

LAPORAN PRAKTIKUM
DATA OPERASIONAL



Disusun Oleh :

Qusnul Diah Mawanti

2341760035

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025



Jurusan Teknologi Informasi – Politeknik Negeri
Malang
Jobsheet 2 : Data Operasional
Mata Kuliah : Data Warehouse
Kelas : 2A – SIB
Nama : Qusnul Diah Mawanti
No absen : 25
NIM : 2341760035

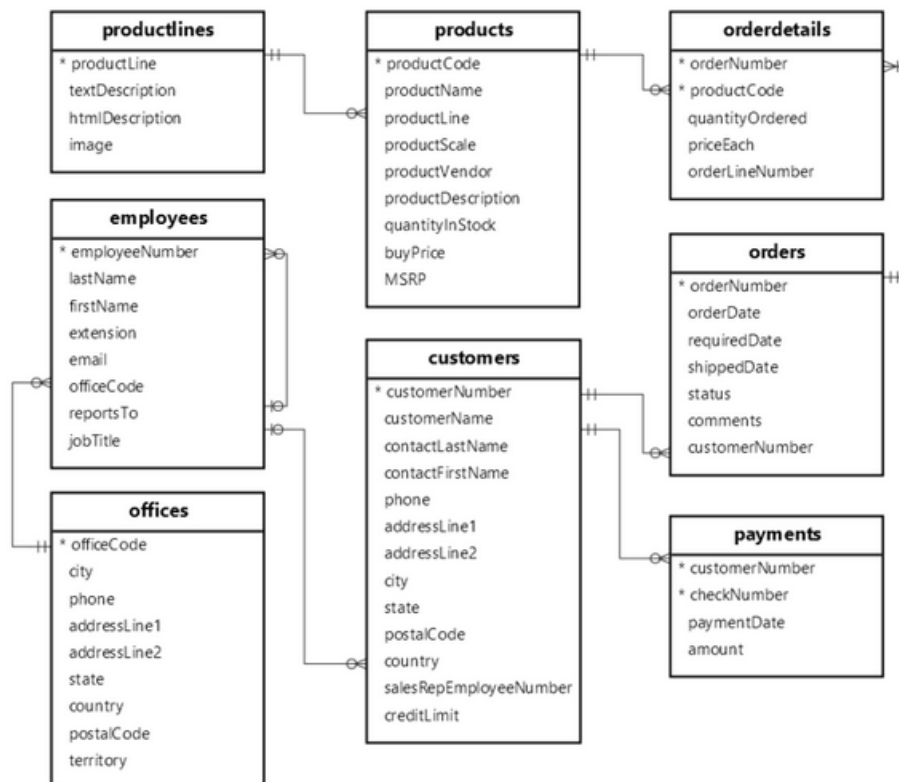
Februari 2025

Tujuan

Setelah melakukan praktikum ini, mahasiswa diharapkan dapat lebih mengenal data sumber, cara menganalisa serta melihat kebutuhan baik fungsional maupun non-fungsional dalam pengembangan data warehouse serta lebih memahami apa itu OLTP.

Studi Kasus

LegendVehicle merupakan perusahaan jual-beli tukar-tambah kendaraan klasik. Perusahaan ini memiliki cabang di berbagai negara. LegendVehicle memiliki sistem informasi ERP sendiri. Salah satu modul dari sistem ERP tersebut adalah modul penjualan. Desain database dari modul tersebut adalah sebagai berikut:



Selain itu proses penjualan kendaraan pada perusahaan tersebut bukan hanya melalui showroom cabang, melainkan reseller-reseller bebas lainnya.

Tugas 1

1. Import data perusahaan tersebut pada DBMS MySQL!

The screenshot shows the phpMyAdmin interface for a database named 'classicmodels'. The 'Structure' tab is selected, displaying a list of 8 tables: customers, employees, offices, orderdetails, orders, payments, productlines, and products. Each table row includes icons for Browse, Structure, Search, Insert, Empty, and Drop, along with statistics for Rows, Type, Collation, Size, and Overhead. A summary row at the bottom indicates 8 tables with a total size of 544.0 KiB.

