

DOKUMENTASI PROJECT

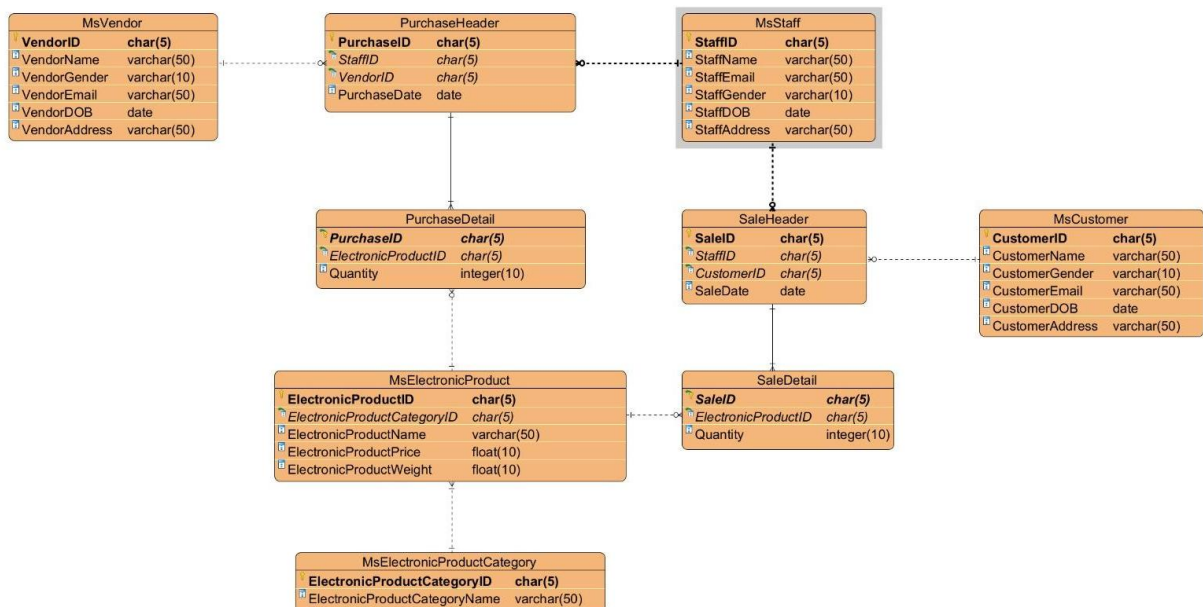
INTRODUCTION TO DATABASE SYSTEMS

KELAS : BB11

KELOMPOK 3 :

- 2702259576 - Vanessa Verensia
- 2702275182 - Edeline Clarissa
- 2702259355 - Giotirta Liweris
- 2702274942 - Delvin Andre Christian

“Jigi BoxZ”



1. ERD (Entity Relationship Diagram) pada project “Jigi BoxZ” terdiri dari 9 tables, yaitu:

- **MsVendor**

Terdiri dari VendorID sebagai **Primary Key**, VendorName, VendorGender, VendorEmail, VendorDOB, dan VendorAddress.

- **MsStaff**

Terdiri dari StaffID sebagai **Primary Key**, StaffName, StaffEmail, StaffGender, StaffDOB, dan StaffAddress.

- **MsCustomer**
Terdiri dari CustomerID sebagai **Primary Key**, CustomerName, CustomerGender, CustomerEmail, CustomerDOB, dan CustomerAddress.
- **MsElectronicProduct**
Terdiri dari ElectronicProductID sebagai **Primary Key**, ElectronicProductCategoryID sebagai **Foreign Key**, ElectronicProductName, ElectronicProductPrice, dan ElectronicProductWeight.
- **MsElectronicProductCategory**
Terdiri dari ElectronicProductCategoryID sebagai **Primary Key** dan ElectronicProductCategoryName.
- **PurchaseHeader**
Terdiri dari PurchaseID sebagai **Primary Key**, StaffID dan VendorID sebagai **Foreign Key**, dan PurchaseDate.
- **PurchaseDetail**
Terdiri dari PurchaseID sebagai **Primary Key sekaligus Foreign Key**, ElectronicProductID sebagai **Foreign Key**, dan Quantity.
- **SaleHeader**
Terdiri dari SaleID sebagai **Primary Key**, StaffID dan CustomerID sebagai **Foreign Key**, dan SaleDate.
- **SaleDetail**
Terdiri dari SaleID sebagai **Primary Key sekaligus Foreign Key**, ElectronicProductID sebagai **Foreign Key**, dan Quantity.

