2502002785 – Jayson Cornelis

ESSAY

1. The main function of maintaining database security : To protect databases from any big or small threats and risks, intentional or accidental like malicious, breaches, hackers, virus, theft, etc. So, databases are protected, have a great privacy, integrity, etc. Not only databases, this security affects to the hardware, software, and people.
   Steps that can be taken to do that :
   - Grant the access to specific persons and protect them from outsiders
   - Don't access the suspicious link that will lead you to phising, money laundering offer, suspicious apps, etc.
   - Prepare the backup in any situation (backup the data and prepare the electricity replacement)
   - Check the credibility of every chats and calls from outsiders to prevent the suspicious things
   - Check physical databases location like adding cameras, locks, or staff security
   - Install proxy server to database that encrypts data travelling
   - Encrypt them using the strong-complicated password
   - 2FA
   - Use firewalls and antivirus (subscribe to features are extremely recommended)
   - Always update your application

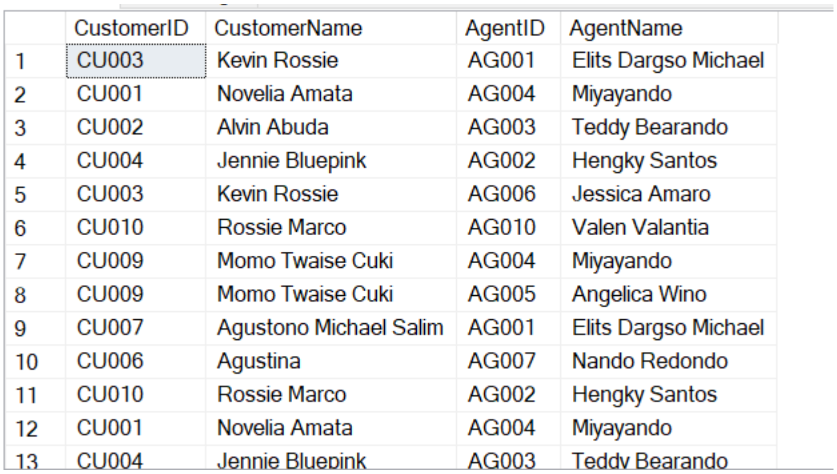
Reference :

Fanchi, C. (2021, December 10). 8 critical database security best practices to Keep Your Data Safe. Backendless. Retrieved July 12, 2022, from https://backendless.com/database-security-best-practices/

tripwire. (2021, February 24). 10 database security best practices you should know. The State of Security. Retrieved July 12, 2022, from https://www.tripwire.com/state-of-security/featured/database-security-best-practices-you-should-know/

2. tripwire. (2021, February 24). 10 database security best practices you should know. The State of Security. Retrieved July 12, 2022, from https://www.tripwire.com/state-of-security/featured/database-security-best-practices-you-should-know/ Join and Union
   a. Join : Combining two different tables and forming the rows into one table and the other selected table becoming the new column rather than combined into rows.
   Example :

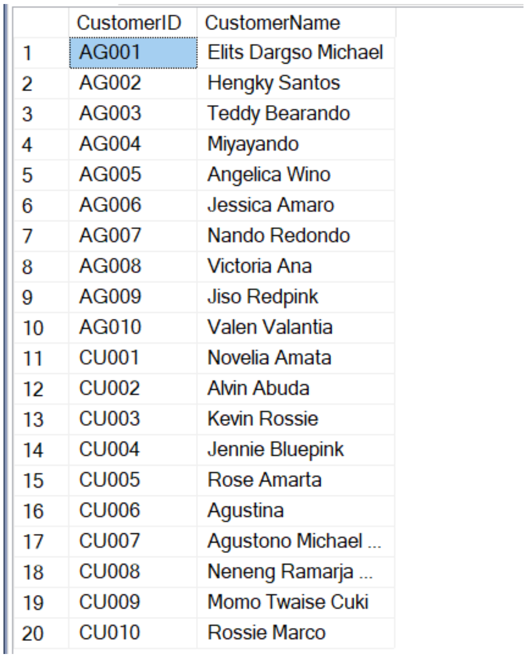
```
select c.CustomerID, CustomerName, a.AgentID, AgentName from
MsCustomer c
join HeaderTransaction ht on ht.CustomerID = c.CustomerID
join MsAgent a on a.AgentID = ht.AgentID
```



	CustomerID	CustomerName	AgentID	AgentName
1	CU003	Kevin Rossie	AG001	Elits Dargso Michael
2	CU001	Novelia Amata	AG004	Miyayando
3	CU002	Alvin Abuda	AG003	Teddy Bearando
4	CU004	Jennie Bluepink	AG002	Hengky Santos
5	CU003	Kevin Rossie	AG006	Jessica Amaro
6	CU010	Rossie Marco	AG010	Valen Valantia
7	CU009	Momo Twaise Cuki	AG004	Miyayando
8	CU009	Momo Twaise Cuki	AG005	Angelica Wino
9	CU007	Agustono Michael Salim	AG001	Elits Dargso Michael
10	CU006	Agustina	AG007	Nando Redondo
11	CU010	Rossie Marco	AG002	Hengky Santos
12	CU001	Novelia Amata	AG004	Miyayando
13	CU004	Jennie Bluepink	AG003	Teddy Bearando

b. Union : Combining two tables (either same or different) and containing all rows from them into a lot of rows
Example :

select CustomerID, CustomerName from
MsCustomer
union
select AgentID, AgentName from
MsAgent



	CustomerID	CustomerName
1	AG001	Elits Dargso Michael
2	AG002	Hengky Santos
3	AG003	Teddy Bearando
4	AG004	Miyayando
5	AG005	Angelica Wino
6	AG006	Jessica Amaro
7	AG007	Nando Redondo
8	AG008	Victoria Ana
9	AG009	Jiso Redpink
10	AG010	Valen Valantia
11	CU001	Novelia Amata
12	CU002	Alvin Abuda
13	CU003	Kevin Rossie
14	CU004	Jennie Bluepink
15	CU005	Rose Amarta
16	CU006	Agustina
17	CU007	Agustono Michael ...
18	CU008	Neneng Ramarja ...
19	CU009	Momo Twaise Cuki
20	CU010	Rossie Marco

Connolly T., & Begg C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management, Global Edition. [VitalSource Bookshelf]. Retrieved from https://online.vitalsource.com/#/books/9781292061849/

3. Data warehouse characteristics :
    - Subject-oriented data : organized around any roles of the enterprise (customers, products, and sales) rather than application areas (invoicing, stock, product, etc.) to make various decisions.
    - Integrated Data : integrating data from any sources of data (mainframe and relational), but that data mostly are inconsistent. So, labelling, codes, and formats are important for analysis of the data.
    - Time-variant data : update, alter, or modify are normal things in managing database and needs the proof with timing. Time is so different and has variants to maintain the data implicit or explicit
    - Non-volatile data : data got refreshed by OS rather than real-time update. Data also is read-only for analytics.

