MySQL Northwind Queries - Part 3

This is part 3 of the tutorial series - converting the popular Microsoft Access Northwind database to MySQL queries. These queries are originated from Access Northwind Traders application. Some of them are relatively complex aggregated queries with sub-queries.

11. Products Above Average Price

This query shows how to use sub-query to get a single value (average unit price) that can be used in the outer-query.

```
select distinct ProductName, UnitPrice
from Products
where UnitPrice > (select avg(UnitPrice) from Products)
order by UnitPrice;
```

Here is the query result. 25 records returned.

ProductName	UnitPrice
Uncle Bob's Organic Dried Pears	30
Ikura	31
Gumbär Gummibärchen	31.23
Mascarpone Fabioli	32
Perth Pasties	32.8
Wimmers gute Semmelknödel	33.25
Camembert Pierrot	34
Mozzarella di Giovanni	34.8
Gudbrandsdalsost	36
Gnocchi di nonna Alice	38
Queso Manchego La Pastora	38
Alice Mutton	39
Northwoods Cranberry Sauce	40

12. Product Sales for 1997

This query shows how to group categories and products by quarters and shows sales amount for each quarter.

```
select distinct a.CategoryName,
    b.ProductName,
    format(sum(c.UnitPrice * c.Quantity * (1 - c.Discount)), 2) as ProductSales,
    concat('Qtr ', quarter(d.ShippedDate)) as ShippedQuarter
from Categories a
inner join Products b on a.CategoryID = b.CategoryID
inner join Order_Details c on b.ProductID = c.ProductID
inner join Orders d on d.OrderID = c.OrderID
where d.ShippedDate between date('1997-01-01') and date('1997-12-31')
group by a.CategoryName,
    b.ProductName,
    concat('Qtr ', quarter(d.ShippedDate))
order by a.CategoryName,
    b.ProductName,
    ShippedQuarter;
```

Here is the query result. 286 records returned.

CategoryName	ProductName	ProductSales	ShippedQuarter
Beverages	Chai	705.60	Qtr 1
Beverages	Chai	878.40	Qtr 2
Beverages	Chai	1,174.50	Qtr 3
Beverages	Chai	2,128.50	Qtr 4
Beverages	Chang	2,720.80	Qtr 1
Beverages	Chang	228.00	Qtr 2
Beverages	Chang	2,061.50	Qtr 3
Beverages	Chang	2,028.25	Qtr 4
Beverages	Chartreuse verte	590.40	Qtr 1
Beverages	Chartreuse verte	360.00	Qtr 2
Beverages	Chartreuse verte	817.20	Qtr 3
Beverages	Chartreuse verte	2,424.60	Qtr 4

13. Category Sales for 1997

This query shows sales figures by categories - mainly just aggregation with sub-query. The inner query aggregates to product level, and the outer query further aggregates the result set from inner-query to category level.

```
select CategoryName, format(sum(ProductSales), 2) as CategorySales
from
(
    select distinct a.CategoryName,
        b.ProductName,
        format(sum(c.UnitPrice * c.Quantity * (1 - c.Discount)), 2) as ProductSales,
        concat('Qtr ', quarter(d.ShippedDate)) as ShippedQuarter
    from Categories as a
    inner join Products as b on a.CategoryID = b.CategoryID
    inner join Order_Details as c on b.ProductID = c.ProductID
    inner join Orders as d on d.OrderID = c.OrderID
    where d.ShippedDate between date('1997-01-01') and date('1997-12-31')
    group by a.CategoryName,
        b.ProductName,
        concat('Qtr ', quarter(d.ShippedDate))
    order by a.CategoryName,
        b.ProductName,
        ShippedQuarter
) as x
group by CategoryName
order by CategoryName;
```

Here is the query result. 8 records returned.

CategoryName	CategorySales
Beverages	7,654.34
Condiments	7,719.90
Confections	13,018.22
Dairy Products	3,610.40
Grains/Cereals	11,611.15
Meat/Poultry	4,971.59
Produce	1,402.81
Seafood	8,371.13

14. Quarterly Orders by Product

This query shows how to convert order dates to the corresponding quarters. It also demonstrates how SUM function is used together with CASE statement to get sales for each quarter, where quarters are converted from OrderDate column.

```
select a.ProductName,
    d.CompanyName,
    year(OrderDate) as OrderYear,
    format(sum(case quarter(c.OrderDate) when '1'
        then b.UnitPrice*b.Quantity*(1-b.Discount) else 0 end), 0) "Qtr 1",
    format(sum(case quarter(c.OrderDate) when '2'
        then b.UnitPrice*b.Quantity*(1-b.Discount) else 0 end), 0) "Qtr 2",
    format(sum(case quarter(c.OrderDate) when '3'
        then b.UnitPrice*b.Quantity*(1-b.Discount) else 0 end), 0) "Qtr 3",
    format(sum(case quarter(c.OrderDate) when '4'
        then b.UnitPrice*b.Quantity*(1-b.Discount) else 0 end), 0) "Qtr 4"
from Products a
inner join Order_Details b on a.ProductID = b.ProductID
inner join Orders c on c.OrderID = b.OrderID
inner join Customers d on d.CustomerID = c.CustomerID
where c.OrderDate between date('1997-01-01') and date('1997-12-31')
group by a.ProductName,
    d.CompanyName,
    year(OrderDate)
order by a.ProductName, d.CompanyName;
```

Here is the query result. 947 records returned.