2. Relationships antar tabel:

- **MsStaff - PurchaseHeader : One-to-Many**
Satu staf dapat menangani banyak pembelian.
- **MsVendor - PurchaseHeader : One-to-Many**
Satu vendor dapat dikaitkan dengan banyak pembelian.
- **PurchaseHeader - PurchaseDetail : One-to-Many**
Satu pembayaran bisa memiliki banyak detail pembelian.
- **MsElectronicProduct - PurchaseDetail : One-to-Many**
Satu produk dapat muncul di banyak detail pembelian.

- **MsStaff - SaleHeader : *One-to-Many***
Satu staf dapat menangani banyak penjualan.
- **MsCustomer - SaleHeader : *One-to-Many***
Satu pelanggan dapat dikaitkan dengan banyak penjualan.
- **SaleHeader - SaleDetail : *One-to-Many***
Satu penjualan bisa memiliki banyak detail penjualan.
- **MsElectronicProduct - SaleDetail : *One-to-Many***
Satu produk bisa muncul di banyak detail penjualan.
- **MsElectronicProductCategory - MsElectronicProduct : *One-to-Many***
Satu kategori produk dapat dikaitkan dengan banyak produk.

3. DDL dan DML

Tabel [MsVendor]

- **CREATE TABLE [MsVendor]:** Membuat tabel "MsVendor" untuk menyimpan informasi terkait vendor.
- **Kolom-kolom:**
 - VendorID: CHAR(5), primary key dengan format ID 'VR' diikuti oleh tiga digit angka, tidak boleh null.
 - VendorName: VARCHAR(50), tidak boleh null.
 - VendorGender: VARCHAR(10), harus 'Male', 'Female', tidak boleh null.
 - VendorEmail: VARCHAR(50), tidak boleh null.
 - VendorDOB: DATE, harus dibawah 2000-01-01, tidak boleh null.
 - VendorAddress: VARCHAR(50), tidak boleh null.
- **INSERT INTO MsVendor:** Menambahkan 12 baris data vendor ke tabel "MsVendor".

Tabel [MsStaff]

- **CREATE TABLE [MsStaff]:** Membuat tabel "MsStaff" untuk menyimpan informasi terkait staf.
- **Kolom-kolom:**
 - StaffID: CHAR(5), primary key dengan format ID 'ST' diikuti oleh tiga digit angka, tidak boleh null.
 - StaffName: VARCHAR(50), tidak boleh null.
 - StaffEmail: VARCHAR(50), tidak boleh null.
 - StaffGender: VARCHAR(10), harus 'Male', 'Female', tidak boleh null.
 - StaffDOB: DATE, harus dibawah 2000-01-01, tidak boleh null.
 - StaffAddress: VARCHAR(50), tidak boleh null.

- **INSERT INTO MsStaff:** Menambahkan 12 baris data staf ke tabel "MsStaff".

Tabel [MsElectronicProductCategory]

- **CREATE TABLE [MsElectronicProductCategory]:** Membuat tabel "MsElectronicProductCategory" untuk menyimpan informasi terkait kategori produk elektronik.
- **Kolom-kolom:**
 - ElectronicProductCategoryID: CHAR(5), primary key dengan format ID 'CT' diikuti oleh tiga digit angka, tidak boleh null.
 - ElectronicProductCategoryName: VARCHAR(50), tidak boleh null.
- **INSERT INTO MsElectronicProductCategory:** Menambahkan 12 baris data kategori produk elektronik ke tabel "MsElectronicProductCategory".

Tabel [MsElectronicProduct]

- **CREATE TABLE [MsElectronicProduct]:** Membuat tabel "MsElectronicProduct" untuk menyimpan informasi terkait produk elektronik.
- **Kolom-kolom:**
 - ElectronicProductID: CHAR(5), primary key dengan format ID 'PT' diikuti oleh tiga digit angka, tidak boleh null.
 - ElectronicProductCategoryID: CHAR(5), foreign key yang mereferensi ke "MsElectronicProductCategory".
 - ElectronicProductName: VARCHAR(50), tidak boleh null.
 - ElectronicProductPrice: FLOAT(10), tidak boleh null.
 - ElectronicProductWeight: FLOAT(10), tidak boleh null.
 - SaleDetailElectronicProductID: CHAR(5), tidak boleh null (tidak dijelaskan hubungan referensinya).
- **INSERT INTO MsElectronicProduct:** Menambahkan 12 baris data produk elektronik ke tabel "MsElectronicProduct".

Tabel [MsCustomer]

- **CREATE TABLE [MsCustomer]:** Membuat tabel "MsCustomer" untuk menyimpan informasi pelanggan.
- **Kolom-kolom:**
 - CustomerID: CHAR(5), primary key dengan format ID 'CU' diikuti oleh tiga digit angka, tidak boleh null.
 - CustomerName: VARCHAR(50), tidak boleh null.
 - CustomerGender: VARCHAR(10) harus 'Male', 'Female', tidak boleh null.

