LAPORAN PRAKTIKUM
DATA WAREHOUSE

JOBSHEET 2
DATABASE OPERASIONAL

Oleh:

NIMAS SEPTIANDINI

2341760087

SIB 2B

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

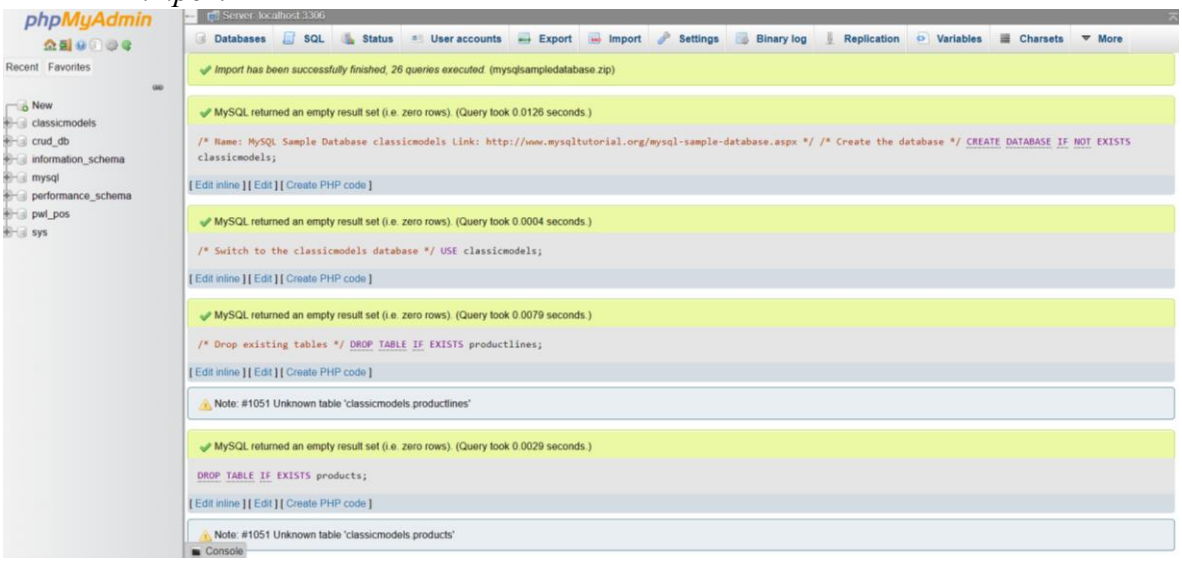
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2025/2026

