DATA WEREHOUSE

JOBSHEET2

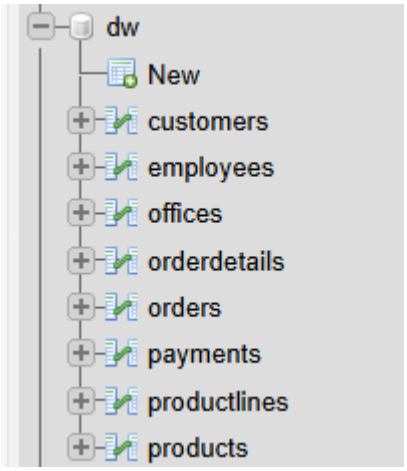
DATABASE OPERATIONAL

NAMA :Bayu Triwibowo

KELAS/NO.ABSEN :SIB 2A/07

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!

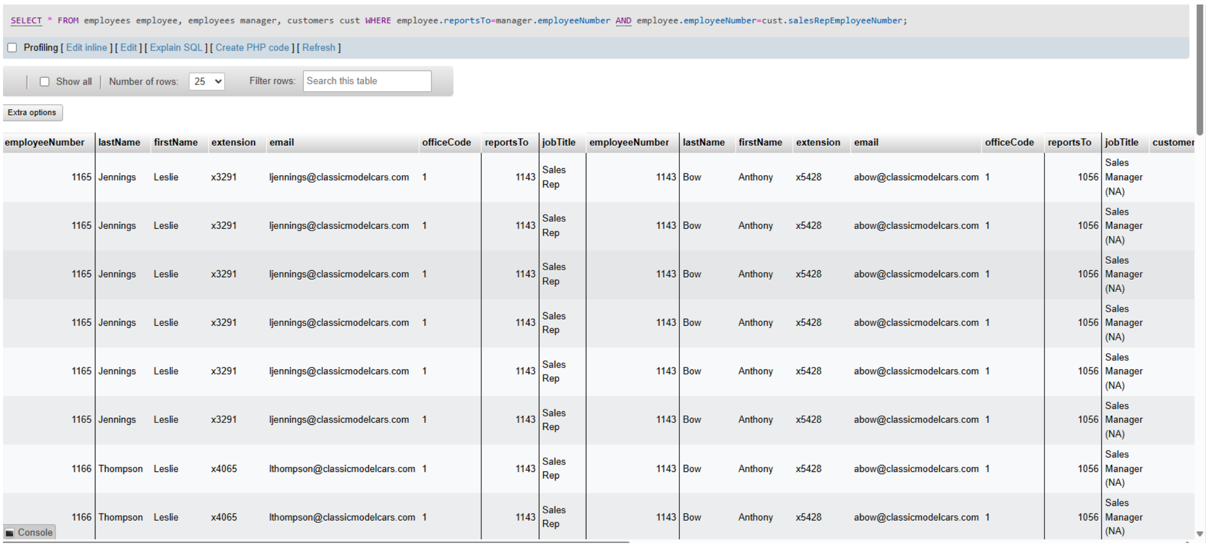
Tabel 1	Tabel 2	Tabel 3
productlines	product	One to many
product	orderdetails	One to many
orders	orderdetails	One to many
customers	orders	One to many
customers	payments	One to many
employees	customers	One to many
offices	employees	One to many
employess	employees	One to many

3. Analisa jumlah field pada setiap tabel!
   Jumlah Field per Tabel
   1. **customers** (Pelanggan) - **13 field**
      - customerNumber, customerName, contactLastName, contactFirstName, phone, addressLine1, addressLine2, city, state, postalCode, country, salesRepEmployeeNumber, creditLimit
   2. **orders** (Pesanan) - **6 field**

   - orderNumber, orderDate, requiredDate, shippedDate, status, comments, customerNumber
3. **orderdetails** (Detail Pesanan) - **4 field**
   - orderNumber, productCode, quantityOrdered, priceEach, orderLineNumber
4. **products** (Produk) - **9 field**
   - productCode, productName, productLine, productScale, productVendor, productDescription, quantityInStock, buyPrice, MSRP
5. **productlines** (Kategori Produk) - **4 field**
   - productLine, textDescription, htmlDescription, image
6. **payments** (Pembayaran) - **4 field**
   - customerNumber, checkNumber, paymentDate, amount
7. **employees** (Karyawan) - **6 field**
   - employeeNumber, lastName, firstName, extension, email, officeCode, reportsTo, jobTitle
8. **offices** (Kantor) - **9 field**
   - officeCode, city, phone, addressLine1, addressLine2, state, country, postalCode, territory

