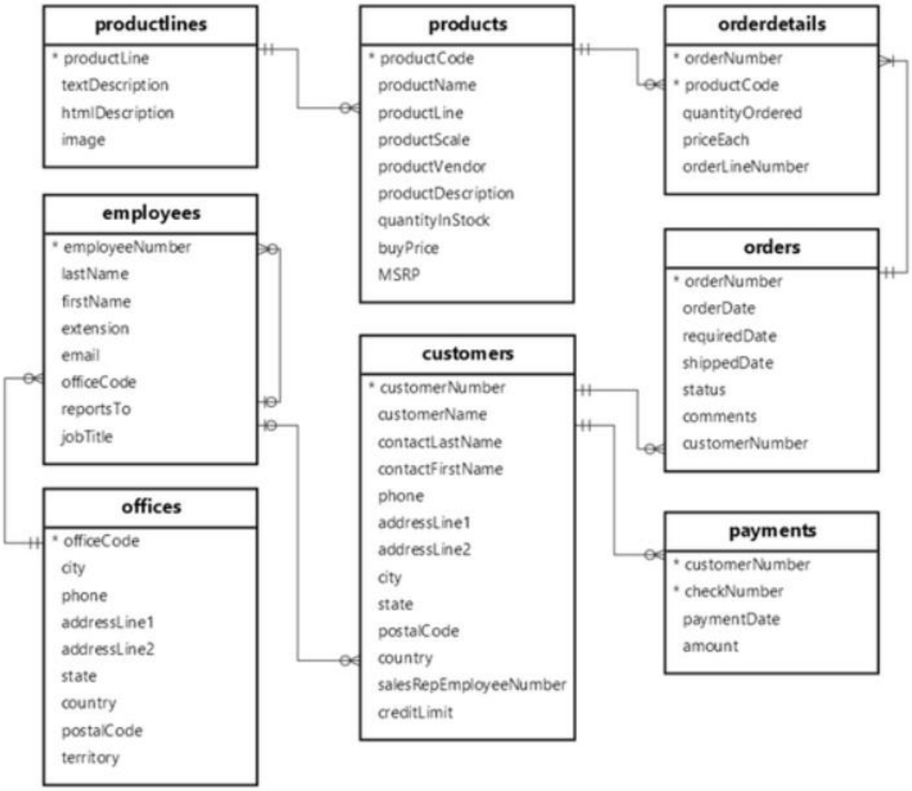
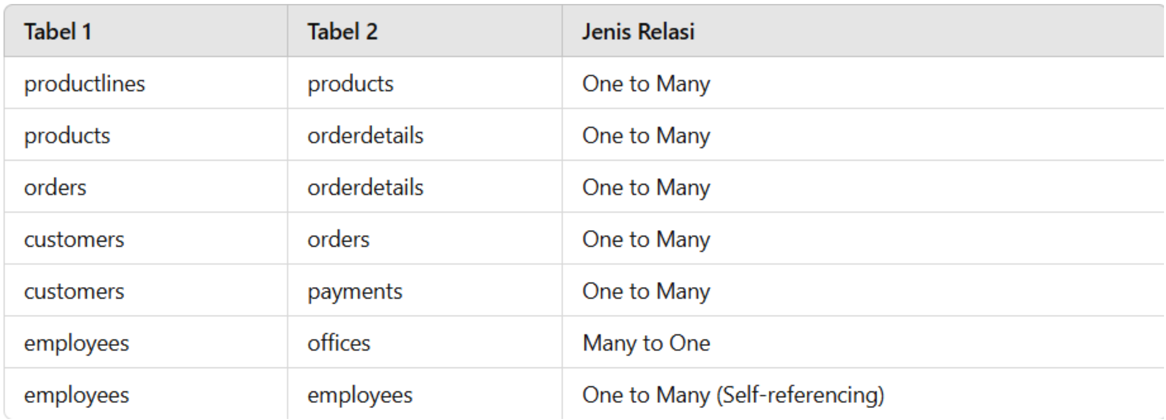
Jobsheet 2: Database Operasional



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!



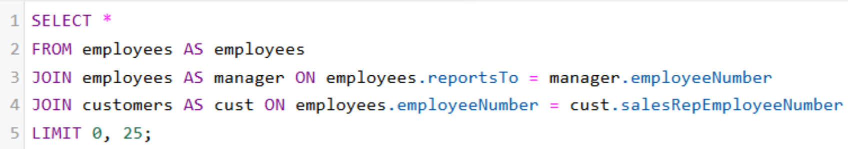
Tabel 1	Tabel 2	Jenis Relasi
productlines	products	One to Many
products	orderdetails	One to Many
orders	orderdetails	One to Many
customers	orders	One to Many
customers	payments	One to Many
employees	offices	Many to One
employees	employees	One to Many (Self-referencing)

