CASE STUDY

softwAEr House

softwAEr House is a famous software store that belongs to Aedwin and its located in Jakarta. **softwAEr House sells many kinds of software to costumers and got its product from the software distributor around the world**.

Every staff who is hired by **softwAEr House** have a task to **serve a customer who wants to buy softwares in a sales transaction** and **serve a distributor when restocking in a purchase transaction**. Every staff must follow several procedures to become a staff and do their job desks, which are:

- Every staff hired must have a personal information such as name, gender, phone number, date of birth, email and address. Every staff has an identification number with the followingformat:

  “SFXXX”
  X => number between 0 – 9

- Every software sold by **softwAEr House** has several types with different names. Every software type has an identification number with the following format:

  “TPXXX”
  X => number between 0 – 9

- As for the software itself has information such as name, version, release date, price, stock. Every software has an identification number with the following format:

  “SWXXX”
  X => number between 0 – 9

- Every customer who is a member must have information such as name, gender, address, and phone. Every customer has an identification number with the following format:

  “CSXXX”
  X => number between 0 – 9

- Every distributor data must have information such as name and company. Every distributor has an identification number with the following format:

  “DTXXX”
  X => number between 0 – 9

- Staff can serve customers who wants to buy products.
- Every **sales transaction** made must have information about the staff, customer, transaction date, software and quantity. Every **sales transaction** has an identification number with the following format:

“SLXXX”
X => number between 0 – 9

- A customer could buy **more than one software** in every transaction.
- Staff can also serve distributors who sells the products
- Every **purchase transaction** made must have information about the staff, distributor, transaction date, software bought and quantity. Every **purchase transaction** has an identification number with the following format:

“PRXXX”
X => number between 0 – 9

- A distributor could sell **more than one software** in every transaction.

Notes:

- Customer’s name must be longer than 5 characters.
- Staff’s phone number length must be equal to 13.
- Staff’s gender must be either “Male” or “Female” (without quote).
- Staff’s address must start with 3 digits number with space at the end of the number and followed by a word (ex: 157 street).
- Staff must be older than 17 years old.
- Software price must be between 20000 and 3000000.
- Software version data type must be varchar and the data are in x.x format (x is a digit,ex:1.1, 1.0, etc)
- Distributor’s name must consist of at least 2 words.

Because **softwAEr House** was established recently, Aedwin asks you to help him design and **create a database system** that can store data and maintain **sales & purchase transactions** and **assets recordings according to requirements** he has requested of you. The tasks that you must do are:

a. Create Entity Relationship Diagram to maintain **sales & purchase transactions** and **assets recordings**.

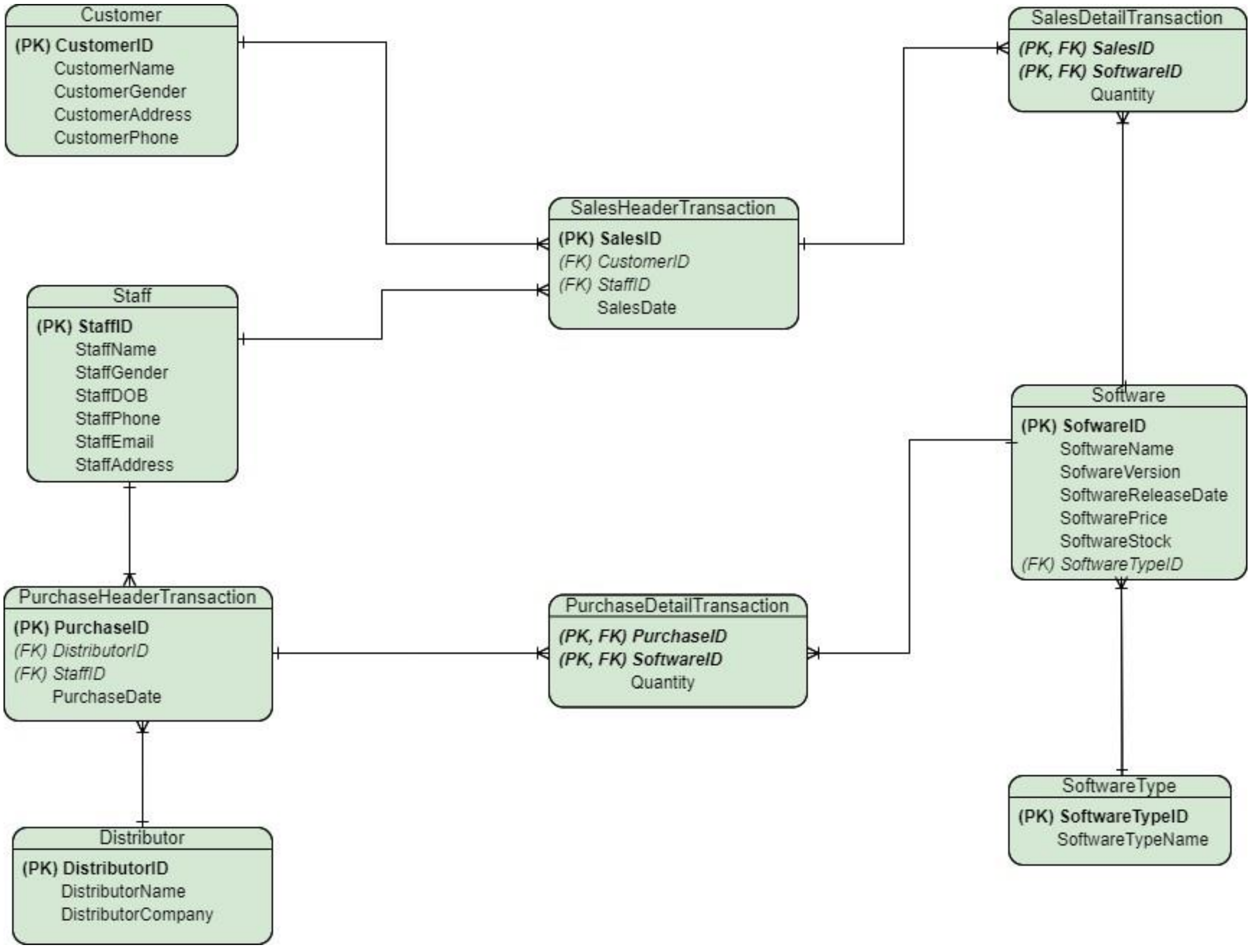
b. Create a database system using DDL syntax that relevant with **sales & purchase transactions** and **assets recordings**. The database system must include **database** and **tables** with the required procedures.

c. Create query using DML syntax to fill the tables in database systems with data based on the specified conditions.

d. Create query using DML syntax to simulate the transactions process for **sales and purchase transactions**.

e. To support database management process in **softwAEr House**, Aedwin asks you to provide some queries which give him important data he needs.