PRAKTIKUM 1

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



```
SELECT * FROM employees employee, employees manager, customers cust WHERE employee.reportsTo=manager.employeeNumber AND employee.employeeNumber=cust.salesRepEmployeeNumber;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

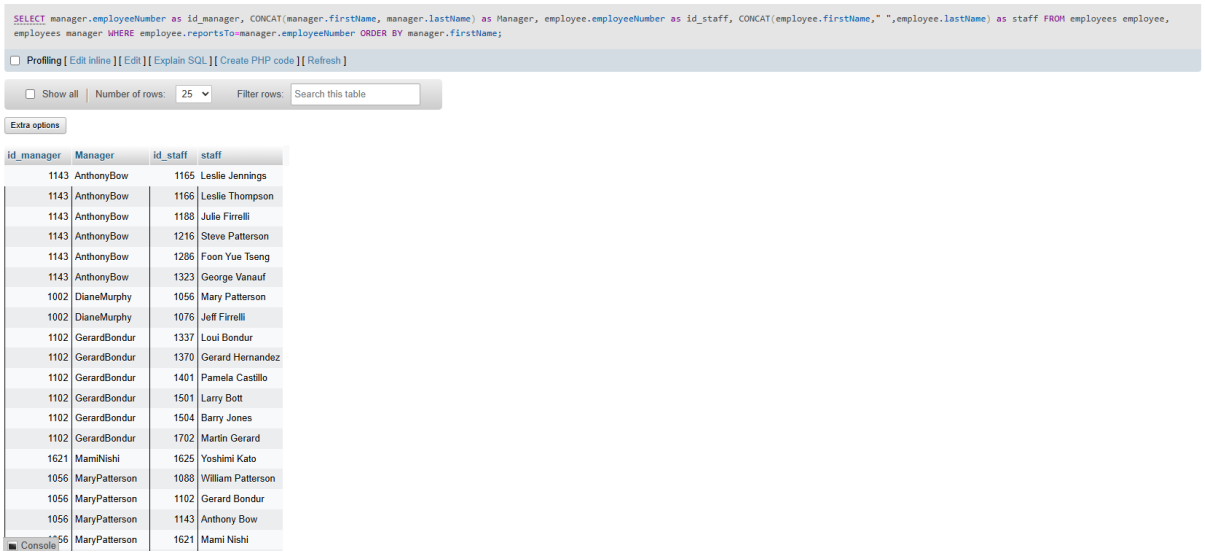
Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle	employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle	customer
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	
1165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	
1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	
1166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1	1056	Sales Manager (NA)	

Console

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 FROM employees employee,
employees manager WHERE employee.reportsTo=manager.employeeNumber ORDER BY manager.firstName;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