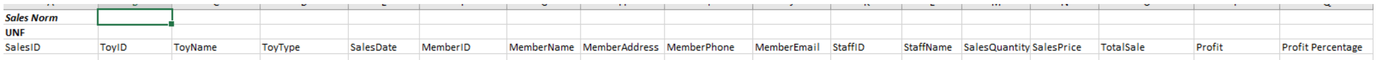
References :

Connolly T., & Begg C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management, Global Edition. [VitalSource Bookshelf]. Retrieved from https://online.vitalsource.com/#/books/9781292061849/

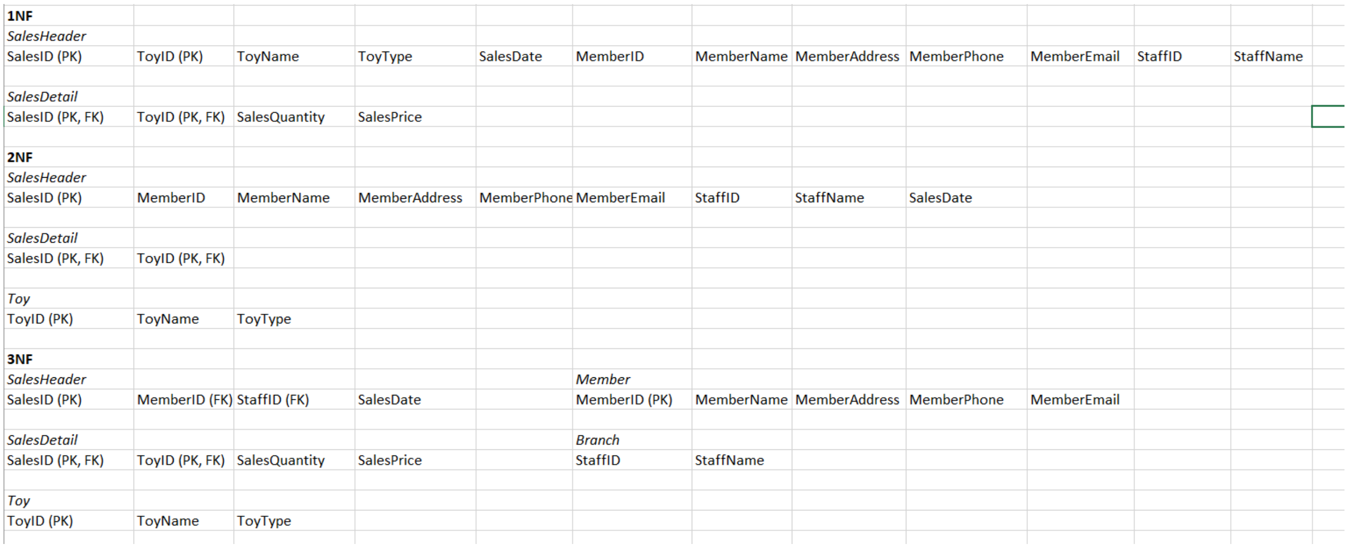
Sharma, M. (2018, October 22). Characteristics and functions of Data Warehouse. GeeksforGeeks. Retrieved July 12, 2022, from https://www.geeksforgeeks.org/characteristics-and-functions-of-data-warehouse/

CASE

4. Sales Normalization



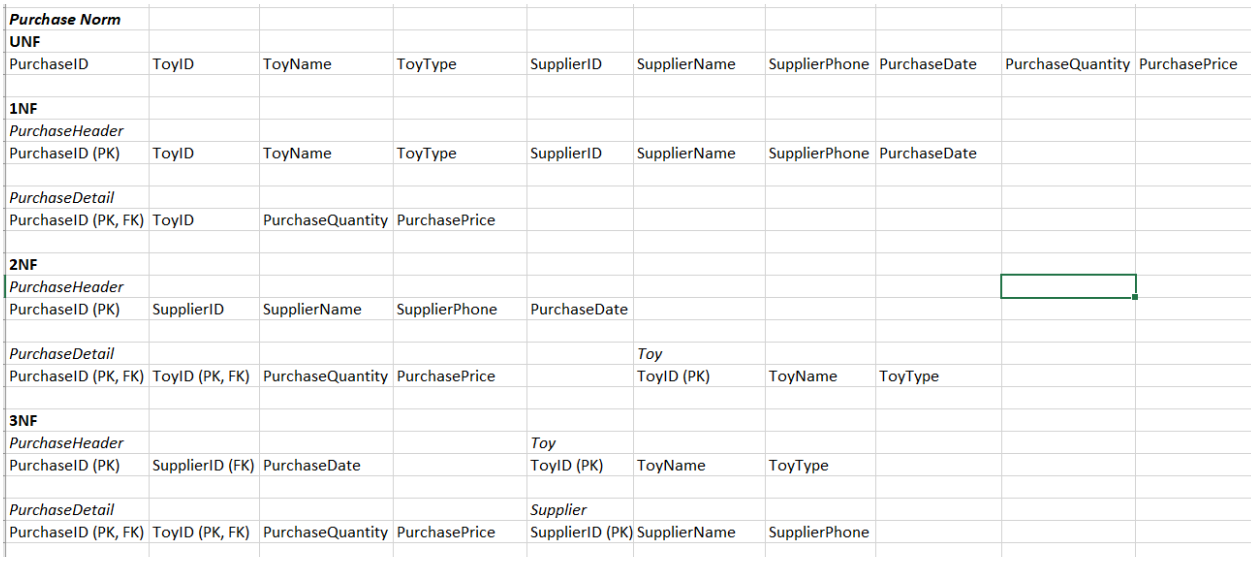
Sales Norm																
UNF																
SalesID	ToyID	ToyName	ToyType	SalesDate	MemberID	MemberName	MemberAddress	MemberPhone	MemberEmail	StaffID	StaffName	SalesQuantity	SalesPrice	TotalSale	Profit	Profit Percentage



1NF											
SalesHeader											
SalesID (PK)	ToyID (PK)	ToyName	ToyType	SalesDate	MemberID	MemberName	MemberAddress	MemberPhone	MemberEmail	StaffID	StaffName
SalesDetail											
SalesID (PK, FK)	ToyID (PK, FK)	SalesQuantity	SalesPrice								
2NF											
SalesHeader											
SalesID (PK)	MemberID	MemberName	MemberAddress	MemberPhone	MemberEmail	StaffID	StaffName	SalesDate			
SalesDetail											
SalesID (PK, FK)	ToyID (PK, FK)										
Toy											
ToyID (PK)	ToyName	ToyType									
3NF											
SalesHeader					*Member*						
SalesID (PK)	MemberID (FK)	StaffID (FK)	SalesDate		MemberID (PK)	MemberName	MemberAddress	MemberPhone	MemberEmail		
SalesDetail					*Branch*						
SalesID (PK, FK)	ToyID (PK, FK)	SalesQuantity	SalesPrice		StaffID	StaffName					
Toy											
ToyID (PK)	ToyName	ToyType									