A. *Create Entity Relationship Diagram to maintain **sales & purchase transactions** and **assets recordings**.*



B. *Create a database system using DDL syntax that relevant with **sales & purchase transactions** and **assets recordings**. The database system must include **database** and **tables** with the required procedures.*

```
CREATE DATABASE softwAErHouse

--Staff
CREATE TABLE Staff(
    StaffID CHAR(5) PRIMARY KEY CHECK (StaffID LIKE 'SF[0-9][0-9][0-9]'),
    StaffName VARCHAR(25),
    StaffGender VARCHAR(8) CHECK (StaffGender LIKE 'Male' OR StaffGender LIKE
'Female'),
    StaffPhone VARCHAR(13) CHECK (LEN(StaffPhone)=13),
    StaffDOB DATE CHECK (DATEDIFF(YEAR, StaffDOB, CONVERT(DATE,
GETDATE())) > 17),
    StaffEmail VARCHAR(25),
    StaffAddress VARCHAR(25) CHECK (StaffAddress LIKE '[0-9][0-9][0-9] %')
)
```

--Software Type

```
CREATE TABLE SoftwareType(
    SoftwareTypeId CHAR(5) PRIMARY KEY CHECK(SoftwareTypeId LIKE 'TP[0-9][0-9][0-
9]'),
    SoftwareTypeName VARCHAR (40) CHECK
    (SoftwareTypeName LIKE 'Multimedia Design' OR SoftwareTypeName LIKE 'Mobile
Application' OR SoftwareTypeName LIKE 'Database Management'
     OR SoftwareTypeName LIKE 'Game Development' OR SoftwareTypeName LIKE 'Browser'
OR SoftwareTypeName LIKE 'Text Editor' OR SoftwareTypeName LIKE 'Web Development'
     OR SoftwareTypeName LIKE 'Business Analytics' OR SoftwareTypeName LIKE 'Integrated
Development Environment' OR SoftwareTypeName LIKE 'Others')
)
```

--Software

```
CREATE TABLE Software(
    SoftwareId CHAR(5) PRIMARY KEY CHECK(SoftwareId LIKE 'SW[0-9][0-9][0-9]'),
    SoftwareName VARCHAR(15),
    SoftwareVersion VARCHAR(5) CHECK(SoftwareVersion LIKE '[0-9].[0-9]'),
    SoftwareReleaseDate DATE,
    SoftwarePrice INT CHECK(SoftwarePrice BETWEEN 20000 AND 3000000),
    SoftwareStock INT
)
```

--Customer

```
CREATE TABLE Customer(
    CustomerID CHAR(5) PRIMARY KEY CHECK(CustomerID LIKE 'CS[0-9][0-9][0-9]'),
    CustomerName VARCHAR(25) CHECK (LEN(CustomerName) > 5),
    CustomerGender VARCHAR(8),
    CustomerAddress VARCHAR(25),
    CustomerPhone VARCHAR(15),
)
```

--Distributor

```
CREATE TABLE Distributor(
    DistributorID CHAR(5) PRIMARY KEY CHECK(DistributorID LIKE 'DT[0-9][0-9][0-9]'),
    DistributorName VARCHAR(25) CHECK (DistributorName LIKE '% %'),
    DistributorCompany VARCHAR(25)
)
```

```
--SalesHeaderTransaction
CREATE TABLE SalesHeaderTransaction(
    SalesID CHAR(5) PRIMARY KEY CHECK(SalesID LIKE 'SL[0-9][0-9][0-9]'),
    CustomerID CHAR(5) REFERENCES Customer(CustomerID),
    StaffID CHAR(5) REFERENCES Staff(StaffID),
    SalesDate DATE
)

--SalesDetailTransaction
CREATE TABLE SalesDetailTransaction(
    SalesID CHAR(5) REFERENCES SalesHeaderTransaction(SalesID),
    SoftwareId CHAR(5) REFERENCES Software(SoftwareID),
    SoftwareTypeId CHAR(5) REFERENCES SoftwareType(SoftwareTypeId),
    Quantity INT,
    PRIMARY KEY (SalesId, SoftwareId)
)

--PurchaseHeaderTransaction
CREATE TABLE PurchaseHeaderTransaction (
    PurchaseID CHAR(5) PRIMARY KEY CHECK (PurchaseID LIKE 'PR[0-9][0-9][0-9]'),
    DistributorID CHAR(5) REFERENCES Distributor(DistributorID),
    StaffID CHAR(5) REFERENCES Staff(StaffID),
    PurchaseDate DATE
)

--PurchaseDetailTransaction
CREATE TABLE PurchaseDetailTransaction (
   PurchaseID CHAR(5) REFERENCES PurchaseHeaderTransaction(PurchaseID),
   SoftwareID CHAR(5) REFERENCES Software(SoftwareID),
   Quantity INT,
   PRIMARY KEY(PurchaseID, SoftwareID)
)
```

C. Create query using DML syntax to fill the tables in database systems with data based on the following conditions:

   a. **Master** table must be filled with more than or equals 15 data.
   b. **Transaction** table must be filled with more than or equals 15 data.
   c. **Transaction detail** table must be filled with more than or equals 25 data.
   d. For the **Software Type** table, the table must be filled with the following data:

Software Type Names	
Multimedia Design	Mobile Application

Database Management	Game Development
Browser	Text Editor
Web Development	Business Analytics
Integrated Development Environment	Others

--Staff

```
INSERT INTO Staff VALUES('SF001', 'Lydia Fortuna', 'Female', '0812648979832', '1999-02-22', 'lydia_fortuna@gmail.com', '233 Street')
INSERT INTO Staff VALUES('SF002', 'Yohanes Purnomo Sanjaya', 'Male', '0867156094451', '2002-08-15', 'yohanes_w_s@gmail.com', '786 Street')
INSERT INTO Staff VALUES('SF003', 'Fajar Saputra', 'Male', '0899974335984', '1997-07-31', 'Fajar_Saputra_@yahoo.com', '666 Street')
INSERT INTO Staff VALUES('SF004', 'Yuliana Gunawan Lestari', 'Female', '0862230007665', '1991-11-09', 'yuliana09gn@gmail.com', '559 Street')
INSERT INTO Staff VALUES('SF005', 'Anastasya Lestari', 'Female', '0811150064275', '1994-07-15', 'A_Lestari94@yahoo.com', '122 Street')
INSERT INTO Staff VALUES('SF006', 'Hailey Frolensa', 'Female', '0890902787779', '2000-12-22', 'hailey_f@yahoo.com', '734 Street')
INSERT INTO Staff VALUES('SF007', 'Tirta Dharma Handoko', 'Male', '0844463127759', '1989-01-01', 'TirtaDHandoko19@gmail.com', '100 Street')
INSERT INTO Staff VALUES('SF008', 'Dendrick Jaya', 'Male', '0888109975226', '1985-03-23', 'jaya.derick@gmail.com', '455 Street')
INSERT INTO Staff VALUES('SF009', 'Moehammad Fauzi Hidayat', 'Male', '0837776123008', '1993-10-14', 'fauziHidayat@yahoo.com', '156 Street')
INSERT INTO Staff VALUES('SF010', 'David Nugrogo', 'Male', '0866347890455', '1991-05-31', 'David_N@gmail.com', '654 Street')
INSERT INTO Staff VALUES('SF011', 'Reyna Cahya Januar ', 'Female', '0811199956656', '1999-02-28', 'reyna_CJ@gmail.com', '679 Street')
INSERT INTO Staff VALUES('SF012', 'Azura', 'Male', '0856343339900', '1884-06-27', 'azura27_84@gmail.com', '900 Street')
INSERT INTO Staff VALUES('SF013', 'Benedick Iskandar Samudra', 'Male', '0888812307661', '1997-04-06', 'IskandarSamudra@gmail.com', '178 Street')
INSERT INTO Staff VALUES('SF014', 'Siti Bulan Kemala Sari', 'Female', '0822665430813', '2000-08-30', 'siti2000@gmail.com', '144 Street')
INSERT INTO Staff VALUES('SF015', 'Anissa Setianingsi', 'Female', '0872437200751', '1994-03-17', 'setianingsi@yahoo.com', '543 Street')
INSERT INTO Staff VALUES('SF016', 'Angkasa Putra', 'Male', '0871634449826', '1999-08-23', 'PutrAngka@gmail.com', '765 Street')
INSERT INTO Staff VALUES('SF017', 'Bulan Lestari', 'Female', '0811187544546', '1996-11-13', 'bulan_L@gmail.com', '489 Street')
INSERT INTO Staff VALUES('SF018', 'Rahmat Sentosa', 'Male', '0876252528553', '1995-08-30', 'rahmat2000@yahoo.com', '378 Street')
INSERT INTO Staff VALUES('SF019', 'Dimas Haraja', 'Male', '0856667882994', '1998-06-14',
'Dimas98@gmail.com', '188 Street')
INSERT INTO Staff VALUES('SF020', 'Raja Dharma Jaya', 'Male', '0867344482137', '2001-12-
15', 'dharmaJaya@yahoo.com', '600 Street')

--SoftwareType
INSERT INTO SoftwareType VALUES('TP001', 'Multimedia Design')
INSERT INTO SoftwareType VALUES('TP002', 'Mobile Application')
INSERT INTO SoftwareType VALUES('TP003', 'Database Management')
INSERT INTO SoftwareType VALUES('TP004', 'Game Development')
INSERT INTO SoftwareType VALUES('TP005', 'Browser')
INSERT INTO SoftwareType VALUES('TP006', 'Text Editor')
INSERT INTO SoftwareType VALUES('TP007', 'Web Development')
INSERT INTO SoftwareType VALUES('TP008', 'Business Analytics')
INSERT INTO SoftwareType VALUES('TP009', 'Integrated Development Environment')
INSERT INTO SoftwareType VALUES('TP010', 'Others')

--Software
INSERT INTO Software VALUES('SW001', 'Photoshop CS5', '5.1', '2002-05-17', '300000', 20)
INSERT INTO Software VALUES('SW002', 'SQL Server 2019', '2.3', '2019-11-04', '900000', 30)