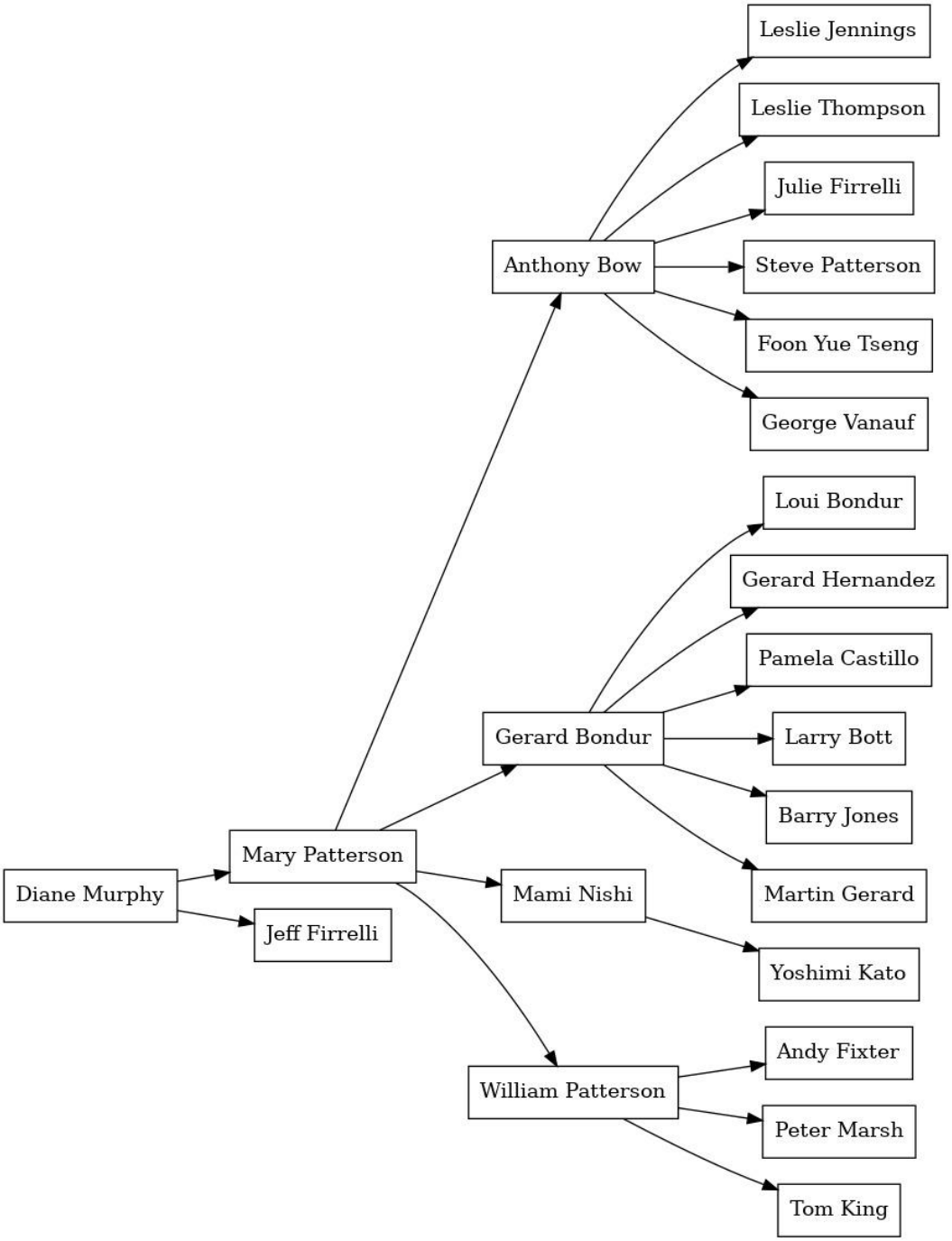
Extra options

id_manager	Manager	id_staff	staff
1143	AnthonyBow	1165	Leslie Jennings
1143	AnthonyBow	1166	Leslie Thompson
1143	AnthonyBow	1188	Julie Firrelli
1143	AnthonyBow	1216	Steve Patterson
1143	AnthonyBow	1286	Foon Yue Tseng
1143	AnthonyBow	1323	George Vanauf
1002	DianeMurphy	1056	Mary Patterson
1002	DianeMurphy	1076	Jeff Firrelli
1102	GerardBondur	1337	Loui Bondur
1102	GerardBondur	1370	Gerard Hernandez
1102	GerardBondur	1401	Pamela Castillo
1102	GerardBondur	1501	Larry Bott
1102	GerardBondur	1504	Barry Jones
1102	GerardBondur	1702	Martin Gerard
1621	MamiNishi	1625	Yoshimi Kato
1056	MaryPatterson	1088	William Patterson
1056	MaryPatterson	1102	Gerard Bondur
1056	MaryPatterson	1143	Anthony Bow
1056	MaryPatterson	1621	Mami Nishi

Console

TUGAS 2

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



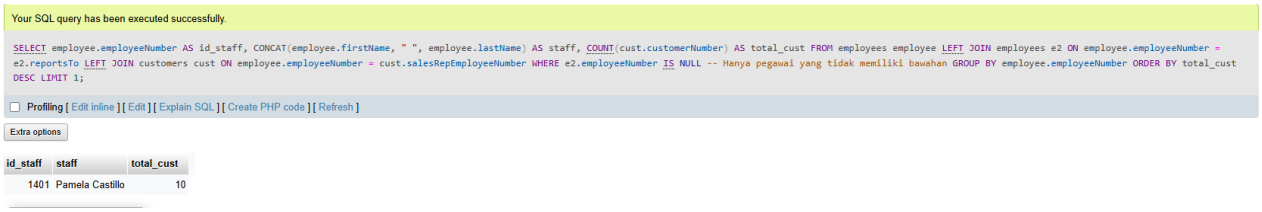
2. Buka **tab baru** pada browser untuk melakukan eksekusi **query** berikut:

```
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 employee.employeeNumber, manager.employeeNumber, manager.firstName, manager.lastName, employee.firstName, employee.lastName ORDER BY manager.firstName;
```

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

TUGAS 3

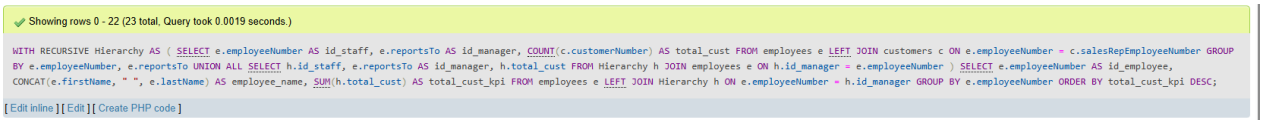
1. Siapakah staff dengan hirarki paling bawah yang berprestasi dilihat dari jumlah customer terbanyak?



```
Your SQL query has been executed successfully.

SELECT employee.employeeNumber AS id_staff, CONCAT(employee.firstName, " ", employee.lastName) AS staff, COUNT(cust.customerNumber) AS total_cust FROM employees employee LEFT JOIN employees e2 ON employee.employeeNumber =
e2.reportsTo LEFT JOIN customers cust ON employee.employeeNumber = cust.salesRepEmployeeNumber WHERE e2.employeeNumber IS NULL -- Hanya pegawai yang tidak memiliki bawahan GROUP BY employee.employeeNumber ORDER BY total_cust
DESC LIMIT 1;

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Extra options
```

id_staff	staff	total_cust
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!



```
Showing rows 0 - 22 (23 total, Query took 0.0019 seconds.)

WITH RECURSIVE Hierarchy AS ( SELECT e.employeeNumber AS id_staff, e.reportsTo AS id_manager, COUNT(c.customerNumber) AS total_cust FROM employees e LEFT JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber GROUP
BY e.employeeNumber, e.reportsTo UNION ALL SELECT h.id_staff, e.reportsTo AS id_manager, h.total_cust FROM Hierarchy h JOIN employees e ON h.id_manager = e.employeeNumber ) SELECT e.employeeNumber AS id_employee,
CONCAT(e.firstName, " ", e.lastName) AS employee_name, SUM(h.total_cust) AS total_cust_kpi FROM employees e LEFT JOIN Hierarchy h ON e.employeeNumber = h.id_manager GROUP BY e.employeeNumber ORDER BY total_cust_kpi DESC;

[ Edit inline ] [ Edit ] [ Create PHP code ]
```

	id_employee	employee_name	total_cust_kpi
Edit Copy Delete	1002	Diane Murphy	100
Edit Copy Delete	1056	Mary Patterson	100
Edit Copy Delete	1102	Gerard Bondur	46
Edit Copy Delete	1143	Anthony Bow	39
Edit Copy Delete	1088	William Patterson	10
Edit Copy Delete	1621	Mami Nishi	0
Edit Copy Delete	1076	Jeff Firrelli	NULL
Edit Copy Delete	1165	Leslie Jennings	NULL
Edit Copy Delete	1166	Leslie Thompson	NULL
Edit Copy Delete	1188	Julie Firrelli	NULL
Edit Copy Delete	1216	Steve Patterson	NULL
Edit Copy Delete	1286	Foon Yue Tseng	NULL
Edit Copy Delete	1323	George Vanauf	NULL
Edit Copy Delete	1337	Loui Bondur	NULL
Edit Copy Delete	1370	Gerard Hernandez	NULL
Edit Copy Delete	1401	Pamela Castillo	NULL
Edit Copy Delete	1501	Larry Bott	NULL
Edit Copy Delete	1504	Barry Jones	NULL
Edit Copy Delete	1611	Andy Fixter	NULL
Edit Copy Delete	1612	Peter Marsh	NULL
Edit Copy Delete	1619	Tom King	NULL
Edit Copy Delete	1625	Yoshimi Kato	NULL
Edit Copy Delete	1702	Martin Gerard	NULL

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

```
SELECT e.employeeNumber AS id_employee, CONCAT(e.firstName, " ", e.lastName) AS employee_name, SUM(p.amount) AS total_revenue FROM employees e JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber JOIN payments p
ON c.customerNumber = p.customerNumber GROUP BY e.employeeNumber ORDER BY total_revenue DESC;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id_employee	employee_name	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

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?
   Jawaban:
   Jumlah field yang dibutuhkan untuk mendapatkan informasi KPI:
   - **Jumlah customer yang bertransaksi:** Dibutuhkan satu field, yaitu **Customer ID** atau **Nama Customer**, untuk menghitung jumlah unik customer yang melakukan transaksi.
   - **Jumlah omset yang didapat:** Dibutuhkan satu field, yaitu **Total Omset (Revenue)**, yang menunjukkan total pendapatan dari transaksi.

