (<<criteria>>) | PRIMARY KEY
(<<primary_key_name>>) ]
```

```
ALTER TABLE <<table_name>>
DROP [ COLUMN <<column_name>> | CONSTRAINT
<<constraint_name>> ]
```

CREATE TABLE digunakan untuk membuat tabel.

DROP TABLE digunakan untuk menghapus tabel.

ALTER TABLE digunakan untuk memperbaharui tabel.

Jenis-jenis tipe data dalam tabel yang sering digunakan adalah sebagai berikut :

Tipe Data	Keterangan
Char	Karakter atau kumpulan karakter, jumlah memori yang digunakan sesuai yang dialokasikan.
Varchar	Sama seperti char, akan tetapi jumlah memori yang digunakan sesuai yang digunakan.
Date	Tanggal
Numeric	Bilangan

Primary key adalah key yang mengidentifikasi setiap record secara unik dari tabel database.

Foreign key adalah primary key dari tabel lain untuk memberikan hubungan antar tabel.

Constraint adalah batasan yang digunakan untuk memberikan ketentuan pada tabel.

Jenis-jenis constraint adalah sebagai berikut :

1. NOT NULL

Constraint untuk memberikan aturan kolom tersebut tidak menerima nilai NULL.

2. UNIQUE

Constraint untuk mengidentifikasi keunikan record dalam tabel.

3. DEFAULT

Constraint untuk memberikan nilai default jika kolom tersebut tidak diisi.

4. **CHECK**
Constraint untuk membatasi nilai yang boleh dimasukkan ke dalam sebuah kolom dalam tabel.
5. **PRIMARY KEY**
Constraint untuk mengidentifikasi keunikan setiap record dalam tabel.
6. **FOREIGN KEY**
Constraint untuk menghubungkan ke tabel lain melalui primary key tabel tersebut.

Referential Integrity digunakan untuk menjaga konsistensi isi data antara 2 tabel yang saling berhubungan melalui FOREIGN KEY. Untuk menjaga referential integrity, saat update dan delete dapat memberikan pilihan sebagai berikut :

1. **NO ACTION**
Jika melakukan update atau delete pada primary key suatu tabel, maka tabel lain tidak melakukan reaksi apapun.
2. **CASCADE**
Saat melakukan update atau delete pada primary key, maka foreign key di tabel lain ikut terupdate sesuai primary key.
3. **SET NULL**
Saat melakukan update atau delete pada primary key, maka foreign key di tabel lain akan berisi NULL.
4. **SET DEFAULT**
Saat melakukan update atau delete pada primary key, maka foreign key di tabel lain akan berisi nilai default sesuai yang telah ditentukan.

3. DML Insert, Update, Delete, Select Data

Insert Data

```
INSERT INTO <<table_name>> (<<column_1>>,
<<column_2>>, ...)
VALUES (<<value_1>>, <<value_2>>, ...)
```

Update Data

```
UPDATE <<table_name>> SET <<column_1>> = <<value_1>>,
<<column_2>> = <<value_2>>, ...
WHERE <<condition>>
```

Delete Data

```
DELETE FROM <<table_name>>
WHERE <<condition>>
```

Select Data

```
SELECT <<column_1>>, <<column_2>>, ...
FROM <<table_name>>
```

```
SELECT *
FROM <<table_name>>
```


4. DML Grouping & Aggregate

Aggregate adalah sarana untuk melakukan perhitungan pada saat pengelompokan data.

Grouping digunakan untuk pengelompokan data, grouping harus digunakan saat adanya aggregate.

Aggregate function :

1. COUNT : menghitung jumlah data.
2. MAX : mencari data terbesar.
3. MIN : mencari data terkecil.
4. SUM : menjumlahkan semua data.
5. AVG : menghitung rata-rata dari data.

Grouping & Aggregate

```
SELECT      <<column_1>>,          <<column_2>>,
<<aggregate_function>> (<<column_3 >>)
FROM <<table_name>>
WHERE <<condition>>
GROUP BY <<column_1>>, <<column_2>>
```

Kolom-kolom yang di SELECT tetapi tidak termasuk ke dalam aggregate function harus dimasukkan ke dalam GROUP BY.

5. DML Order By & Compute By

Order By digunakan untuk mengurutkan data, baik ascending maupun descending.

Compute By digunakan untuk membuat subtotal. Untuk menggunakan compute by harus menggunakan order by.

Order By & Compute By

```
SELECT      <<column 1>>,          <<column 2>>,
<<aggregate_function>> (<<column 3 >>)
FROM <<table_name>>
WHERE <<condition>>
GROUP BY <<column 1>>, <<column 2>>
ORDER BY <<column>> ASC|DESC
COMPUTE    <<compute_function>>    (<<column>>)    BY
<<column>>
```

6. DML String and Date Function

String Function

Sintaks	Format	Fungsi
CHARINDEX	CHARINDEX (string yang dicari, nama kolom, indeks mulai)	Mencari indeks dari string
LEFT	LEFT (nama kolom, jumlah huruf)	Ambil karakter dari sebelah kiri (awal)
RIGHT	RIGHT (nama kolom, jumlah huruf)	Ambil karakter dari sebelah kanan (akhir)
SUBSTRING	SUBSTRING (nama kolom, indeks mulai, jumlah karakter)	Mengambil beberapa karakter dari indeks tertentu
RTRIM	RTRIM (nama kolom)	Menghilangkan spasi di sebelah kanan
REVERSE	REVERSE (nama kolom)	Membalikkan kata atau kalimat
LEN	LEN (nama kolom)	Mencari panjang kata atau kalimat
STUFF	STUFF (String penerima, indeks mulai, jumlah karakter, string yang akan ditambah)	Memasukkan kata ke dalam kata lainnya.
CONVERT	CONVERT (tipe data tujuan, nama kolom, jenis style)	Mengubah tipe data suatu nilai ke tipe data lainnya.
CAST	CAST (nama kolom AS tipe data tujuan)	Untuk mengubah tipe data suatu nilai ke tipe data lainnya.

Date Function

Sintaks	Format	Fungsi
DATEDIFF	DATEDIFF (satuan, tanggal1, tanggal2)	Mendapatkan selisih tanggal.
DATENAME	DATENAME (satuan, tanggal)	Mendapatkan nama dari bagian tanggal tertentu.
DATEPART	DATEPART (satuan, tanggal)	Mengambil data tertentu dari tanggal
DATEADD	DATEADD (satuan, tambahan, tanggal)	Menambahkan satuan waktu tertentu ke tanggal.
MONTH	MONTH (tanggal)	Mengambil bulan
DAY	DAY (tanggal)	Mengambil tanggal
YEAR	YEAR (tanggal)	Mengambil tahun

Operator Logika

Sintaks	Fungsi
AND	Mengembalikan nilai TRUE jika kedua ekspresi TRUE.
OR	Mengembalikan nilai TRUE jika salah satu TRUE.
BETWEEN	Mengembalikan nilai TRUE jika operand yang diuji dalam jangkauan BETWEEN
LIKE	Mengembalikan nilai TRUE jika operand sesuai pattern yang ditentukan.
NOT	Membalikkan nilai boolean
EXISTS	Mengembalikan nilai TRUE jika subquery mengembalikan setidaknya 1 baris.
IN	Mengembalikan nilai TRUE jika operand yang diuji sesuai dengan salah satu ekspresi dalam IN.

7. DML Inner Join, Outer Join, Union

Join digunakan untuk query data dari dua atau lebih tabel.

Ada beberapa jenis Join antara lain :

1. JOIN (INNER JOIN).
Mengembalikan baris jika setidaknya terdapat satu kesamaan foreign key dan primary key yang dihubungkan dengan sintaks JOIN.
2. LEFT JOIN (LEFT OUTER JOIN).
Mengembalikan semua baris dari tabel yang ada di sebelah kiri perintah JOIN, bahkan jika tidak terdapat kesamaan antara foreign key dan primary key di kedua tabel.
3. RIGHT JOIN (RIGHT OUTER JOIN).
Mengembalikan semua baris dari tabel yang ada di sebelah kanan perintah JOIN, bahkan jika tidak terdapat kesamaan antara foreign key dan primary key di kedua tabel.
4. FULL JOIN (FULL OUTER JOIN)
Mengembalikan semua baris dari tabel yang ada di sebelah kiri dan kanan JOIN.

Join

```
SELECT *
FROM <<first table>>
[INNER JOIN / LEFT JOIN / RIGHT JOIN / FULL
JOIN] <<second table>>
ON <<first table>>.<<column>> =
<<second table>>.<<column>>
```

Union digunakan untuk menggabungkan semua nilai hasil SELECT dari masing-masing tabel.

UNION digunakan untuk mengurangi redudansi data/baris yang berulang dengan cara dihapus dari hasil UNION.

UNION ALL digunakan untuk mengembalikan semua baris hasil UNION tanpa mengurangi baris yang redudansi.

Union

<<SQL Statement 1>>

[UNION / UNION ALL]

<<SQL Statement 2>>

8. DML Subquery, Alias Subquery

Subquery adalah query select yang terdapat di dalam query select lainnya. Subquery disebut juga inner query atau inner select, sedangkan query yang mengandung subquery disebut outer query atau outer select.

Contoh subquery :

```
DELETE FROM orders
WHERE employee_id = (SELECT employee_id FROM
employees WHERE employee_name = 'Anto')
```

Alias subquery adalah subquery yang diberi nama.

Contoh alias subquery :

```
SELECT a.employee_name
FROM (SELECT * FROM employees) as a
```

9. DML IN & EXISTS

IN digunakan untuk mengambil nilai tertentu. Selain itu, IN juga digunakan dalam subquery. Subquery menggunakan IN jika sudah mengetahui kolom yang akan dikembalikan dari subquery.

In

```
SELECT <<column1>>  
FROM <<table1>>  
WHERE <<column1>> IN (SELECT <<column1>> FROM  
<<table2>>)
```

EXISTS digunakan untuk mengembalikan nilai boolean dari subquery. Nilai TRUE jika subquery dalam EXISTS mengandung minimal satu baris data. Nilai FALSE jika subquery dalam EXISTS tidak mengandung baris data sama sekali.

Exists

```
SELECT <<column1>>  
FROM <<table1>>  
WHERE EXISTS (SELECT * FROM <<table2>>)
```


10. DML ANY, SOME, ALL

ANY, SOME, dan ALL merupakan operator pembandingan yang biasa digunakan dalam subquery EXISTS.

ANY : jika salah satu nilai terpenuhi, maka kondisi benar.

SOME : sama seperti ANY.

ALL : harus memenuhi semua nilai untuk memperoleh kondisi benar.

11. View

View merupakan tabel virtual yang dapat dipakai untuk pemanggilan data, namun tidak tersimpan data secara fisik pada disk.

Create View

```
CREATE VIEW <<view_name>>  
AS  
<<query_select>>
```

Drop View

```
DROP VIEW <<view_name>>
```

12. Data Control Language

Grant digunakan untuk menambahkan hak akses ke user. Hak akses dapat berupa select, insert, update, delete, references, alter, dan index. Grant menggunakan sintaks GRANT.

Revoke digunakan untuk menghapus hak akses yang telah diberikan sebelumnya.

Revoke menggunakan sintaks REVOKE.

13. Store Procedure

Procedure adalah program yang dapat dipanggil dan dieksekusi untuk melakukan tugas tertentu.

Create Store Procedure

```
CREATE (PROC / PROCEDURE) <<procedure_name>>  
[ (<<parameter_data_type>>, ... ) ]  
AS <<query>>
```

Execute Store Procedure

```
[EXEC / EXECUTE] <<procedure_name>>  
[ (<<parameter>>, ... ) ]
```

14. Cursor

Cursor adalah object database yang digunakan oleh aplikasi untuk memanipulasi data dalam set per baris. Biasanya cursor digunakan dalam PL/SQL.

```
DECLARE <<cursor_name>> CURSOR FOR <<query>>
OPEN <<cursor_name>>
FETCH NEXT FROM <<cursor_name>>
WHILE @@FETCH_STATUS = 0
    BEGIN
        <<query>>
        FETCH NEXT FROM <<cursor_name>>
    END
CLOSE <<cursor_name>>
DEALLOCATE <<cursor_name>>
```

15. Trigger

Trigger adalah PL/SQL yang diasosiasikan dengan tabel agar ketika eksekusi SQL statement akan muncul event. Trigger dapat digunakan untuk keamanan sistem, membuat history record, dan lainnya.

Create Trigger

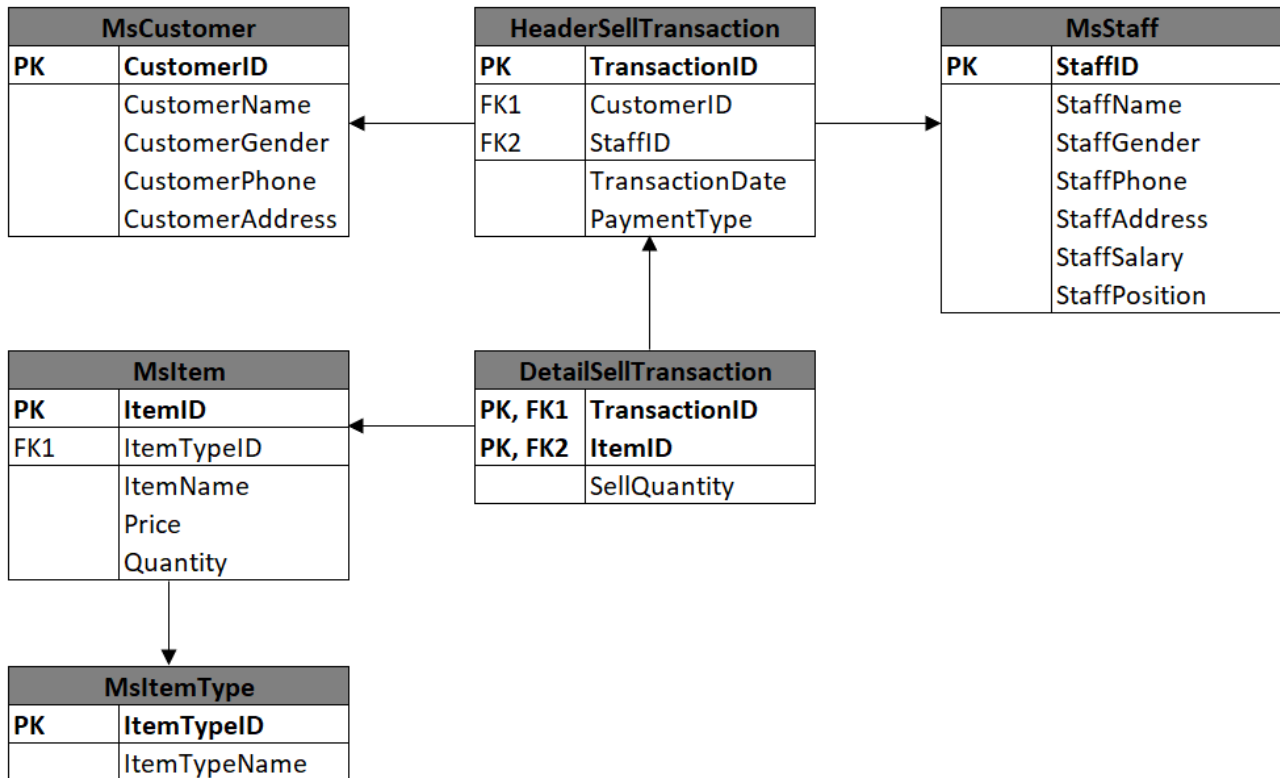
```
CREATE TRIGGER ON <<table_name>>  
FOR (INSERT / UPDATE / DELETE)  
AS <<query>>
```

Drop Trigger