INSERT INTO Software VALUES('SW003', 'Oracle RDBMS', '1.9', '2019-02-13', '250000', 25)
INSERT INTO Software VALUES('SW004', 'Brave', '8.9', '2020-05-12', '150000', 35)
INSERT INTO Software VALUES('SW005', 'Bootstrap', '4.5', '2020-05-12', '700000', 30)
INSERT INTO Software VALUES('SW006', 'Axure RP Pro', '9.0', '2019-04-15', '800000', 25)
INSERT INTO Software VALUES('SW007', 'IntelliJ IDEA', '2.0', '2020-04-09', '1300000', 20)
INSERT INTO Software VALUES('SW008', 'Eclipse', '4.1', '2018-09-16', '100000', 35)
INSERT INTO Software VALUES('SW009', 'Felgo', '3.5', '2002-04-22', '165000', 24)
INSERT INTO Software VALUES('SW010', 'Unity', '3.1', '2020-05-06', '140000', 30)
INSERT INTO Software VALUES('SW011', 'Sublime Text', '2.2', '2019-10-01', '245000', 22)
INSERT INTO Software VALUES('SW012', 'Qlik Sense', '3.7', '2020-04-14', '270000', 18)
INSERT INTO Software VALUES('SW013', 'CorelDraw 2019', '2.5', '2020-03-12', '255000', 30)
INSERT INTO Software VALUES('SW014', 'Office 2019', '5.4', '2018-09-24', '130000', 21)
INSERT INTO Software VALUES('SW015', 'VLC Player', '3.0', '2017-11-30', '50000', 20)
INSERT INTO Software VALUES('SW016', 'PostgreSQL', '9.2', '2012-05-14', '165000', 37)
INSERT INTO Software VALUES('SW017', 'SAP ERP', '6.0', '2006-12-21', '300000', 45)
INSERT INTO Software VALUES('SW018', 'Komodo Edit', '1.1', '2018-06-21', '230000', 25)
INSERT INTO Software VALUES('SW019', 'NetBeans IDE', '7.5', '2019-07-22', '170000', 55)
INSERT INTO Software VALUES('SW020', 'Django', '8.6', '2020-05-04', '235000', 15)

--Customer
INSERT INTO Customer VALUES('CS001', 'Bryan Elpizochari', 'Male', 'Jl.Kebon Jeruk No.12',
'081254398212')
INSERT INTO Customer VALUES('CS002', 'Josh Ezekeil', 'Male', 'Jl.boulevard Kelapa' ,
'081564891235')
```

```
INSERT INTO Customer VALUES('CS003', 'Annabella Nathalia', 'Female', 'Jl.Sunter Icon No.30',
'081249761349')
INSERT INTO Customer VALUES('CS004', 'Andy Leonard', 'Male', 'Jl.Pegangsaan No.1',
'081589456123')
INSERT INTO Customer VALUES('CS005', 'Marc Setiawan', 'Male', 'Jl.Bukit Indah No.15',
'081282461593')
INSERT INTO Customer VALUES('CS006', 'Natalie Norman', 'Female', 'Jl.Keraton No. 21',
'08137914582')
INSERT INTO Customer VALUES('CS007', 'Abigail Clarissa', 'Female', 'Jl.Palangkaraya No.30 ',
'081212457896')
INSERT INTO Customer VALUES('CS008', 'Alexa Bonner', 'Female', 'Jl.Dr Sam Ratulangi 9',
'02123550208')
INSERT INTO Customer VALUES('CS009', 'Saarah Goodman', 'Female', 'Jl.MH Thamrin Kav 28',
'02131441712')
INSERT INTO Customer VALUES('CS010', 'Alan Bradford', 'Male', 'Jl.HR Rasuna Said Kav 2', '
02152742682')
INSERT INTO Customer VALUES('CS011', 'Jamie Connor', 'Male', 'Jl.Pegangsaan Tmr 1-A', '
02123032288')
INSERT INTO Customer VALUES('CS012', 'Orson Ramsey', 'Male', 'Jl.Kompl Mangga Dua',
'082162870332')
INSERT INTO Customer VALUES('CS013', 'Taren Brandt', 'Male', 'Jl.Madrasah Tanah Koja',
'081584267264')
INSERT INTO Customer VALUES('CS014', 'Jane Barber', 'Female', 'JL.Kasepuhan No.7',
'081535503311')
INSERT INTO Customer VALUES('CS015', 'Callie Woodley', 'Female', 'Jl.Kom L Yos Sudarso',
'0216511715')
INSERT INTO Customer VALUES('CS016', 'Sila Li', 'Female', 'JL Raden Saleh No.20',
'081564897845')
INSERT INTO Customer VALUES('CS017', 'Flynn Shea', 'Female', 'Jl.Cikini IV 12',
'0211649502')
INSERT INTO Customer VALUES('CS018', 'Alexandre East', 'Male', 'Jl.Teuku Umar 30 B',
'0211643598')
INSERT INTO Customer VALUES('CS019', 'Rodney Hatfield', 'Male', 'Jl.Kebon Sirih No. 48 ',
'0211321715')
INSERT INTO Customer VALUES('CS020', 'Kanaya Wong', 'Female', 'Jl.Radio Raya No. 1',
'0218845223')
```