Tugas 1

1. Import data perusahaan tersebut pada DBMS MySQL!
   - Proses *import*



   - Hasil *import*

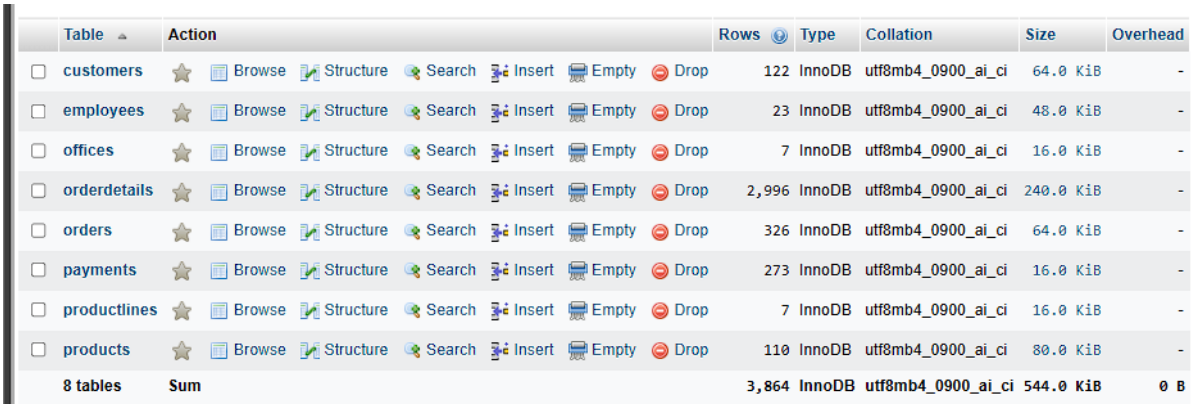


	Table	Action	Rows	Type	Collation	Size	Overhead
☐	customers	Browse Structure Search Insert Empty Drop	122	InnoDB	utf8mb4_0900_ai_ci	64.0 KiB	-
☐	employees	Browse Structure Search Insert Empty Drop	23	InnoDB	utf8mb4_0900_ai_ci	48.0 KiB	-
☐	offices	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_0900_ai_ci	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
productlines	products	one to many
employees	offices	many to one
employees	employees	one to many (self-referencing)
customers	employees	many to one
customers	orders	one to many
orders	ordeerdetails	one to many
orderdetails	products	many to one
payment	customers	many to one

3. Analisa jumlah field pada setiap tabel!

Nama Tabel	Jumlah field
Productlines	4
Products	9
Employees	8

Offices	9
Customers	9
Orders	6
Orderdetails	4
payments	4

A. Analisa Data

Praktikum 1

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

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

Hasil run *query*:

Error

SQL query: Copy

```
SELECT *
FROM employees employe, employes manager, customer cust
WHERE employee.reportsTo=manager.employeeNumber
AND employee.employeeNumber=cust.salesRepEmployeeNumber LIMIT 0, 25
```

MySQL said:

```
#1146 - Table 'classicmodels.employes' doesn't exist
```

= Terjadi *error* dikarenakan terjadi kesalahan penulisan employes yang seharusnya employees dan customer yang harusnya customers. Berikut adalah *query* yang telah diperbaiki beserta hasil run nya. Hasil run query akan menampilkan data Employee beserta Manajer dan Customernya.

```
1 SELECT *
2 FROM employees employee
3 JOIN employees manager ON employee.reportsTo = manager.employeeNumber
4 JOIN customers cust ON employee.employeeNumber = cust.salesRepEmployeeNumber;
```

2. Buka **tab baru** pada browser untuk melakukan eksekusi **query** berikut. dari hasil **query** diatas maka akan ditemukan atasan dari setiap pegawai.

```
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;
```

Saat *query* tersebut dijalankan, terjadi *error* seperti berikut:

Error

Static analysis:

1 errors were found during analysis.

1. Ending quote " was expected. (near "" at position 327)

SQL query: Copy

```
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;;
```

MySQL said:

```
#1064 - You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '",employee.lastName) as staff
FROM employees employee, employees manager
WHE' at line 4
```

= Error tersebut terjadi karena terjadi kesalahan pada penulisan CONCAT(), khususnya pada kutipan (") yang tidak tertutup dengan benar dalam parameter manager.lastName. Berikut adalah query yang telah diperbaiki serta hasilnya

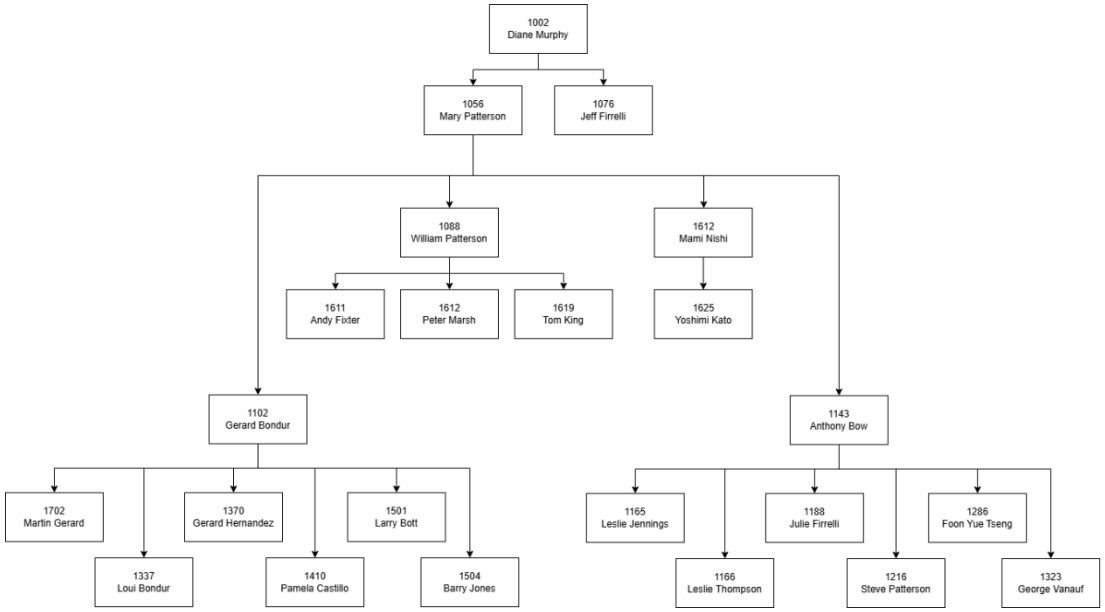
```sql
SELECT
    manager.employeeNumber AS id_manager,
    CONCAT(manager.firstName, " ", manager.lastName) AS Manager,
    employee.employeeNumber AS id_staff,
    CONCAT(employee.firstName, " ", employee.lastName) AS staff
FROM employees employee
JOIN employees manager ON employee.reportsTo = manager.employeeNumber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

