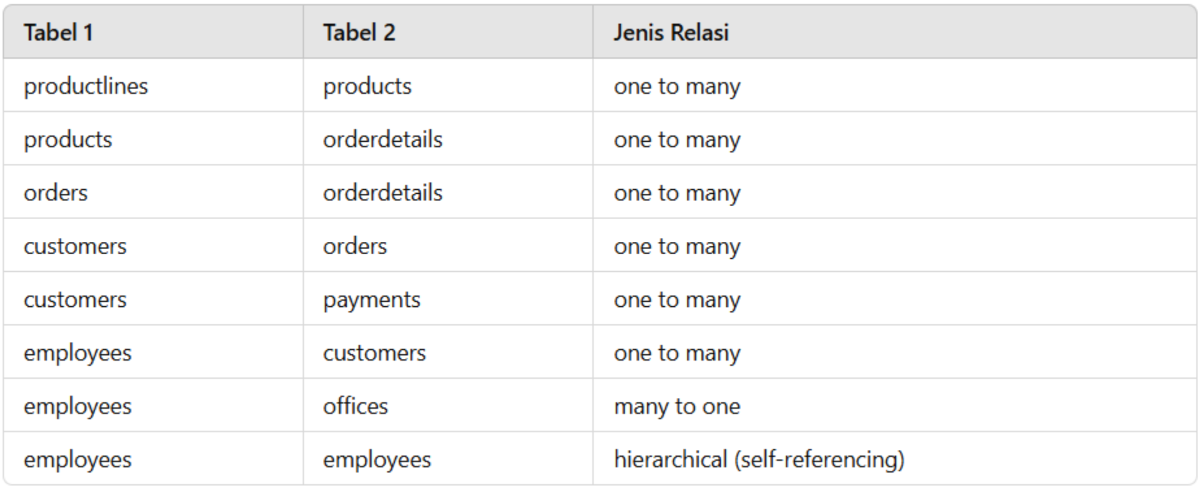
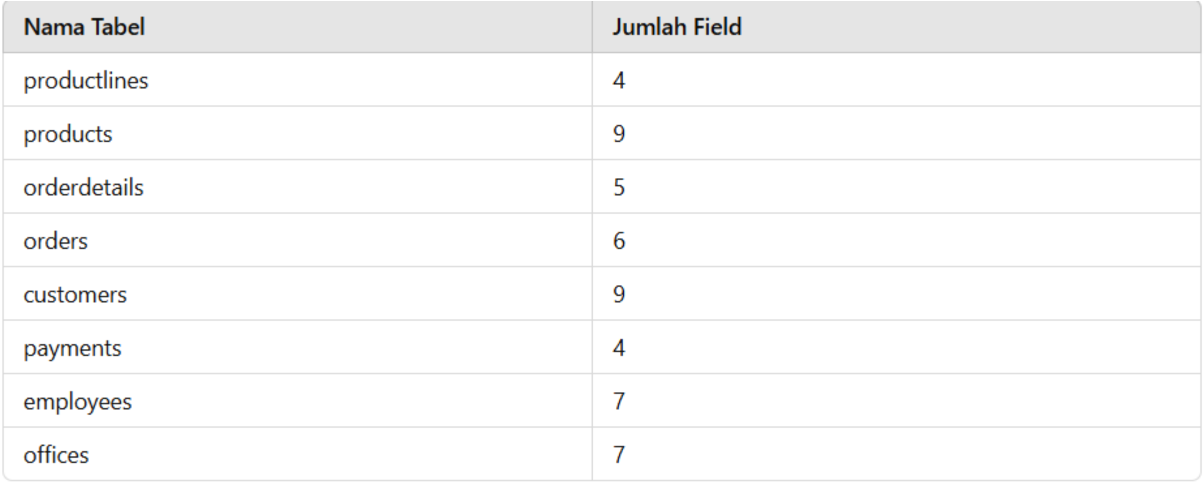
Laporan Praktikum Data Warehouse
Jobsheet 2