--Distributor

```
INSERT INTO Distributor VALUES('DT001', 'Sami Frost', 'Caesars Entertainment')
INSERT INTO Distributor VALUES('DT002', 'Ritchie Baird', 'Advanced Micro Devices')
INSERT INTO Distributor VALUES('DT003', 'Mary Wang', 'Laboratory of America')
INSERT INTO Distributor VALUES('DT004', 'Kelsi Stout', 'Graphic Packaging Holding')
INSERT INTO Distributor VALUES('DT005', 'Heather Brooks', 'L3 Technologies')
INSERT INTO Distributor VALUES('DT006', 'Tori Mclean', 'Charter Communication')
```

```sql
INSERT INTO Distributor VALUES('DT007', 'Nuha Atherton', 'Blackstone Group')
INSERT INTO Distributor VALUES('DT008', 'Maximus Lancaster', 'Anixter International')
INSERT INTO Distributor VALUES('DT009', 'Kim Edwards', 'Polaris Industries')
INSERT INTO Distributor VALUES('DT010', 'Carolina Mcdonald', 'General Dynamics')
INSERT INTO Distributor VALUES('DT011', 'Maya Owen', 'United Technologies')
INSERT INTO Distributor VALUES('DT012', 'Mckenzie Soto', 'Boston Scientific')
INSERT INTO Distributor VALUES('DT013', 'Kush Perry', 'NextEra Energy')
INSERT INTO Distributor VALUES('DT014', 'Amirah Blair', 'CenterPoint Energy')
INSERT INTO Distributor VALUES('DT015', 'Yousef Mays', 'Blackstone Group')
INSERT INTO Distributor VALUES('DT016', 'Niam Clifford', 'Burlington Stores')
INSERT INTO Distributor VALUES('DT017', 'Callen Robins', 'R.R. Donnelley & Sons')
INSERT INTO Distributor VALUES('DT018', 'Rylee Mackay', 'Adobe')
INSERT INTO Distributor VALUES('DT019', 'Kirsty Mcdonnell', 'Goldman Sachs Group')
INSERT INTO Distributor VALUES('DT020', 'Amos Cantrell', 'Duke Energy')

--SalesHeaderTransaction
INSERT INTO SalesHeaderTransaction VALUES('SL001', 'CS001', 'SF002', '2018-08-11')
INSERT INTO SalesHeaderTransaction VALUES('SL002', 'CS014', 'SF003', '2018-08-20')
INSERT INTO SalesHeaderTransaction VALUES('SL003', 'CS012', 'SF003', '2018-09-25')
INSERT INTO SalesHeaderTransaction VALUES('SL004', 'CS007', 'SF007', '2018-10-16')
INSERT INTO SalesHeaderTransaction VALUES('SL005', 'CS004', 'SF009', '2018-10-30')
INSERT INTO SalesHeaderTransaction VALUES('SL006', 'CS010', 'SF005', '2018-11-02')
INSERT INTO SalesHeaderTransaction VALUES('SL007', 'CS005', 'SF001', '2018-11-17')
INSERT INTO SalesHeaderTransaction VALUES('SL008', 'CS002', 'SF006', '2018-11-20')
INSERT INTO SalesHeaderTransaction VALUES('SL009', 'CS013', 'SF004', '2018-12-05')
INSERT INTO SalesHeaderTransaction VALUES('SL010', 'CS007', 'SF003', '2019-01-03')
INSERT INTO SalesHeaderTransaction VALUES('SL011', 'CS011', 'SF010', '2019-01-20')
INSERT INTO SalesHeaderTransaction VALUES('SL012', 'CS008', 'SF004', '2019-02-16')
INSERT INTO SalesHeaderTransaction VALUES('SL013', 'CS004', 'SF009', '2019-02-20')
INSERT INTO SalesHeaderTransaction VALUES('SL014', 'CS006', 'SF012', '2019-02-26')
INSERT INTO SalesHeaderTransaction VALUES('SL015', 'CS003', 'SF014', '2019-03-30')
INSERT INTO SalesHeaderTransaction VALUES('SL016', 'CS009', 'SF019', '2019-01-20')
INSERT INTO SalesHeaderTransaction VALUES('SL017', 'CS017', 'SF020', '2019-02-16')
INSERT INTO SalesHeaderTransaction VALUES('SL018', 'CS002', 'SF018', '2019-02-20')
INSERT INTO SalesHeaderTransaction VALUES('SL019', 'CS020', 'SF016', '2019-02-26')
INSERT INTO SalesHeaderTransaction VALUES('SL020', 'CS016', 'SF017', '2019-03-30')

--SalesDetailTransaction
INSERT INTO SalesDetailTransaction VALUES('SL001', 'SW002', 'TP001', 5)
INSERT INTO SalesDetailTransaction VALUES('SL002', 'SW005', 'TP002', 3)
INSERT INTO SalesDetailTransaction VALUES('SL003', 'SW006', 'TP010', 9)
INSERT INTO SalesDetailTransaction VALUES('SL004', 'SW008', 'TP009', 7)
INSERT INTO SalesDetailTransaction VALUES('SL005', 'SW001', 'TP003', 6)
INSERT INTO SalesDetailTransaction VALUES('SL006', 'SW004', 'TP005', 2)
INSERT INTO SalesDetailTransaction VALUES('SL007', 'SW010', 'TP007', 12)
INSERT INTO SalesDetailTransaction VALUES('SL008', 'SW009', 'TP003', 1)
INSERT INTO SalesDetailTransaction VALUES('SL009', 'SW003', 'TP008', 10)
INSERT INTO SalesDetailTransaction VALUES('SL010', 'SW007', 'TP001', 5)
INSERT INTO SalesDetailTransaction VALUES('SL011', 'SW011', 'TP004', 2)
INSERT INTO SalesDetailTransaction VALUES('SL012', 'SW012', 'TP007', 6)
INSERT INTO SalesDetailTransaction VALUES('SL013', 'SW017', 'TP009', 4)
INSERT INTO SalesDetailTransaction VALUES('SL014', 'SW019', 'TP005', 10)
INSERT INTO SalesDetailTransaction VALUES('SL015', 'SW020', 'TP008', 9)