3. Analisa jumlah field pada setiap tabel!

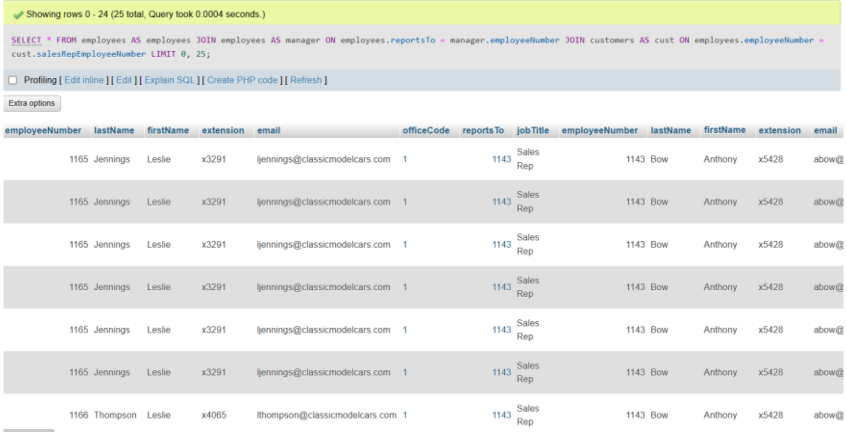
Nama Tabel	Jumlah Field
productlines	4
products	9
orderdetails	5
orders	6
customers	10
payments	4
employees	7
offices	7

PRAKTIKUM 1

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



```sql
SELECT *
FROM employees AS employees
JOIN employees AS manager ON employees.reportsTo = manager.employeeNumber
JOIN customers AS cust ON employees.employeeNumber = cust.salesRepEmployeeNumber
LIMIT 0, 25;
```



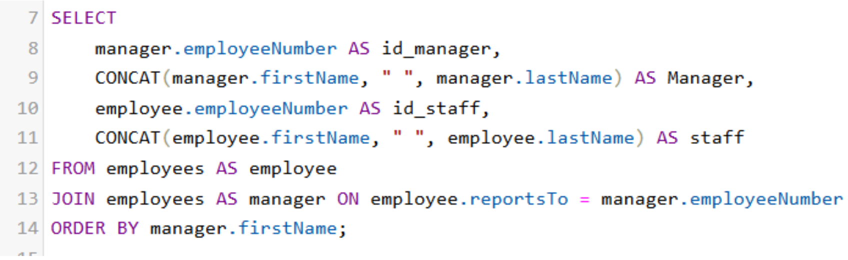
✓ Showing rows 0 - 24 (25 total, Query took 0.0004 seconds.)

```sql
SELECT * FROM employees AS employees JOIN employees AS manager ON employees.reportsTo = manager.employeeNumber JOIN customers AS cust ON employees.employeeNumber =
cust.salesRepEmployeeNumber LIMIT 0, 25;
```

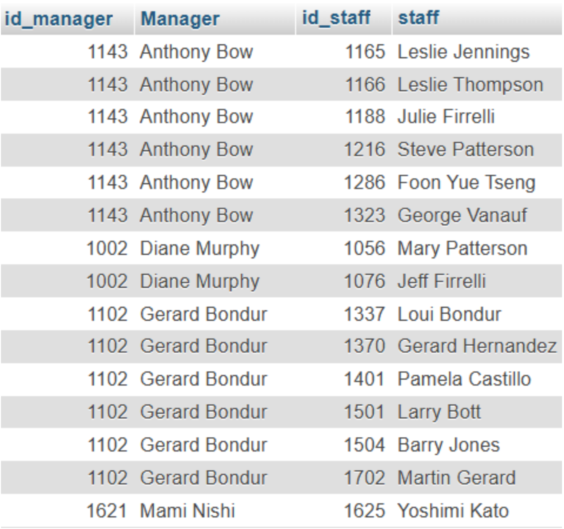
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options

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

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



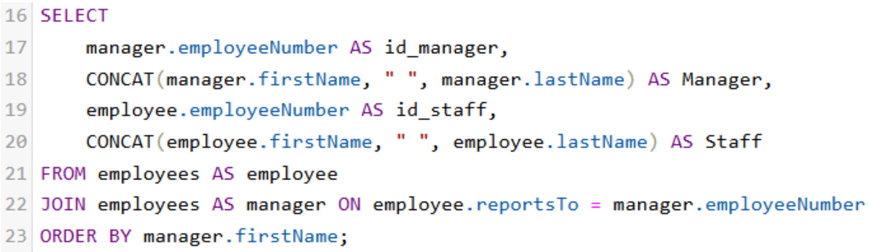
```sql
SELECT
    manager.employeeNumber AS id_manager,
    CONCAT(manager.firstName, " ", manager.lastName) AS Manager,
    employee.employeeNumber AS id_staff,
    CONCAT(employee.firstName, " ", employee.lastName) AS staff
FROM employees AS employee
JOIN employees AS manager ON employee.reportsTo = manager.employeeNumber
ORDER BY manager.firstName;
```