1. Analisis Struktur Data dan Hubungan Antar Tabel



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	customers	one to many
employees	offices	many to one
employees	employees	hierarchical (self-referencing)

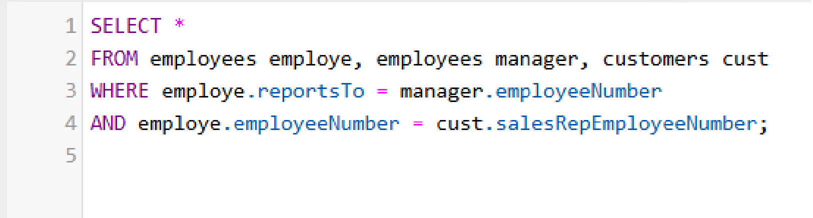
2. Analisis Jumlah Field dalam setiap Tabel



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

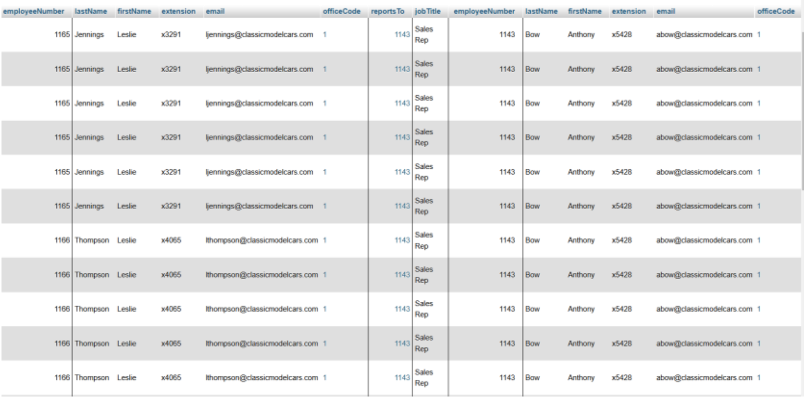
Praktikum 1

1. Query :

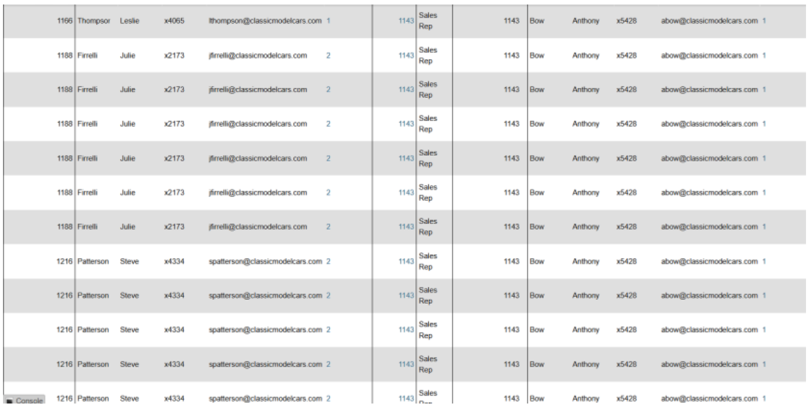


```sql
SELECT *
FROM employees employe, employees manager, customers cust
WHERE employe.reportsTo = manager.employeeNumber
AND employe.employeeNumber = cust.salesRepEmployeeNumber;
```

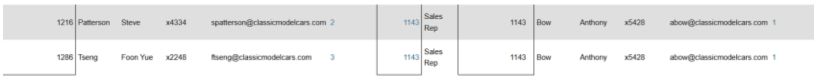
Output :