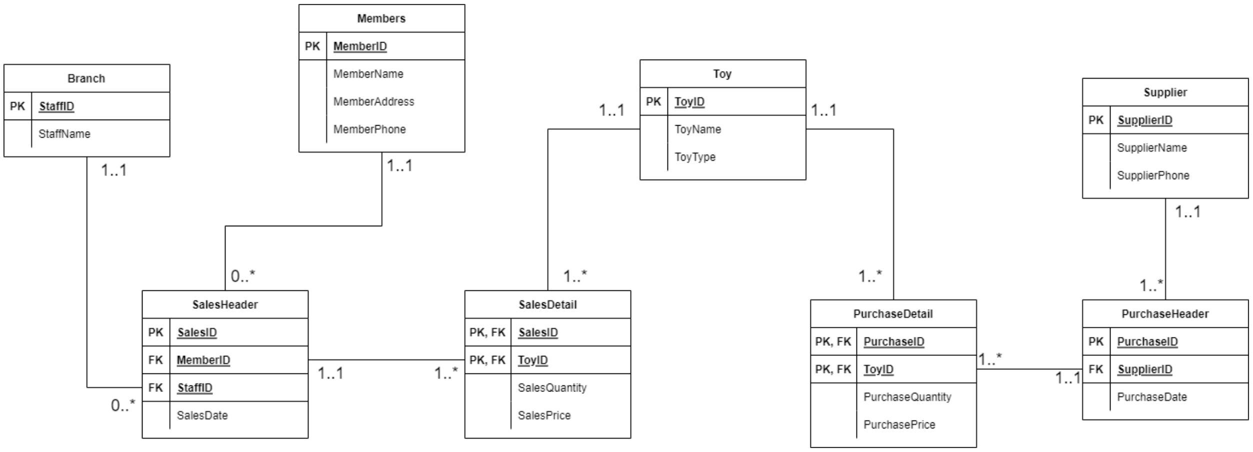
2502002785 – Jayson Cornelis

Purchase Normalization



Purchase Norm									
UNF									
PurchaseID	ToyID	ToyName	ToyType	SupplierID	SupplierName	SupplierPhone	PurchaseDate	PurchaseQuantity	PurchasePrice
1NF									
PurchaseHeader									
PurchaseID (PK)	ToyID	ToyName	ToyType	SupplierID	SupplierName	SupplierPhone	PurchaseDate		
PurchaseDetail									
PurchaseID (PK, FK)	ToyID	PurchaseQuantity	PurchasePrice						
2NF									
PurchaseHeader									
PurchaseID (PK)	SupplierID	SupplierName	SupplierPhone	PurchaseDate					
PurchaseDetail					*Toy*				
PurchaseID (PK, FK)	ToyID (PK, FK)	PurchaseQuantity	PurchasePrice		ToyID (PK)	ToyName	ToyType		
3NF									
PurchaseHeader				*Toy*					
PurchaseID (PK)	SupplierID (FK)	PurchaseDate		ToyID (PK)	ToyName	ToyType			
PurchaseDetail				*Supplier*					
PurchaseID (PK, FK)	ToyID (PK, FK)	PurchaseQuantity	PurchasePrice	SupplierID (PK)	SupplierName	SupplierPhone			

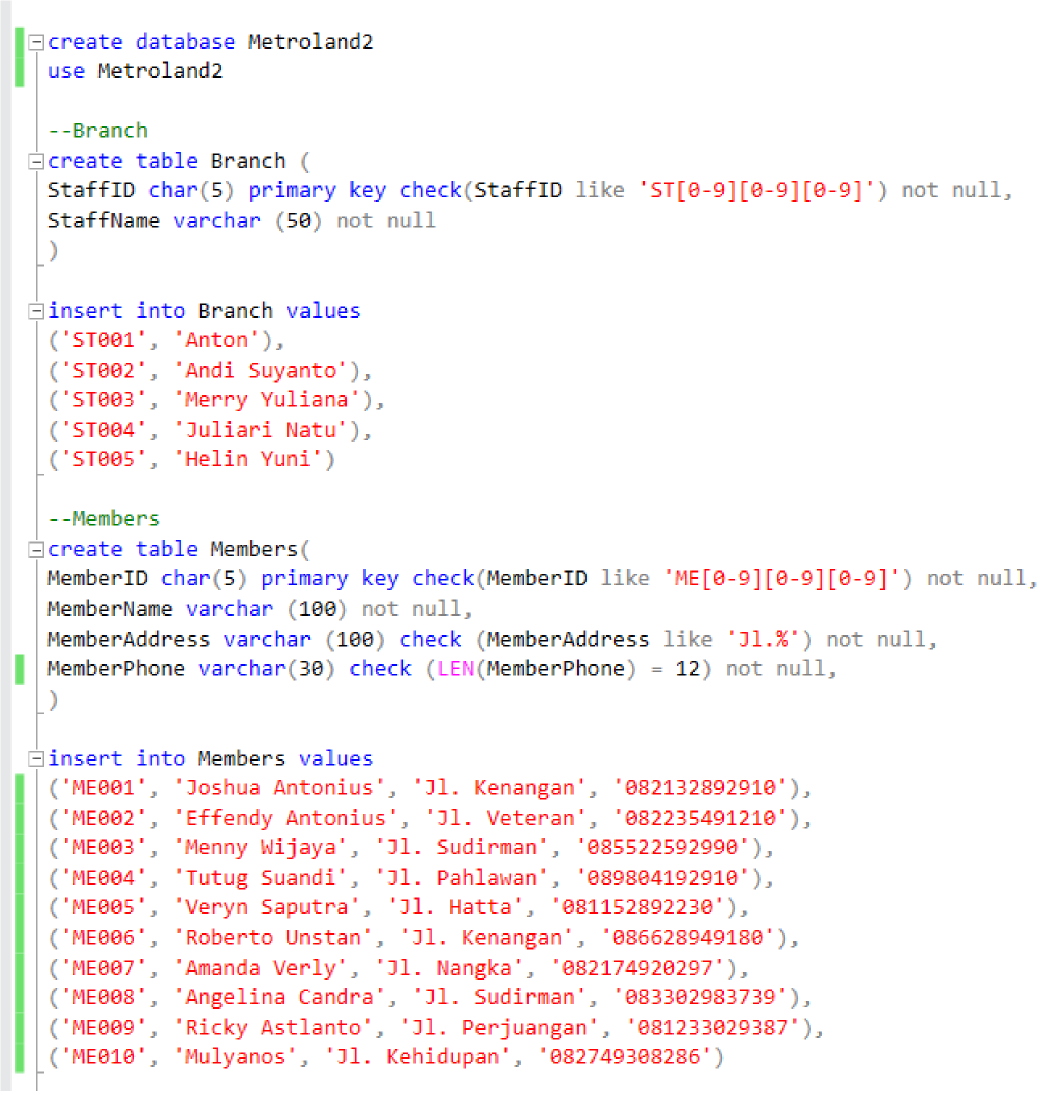
5. ERD



Toy table are the one with multiplicity because it connects with sales and purchases table