- CustomerEmail: VARCHAR(50), tidak boleh null.
- CustomerDOB: DATE, harus dibawah 2000-01-01, tidak boleh null.
- CustomerAddress: VARCHAR(50), tidak boleh null.
- **INSERT INTO MsCustomer:** Menambahkan 12 baris data pelanggan ke tabel "MsCustomer".

Tabel [PurchaseHeader]

- **CREATE TABLE [PurchaseHeader]:** Membuat tabel "PurchaseHeader" untuk menyimpan informasi terkait header pembelian.
- **Kolom-kolom:**
 - PurchaseID: CHAR(5), primary key dengan format ID 'PH' diikuti oleh tiga digit angka, tidak boleh null.
 - StaffID: CHAR(5), foreign key yang mereferensi ke "MsStaff".
 - VendorID: CHAR(5), foreign key yang mereferensi ke "MsVendor".
 - PurchaseDate: DATE, tidak boleh null.
- **INSERT INTO PurchaseHeader:** Menambahkan 17 baris data header pembelian ke tabel "PurchaseHeader".

Tabel [PurchaseDetail]

- **CREATE TABLE [PurchaseDetail]:** Membuat tabel "PurchaseDetail" untuk menyimpan informasi terkait detail pembelian.
- **Kolom-kolom:**
 - PurchaseID: CHAR(5), foreign key yang mereferensi ke "PurchaseHeader".
 - ElectronicProductID: CHAR(5), foreign key yang mereferensi ke "MsElectronicProduct".
 - Quantity: INT, tidak boleh null.
 - PRIMARY KEY (PurchaseID dan ElectronicProductID).
- **INSERT INTO PurchaseDetail:** Menambahkan 27 baris data detail pembelian ke tabel "PurchaseDetail".

Tabel [SaleHeader]

- **CREATE TABLE [SaleHeader]:** Membuat tabel "SaleHeader" untuk menyimpan informasi header penjualan.
- **Kolom-kolom:**
 - SaleID: CHAR(5), primary key dengan format ID 'SH' diikuti oleh tiga digit angka, tidak boleh null.
 - StaffID: CHAR(5), foreign key yang mereferensi ke "MsStaff".
 - CustomerID: CHAR(5), foreign key yang mereferensi ke "MsCustomer".

- SaleDate: DATE, tidak boleh null.
- **INSERT INTO SaleHeader:** Menambahkan 17 baris data header penjualan ke tabel "SaleHeader".

Tabel [SaleDetail]

- **CREATE TABLE [SaleDetail]:** Membuat tabel "SaleDetail" untuk menyimpan informasi terkait detail penjualan.
- **Kolom-kolom:**
 - SaleID: CHAR(5), foreign key yang mereferensi ke "SaleHeader".
 - ElectronicProductID: CHAR(5), foreign key yang mereferensi ke "MsElectronicProduct".
 - Quantity: INT, tidak boleh null.
 - PRIMARY KEY (SaleID dan ElectronicProductID).
- **INSERT INTO SaleDetail:** Menambahkan 30 baris data detail penjualan ke tabel "SaleDetail".

4. Stimulate Transaction

Tabel [MsVendor]

- **INSERT INTO MsVendor:** Menambahkan 12 baris data vendor ke tabel "MsVendor".

Tabel [MsStaff]

- **INSERT INTO MsStaff:** Menambahkan 12 baris data staf ke tabel "MsStaff".

Tabel [MsElectronicProductCategory]

- **INSERT INTO MsElectronicProductCategory:** Menambahkan 12 baris data kategori produk elektronik ke tabel "MsElectronicProductCategory".

Tabel [MsElectronicProduct]

- **INSERT INTO MsElectronicProduct:** Menambahkan 12 baris data produk elektronik ke tabel "MsElectronicProduct".

Tabel [MsCustomer]

- **INSERT INTO MsCustomer:** Menambahkan 12 baris data pelanggan ke tabel "MsCustomer".

Tabel [PurchaseHeader]

- **INSERT INTO PurchaseHeader:** Menambahkan 17 baris data header pembelian ke tabel "PurchaseHeader".

Tabel [PurchaseDetail]

- **INSERT INTO PurchaseDetail:** Menambahkan 27 baris data detail pembelian ke tabel "PurchaseDetail".

Tabel [SaleHeader]

- **INSERT INTO SaleHeader:** Menambahkan 17 baris data header penjualan ke tabel "SaleHeader".