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

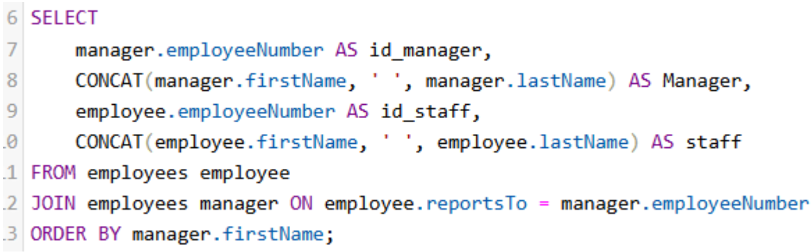


1166	Thompsor	Leslie	x4065	lthompson@classicmodelcars.com	1	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1



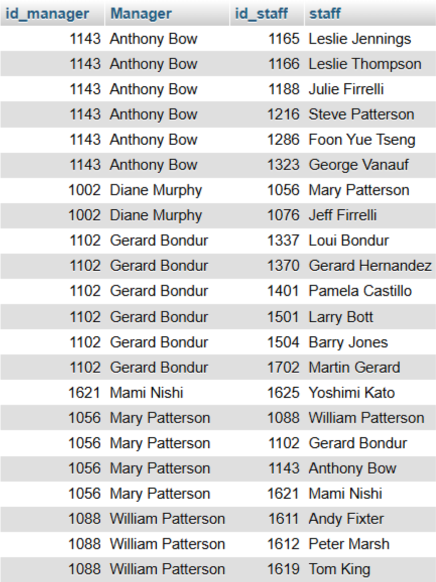
1216	Patterson	Steve	x4334	spatterson@classicmodelcars.com	2	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1
1286	Tseng	Foon Yue	x2248	ftseng@classicmodelcars.com	3	1143	Sales Rep	1143	Bow	Anthony	x5428	abow@classicmodelcars.com	1

2. Query:



```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 employee
JOIN employees manager ON employee.reportsTo = manager.employeeNumber
ORDER BY manager.firstName;
```

Output :

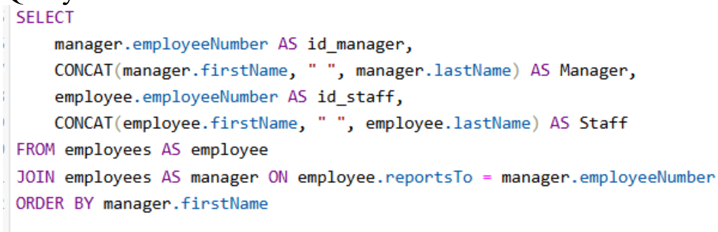


id_manager	Manager	id_staff	staff
1143	Anthony Bow	1165	Leslie Jennings
1143	Anthony Bow	1166	Leslie Thompson
1143	Anthony Bow	1188	Julie Firrelli
1143	Anthony Bow	1216	Steve Patterson
1143	Anthony Bow	1286	Foon Yue Tseng
1143	Anthony Bow	1323	George Vanauf
1002	Diane Murphy	1056	Mary Patterson
1002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

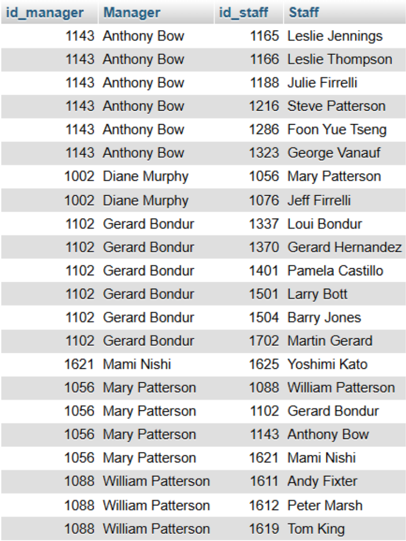
1. Gambarlah hirarki organisasi berdasarkan atasan dari setiap pegawai sesuai dengan hasil prkatikum diatas!