```

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!
   - Gambar hirarki



- Tautan draw.io: https://drive.google.com/file/d/18tmzozcXNmRwrpEunGEyFkO8XTU6iNdc/view?usp=sharing

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.reportsTomanager.employeeNumber
left join customers cust on
employee.employeeNumber=cust.salesRepEmployeeNumber
GROUP BY employee.employeeNumber
ORDER BY manager.firstName;
```

= Saat *query* tersebut dieksekusi, terjadi error seperti gambar berikut. Error berikut terjadi karena reportsTomanager bukan kolom yang valid, harusnya employee.reportsTo, dari skema database yang menunjukkan atasan pegawai adalah reportsTo.

Error

SQL query: Copy

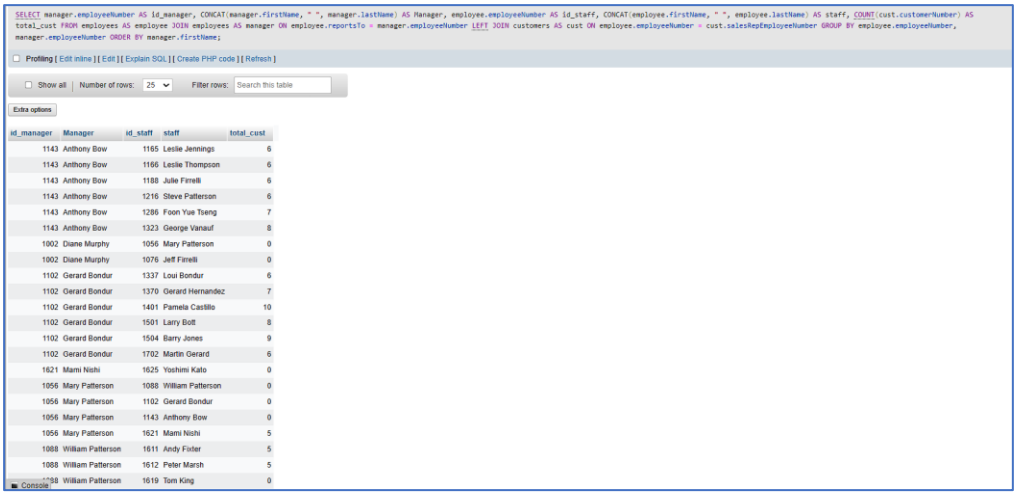
```
SELECT manager.employeeNumber as id_manager,
concat(manager.firstName," ",manager.lastName) as Manager,
employee.employeeNumber as id_staff, concat(employee.firstName," ",employee.lastName) as staff,
count(cust.customerNumber) as total_cust
FROM employees employee join employees manager on employee.reportsTomanager.employeeNumber
left join customers cust on employee.employeeNumber=cust.salesRepEmployeeNumber
```

MySQL said:

```
#1054 - Unknown column 'employee.reportsTomanager.employeeNumber' in 'on clause'
```

= Berikut adalah *query* yang telah diperbaiki beserta hasil eksekusinya

```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 AS employee
JOIN employees AS manager ON employee.reportsTo = manager.employeeNumber
LEFT JOIN customers AS cust ON employee.employeeNumber = cust.salesRepEmployeeNumber
GROUP BY employee.employeeNumber, manager.employeeNumber
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?
   = Pamela Catillo dengan total 10 customer
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!