Table	Action	Rows	Type	Collation	Size	Overhead
customers	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	122	InnoDB	utf8mb4_0900_ai_ci	64.0 KiB	-
employees	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	23	InnoDB	utf8mb4_0900_ai_ci	48.0 KiB	-
offices	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	7	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
orderdetails	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	2,996	InnoDB	utf8mb4_0900_ai_ci	240.0 KiB	-
orders	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	326	InnoDB	utf8mb4_0900_ai_ci	64.0 KiB	-
payments	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	273	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
productlines	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	7	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
products	[Browse] [Structure] [Search] [Insert] [Empty] [Drop]	110	InnoDB	utf8mb4_0900_ai_ci	80.0 KiB	-
8 tables	Sum	3,864	InnoDB	utf8mb4_0900_ai_ci	544.0 KiB	0 B

2. Analisa struktur data dari database perusahaan tersebut, dalam bentuk tabel, analisa hubungan setiap tabel nya!

Tabel 1	Tabel 2	Jenis Relasi	Keterangan
productlines	products	One to Many	Satu kategori produk memiliki banyak produk
products	orderdetails	One to Many	Satu produk bisa masuk ke banyak detail pesanan
orders	orderdetails	One to Many	Satu pesanan memiliki banyak detail pesanan
orders	customers	Many to One	Banyak pesanan bisa berasal dari satu pelanggan
employees	customers	One to Many	Setiap pelanggan memiliki seorang sales representative (pegawai)
customers	payments	One to Many	Setiap pelanggan bisa melakukan lebih dari satu pembayaran
Offices	employees	One to Many	Satu kantor memiliki banyak pegawai

➤ employees

Your SQL query has been executed successfully.

```
DESC employees;
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Extra options

Field	Type	Null	Key	Default	Extra
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	

➤ offices

Your SQL query has been executed successfully.

```
DESC offices;
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Extra options

Field	Type	Null	Key	Default	Extra
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	

➤ customers

Your SQL query has been executed successfully.

```
DESC customers;
```

Field	Type	Null	Key	Default	Extra
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	
salesRepEmployeeNumber	int	YES	MUL	NULL	
creditLimit	decimal(10,2)	YES		NULL	

➤ orderdetails

Your SQL query has been executed successfully.

```
DESC orderdetails;
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

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

➤ orders

Your SQL query has been executed successfully.

```
DESC orders;
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

Field	Type	Null	Key	Default	Extra
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	

➤ payments

Your SQL query has been executed successfully.

```
DESC payments;
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

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

➤ productlines

Your SQL query has been executed successfully.

```
DESC productlines;
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

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

➤ products

Your SQL query has been executed successfully.

```
DESC products;
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

Extra options

Field	Type	Null	Key	Default	Extra
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	

3. Analisa jumlah field pada setiap tabel!

✓ Showing rows 0 - 7 (8 total, Query took 0.0020 seconds.)

```
SELECT TABLE_NAME, COUNT(COLUMN_NAME) AS jumlah_field FROM  
INFORMATION_SCHEMA.COLUMNS WHERE TABLE_SCHEMA = 'classicmodels' GROUP BY  
TABLE_NAME;
```

TABLE_NAME	jumlah_field
customers	13
employees	8
offices	9
orderdetails	5
orders	7
payments	4
productlines	4
products	9

A. Analisis Data

PRAKTIKUM 1

1. Jalankan **query** berikut pada **DBMS MySQL** yang telah tersedia **data Perusahaan LegendVehicle**.