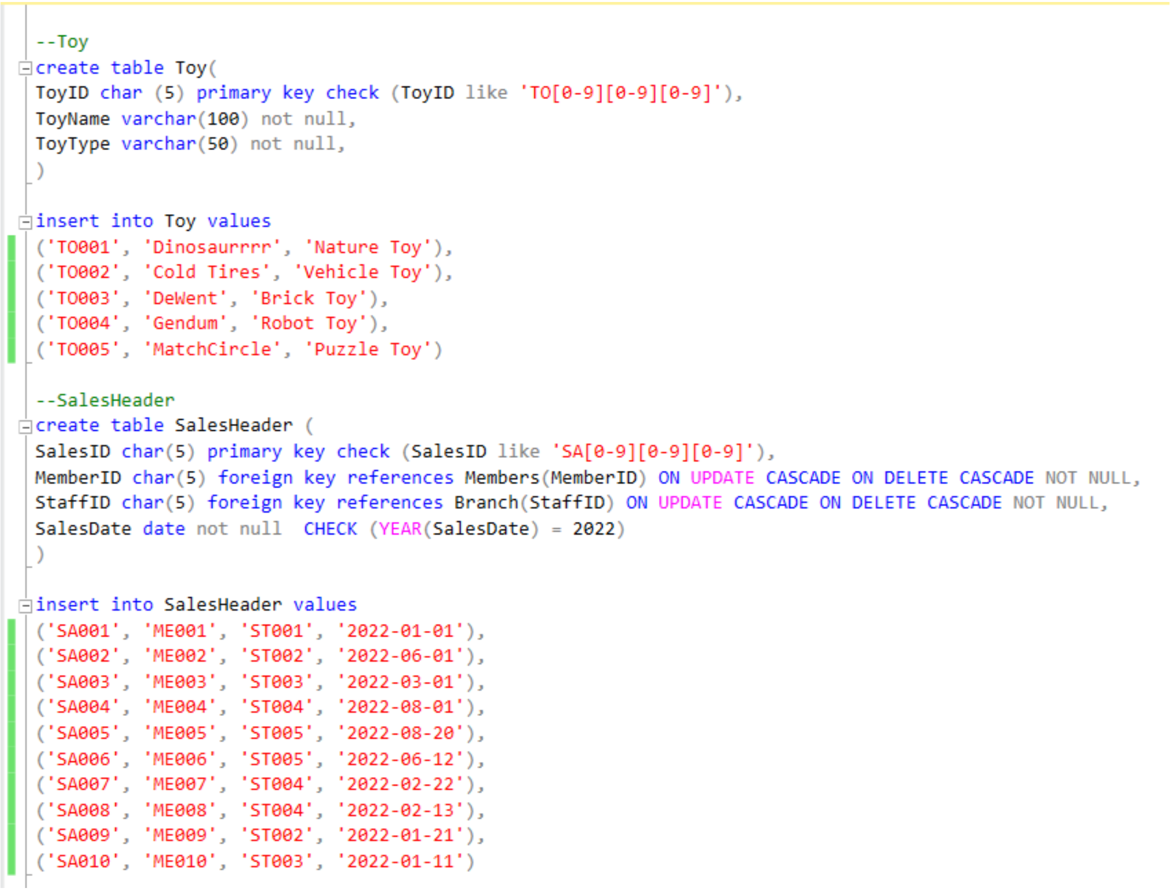
Before we continue to No. 6 – 8, let me give you the Create-Insert Query Screenshots