ProductName	CompanyName	OrderYear	Qtr 1	Qtr 2	Qtr 3	Qtr 4
Alice Mutton	Antonio Moreno Taquería	1997	0	702	0	0
Alice Mutton	Berglunds snabbköp	1997	312	0	0	0
Alice Mutton	Bólido Comidas preparadas	1997	0	0	0	1,170
Alice Mutton	Bottom-Dollar Markets	1997	1,170	0	0	0
Alice Mutton	Ernst Handel	1997	1,123	0	0	2,607
Alice Mutton	Godos Cocina Típica	1997	0	281	0	0
Alice Mutton	Hungry Coyote Import Store	1997	62	0	0	0
Alice Mutton	Piccolo und mehr	1997	0	1,560	936	0
Alice Mutton	Rattlesnake Canyon Grocery	1997	0	593	0	0
Alice Mutton	Reggiani Caseifici	1997	0	0	0	741
Alice Mutton	Save-a-lot Markets	1997	0	0	3,900	790

15. Invoice

A simple query to get detailed information for each sale so that invoice can be issued.

```
select distinct b.ShipName,
    b.ShipAddress,
    b.ShipCity,
    b.ShipRegion,
    b.ShipPostalCode,
    b.ShipCountry,
    b.CustomerID,
    c.CompanyName,
    c.Address,
    c.City,
    c.Region,
    c.PostalCode,
    c.Country,
    concat(d.FirstName,  ' ', d.LastName) as Salesperson,
    b.OrderID,
    b.OrderDate,
    b.RequiredDate,
    b.ShippedDate,
    a.CompanyName,
    e.ProductID,
    f.ProductName,
    e.UnitPrice,
    e.Quantity,
    e.Discount,
    e.UnitPrice * e.Quantity * (1 - e.Discount) as ExtendedPrice,
    b.Freight
from Shippers a
inner join Orders b on a.ShipperID = b.ShipVia
inner join Customers c on c.CustomerID = b.CustomerID
inner join Employees d on d.EmployeeID = b.EmployeeID
inner join Order_Details e on b.OrderID = e.OrderID
inner join Products f on f.ProductID = e.ProductID
order by b.ShipName;
```

Here is the query result. 2,155 records returned.

ShipName	ShipAddress	ShipCity	ShipRegion	ShipPostalCode	Sh
Alfreds Futterkiste	Obere Str. 57	Berlin		12209	Ge
Alfreds Futterkiste	Obere Str. 57	Berlin		12209	Ge
Alfreds Futterkiste	Obere Str. 57	Berlin		12209	Ge
Alfreds Futterkiste	Obere Str. 57	Berlin		12209	Ge
Ana Trujillo Emparedac	Avda. de la Constitución 2222	México D.F.		05021	Me
Ana Trujillo Emparedac	Avda. de la Constitución 2222	México D.F.		05021	Me
Ana Trujillo Emparedac	Avda. de la Constitución 2222	México D.F.		05021	Me
Ana Trujillo Emparedac	Avda. de la Constitución 2222	México D.F.		05021	Me
Ana Trujillo Emparedac	Avda. de la Constitución 2222	México D.F.		05021	Me
Ana Trujillo Emparedac	Avda. de la Constitución 2222	México D.F.		05021	Me

16. Number of units in stock by category and supplier continent

This query shows that case statement is used in GROUP BY clause to list the number of units in stock for each product category and supplier's continent. Note that, if only s.Country (not the case statement) is used in the GROUP BY, duplicated rows will exist for each product category and supplier continent.

```
select c.CategoryName as "Product Category",
       case when s.Country in
                 ('UK','Spain','Sweden','Germany','Norway',
                  'Denmark','Netherlands','Finland','Italy','France')
            then 'Europe'
            when s.Country in ('USA','Canada','Brazil')
            then 'America'
            else 'Asia-Pacific'
        end as "Supplier Continent",
        sum(p.UnitsInStock) as UnitsInStock
from Suppliers s
inner join Products p on p.SupplierID=s.SupplierID
inner join Categories c on c.CategoryID=p.CategoryID
group by c.CategoryName,
         case when s.Country in
                 ('UK','Spain','Sweden','Germany','Norway',
                  'Denmark','Netherlands','Finland','Italy','France')
              then 'Europe'
              when s.Country in ('USA','Canada','Brazil')
              then 'America'
              else 'Asia-Pacific'
         end;
```

Here is the query result. 21 records returned.

Product Category	Supplier Continent	UnitsInStock
Beverages	America	203
Beverages	Asia-Pacific	32
Beverages	Europe	324
Condiments	America	372
Condiments	Asia-Pacific	90
Condiments	Europe	45
Confections	America	17
Confections	Asia-Pacific	29
Confections	Europe	340
Dairy Products	Europe	393
Grains/Cereals	Asia-Pacific	64
Grains/Cereals	Europe	244
Meat/Poultry	America	136

Here comes the end of this article series. I hope you find it useful in your day-to-day job of SQL coding! Don't forget to download the full script which can be found at the beginning of the first two parts of this article series.

Other tutorials in this category

1. MySQL Northwind Queries - Part 1
2. MySQL Northwind Queries - Part 2
3. How to work with two unrelated values in MySQL

Back to Tutorial Index Page