

**LAPORAN PRAKTIKUM**  
**MATA KULIAH DATA WAREHOUSE**

Dosen Pengampu : Vipkas Al Hadid Firdaus, S.T., M.T.

**JOBSHEET 2: DATABASE OPERASIONAL**



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2024**

## TUGAS 1

1. Import data perusahaan tersebut pada DBMS MySQL!
2. Analisa struktur data dari database perusahaan tersebut, dalam bentuk tabel, analisa hubungan setiap tabel nya!

### Jawab:

Berikut merupakan daftar tabel yang ada pada db mysqlsampledatabse.sql:

>	customers
>	employees
>	offices
>	orderdetails
>	orders
>	payments
>	productlines
>	products

Berikut merupakan struktur tiap databasenya:

#### a. Customers

Q	Field varchar	Type blob	Null varchar	Key string	Default blob	Extra varchar
>	customerNumber	int	NO	PRI	(NULL)	
>	customerName	varchar(50)	NO		(NULL)	
>	contactLastName	varchar(50)	NO		(NULL)	
>	contactFirstName	varchar(50)	NO		(NULL)	
>	phone	varchar(50)	NO		(NULL)	
>	addressLine1	varchar(50)	NO		(NULL)	
>	addressLine2	varchar(50)	YES		(NULL)	
>	city	varchar(50)	NO		(NULL)	
>	state	varchar(50)	YES		(NULL)	
>	postalCode	varchar(15)	YES		(NULL)	
>	country	varchar(50)	NO		(NULL)	
>	salesRepEmployeeN	int	YES	MUL	(NULL)	
>	creditLimit	decimal(10,2)	YES		(NULL)	

b. Employees

Field varchar	Type blob	Null varchar	Key string	Default blob	Extra varchar
employeeNumber	int	NO	PRI	(NULL)	
lastName	varchar(50)	NO		(NULL)	
firstName	varchar(50)	NO		(NULL)	
extension	varchar(10)	NO		(NULL)	
email	varchar(100)	NO		(NULL)	
officeCode	varchar(10)	NO	MUL	(NULL)	
reportsTo	int	YES	MUL	(NULL)	
jobTitle	varchar(50)	NO		(NULL)	

c. Offices

Field varchar	Type blob	Null varchar	Key string	Default blob	Extra varchar
officeCode	varchar(10)	NO	PRI	(NULL)	
city	varchar(50)	NO		(NULL)	
phone	varchar(50)	NO		(NULL)	
addressLine1	varchar(50)	NO		(NULL)	
addressLine2	varchar(50)	YES		(NULL)	
state	varchar(50)	YES		(NULL)	
country	varchar(50)	NO		(NULL)	
postalCode	varchar(15)	NO		(NULL)	
territory	varchar(10)	NO		(NULL)	

d. Orderdetails

Field varchar	Type blob	Null varchar	Key string	Default blob	Extra varchar
orderNumber	int	NO	PRI	(NULL)	
productCode	varchar(15)	NO		(NULL)	
quantityOrdered	int	NO		(NULL)	
priceEach	decimal(10,2)	NO		(NULL)	
orderLineNumber	smallint	NO		(NULL)	

e. Orders

Field varchar	Type blob	Null varchar	Key string	Default blob	Extra varchar
orderNumber	int	NO	PRI	(NULL)	
orderDate	date	NO		(NULL)	
requiredDate	date	NO		(NULL)	
shippedDate	date	YES		(NULL)	
status	varchar(15)	NO		(NULL)	
comments	text	YES		(NULL)	
customerNumber	int	NO	MUL	(NULL)	

f. Payments

Field varchar	Type blob	Null varchar	Key string	Default blob	Extra varchar
customerNumber	int	NO	PRI	(NULL)	
checkNumber	varchar(50)	NO		(NULL)	
paymentDate	date	NO		(NULL)	
amount	decimal(10,2)	NO		(NULL)	

g. Productlines

Field varchar	Type blob	Null varchar	Key string	Default blob	Extra varchar
productLine	varchar(50)	NO	PRI	(NULL)	
textDescription	varchar(4000)	YES		(NULL)	
htmlDescription	mediumtext	YES		(NULL)	
image	mediumblob	YES		(NULL)	

h. Products

Field varchar	Type blob	Null varchar	Key string	Default blob	Extra varchar
productCode	varchar(15)	NO	PRI	(NULL)	
productName	varchar(70)	NO		(NULL)	
productLine	varchar(50)	NO	MUL	(NULL)	
productScale	varchar(10)	NO		(NULL)	
productVendor	varchar(50)	NO		(NULL)	
productDescription	text	NO		(NULL)	
quantityInStock	smallint	NO		(NULL)	
buyPrice	decimal(10,2)	NO		(NULL)	
MSRP	decimal(10,2)	NO		(NULL)	