2502002785 – Jayson Cornelis



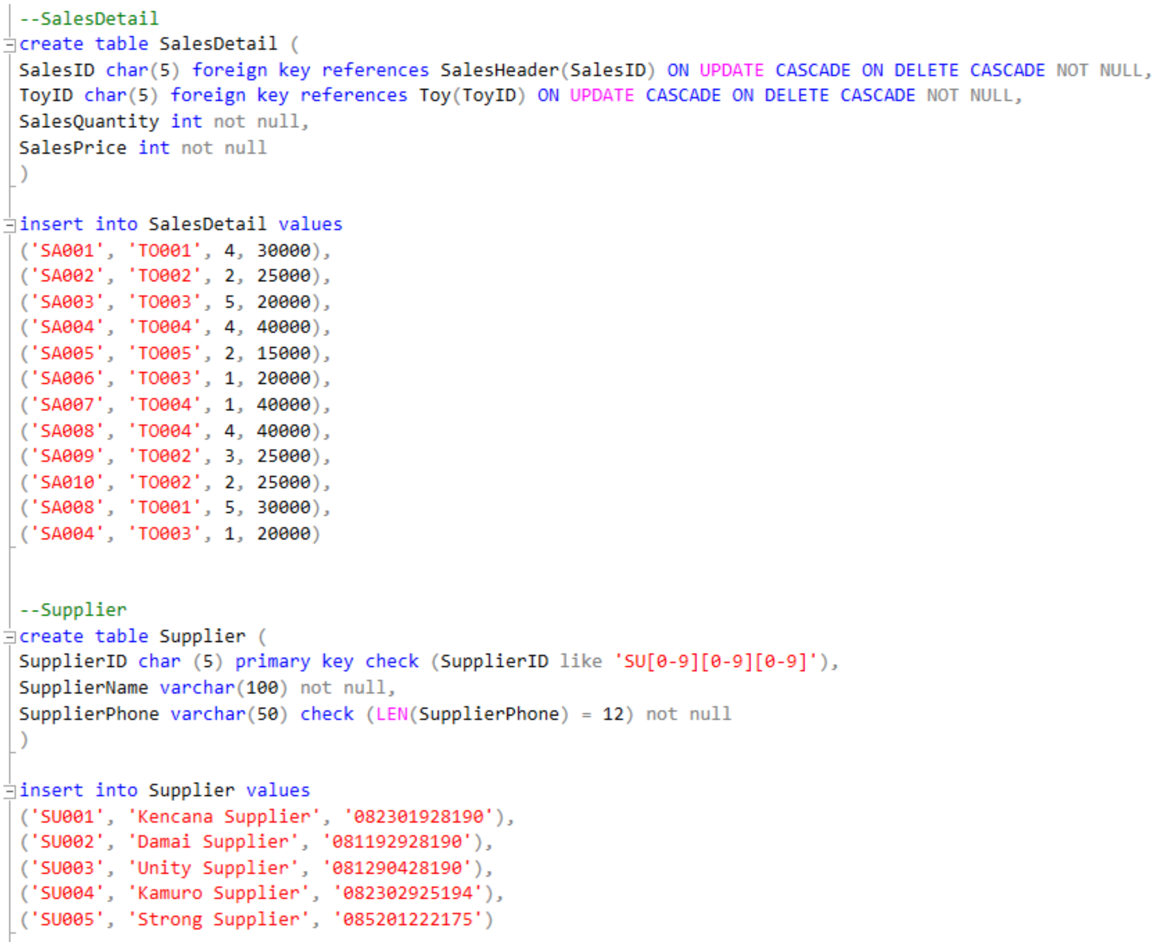
```sql
create database Metroland2
use Metroland2

--Branch
create table Branch (
StaffID char(5) primary key check(StaffID like 'ST[0-9][0-9][0-9]') not null,
StaffName varchar (50) not null
)

insert into Branch values
('ST001', 'Anton'),
('ST002', 'Andi Suyanto'),
('ST003', 'Merry Yuliana'),
('ST004', 'Juliari Natu'),
('ST005', 'Helin Yuni')

--Members
create table Members(
MemberID char(5) primary key check(MemberID like 'ME[0-9][0-9][0-9]') not null,
MemberName varchar (100) not null,
MemberAddress varchar (100) check (MemberAddress like 'Jl.%') not null,
MemberPhone varchar(30) check (LEN(MemberPhone) = 12) not null,
)

insert into Members values
('ME001', 'Joshua Antonius', 'Jl. Kenangan', '082132892910'),
('ME002', 'Effendy Antonius', 'Jl. Veteran', '082235491210'),
('ME003', 'Menny Wijaya', 'Jl. Sudirman', '085522592990'),
('ME004', 'Tutug Suandi', 'Jl. Pahlawan', '089804192910'),
('ME005', 'Veryn Saputra', 'Jl. Hatta', '081152892230'),
('ME006', 'Roberto Unstan', 'Jl. Kenangan', '086628949180'),
('ME007', 'Amanda Verly', 'Jl. Nangka', '082174920297'),
('ME008', 'Angelina Candra', 'Jl. Sudirman', '083302983739'),
('ME009', 'Ricky Astlanto', 'Jl. Perjuangan', '081233029387'),
('ME010', 'Mulyanos', 'Jl. Kehidupan', '082749308286')
```

2502002785 – Jayson Cornelis



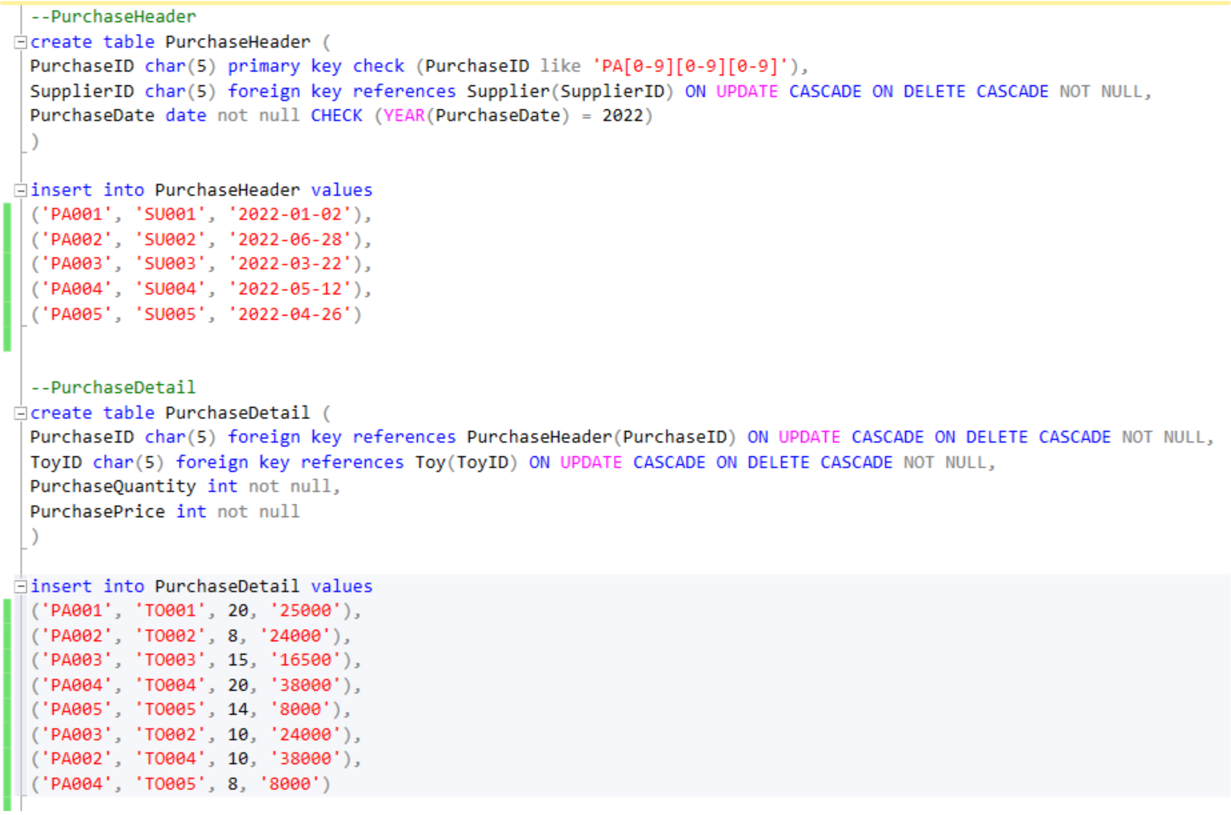
```
--Toy
create table Toy(
ToyID char (5) primary key check (ToyID like 'TO[0-9][0-9][0-9]'),
ToyName varchar(100) not null,
ToyType varchar(50) not null,
)

insert into Toy values
('TO001', 'Dinosaurrrr', 'Nature Toy'),
('TO002', 'Cold Tires', 'Vehicle Toy'),
('TO003', 'DeWent', 'Brick Toy'),
('TO004', 'Gendum', 'Robot Toy'),
('TO005', 'MatchCircle', 'Puzzle Toy')

--SalesHeader
create table SalesHeader (
SalesID char(5) primary key check (SalesID like 'SA[0-9][0-9][0-9]'),
MemberID char(5) foreign key references Members(MemberID) ON UPDATE CASCADE ON DELETE CASCADE NOT NULL,
StaffID char(5) foreign key references Branch(StaffID) ON UPDATE CASCADE ON DELETE CASCADE NOT NULL,
SalesDate date not null  CHECK (YEAR(SalesDate) = 2022)
)

insert into SalesHeader values
('SA001', 'ME001', 'ST001', '2022-01-01'),
('SA002', 'ME002', 'ST002', '2022-06-01'),
('SA003', 'ME003', 'ST003', '2022-03-01'),
('SA004', 'ME004', 'ST004', '2022-08-01'),
('SA005', 'ME005', 'ST005', '2022-08-20'),
('SA006', 'ME006', 'ST005', '2022-06-12'),
('SA007', 'ME007', 'ST004', '2022-02-22'),
('SA008', 'ME008', 'ST004', '2022-02-13'),
('SA009', 'ME009', 'ST002', '2022-01-21'),
('SA010', 'ME010', 'ST003', '2022-01-11')
```