Tabel [SaleDetail]

- **INSERT INTO SaleDetail:** Menambahkan 30 baris data detail penjualan ke tabel "SaleDetail".

INSERT NEW DATA

--NEW VENDOR (menambahkan data baru kedalam table Vendor)

INSERT INTO MsVendor (VendorID, VendorName, VendorGender, VendorEmail, VendorDOB, VendorAddress) VALUES (dengan atribut, VendorID, VendorName, VendorGender, VendorEmail, VendorDOB, VendorAddress)

('VR013', 'Dicky Larson', 'Male', 'dicky098@gmail.com', '1996-07-11', 'Jl. Jelambar Raya No. 13'), (data ke satu)

('VR014', 'Sherry Malvin', 'Female', 'sherrywa123@gmail.com', '1997-11-23', 'Jl. Empang Bahagia No. 10') (data ke dua)

Menambahkan 2 data baru dengan atribut VendorID, VendorName, VendorGender, VendorEmail, VenderDOB, VendorAddress ke dalam table Vendor

–NEW PURCHASE DETAIL (menambahkan data baru kedalam table Purchase Detail)

```
INSERT INTO PurchaseDetail (PurchaseID, ElectronicProductID, Quantity)
VALUES
```

('PH010', 'PT004', 5), (data ke satu)

('PH010', 'PT005', 2), (data ke dua)

('PH010', 'PT006', 3), (data ke tiga)

('PH011', 'PT007', 4), (data ke empat)

('PH011', 'PT008', 2), (data ke lima)

('PH011', 'PT009', 3) (data ke enam)

Menambahkan 6 data baru dengan atribut PurchaseID, ElectronicProductID, Quantity ke dalam table Purchase Detail

(satu PurchaseID dapat memiliki lebih dari satu ElectronicProductID)

--NEW SALE DETAIL (menambahkan data baru kedalam table Sale Detail)

```
INSERT INTO SaleDetail (SaleID, ElectronicProductID, Quantity) VALUES
```

('SH011', 'PT007', 3), (data ke satu)

('SH011', 'PT008', 5), (data ke dua)

('SH011', 'PT009', 6), (data ke tiga)

('SH012', 'PT010', 4), (data ke empat)

('SH012', 'PT011', 2), (data ke lima)

('SH012', 'PT012', 3) (data ke enam)

Menambahkan 6 data baru dengan atribut SaleID, ElectronicProductID, Quantity ke dalam table Sale Detail

(satu SaleID dapat memiliki lebih dari satu ElectronicProductID)

–UPDATE NEW DATA TO PURCHASE DETAIL

UPDATE PurchaseDetail (memperbarui data table PurchaseDetail)

SET Quantity = 10 (memperbarui atau menetapkan quantity baru, yaitu menjadi 10)

WHERE PurchaseID = 'PH010' AND ElectronicProductID = 'PT005' (dimana PurchaseID berupa 'PH010' dan ElectronicProductID berupa 'PT005')

5. Query (10 Soal)

1. Display Customer Name (obtained from the CustomerName starts with 'Mr./Mrs. '), Staff Name (obtained from StaffID concat by ' - ' and StaffName), and Total Transaction Made (obtained from total number of sales) for each sales date occurred in 2023 and Customer that born on 1999.
2. Display the Sales ID (obtained by replacing 'SH' in the salesID with 'SalesID '), ProductName, and Total Product Sold (obtained from the sum of quantity) for each SalesID where the last three digits are even, and the quantity sold was between 1 and 5 (inclusive).
3. Display the PurchaseID, Staff Email (obtained by replacing the '@example.com' with '@DigiBoxZ.com'), VendorName, Total Purchases Made (obtained from the total number of purchases), and Total Quantity (obtained from the sum of quantity) for each staff whose name length more than 5 characters and whose purchase quantity was between 1 and 5 (inclusive).
4. Display the Sales ID (obtained by replacing the 'SH' with 'SalesID '), CustomerName, Sales Date (obtained from salesDate formatted as 'MON DD, YYYY'), Sales Date Quarter (obtained from the quarter of the sales date), Total Product Sold (obtained from the sum of quantity), and Total Sales Transactions Made (obtained from total number of sales) for each sale that occurred in the 1st or 2nd quarter of 2023.
5. Display Product ID (obtained from the last 3 digits of productID), Staff Name (obtained from replacing 'ST' of StaffID with 'Staff' and concatenating it with ' - ' and StaffName), Product Name (obtained by replacing 'PT' with 'Product' and

concatenating it with ' - ' and ProductName), Product Price, and Average Price (calculated from the average of ProductPrice) for each PurchaseID where the last three digits are odd and ProductPrice more than Average Price.