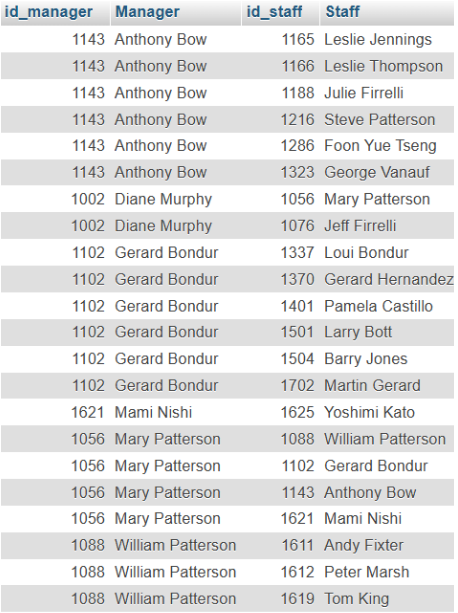
id_manager	Manager	id_staff	staff
1143	Anthony Bow	1165	Leslie Jennings
1143	Anthony Bow	1166	Leslie Thompson
1143	Anthony Bow	1188	Julie Firrelli
1143	Anthony Bow	1216	Steve Patterson
1143	Anthony Bow	1286	Foon Yue Tseng
1143	Anthony Bow	1323	George Vanauf
1002	Diane Murphy	1056	Mary Patterson
1002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

TUGAS 2

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

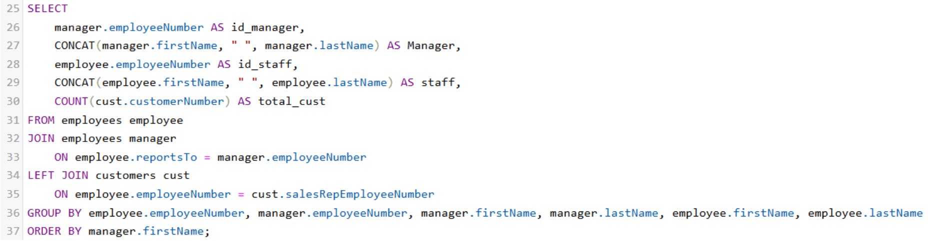


```sql
SELECT
    manager.employeeNumber AS id_manager,
    CONCAT(manager.firstName, " ", manager.lastName) AS Manager,
    employee.employeeNumber AS id_staff,
    CONCAT(employee.firstName, " ", employee.lastName) AS Staff
FROM employees AS employee
JOIN employees AS manager ON employee.reportsTo = manager.employeeNumber
ORDER BY manager.firstName;
```

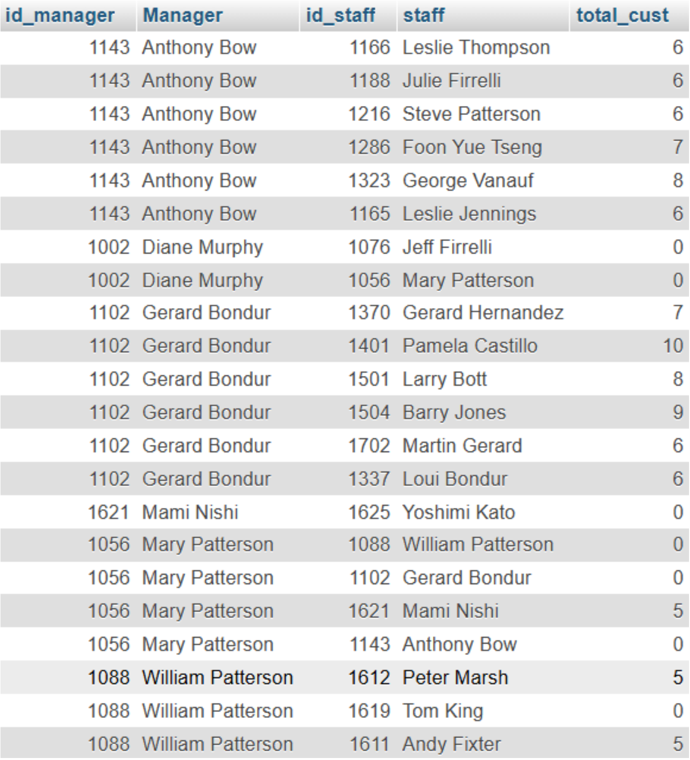


id_manager	Manager	id_staff	Staff
1143	Anthony Bow	1165	Leslie Jennings
1143	Anthony Bow	1166	Leslie Thompson
1143	Anthony Bow	1188	Julie Firrelli
1143	Anthony Bow	1216	Steve Patterson
1143	Anthony Bow	1286	Foon Yue Tseng
1143	Anthony Bow	1323	George Vanauf
1002	Diane Murphy	1056	Mary Patterson
1002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