```
DROP TRIGGER <<trigger_name>>
```

Pertemuan 1

Database Environment and Data Definition



Soal

1. Buatlah tabel-tabel dibawah ini dengan ketentuan sebagai berikut :

Nama Tabel : MsCustomer*Primary Key : CustomerId*

Nama Field	Tipe Data	Length	Keterangan
CustomerId	char	5	Harus diisi dan panjang = 5, harus diawali dengan 'CU' dan 3 digit terakhirnya berupa angka
CustomerName	varchar	50	Tidak boleh kosong
CustomerGender	varchar	10	Harus berupa 'Male' atau 'Female'
CustomerPhone	varchar	13	-
CustomerAddress	varchar	100	-

Nama Tabel : MsStaff*Primary Key : StaffId*

Nama Field	Tipe Data	Length	Keterangan
StaffId	char	5	Harus diisi dan panjang = 5, harus diawali dengan 'SF' dan 3 digit terakhirnya berupa angka
StaffName	varchar	50	Tidak boleh kosong
StaffGender	varchar	10	Harus berupa 'Male' atau 'Female'
StaffPhone	varchar	13	-
StaffAddress	varchar	100	-
StaffSalary	numeric	11, 2	-
StaffPosition	varchar	20	-

Nama Tabel : MsItemType*Primary Key : ItemTypeId*

Nama Field	Tipe Data	Length	Keterangan
ItemTypeId	char	5	Harus diisi dan panjang = 5, harus diawali dengan 'IT' dan 3 digit terakhirnya berupa angka
ItemTypeName	varchar	50	Tidak boleh kosong

Nama Tabel : MsItem*Primary Key : ItemId*

Nama Field	Tipe Data	Length	Keterangan
ItemId	char	5	Harus diisi dan panjang = 5, harus diawali dengan 'IM' dan 3 digit terakhirnya berupa angka
ItemTypeId	char	5	Tidak boleh kosong
ItemName	varchar	10	Tidak boleh kosong
Price	numeric	11, 2	-
Quantity	numeric	11, 2	-

Nama Tabel : HeaderSellTransaction*Primary Key : TransactionId*

Nama Field	Tipe Data	Length	Keterangan
TransactionId	Char	5	Harus diisi dan panjang = 5, harus diawali dengan 'TR' dan 3 digit terakhirnya berupa angka
CustomerId	Char	5	Tidak boleh kosong
StaffId	char	5	Tidak boleh kosong
TransactionDate	date	-	-
PaymentType	varchar	20	-

Nama Tabel : DetailSellTransaction*Primary Key : TransactionId, ItemId*

Nama Field	Tipe Data	Length	Keterangan
TransactionId	char	5	Tidak boleh kosong
ItemId	char	5	Tidak boleh kosong
SellQuantity	numeric	11,2	-

- Hapuslah tabel DetailSellTransaction.
(**drop table**)
- Buatlah tabel dengan ketentuan berikut :

Nama Tabel : DetailSellTransaction

Nama Field	Tipe Data	Length	Keterangan
TransactionId	char	5	Tidak boleh kosong
ItemId	char	5	Tidak boleh kosong
SellQuantity	numeric	11,2	-

Kemudian dengan perintah alter, ubahlah *field* TransactionId dan ItemId sehingga menjadi *primary key* dari tabel tersebut.

(**create table, alter table, primary key**)

4. Tambahkan *constraint* untuk tabel MsStaff dimana panjang StaffName harus diantara 5 dan 20, data yang terdahulu tidak perlu di cek. Kemudian hapuslah *constraint* tersebut.
(alter table, with nocheck, add constraint, drop)
5. Tambahkan kolom 'Description' pada tabel MsItem dengan tipe data varchar dengan panjang 100 karakter. Lalu hapuslah kolom tersebut.
(alter table, add, drop column)

Jawaban

--1

-- Create MsCustomer

```
CREATE TABLE MsCustomer(  
    CustomerId CHAR(5) PRIMARY KEY,  
    CustomerName VARCHAR(50),  
    CustomerGender VARCHAR(10),  
    CustomerPhone VARCHAR(13),  
    CustomerAddress VARCHAR(100),  
    CONSTRAINT cekIDCust CHECK(CustomerId LIKE 'CU[0-9][0-9][0-9]'),  
    CONSTRAINT cekGenCust CHECK(CustomerGender = 'Male' or CustomerGender =  
'Female')  
)
```

-- Create MsStaff

```
CREATE TABLE MsStaff(  
    StaffId CHAR(5) PRIMARY KEY,  
    StaffName VARCHAR(50),  
    StaffGender VARCHAR(10),  
    StaffPhone VARCHAR(13),  
    StaffAddress VARCHAR(100),  
    StaffSalary NUMERIC(11,2),  
    StaffPosition VARCHAR(20),  
    CONSTRAINT cekIDStaff CHECK(StaffId LIKE 'SF[0-9][0-9][0-9]'),  
    CONSTRAINT cekGenStaff CHECK(StaffGender = 'Male' or StaffGender = 'Female')  
)
```

-- Create MsItemType

```
CREATE TABLE MsItemType(  
    ItemTypeId CHAR(5) PRIMARY KEY,  
    ItemTypeName VARCHAR(50),  
    CONSTRAINT cekIDItemType CHECK(ItemTypeId LIKE 'IT[0-9][0-9][0-9]')  
)
```

-- Create MsItem

```
CREATE TABLE MsItem(  
    ItemId CHAR(5) PRIMARY KEY,  
    ItemTypeId CHAR(5) REFERENCES MsItemType ON UPDATE CASCADE ON  
DELETE CASCADE,  
    ItemName VARCHAR(50),  
    Price NUMERIC(11,2),  
    Quantity NUMERIC(11,2),  
    CONSTRAINT cekIDItem CHECK(ItemId LIKE 'IM[0-9][0-9][0-9]')  
)
```

```
-- Create HeaderSellTransaction
CREATE TABLE HeaderSellTransaction(
    TransactionId CHAR(5) PRIMARY KEY,
    CustomerId CHAR(5) REFERENCES MsCustomer ON UPDATE CASCADE ON
DELETE CASCADE,
    StaffId CHAR(5) REFERENCES MsStaff ON UPDATE CASCADE ON DELETE
CASCADE,
    TransactionDate DATE,
    PaymentType VARCHAR(20),
    CONSTRAINT cekIDTrans CHECK(TransactionId LIKE 'TR[0-9][0-9][0-9]')
)
```

```
-- Create DetailSellTransaction
CREATE TABLE DetailSellTransaction(
    TransactionId CHAR(5) REFERENCES HeaderSellTransaction ON UPDATE CASCADE
ON DELETE CASCADE,
    ItemId CHAR(5) REFERENCES MsItem ON UPDATE CASCADE ON DELETE
CASCADE,
    PRIMARY KEY(TransactionId, ItemId)
)
```

```
--2
begin tran
drop table DetailSellTransaction
rollback
select * from DetailSellTransaction
```

```
--3
CREATE TABLE DetailSellTransaction(
    TransactionId CHAR(5) REFERENCES HeaderSellTransaction ON UPDATE CASCADE
ON DELETE CASCADE not null,
    ItemId CHAR(5) REFERENCES MsItem ON UPDATE CASCADE ON DELETE
CASCADE not null,
)
```

```
begin tran
alter table DetailSellTransaction add PRIMARY KEY(TransactionId, ItemId)
```

```
--4
begin tran
alter table MsStaff with nocheck
add constraint CheckStaff check(len(StaffName) between 5 and 20)

alter table MsStaff
drop CheckStaff

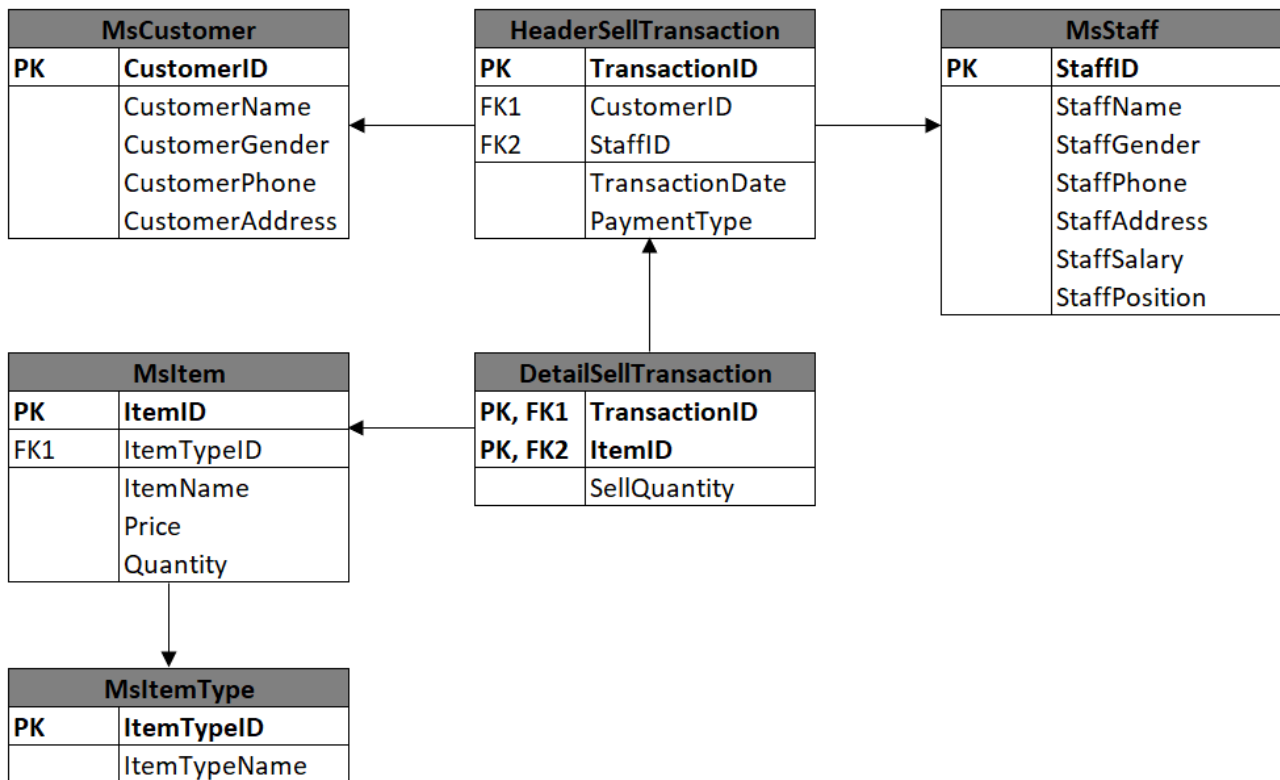
select * from MsStaff

--5
begin tran
alter table MsItem
add "Description" varchar(100)

alter table MsItem
drop column "Description"