2502002785 – Jayson Cornelis



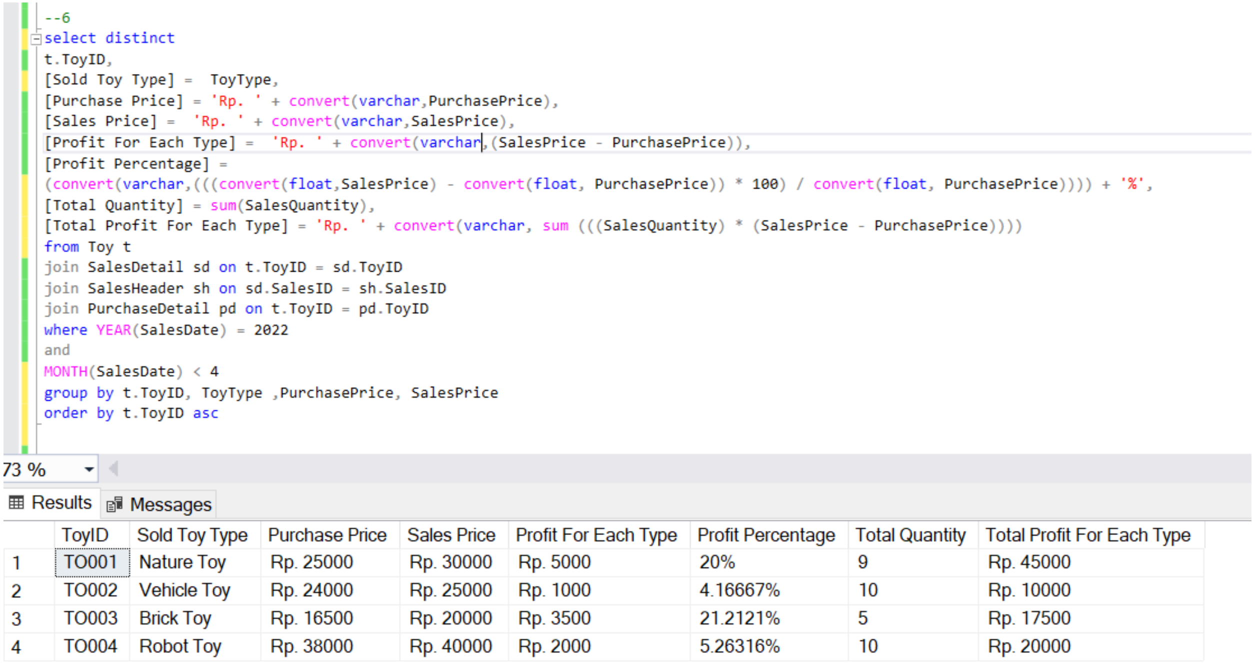
```
--SalesDetail
create table SalesDetail (
SalesID char(5) foreign key references SalesHeader(SalesID) ON UPDATE CASCADE ON DELETE CASCADE NOT NULL,
ToyID char(5) foreign key references Toy(ToyID) ON UPDATE CASCADE ON DELETE CASCADE NOT NULL,
SalesQuantity int not null,
SalesPrice int not null
)

insert into SalesDetail values
('SA001', 'TO001', 4, 30000),
('SA002', 'TO002', 2, 25000),
('SA003', 'TO003', 5, 20000),
('SA004', 'TO004', 4, 40000),
('SA005', 'TO005', 2, 15000),
('SA006', 'TO003', 1, 20000),
('SA007', 'TO004', 1, 40000),
('SA008', 'TO004', 4, 40000),
('SA009', 'TO002', 3, 25000),
('SA010', 'TO002', 2, 25000),
('SA008', 'TO001', 5, 30000),
('SA004', 'TO003', 1, 20000)


--Supplier
create table Supplier (
SupplierID char (5) primary key check (SupplierID like 'SU[0-9][0-9][0-9]'),
SupplierName varchar(100) not null,
SupplierPhone varchar(50) check (LEN(SupplierPhone) = 12) not null
)

insert into Supplier values
('SU001', 'Kencana Supplier', '082301928190'),
('SU002', 'Damai Supplier', '081192928190'),
('SU003', 'Unity Supplier', '081290428190'),
('SU004', 'Kamuro Supplier', '082302925194'),
('SU005', 'Strong Supplier', '085201222175')
```

2502002785 – Jayson Cornelis



```
--PurchaseHeader
create table PurchaseHeader (
PurchaseID char(5) primary key check (PurchaseID like 'PA[0-9][0-9][0-9]'),
SupplierID char(5) foreign key references Supplier(SupplierID) ON UPDATE CASCADE ON DELETE CASCADE NOT NULL,
PurchaseDate date not null CHECK (YEAR(PurchaseDate) = 2022)
)

insert into PurchaseHeader values
('PA001', 'SU001', '2022-01-02'),
('PA002', 'SU002', '2022-06-28'),
('PA003', 'SU003', '2022-03-22'),
('PA004', 'SU004', '2022-05-12'),
('PA005', 'SU005', '2022-04-26')


--PurchaseDetail
create table PurchaseDetail (
PurchaseID char(5) foreign key references PurchaseHeader(PurchaseID) ON UPDATE CASCADE ON DELETE CASCADE NOT NULL,
ToyID char(5) foreign key references Toy(ToyID) ON UPDATE CASCADE ON DELETE CASCADE NOT NULL,
PurchaseQuantity int not null,
PurchasePrice int not null
)

insert into PurchaseDetail values
('PA001', 'TO001', 20, '25000'),
('PA002', 'TO002', 8, '24000'),
('PA003', 'TO003', 15, '16500'),
('PA004', 'TO004', 20, '38000'),
('PA005', 'TO005', 14, '8000'),
('PA003', 'TO002', 10, '24000'),
('PA002', 'TO004', 10, '38000'),
('PA004', 'TO005', 8, '8000')
```