Rank.	Staff	Total_cust	Keterangan
1.	Gerrard Bondur	46	Bawahan: 6 staff Total_cust: 6+7+10+8+9+6 = 46
2.	Marry Patterson	22	Bawahan: William Patterson (5+5+5=15) + Gerard Bondur (6+7+10+8+9+6=46) + Anthony Bow (8) + Mami Nishi (5)
3.	Dianer Murphy	8	Bawahan: Mary Patterson (0) + Jeff Firrelli (0) + Anthony Bow (8)
4.	Anthony Bow	8	Bawahan: 6 staff Total_cust: 6+6+6+6+7+8 = 39

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 Nama_Pegawai,
    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
GROUP BY e.employeeNumber
ORDER BY Total_Omset DESC;
```

employeeNumber	Nama_Pegawai	Total_Omset ▾ 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?
   = Jumlah customer yang bertransaksi membutuhkan 7 field dari tabel employees, customers, dan hasil perhitungan dari COUNT(customerNumber). Sedangkan jumlah omset yang didapat membutuhkan 9 field, termasuk amount dari tabel payments dan hasil perhitungan dari SUM(amount). Berikut ini adalah tabel jumlah field yang dibutuhkan untuk mendapatkan informasi tersebut

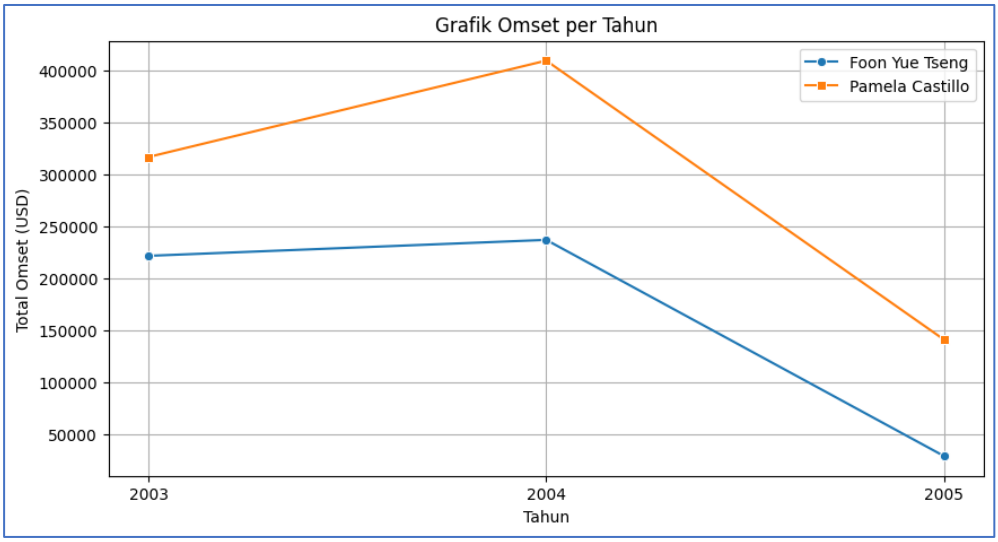
KPI	Jumlah Field
Jumlah customer yang bertransaksi	7 Field
Jumlah omset yang didapat	9 Field

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