Query:



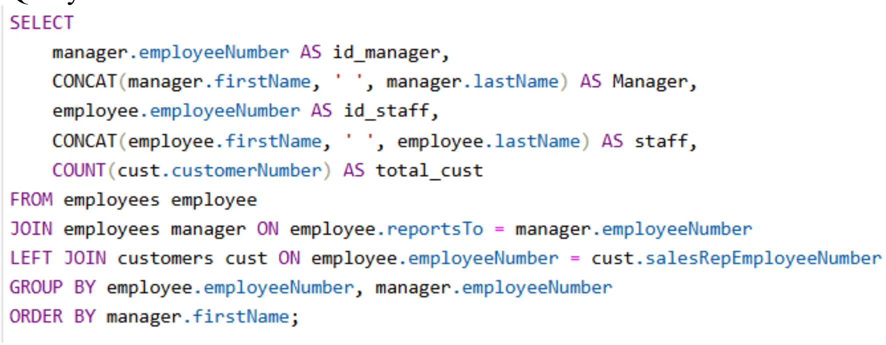
```
SELECT
    manager.employeeNumber AS id_manager,
    CONCAT(manager.firstName, " ", manager.lastName) AS Manager,
    employee.employeeNumber AS id_staff,
    CONCAT(employee.firstName, " ", employee.lastName) AS Staff
FROM employees AS employee
JOIN employees AS manager ON employee.reportsTo = manager.employeeNumber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

Output :



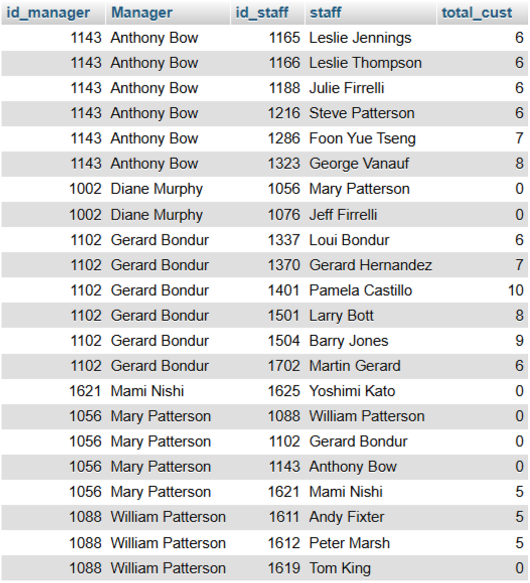
id_manager	Manager	id_staff	Staff
1143	Anthony Bow	1165	Leslie Jennings
1143	Anthony Bow	1166	Leslie Thompson
1143	Anthony Bow	1188	Julie Firrelli
1143	Anthony Bow	1216	Steve Patterson
1143	Anthony Bow	1286	Foon Yue Tseng
1143	Anthony Bow	1323	George Vanauf
1002	Diane Murphy	1056	Mary Patterson
1002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



```
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
ORDER BY manager.firstName;
```

Output :

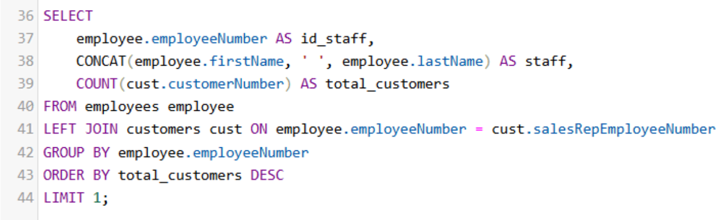


id_manager	Manager	id_staff	staff	total_cust
1143	Anthony Bow	1165	Leslie Jennings	6
1143	Anthony Bow	1166	Leslie Thompson	6
1143	Anthony Bow	1188	Julie Firrelli	6
1143	Anthony Bow	1216	Steve Patterson	6
1143	Anthony Bow	1286	Foon Yue Tseng	7
1143	Anthony Bow	1323	George Vanauf	8
1002	Diane Murphy	1056	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

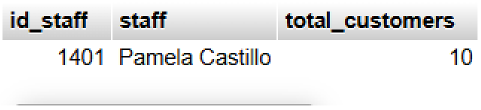
1. Siapakah staff dengan hirarki paling bawah yang berprestasi dilihat dari jumlah customer terbanyak?

Query :