select * from MsItem
```

Pertemuan 2
DML Insert, Update, Delete, Select Data



Soal

1. Tampilkan semua kolom dari MsStaff **dimana** StaffGender adalah 'Female'.

	StaffId	StaffName	StaffGender	StaffPhone	StaffAddress	StaffSalary	StaffPosition
1	SF002	Poppy Hwangsa Iswara	Female	085714263367	Budi Raya Street no 11	10000000.00	Supervisor

2. Tampilkan StaffName dan StaffSalary (didapat dari StaffSalary dengan menambahkan kata 'Rp. ' diawalnya) **dimana** StaffSalary lebih besar atau sama dengan 5.000.000 dan StaffName mengandung huruf 's'.

(cast, like)

	StaffName	StaffSalary
1	Poppy Hwangsa Iswara	Rp. 10000000.00
2	Louis Rudy Valen	Rp. 6000000.00

3. Tampilkan ItemName, Price dan Quantity **dimana** ItemTypeName adalah 'Health' atau 'Furniture'.

(in)

	ItemName	Price	Quantity
1	Bed	2100000.00	32.00
2	Stove	320000.00	20.00
3	Wardrobe	760000.00	17.00
4	Vitamin	290000.00	99.00
5	Sunblock	125000.00	32.00
6	Body lotion	320000.00	48.00

4. Tampilkan StaffName, StaffPosition dan TransactionDate (dengan format 'Mon dd,yyyy') **dimana** StaffSalary diantara 7000000 dan 10000000.

(convert, between)

	StaffName	StaffPosition	TransactionDate
1	Nathaniel Putera	Cashier	Dec 20, 2017
2	Nathaniel Putera	Cashier	Dec 20, 2017
3	Nathaniel Putera	Cashier	Dec 21, 2017
4	Poppy Hwangsa Iswara	Supervisor	Dec 22, 2017

5. Tampilkan Name (didapat dari nama pertama CustomerName), Gender (didapat dari huruf pertama CustomerGender) dan PaymentType **dimana** PaymentType adalah 'Debit'.

(substring, charindex, left)

	Name	Gender	PaymentType
1	Naufal	M	Debit
2	Arya	M	Debit
3	Arya	M	Debit

6. Tampilkan Initial (didapat dari huruf pertama CustomerName dan huruf pertama dari kata kedua CustomerName dengan format huruf kapital) dan Day (didapat dari hari terjadinya transaksi) **dimana** selisih tanggal terjadinya transaksi dengan tanggal 24 Desember 2017 kurang dari tiga hari.

(upper, left, substring, charindex, datename, weekday, datediff, day)

	Initial	Day
1	KA	Friday
2	FL	Friday
3	AT	Friday

7. Tampilkan TransactionDate dan CustomerName (didapat dari nama terakhir CustomerName) **dimana** CustomerName terdiri dari dua kata atau lebih dan customer tersebut pernah melakukan transaksi pada hari Jumat.

(right, charindex, reverse, like, datename, weekday)

	TransactionDate	CustomerName
1	2017-12-22	Augustin
2	2017-12-22	Lioexander
3	2017-12-22	Thomas

8. Tampilkan StaffName, CustomerName, CustomerPhone (didapat dari CustomerPhone dengan mengganti bilangan '0' dengan '+62') dan CustomerAddress **dimana** CustomerName diawali huruf vokal (a,i,u,e,o) dan StaffName terdiri dari tiga kata atau lebih.

(replace, like)

	StaffName	CustomerName	CustomerPhone	CustomerAddress
1	Louis Rudy Valen	Arya Thomas	+6281811992617	Anggrek Street no 12

9. Tampilkan StaffName, ItemName dan Term of Transaction (didapat dari selisih tanggal transaksi dengan 24 Desember 2017) **dimana** panjang karakter dari ItemName lebih besar dari 20 atau ItemName terdiri dari dua kata atau lebih.

(datediff, day, len, charindex)

	StaffName	ItemName	Term of Transaction
1	Andika Leonardo	Rice cooker	4
2	Venky Eduardo	Rice cooker	2
3	Andika Leonardo	Rice cooker	2

10. Tampilkan TransactionDate, CustomerName, ItemName, Discount (didapat dari 20% Price dengan format bilangan bulat) dan PaymentType **dimana** transaksi terjadi pada tanggal 22.

(cast, day)

	TransactionDate	CustomerName	ItemName	Discount	PaymentType
1	2017-12-22	Kerin Augustin	Television	660000	Cash
2	2017-12-22	Kerin Augustin	Rice cooker	190000	Cash
3	2017-12-22	Fernando Lioexander	Television	660000	Credit
4	2017-12-22	Fernando Lioexander	Lamp	19000	Credit
5	2017-12-22	Arya Thomas	Rice cooker	190000	Debit
6	2017-12-22	Arya Thomas	Lamp	19000	Debit

11. Insertlah data berikut!
(insert)

Table : MsStaff

StaffID	StaffName	StaffGender	StaffPhone	StaffAddress	StaffSalary	StaffPosition
SF006	Ryifki Yidan	Male	081901927772	Kedoya Street no 10	3000000	Cashier
SF007	Muhammad Farrel Adivia	Male	085700018281	Sudirman Street no 55	5000000	Cashier
SF008	Gilbert Christopher	Male	085737738292	Diponegoro Street no 124	7000000	Supervisor
SF009	Eriksen Finarto	Male	085382819384	Palmerah Street no 2	6800000	Supervisor

Table : HeaderSellTransaction

TransactionID	CustomerID	StaffID	TransactionDate	PaymentType
TR010	CU001	SF004	2017/12/23	Credit
TR011	CU002	SF006	2017/12/24	Credit
TR012	CU003	SF007	2017/12/24	Cash
TR013	CU004	SF005	2017/12/25	Debit
TR014	CU005	SF007	2017/12/25	Debit
TR015	CU005	SF005	2017/12/26	Credit
TR016	CU002	SF001	2017/12/26	Cash
TR017	CU003	SF002	2017/12/26	Credit
TR018	CU005	SF001	2017/12/27	Debit

Table : DetailSellTransaction

TransactionID	ItemID	SellQuantity
TR010	IM003	1
TR010	IM005	2
TR010	IM010	1
TR011	IM015	2
TR011	IM025	2
TR012	IM009	1
TR013	IM003	1
TR013	IM006	1
TR013	IM015	2
TR014	IM016	1
TR015	IM016	1
TR015	IM006	1
TR016	IM015	1
TR016	IM003	1
TR016	IM005	1
TR017	IM003	2
TR018	IM006	3
TR018	IM005	1
TR018	IM007	1

12. Insertlah data berikut!
(insert, dateadd, day, getdate)

Table : HeaderSellTransaction

TransactionID	CustomerID	StaffID	TransactionDate	PaymentType
TR019	CU005	SF004	Didapat dari tanggal hari ini ditambah 3 hari.	Credit

13. Insertlah data berikut!
(insert, round, rand)

Table : MsStaff

StaffID	StaffName	StaffGender	StaffPhone	StaffAddress	StaffSalary	StaffPosition
SF010	Berlianto	Male	081128391844	Tangerang City Street no 88	Didapat dari hasil random dengan range antara 5000000 sampai 7000000	Supervisor

14. Ubahlah CustomerPhone dengan mengganti bilangan '08' menjadi '628' pada table MsCustomer. Kemudian tampilkan seluruh data dari MsCustomer.
(update, replace)

	CustomerId	CustomerName	CustomerGender	CustomerPhone	CustomerAddress
1	CU001	Kevin Axellino Triantio	Male	6281267381930	Jelambar Street no 17
2	CU002	Kerin Augustin	Female	6281372116372	Ancol Barat Street no 190
3	CU003	Fernando Lioexander	Male	6287824153627	Palmerah Street no 20
4	CU004	Naufal Hafiz	Male	6281127173829	Duri Kepa Street no 9
5	CU005	Arya Thomas	Male	6281811992617	Anggrek Street no 12

15. Ubahlah StaffPosition menjadi 'Supervisor' dan StaffSalary menjadi StaffSalary awal ditambahkan dengan 2000000 pada table MsStaff dimana StaffName adalah 'Ryifki Yidan'. Kemudian tampilkan seluruh data dari MsStaff.
(update, like)

	StaffId	StaffName	StaffGender	StaffPhone	StaffAddress	StaffSalary	StaffPosition
1	SF001	Venky Eduardo	Male	081927183617	Budi Raya Street no 100	15000000.00	Manager
2	SF002	Poppy Hwangsa Iswara	Female	085714263367	Budi Raya Street no 11	10000000.00	Supervisor
3	SF003	Louis Rudy Valen	Male	089915276278	Serpong Raya Street no 109	6000000.00	Cashier
4	SF004	Andika Leonardo	Male	081189227888	BSD Raya Street no 1	6500000.00	Cashier
5	SF005	Nathaniel Putera	Male	087715551782	Ciledug Street no 233	7300000.00	Cashier
6	SF006	Ryifki Yidan	Male	081901927772	Kedoya Street no 10	5000000.00	Supervisor
7	SF007	Muhammad Farrel Adivia	Male	085700018281	Sudiman Street no 55	5000000.00	Cashier
8	SF008	Gilbert Christopher	Male	085737738292	Diponegoro Street no 124	7000000.00	Supervisor
9	SF009	Eriksen Finarto	Male	085382819384	Palmerah Street no 2	6800000.00	Supervisor
10	SF010	Berlianto	Male	081128391844	Tangerang City Street no 88	9217546.00	Supervisor

16. Ubahlah CustomerName menjadi nama pertama dari CustomerName pada table MsCustomer **dimana** Customer tersebut pernah melakukan transaksi pada tanggal 24. Kemudian tampilkan seluruh data dari MsCustomer.

(update, left, charindex, join, datepart, day)

	CustomerId	CustomerName	CustomerGender	CustomerPhone	CustomerAddress
1	CU001	Kevin Axellino Triantio	Male	081267381930	Jelambar Street no 17
2	CU002	Kerin	Female	081372116372	Ancol Barat Street no 190
3	CU003	Fernando	Male	087824153627	Palmerah Street no 20
4	CU004	Naufal Hafiz	Male	081127173829	Duri Kepa Street no 9
5	CU005	Arya Thomas	Male	081811992617	Anggrek Street no 12

17. Ubahlah CustomerName menjadi CustomerName dengan menambahkan kata 'Ms. ' diawal pada table MsCustomer **dimana** CustomerId adalah 'CU002'. Kemudian tampilkan seluruh data dari MsCustomer.

(update)

	CustomerId	CustomerName	CustomerGender	CustomerPhone	CustomerAddress
1	CU001	Kevin Axellino Triantio	Male	081267381930	Jelambar Street no 17
2	CU002	Ms. Kerin Augustin	Female	081372116372	Ancol Barat Street no 190
3	CU003	Fernando Lioexander	Male	087824153627	Palmerah Street no 20
4	CU004	Naufal Hafiz	Male	081127173829	Duri Kepa Street no 9
5	CU005	Arya Thomas	Male	081811992617	Anggrek Street no 12

18. Ubahlah CustomerAddress menjadi 'Meruya Street No. 123' pada table MsCustomer **dimana** Customer tersebut pernah dilayani oleh Staff yang bernama 'Nathaniel Putera' dan transaksi terjadi pada hari kamis. Kemudian tampilkan seluruh data dari MsCustomer.

(update, in, exists, datename, weekday)

	CustomerId	CustomerName	CustomerGender	CustomerPhone	CustomerAddress
1	CU001	Kevin Axellino Triantio	Male	081267381930	Meruya Street No. 123
2	CU002	Kerin Augustin	Female	081372116372	Ancol Barat Street no 190
3	CU003	Fernando Lioexander	Male	087824153627	Palmerah Street no 20
4	CU004	Naufal Hafiz	Male	081127173829	Duri Kepa Street no 9
5	CU005	Arya Thomas	Male	081811992617	Anggrek Street no 12

19. Update CustomerName menjadi Kerin untuk CustomerId CU002. Kemudian hapuslah data pada table HeaderSellTransaction **dimana** CustomerName hanya terdiri dari satu kata. Kemudian tampilkan seluruh data dari HeaderSellTransaction.
(delete, join, charindex)

Before

	TransactionId	CustomerId	StaffId	TransactionDate	Payment Type
1	TR001	CU001	SF004	2017-12-20	Credit
2	TR002	CU002	SF005	2017-12-20	Credit
3	TR003	CU003	SF003	2017-12-20	Cash
4	TR004	CU004	SF005	2017-12-20	Debit
5	TR005	CU005	SF003	2017-12-21	Debit
6	TR006	CU001	SF005	2017-12-21	Credit
7	TR007	CU002	SF001	2017-12-22	Cash
8	TR008	CU003	SF002	2017-12-22	Credit
9	TR009	CU005	SF004	2017-12-22	Debit
10	TR010	CU001	SF004	2017-12-23	Credit
11	TR011	CU002	SF006	2017-12-24	Credit
12	TR012	CU003	SF007	2017-12-24	Cash
13	TR013	CU004	SF005	2017-12-25	Debit
14	TR014	CU005	SF007	2017-12-25	Debit
15	TR015	CU005	SF005	2017-12-26	Credit
16	TR016	CU002	SF001	2017-12-26	Cash
17	TR017	CU003	SF002	2017-12-26	Credit
18	TR018	CU005	SF001	2017-12-27	Debit
19	TR019	CU005	SF004	2018-09-17	Credit

After

	TransactionId	CustomerId	StaffId	TransactionDate	Payment Type
1	TR001	CU001	SF004	2017-12-20	Credit
2	TR003	CU003	SF003	2017-12-20	Cash
3	TR004	CU004	SF005	2017-12-20	Debit
4	TR005	CU005	SF003	2017-12-21	Debit
5	TR006	CU001	SF005	2017-12-21	Credit
6	TR008	CU003	SF002	2017-12-22	Credit
7	TR009	CU005	SF004	2017-12-22	Debit
8	TR010	CU001	SF004	2017-12-23	Credit
9	TR012	CU003	SF007	2017-12-24	Cash
10	TR013	CU004	SF005	2017-12-25	Debit
11	TR014	CU005	SF007	2017-12-25	Debit
12	TR015	CU005	SF005	2017-12-26	Credit
13	TR017	CU003	SF002	2017-12-26	Credit
14	TR018	CU005	SF001	2017-12-27	Debit
15	TR019	CU005	SF004	2018-09-17	Credit

20. Hapuslah data pada table MsItem **dimana** ItemTypeName tidak mengandung kata 'ni'.
Kemudian tampilkan seluruh data dari MsItem.
(delete, in, not like)

Before

	ItemId	ItemTypeId	ItemName	Price	Quantity
1	IM001	IT001	Camera	9000000.00	10.00
2	IM002	IT001	Television	3300000.00	19.00
3	IM003	IT001	Handphone	5100000.00	9.00
4	IM004	IT001	Laptop	10050000.00	20.00
5	IM005	IT001	Rice cooker	950000.00	17.00
6	IM006	IT001	Lamp	95000.00	90.00
7	IM007	IT001	Modem	420000.00	77.00
8	IM008	IT001	Harddisk	7900000.00	16.00
9	IM009	IT001	Mouse	210000.00	29.00
10	IM010	IT001	Keyboard	170000.00	41.00
11	IM011	IT002	Racket	630000.00	22.00
12	IM012	IT002	Volleyball	420000.00	28.00
13	IM013	IT002	Golf stick	1900000.00	12.00
14	IM014	IT002	Basketball	920000.00	14.00
15	IM015	IT002	Bicycle	2300000.00	25.00
16	IM016	IT002	Diving	4400000.00	12.00
17	IM017	IT003	Bed	2100000.00	32.00
18	IM018	IT003	Stove	320000.00	20.00
19	IM019	IT003	Wardrobe	760000.00	17.00
20	IM020	IT004	Blazer	420000.00	41.00
21	IM021	IT004	Shirt	320000.00	64.00
22	IM022	IT004	Pants	580000.00	52.00
23	IM023	IT004	Hat	210000.00	58.00
24	IM024	IT005	Vitamin	290000.00	99.00
25	IM025	IT005	Sunblock	125000.00	32.00
26	IM026	IT005	Body lotion	320000.00	48.00

After

	ItemId	ItemTypeId	ItemName	Price	Quantity
1	IM001	IT001	Camera	9000000.00	10.00
2	IM002	IT001	Television	3300000.00	19.00
3	IM003	IT001	Handphone	5100000.00	9.00
4	IM004	IT001	Laptop	10050000.00	20.00
5	IM005	IT001	Rice cooker	950000.00	17.00
6	IM006	IT001	Lamp	95000.00	90.00
7	IM007	IT001	Modem	420000.00	77.00
8	IM008	IT001	Harddisk	7900000.00	16.00
9	IM009	IT001	Mouse	210000.00	29.00
10	IM010	IT001	Keyboard	170000.00	41.00
11	IM017	IT003	Bed	2100000.00	32.00
12	IM018	IT003	Stove	320000.00	20.00
13	IM019	IT003	Wardrobe	760000.00	17.00

Jawaban

--1.

```
select * from MsStaff where StaffGender = 'Female'
```

--2.