Berikut merupakan relasinya:

Table 1	Table 2	Relasi
productlines	products	1 : M
offices	employees	1 : M
employees	employees	1 : M
employees	customers	1 : M
customers	payments	1 : M
customers	orders	1 : M
orders	ordersdetails	1 : M
products	ordersdetails	1 : M

3. Analisa jumlah field pada setiap tabel!

**Jawab:**

Nama Table	Jumlah Fields
productlines	4
products	9
ordersdetails	5
offices	9
customers	13
payments	4
orders	7
employees	8

## PRAKTIKUM 1

1. Jalankan query berikut:

```
SELECT * FROM employees as employe, employees AS manager, customers as cust
WHERE employe.reportsTo = manager.employeeNumber
AND employe.employeeNumber = cust.salesRepEmployeeNumber; 20ms
```

Maka hasil dari query tersebut adalah data Employee beserta Manajernya dan Customer yang ia miliki. Perhatikan hasil data dengan seksama:

employeeNumber int	lastName varchar	firstName varchar	extension varchar	email varchar	officeCode varchar	reportsTo int	jobTitle varchar	employeeNumber int	lastName varchar	firstName varchar	extension varchar
1165	Jennings	Leslie	x3291	ljennings@clas	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1165	Jennings	Leslie	x3291	ljennings@clas	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1165	Jennings	Leslie	x3291	ljennings@clas	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1165	Jennings	Leslie	x3291	ljennings@clas	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1165	Jennings	Leslie	x3291	ljennings@clas	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1166	Thompson	Leslie	x4065	lthompson@cl	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1166	Thompson	Leslie	x4065	lthompson@cl	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1166	Thompson	Leslie	x4065	lthompson@cl	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1166	Thompson	Leslie	x4065	lthompson@cl	1	1143	Sales Rep	1143	Bow	Anthony	x5428
1166	Thompson	Leslie	x4065	lthompson@cl	1	1143	Sales Rep	1143	Bow	Anthony	x5428

2. Buka tab baru pada browser untuk melakukan eksekusi query berikut:

```
SELECT
  manager.employeeNumber as id_manager,
  CONCAT(manager.firstName, ' ', manager.lastName) as manager,
  employe.employeeNumber as id_staff,
  CONCAT(employe.firstName, ' ', employe.lastName) as staff
FROM
  employees as employe, employees AS manager
WHERE
  employe.reportsTo = manager.employeeNumber
ORDER BY
  manager.firstName ASC; 8ms
```

dari hasil query diatas maka akan ditemukan atasan dari setiap pegawai.

id_manager int	manager varchar	id_staff int	staff varchar
1143	Anthony Bow	1165	Leslie Jennings
1143	Anthony Bow	1166	Leslie Thompson
1143	Anthony Bow	1188	Julie Firrelli
1143	Anthony Bow	1216	Steve Patterson
1143	Anthony Bow	1286	Foon Yue Tseng
1143	Anthony Bow	1323	George Vanauf
1002	Diane Murphy	1056	Mary Patterson

## TUGAS 2

1. Gambarkanlah hirarki organisasi berdasarkan atasan dari setiap pegawai sesuai dengan hasil prkatikum diatas!

**Jawab:**

1143	Anthony Bow	1165	Leslie Jennings
1143	Anthony Bow	1166	Leslie Thompson
1143	Anthony Bow	1188	Julie Firrelli
1143	Anthony Bow	1216	Steve Patterson
1143	Anthony Bow	1286	Foon Yue Tseng
1143	Anthony Bow	1323	George Vanauf
1002	Diane Murphy	1056	Mary Patterson
1002	Diane Murphy	1076	Jeff Firrelli
1102	Gerard Bondur	1337	Loui Bondur
1102	Gerard Bondur	1370	Gerard Hernandez
1102	Gerard Bondur	1401	Pamela Castillo
1102	Gerard Bondur	1501	Larry Bott
1102	Gerard Bondur	1504	Barry Jones
1102	Gerard Bondur	1702	Martin Gerard
1621	Mami Nishi	1625	Yoshimi Kato
1056	Mary Patterson	1088	William Patterson
1056	Mary Patterson	1102	Gerard Bondur
1056	Mary Patterson	1143	Anthony Bow
1056	Mary Patterson	1621	Mami Nishi
1088	William Patterson	1611	Andy Fixter
1088	William Patterson	1612	Peter Marsh
1088	William Patterson	1619	Tom King