```
1 SELECT e.employeeNumber AS id_staff,  
2       e.firstName AS staff_firstname,  
3       e.lastName AS staff_lastname,  
4       m.employeeNumber AS id_manager,  
5       m.firstName AS manager_firstname,  
6       m.lastName AS manager_lastname,  
7       c.customerNumber AS id_customer,  
8       c.customerName AS customer_name  
9 FROM employees e  
10 JOIN employees m ON e.reportsTo = m.employeeNumber  
11 JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber;  
12
```

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

id_staff	staff_firstname	staff_lastname	id_manager	manager_firstname	manager_lastname	id_customer	customer_name
1165	Leslie	Jennings	1143	Anthony	Bow	124	Mini Gifts Distributors Ltd.
1165	Leslie	Jennings	1143	Anthony	Bow	129	Mini Wheels Co.
1165	Leslie	Jennings	1143	Anthony	Bow	161	Technics Stores Inc.
1165	Leslie	Jennings	1143	Anthony	Bow	321	Corporate Gift Ideas Co.
1165	Leslie	Jennings	1143	Anthony	Bow	450	The Sharp Gifts Warehouse
1165	Leslie	Jennings	1143	Anthony	Bow	487	Signal Collectibles Ltd.
1166	Leslie	Thompson	1143	Anthony	Bow	112	Signal Gift Stores
1166	Leslie	Thompson	1143	Anthony	Bow	205	Toys4GrownUps.com
1166	Leslie	Thompson	1143	Anthony	Bow	219	Boards & Toys Co.
1166	Leslie	Thompson	1143	Anthony	Bow	239	Collectable Mini Designs Co.
1166	Leslie	Thompson	1143	Anthony	Bow	347	Men 'R' US Retailers, Ltd.
1166	Leslie	Thompson	1143	Anthony	Bow	475	West Coast Collectables Co.
1188	Julie	Firrelli	1143	Anthony	Bow	173	Cambridge Collectables Co.
1188	Julie	Firrelli	1143	Anthony	Bow	204	Online Mini Collectables
1188	Julie	Firrelli	1143	Anthony	Bow	320	Mini Creations Ltd.
1188	Julie	Firrelli	1143	Anthony	Bow	339	Classic Gift Ideas, Inc
1188	Julie	Firrelli	1143	Anthony	Bow	379	Collectables For Less Inc.
1188	Julie	Firrelli	1143	Anthony	Bow	495	Diecast Collectables
1216	Steve	Patterson	1143	Anthony	Bow	157	Diecast Classics Inc.
1216	Steve	Patterson	1143	Anthony	Bow	198	Auto-Moto Classics Inc.
1216	Steve	Patterson	1143	Anthony	Bow	286	Marta's Replicas Co.
1216	Steve	Patterson	1143	Anthony	Bow	362	Gifts4AllAges.com
1216	Steve	Patterson	1143	Anthony	Bow	363	Online Diecast Creations Co.
1216	Steve	Patterson	1143	Anthony	Bow	462	FunGiftIdeas.com
1286	Foon Yue	Tseng	1143	Anthony	Bow	151	Muscle Machine Inc

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

```

1  -- Data atasan dari setiap pegawai
2  SELECT m.employeeNumber AS id_manager,
3         CONCAT(m.firstName, " ", m.lastName) AS Manager,
4         e.employeeNumber AS id_staff,
5         CONCAT(e.firstName, " ", e.lastName) AS staff
6  FROM employees e
7  JOIN employees m ON e.reportsTo = m.employeeNumber
8  ORDER BY m.firstName;

```

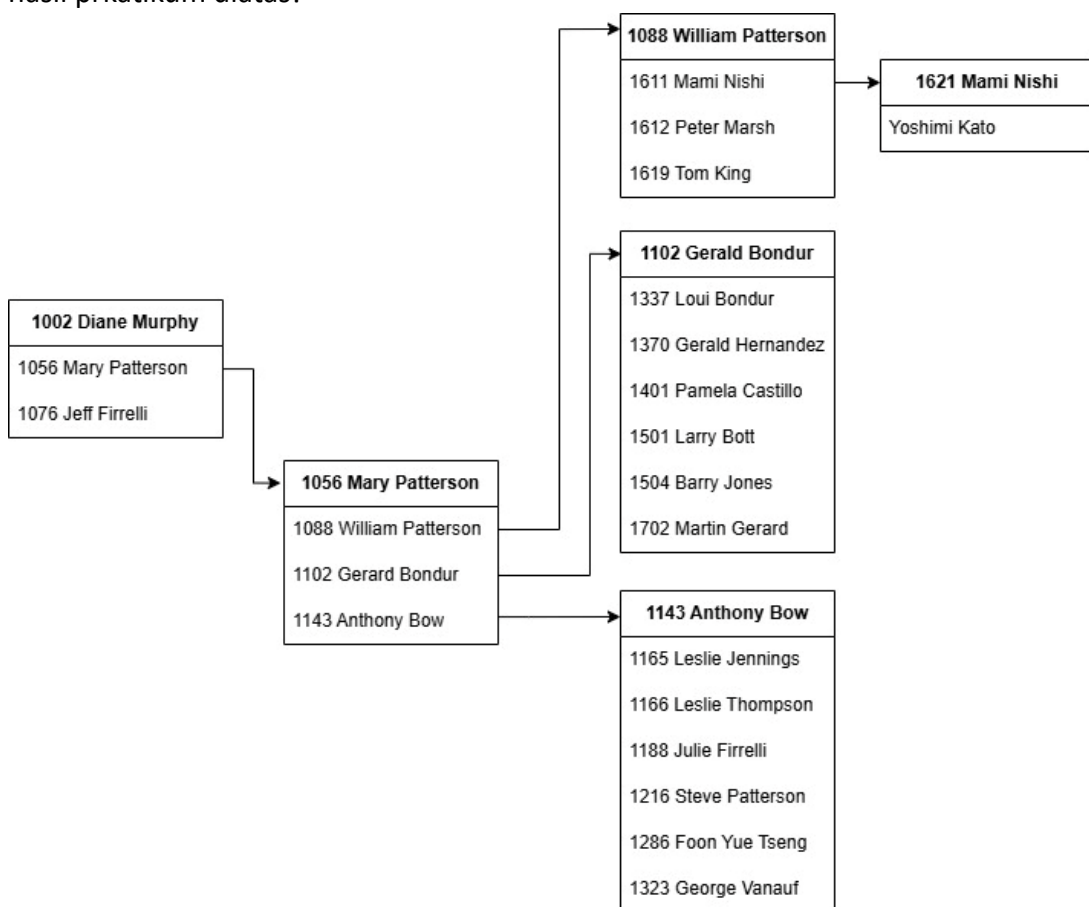
Dari hasil **query** diatas maka akan ditemukan atasan dari setiap pegawai.

id_manager	Manager	id_staff	staff
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

Tugas 2

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



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