```
select StaffName, StaffSalary = 'Rp. ' + CAST(StaffSalary as varchar)
from MsStaff
where StaffSalary >= 5000000
and
StaffName like '%s%'
```

--3.

```
select ItemName, Price , Quantity
from MsItem mi, MsItemType mit
where
mi.ItemTypeId = mit.ItemTypeId
and ItemTypeName in ('Health', 'Furniture')
```

--4.

```
select StaffName, StaffPosition, TransactionDate = CONVERT(varchar, TransactionDate, 107)
from MsStaff ms, HeaderSellTransaction hst
where ms.StaffId = hst.StaffId
and StaffSalary between 7000000 and 10000000
```

--5.

```
select Name = SUBSTRING(CustomerName, 1,CHARINDEX(' ', CustomerName)), Gender =
LEFT(CustomerGender, 1), PaymentType
from MsCustomer mc, HeaderSellTransaction hst
where mc.CustomerId = hst.CustomerId
and PaymentType = 'Debit'
```

--6.

```
select Initial = UPPER(LEFT(CustomerName, 1)+SUBSTRING(CustomerName, CHARINDEX(' ',
CustomerName)+1, 1)),[Day] = DATENAME(weekday, TransactionDate)
from MsCustomer mc, HeaderSellTransaction hst
where mc.CustomerId = hst.CustomerId
and DATEDIFF(DAY, TransactionDate, '12/24/2017') < 3
```

--7.

```
select TransactionDate, CustomerName = RIGHT(CustomerName,CHARINDEX('
',REVERSE(CustomerName))-1)
from MsCustomer mc, HeaderSellTransaction hst
where mc.CustomerId = hst.CustomerId
and CustomerName like '% %'
and DATENAME(WEEKDAY, TransactionDate) = 'Friday'
```

--8.

```
select StaffName, CustomerName, CustomerPhone = REPLACE(CustomerPhone, '0', '+62'),  
CustomerAddress  
from MsStaff ms, HeaderSellTransaction hs, MsCustomer mc  
where ms.StaffId=hs.StaffId  
AND mc.CustomerId=hs.CustomerId  
and CustomerName like '[aiueo]%'  
and StaffName like '% % %'
```

--9.

```
select StaffName, ItemName, [Term of Transaction] = DATEDIFF(day, TransactionDate,  
'12/24/2017')  
from MsStaff ms, HeaderSellTransaction hst, DetailSellTransaction dst, MsItem mi  
where ms.StaffId = hst.StaffId  
and dst.TransactionId = hst.TransactionId  
and dst.ItemId = mi.ItemId  
and (LEN(ItemName) > 20  
or  
CHARINDEX(' ', ItemName)!=0)
```

--10.

```
select TransactionDate, CustomerName, ItemName, Discount = CAST(0.2*Price AS INT),  
PaymentType  
from MsCustomer mc, HeaderSellTransaction hst, DetailSellTransaction dst, MsItem mi  
where mc.CustomerId = hst.CustomerId  
and hst.TransactionId = dst.TransactionId  
and dst.ItemId = mi.ItemId  
and DAY(TransactionDate) = 22
```

--11.

BEGIN TRAN

INSERT INTO MSSTAFF VALUES

('SF006', 'Ryifki Yidan', 'Male', '081901927772', 'Kedoya Street no 10', 3000000, 'Cashier'),
('SF007', 'Muhammad Farrel Adivia', 'Male', '085700018281', 'Sudirman Street no 55', 5000000,
'Cashier'),
('SF008', 'Gilbert Christopher', 'Male', '085737738292', 'Diponegoro Street no 124', 7000000,
'Supervisor'),
('SF009', 'Eriksen Finarto', 'Male', '085382819384', 'Palmerah Street no 2', 6800000, 'Supervisor')

INSERT INTO HeaderSellTransaction VALUES

('TR010', 'CU001', 'SF004', '2017/12/23', 'Credit'),
('TR011', 'CU002', 'SF006', '2017/12/24', 'Credit'),
('TR012', 'CU003', 'SF007', '2017/12/24', 'Cash'),
('TR013', 'CU004', 'SF005', '2017/12/25', 'Debit'),
('TR014', 'CU005', 'SF007', '2017/12/25', 'Debit'),
('TR015', 'CU005', 'SF005', '2017/12/26', 'Credit'),
('TR016', 'CU002', 'SF001', '2017/12/26', 'Cash'),
('TR017', 'CU003', 'SF002', '2017/12/26', 'Credit'),
('TR018', 'CU005', 'SF001', '2017/12/27', 'Debit')

INSERT INTO DetailSellTransaction VALUES

('TR010', 'IM003',1),
('TR010', 'IM005',2),
('TR010', 'IM010',1),
('TR011', 'IM015',2),
('TR011', 'IM025',2),
('TR012', 'IM009',1),
('TR013', 'IM003',1),
('TR013', 'IM006',1),
('TR013', 'IM015',2),
('TR014', 'IM016',1),
('TR015', 'IM016',1),
('TR015', 'IM006',1),
('TR016', 'IM015',1),
('TR016', 'IM003',1),
('TR016', 'IM005',1),
('TR017', 'IM003',2),
('TR018', 'IM006',3),
('TR018', 'IM005',1),
('TR018', 'IM007',1)

--12

```
INSERT INTO HeaderSellTransaction  
VALUES('TR019', 'CU005', 'SF004', DATEADD(DAY, 3, GETDATE()), 'Credit')
```

--13

```
INSERT INTO MsStaff  
VALUES('SF010', 'Berlianto', 'Male', '081128391844', 'Tangerang City Street no 88',  
ROUND(rand()*2000000 + 5000000, 0), 'Supervisor')  
select * from MsStaff
```

--14

```
begin tran  
UPDATE MsCustomer SET CustomerPhone = REPLACE(CustomerPhone, '08', '628')  
select * from MsCustomer  
rollback
```

--15

```
begin tran  
UPDATE MsStaff set StaffPosition = 'Supervisor', StaffSalary = StaffSalary + 2000000  
where StaffName like 'Ryifki Yidan'  
select * from MsStaff  
rollback
```

--16

```
begin tran  
UPDATE MsCustomer  
set CustomerName = LEFT(CustomerName, CHARINDEX(' ', CustomerName)-1)  
from MsCustomer a join HeaderSellTransaction b  
on a.CustomerId = b.CustomerId  
where DATEPART(DAY, TransactionDate) = 24  
select * from MsCustomer  
rollback
```

--17

```
begin tran  
UPDATE MsCustomer  
set CustomerName = 'Ms. ' + CustomerName  
where CustomerId = 'CU002'  
select * from MsCustomer  
rollback
```

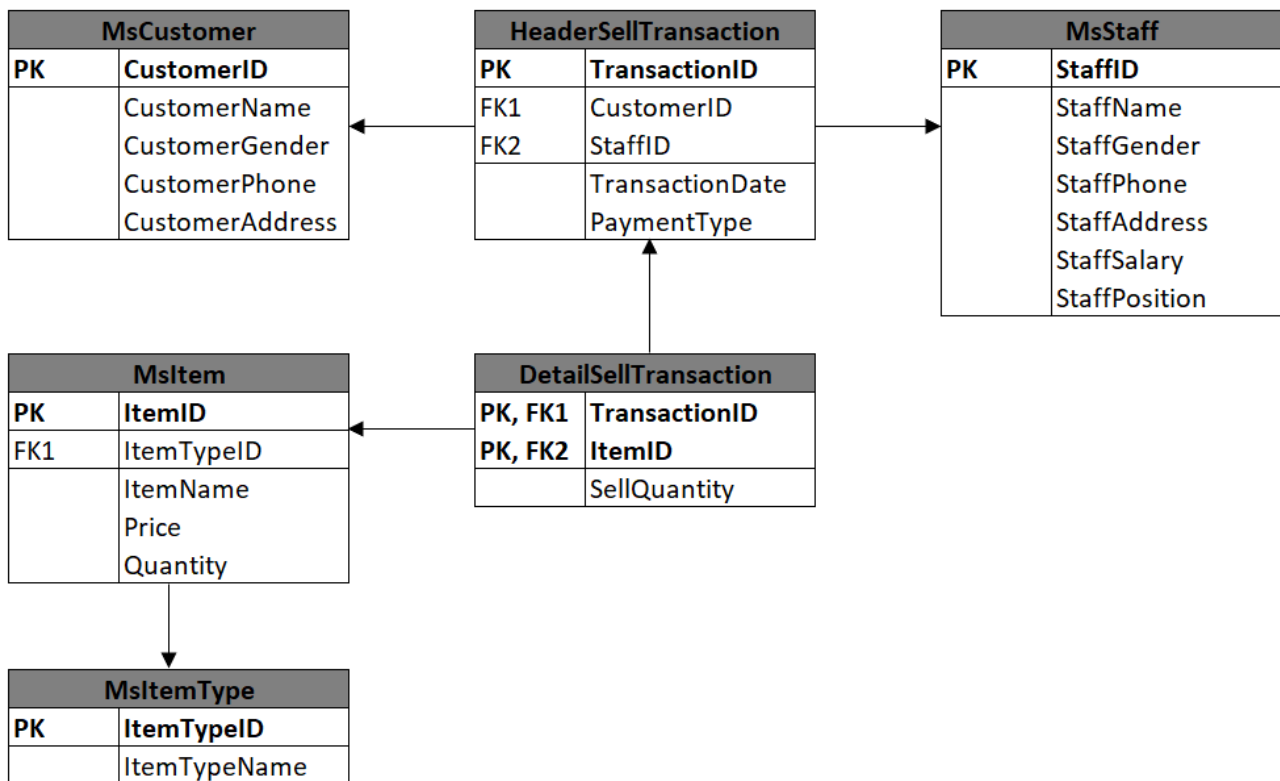
```
--18
begin tran
UPDATE MsCustomer
set CustomerAddress = 'Meruya Street No. 123'
from MsCustomer a
where CustomerId in(
    select CustomerId from HeaderSellTransaction b
    where exists(
        select * from MsStaff c
        where b.StaffId = c.StaffId
        and StaffName = 'Nathaniel Putera'
    )
    and DATENAME(WEEKDAY, TransactionDate) = 'Thursday'
)
select * from MsCustomer
rollback

--19
begin tran
update MsCustomer set CustomerName = 'Kerin'
where CustomerId = 'CU002'
select * from HeaderSellTransaction
delete HeaderSellTransaction
from HeaderSellTransaction a join MsCustomer b on a.CustomerId = b.CustomerId
where CHARINDEX(' ', CustomerName) = 0
select * from HeaderSellTransaction
rollback

--20
begin tran
select * from MsItem
delete MsItem
where ItemTypeId in(
    select ItemTypeId
    from MsItemType
    where ItemTypeName not like '%ni%'
)
select * from MsItem
rollback
```

Pertemuan 3

DML Grouping, Aggregate, Order By, Compute



Soal

1. Tampilkan Maximum Price (didapat dari Price tertinggi), Minimum Price (didapat dari Price terendah) dan Average Price (didapat dari rata-rata Price yang dibulatkan dengan format dua digit di belakang koma).

(max, min, cast, round, avg)

	Maximum Price	Minimum Price	Average Price
1	10050000.00	95000.00	2046538.00

2. Tampilkan StaffPosition, Gender (didapat dari huruf pertama StaffGender) dan Average Salary (didapat dari rata-rata StaffSalary yang dibulatkan dengan format dua digit di belakang koma dan diawali dengan kata 'Rp. ').

(left, cast, avg, group by)

	StaffPosition	Gender	Average Salary
1	Supervisor	F	Rp. 10000000.00
2	Cashier	M	Rp. 6600000.00
3	Manager	M	Rp. 15000000.00

3. Tampilkan Transaction Date (didapat dari TransactionDate dengan format 'Mon dd, yyyy') dan Total Transaction per Day (didapat dari banyaknya transaksi per hari). Kemudian tampilkan pula rata-rata dari Total Transaction per Day.

(convert, count, group by, compute, avg)

	TransactionDate	Total Transaction per Day
1	Dec 20, 2017	4
2	Dec 21, 2017	2
3	Dec 22, 2017	3

	avg
1	3

4. Tampilkan CustomerGender (didapat dari CustomerGender dengan format huruf kapital) dan Total Transaction (didapat dari banyaknya transaksi berdasarkan jenis kelamin Customer). Dan tampilkan pula jumlah Total Transaction.

(upper, count, group by, compute, sum)

	CustomerGender	Total Transaction
1	FEMALE	2
2	MALE	7

	sum
1	9

5. Tampilkan ItemTypeName, Total Transaction (didapat dari banyaknya transaksi) kemudian urutkan data berdasarkan Total Transaction secara descending (count, group by, order by)

	ItemTypeName	Total Transaction
1	Electronic	13
2	Sport	5
3	Health	1

6. Tampilkan Date (didapat dari tanggal terjadinya transaksi dengan format 'dd mon yy') dan Revenue per Day (didapat dari jumlah Price dan diawali dengan kata 'Rp. ') **dimana** Revenue per Day diantara 5000000 dan 10000000. (convert, cast, sum, group by, having, between)

	Date	Revenue per Day
1	21 Dec 2017	Rp. 6795000.00
2	22 Dec 2017	Rp. 8690000.00

7. Tampilkan ID (didapat dari ItemTypeId dengan mengganti kata 'IT0' menjadi 'Item Type '), ItemTypeName dan Total Item per Type (didapat dari banyaknya ItemID dengan diakhiri kata ' Item') **dimana** Total Item per Type lebih besar dari 5. Kemudian urutkan data berdasarkan Total Item per Type secara descending. (replace, cast, count, group by, having, order by)

	ID	ItemTypeName	Total Item per Type
1	Item Type 01	Electronic	10 Item
2	Item Type 02	Sport	6 Item

8. Tampilkan StaffName (didapat dari nama pertama StaffName), TransactionID dan Total Item per Transaction (didapat dari banyaknya item per transaksi). Tampilkan pula Total Item per Transaction terbesar. (left, charindex, count, group by, compute, max)

	StaffName	TransactionID	Total Item per Transaction
2	Nathaniel	TR002	2
3	Louis	TR003	2
4	Nathaniel	TR004	4
5	Louis	TR005	1
6	Nathaniel	TR006	2
7	Venky	TR007	2
8	Poppy	TR008	2
9	Andika	TR009	2

	max
1	4

9. Tampilkan TransactionDate, CustomerName, ItemName dan Price **dimana** Transaksi terjadi pada hari Rabu dan StaffName diawali kata 'Nathan'. Tampilkan pula total Price berdasarkan TransactionDate dan CustomerName.
(datetime, weekday, like, order by, compute by, sum)

	TransactionDate	CustomerName	ItemName	Price
1	2017-12-20	Kerin Augustin	Keyboard	170000.00
2	2017-12-20	Kerin Augustin	Bicycle	2300000.00
sum				
1	2470000.00			

	TransactionDate	CustomerName	ItemName	Price
1	2017-12-20	Naufal Hafiz	Camera	9000000.00
2	2017-12-20	Naufal Hafiz	Lamp	95000.00
3	2017-12-20	Naufal Hafiz	Bicycle	2300000.00
4	2017-12-20	Naufal Hafiz	Diving	4400000.00
sum				
1	15795000.00			

10. Tampilkan TransactionDate, CustomerName dan TotalPrice (didapat dari total Price) **dimana** transaksi terjadi setelah tanggal 20. Tampilkan pula grand TotalPrice dan minimum TotalPrice berdasarkan tanggal transaksi. Urutkan pula data berdasarkan tanggal transaksi.
(sum, day, group by, order by, compute by, min)

	TransactionDate	CustomerName	TotalPrice
1	2017-12-21	Aya Thomas	4400000.00
2	2017-12-21	Kevin Axellino Triantio	2395000.00
sum min			
1	6795000.00	2395000.00	

	TransactionDate	CustomerName	TotalPrice
1	2017-12-22	Aya Thomas	1045000.00
2	2017-12-22	Femando Lioexander	3395000.00
3	2017-12-22	Kerin Augustin	4250000.00
sum min			
1	8690000.00	1045000.00	

Jawaban

--1.