(ALIAS SUBQUERY)

6. Display SalesID, Sales Date (obtained from salesDate formatted as 'DD MON YYYY'), ProductName, CategoryName, Quantity (obtained from concatenating Quantity with ' PCS'), and Product Price (obtained from concatenating ProductPrice with '\$ ' in front of ProductPrice) for each sales occurred on 2023 and TotalPrice(obtained from total of all product price) subtract by ProductPrice must be more than 14000.

(ALIAS SUBQUERY)

7. Display SalesID, Customer Email (obtained by replacing '@example.com' with '@DigiBoxZ.com'), Sales Date (obtained from SalesDate formatted as 'MON DD, YYYY'), ProductName, ProductPrice, and Result of MaxPrice – MinPrice (obtained from maximum of ProductPrice subtract by minimum of ProductPrice) for every sales quantity is more than 4 and the product price is less than or equal to the Result of Max Price - Min Price.

(ALIAS SUBQUERY)

8. Display SalesID, VIP Customer (obtained from 'VIP CUSTOMER', ' - ' concatenated with the CustomerName in uppercase format), Month (obtained from the month of the SalesDate), Total Expenses (obtained by casting the total multiplication of ProductPrice and Quantity and adding '\$ ' in front of Total Expenses) for every sales that the Sales Month occurred in February, April, or July, and the total multiplication of ProductPrice and Quantity is greater than the average total multiplication of ProductPrice and Quantity.

(ALIAS SUBQUERY)

9. Create a view named 'MostAndLessBoughtProductPerCustomer' to display Sales ID (obtained from replacing 'SH' of SalesID with 'Sales ID '), ProductID (obtained from replacing 'PT' of ProductID with 'Product ID '), SalesDate (obtained from

SalesDate formatted as 'MON DD, YYYY'), Most Bought Product (obtained from maximum of Quantity), and Less Bought Product (obtained from minimum of Quantity) for every product that the product weight is more than 180 and the product price is more than 1500.

10. Create a view named 'SpentAboveMaximumTotalSalesCustomerIn2023' to display Customer Name (obtained from uppercase CustomerName), Total Transaction (obtained from total multiplication of ProductPrice and Quantity), and Max Total Transaction (obtained from max of multiplication ProductPrice and Quantity) for each Sales that occurred in 2023 and the Total multiplication of ProductPrice and Quantity is more than maximum of multiplication ProductPrice and Quantity.

Penjelasan :

```
-- 1*
SELECT CONCAT('Mr. ', CustomerName) AS [Customer Name],
CONCAT(ms.StaffID, '-', StaffName) AS [Staff Name],
COUNT(sh.SaleID) AS [Total Transaction Made]
FROM MsCustomer mc JOIN SaleHeader sh
ON mc.CustomerID = sh.CustomerID
JOIN MsStaff ms ON sh.StaffID = ms.StaffID
WHERE CustomerGender LIKE 'Male' AND YEAR(SaleDate) = 2023 AND YEAR(CustomerDOB) = 1999
GROUP BY mc.CustomerName, ms.StaffName, ms.StaffID
UNION
SELECT CONCAT('Mrs. ', CustomerName) AS [Customer Name],
CONCAT(ms.StaffID, '-', StaffName) AS [Staff Name],
COUNT(sh.SaleID) AS [Total Transaction Made]
FROM MsCustomer mc JOIN SaleHeader sh
ON mc.CustomerID = sh.CustomerID
JOIN MsStaff ms ON sh.StaffID = ms.StaffID
WHERE CustomerGender LIKE 'Female' AND YEAR(SaleDate) = 2023 AND YEAR(CustomerDOB) = 1999
GROUP BY mc.CustomerName, ms.StaffName, ms.StaffID
```

1. Query ini menggunakan CONCAT untuk menggabungkan value CustomerName serta StaffName. COUNT digunakan untuk mendapatkan jumlah transaksi. WHERE digunakan agar output yang dihasilkan sesuai dengan ketentuan, ketentuannya adalah Gender Customer adalah Male dan Tahun penjualan pada 2023 dan Tahun kelahiran Customer adalah 1999 serta Gender Customer adalah Female dan Tahun penjualan pada 2023 dan Tahun kelahiran Customer adalah 1999. UNION digunakan agar dapat menggabungkan dua buah line query. Sehingga query yang sudah dibuat akan menghasilkan output yang sesuai dengan soal.

```
--2*
SELECT REPLACE(sd.SaleID, 'SH', 'SalesID') AS [Sales ID],
ElectronicProductName AS ProductName,
SUM(sd.Quantity) AS [Total Product Sold]
FROM SaleDetail sd JOIN MsElectronicProduct mep
ON sd.ElectronicProductID = mep.ElectronicProductID
WHERE RIGHT(sd.SaleID, 3) % 2 = 0 AND Quantity BETWEEN 1 AND 5
GROUP BY REPLACE(sd.SaleID, 'SH', 'SalesID'), ElectronicProductName
```