```
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
GROUP BY employee.employeeNumber
ORDER BY total_customers DESC
LIMIT 1;
```

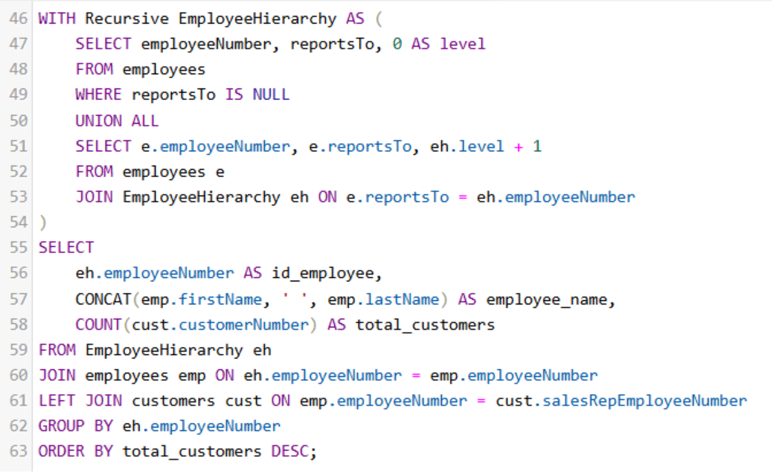
Output :



id_staff	staff	total_customers
1401	Pamela Castillo	10

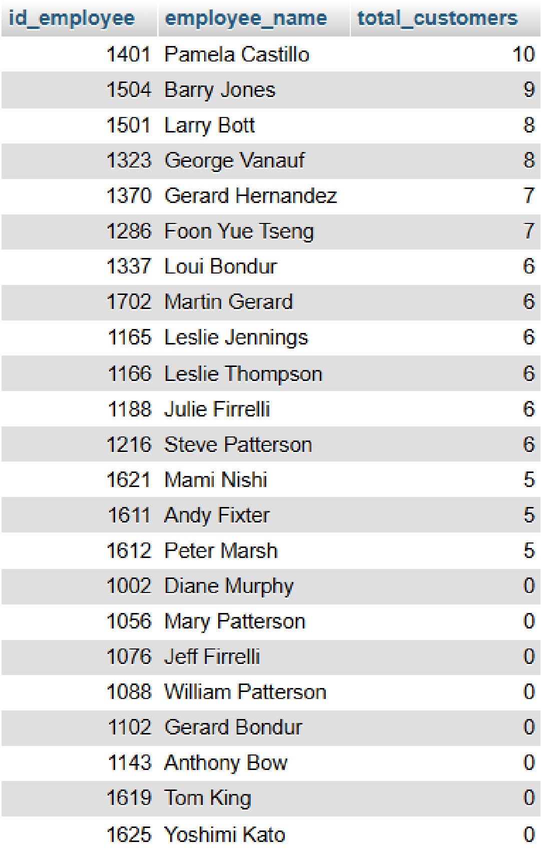
2. Jika KPI atasan dihitung dari customer yang dimilikinya dijumlah dengan customer dari staff dibawahnya, urutkan ranking prestasi keseluruhan pegawai beserta keterangan jumlah customer yang dimilikinya!

Query:



```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
)
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;
```

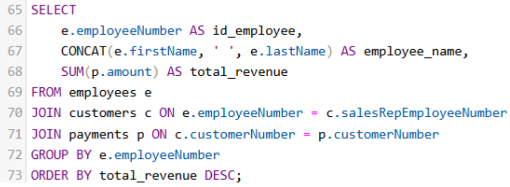
Output:



id_employee	employee_name	total_customers
1401	Pamela Castillo	10
1504	Barry Jones	9
1501	Larry Bott	8
1323	George Vanauf	8
1370	Gerard Hernandez	7
1286	Foon Yue Tseng	7
1337	Loui Bondur	6
1702	Martin Gerard	6
1165	Leslie Jennings	6
1166	Leslie Thompson	6
1188	Julie Firrelli	6
1216	Steve Patterson	6
1621	Mami Nishi	5
1611	Andy Fixter	5
1612	Peter Marsh	5
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

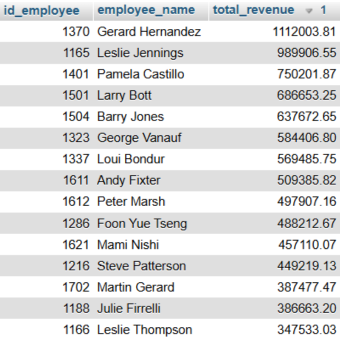
3. Analisa kembali data LegendVehicle untuk mendapatkan ranking pegawai berdasarkan KPI **"Jumlah omset yang didapat".** Urutkan ranking pegawai beserta keterangan dana yang didapat!

   Query:



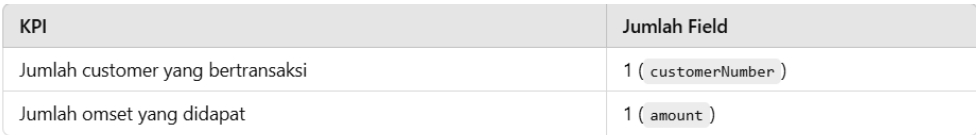
```
SELECT
    e.employeeNumber AS id_employee,
    CONCAT(e.firstName, ' ', e.lastName) AS employee_name,
    SUM(p.amount) AS total_revenue
FROM employees e
JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN payments p ON c.customerNumber = p.customerNumber
GROUP BY e.employeeNumber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

   Output :



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?

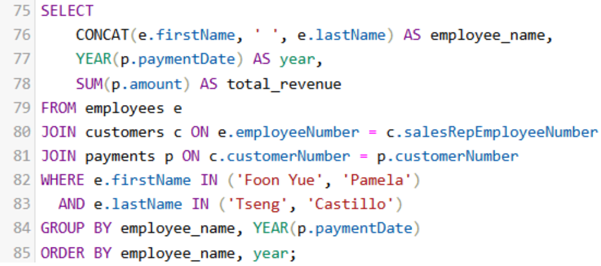


KPI	Jumlah Field
Jumlah customer yang bertransaksi	1 (`customerNumber`)
Jumlah omset yang didapat	1 (`amount`)

   Penjelasan :
   Jumlah customer yang bertransaksi dapat diperoleh dari jumlah unik customerNumber yang telah melakukan transaksi.
   Jumlah omset yang didapat dihitung dari total amount yang ada dalam tabel payments.

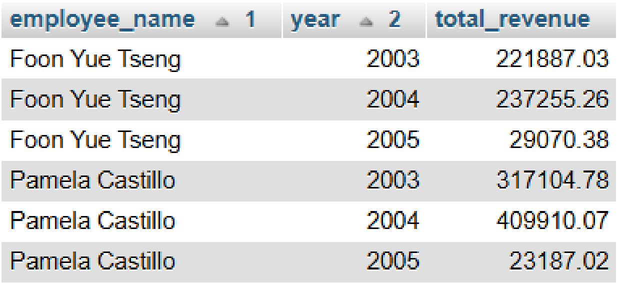
5. Buatlah report pertahun untuk KPI **"Jumlah omset yang didapat"** pada **Foon Yue Tseng** dan **Pamela Castillo**. Serta gambarkan grafiknya (grafik garis).

   Query :