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



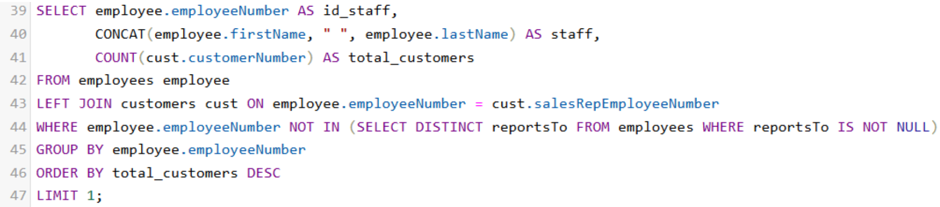
```sql
SELECT
    manager.employeeNumber AS id_manager,
    CONCAT(manager.firstName, " ", manager.lastName) AS Manager,
    employee.employeeNumber AS id_staff,
    CONCAT(employee.firstName, " ", employee.lastName) AS staff,
    COUNT(cust.customerNumber) AS total_cust
FROM employees employee
JOIN 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;
```



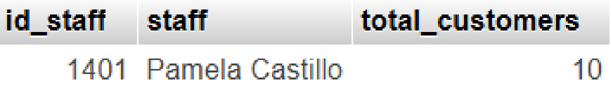
id_manager	Manager	id_staff	staff	total_cust
1143	Anthony Bow	1166	Leslie Thompson	6
1143	Anthony Bow	1188	Julie Firrelli	6
1143	Anthony Bow	1216	Steve Patterson	6
1143	Anthony Bow	1286	Foon Yue Tseng	7
1143	Anthony Bow	1323	George Vanauf	8
1143	Anthony Bow	1165	Leslie Jennings	6
1002	Diane Murphy	1076	Jeff Firrelli	0
1002	Diane Murphy	1056	Mary Patterson	0
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
1102	Gerard Bondur	1337	Loui Bondur	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	1621	Mami Nishi	5
1056	Mary Patterson	1143	Anthony Bow	0
1088	William Patterson	1612	Peter Marsh	5
1088	William Patterson	1619	Tom King	0
1088	William Patterson	1611	Andy Fixter	5

TUGAS 3

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

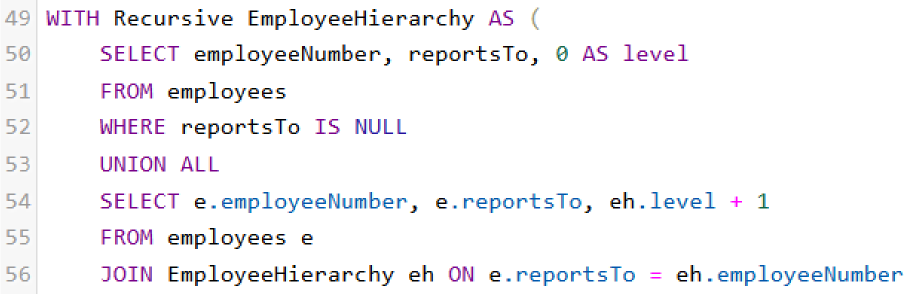


```sql
SELECT employee.employeeNumber AS id_staff,
       CONCAT(employee.firstName, " ", employee.lastName) AS staff,
       COUNT(cust.customerNumber) AS total_customers
FROM employees employee
LEFT JOIN customers cust ON employee.employeeNumber = cust.salesRepEmployeeNumber
WHERE employee.employeeNumber NOT IN (SELECT DISTINCT reportsTo FROM employees WHERE reportsTo IS NOT NULL)
GROUP BY employee.employeeNumber
ORDER BY total_customers DESC
LIMIT 1;
```