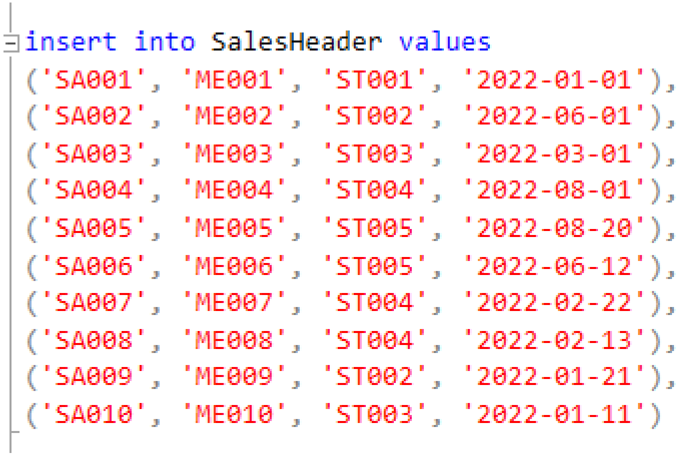
6. Please create a profitability report of sales transaction during the period Jan – Mar 2022. The report will show some require fields quantity of each sold toys type, purchase price, sales price, profit for each type, and the profitability percentage.



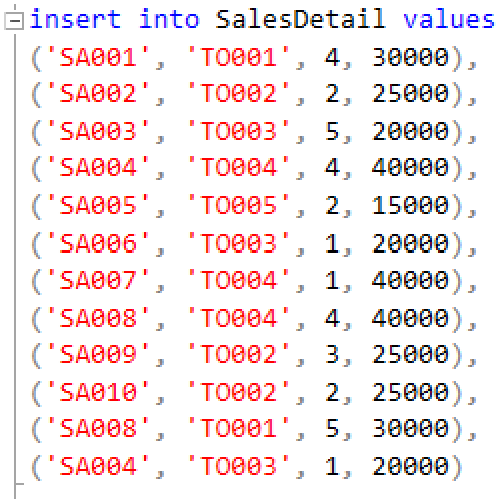
```
--6
select distinct
t.ToyID,
[Sold Toy Type] =  ToyType,
[Purchase Price] = 'Rp. ' + convert(varchar,PurchasePrice),
[Sales Price] =  'Rp. ' + convert(varchar,SalesPrice),
[Profit For Each Type] =  'Rp. ' + convert(varchar,(SalesPrice - PurchasePrice)),
[Profit Percentage] =
(convert(varchar,(((convert(float,SalesPrice) - convert(float, PurchasePrice)) * 100) / convert(float, PurchasePrice)))) + '%',
[Total Quantity] = sum(SalesQuantity),
[Total Profit For Each Type] = 'Rp. ' + convert(varchar, sum (((SalesQuantity) * (SalesPrice - PurchasePrice))))
from Toy t
join SalesDetail sd on t.ToyID = sd.ToyID
join SalesHeader sh on sd.SalesID = sh.SalesID
join PurchaseDetail pd on t.ToyID = pd.ToyID
where YEAR(SalesDate) = 2022
and
MONTH(SalesDate) < 4
group by t.ToyID, ToyType ,PurchasePrice, SalesPrice
order by t.ToyID asc
```

	ToyID	Sold Toy Type	Purchase Price	Sales Price	Profit For Each Type	Profit Percentage	Total Quantity	Total Profit For Each Type
1	TO001	Nature Toy	Rp. 25000	Rp. 30000	Rp. 5000	20%	9	Rp. 45000
2	TO002	Vehicle Toy	Rp. 24000	Rp. 25000	Rp. 1000	4.16667%	10	Rp. 10000
3	TO003	Brick Toy	Rp. 16500	Rp. 20000	Rp. 3500	21.2121%	5	Rp. 17500
4	TO004	Robot Toy	Rp. 38000	Rp. 40000	Rp. 2000	5.26316%	10	Rp. 20000

All of the 4 types of toys are in the sales process on January until March. Use distinct to eliminate the multiple data.



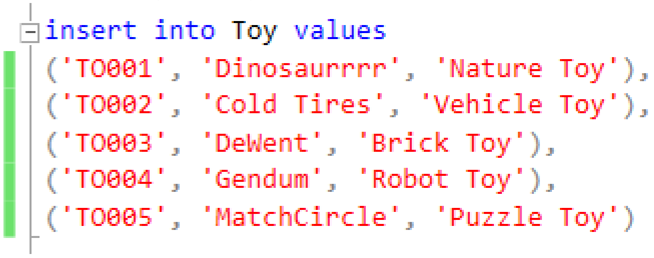
```
insert into SalesHeader values
('SA001', 'ME001', 'ST001', '2022-01-01'),
('SA002', 'ME002', 'ST002', '2022-06-01'),
('SA003', 'ME003', 'ST003', '2022-03-01'),
('SA004', 'ME004', 'ST004', '2022-08-01'),
('SA005', 'ME005', 'ST005', '2022-08-20'),
('SA006', 'ME006', 'ST005', '2022-06-12'),
('SA007', 'ME007', 'ST004', '2022-02-22'),
('SA008', 'ME008', 'ST004', '2022-02-13'),
('SA009', 'ME009', 'ST002', '2022-01-21'),
('SA010', 'ME010', 'ST003', '2022-01-11')
```



```
insert into SalesDetail values
('SA001', 'TO001', 4, 30000),
('SA002', 'TO002', 2, 25000),
('SA003', 'TO003', 5, 20000),
('SA004', 'TO004', 4, 40000),
('SA005', 'TO005', 2, 15000),
('SA006', 'TO003', 1, 20000),
('SA007', 'TO004', 1, 40000),
('SA008', 'TO004', 4, 40000),
('SA009', 'TO002', 3, 25000),
('SA010', 'TO002', 2, 25000),
('SA008', 'TO001', 5, 30000),
('SA004', 'TO003', 1, 20000)
```

SalesID that did transaction in January – March 2022 :