INSERT INTO SalesDetailTransaction VALUES('SL016', 'SW013', 'TP004', 7)
INSERT INTO SalesDetailTransaction VALUES('SL017', 'SW014', 'TP002', 3)
INSERT INTO SalesDetailTransaction VALUES('SL018', 'SW018', 'TP010', 2)
INSERT INTO SalesDetailTransaction VALUES('SL019', 'SW016', 'TP006', 1)
INSERT INTO SalesDetailTransaction VALUES('SL020', 'SW015', 'TP007', 5)
INSERT INTO SalesDetailTransaction VALUES('SL008', 'SW014', 'TP002', 6)
INSERT INTO SalesDetailTransaction VALUES('SL011', 'SW003', 'TP008', 6)
INSERT INTO SalesDetailTransaction VALUES('SL002', 'SW007', 'TP001', 8)
INSERT INTO SalesDetailTransaction VALUES('SL003', 'SW009', 'TP003', 9)
INSERT INTO SalesDetailTransaction VALUES('SL014', 'SW018', 'TP010', 7)
INSERT INTO SalesDetailTransaction VALUES('SL015', 'SW014', 'TP002', 4)

--PurchaseHeaderTransaction
INSERT INTO PurchaseHeaderTransaction VALUES('PR001', 'DT007', 'SF005', '2018-03-22')
INSERT INTO PurchaseHeaderTransaction VALUES('PR002', 'DT005', 'SF007', '2018-05-15')
INSERT INTO PurchaseHeaderTransaction VALUES('PR003', 'DT008', 'SF008', '2018-06-20')
INSERT INTO PurchaseHeaderTransaction VALUES('PR004', 'DT003', 'SF002', '2018-08-01')
INSERT INTO PurchaseHeaderTransaction VALUES('PR005', 'DT002', 'SF001', '2019-01-10')
INSERT INTO PurchaseHeaderTransaction VALUES('PR006', 'DT006', 'SF003', '2019-02-28')
INSERT INTO PurchaseHeaderTransaction VALUES('PR007', 'DT001', 'SF004', '2019-04-24')
INSERT INTO PurchaseHeaderTransaction VALUES('PR008', 'DT004', 'SF006', '2019-05-05')
INSERT INTO PurchaseHeaderTransaction VALUES('PR009', 'DT011', 'SF009', '2018-09-25')
INSERT INTO PurchaseHeaderTransaction VALUES('PR010', 'DT015', 'SF020', '2018-10-22')
INSERT INTO PurchaseHeaderTransaction VALUES('PR011', 'DT019', 'SF010', '2018-04-14')
INSERT INTO PurchaseHeaderTransaction VALUES('PR012', 'DT016', 'SF017', '2018-07-15')
INSERT INTO PurchaseHeaderTransaction VALUES('PR013', 'DT017', 'SF015', '2018-08-09')
INSERT INTO PurchaseHeaderTransaction VALUES('PR014', 'DT013', 'SF012', '2018-05-16')
INSERT INTO PurchaseHeaderTransaction VALUES('PR015', 'DT009', 'SF018', '2018-02-08')
INSERT INTO PurchaseHeaderTransaction VALUES('PR016', 'DT012', 'SF013', '2018-01-02')
INSERT INTO PurchaseHeaderTransaction VALUES('PR017', 'DT014', 'SF014', '2018-06-14')
INSERT INTO PurchaseHeaderTransaction VALUES('PR018', 'DT010', 'SF019', '2018-05-25')
INSERT INTO PurchaseHeaderTransaction VALUES('PR019', 'DT020', 'SF016', '2018-12-20')
INSERT INTO PurchaseHeaderTransaction VALUES('PR020', 'DT018', 'SF011', '2018-11-30')
```

```
--PurchaseDetailTransaction
INSERT INTO PurchaseDetailTransaction VALUES('PR001','SW004', 3)
INSERT INTO PurchaseDetailTransaction VALUES('PR002','SW005', 4)
INSERT INTO PurchaseDetailTransaction VALUES('PR003','SW001', 1)
INSERT INTO PurchaseDetailTransaction VALUES('PR004','SW004', 9)
INSERT INTO PurchaseDetailTransaction VALUES('PR005','SW003', 8)
INSERT INTO PurchaseDetailTransaction VALUES('PR006','SW002', 5)
INSERT INTO PurchaseDetailTransaction VALUES('PR007','SW007', 6)
INSERT INTO PurchaseDetailTransaction VALUES('PR008','SW008', 7)
INSERT INTO PurchaseDetailTransaction VALUES('PR009','SW006', 3)
INSERT INTO PurchaseDetailTransaction VALUES('PR010','SW015', 4)
INSERT INTO PurchaseDetailTransaction VALUES('PR011','SW016', 1)
INSERT INTO PurchaseDetailTransaction VALUES('PR012','SW012', 9)
INSERT INTO PurchaseDetailTransaction VALUES('PR013','SW009', 8)
INSERT INTO PurchaseDetailTransaction VALUES('PR014','SW014', 5)
INSERT INTO PurchaseDetailTransaction VALUES('PR015','SW010', 6)
INSERT INTO PurchaseDetailTransaction VALUES('PR016','SW011', 7)
INSERT INTO PurchaseDetailTransaction VALUES('PR017','SW017', 3)
INSERT INTO PurchaseDetailTransaction VALUES('PR018','SW018', 4)
INSERT INTO PurchaseDetailTransaction VALUES('PR019','SW012', 5)
INSERT INTO PurchaseDetailTransaction VALUES('PR020','SW013', 9)
INSERT INTO PurchaseDetailTransaction VALUES('PR005','SW019', 3)
INSERT INTO PurchaseDetailTransaction VALUES('PR006','SW020', 5)
INSERT INTO PurchaseDetailTransaction VALUES('PR007','SW016', 6)
INSERT INTO PurchaseDetailTransaction VALUES('PR008','SW017', 2)
INSERT INTO PurchaseDetailTransaction VALUES('PR001','SW018', 1)
INSERT INTO PurchaseDetailTransaction VALUES('PR010','SW008', 6)
INSERT INTO PurchaseDetailTransaction VALUES('PR020','SW014', 2)
INSERT INTO PurchaseDetailTransaction VALUES('PR016','SW009', 1)
```