```
select [Maximum Price] = MAX(Price), [Minimum Price] = MIN(Price), [Average Price] =  
CAST(ROUND(AVG(Price), 0) AS NUMERIC(10,2))  
from MsItem
```

--2.

```
select StaffPosition, Gender = LEFT(StaffGender, 1), [Average Salary] = 'Rp. ' +  
CAST(CAST(AVG(StaffSalary) AS NUMERIC(10,2)) as VARCHAR)  
from MsStaff  
group by StaffPosition, StaffGender
```

--3.

```
select TransactionDate = CONVERT(vchar, TransactionDate, 107), [Total Transaction per Day]  
= COUNT(TransactionId)  
from HeaderSellTransaction hst  
group by TransactionDate  
compute AVG(COUNT(TransactionId))
```

--4.

```
select CustomerGender = UPPER(CustomerGender), [Total Transaction] = COUNT(TransactionId)  
from MsCustomer mc, HeaderSellTransaction hst  
where mc.CustomerId = hst.CustomerId  
group by CustomerGender  
compute SUM(COUNT(TransactionId))
```

--5.

```
select ItemTypeName, [Total Transaction] = COUNT(dst.TransactionId)  
from DetailSellTransaction dst, MsItem im, MsItemType it  
where dst.ItemId = im.ItemId  
and im.ItemTypeId = it.ItemTypeId  
group by ItemTypeName  
order by [Total Transaction] desc
```

--6.

```
select [Date] = CONVERT(vchar, TransactionDate, 106), [Revenue per Day] = 'Rp. ' +  
CAST(SUM(Price) as vchar)  
from HeaderSellTransaction hst, DetailSellTransaction dst, MsItem im  
where hst.TransactionId = dst.TransactionId  
and dst.ItemId = im.ItemId  
group by TransactionDate  
having SUM(Price) between 5000000 and 10000000
```

--7.

```
select [ID] = REPLACE(mt.ItemTypeId, 'IT0', 'ItemType '), ItemTypeName, [Total Item per Type]
= CAST(COUNT(ItemID) AS VARCHAR) + ' Item'
from MsItemType mtt, MsItem mt
where mtt.ItemTypeId = mt.ItemTypeId
group by mt.ItemTypeId, ItemTypeName
having COUNT(ItemId) > 5
order by COUNT(ItemId) desc
```

--8.

```
select StaffName = LEFT(StaffName, CHARINDEX(' ', StaffName)-1), hs.TransactionID, [Total
Item per Transaction] = COUNT(ItemId)
from MsStaff ms, HeaderSellTransaction hs, DetailSellTransaction ds
where ms.StaffId = hs.StaffId
and hs.TransactionId = ds.TransactionId
group by StaffName, hs.TransactionId
compute MAX(COUNT(ItemId))
```

--9.

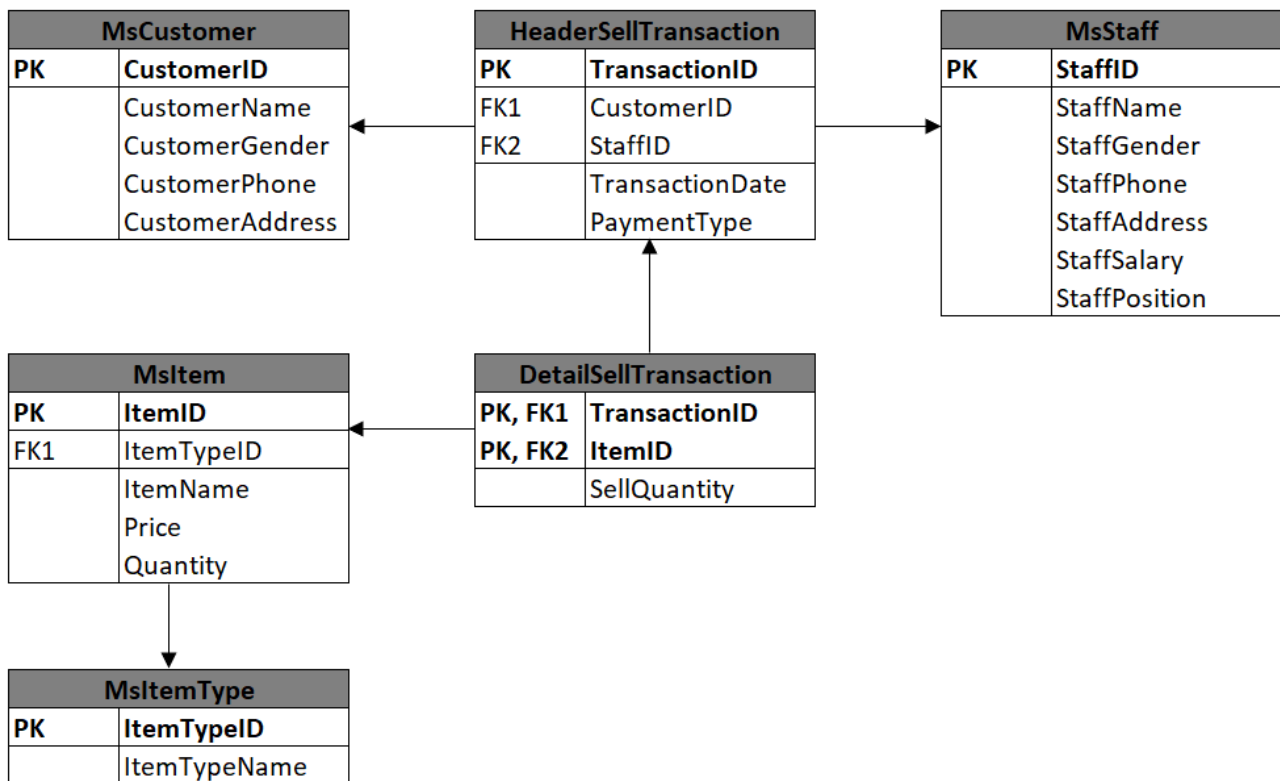
```
select TransactionDate, CustomerName, ItemName, Price
from HeaderSellTransaction hs, DetailSellTransaction ds, MsCustomer mc, MsItem mt, MsStaff ms
where hs.TransactionId = ds.TransactionId
and ms.StaffId = hs.StaffId
and mc.CustomerId = hs.CustomerId
and ds.ItemId = mt.ItemId
and DATENAME(weekday, TransactionDate) = 'Wednesday'
and StaffName like 'Nathan%'
order by TransactionDate, CustomerName
compute SUM(Price) by TransactionDate, CustomerName
```

--10.

```
select TransactionDate, CustomerName, TotalPrice = SUM(Price)
from HeaderSellTransaction hs, MsCustomer mc, DetailSellTransaction ds, MsItem mt
where mc.CustomerId = hs.CustomerId
and hs.TransactionId = ds.TransactionId
and ds.ItemId = mt.ItemId
and DAY(TransactionDate) > 20
group by TransactionDate, CustomerName
order by TransactionDate
compute SUM(SUM(Price)), MIN(SUM(PRICE)) by TransactionDate
```


Pertemuan 4

Join & Union



Soal

1. Tampilkan ItemTypeName, ItemName dan Price **dimana** ItemTypeName mengandung kata 'th' atau diawali huruf 'F' dan harganya kurang dari 500000.

(join, like)

	ItemTypeName	ItemName	Price
1	Furniture	Stove	320000.00
2	Fashion	Blazer	420000.00
3	Fashion	Shirt	320000.00
4	Fashion	Hat	210000.00
5	Health	Vitamin	290000.00
6	Health	Sunblock	125000.00
7	Health	Body lotion	320000.00

2. Tampilkan StaffName dan StaffEmail (didapat dari huruf pertama StaffName ditambah nama terakhir StaffName dan diakhiri kata '@course-net.com' dengan format huruf kecil) **dimana** transaksi terjadi pada hari Rabu dan data yang berulang hanya ditampilkan satu kali. (distinct, lower, left, reverse, charindex, join, datetime, weekday, like)

	StaffName	StaffEmail
1	Andika Leonardo	aleonardo@course-net.com
2	Louis Rudy Valen	lvalen@course-net.com
3	Nathaniel Putera	nputera@course-net.com

3. Tampilkan New Transaction ID (didapat dari TransactionId dengan mengganti kata 'TR' menjadi 'Trans'), Old Transaction ID (di dapat dari TransactionID), TransactionDate, StaffName dan CustomerName **dimana** selisih tanggal transaksi dan tanggal '24 Desember 2017' kurang dari atau sama dengan 2 hari.

(replace, join, datediff, day)

	New Transaction ID	Old Transaction ID	TransactionDate	StaffName	CustomerName
1	Trans007	TR007	2017-12-22	Venky Eduardo	Kerin Augustin
2	Trans008	TR008	2017-12-22	Poppy Hwangsa Iswara	Fernando Lioexander
3	Trans009	TR009	2017-12-22	Andika Leonardo	Arya Thomas

4. Tampilkan New Transaction Date (didapat dari TransactionDate di tambah 5 hari), Old Transaction Date (didapat dari TransactionDate) dan CustomerName **dimana** transaksi terjadi bukan pada tanggal 20.

(dateadd, day, join, datepart)

	New Transaction Date	Old Transaction Date	CustomerName
1	2017-12-26	2017-12-21	Arya Thomas
2	2017-12-26	2017-12-21	Kevin Axellino Triantio
3	2017-12-27	2017-12-22	Kerin Augustin
4	2017-12-27	2017-12-22	Fernando Lioexander
5	2017-12-27	2017-12-22	Arya Thomas

5. Tampilkan Day (didapat dari nama hari TransactionDate), CustomerName dan ItemName **dimana** StaffPosition diawali kata 'Super' atau StaffGender adalah 'Female'. Kemudian urutkan data berdasarkan CustomerName secara ascending.

(datetime, weekday, join, in, like, order by)

	Day	CustomerName	ItemName
1	Friday	Fernando Liexander	Television
2	Friday	Fernando Liexander	Lamp

6. Tampilkan 1 data teratas CustomerId, CustomerName, TransactionId dan Total Item (didapat dari banyaknya Item per transaksi). Dan urutkan data berdasarkan Total Item secara descending.

(top, count, join, group by, order by)

	CustomerId	CustomerName	TransactionId	Total Item
1	CU004	Naufal Hafiz	TR004	4

7. Tampilkan CustomerId, TransactionId, CustomerName, dan Total Price (didapat dari jumlah Price per transaksi) **dimana** Total Price di atas rata-rata Total Price. Kemudian urutkan data berdasarkan Total Price secara descending.

(sum, join, avg, sum, group by, alias subquery, having, order by)

	CustomerId	TransactionId	CustomerName	Total Price
1	CU004	TR004	Naufal Hafiz	15795000.00
2	CU001	TR001	Kevin Axellino Triantio	9950000.00

8. Tampilkan Name (didapat dari StaffName dengan menambahkan kata 'Mr. ' di awal), StaffPosition dan StaffSalary **dimana** StaffGender adalah 'Male'. Kemudian gabungkan dengan Name (didapat dari StaffName dengan menambahkan kata 'Ms. ' di awal), StaffPosition dan StaffSalary **dimana** StaffGender adalah 'Female'. Kemudian urutkan data berdasarkan Name dan StaffPosition secara ascending.

(union, order by)

	Name	StaffPosition	StaffSalary
1	Mr. Andika Leonardo	Cashier	6500000.00
2	Mr. Louis Rudy Valen	Cashier	6000000.00
3	Mr. Nathaniel Putera	Cashier	7300000.00
4	Mr. Venky Eduardo	Manager	15000000.00
5	Ms. Poppy Hwangsa Iswara	Supervisor	10000000.00

9. Tampilkan ItemName, Price (didapat dari Price dengan menambahkan kata 'Rp. ' di awal) dan Status adalah 'Maximum Price' **dimana** Price merupakan Price terbesar dari daftar harga. Kemudian **gabungkan** dengan ItemName, Price (didapat dari Price dengan menambahkan kata 'Rp. ' di awal) dan Status adalah 'Minimum Price' **dimana** Price merupakan Price terkecil dari daftar harga.

(cast, max, alias subquery, union, min)

	ItemName	Price	Status
1	Lamp	Rp. 95000.00	Minimum Price
2	Laptop	Rp. 10050000.00	Maximum Price

10. Tampilkan Longest Name of Staff and Customer (didapat dari CustomerName), Length of Name (didapat dari panjang karakter CustomerName) dan Status adalah 'Customer' **dimana** Length of Name merupakan Length of Name terpanjang. Kemudian **gabungkan** dengan Longest Name of Staff and Customer (didapat dari StaffName), Length of Name (didapat dari panjang karakter StaffName) dan Status adalah 'Staff' **dimana** Length of Name merupakan Length of Name terpanjang.

(len, max, alias subquery, union)

	Longest Name of Staff and Customer	Length of Name	Status
1	Kevin Axellino Triantio	23	Customer
2	Poppy Hwangsa Iswara	20	Staff

Jawaban

--1.

```
select ItemTypeName, ItemName, Price
from MsItem mt
join MsItemType mtt on mt.ItemTypeId = mtt.ItemTypeId
where (ItemTypeName like '%th%' or ItemTypeName like 'F%')
and Price < 500000
```

--2.