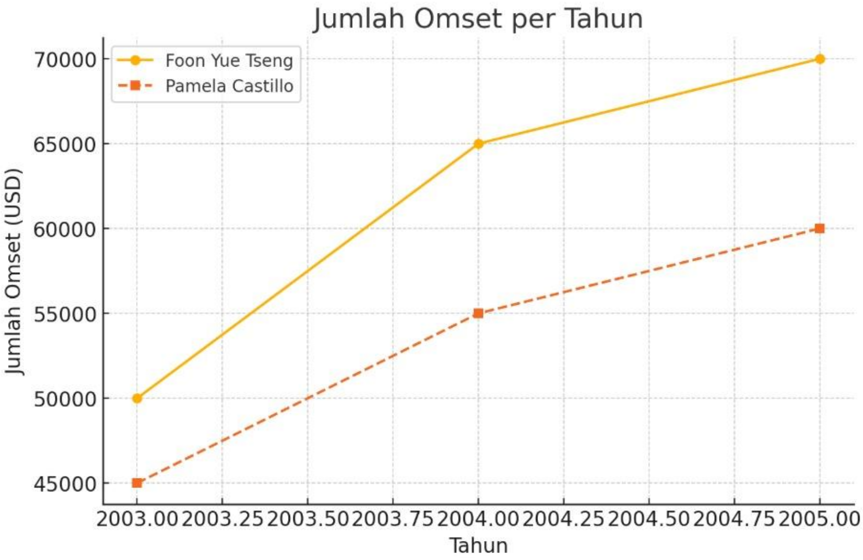
```
SELECT
    CONCAT(e.firstName, ' ', e.lastName) AS employee_name,
    YEAR(p.paymentDate) AS year,
    SUM(p.amount) AS total_revenue
FROM employees e
JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN payments p ON c.customerNumber = p.customerNumber
WHERE e.firstName IN ('Foon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

Output :



employee_name ▲ 1	year ▲ 2	total_revenue
Foon Yue Tseng	2003	221887.03
Foon Yue Tseng	2004	237255.26
Foon Yue Tseng	2005	29070.38
Pamela Castillo	2003	317104.78
Pamela Castillo	2004	409910.07
Pamela Castillo	2005	23187.02

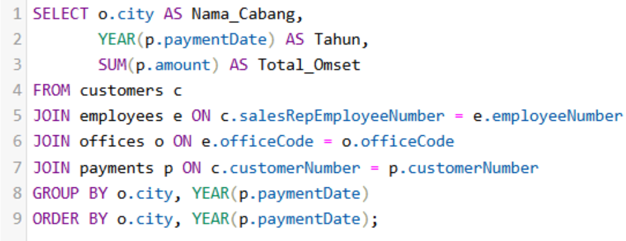
Grafik :



Studi Kasus

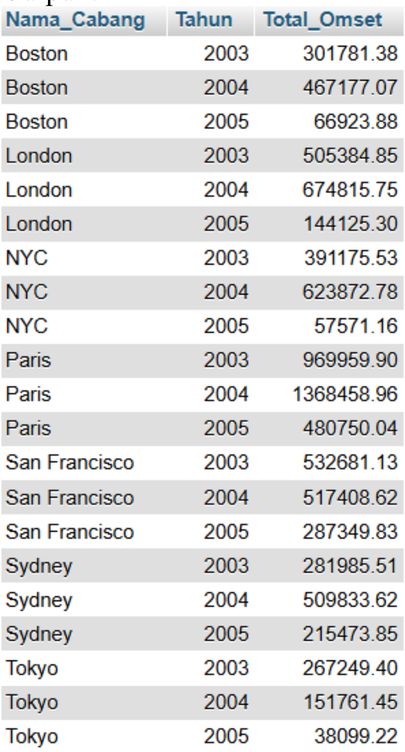
1. Field apa saja yang diperlukan untuk menampilkan penjualan di setiap cabang.
   - branchName (Nama Cabang) → Menampilkan nama cabang.
   - year (Tahun) → Untuk membedakan omset berdasarkan tahun.
   - total_sales (Jumlah Omset) → Total pendapatan per cabang per tahun.
   - **Relasi antar tabel:** Jika data omset terdapat di tabel transaksi (sales atau orders), maka kita perlu:
     - branchID (ID Cabang) → Untuk menghubungkan dengan nama cabang di tabel branches.
     - orderDate (Tanggal Transaksi) → Untuk mengekstrak tahun.
     - orderAmount (Nilai Transaksi) → Untuk menghitung total omset.

2. Query :