```sql
SELECT e.employeeNumber,
    CONCAT(e.firstName, ' ', e.lastName) AS Nama,
    YEAR(o.orderDate) AS Tahun,
    SUM(od.quantityOrdered * od.priceEach) AS Total_Omset
FROM employees e
JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN orders o ON c.customerNumber = o.customerNumber
JOIN orderdetails od ON o.orderNumber = od.orderNumber
WHERE e.employeeNumber IN (1286, 1401)
GROUP BY e.employeeNumber, Tahun
ORDER BY Tahun;
```

employeeNumber	Nama	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	141205.70

Hasil konversi kedalam bentuk grafik:



= Dari grafik tersebut, dapat dilihat bahwa tren omset per tahun Foon Yue Tseng naik pada tahun 2004 kemudian turun pada tahun 2005, sedangkan tren omset tahunan Pamela Castillo mengalami kenaikan pada tahun 2004 dan penurunan pada tahun 2005 yang fluktuatif. Kinerja Pamela Catillo terlihat lebih tinggi di setiap tahunnya daripada Foon Yue Tseng. Kedua Staff tersebut mengalami penurunan drastic pada tahun 2005.

Google colab grafik:
https://colab.research.google.com/drive/1nxNyu5ug1A7m23ClJedQnqsA5PGAxIgm?usp=sharing

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
...			
...			

Analisis terlebih dahulu:

1. Field apa saja yang diperlukan untuk menampilkan penjualan di setiap cabang.
   - branchName: Nama Cabang
   - year(orderDate): Tahun Penjualan
   - SUM(amount): Total Omset per Cabang per Tahun
   - orderNumber: ID Transaksi
   - customerNumber: ID Pelanggan
   - salesRepEmployeeNumber: ID Sales yang menangani transaksi
2. Bentuk *query* dengan memperhatikan relasi antar tabel
   - offices: Menyimpan data cabang perusahaan.
   - employees: Menyimpan data karyawan (termasuk sales).
   - customers: Menyimpan data pelanggan, termasuk sales yang menangani.
   - orders: Menyimpan informasi pesanan, termasuk tanggal order dan pelanggan.
   - orderdetails: Menyimpan detail pesanan, seperti jumlah dan harga barang yang dibeli.
   - payments: Menyimpan informasi pembayaran dari pelanggan (total omset).

```sql
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, YEAR(ord.orderDate);
```

branchName	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

Showing rows 0 - 6 (7 total, Query took 0.0167 seconds.)

```
SELECT branchName AS Nama_Cabang, SUM(CASE WHEN Tahun = 2003 THEN Total_Omset ELSE
0 END) AS '2003', SUM(CASE WHEN Tahun = 2004 THEN Total_Omset ELSE 0 END) AS
'2004', SUM(CASE WHEN Tahun = 2005 THEN Total_Omset ELSE 0 END) AS '2005' FROM (
SELECT o.city AS branchName, 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 =
```

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

Show all | Number of rows: 25

Extra options

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.

= Analisis waktu paling sibuk dalam setahun dengan menentukan bulan paling sibuk tiap tahunnya berdasarkan jumlah transaksi

1. Field yang dibutuhkan.
    - Tahun (YEAR(orderDate)): Untuk melihat perubahan tiap tahun.
    - Bulan (MONTH(orderDate)): Untuk melihat pola bulanan.
    - Jumlah Order (COUNT(orderNumber)): Untuk menghitung total transaksi tiap bulan.
    - Total Omset (SUM(priceEach * quantityOrdered)): Untuk melihat dampak penjualan.
2. *Query* SQL untuk menampilkan report

```
SELECT YEAR(o.orderDate) AS Tahun,
       MONTH(o.orderDate) AS Bulan,
       COUNT(o.orderNumber) AS Total_Transaksi,
       SUM(od.priceEach * od.quantityOrdered) AS Total_Omset
FROM orders o
JOIN orderdetails od ON o.orderNumber = od.orderNumber
GROUP BY Tahun, Bulan ORDER BY Tahun ASC, Bulan ASC;
```

```
SELECT YEAR(o.orderDate) AS Tahun, MONTH(o.orderDate) AS Bulan, COUNT(o.orderNumber) AS Total_Transaksi, SUM(od.priceEach *
od.quantityOrdered) AS Total_Omset FROM orders o JOIN orderdetails od ON o.orderNumber = od.orderNumber GROUP BY Tahun, Bulan ORDER BY
Tahun ASC, Bulan ASC;
```

3. Struktur tabel report

Tahun	Bulan	Total_Transaksi	Total_omset
2003	Jan	120	320.000
2003	Feb	90	275.000
2003	Mar	150	400.000
2004	Jan	130	350.000
2004	Feb	110	300.000
2004	Mar	180	500.000
2005	Jan	160	420.000
2005	Feb	140	380.000
2005	Mar	200	550.000

4. Grafik tren transaksi bulanan tiap tahun