2. Query ini menggunakan REPLACE untuk mengganti bagian teks dalam kolom SaleID pada tabel SaleDetail. Fungsi ini mengubah teks 'SH' di awal SaleID menjadi 'SalesID'. SUM digunakan untuk menghitung jumlah produk elektronik yang terjual pada setiap SalesID dan ProductName di tabel SaleDetail. Dan WHERE digunakan agar output yang dihasilkan sesuai dengan ketentuan pada soal.

```
--3*
SELECT pd.PurchaseID,
REPLACE(StaffEmail, '@example.com', '@DigiBoxZ.com') AS [Staff Email],
VendorName,
COUNT(pd.PurchaseID) AS [Total Purchases Made],
SUM(pd.Quantity) AS [Total Quantity]
FROM PurchaseHeader ph JOIN MsStaff ms
ON ms.StaffID = ph.StaffID
JOIN MsVendor mv ON mv.VendorID = ph.VendorID
JOIN PurchaseDetail pd ON ph.PurchaseID = pd.PurchaseID
WHERE LEN(StaffName) > 5 AND pd.Quantity BETWEEN 1 AND 5
GROUP BY pd.PurchaseID, REPLACE(StaffEmail, '@example.com', '@DigiBoxZ.com'), VendorName
```

3. Query ini menggunakan REPLACE untuk mengganti '@example.com' menjadi '@DigiBoxZ.com' dari kolom StaffEmail. COUNT digunakan untuk menghitung total pembelian. SUM digunakan untuk menghitung Total Quantity. WHERE digunakan agar output yang dihasilkan sesuai dengan ketentuan pada soal, yaitu panjang karakter pada StaffName lebih dari 5 dan quantity pembelian antara 1 dan 5.

```
--4*
SELECT REPLACE(sh.SaleID, 'SH', 'SalesID') AS [Sales ID],
CustomerName,
CONVERT(VARCHAR, SaleDate, 107) AS [Sales Date],
DATEPART(QUARTER, SaleDate) AS [Sales Date Quarter],
SUM(sd.Quantity) AS [Total Product Sold],
COUNT(sh.SaleID) AS [Total Sales Transactions Made]
FROM MsCustomer mc JOIN SaleHeader sh
ON mc.CustomerID = sh.CustomerID
JOIN SaleDetail sd ON sh.SaleID = sd.SaleID
WHERE YEAR(SaleDate) = 2023 AND DATEPART(QUARTER, SaleDate) IN (1, 2)
GROUP BY REPLACE(sh.SaleID, 'SH', 'SalesID'), CustomerName, CONVERT(VARCHAR, SaleDate, 107), DATEPART(QUARTER, SaleDate)
```

4. Query ini menggunakan REPLACE untuk mengganti bagian teks dalam kolom SaleID pada tabel SaleDetail. Fungsi ini mengubah teks 'SH' di awal SaleID menjadi 'SalesID'. CONVERT digunakan untuk mengkonversi kolom SaleDate pada tabel SaleDetail dari tipe data date ke tipe data varchar. DATEPART digunakan untuk mengekstrak komponen tertentu dari tanggal atau waktu. QUARTER digunakan untuk mengekstrak nomor kuartal dari kolom SaleDate pada tabel SaleDetail. SUM digunakan untuk menghitung jumlah produk elektronik yang terjual pada setiap SalesID dan ProductName di tabel SaleDetail. COUNT digunakan untuk menghitung jumlah transaksi penjualan pada tabel SaleDetail. Dan WHERE digunakan agar output yang dihasilkan sesuai dengan ketentuan, ketentuannya adalah tahun penjualan tahun 2023.

```
--5*
SELECT DISTINCT RIGHT(mep.ElectronicProductID, 3) AS [Product ID],
CONCAT(REPLACE(ms.StaffID, 'ST', 'Staff'), '-', StaffName) AS [Staff Name],
CONCAT(REPLACE(mep.ElectronicProductID, 'PT', 'Product'), '-', ElectronicProductName) AS [Product Name],
ElectronicProductPrice AS [Product Price], z.AveragePrice AS [Average Price]
FROM MsElectronicProduct mep JOIN SaleDetail sd
ON mep.ElectronicProductID = sd.ElectronicProductID
JOIN SaleHeader sh ON sd.SaleID = sh.SaleID
JOIN MsStaff ms ON sh.StaffID = ms.StaffID
JOIN PurchaseHeader ph ON ms.StaffID = ph.StaffID,
(SELECT AVG(ElectronicProductPrice) AS AveragePrice FROM MsElectronicProduct)
as z
WHERE RIGHT(PurchaseID, 3) % 2 = 1 AND ElectronicProductPrice > z.AveragePrice
```