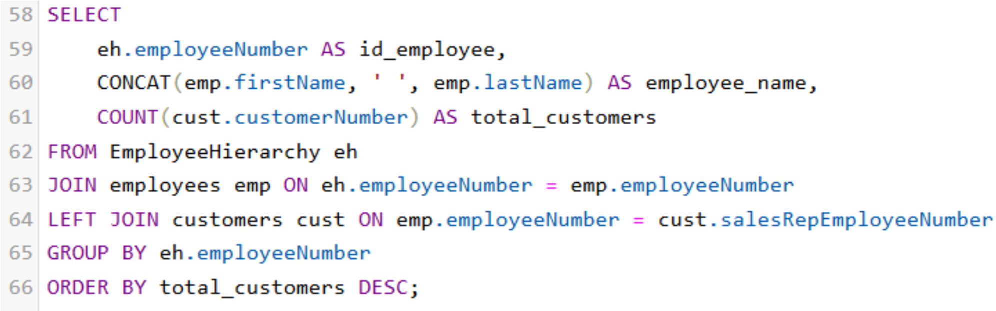


id_staff	staff	total_customers
1401	Pamela Castillo	10

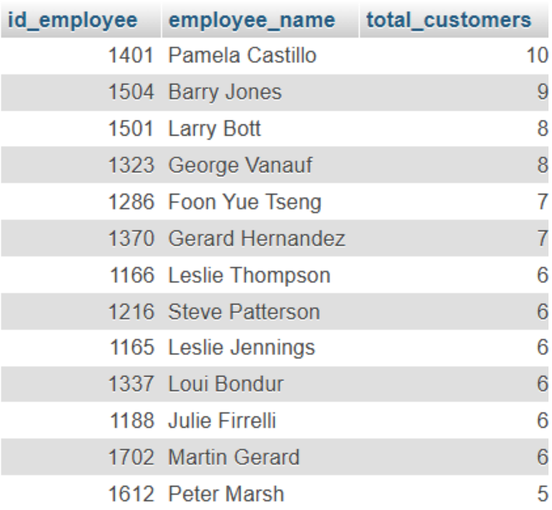
2. Jika KPI atasan dihitung dari customer yang dimilikinya dijumlah dengan customer dari staff dibawahnya, urutkan ranking prestasi keseluruhan pegawai beserta keterangan jumlah customer yang dimilikinya!



```sql
WITH Recursive EmployeeHierarchy AS (
    SELECT employeeNumber, reportsTo, 0 AS level
    FROM employees
    WHERE reportsTo IS NULL
    UNION ALL
    SELECT e.employeeNumber, e.reportsTo, eh.level + 1
    FROM employees e
    JOIN EmployeeHierarchy eh ON e.reportsTo = eh.employeeNumber
```

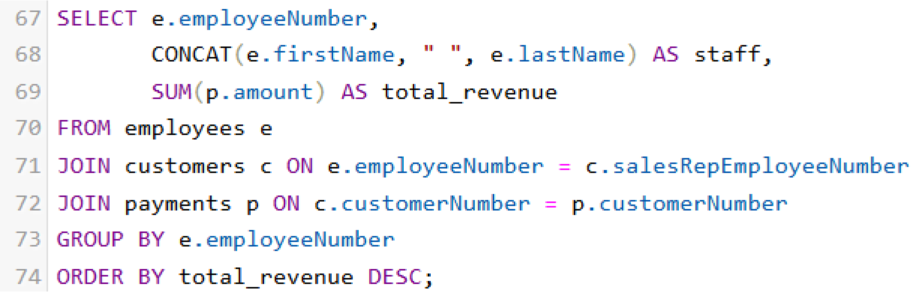


```sql
SELECT
    eh.employeeNumber AS id_employee,
    CONCAT(emp.firstName, ' ', emp.lastName) AS employee_name,
    COUNT(cust.customerNumber) AS total_customers
FROM EmployeeHierarchy eh
JOIN employees emp ON eh.employeeNumber = emp.employeeNumber
LEFT JOIN customers cust ON emp.employeeNumber = cust.salesRepEmployeeNumber
GROUP BY eh.employeeNumber
ORDER BY total_customers DESC;
```

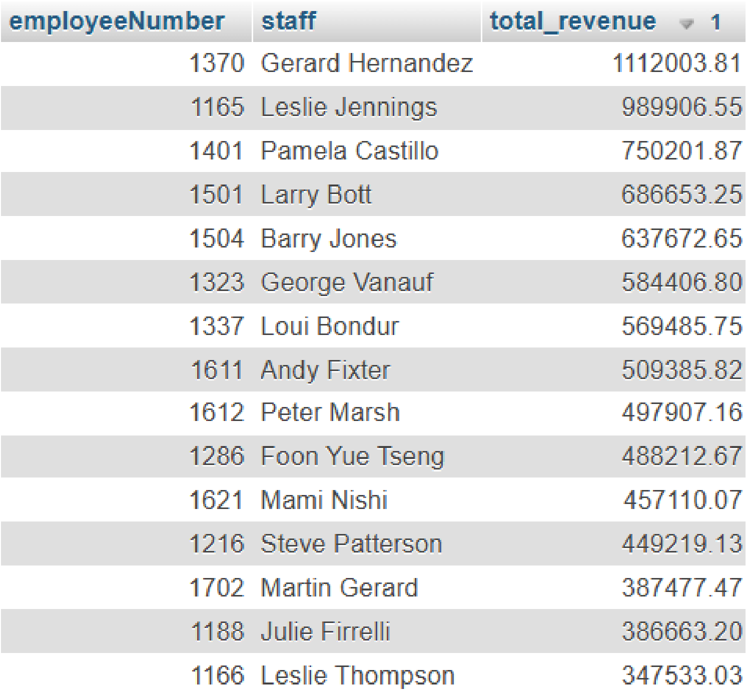


id_employee	employee_name	total_customers
1401	Pamela Castillo	10
1504	Barry Jones	9
1501	Larry Bott	8
1323	George Vanauf	8
1286	Foon Yue Tseng	7
1370	Gerard Hernandez	7
1166	Leslie Thompson	6
1216	Steve Patterson	6
1165	Leslie Jennings	6
1337	Loui Bondur	6
1188	Julie Firrelli	6
1702	Martin Gerard	6
1612	Peter Marsh	5