```
select distinct StaffName, StaffEmail = LOWER(LEFT(StaffName, 1) +
REVERSE(LEFT(REVERSE(StaffName), CHARINDEX(' ', REVERSE(StaffName))-1))) +
'@course-net.com'
from MsStaff ms join HeaderSellTransaction hs
on ms.StaffId = hs.StaffId
where DATENAME(WEEKDAY, TransactionDate) like 'Wednesday'
```

--3.

```
select [New Transaction ID] = REPLACE(TransactionId, 'TR', 'Trans'), [Old Transaction ID] =
TransactionId, TransactionDate, StaffName, CustomerName
from MsStaff ms join HeaderSellTransaction hs
on ms.StaffId = hs.StaffId
join MsCustomer mc on hs.CustomerId = mc.CustomerId
where DATEDIFF(DAY, TransactionDate, '12/24/2017') = 2
```

--4.

```
select [New Transaction Date] = DATEADD(DAY, 5, TransactionDate), [Old Transaction Date] =
TransactionDate, CustomerName
from HeaderSellTransaction hs join MsCustomer mc
on hs.CustomerId = mc.CustomerId
where DATEPART(DAY, TransactionDate) != 20
```

--5.

```
select [Day] = DATENAME(WEEKDAY, TransactionDate), CustomerName, ItemName
from HeaderSellTransaction hs join MsCustomer mc on hs.CustomerId = mc.CustomerId
join DetailSellTransaction ds on hs.TransactionId = ds.TransactionId
join MsItem mt on ds.ItemId = mt.ItemId
where StaffId in(
    select StaffId from MsStaff
    where (StaffPosition like 'Super%'
    or StaffGender = 'Female')
)
order by CustomerName
```

--6.

```
select top 1 mc.CustomerId, CustomerName, hs.TransactionId, [Total Item] = COUNT(ItemId)
from MsCustomer mc join HeaderSellTransaction hs on mc.CustomerId = hs.CustomerId
join DetailSellTransaction ds on hs.TransactionId = ds.TransactionId
group by mc.CustomerId, CustomerName, hs.TransactionId
order by [Total Item] desc
```

--7.

```
select mc.CustomerId, ds.TransactionId, CustomerName, [Total Price] = SUM(Price)
from MsCustomer mc
join HeaderSellTransaction hs on mc.CustomerId = hs.CustomerId
join DetailSellTransaction ds on hs.TransactionId = ds.TransactionId
join MsItem mt on ds.ItemId = mt.ItemId,
(select AVG([Total Price]) as rata2
    from
        (select TransactionId, SUM(Price) as [Total Price]
         from DetailSellTransaction ds join
             MsItem mt on ds.ItemId = mt.ItemId
         group by TransactionId
        )b
    )c
group by mc.CustomerId, ds.TransactionId, CustomerName, c.rata2
having SUM(Price) > c.rata2
order by [Total Price] desc
```

--8.

```
select Name = 'Mr. ' + StaffName, StaffPosition, StaffSalary
from MsStaff
where StaffGender = 'Male'
union
select Name = 'Ms. ' + StaffName, StaffPosition, StaffSalary
from MsStaff
where StaffGender = 'Female'
order by Name, StaffPosition
```

--9.

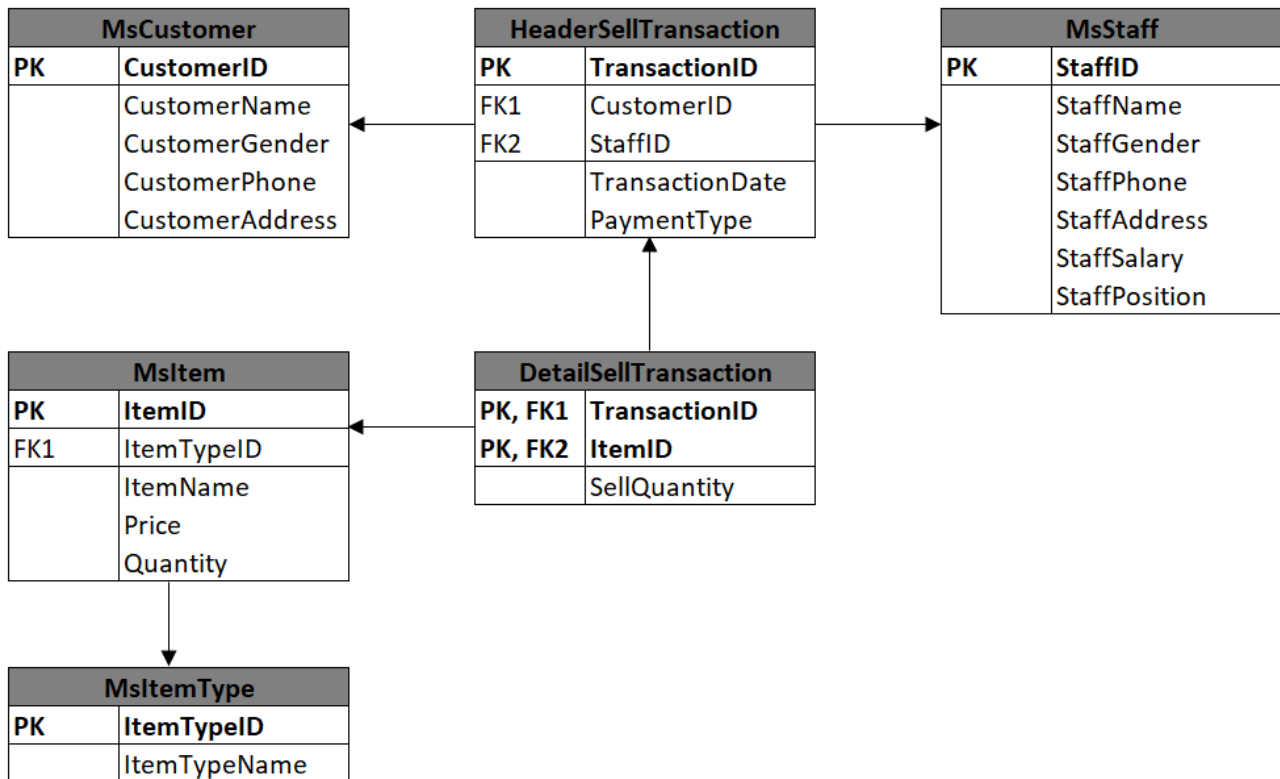
```
select ItemName, Price = 'Rp. ' + CAST(Price as varchar), Status = 'Maximum Price'
from MsItem, (select MAX(Price) as maks from MsItem) as x
where Price = x.maks
union
select ItemName, Price = 'Rp. ' + CAST(Price as varchar), Status = 'Minimum Price'
from MsItem, (select MIN(Price) as mins from MsItem) as x
where Price = x.mins
```

--10.

```
select [Longest Name of Staff and Customer] = CustomerName, [Length of Name] =  
LEN(CustomerName), Status = 'Customer'  
FROM MsCustomer, (select MAX(LEN(CustomerName)) as makslen from MsCustomer) as x  
where LEN(CustomerName) = x.makslen  
union  
select [Longest Name of Staff and Customer] = StaffName, [Length of Name] = LEN(StaffName),  
Status = 'Staff'  
FROM MsStaff, (select MAX(LEN(StaffName)) as makslen from MsStaff) as x  
where LEN(StaffName) = x.makslen
```

Pertemuan 5

Subquery, In, Exists



Soal

1. Tampilkan ItemId dan ItemName **dimana** ItemI adalah 'IM001' atau 'IM002'.
(in)

	ItemId	ItemName
1	IM001	Camera
2	IM002	Television

2. Tampilkan ItemName, Price dan Quantity **dimana** ItemTypeName bukanlah 'Electronic', 'Health' maupun 'Furniture'.
(in, not in)

	ItemName	Price	Quantity
1	Racket	630000.00	22.00
2	Volleyball	420000.00	28.00
3	Golf stick	1900000.00	12.00
4	Basketball	920000.00	14.00
5	Bicycle	2300000.00	25.00
6	Diving	4400000.00	12.00
7	Blazer	420000.00	41.00
8	Shirt	320000.00	64.00
9	Pants	580000.00	52.00
10	Hat	210000.00	58.00

3. Tampilkan CustomerName, CustomerPhone dan CustomerAddress **dimana** Customer tersebut pernah melakukan transaksi pada hari jumat dan panjang karakter CustomerName lebih dari 8 digit.
(in, datename, weekday,len)

	CustomerName	CustomerPhone	CustomerAddress
1	Kerin Augustin	081372116372	Ancol Barat Street no 190
2	Fernando Lioexander	087824153627	Palmerah Street no 20
3	Arya Thomas	081811992617	Anggrek Street no 12

4. Tampilkan ItemTypeName, ItemName dan Price **dimana** CustomerName mengandung kata 'Kerin' dan transaksi terjadi pada tanggal 22.
(in, like, day)

	ItemTypeName	ItemName	Price
1	Electronic	Television	3300000.00
2	Electronic	Rice cooker	950000.00

5. Tampilkan StaffName, CustomerName dan TransactionDate (didapat dari TransactionDate dengan format 'Mon dd, yyyy') **dimana** digit terakhir ItemId merupakan genap.

(convert, exists, right)

	StaffName	CustomerName	TransactionDate
1	Venky Eduardo	Kerin Augustin	Dec 22, 2017
2	Poppy Hwangsa Iswara	Fernando Lioexander	Dec 22, 2017
3	Louis Rudy Valen	Arya Thomas	Dec 21, 2017
4	Andika Leonardo	Arya Thomas	Dec 22, 2017
5	Nathaniel Putera	Kerin Augustin	Dec 20, 2017
6	Nathaniel Putera	Naufal Hafiz	Dec 20, 2017
7	Nathaniel Putera	Kevin Axellino Triantio	Dec 21, 2017

6. Tampilkan CustomerName, CustomerPhone dan CustomerAddress **dimana** Customer tersebut pernah dilayani oleh Staff yang panjang karakter nama-nya merupakan bilangan ganjil.

(exists, len)

	CustomerName	CustomerPhone	CustomerAddress
1	Kevin Axellino Triantio	081267381930	Jelambar Street no 17
2	Kerin Augustin	081372116372	Ancol Barat Street no 190
3	Arya Thomas	081811992617	Anggrek Street no 12

7. Tampilkan ID (didapat dari tiga digit terakhir StaffId) dan Name (didapat dari nama tengah StaffName) **dimana** Staff tersebut tidak pernah melayani Customer berjenis kelamin 'Female' dan StaffName terdiri dari tiga kata atau lebih.

(right, substring, charindex, len, exists, in, not like, like)

	ID	Name
1	002	Hwangsa
2	003	Rudy

8. Tampilkan ItemTypeName, ItemName dan Price **dimana** Price lebih besar dari rata-rata seluruh Price.

(alias subquery, avg)

	Item TypeName	ItemName	Price
1	Electronic	Camera	9000000.00
2	Electronic	Television	3300000.00
3	Electronic	Handphone	5100000.00
4	Electronic	Laptop	10050000.00
5	Electronic	Harddisk	7900000.00
6	Sport	Bicycle	2300000.00
7	Sport	Diving	4400000.00
8	Furniture	Bed	2100000.00

9. Tampilkan StaffName, StaffPosition dan StaffSalary **dimana** StaffSalary merupakan salary tertinggi atau terendah.

(alias subquery, max, min)

	StaffName	StaffPosition	StaffSalary
1	Venky Eduardo	Manager	15000000.00
2	Louis Rudy Valen	Cashier	6000000.00

10. Tampilkan CustomerName, CustomerPhone, CustomerAddress dan Count Item (didapat dari banyaknya Item yang dibeli tiap transaksi) **dimana** Count Item merupakan Count Item terbesar.

(count, alias subquery, max)

	CustomerName	CustomerPhone	CustomerAddress	Count Item
1	Naufal Hafiz	081127173829	Duri Kepa Street no 9	4

Jawaban

--1.

```
select ItemId, ItemName
from MsItem
where ItemId in ('IM001', 'IM002')
```

--2.

```
select ItemName, Price, Quantity
from MsItem mt
where ItemTypeId in(
    select ItemTypeId
    from MsItemType
    where ItemTypeName not in('Electronic', 'Health', 'Furniture')
)
```

--3.

```
select CustomerName, CustomerPhone, CustomerAddress
from MsCustomer
where CustomerId in (
    select CustomerId
    from HeaderSellTransaction hs
    where DATENAME(weekday, TransactionDate) = 'Friday'
)
and LEN(CustomerName) > 8
```

--4.

```
select ItemTypeName, ItemName, Price
from MsItemType mtt, MsItem mt
where mtt.ItemTypeId = mt.ItemTypeId
and ItemId in(
    select ItemId
    from DetailSellTransaction ds, HeaderSellTransaction hs, MsCustomer mc
    where ds.TransactionId = hs.TransactionId
    and hs.CustomerId = mc.CustomerId
    and CustomerName like '%Kerin%'
    and DAY(TransactionDate) = 22
)
```

--5.

```
select StaffName, CustomerName, TransactionDate = CONVERT(varchar, TransactionDate, 107)
from MsStaff ms, HeaderSellTransaction hs, MsCustomer mc
where ms.StaffId = hs.StaffId
and hs.CustomerId = mc.CustomerId
and exists(
    select * from DetailSellTransaction ds
    where hs.TransactionId = ds.TransactionId
    and RIGHT(ds.ItemId, 1) % 2 = 0
)
```

--6.

```
select CustomerName, CustomerPhone, CustomerAddress
from MsCustomer mc
where exists(
    select *
    from HeaderSellTransaction hs, MsStaff ms
    where hs.CustomerId = mc.CustomerId
    and hs.StaffId = ms.StaffId
    and LEN(StaffName) % 2 = 1
)
```

--7.

```
select [ID] = RIGHT(StaffId, 3), Name=Substring(Substring(StaffName,CHARINDEX('
',StaffName)+1,LEN(StaffName)),0,CHARINDEX('
',Substring(StaffName,CHARINDEX('
',StaffName)+1,LEN(StaffName))))
FROM MsStaff ms
where exists(
    select * from HeaderSellTransaction hs
    where hs.StaffId = ms.StaffId
    and CustomerId in(
        select CustomerId
        from MsCustomer
        where CustomerGender not like 'Female'
    )
)
and StaffName like '% % %'
```

--8.