5. Query ini menggunakan DISTINCT agar data yang berulang tidak muncul kembali. RIGHT digunakan untuk mengambil data dari kanan sesuai dengan ketentuan soal, yaitu 3 karakter dari kanan pada kolom ElectronicProductID. REPLACE digunakan untuk mengganti 'ST' menjadi 'Staff' dari value StaffID dan menggantikan 'PT' menjadi 'Product' dari kolom ElectronicProductID. CONCAT digunakan menggabungkan value StaffID dengan StaffName dan menggabungkan kolom ElectronicProductID dengan ElectronicProductName. AVERAGE digunakan untuk menghitung rata-rata dari ElectronicProductPrice. WHERE digunakan agar output yang dihasilkan sesuai dengan ketentuan pada soal, yaitu 3 karakter dari kanan pada kolom PurchaseID yang tidak habis dibagi 2 dan ElectronicProductPrice lebih besar dari AveragePrice.

```
--6*
SELECT sh.SaleID AS SalesID,
    CONVERT(VARCHAR, SaleDate, 106) AS [Sales Date],
    ElectronicProductName AS ProductName,
    ElectronicProductCategoryName AS CategoryName,
    CONCAT(sd.Quantity, ' PCS') AS [Quantity],
    CONCAT('$', ElectronicProductPrice) AS [Product Price]
FROM SaleHeader sh JOIN SaleDetail sd
ON sh.SaleID = sd.SaleID
JOIN MsElectronicProduct mep ON sd.ElectronicProductID = mep.ElectronicProductID
JOIN MsElectronicProductCategory mepc ON mep.ElectronicProductCategoryID = mepc.ElectronicProductCategoryID,
(SELECT SUM(ElectronicProductPrice) AS TotalPrice FROM MsElectronicProduct)
as z
WHERE YEAR(SaleDate) = 2023 AND (z.TotalPrice - ElectronicProductPrice) > 14000
```

6. CONVERT digunakan untuk mengubah format data dari kolom SaleDate pada tabel SaleHeader menjadi format teks dengan format "yyyy-MM-dd". CONCAT digunakan untuk menambahkan 'PCS' di belakang quantity dan '\$' didepan harga produk elektronik. SUM digunakan untuk menjumlahkan harga produk elektronik dari tabel MsElectronicProduct. WHERE digunakan agar output yang dihasilkan sesuai dengan ketentuan, ketentuannya adalah tahun penjualan pada tahun 2023.

```
--7*
SELECT sh.SaleID AS SalesID,
    STUFF(CustomerEmail, CHARINDEX('@', CustomerEmail), LEN(CustomerEmail), '@DigiBoxZ.com') AS [Customer Email],
    CONVERT(VARCHAR, SaleDate, 107) AS [Sales Date],
    ElectronicProductName AS ProductName,
    ElectronicProductPrice AS ProductPrice,
    z.[Result of MaxPrice - MinPrice]
FROM MsCustomer mc JOIN SaleHeader sh ON mc.CustomerID = sh.CustomerID
JOIN SaleDetail sd ON sh.SaleID = sd.SaleID
JOIN MsElectronicProduct mep ON sd.ElectronicProductID = mep.ElectronicProductID,
(SELECT MAX(ElectronicProductPrice) - MIN(ElectronicProductPrice) AS [Result of MaxPrice - MinPrice] FROM MsElectronicProduct)z
WHERE sd.Quantity > 4 AND ElectronicProductPrice <= z.[Result of MaxPrice - MinPrice]
```