```

1 SELECT m.employeeNumber AS id_manager,
2       CONCAT(m.firstName, " ", m.lastName) AS Manager,
3       e.employeeNumber AS id_staff,
4       CONCAT(e.firstName, " ", e.lastName) AS staff,
5       COUNT(c.customerNumber) AS total_cust
6 FROM employees e
7 JOIN employees m ON e.reportsTo = m.employeeNumber
8 LEFT JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
9 GROUP BY e.employeeNumber
10 ORDER BY m.firstName;

```

Dari query tersebut menghasilkan jumlah **customer** dari setiap **staff**.

id_manager	Manager	id_staff	staff	total_cust
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

Jika perusahaan tersebut memiliki **KPI (Key Performances Indicator)** "Jumlah customer yang bertransaksi" maka jawablah pertanyaan-pertanyaan berikut!

Tugas 3

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

```
1 SELECT e.employeeNumber, CONCAT(e.firstName, " ", e.lastName)
2       AS staff, COUNT(c.customerNumber) AS total_customers
3 FROM employees e
4 LEFT JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
5 GROUP BY e.employeeNumber
6 ORDER BY total_customers DESC
7 LIMIT 1;
```

employeeNumber	staff	total_customers
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!

```
1 WITH RECURSIVE EmployeeHierarchy AS (
2     -- Step 1: Ambil semua pegawai
3     SELECT e.employeeNumber, e.reportsTo, e.firstName, e.lastName
4     FROM employees e
5
6     UNION ALL
7
8     -- Step 2: Hubungkan pegawai dengan bawahannya
9     SELECT e.employeeNumber, e.reportsTo, e.firstName, e.lastName
10    FROM employees e
11   INNER JOIN EmployeeHierarchy eh ON e.reportsTo = eh.employeeNumber
12 )
13
14 -- Step 3: Hitung total customer dari pegawai dan semua bawahannya
15 SELECT eh.employeeNumber,
16        CONCAT(eh.firstName, ' ', eh.lastName) AS Employee,
17        COUNT(c.customerNumber) AS total_customers_including_team
18 FROM EmployeeHierarchy eh
19 LEFT JOIN customers c ON eh.employeeNumber = c.salesRepEmployeeNumber
20 GROUP BY eh.employeeNumber, eh.firstName, eh.lastName
21 ORDER BY total_customers_including_team DESC;
```

employeeNumber	Employee	total_customers_including_team
1401	Pamela Castillo	40
1504	Barry Jones	36
1323	George Vanauf	32
1501	Larry Bott	32
1286	Foon Yue Tseng	28
1370	Gerard Hernandez	28
1165	Leslie Jennings	24
1166	Leslie Thompson	24
1188	Julie Firrelli	24
1216	Steve Patterson	24
1337	Loui Bondur	24
1702	Martin Gerard	24
1611	Andy Fixter	20
1612	Peter Marsh	20
1621	Mami Nishi	15
1002	Diane Murphy	0
1056	Mary Patterson	0
1076	Jeff Firrelli	0
1088	William Patterson	0
1102	Gerard Bondur	0
1143	Anthony Bow	0
1619	Tom King	0
1625	Yoshimi Kato	0

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