   Total jumlah field yang dibutuhkan = 2 field (Customer ID dan Total Omset).

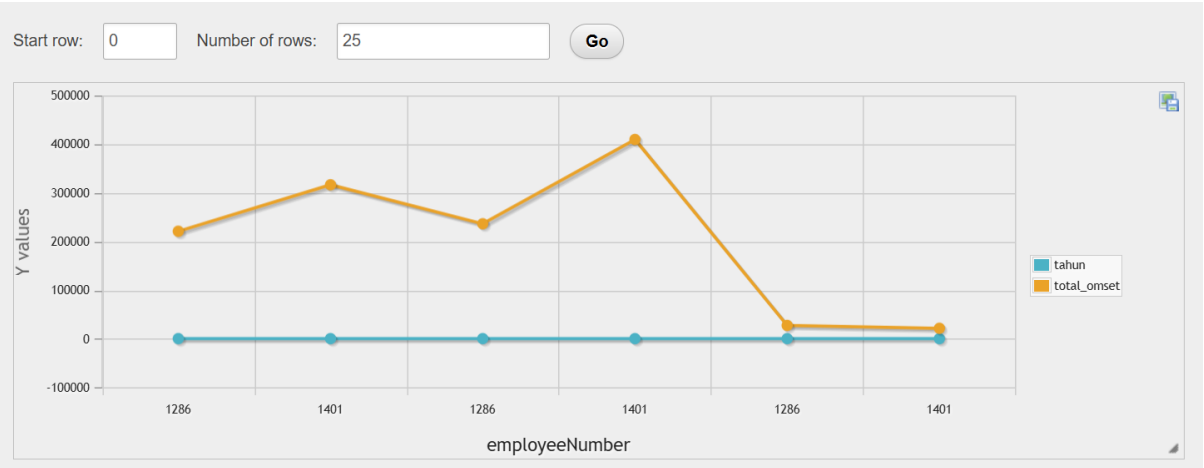
5. Buatlah report pertahun untuk KPI "Jumlah omset yang didapat" pada Foon Yue Tseng dan Pamela Castillo. Serta gambarkan grafiknya (grafik garis).
   Jawaban:
   Query:

```
SELECT e.employeeNumber, CONCAT(e.firstName, ' ', e.lastName) AS employee_name,
YEAR(p.paymentDate) AS tahun, SUM(p.amount) AS total_omset FROM employees e JOIN customers
c ON e.employeeNumber = c.salesRepEmployeeNumber JOIN payments p ON c.customerNumber =
p.customerNumber WHERE e.firstName = 'Foon Yue' AND e.lastName = 'Tseng' OR e.firstName =
'Pamela' AND e.lastName = 'Castillo' GROUP BY e.employeeNumber, YEAR(p.paymentDate) ORDER
BY tahun;
```

Output:

employeeNumber	employee_name	tahun ▲ 1	total_omset
1286	Foon Yue Tseng	2003	221887.03
1401	Pamela Castillo	2003	317104.78
1286	Foon Yue Tseng	2004	237255.26
1401	Pamela Castillo	2004	409910.07
1286	Foon Yue Tseng	2005	29070.38
1401	Pamela Castillo	2005	23187.02

Chart:



Analisis Data:

1. **Tahun 2003:**
   - Foon Yue Tseng memperoleh omset sebesar **221,887.03**
   - Pamela Castillo memperoleh omset lebih tinggi, yaitu **317,104.78**
2. **Tahun 2004:**
   - Foon Yue Tseng mengalami peningkatan omset menjadi **237,255.26**
   - Pamela Castillo juga meningkat signifikan menjadi **409,910.07**, menjadi yang tertinggi dalam tabel
3. **Tahun 2005:**
   - Omset Foon Yue Tseng menurun drastis ke **29,070.38**
   - Pamela Castillo juga mengalami penurunan lebih tajam ke **23,187.02**

Studi kasus

Pak Huhut merupakan pemegang saham LegendVehicle. dia membutuhkan dashboard untuk melihat perkembangan penjualan (omset) disetiap cabang di tiap tahunnya. Dikarenakan perusahaan tersebut belum merekrut Data Engineer maka, penarikan informasi hanya bisa dilakukan melaluai OLTP yang ada.
Hasil report yang diinginkan adalah grafik berdasarkan tabel berikut:

Nama Cabang	2003	2004	2005
...			
...			

