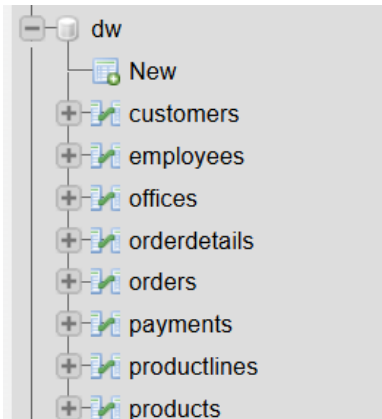


Tugas 1

1. Import data perusahaan tersebut pada DBMS MySQL!



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

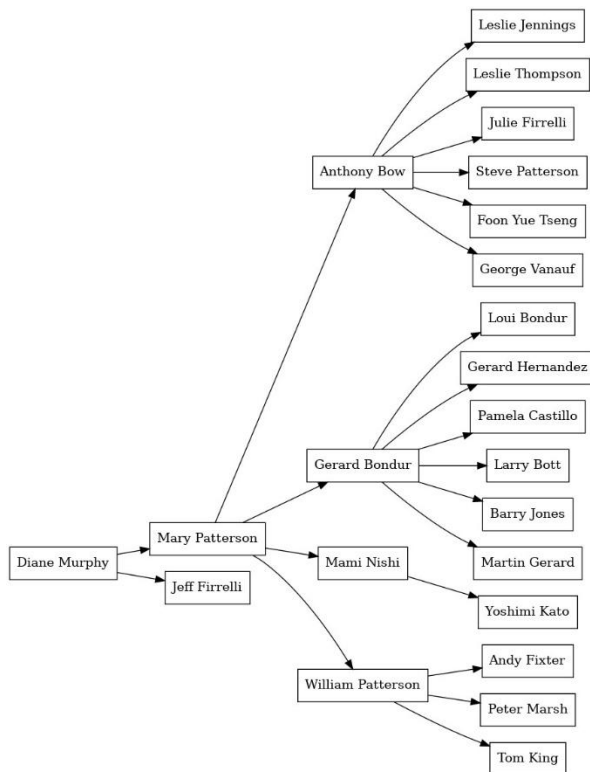
Tabel 1	Tabel 2	Jenis Relasi
productlines	products	One to many
products	orderdetails	One to many
orders	orderdetail	One to many
customers	orders	One to many
customers	payments	One to many
offices	employees	One to many
employees	employess	One to many
employees	customers	One to many

3. Analisa jumlah field pada setiap tabel!

Nama Tabel	Jumlah Field
productlines	4
product	9
orderdetails	5
orders	7
offices	9
employees	8
customers	13
payments	4

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 GROUP BY employee.employeeNumber, employee.firstName, employee.lastName 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 0.0071 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;
```

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

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

Extra options

id_employee	employee_name	total_cust_kpi
1002	Diane Murphy	100
1056	Mary Patterson	100
1102	Geord Bondu	46
1143	Anthony Bon	39
1088	William Patterson	10
1621	Mami Nahi	0
1076	Jeff Firrell	NULL
1185	Leslie Jennings	NULL
1166	Leslie Thompson	NULL
1180	Julie Firrell	NULL
1216	Steve Patterson	NULL
1286	Foon Yue Tseng	NULL
1323	George Vanauf	NULL
1337	Loui Bondu	NULL
1370	Gerard Hernandez	NULL
1401	Pamela Castillo	NULL
1501	Larry Bott	NULL
1504	Barry Jones	NULL
1611	Andy Fider	NULL
1612	Peter Marsh	NULL
1619	Tom King	NULL
1625	Yoshimi Kato	NULL
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 [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

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

Extra options

id_employee	employee_name	total_revenue
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 Bondu	569485.75
1611	Andy Fider	509385.82
1612	Peter Marsh	497907.16
1286	Foon Yue Tseng	486212.67
1621	Mami Nahi	457110.07
1216	Steve Patterson	449219.13
1702	Martin Gerard	387477.47
1180	Julie Firrell	386663.20
1166	Leslie Thompson	347533.93

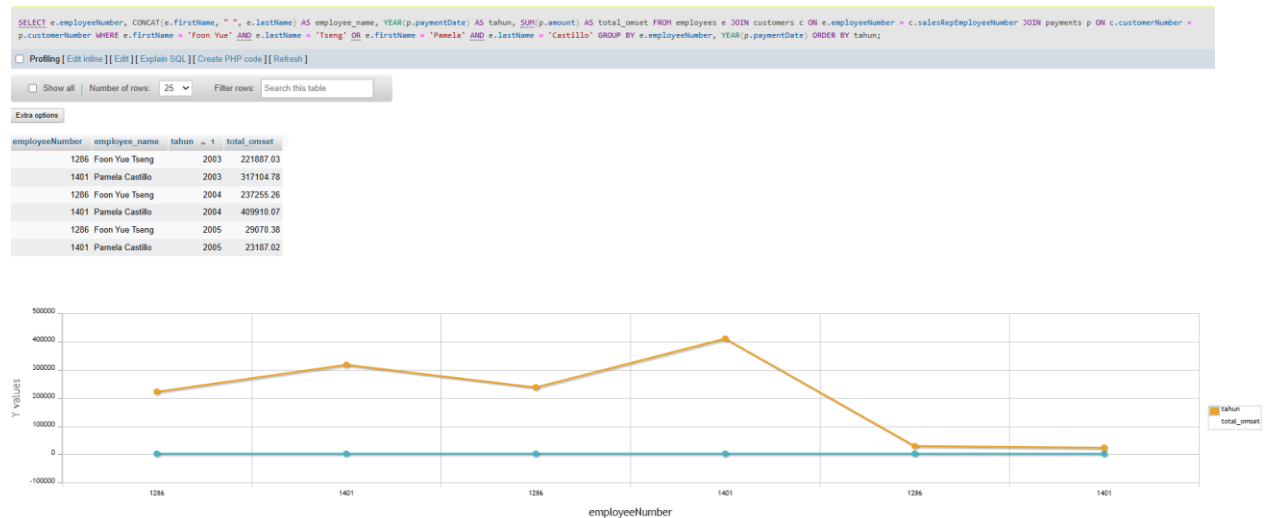
- Jika KPI yang pertama merupakan "Jumlah customer yang bertransaksi" sedangkan KPI yang kedua "Jumlah omset yang didapat". Maka, berapakah jumlah field yang dibutuhkan untuk mendapatkan informasi tersebut?

Jumlah customer yang bertransaksi: Membutuhkan satu field, yaitu Customer ID untuk menghitung jumlah customer yang melakukan transaksi.

Jumlah omset yang didapat: Membutuhkan satu field, yaitu Total Omset, yang menunjukkan total pendapatan dari transaksi.

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

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

Analisalah terlebih dahulu:

- Field apa saja yang diperlukan untuk menampilkan penjualan di setiap cabang
 - **offices**: Berisi informasi tentang cabang, seperti officeCode, city, country, dll.
 - **employees**: Menghubungkan cabang dengan karyawan (sales representative) melalui officeCode.
 - **customers**: Menghubungkan karyawan (sales representative) dengan pelanggan melalui salesRepEmployeeNumber.
 - **orders**: Berisi informasi pesanan pelanggan, seperti orderNumber, customerNumber, dll.
 - **orderdetails**: Berisi detail pesanan, seperti productCode, quantityOrdered, dan priceEach, yang digunakan untuk menghitung total penjualan.
 - **products**: Berisi informasi produk, seperti productLine

- Bentuk query dengan memperhatikan relasi antara table

Query:

```
SELECT o.city AS Nama_Cabang, SUM(CASE WHEN YEAR(ord.orderDate) = 2003 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2003', SUM(CASE WHEN YEAR(ord.orderDate) = 2004 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2004', SUM(CASE WHEN YEAR(ord.orderDate) = 2005 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2005'
FROM offices o JOIN employees e ON o.officeCode = e.officeCode JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber JOIN orders ord ON c.customerNumber = ord.customerNumber JOIN orderdetails od ON ord.orderNumber = od.orderNumber WHERE YEAR(ord.orderDate) IN (2003, 2004, 2005) GROUP BY o.city ORDER BY o.city;
```

Output:

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

1. buatlah report lain dengan sumber data OLTP yang sama, analisa field yang digunakan, bentuk struktur query dan tuliskan dalam tabel serta grafiknya.
 - Field yang digunakan:
 1. ProductLines:
 - productLine: Field utama untuk mengidentifikasi kategori produk (misalnya, Classic Cars, Motorcycles, Planes, dll.).
 - Ini akan menjadi dasar pengelompokan penjualan per kategori.
 2. Products:
 - productCode: Kunci utama untuk menghubungkan tabel products dengan orderdetails.
 - productLine: Foreign key yang menghubungkan ke tabel productlines untuk mendapatkan kategori produk.
 3. OrderDetails:
 - orderNumber: Untuk menghubungkan dengan tabel orders.
 - productCode: Untuk menghubungkan dengan tabel products.
 - quantityOrdered: Jumlah barang yang dipesan.
 - priceEach: Harga per unit, digunakan untuk menghitung total penjualan ($\text{quantityOrdered} * \text{priceEach}$).
 4. Orders:
 - orderNumber: Kunci utama untuk menghubungkan dengan orderdetails.
 - orderDate: Digunakan untuk mengekstrak tahun (2003, 2004, 2005) dengan fungsi YEAR(orderDate).
 - customerNumber: Untuk menghubungkan dengan tabel customers (opsional, jika diperlukan informasi pelanggan).

- Struktur Query

```
SELECT pl.productline AS Kategori_Produk, SUM(CASE WHEN YEAR(ord.orderDate) = 2003 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2003', SUM(CASE WHEN YEAR(ord.orderDate) = 2004 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2004', SUM(CASE WHEN YEAR(ord.orderDate) = 2005 THEN od.quantityOrdered * od.priceEach ELSE 0 END) AS '2005' FROM productlines pl JOIN products p ON pl.productline = p.productline JOIN orderdetails od ON p.productCode = od.productCode JOIN orders ord ON od.orderNumber = ord.orderNumber WHERE YEAR(ord.orderDate) IN (2003, 2004, 2005) GROUP BY pl.productline ORDER BY pl.productline;
```

Kategori_Produk	2003	2004	2005
Classic Cars	1374832.22	1763136.73	715953.54
Motorcycles	348909.24	527243.84	245273.04
Planes	309784.20	471971.46	172881.88
Ships	222182.08	337326.10	104490.16
Trains	65822.05	96285.53	26425.34
Trucks and Buses	376657.12	465390.00	182066.45
Vintage Cars	619161.48	854551.85	323846.30