2. Buka tab baru pada browser untuk melakukan eksekusi query berikut:

```
SELECT
    manager.employeeNumber as id_manager,
    CONCAT(manager.firstName, " ", manager.lastName) as Manager,
    employee.employeeNumber as id_staff,
    CONCAT(employee.firstName, " ", employee.lastName) as staff,
    COUNT(cust.customerNumber) as total_cust
FROM
    employees employee
JOIN
    employees manager
ON
    employee.reportsTo = manager.employeeNumber
LEFT JOIN
    customers cust
ON
    employee.employeeNumber = cust.salesRepEmployeeNumber
GROUP BY
    manager.employeeNumber, employee.employeeNumber
ORDER BY
    manager.firstName; 14ms
```

dari query tersebut menghasilkan jumlah customer dari setiap staff:

id_manager	Manager	id_staff	staff	total_cust
int	varchar	int	varchar	bigint
1143	Anthony Bow	1165	Leslie Jennings	6
1143	Anthony Bow	1166	Leslie Thompson	6
1143	Anthony Bow	1188	Julie Firrelli	6
1143	Anthony Bow	1216	Steve Patterson	6
1143	Anthony Bow	1286	Foon Yue Tseng	7
1143	Anthony Bow	1323	George Vanauf	8
1002	Diane Murphy	1056	Mary Patterson	0
1002	Diane Murphy	1076	Jeff Firrelli	0
1102	Gerard Bondur	1337	Loui Bondur	6
1102	Gerard Bondur	1370	Gerard Hernandez	7
1102	Gerard Bondur	1401	Pamela Castillo	10
1102	Gerard Bondur	1501	Larry Bott	8
1102	Gerard Bondur	1504	Barry Jones	9
1102	Gerard Bondur	1702	Martin Gerard	6
1621	Mami Nishi	1625	Yoshimi Kato	0
1056	Mary Patterson	1088	William Patterson	0
1056	Mary Patterson	1102	Gerard Bondur	0
1056	Mary Patterson	1143	Anthony Bow	0
1056	Mary Patterson	1621	Mami Nishi	5
1088	William Patterson	1611	Andy Fixter	5
1088	William Patterson	1612	Peter Marsh	5
1088	William Patterson	1619	Tom King	0



### TUGAS 3

1. Siapakah staff dengan hirarki paling bawah yang berprestasi dilihat dari jumlah customer terbanyak?

**Jawab:**

```
SELECT
  employee.employeeNumber as id_staff,
  CONCAT(employee.firstName, " ", employee.lastName) as staff,
  COUNT(cust.customerNumber) as total_cust
FROM
  employees employee
LEFT JOIN
  customers cust
ON
  employee.employeeNumber = cust.salesRepEmployeeNumber
GROUP BY
  employee.employeeNumber
ORDER BY
  total_cust DESC
LIMIT 1; 10ms
```

Query tersebut akan mengembalikan staff dengan jumlah cust terbanyak, yaitu Pamela Castillo dengan 10 customer

id_staff int	staff varchar	total_cust bigint
1401	Pamela Castillo	10

2. Jika KPI atasan dihitung dari customer yang dimilikinya dijumlah dengan customer dari staff dibawahnya, urutkan ranking prestasi keseluruhan pegawai beserta keterangan jumlah customer yang dimilikinya!

**Jawab:**

Menghitung KPI atasan dari jumlah customer yang dimilikinya ditambah dengan customer dari staff di bawahnya

```

WITH RECURSIVE EmployeeHierarchy AS (
    SELECT
        employeeNumber AS id_staff,
        reportsTo AS id_manager,
        0 AS level
    FROM
        employees
    WHERE
        reportsTo IS NOT NULL
    UNION ALL
    SELECT
        e.employeeNumber AS id_staff,
        e.reportsTo AS id_manager,
        eh.level + 1 AS level
    FROM
        employees e
    INNER JOIN
        EmployeeHierarchy eh ON e.reportsTo = eh.id_staff
),
CustomerCount AS (
    SELECT
        e.employeeNumber AS id_staff,
        COUNT(c.customerNumber) AS total_cust
    FROM
        employees e
    LEFT JOIN
        customers c ON e.employeeNumber = c.salesRepEmployeeNumber
    GROUP BY
        e.employeeNumber
),
KPI AS (
    SELECT
        eh.id_manager,
        SUM(cc.total_cust) AS total_cust
    FROM
        EmployeeHierarchy eh
    INNER JOIN
        CustomerCount cc ON eh.id_staff = cc.id_staff
    GROUP BY
        eh.id_manager
)
SELECT
    e.employeeNumber AS id_staff,
    CONCAT(e.firstName, ' ', e.lastName) AS staff,
    IFNULL(cc.total_cust, 0) + IFNULL(kpi.total_cust, 0) AS total_cust
FROM
    employees e
LEFT JOIN
    CustomerCount cc ON e.employeeNumber = cc.id_staff
LEFT JOIN
    KPI kpi ON e.employeeNumber = kpi.id_manager
ORDER BY
    total_cust DESC; 9ms