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

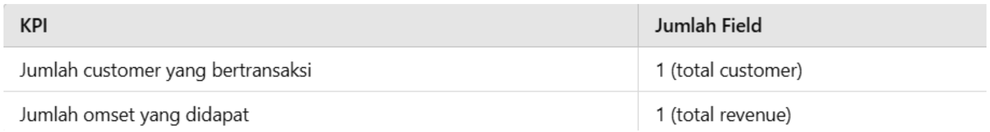


```sql
SELECT e.employeeNumber,
       CONCAT(e.firstName, " ", e.lastName) AS staff,
       SUM(p.amount) AS total_revenue
FROM employees e
JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN payments p ON c.customerNumber = p.customerNumber
GROUP BY e.employeeNumber
ORDER BY total_revenue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



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

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)

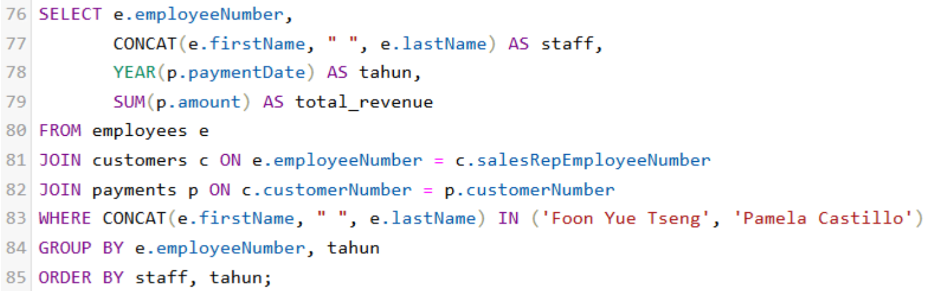
Jumlah customer yang bertransaksi

- Mengukur jumlah customer yang melakukan transaksi melalui seorang pegawai (staff).
- Field yang dibutuhkan: total customer (jumlah customer yang terkait dengan pegawai).

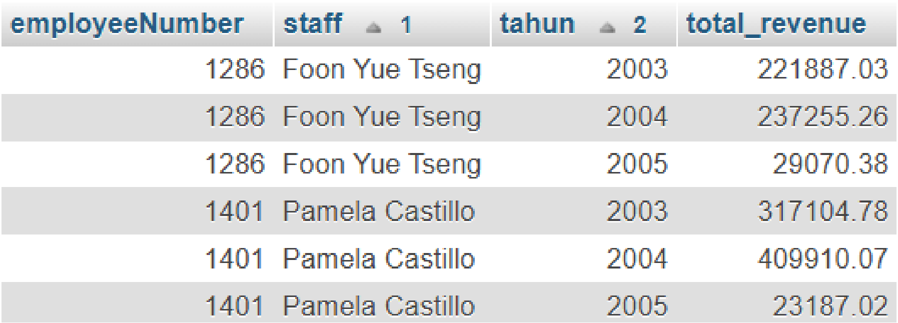
Jumlah omset yang didapat

- Mengukur total pendapatan (revenue) yang diperoleh dari transaksi customer yang dilayani oleh seorang pegawai.
- Field yang dibutuhkan: total revenue (jumlah uang yang diperoleh dari customer yang bertransaksi).

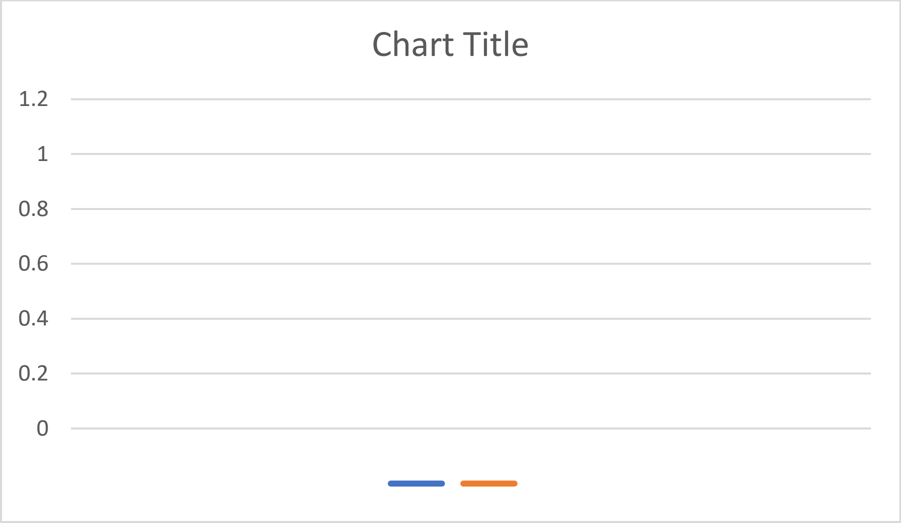
5. Buatlah report pertahun untuk KPI "Jumlah omset yang didapat" pada Foon Yue Tseng dan Pamela Castillo. Serta gambarkan grafiknya (grafik garis).