7. Query ini menggunakan STUFF mengganti domain CustomerEmail menjadi DigiboxZ.com. CONVERT digunakan untuk mengubah format data dari kolom SaleDate pada tabel SaleHeader menjadi format 'Mon dd, yyyy'. MAX digunakan untuk mencari nilai maksimum dari kolom ElectronicProductPrice. MIN digunakan untuk mencari nilai minimum dari kolom ElectronicProductPrice. WHERE digunakan untuk agar output yang dihasilkan sesuai dengan ketentuan pada soal, yaitu Quantity lebih dari 4 dan ElectronicProductPrice lebih kecil dari sama dengan hasil dari pengurangan MaxPrice dikurang dengan MinPrice.


```
--8
SELECT sh.SaleID AS SalesID,
CONCAT('VIP CUSTOMER - ', UPPER(CustomerName)) AS [VIP Customer],
MONTH(sh.SaleDate) AS "Month",
CONCAT('$ ', CAST(SUM(mep.ElectronicProductPrice * sd.Quantity) AS VARCHAR(10))) AS [Total Expenses]
FROM SaleHeader sh JOIN SaleDetail sd ON sh.SaleID = sd.SaleID
JOIN MsElectronicProduct mep ON sd.ElectronicProductID = mep.ElectronicProductID
JOIN MsCustomer mc ON sh.CustomerID = mc.CustomerID,
(SELECT AVG(mep.ElectronicProductPrice * sd.Quantity) Average FROM SaleDetail sd JOIN MsElectronicProduct mep ON sd.ElectronicProductID = mep.ElectronicProductID)z
WHERE
MONTH(sh.SaleDate) IN (2, 4, 7)
AND mep.ElectronicProductPrice * sd.Quantity > z.Average
GROUP BY sh.SaleID, mc.CustomerName, sh.SaleDate
```

8. CONCAT digunakan untuk menambahkan VIP CUSTOMER - didepan CustomerName yang sudah diubah menjadi huruf besar secara keseluruhan dan menggabungkan '\$' pada total expenses. MONTH digunakan untuk menentukan bulan pada value data. AVG digunakan untuk mencari rata-rata. Dan WHERE digunakan untuk mencari bulan penjualan antara 2, 4, atau 7 dan hasil perkalian dari ElectronicProductPrice dengan Quantity lebih dari average.

```
--9*
CREATE VIEW MostAndLessBoughtProductPerCustomer AS
SELECT REPLACE(sh.SaleID, 'SH', 'Sales ID') AS [Sales ID],
REPLACE(mep.ElectronicProductID, 'PT', 'Product ID') AS ProductID,
CONVERT(VARCHAR, SaleDate, 107) AS SalesDate,
MAX(pd.Quantity) AS [Most Bought Product],
MIN(pd.Quantity) AS [Less Bought Product]
FROM SaleHeader sh JOIN SaleDetail sd ON sh.SaleID = sd.SaleID
JOIN MsElectronicProduct mep ON sd.ElectronicProductID = mep.ElectronicProductID
JOIN PurchaseDetail pd ON mep.ElectronicProductID = pd.ElectronicProductID
WHERE ElectronicProductWeight > 180 AND ElectronicProductPrice > 1500
GROUP BY REPLACE(sh.SaleID, 'SH', 'Sales ID'),
REPLACE(mep.ElectronicProductID, 'PT', 'Product ID'),
CONVERT(VARCHAR, SaleDate, 107)

SELECT * FROM MostAndLessBoughtProductPerCustomer
```

9. Query ini menggunakan REPLACE untuk mengganti 'SH' menjadi 'Sales ID' dari kolom SaleID dan mengganti 'PT' menjadi 'Product ID' dari kolom ElectronicProductID. CONVERT digunakan untuk mengubah format data dari kolom SaleDate pada tabel SaleHeader menjadi format 'Mon dd, yyyy'. MAX digunakan untuk mencari nilai maksimum dari Quantity. MIN digunakan untuk mencari nilai minimum dari Quantity. WHERE digunakan agar output yang dihasilkan sesuai dengan ketentuan, yaitu ElectronicProductWeight lebih dari 180 dan ElectronicProductPrice lebih dari 1500.

```
--10*
CREATE VIEW SpendAboveMaximumTotalSalesCustomerIn2023 AS
SELECT UPPER(CustomerName) AS [Customer Name],
SUM(ElectronicProductPrice*sd.Quantity) AS [Total Transaction],
MAX(ElectronicProductPrice*sd.Quantity) AS [Max Total Transaction]
FROM MsCustomer mc JOIN SaleHeader sh ON mc.CustomerID = sh.CustomerID
JOIN SaleDetail sd ON sh.SaleID = sd.SaleID
JOIN MsElectronicProduct mep ON sd.ElectronicProductID = mep.ElectronicProductID
WHERE YEAR(SaleDate) = 2023
GROUP BY UPPER(CustomerName)
```

10. UPPER digunakan untuk mengubah CustomerName menjadi huruf besar secara keseluruhan.
SUM digunakan untuk menjumlahkan ElectronicProductPrice dikalikan dengan Quantity.
MAX digunakan untuk mengambil nilai maksimum dari hasil kali ElectronicProductPrice dengan Quantity. WHERE digunakan untuk mencari tahun penjualan pada tahun 2023.