```

Query ini akan mengembalikan daftar pegawai dengan jumlah customer yang dimilikinya ditambah dengan customer dari staff di bawahnya, diurutkan berdasarkan jumlah customer terbanyak. Hasilnya sebagai berikut:

id_staff int	staff varchar	total_cust decimal
1102	Gerard Bondur	138
1143	Anthony Bow	117
1088	William Patterson	30
1056	Mary Patterson	10
1401	Pamela Castillo	10
1504	Barry Jones	9
1323	George Vanauf	8
1501	Larry Bott	8
1286	Foon Yue Tseng	7
1370	Gerard Hernandez	7
1165	Leslie Jennings	6
1166	Leslie Thompson	6
1188	Julie Firrelli	6
1216	Steve Patterson	6
1337	Loui Bondur	6
1702	Martin Gerard	6
1611	Andy Fixter	5
1612	Peter Marsh	5
1621	Mami Nishi	5
1002	Diane Murphy	0
1076	Jeff Firrelli	0
1619	Tom King	0
1625	Yoshimi Kato	0

- Analisa kembali data LegendVehicle untuk mendapatkan ranking pegawai berdasarkan KPI "Jumlah omset yang didapat". Urutkan ranking pegawai beserta keterangan dana yang didapat!

**Jawab:**

Untuk menghitung KPI pegawai berdasarkan jumlah omset yang didapat, kita perlu menggabungkan tabel employees dan customers dengan tabel orders dan orderdetails untuk mendapatkan total omset yang dihasilkan oleh setiap pegawai.

```

WITH EmployeeSales AS (
    SELECT
        e.employeeNumber AS id_staff,
        CONCAT(e.firstName, ' ', e.lastName) AS staff,
        SUM(od.quantityOrdered * od.priceEach) AS total_sales
    FROM
        employees e
    LEFT JOIN
        customers c ON e.employeeNumber = c.salesRepEmployeeNumber
    LEFT JOIN
        orders o ON c.customerNumber = o.customerNumber
    LEFT JOIN
        orderdetails od ON o.orderNumber = od.orderNumber
    GROUP BY
        e.employeeNumber
)
SELECT
    id_staff,
    staff,
    IFNULL(total_sales, 0) AS total_sales
FROM
    EmployeeSales
ORDER BY
    total_sales DESC; 22ms

```

Query ini akan mengembalikan daftar pegawai dengan jumlah omset yang didapat, diurutkan berdasarkan jumlah omset terbanyak. Berikut adalah hasilnya:

id_staff int	staff varchar	total_sales decimal
1370	Gerard Hernandez	1258577.81
1165	Leslie Jennings	1081530.54
1401	Pamela Castillo	868220.55
1501	Larry Bott	732096.79
1504	Barry Jones	704853.91
1323	George Vanauf	669377.05
1612	Peter Marsh	584593.76
1337	Loui Bondur	569485.75
1611	Andy Fixter	562582.59
1216	Steve Patterson	505875.42
1286	Foon Yue Tseng	488212.67
1621	Mami Nishi	457110.07
1702	Martin Gerard	387477.47
1188	Julie Firrelli	386663.20
1166	Leslie Thompson	347533.03
1002	Diane Murphy	0.00
1056	Mary Patterson	0.00
1076	Jeff Firrelli	0.00
1088	William Patterson	0.00
1102	Gerard Bondur	0.00
1143	Anthony Bow	0.00
1619	Tom King	0.00
1625	Yoshimi Kato	0.00

4. Jika KPI yang pertama merupakan "Jumlah customer yang bertransaksi" sedangkan KPI yang kedua "Jumlah omset yang didapat". Maka, berapakah jumlah field yang dibutuhkan untuk mendapatkan informasi tersebut?