```
SELECT e.employeeNumber,
       CONCAT(e.firstName, " ", e.lastName) AS staff,
       YEAR(p.paymentDate) AS tahun,
       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
WHERE CONCAT(e.firstName, " ", e.lastName) IN ('Foon Yue Tseng', 'Pamela Castillo')
GROUP BY e.employeeNumber, tahun
ORDER BY staff, tahun;
```

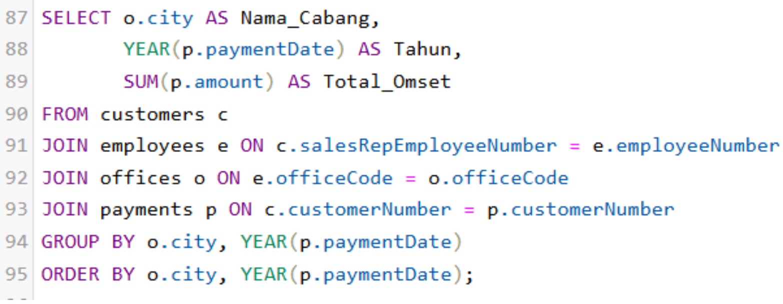


employeeNumber	staff ▲ 1	tahun ▲ 2	total_revenue
1286	Foon Yue Tseng	2003	221887.03
1286	Foon Yue Tseng	2004	237255.26
1286	Foon Yue Tseng	2005	29070.38
1401	Pamela Castillo	2003	317104.78
1401	Pamela Castillo	2004	409910.07
1401	Pamela Castillo	2005	23187.02

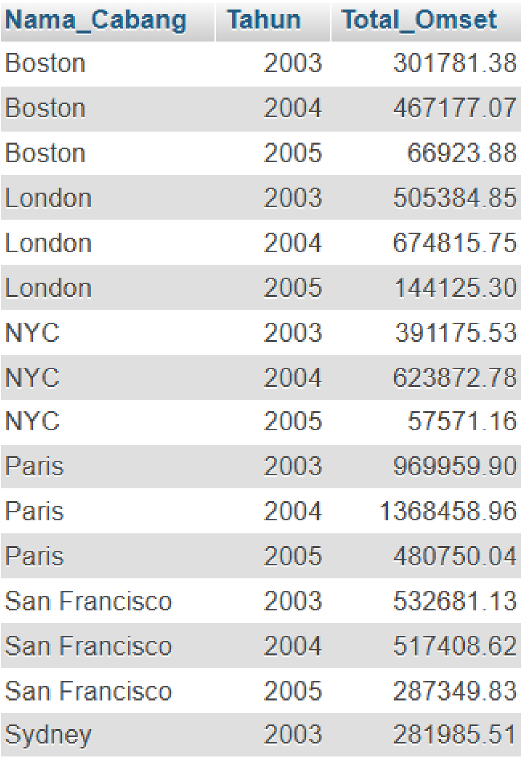


Studi Kasus

1. Field apa saja yang diperlukan untuk menampilkan penjualan di setiap cabang.
   - BranchName (Nama Cabang)
   - Year (Tahun transaksi)
   - TotalSales (Total penjualan/omset)
2. Bentuk query dengan memperhatikan relasi antar tabel.