```
select ItemTypeName, ItemName, Price
from MsItemType mtt, MsItem mt, (select AVG(Price) as Rata2 from MsItem) as x
where mtt.ItemTypeId = mt.ItemTypeId
and Price > x.Rata2
```

--9.

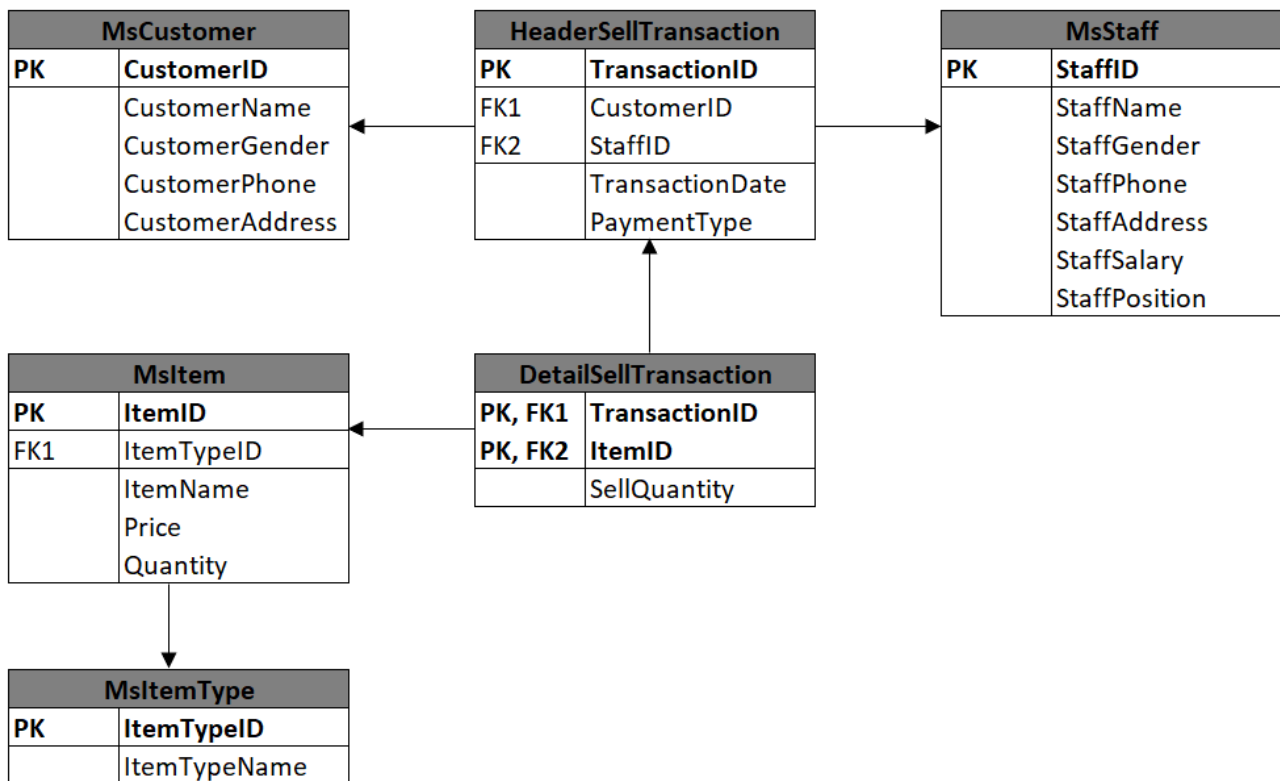
```
select StaffName, StaffPosition, StaffSalary
from MsStaff,
(select MAX(StaffSalary) as maksal, MIN(StaffSalary) as minsal from MsStaff) as x
where (StaffSalary = maksal or StaffSalary = minsal)
```

--10.

```
SELECT CustomerName, CustomerPhone, CustomerAddress, [Count Item]
FROM MsCustomer mc, HeaderSellTransaction hs, (
    SELECT TransactionId, COUNT(ItemId) AS [Count Item]
    FROM DetailSellTransaction
    GROUP BY TransactionId
)a, (
    SELECT MAX([Count Item]) AS [Max Count Item]
    FROM (
        SELECT TransactionId, COUNT(ItemId) AS [Count Item]
        FROM DetailSellTransaction
        GROUP BY TransactionId
    )a
)b
WHERE mc.CustomerId=hs.CustomerId
    AND hs.TransactionId=a.TransactionId
    AND [Count Item]=[Max Count Item]
```

Pertemuan 7

View, Grant & Revoke



Soal

1. Buatlah sebuah *view* dengan nama ViewBonus untuk menampilkan BonusId (didapat dari CustomerId yang 2 huruf pertamanya diganti dengan kata 'BN') dan CustomerName **dimana** panjang kata CustomerName lebih besar dari 10.

(create view, stuff, len)

	BonusId	CustomerName
1	BN001	Kevin Axellino Triantio
2	BN002	Kerin Augustin
3	BN003	Fernando Lioexander
4	BN004	Naufal Hafiz
5	BN005	Arya Thomas

2. Buatlah sebuah *view* dengan nama ViewCustomerData untuk menampilkan Name (didapat dari kata pertama CustomerName), Address (didapat dari CustomerAddress) dan Phone (didapat dari CustomerPhone) **dimana** CustomerName terdiri dari dua kata atau lebih.

(create view, substring, charindex)

	Name	Address	Phone
1	Kevin	Jelambar Street no 17	081267381930
2	Kerin	Ancol Barat Street no 190	081372116372
3	Fernando	Palmerah Street no 20	087824153627
4	Naufal	Duri Kepa Street no 9	081127173829
5	Arya	Anggrek Street no 12	081811992617

3. Buatlah sebuah *view* dengan nama ViewItem untuk menampilkan ItemName, ItemTypeName, Price (didapat dari Price ditambahkan 'Rp. ' di awal) **dimana** ItemTypeName adalah 'Sport' dan Price di antara 200000 dan 1000000.

(create view, cast, between)

	ItemName	ItemTypeName	Price
1	Racket	Sport	Rp. 630000.00
2	Volleyball	Sport	Rp. 420000.00
3	Basketball	Sport	Rp. 920000.00

4. Buatlah sebuah *view* dengan nama ViewTransaction untuk menampilkan StaffName, CustomerName, TransactionDate (didapat dari TransactionDate dengan format 'dd Mon yyyy') dan PaymentType **dimana** tanggal terjadinya transaksi diantara 21 dan 25 dan PaymentType merupakan 'Credit'.

(create view, convert, day, between)

	StaffName	CustomerName	TransactionDate	PaymentType
1	Nathaniel Putera	Kevin Axellino Triantio	21 Dec 2017	Credit
2	Poppy Hwangsa Iswara	Fernando Lioexander	22 Dec 2017	Credit

5. Buatlah sebuah *view* dengan nama ViewBonusCustomer untuk menampilkan BonusId (didapat dari CustomerId dengan menggantikan kata 'CU' dengan kata 'BN'), Name (didapat dari nama terakhir Customer dengan format huruf kecil), Day (didapat dari nama hari terjadinya transaksi) dan TransactionDate (didapat dari TransactionDate dengan format 'mm/dd/yyyy') **dimana** CustomerName terdiri dari 2 kata atau lebih dan Name mengandung huruf 'a'.
(create view, replace, lower, substring, charindex, len, datename, convert, like)

	BonusId	Name	Day	TransactionDate
1	BN001	axellino triantio	Wednesday	12/20/2017
2	BN002	augustin	Wednesday	12/20/2017
3	BN003	lioexander	Wednesday	12/20/2017
4	BN004	hafiz	Wednesday	12/20/2017
5	BN005	thomas	Thursday	12/21/2017
6	BN001	axellino triantio	Thursday	12/21/2017
7	BN002	augustin	Friday	12/22/2017
8	BN003	lioexander	Friday	12/22/2017
9	BN005	thomas	Friday	12/22/2017

6. Buatlah sebuah *view* dengan nama ViewTransactionByLouis yang menampilkan TransactionId, Date (didapat dari TransactionDate dengan format 'Mon dd, yyyy'), dan ItemName **dimana** Staff yang menangani transaksi bernama 'Louis Rudy Valen' dan transaksi terjadi pada tanggal 21.
(create view, convert, like, day)

	TransactionId	Date	ItemName
1	TR005	Dec 21, 2017	Diving

7. Ubahlah *view* ViewCustomerData menjadi menampilkan ID (didapat dari 3 digit terakhir CustomerId), Name (didapat dari kata CustomerName), Address (didapat dari CustomerAddress) dan Phone (didapat dari CustomerPhone) **dimana** CustomerName terdiri dari dua kata atau lebih.
(alter, right, charindex)

	ID	Name	Address	Phone
1	001	Kevin Axellino Triantio	Jelambar Street no 17	081267381930
2	002	Kerin Augustin	Ancol Barat Street no 190	081372116372
3	003	Fernando Lioexander	Palmerah Street no 20	087824153627
4	004	Naufal Hafiz	Duri Kepa Street no 9	081127173829
5	005	Arya Thomas	Anggrek Street no 12	081811992617

8. Buatlah sebuah *view* dengan nama ViewCustomer untuk menampilkan CustomerId, CustomerName dan CustomerGender. Kemudian tambahkan data ke ViewCustomer dengan spesifikasi tabel di bawah ini:

CustomerId	CustomerName	CustomerGender
CU006	Cristian	Male

Kemudian tampilkan seluruh data dari MsCustomer.

	CustomerId	CustomerName	CustomerGender	CustomerPhone	CustomerAddress
1	CU001	Kevin Axellino Triantio	Male	081267381930	Jelambar Street no 17
2	CU002	Kerin Augustin	Female	081372116372	Ancol Barat Street no 190
3	CU003	Femando Lioexander	Male	087824153627	Palmerah Street no 20
4	CU004	Naufal Hafiz	Male	081127173829	Duri Kepa Street no 9
5	CU005	Arya Thomas	Male	081811992617	Anggrek Street no 12
6	CU006	Cristian	Male	NULL	NULL

9. Hapuslah data yang berada di dalam *view* ViewCustomerData yang memiliki ID '005'. Kemudian tampilkan seluruh data pada MsCustomer.

Sebelum di Delete:

	ID	Name	Address	Phone
1	001	Kevin Axellino Triantio	Jelambar Street no 17	081267381930
2	002	Kerin Augustin	Ancol Barat Street no 190	081372116372
3	003	Femando Lioexander	Palmerah Street no 20	087824153627
4	004	Naufal Hafiz	Duri Kepa Street no 9	081127173829
5	005	Arya Thomas	Anggrek Street no 12	081811992617

Setelah di Delete:

	ID	Name	Address	Phone
1	001	Kevin Axellino Triantio	Jelambar Street no 17	081267381930
2	002	Kerin Augustin	Ancol Barat Street no 190	081372116372
3	003	Femando Lioexander	Palmerah Street no 20	087824153627
4	004	Naufal Hafiz	Duri Kepa Street no 9	081127173829

10. Hapuslah *view* ViewCustomerData.
11. Buatlah sebuah *login* dengan nama 'ManagerUser' yang memiliki password 'manager'.
(create login)
12. Buatlah sebuah *database user* dengan nama 'Manager' untuk *login* 'ManagerUser'.
(create user)
13. Buatlah sebuah *login* dengan nama 'EmployeeUser' yang memiliki password 'employee'.
(create login)
14. Buatlah sebuah *database user* dengan nama 'Employee' untuk *login* 'EmployeeUser'.
(create user)
15. Berilah hak akses kepada publik (*select, insert, update, delete, dll*) terhadap tabel MsItem.
(grant all)

16. Berilah hak akses kepada 'Manager' untuk melakukan *insert*, *update* dan *delete* terhadap tabel MsStaff dan 'Manager' juga dapat memberikan hak akses pada tabel MsStaff kepada user lainnya.
(**grant**)
17. Berilah hak akses kepada 'Manager' untuk melakukan *select* terhadap tabel MsItemType.
(**grant**)
18. Berilah hak akses kepada 'Employee' untuk melakukan *select* terhadap tabel MsItem.
(**grant**)
19. Cabutlah semua hak akses tabel MsItem dari publik.
(**revoke**)
20. Cabutlah hak akses untuk *select* pada tabel MsItemType dari 'Manager'.
(**revoke**)

Jawaban

--1

```
CREATE VIEW ViewBonus
AS
SELECT
    BinusId = STUFF(CustomerID, 1, 2, 'BN'),
    CustomerName
FROM
    MsCustomer
WHERE
    LEN(CustomerName) > 10
Select * from ViewBonus
```

--2

```
CREATE VIEW ViewCustomerData
AS
SELECT
    Name = SUBSTRING(CustomerName, 0, CHARINDEX(' ', CustomerName)),
    Address = CustomerAddress,
    Phone = CustomerPhone
FROM
    MsCustomer
WHERE
    CHARINDEX(' ', CustomerName) > 0
```

```
SELECT * FROM ViewCustomerData
```

--3

```
CREATE VIEW ViewItem
AS
SELECT
    ItemName,
    ItemTypeName,
    Price = 'Rp. ' + CAST(Price AS VARCHAR)
FROM
    MsItem mt,
    MsItemType mtt
WHERE
    mt.ItemTypeId = mtt.ItemTypeId
    AND ItemTypeName = 'Sport'
    AND Price BETWEEN 200000 AND 1000000
```

```
SELECT ItemName FROM ViewItem
```

--4

CREATE VIEW ViewTransaction

AS

SELECT

StaffName,

CustomerName,

TransactionDate = CONVERT(varchar, TransactionDate, 106),

PaymentType

from

MsStaff ms,

HeaderSellTransaction hs,

MsCustomer mc

where

ms.StaffId = hs.StaffId

AND hs.CustomerId = mc.CustomerId

AND DAY(TransactionDate) BETWEEN 21 AND 25

AND PaymentType = 'Credit'

select * from ViewTransaction

--5

CREATE VIEW ViewBonusCustomer

AS

SELECT

BonusId = REPLACE(mc.CustomerId, 'CU', 'BN'),

Name = LOWER(SUBSTRING(CustomerName, CHARINDEX(' ', CustomerName)+1,

LEN(CustomerName)-1)),

[Day] = DATENAME(WEEKDAY, TransactionDate), TransactionDate =

CONVERT(VARCHAR, TransactionDate, 101)

FROM

MsCustomer mc,

HeaderSellTransaction hs

WHERE

mc.CustomerId = hs.CustomerId

and CHARINDEX(' ', CustomerName) > 0

and SUBSTRING(CustomerName, CHARINDEX(' ', CustomerName)+1,

LEN(CustomerName)-1) like '%a%'

select * from ViewBonusCustomer

--6