**Jawab:**

Untuk mendapatkan informasi mengenai KPI "Jumlah customer yang bertransaksi" dan "Jumlah omset yang didapat", kita memerlukan beberapa field dari tabel yang berbeda:

- a. Tabel employees:

- employeeNumber (ID Pegawai)
  - firstName (Nama Depan Pegawai)
  - lastName (Nama Belakang Pegawai)
- b. Tabel customers:
- salesRepEmployeeNumber (ID Pegawai yang bertanggung jawab atas customer)
  - customerNumber (ID Customer)
- c. Tabel Orders:
- customerNumber (ID Customer)
  - orderNumber (ID Order)
- d. Tabel OrderDetails:
- orderNumber (ID Order)
  - quantityOrdered (Jumlah Barang yang Dipesan)
  - priceEach (Harga per Barang)

Dengan field-field tersebut, kita dapat menghitung jumlah customer yang bertransaksi dan jumlah omset yang didapat oleh setiap pegawai.

```

WITH CustomerCount AS (
    SELECT
        e.employeeNumber AS id_staff,
        CONCAT(e.firstName, ' ', e.lastName) AS staff,
        COUNT(DISTINCT c.customerNumber) AS total_customers
    FROM
        employees e
    LEFT JOIN
        customers c ON e.employeeNumber = c.salesRepEmployeeNumber
    LEFT JOIN
        orders o ON c.customerNumber = o.customerNumber
    GROUP BY
        e.employeeNumber
),
EmployeeSales AS (
    SELECT
        e.employeeNumber AS id_staff,
        SUM(od.quantityOrdered * od.priceEach) AS total_sales
    FROM
        employees e
    LEFT JOIN
        customers c ON e.employeeNumber = c.salesRepEmployeeNumber
    LEFT JOIN
        orders o ON c.customerNumber = o.customerNumber
    LEFT JOIN
        orderdetails od ON o.orderNumber = od.orderNumber
    GROUP BY
        e.employeeNumber
)
SELECT
    cc.id_staff,
    cc.staff,
    cc.total_customers,
    IFNULL(es.total_sales, 0) AS total_sales
FROM
    CustomerCount cc
LEFT JOIN
    EmployeeSales es ON cc.id_staff = es.id_staff
ORDER BY
    total_sales DESC, total_customers DESC; 18ms

```

Query ini akan mengembalikan daftar pegawai dengan jumlah customer yang bertransaksi dan jumlah omset yang didapat, diurutkan berdasarkan jumlah omset terbanyak dan jumlah customer terbanyak.

id_staff int	staff varchar	total_customers bigint	total_sales decimal
1370	Gerard Hernandez	7	1258577.81
1165	Leslie Jennings	6	1081530.54
1401	Pamela Castillo	10	868220.55
1501	Larry Bott	8	732096.79
1504	Barry Jones	9	704853.91
1323	George Vanauf	8	669377.05
1612	Peter Marsh	5	584593.76
1337	Loui Bondur	6	569485.75
1611	Andy Fixter	5	562582.59
1216	Steve Patterson	6	505875.42
1286	Foon Yue Tseng	7	488212.67
1621	Mami Nishi	5	457110.07
1702	Martin Gerard	6	387477.47
1188	Julie Firrelli	6	386663.20
1166	Leslie Thompson	6	347533.03
1002	Diane Murphy	0	0.00
1056	Mary Patterson	0	0.00
1076	Jeff Firrelli	0	0.00
1088	William Patterson	0	0.00
1102	Gerard Bondur	0	0.00
1143	Anthony Bow	0	0.00
1619	Tom King	0	0.00
1625	Yoshimi Kato	0	0.00

5. Buatlah report pertahun untuk KPI "Jumlah omset yang didapat" pada Foon Yue Tseng dan Pamela Castillo. Serta gambarkan grafiknya (grafik garis).

**Jawab:**

Untuk membuat laporan tahunan untuk KPI "Jumlah omset yang didapat" pada Foon Yue Tseng dan Pamela Castillo, kita perlu melakukan beberapa langkah:

- a. Mengidentifikasi employeeNumber dari Foon Yue Tseng dan Pamela Castillo.

```
-- Mengidentifikasi employeeNumber dari Foon Yue Tseng dan Pamela Castillo
> Run | +Tab | JSON | Ask Copilot
SELECT employeeNumber, firstName, lastName
FROM employees
WHERE firstName IN ('Foon Yue', 'Pamela') AND lastName IN ('Tseng',
'Castillo'); 5ms
```



* employeeNumber	* firstName	* lastName
int	varchar(50)	varchar(50)
1286	Foon Yue	Tseng
1401	Pamela	Castillo