*D. Create query using DML syntax to simulate the transactions process for **sales andpurchase transactions**.*

Note: DML syntax to **fill database** and DML syntax to **simulate** the **transactions process** should be a **different query**.

```
--Purchase Simulation
INSERT INTO PurchaseHeaderTransaction VALUES ('PR021', 'DT004', 'SF002', '2019-03-20')
INSERT INTO PurchaseHeaderTransaction VALUES ('PR022', 'DT001', 'SF014', '2019-02-08')
INSERT INTO PurchaseHeaderTransaction VALUES ('PR023', 'DT002', 'SF003', '2019-6-10')

INSERT INTO PurchaseDetailTransaction VALUES ('PR021', 'SW007', 10)
INSERT INTO PurchaseDetailTransaction VALUES ('PR022', 'SW010', 8)
INSERT INTO PurchaseDetailTransaction VALUES ('PR022', 'SW016', 4)
INSERT INTO PurchaseDetailTransaction VALUES ('PR023', 'SW005', 3)
```

INSERT INTO PurchaseDetailTransaction VALUES ('PR023', 'SW003', 7)

--Sales Simulation
INSERT INTO SalesHeaderTransaction VALUES ('SL021', 'CS007', 'SF003', '2019-05-28')
INSERT INTO SalesHeaderTransaction VALUES ('SL022', 'CS001', 'SF005', '2019-01-12')
INSERT INTO SalesHeaderTransaction VALUES ('SL023', 'CS005', 'SF002', '2019-11-08')

INSERT INTO SalesDetailTransaction VALUES ('SL021', 'SW008', 'TP009', 7)
INSERT INTO SalesDetailTransaction VALUES ('SL021', 'SW002', 'TP001', 2)
INSERT INTO SalesDetailTransaction VALUES ('SL022', 'SW005', 'TP002', 4)
INSERT INTO SalesDetailTransaction VALUES ('SL022', 'SW006', 'TP010', 5)
INSERT INTO SalesDetailTransaction VALUES ('SL023', 'SW011', 'TP004', 9)

E. *To support database management process in* ***softwAEr House****, Aedwin asks you to providesome queries which give him important data he needs. The requirements requested are:*

1. **Display SoftwareName and Income (obtained by adding ‘Rp. ’ in front of the sum of software price multiplied by quantity) for every sales transaction which the software type is either ‘Web Development’ or ‘Browser’ and the software stock is more than 10.**

```
SELECT SoftwareName,
       [Income] = 'Rp.' + CAST((SoftwarePrice*Quantity) AS VARCHAR)
FROM Software s
JOIN SalesDetailTransaction sdt
ON s.SoftwareId = sdt.SoftwareId
JOIN SoftwareType st
ON st.SoftwareTypeId = sdt.SoftwareTypeId
WHERE SoftwareTypeName IN ('Web Development', 'Browser')
AND SoftwareStock > 10
```

2. **Display distributor company and Total Software Bought (obtained from the sum of quantity bought) for every purchase transaction which handled by a distributor whose name starts with ‘A’ and occurred after the 10th date of every month.**

```
SELECT DistributorCompany, [Total Software Bought] = SUM(Quantity)
FROM Distributor d
JOIN PurchaseHeaderTransaction pht
ON d.DistributorID = pht.DistributorID
JOIN PurchaseDetailTransaction pdt
ON pht.PurchaseID = pdt.PurchaseID
WHERE DistributorName LIKE 'A%'
AND DATEPART(DAY, PurchaseDate) > 10
GROUP BY DistributorCompany
```

3. **Display Average Revenue per Day (obtained by adding ‘Rp. ’in front of the average of software price multiplied by quantity), TransactionDate and Male Staff Count (obtained from distinct count of staff and ended with ‘ person’) for every sales transaction which is handled by male staff and occurred in 2018.**

```sql
SELECT [Average Revenue per Day] =
           'Rp.' + CAST(AVG(SoftwarePrice*Quantity) AS VARCHAR),
       [Transaction Date] = SalesDate,
       [Male Staff Count] =
           CAST(COUNT(DISTINCT sht.StaffID) AS VARCHAR) + ' person'
FROM Software s
JOIN SalesDetailTransaction sdt
ON s.SoftwareId = sdt.SoftwareId
JOIN SalesHeaderTransaction sht
on sht.SalesID = sdt.SalesID
JOIN Staff st
ON st.StaffID = sht.StaffID
WHERE StaffGender = 'Male'
AND YEAR(SalesDate) = 2018
GROUP BY SalesDate
```

4. **Display Gender (obtained from the first letter of Gender), Total Transactions (obtained from number of sales transaction ended with ‘ transaction(s)’) and Total Sold (obtainedfrom sum of quantity ended with ‘ item(s)’) for every sales transactions that is handled by a male staff and the price multiplied with quantity is higher than 100000. And thencombine it with Gender (obtained from the first letter of Gender), Total Transactions (total sales transaction ended with ‘ transaction(s)’) and Total Sold (obtained from sum of quantity ended with ‘ item(s)’) for every sales transactions that is handled by a female staff and the price multiplied with quantity is higher than 200000.**

```sql
SELECT [Gender] = LEFT(StaffGender, 1),
       [Total Transactions] = CAST(COUNT(sht.SalesID) AS VARCHAR) +
                              ' transaction(s)',
       [Total Sold] = CAST(SUM(Quantity) AS VARCHAR) + ' item(s)'
FROM Staff st JOIN SalesHeaderTransaction sht
ON st.StaffID = sht.StaffID
JOIN SalesDetailTransaction sdt
ON sht.SalesID = sdt.SalesID
JOIN Software s
ON sdt.SoftwareId = s.SoftwareId
WHERE StaffGender = 'Male'
AND SoftwarePrice*Quantity > 100000
GROUP BY StaffGender
UNION
```

```
SELECT [Gender] = LEFT(StaffGender, 1),
       [Total Transactions] = CAST(COUNT(sht.SalesID) AS VARCHAR) +
                             ' transaction(s)',
       [Total Sold] = CAST(SUM(Quantity) AS VARCHAR) + ' item(s)'
FROM Staff st JOIN SalesHeaderTransaction sht
ON st.StaffID = sht.StaffID
JOIN SalesDetailTransaction sdt
ON sht.SalesID = sdt.SalesID
JOIN Software s
ON sdt.SoftwareId = s.SoftwareId
WHERE StaffGender = 'Female'
AND SoftwarePrice*Quantity > 200000
GROUP BY StaffGender
```

5. **Display SoftwareId, SoftwareName, SoftwarePrice (obtained by adding ‘Rp. ’in front of SoftwarePrice) for every purchase transaction which SoftwarePrice is higher than the average of SoftwarePrice from every purchase transaction and for every purchase transaction handled by a staff whose StaffId is either ‘SF003’, ‘SF004’, ‘SF009’. Showthe data based on SoftwarePrice in descending order.**

```
SELECT pdt.SoftwareID, SoftwareName,
       SoftwarePrice = 'Rp. ' + CAST(SoftwarePrice AS VARCHAR)
FROM Software s JOIN PurchaseDetailTransaction pdt
ON s.SoftwareId = pdt.SoftwareID
JOIN PurchaseHeaderTransaction pht
ON pht.PurchaseID = pdt.PurchaseID
WHERE
    SoftwarePrice > (SELECT AVG(SoftwarePrice) AS [Avg]
                     FROM Software s, PurchaseDetailTransaction pdt
                     WHERE s.SoftwareId = pdt.SoftwareID)
AND StaffID IN ('SF003', 'SF004', 'SF009')
ORDER BY SoftwarePrice DESC
```