```

1 SELECT e.employeeNumber, CONCAT(e.firstName, " ", e.lastName) AS staff,
2       SUM(p.amount) AS total_revenue
3 FROM employees e
4 LEFT JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
5 LEFT JOIN payments p ON c.customerNumber = p.customerNumber
6 GROUP BY e.employeeNumber
7 ORDER BY total_revenue DESC;

```

employeeNumber	staff	total_revenue ▾ 1
1370	Gerard Hernandez	1112003.81
1165	Leslie Jennings	989906.55
1401	Pamela Castillo	750201.87
1501	Larry Bott	686653.25
1504	Barry Jones	637672.65
1323	George Vanauf	584406.80
1337	Loui Bondur	569485.75
1611	Andy Fixter	509385.82
1612	Peter Marsh	497907.16
1286	Foon Yue Tseng	488212.67
1621	Mami Nishi	457110.07
1216	Steve Patterson	449219.13
1702	Martin Gerard	387477.47
1188	Julie Firrelli	386663.20
1166	Leslie Thompson	347533.03
1002	Diane Murphy	NULL
1056	Mary Patterson	NULL
1076	Jeff Firrelli	NULL
1088	William Patterson	NULL
1102	Gerard Bondur	NULL
1143	Anthony Bow	NULL
1619	Tom King	NULL
1625	Yoshimi Kato	NULL

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?

KPI	Jumlah Field
Jumlah customer yang bertransaksi	1 (total_customer)
Jumlah omset yang didapat	1 (total_revenue)

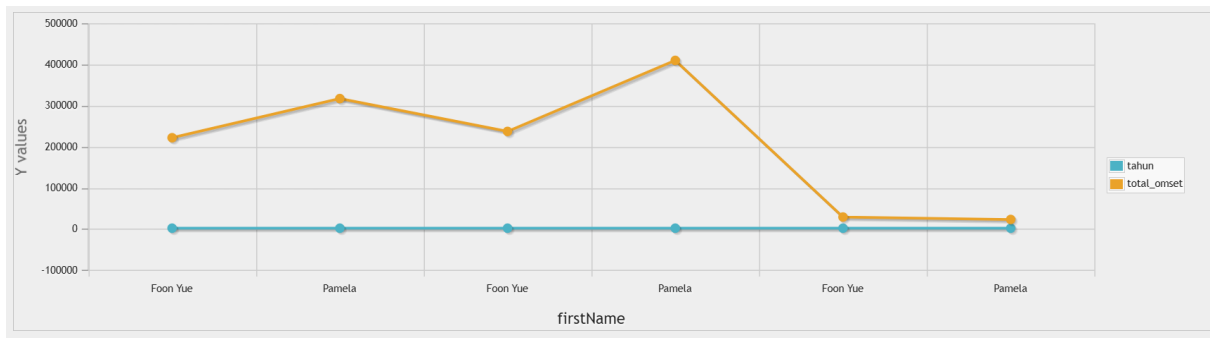
5. Buatlah report pertahun untuk KPI **"Jumlah omset yang didapat"** pada **Foon Yue Tseng** dan **Pamela Castillo**.