- b. Menghitung jumlah omset yang didapat oleh masing-masing pegawai per tahun.

```
-- Menghitung jumlah omset yang didapat oleh masing-masing pegawai per tahun
> Run | +Tab | JSON | Select | Ask Copilot
SELECT
  e.employeeNumber,
  CONCAT(e.firstName, ' ', e.lastName) AS staff,
  YEAR(o.orderDate) AS year,
  SUM(od.quantityOrdered * od.priceEach) AS total_sales
FROM
  employees e
JOIN
  customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN
  orders o ON c.customerNumber = o.customerNumber
JOIN
  orderdetails od ON o.orderNumber = od.orderNumber
WHERE
  e.employeeNumber IN (1286, 1401)
GROUP BY
  e.employeeNumber, YEAR(o.orderDate)
ORDER BY
  e.employeeNumber, YEAR(o.orderDate); 10ms
```

- c. Menggambar grafik garis berdasarkan data tersebut.

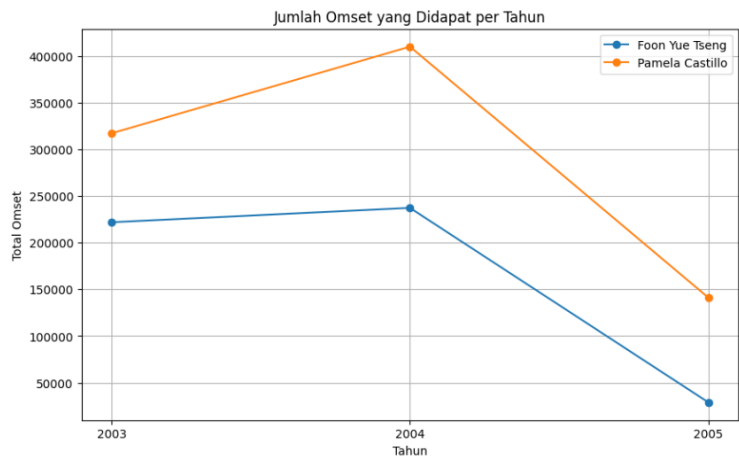
```
import matplotlib.pyplot as plt
import pandas as pd

# Data hasil query SQL
data = {
    'employeeNumber': [1286, 1286, 1286, 1401, 1401, 1401],
    'staff': ['Foon Yue Tseng', 'Foon Yue Tseng', 'Foon Yue Tseng', 'Pamela Castillo', 'Pamela Castillo', 'Pamela Castillo'],
    'year': ['2003', '2004', '2005', '2003', '2004', '2005'],
    'total_sales': [221887.03, 237255.26, 29070.38, 317104.78, 409910.07, 141205.70]
}

# Membuat DataFrame dari data
df = pd.DataFrame(data)

# Membuat grafik garis
plt.figure(figsize=(10, 6))
for staff in df['staff'].unique():
    staff_data = df[df['staff'] == staff]
    plt.plot(staff_data['year'], staff_data['total_sales'], marker='o', label=staff)

plt.title('Jumlah Omset yang Didapat per Tahun')
plt.xlabel('Tahun')
plt.ylabel('Total Omset')
plt.legend()
plt.grid(True)
plt.show()
```



## STUDI KASUS

Pak Huhut merupakan pemegang saham LegendVehicle. dia membutuhkan dashboard untuk melihat perkembangan penjualan (omset) di setiap cabang di tiap tahunnya. Dikarenakan perusahaan tersebut belum merekrut Data Engineer maka, penarikan informasi hanya bisa dilakukan melalui OLTP yang ada.

### Jawab:

1. Field apa saja yang diperlukan untuk menampilkan penjualan di setiap cabang

### Jawab:

Berikut adalah field yang digunakan dalam query ini beserta tabel asalnya dan fungsinya:

- offices.city (dari tabel offices). Digunakan sebagai nama cabang (Branch) dalam hasil output dan untuk mengelompokkan data penjualan berdasarkan cabang (via GROUP BY)
- orders.orderDate (dari tabel orders). Digunakan untuk menentukan tahun penjualan (2003, 2004, atau 2005) dengan fungsi YEAR().
- orderdetails.quantityOrdered (dari tabel orderdetails). Jumlah barang yang dipesan dalam setiap pesanan. Digunakan untuk menghitung omzet dengan mengalikan jumlah ini dengan harga satuan (priceEach).
- orderdetails.priceEach (dari tabel orderdetails). Harga satuan setiap item dalam pesanan. Digabungkan dengan quantityOrdered untuk menghitung total omzet ( $\text{quantityOrdered} * \text{priceEach}$ )
- orders.orderNumber (dari tabel orders dan orderdetails). Digunakan sebagai kunci untuk menggabungkan tabel orders dan orderdetails melalui JOIN
- orders.customerNumber (dari tabel orders dan customers). Digunakan sebagai kunci untuk menggabungkan tabel orders dan customers melalui JOIN
- customers.customerNumber (dari tabel customers). Kunci primer di tabel customers yang cocok dengan orders.customerNumber untuk proses JOIN.
- customers.salesRepEmployeeNumber (dari tabel customers). Digunakan sebagai kunci untuk menggabungkan tabel customers dan employees melalui JOIN.