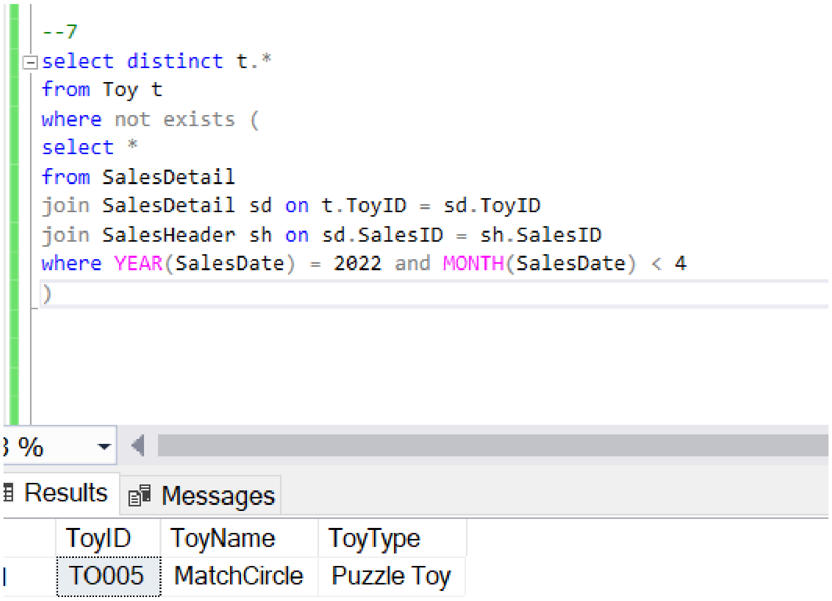
1. SA001 (TO001)
2. SA003 (TO003)
3. SA007 (TO004)
4. SA008 (TO004 & TO001)
5. SA009 (TO002)
6. SA010 (TO002)



```
insert into Toy values
('TO001', 'Dinosaurrrr', 'Nature Toy'),
('TO002', 'Cold Tires', 'Vehicle Toy'),
('TO003', 'DeWent', 'Brick Toy'),
('TO004', 'Gendum', 'Robot Toy'),
('TO005', 'MatchCircle', 'Puzzle Toy')
```

Conclusion, toy types that included here are : TO001, TO002. TO003, TO004

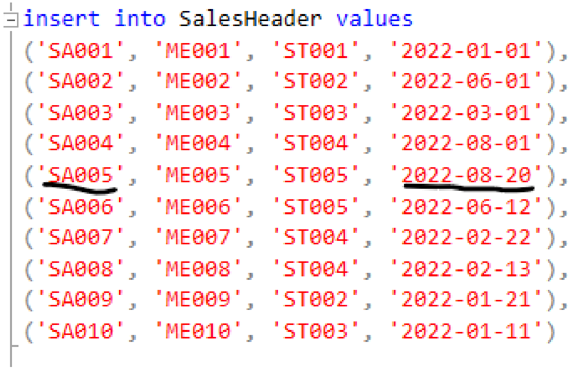
7. Please create a report using subquery to show items that have not sold during Jan – Mar 2022



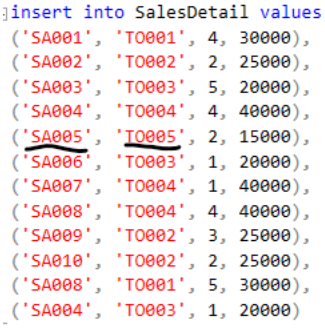
```
--7
select distinct t.*
from Toy t
where not exists (
select *
from SalesDetail
join SalesDetail sd on t.ToyID = sd.ToyID
join SalesHeader sh on sd.SalesID = sh.SalesID
where YEAR(SalesDate) = 2022 and MONTH(SalesDate) < 4
)
```

ToyID	ToyName	ToyType
TO005	MatchCircle	Puzzle Toy

Here’s why TO005 showed up :



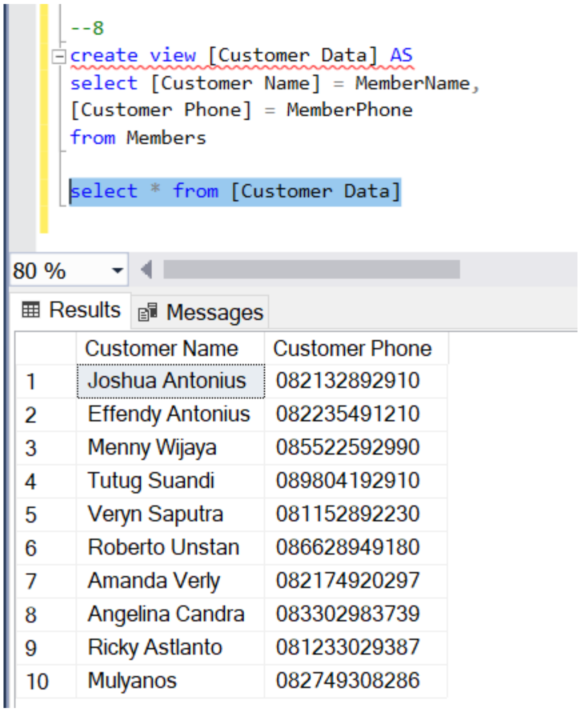
```
insert into SalesHeader values
('SA001', 'ME001', 'ST001', '2022-01-01'),
('SA002', 'ME002', 'ST002', '2022-06-01'),
('SA003', 'ME003', 'ST003', '2022-03-01'),
('SA004', 'ME004', 'ST004', '2022-08-01'),
('SA005', 'ME005', 'ST005', '2022-08-20'),
('SA006', 'ME006', 'ST005', '2022-06-12'),
('SA007', 'ME007', 'ST004', '2022-02-22'),
('SA008', 'ME008', 'ST004', '2022-02-13'),
('SA009', 'ME009', 'ST002', '2022-01-21'),
('SA010', 'ME010', 'ST003', '2022-01-11')
```



```
insert into SalesDetail values
('SA001', 'TO001', 4, 30000),
('SA002', 'TO002', 2, 25000),
('SA003', 'TO003', 5, 20000),
('SA004', 'TO004', 4, 40000),
('SA005', 'TO005', 2, 15000),
('SA006', 'TO003', 1, 20000),
('SA007', 'TO004', 1, 40000),
('SA008', 'TO004', 4, 40000),
('SA009', 'TO002', 3, 25000),
('SA010', 'TO002', 2, 25000),
('SA008', 'TO001', 5, 30000),
('SA004', 'TO003', 1, 20000)
```

As I underline it, TO005 sales happened on August only and no sales on January until March.

8. Display Customer Data



```
--8
create view [Customer Data] AS
select [Customer Name] = MemberName,
[Customer Phone] = MemberPhone
from Members

select * from [Customer Data]
```

	Customer Name	Customer Phone
1	Joshua Antonius	082132892910
2	Effendy Antonius	082235491210
3	Menny Wijaya	085522592990
4	Tutug Suandi	089804192910
5	Veryn Saputra	081152892230
6	Roberto Unstan	086628949180
7	Amanda Verly	082174920297
8	Angelina Candra	083302983739
9	Ricky Astlanto	081233029387
10	Mulyanos	082749308286

Video link : https://youtu.be/JTKYDZW3LDw