6. **Display Staff First Name (obtained from the staff’s first name), StaffPhone, Transaction Date (obtained from the transaction date in dd mon yyyy format) for everysales transaction which quantity is lower than the average of quantity from all sales transaction and occurred before ‘1 January 2019’.**

```
SELECT [Staff First Name] = LEFT(StaffName,
                            CHARINDEX(' ', StaffName + ' ')-1),
       StaffPhone,
       [Transaction Date] = CONVERT(VARCHAR, SalesDate, 106)
FROM Staff s
JOIN SalesHeaderTransaction sht
ON s.StaffID = sht.StaffID
JOIN SalesDetailTransaction sdt
ON sht.SalesID = sdt.SalesID
```

```
WHERE sdt.Quantity < (SELECT AVG(Quantity) FROM
SalesDetailTransaction)
AND SalesDate < '1 January 2019'
```

7. **Display PurchaseTransactionId, Distributor Last Name (obtained by adding 'Mx. ' infront of distributor's last name), DistributorCompany, TransactionDate (obtained from the transaction date in Mon dd, yyyy format), for every purchase transaction which Software Price is higher than the average price of every software but lower than the maximum price of every software and occurred between the year 2017 and 2018.**

```
SELECT pht.PurchaseID AS PurchaseTransactionId,
       [Distributor Last Name] = 'Mx. ' + SUBSTRING(DistributorName,
             CHARINDEX(' ', DistributorName)+1, LEN(DistributorName)),
       DistributorCompany,
       [TransactionDate] = CONVERT(VARCHAR, PurchaseDate, 107)
FROM PurchaseHeaderTransaction pht, Distributor d,
(
  SELECT PurchaseID, SoftwarePrice
  FROM Software s, PurchaseDetailTransaction pdt
  WHERE s.SoftwareId = pdt.SoftwareID
) AS a,
(
  SELECT AVG(SoftwarePrice) AS [Avg Software Price],
         MAX(SoftwarePrice) AS [Max Software Price]
  FROM Software
) AS b
WHERE pht.DistributorID = d.DistributorID
AND pht.PurchaseID = a.PurchaseID
AND a.SoftwarePrice > [Avg Software Price]
AND a.SoftwarePrice < [Max Software Price]
AND YEAR(PurchaseDate) BETWEEN '2017' AND '2018'
```

8. **Display DistributorName, TransactionDate, Total Transactions (obtained from the number of Transactions and ended with ' transaction(s)') for every purchase transaction where the software's version bought is higher than the average software's version available and the distributor id is either 'DT001', 'DT005', 'DT006'.**

```
 WITH Total_Purchase_Transaction AS
  (
   SELECT PurchaseID, CAST(SoftwareVersion AS FLOAT) AS [Software
   Version], CAST(Quantity AS VARCHAR) + ' transaction(s)' AS [Total
   Transactions]
   FROM PurchaseDetailTransaction pdt, Software s
   WHERE pdt.SoftwareID = s.SoftwareId
  )
```

```
SELECT DistributorName, [Transaction Date] = PurchaseDate,
       [Total Transactions]
FROM PurchaseHeaderTransaction pht
JOIN Distributor d
ON pht.DistributorID = d.DistributorID
JOIN Total_Purchase_Transaction tp
ON tp.PurchaseID = pht.PurchaseID
WHERE tp.[Software Version] >
       (SELECT AVG(CAST(SoftwareVersion AS FLOAT)) FROM Software)
AND pht.DistributorID IN ('DT001', 'DT005','DT006')
```

9. **Create a View named ‘StaffSalesReport’ to display StaffName, StaffGender, Transaction Count (obtained from the number of transactions), Total Sales Income (obtained by adding ‘Rp. ’ in front of the sum software price multiplied by quantity) for every sales transaction which the Total Sales Income is higher than 100000 and thestaff name consists of at least 3 words.**

```
GO
CREATE VIEW StaffSalesReport AS
SELECT StaffName, StaffGender,
       [Transaction Count] = COUNT(sht.SalesID),
       [Total Sales Income] = 'Rp. ' +
            CAST(SUM(SoftwarePrice*Quantity) AS VARCHAR)
FROM Staff st
JOIN SalesHeaderTransaction sht
ON st.StaffID = sht.StaffID
JOIN SalesDetailTransaction sdt
ON sht.SalesID = sdt.SalesID
JOIN Software s
ON s.SoftwareId = sdt.SoftwareId
WHERE StaffName LIKE '% % %'
GROUP BY StaffName, StaffGender
HAVING SUM(SoftwarePrice*Quantity) > 100000
GO
```

10. **Create view named ‘Recurring Members’ to display CustomerName, Total Transactions (obtained from total number of transactions), Total Spent (obtained by adding ‘Rp. ’ in front of total sum of software price multiplied by quantity), for every customer who has done more than 2 transactions and SoftwarePrice is higher than 50000.**

```
GO
CREATE VIEW [Recurring Members] AS
SELECT CustomerName,
       [Total Transactions] = COUNT(sht.SalesID),
       [Total Spent] = 'Rp. ' + CAST(SUM(SoftwarePrice*Quantity) AS
                        VARCHAR)
```

```
FROM Customer c
JOIN SalesHeaderTransaction sht
ON c.CustomerID = sht.CustomerID
JOIN SalesDetailTransaction sdt
ON sht.SalesID = sdt.SalesID
JOIN Software s
ON s.SoftwareId = sdt.SoftwareId
WHERE SoftwarePrice > 50000
GROUP BY CustomerName
HAVING COUNT(sht.SalesID) > 2
GO
```