- employees.employeeNumber (dari tabel employees). Kunci primer di tabel employees yang cocok dengan customers.salesRepEmployeeNumber untuk proses JOIN
- employees.officeCode (dari tabel employees). Digunakan sebagai kunci untuk menggabungkan tabel employees dan offices melalui JOIN
- offices.officeCode (dari tabel offices). Kunci primer di tabel offices yang cocok dengan employees.officeCode untuk proses JOIN

## 2. Bentuk query dengan memperhatikan relasi antar tabel

```

1 SELECT
2   offices.city AS Branch,
3   SUM(CASE WHEN YEAR(orders.orderDate) = 2003 THEN orderdetails.quantityOrdered * orderdetails.priceEach ELSE 0 END) AS '2003',
4   SUM(CASE WHEN YEAR(orders.orderDate) = 2004 THEN orderdetails.quantityOrdered * orderdetails.priceEach ELSE 0 END) AS '2004',
5   SUM(CASE WHEN YEAR(orders.orderDate) = 2005 THEN orderdetails.quantityOrdered * orderdetails.priceEach ELSE 0 END) AS '2005'
6 FROM
7   orders
8 JOIN
9   orderdetails ON orders.orderNumber = orderdetails.orderNumber
10 JOIN
11  customers ON orders.customerNumber = customers.customerNumber
12 JOIN
13  employees ON customers.salesRepEmployeeNumber = employees.employeeNumber
14 JOIN
15  offices ON employees.officeCode = offices.officeCode
16 GROUP BY
17   offices.city
18 ORDER BY
19   offices.city;

```

Branch	2003	2004	2005
Boston	301781.38	467177.07	123580.17
London	549551.94	706014.52	181384.24
NYC	391175.53	665317.99	101096.20
Paris	969959.90	1465229.84	648571.84
San Francisco	532681.13	517408.62	378973.82
Sydney	304949.11	542996.02	299231.22
Tokyo	267249.40	151761.45	38099.22

**SOAL BONUS:** buatlah report lain dengan sumber data OLTP yang sama, analisa field yang digunakan, bentuk struktur query dan tuliskan dalam tabel serta grafiknya.

**Jawab:**

Soal sebelumnya menampilkan omzet per cabang per tahun. Kali ini, saya akan membuat laporan yang menunjukkan total penjualan per kategori produk (productLine) per cabang. Laporan ini akan berguna untuk Pak Huhut untuk melihat performa penjualan berdasarkan jenis produk di setiap cabang.

a. Struktur Query

```
1 SELECT
2     offices.city AS Branch,
3     productlines.productLine AS Product_Category,
4     SUM(orderdetails.quantityOrdered * orderdetails.priceEach) AS Total_Sales
5 FROM
6     orders
7 JOIN
8     orderdetails ON orders.orderNumber = orderdetails.orderNumber
9 JOIN
10    customers ON orders.customerNumber = customers.customerNumber
11 JOIN
12    employees ON customers.salesRepEmployeeNumber = employees.employeeNumber
13 JOIN
14    offices ON employees.officeCode = offices.officeCode
15 JOIN
16    products ON orderdetails.productCode = products.productCode
17 JOIN
18    productlines ON products.productLine = productlines.productLine
19 GROUP BY
20     offices.city, productlines.productLine
21 ORDER BY
22     offices.city, productlines.productLine;
```

## Hasilnya:

Branch	Product_Category	Total_Sales
Boston	Classic Cars	335968.81
Boston	Motorcycles	118533.76
Boston	Planes	67814.36
Boston	Ships	76402.78
Boston	Trains	17046.51
Boston	Trucks and Buses	103893.36
Boston	Vintage Cars	172879.04
London	Classic Cars	644626.76
London	Motorcycles	141552.32
London	Planes	139653.57
London	Ships	121948.90
London	Trains	32369.86
London	Trucks and Buses	137721.41
London	Vintage Cars	219077.88
NYC	Classic Cars	469355.06
NYC	Motorcycles	189573.30
NYC	Planes	122638.96
NYC	Ships	83107.75
NYC	Trains	24582.00
NYC	Trucks and Buses	140033.20
NYC	Vintage Cars	128299.45
Paris	Classic Cars	1335414.46
Paris	Motorcycles	287669.45
Paris	Planes	307551.29
Paris	Ships	251463.11