```
CREATE VIEW ViewTransactionByLouis
AS
SELECT
    hss.TransactionId,
    Date = CONVERT(VARCHAR, TransactionDate, 107),
    ItemName
FROM
    MsStaff ms,
    HeaderSellTransaction hss,
    DetailSellTransaction dss,
    MsItem mt
WHERE
    ms.StaffId = hss.StaffId
    AND hss.TransactionId = dss.TransactionId
    AND mt.ItemId = dss.ItemId
    AND StaffName LIKE 'Louis Rudy Valen'
    AND DAY(TransactionDate) = 21
```

```
select * from ViewTransactionByLouis
```

--7

```
ALTER VIEW ViewCustomerData
AS
SELECT
    ID = RIGHT(CustomerID, 3),
    Name = CustomerName,
    Address = CustomerAddress,
    Phone = CustomerPhone
FROM
    MsCustomer
WHERE
    CHARINDEX(' ', CustomerName) > 0
```

```
SELECT * FROM ViewCustomerData
```

--8

```
CREATE VIEW ViewCustomer
AS
SELECT CustomerId, CustomerName, CustomerGender
FROM MsCustomer

BEGIN TRAN
INSERT INTO ViewCustomer VALUES('CU006', 'Cristian', 'Male')
SELECT * FROM MsCustomer
ROLLBACK
```

```
--9
BEGIN TRAN
SELECT * FROM ViewCustomerData

DELETE FROM ViewCustomerData
WHERE ID = '005'

SELECT * FROM ViewCustomerData
ROLLBACK

--10
DROP VIEW ViewCustomerData

--11.
CREATE LOGIN ManagerUser with password='manager'

--12.
CREATE USER Manager FOR ManagerUser

--13.
CREATE LOGIN EmployeeUser with password='employee'

--14.
CREATE USER Employee FOR EmployeeUser

--15.
GRANT ALL PRIVILEGES ON MsItem TO PUBLIC

--16.
GRANT INSERT, UPDATE, DELETE ON MsStaff TO Manager WITH GRANT OPTION

--17.
GRANT SELECT ON MsItemType TO Manager

--18.
GRANT SELECT ON MsItem TO Employee

--19.
revoke ALL PRIVILEGES ON MsItem FROM PUBLIC

--20.
revoke SELECT ON MsItemType FROM Manager CASCADE
```

Pertemuan 8 Normalization

Ada 4 komponen pada ER Modelling :

- Entity / Entity type → sekumpulan objek yang mempunyai property yang sama yang mengidentifikasi perusahaan.
- Relationship → hubungan antar entity, digambarkan dengan garis.
- Attribute → kolom – kolom pada table.
- Multiplicity → banyaknya hubungan antar records dalam 2 entitas.

Enhacement type pada ER models :

- Generalization → menggabung subclass menjadi table.
- Specialization → memecah table menjadi beberapa subclass Karena perbedaan role.
- Constraint : {Optional, Or}, {Mandatory, And}, {Mandatory, Or} :
 - Participants → jumlah member yang berpartisipasi.
 - Disjoint constraint → jumlah member yang ada dalam subclass.
 - Optional → boleh ada, boleh tidak.
 - Mandatory → harus ada.
 - AND → boleh lebih dari 1 tabel. (Nondisjoint)
 - OR → boleh 1 tabel. (Disjoint)
- Aggregation → hubungan antar 2 tabel yang dapat saling bergantung atau tidak. Terjadi jika hubungannya has (memiliki) dan offer (menawarkan).
- Composition → menentukan hubungan dependency satu sama lain. Misal A tidak bisa dianggap A jika tanpa B.

Normalisasi → teknik untuk menghasilkan table – table yang sesuai atau dibutuhkan oleh perusahaan.


Fungsi normalisasi :

- Table mudah diakses dan di maintenance → dalam normalisasi, menghasilkan table master.
- Agar meminimalkan kapasitas komputer kita → meminimalkan redundancy.

Kondisi normalisasi (semakin besar, semakin ternormalisasi / normal) :

- UNF → table yang berisi satu atau lebih repeating group.
- 1NF → relasi di mana tiap kolom dan baris hanya mengandung satu nilai. Di dapat dari eliminasi repeating group dari UNF.
- 2NF → suatu hubungan pada 1NF dan setiap atribut yang bukan PK fungsional berkaitan pada PK. Di dapat dari eliminasi partial dependencies.
- 3NF → hubungan dalam bentuk 1NF dan 2NF tidak ada atribut yang merupakan PK yang berkaitan kuat dengan PK. Di dapat dari eliminasi transitive dependencies.
- BCNF/3.5NF → semua kolom candidate key.
- 4NF → ada multiple value.
- 5NF

Case 1:



Laundry Receipt
 Kemanggisan Street Number 16
 (021) - 5014986

SalesID : SA001	Customer Name : Justinus Hermawan
Date : 1 Dec 2015	Phone : 08999465851
Payment Type : Credit	Address : Anggrek Street 6

No	ItemID	ItemName	Price	Weight	SubTotal
1	IT001	Pants	Rp 20.000.00	1.5 Kg	Rp 30.000.00
2	IT002	Shirt	Rp 15.000.00	1.0 Kg	Rp 15.000.00
Total					Rp 45.000.00

 Cashier (Komala Surya)

UNF:

*Masukkan semua data yang ada di faktur/struk

Logo, StoreName, Address, Phone,

SalesID, Date, PaymentType,

CustomerName, Phone, Address,

{ no, ItemId, ItemName, Price, Weight, SubTotal }, Total, cashierName

1NF:

*pisahkan data yang berulang dengan data yang tidak berulang

*copy semua data berulang ke Detail, dan sisanya ke Header

*Buang semua data yang tidak perlu

*perhitungan tidak perlu dimasukkan ke database

*tentukan PK, kemudian tentukan FK

Header (SalesID, Date, PaymentType, CustomerName, Phone, Address, cashierName)

Detail (SalesID, ItemId, ItemName, Price, Weight)

2NF:

*(focus pada table yang memiliki 2 PK/FK)

* remove partial dependency | pecah key yang non-relatable menjadi table yang lain (yang bergantung pada satu PK)

Header (SalesID, Date, PaymentType, CustomerName, Phone, Address, cashierName)

Detail (SalesID, itemId)

Item(ItemId, ItemName, Price, Weight)

3NF:

*remove transitive dependency

Header (SalesID, Date, PaymentType, cashierName)

Detail (SalesID, ItemId)

Item(ItemId, ItemName, Price, Weight)


Cust(CustomerName, Phone, Address)

Case 2:

1					
2	KodeTrans : T001		ID Kasir : K001		
3	TglTrans : 7-Mei-14		Nama Kasir : Adi		
4					
5	KodeBrg	NamaBrg	Qty	Harga	Total
6	P001	Pensil	2	1000	2000
7	P002	Stabilo	1	5000	5000
8	Grand Total				7000
9					
10					
11	Create UNF, NF1, NF2, NF3				
12					

Case 3:





Nama Pemesan: Wil
Tanggal Pemesanan: Jumat, 25 September 2015

E-Tiket

Kode Booking: **3M4TXX**

PENERBANGAN	KEBERANGKATAN	KEDATANGAN
GA 640/GA 7640 Class H 2 Transit <small>Flangan pemberangkatan di halaman 2</small>	23:45 Selasa, 6 Oktober 2015 Jakarta - Cengkareng (CGK) Soekarno Hatta International Airport	17:20 Rabu, 7 Oktober 2015 Tual (LUV) Tual, Dumatubin Airport

DETIL PENUMPANG

No.	Titel	Nama Penumpang	Jenis Tiket	Bagasi	Nomor Tiket
1.	Mrs	Nore	Dewasa	20 Kg	126247275XXXX

PT GARUDA INDONESIA (PERSERO) Tbk. (NPWP : 01.001.634.3-093.000) | JL. KEBON SIRIH NO 44 GAMBIR JAKARTA-PUSAT

Pertemuan 9

Transaction

Properti transaksi yang committed ada 4 kondisi / karakteristik :

- Atomicity → semua / tidak sama sekali. Misalkan kirim 2 data, apabila ada 1 data yang tidak commit, maka gagal.
- Consistency → perubahan database konsisten, contoh kolom NIM, yang berubah datanya, tetapi strukturnya tetap.
- Isolation → semua transaksi tidak bisa melihat ada transaksi lain, hanya menjalankan yang dia lakukan.
- Durability → ketahanan, kalau sudah commit, maka sifatnya permanen, tidak akan hilang.

Concurrency control → proses me-manage operasi database agar tidak bertabrakan (mengatur lalu lintas data) / agar tidak tumpang tindih.

3 masalah disebabkan oleh concurrency :

- Masalah lost update → masalah dimana 1 data / transaksi overwrite / terupdate data atau transaksi lain.

Time	T ₁	T ₂	bal _x
t ₁		begin_transaction	100
t ₂	begin_transaction	read(bal _x)	100
t ₃	read(bal _x)	bal _x = bal _x + 100	100
t ₄	bal _x = bal _x - 10	write(bal _x)	200
t ₅	write(bal _x)	commit	90
t ₆	commit		90

- Uncommitted dependency problem → satu transaksi berjalan, kemudian dibatalkan sebelum di commit (rollback).

Time	T ₃	T ₄	bal _x
t ₁		begin_transaction	100
t ₂		read(bal _x)	100
t ₃		bal _x = bal _x + 100	100
t ₄	begin_transaction	write(bal _x)	200
t ₅	read(bal _x)	:	200
t ₆	bal _x = bal _x - 10	rollback	100
t ₇	write(bal _x)		190
t ₈	commit		190

- Inconsistent analysis problem → jika menganalisa transaksi, ternyata hasil salah Karena ada transaksi lain saat kita menganalisa.

Time	T ₅	T ₆	bal _x	bal _y	bal _z	sum
t ₁		begin_transaction	100	50	25	
t ₂	begin_transaction	sum = 0	100	50	25	0
t ₃	read(bal _x)	read(bal _x)	100	50	25	0
t ₄	bal _x = bal _x - 10	sum = sum + bal _x	100	50	25	100
t ₅	write(bal _x)	read(bal _y)	90	50	25	100
t ₆	read(bal _z)	sum = sum + bal _y	90	50	25	150
t ₇	bal _z = bal _z + 10		90	50	25	150
t ₈	write(bal _z)		90	50	35	150
t ₉	commit	read(bal _z)	90	50	35	150
t ₁₀		sum = sum + bal _z	90	50	35	185
t ₁₁		commit	90	50	35	185

Cara mengatasi masalah pada transaksi :

- Serializability → memberikan urutan / schedule pada transaksi, kapan harus jalan / berhenti. Berjalan secara serial (bergantian), kapan read / kapan update. Tidak menjamin hasil akan sama / benar.
 - Serial schedule → schedule yang diberikan pada transaksi agar tidak interleaved / tidak saling mendahului.
 - Tujuan → mencari transaksi non serial agar menjadi serial.
 - Ketentuan :
 - Jika ada 2 transaksi, membaca akun yang sama.
 - Jika 2 transaksi, yang 1 baca / update data yang berbeda, tidak konflik.
 - Jika transaksi, yang 1 baca dan 1 update data yang sama, urutan penting untuk diperhatikan.
 - Contoh :

Time	T ₁₁	T ₁₂	T ₁₃
t ₁	begin_transaction		
t ₂	read(bal _x)		
t ₃		begin_transaction	
t ₄		write(bal _x)	
t ₅		commit	
t ₆	write(bal _x)		
t ₇	commit		
t ₈			begin_transaction
t ₉			write(bal _x)
t ₁₀			commit

- Locking → transaksi diberikan kunci untuk menolak akses ke transaksi lain. Setelah di lock menjadi serial (diatur).
 - Aturan :
 - Shared lock → hanya bisa baca.
 - Exclusive lock → bisa baca dan update.
 - Membaca / reads tidak konflik.
 - Exclusive lock memberi akses eksklusif ke data.
 - Upgrade → shared lock ke exclusive lock.
 - Downgrade → exclusive lock ke shared lock.
 - Tahapan / fase :
 - Growing → locking membuat gembok.
 - Shrinking → locking membuat kunci agar bisa melepas gembok.
 - Rollback / commit dianggap melepas kunci.
 - Cascading rollback → jika 1 rollback, maka rollback semua transaksi.
 - Deadlock → 2 transaksi / lebih saling menunggu transaksi lain. Cara menyelesaikan : di abort salah satu.
 - Timeout → menunggu kunci dibuka.
 - Contoh :

Time	T ₁₇	T ₁₈
t ₁	begin_transaction	
t ₂	write_lock(bal_x)	begin_transaction
t ₃	read(bal_x)	write_lock(bal_y)
t ₄	bal_x = bal_x - 10	read(bal_y)
t ₅	write(bal_x)	bal_y = bal_y + 100
t ₆	write_lock(bal_y)	write(bal_y)
t ₇	WAIT	write_lock(bal_x)
t ₈	WAIT	WAIT
t ₉	WAIT	WAIT
t ₁₀	:	WAIT
t ₁₁	:	:

- Time Stamping → memberikan identitas unik kepada transaksi berupa waktu kapan transaksi dimulai. Diurutkan berdasarkan waktu.

Time	Op	T ₁₉	T ₂₀	T ₂₁
t ₁		begin_transaction		
t ₂	read(bal_x)	read(bal_x)		
t ₃	bal_x = bal_x + 10	bal_x = bal_x + 10		
t ₄	write(bal_x)	write(bal_x)	begin_transaction	
t ₅	read(bal_y)		read(bal_y)	
t ₆	bal_y = bal_y + 20		bal_y = bal_y + 20	begin_transaction
t ₇	read(bal_y)			read(bal_y)
t ₈	write(bal_y)		write(bal_y) ⁺	
t ₉	bal_y = bal_y + 30			bal_y = bal_y + 30
t ₁₀	write(bal_y)			write(bal_y)
t ₁₁	bal_z = 100			bal_z = 100
t ₁₂	write(bal_z)			write(bal_z)
t ₁₃	bal_z = 50	bal_z = 50		commit
t ₁₄	write(bal_z)	write(bal_z) [‡]	begin_transaction	
t ₁₅	read(bal_y)	commit	read(bal_y)	
t ₁₆	bal_y = bal_y + 20		bal_y = bal_y + 20	
t ₁₇	write(bal_y)		write(bal_y)	
t ₁₈			commit	

⁺ At time t₈, the write by transaction T₂₀ violates the first timestamping write rule described above and therefore is aborted and restarted at time t₁₄.

[‡] At time t₁₄, the write by transaction T₁₉ can safely be ignored using the ignore obsolete write rule, as it would have been overwritten by the write of transaction T₂₁ at time t₁₂.