```sql
SELECT o.city AS Nama_Cabang,
       YEAR(p.paymentDate) AS Tahun,
       SUM(p.amount) AS Total_Omset
FROM customers c
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN offices o ON e.officeCode = o.officeCode
JOIN payments p ON c.customerNumber = p.customerNumber
GROUP BY o.city, YEAR(p.paymentDate)
ORDER BY o.city, YEAR(p.paymentDate);
```

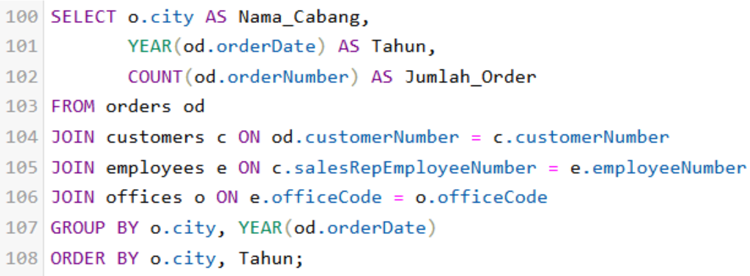


Nama_Cabang	Tahun	Total_Omset
Boston	2003	301781.38
Boston	2004	467177.07
Boston	2005	66923.88
London	2003	505384.85
London	2004	674815.75
London	2005	144125.30
NYC	2003	391175.53
NYC	2004	623872.78
NYC	2005	57571.16
Paris	2003	969959.90
Paris	2004	1368458.96
Paris	2005	480750.04
San Francisco	2003	532681.13
San Francisco	2004	517408.62
San Francisco	2005	287349.83
Sydney	2003	281985.51

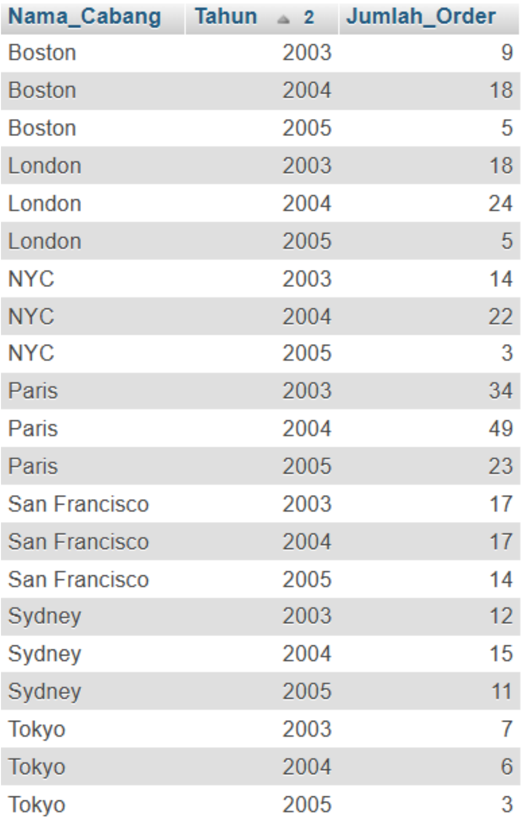
SOAL BONUS:
buatlah report lain dengan sumber data OLTP yang sama, analisa field yang digunakan, bentuk struktur query dan tuliskan dalam tabel serta grafiknya.

Field yang dibutuhkan

- orderNumber untuk menghitung jumlah order
- orderDate untuk mengambil tahun order
- officeCode untuk menghubungkan dengan cabang
- city sebagai nama cabang (dari tabel offices)



```
SELECT o.city AS Nama_Cabang,
       YEAR(od.orderDate) AS Tahun,
       COUNT(od.orderNumber) AS Jumlah_Order
FROM orders od
JOIN customers c ON od.customerNumber = c.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN offices o ON e.officeCode = o.officeCode
GROUP BY o.city, YEAR(od.orderDate)
ORDER BY o.city, Tahun;
```



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

Grafik

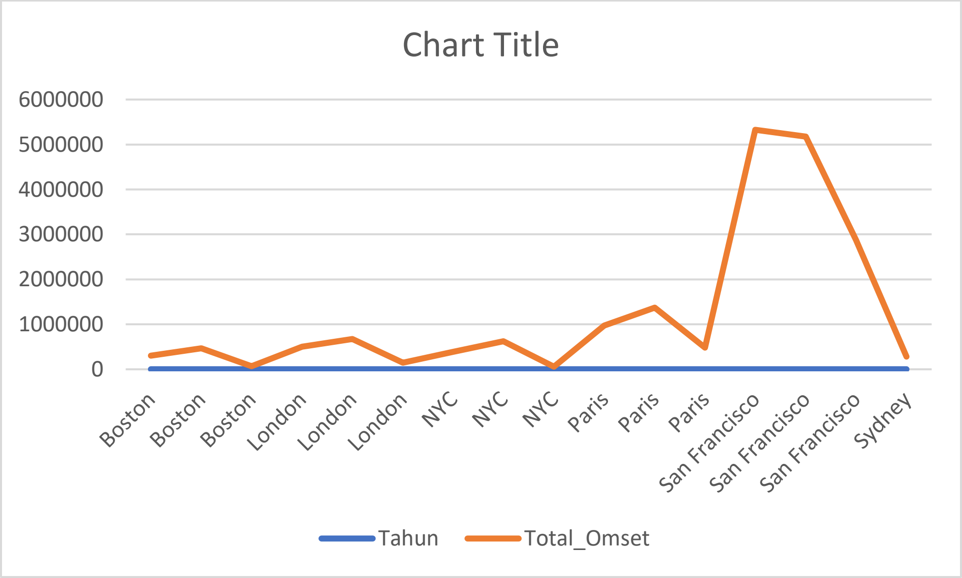


Chart Title
Boston
Boston
Boston
London
London
London
NYC
NYC
NYC
Paris
Paris
Paris
San Francisco
San Francisco
San Francisco
Sydney
Tahun
Total_Omset