Analisalah terlebih dahulu:

1. Field apa saja yang diperlukan untuk menampilkan penjualan di setiap cabang.
    - **`orders`** – berisi informasi transaksi dengan tanggal (`orderDate`).
    - **`orderdetails`** – berisi jumlah produk yang dibeli (`quantityOrdered`) dan harga per unit (`priceEach`).
    - **`customers`** – menghubungkan pesanan dengan sales representative (`salesRepEmployeeNumber`).
    - **`employees`** – menghubungkan sales representative dengan kantor (`officeCode`).
    - **`offices`** – berisi data kantor cabang (`officeCode`, `city`).
2. Bentuk query dengan memperhatikan relasi antar tabel.
   QUERY:

   Showing rows 0 - 20 (21 total, Query took 0.0054 seconds.)

```
SELECT o.city AS Nama_Cabang, YEAR(ord.orderDate) AS Tahun, SUM(od.quantityOrdered * od.priceEach) AS Total_Omset FROM orders ord JOIN orderdetails od ON ord.orderNumber = od.orderNumber JOIN customers c ON ord.customerNumber = c.customerNumber JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber JOIN offices o ON e.officeCode = o.officeCode GROUP BY o.city, YEAR(ord.orderDate) ORDER BY o.city, Tahun;
```

   OUTPUT:

Nama_Cabang	Tahun ▲ 2	Total_Omset
Boston	2003	301781.38
Boston	2004	467177.07
Boston	2005	123580.17
London	2003	549551.94
London	2004	706014.52
London	2005	181384.24
NYC	2003	391175.53
NYC	2004	665317.99
NYC	2005	101096.20
Paris	2003	969959.90
Paris	2004	1465229.84
Paris	2005	648571.84
San Francisco	2003	532681.13
San Francisco	2004	517408.62
San Francisco	2005	378973.82
Sydney	2003	304949.11
Sydney	2004	542996.02
Sydney	2005	299231.22
Tokyo	2003	267249.40
Tokyo	2004	151761.45
Tokyo	2005	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.
  Jawaban : **Tren Penjualan Produk Paling Laris Tiap Tahun**
  - Field yang digunakan

Tabel	Field	Keterangan
products	productName	Nama produk

orderdetails	productCode	ID produk yang dipesan
orderdetails	quantityOrdered	Jumlah unit yang dipesan
orders	orderDate	Tanggal pesanan
orders	orderNumber	ID pesanan

- Query:

Showing rows 0 - 24 (327 total, Query took 0.0068 seconds.)

```
SELECT p.productName, YEAR(ord.orderDate) AS Tahun, SUM(od.quantityOrdered) AS Total_Unit_Terjual FROM orders ord JOIN orderdetails od ON ord.orderNumber = od.orderNumber JOIN products p ON od.productCode = p.productCode GROUP BY p.productName, YEAR(ord.orderDate) ORDER BY Tahun, Total_Unit_Terjual DESC;
```

- Output:

productName	Tahun 1	Total_Unit_Terjual 2
1992 Ferrari 360 Spider red	2003	672
1936 Mercedes-Benz 500K Special Roadster	2003	429
1964 Mercedes Tour Bus	2003	427
1940s Ford truck	2003	408
1926 Ford Fire Engine	2003	393
1956 Porsche 356A Coupe	2003	389
1948 Porsche Type 356 Roadster	2003	382
1965 Aston Martin DB5	2003	382
1996 Peterbilt 379 Stake Bed with Outrigger	2003	373
0's Chicago Surface Lines Streetcar	2003	372
1968 Dodge Charger	2003	371
Diamond T620 Semi-Skirted Tanker	2003	369
1939 Cadillac Limousine	2003	369
1948 Porsche 356-A Roadster	2003	366
1995 Honda Civic	2003	366
1937 Lincoln Berline	2003	365
1954 Greyhound Scenicruiser	2003	363
1928 British Royal Navy Airplane	2003	363
1969 Ford Falcon	2003	363
1998 Chrysler Plymouth Prowler	2003	363
1917 Maxwell Touring Car	2003	360
1966 Shelby Cobra 427 S/C	2003	357
1999 Yamaha Speed Boat	2003	357
18th century schooner	2003	356
1932 Alfa Romeo 8C2300 Spider Sport	2003	354