```

1 SELECT e.firstName, e.lastName, YEAR(p.paymentDate) AS tahun, SUM(p.amount) AS total_omset
2 FROM employees e
3 JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
4 JOIN payments p ON c.customerNumber = p.customerNumber
5 WHERE e.firstName IN ('Foon Yue', 'Pamela') AND e.lastName IN ('Tseng', 'Castillo')
6 GROUP BY e.firstName, e.lastName, tahun
7 ORDER BY tahun;
```

firstName	lastName	tahun	total_omset
Foon Yue	Tseng	2003	221887.03
Pamela	Castillo	2003	317104.78
Foon Yue	Tseng	2004	237255.26
Pamela	Castillo	2004	409910.07
Foon Yue	Tseng	2005	29070.38
Pamela	Castillo	2005	23187.02

Serta gambarkan grafiknya (grafik garis).



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.

Hasil report yang diinginkan adalah grafik berdasarkan tabel berikut:

Nama Cabang	2003	2004	2005

Analisislah terlebih dahulu:

- Field apa saja yang diperlukan untuk menampilkan penjualan di setiap cabang.
 - Tabel **orders**
 - **orderNumber** → ID transaksi
 - **orderDate** → Tanggal transaksi (digunakan untuk menentukan tahun)
 - **customerNumber** → Pelanggan yang melakukan transaksi
 - Tabel **orderdetails**
 - **orderNumber** → Referensi ke tabel orders
 - **quantityOrdered** → Jumlah produk yang dibeli
 - **priceEach** → Harga satuan barang
 - Tabel **customers**

- **customerNumber** → ID pelanggan
- **salesRepEmployeeNumber** → Pegawai yang menangani transaksi
- Tabel **employees**
 - **employeeNumber** → ID pegawai
 - **officeCode** → Kantor cabang tempat pegawai bekerja
- Tabel **office**
 - **officeCode** → Kode kantor cabang
 - **city** → Nama caban

2. Bentuk query dengan memperhatikan relasi antar tabel.

```

1 SELECT
2     o.city AS Nama_Cabang,
3     SUM(CASE WHEN YEAR(ord.orderDate) = 2003 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2003',
4     SUM(CASE WHEN YEAR(ord.orderDate) = 2004 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2004',
5     SUM(CASE WHEN YEAR(ord.orderDate) = 2005 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2005'
6 FROM orders ord
7 JOIN orderdetails od ON ord.orderNumber = od.orderNumber
8 JOIN customers c ON ord.customerNumber = c.customerNumber
9 JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
10 JOIN offices o ON e.officeCode = o.officeCode
11 GROUP BY o.city
12 ORDER BY o.city;

```

Nama_Cabang	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

Relasi antar Tabel:

- Menghubungkan **orders** dengan **orderdetails**
 - Untuk mendapatkan jumlah dan harga dari setiap item yang dibeli.
- Menghubungkan **orders** dengan **customers**
 - Untuk mengetahui pelanggan yang melakukan transaksi.
- Menghubungkan **customers** dengan **employees**
 - Untuk mengetahui pegawai yang menangani transaksi.
- Menghubungkan **employees** dengan **offices**
 - Untuk mengetahui kantor cabang dari pegawai tersebut.
- Mengelompokkan data berdasarkan **city** (cabang) dan **YEAR**(orderDate) (tahun)
 - Supaya bisa melihat omzet per cabang dalam setiap tahun.

Nama_Cabang	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.

Report untuk menghitung jumlah transaksi yang dilakukan setiap cabang per tahun.

- Field yang dibutuhkan.
 - **orders.orderDate** → Tanggal transaksi
 - **offices.city** → Nama cabang
 - **orders.orderNumber** → Untuk menghitung jumlah transaksi
- Query Jumlah Transaksi per Cabang per Tahun.

```

1 SELECT o.city AS Nama_Cabang,
2        YEAR(ord.orderDate) AS Tahun,
3        COUNT(ord.orderNumber) AS Jumlah_Transaksi
4 FROM orders ord
5 JOIN customers c ON ord.customerNumber = c.customerNumber
6 JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
7 JOIN offices o ON e.officeCode = o.officeCode
8 GROUP BY o.city, YEAR(ord.orderDate)
9 ORDER BY Tahun, Jumlah_Transaksi DESC;

```

- Output Tabel

Nama_Cabang	Tahun ▲ 1	Jumlah_Transaksi ▼ 2
Paris	2003	34
London	2003	18
San Francisco	2003	17
NYC	2003	14
Sydney	2003	12
Boston	2003	9
Tokyo	2003	7
Paris	2004	49
London	2004	24
NYC	2004	22
Boston	2004	18
San Francisco	2004	17
Sydney	2004	15
Tokyo	2004	6
Paris	2005	23
San Francisco	2005	14
Sydney	2005	11
Boston	2005	5
London	2005	5
NYC	2005	3
Tokyo	2005	3

4. Grafik