```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);
```

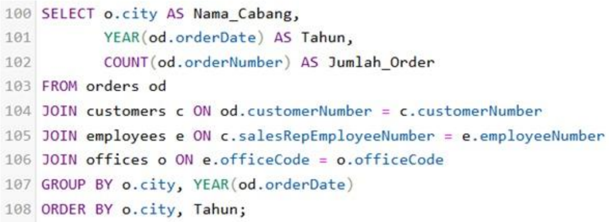
Output :



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
Sydney	2004	509833.62
Sydney	2005	215473.85
Tokyo	2003	267249.40
Tokyo	2004	151761.45
Tokyo	2005	38099.22

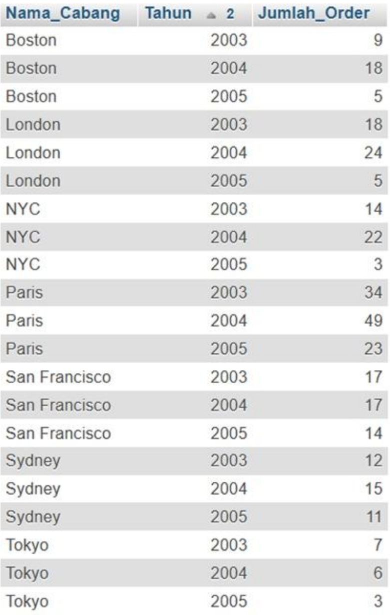
Soal Bonus

Query :



```
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;
```

Output :



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

Graph:

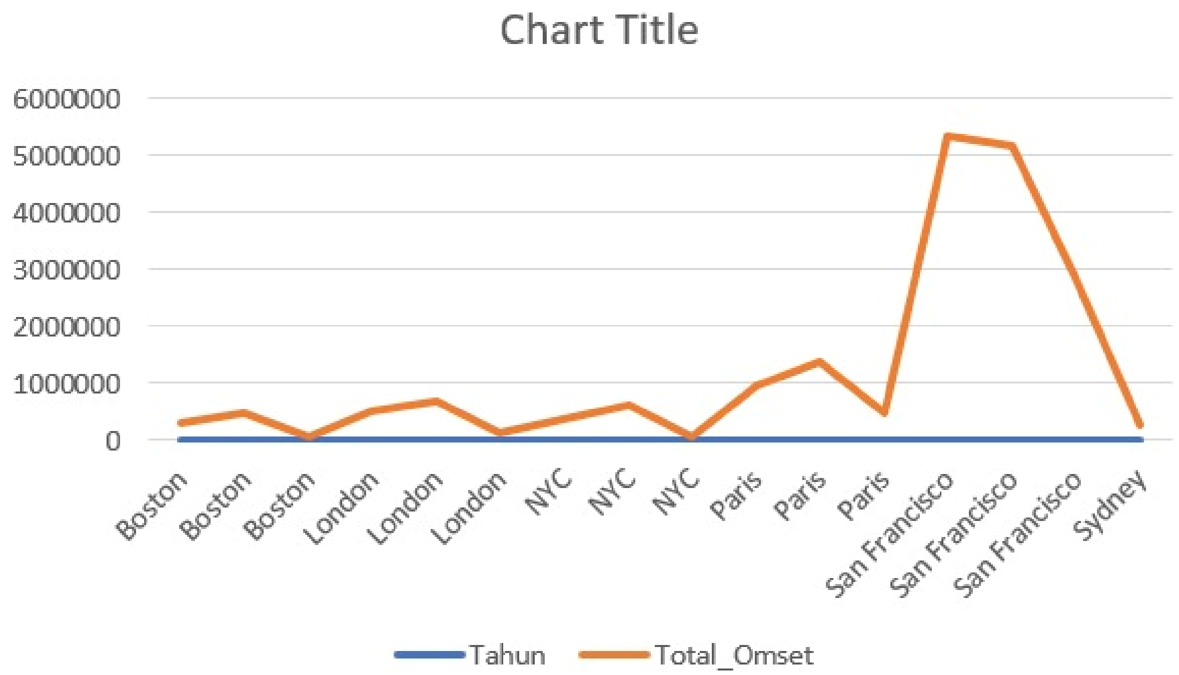


Chart Title
6000000
5000000
4000000
3000000
2000000
1000000
0
Boston
Boston
Boston
London
London
London
NYC
NYC
NYC
Paris
Paris
Paris
San Francisco
San Francisco
San Francisco
Sydney
Tahun
Total_Omset