## Penjelasan:

- Field untuk Total Penjualan: `quantityOrdered * priceEach` dari `orderdetails` digunakan untuk menghitung total penjualan
- Field untuk Cabang: `offices.city` ditentukan melalui rantai hubungan `orders` → `customers` → `employees` → `offices`.
- Field untuk Kategori Produk: `productlines.productLine` diperoleh melalui hubungan `orderdetails` → `products` → `productlines`
- JOIN Tambahan: Dibandingkan laporan sebelumnya, kita menambahkan `products` dan `productlines` untuk mendapatkan informasi kategori produk

Berikut merupakan grafiknya:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Data yang diberikan
data = {
    'Branch': ['Boston', 'Boston', 'Boston', 'Boston', 'Boston', 'Boston', 'Boston',
              'London', 'London', 'London', 'London', 'London', 'London', 'London',
              'NYC', 'NYC', 'NYC', 'NYC', 'NYC', 'NYC', 'NYC',
              'Paris', 'Paris', 'Paris', 'Paris'],
    'Product_Category': ['Classic Cars', 'Motorcycles', 'Planes', 'Ships', 'Trains', 'Trucks and Buses', 'Vintage Cars',
                        'Classic Cars', 'Motorcycles', 'Planes', 'Ships', 'Trains', 'Trucks and Buses', 'Vintage Cars',
                        'Classic Cars', 'Motorcycles', 'Planes', 'Ships', 'Trains', 'Trucks and Buses', 'Vintage Cars',
                        'Classic Cars', 'Motorcycles', 'Planes', 'Ships'],
    'Total_Sales': [335968.81, 118533.76, 67814.36, 76482.78, 17846.51, 183893.36, 172879.04,
                   644626.76, 141552.32, 139653.57, 121948.90, 32369.86, 137721.41, 219077.88,
                   469355.06, 189573.30, 122638.96, 83187.75, 24582.00, 140033.20, 128299.45,
                   1335414.46, 287669.45, 307551.29, 251463.11]
}

# Membuat DataFrame
df = pd.DataFrame(data)

# Menampilkan tabel data
print("Tabel Total Penjualan per Kategori Produk per Cabang:")
print(df)

# Grafik 1: Clustered Bar Chart
plt.figure(figsize=(12, 6))
sns.barplot(data=df, x='Branch', y='Total_Sales', hue='Product_Category')
plt.title('Total Penjualan per Kategori Produk per Cabang')
plt.xlabel('Cabang')
plt.ylabel('Total Penjualan ($)')
plt.xticks(rotation=45)
plt.legend(title='Kategori Produk', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()

# Grafik 2: Stacked Bar Chart
plt.figure(figsize=(12, 6))
sns.barplot(data=df, x='Branch', y='Total_Sales', hue='Product_Category', dodge=False)
plt.title('Total Penjualan per Kategori Produk per Cabang (Bertumpuk)')
plt.xlabel('Cabang')
plt.ylabel('Total Penjualan ($)')
plt.xticks(rotation=45)
plt.legend(title='Kategori Produk', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

Tabel Total Penjualan per Kategori Produk per Cabang:

Branch	Product_Category	Total_Sales
0 Boston	Classic Cars	335968.81
1 Boston	Motorcycles	118533.76
2 Boston	Planes	67814.36
3 Boston	Ships	76482.78
4 Boston	Trains	17046.51
5 Boston	Trucks and Buses	103893.36
6 Boston	Vintage Cars	172879.04
7 London	Classic Cars	644626.76
8 London	Motorcycles	141552.32
9 London	Planes	139653.57
10 London	Ships	121948.90
11 London	Trains	32369.86
12 London	Trucks and Buses	137721.41
13 London	Vintage Cars	219077.88
14 NYC	Classic Cars	469355.06
15 NYC	Motorcycles	189573.30
16 NYC	Planes	122638.96
17 NYC	Ships	83187.75
18 NYC	Trains	24582.00
19 NYC	Trucks and Buses	140033.20
20 NYC	Vintage Cars	128299.45
21 Paris	Classic Cars	1335414.46
22 Paris	Motorcycles	287669.45
23 Paris	Planes	307551.29
24 Paris	Ships	251463.11

