DSO 552: SQL Databases for Business Analysts

Extra Practice Problems (Joins, Subqueries, Aggregations)

Northwind Traders Company

Northwind Traders is a company that imports and exports food globally. The database captures all the sales transactions that occurs between the company i.e. Northwind traders and its customers as well as the purchase transactions between Northwind and its suppliers.

The following explains each table (used in this assignment) in the Northwind database:

Table	Description
Customers	who buy from Northwind
Orders	stores transaction sale orders from customers
OrderDetails	stores line items of sale orders
Products	the products that Northwind trades in
Suppliers	who supply to the company
Shippers	details of the shippers who ship the products from the traders to the end-customers
Employees	who work for Northwind

Check Figure 1 (on the last page) for more details for each of the above tables.

1. ***(1 point)*** **Using a `GROUP BY` statement, produce a report containing the `companyname`, `contactname`, and `num_orders` (number of orders) that each company has made. Show the top five companies that have had the most orders. Do not use a subquery in this question.**

```
SELECT c.companyname, c.contactname, COUNT(DISTINCT o.orderid) AS num_orders
FROM customers c
         JOIN orders o ON c.customerid = o.customerid
GROUP BY 1, 2
ORDER BY 3 DESC
LIMIT 5;
```

Sample Output:

companyname	contactname	num_orders
Save-a-lot Markets	Jose Pavarotti	31
Ernst Handel	Roland Mendel	30
QUICK-Stop	Horst Kloss	28
Folk och fä HB	Maria Larsson	19
Hungry Owl All-Night Grocers	Patricia McKenna	19

2. ***(1 point)*** **Using a subquery, produce a report containing the `companyname`, `contactname`, and `num_orders` (number of orders) that each company has made. Do not use a GROUP BY to produce this report. Show the top five companies that have had the most orders.**

```sql
SELECT c.companyname,
       c.contactname,
       (
           SELECT COUNT(*)
           FROM orders o
           WHERE o.customerid = c.customerid) num_orders
FROM customers c
ORDER BY 3 DESC
LIMIT 5;
```

Sample Output:

companyname	contactname	num_orders
Save-a-lot Markets	Jose Pavarotti	31
Ernst Handel	Roland Mendel	30
QUICK-Stop	Horst Kloss	28
Folk och fä HB	Maria Larsson	19
Hungry Owl All-Night Grocers	Patricia McKenna	19

3. ***(1 point)*** **Northwind wants to ensure that all customers purchase at least one order. Produce a report showing a list of customers where customers have not ordered, and how many customers there are in these countries.**

```sql
SELECT country, COUNT(DISTINCT no_order_customers.customerid) number_of_no_order_customers
FROM (
         SELECT c.*
         FROM customers c
                  LEFT JOIN orders o ON c.customerid = o.customerid
         WHERE o.orderid IS NULL) no_order_customers
GROUP BY 1
```

Sample Output:

country	number_of_no_order_customers
France	1
Spain	1

4. ***(1 point)*** **Produce a report showing all customers IDs and number of orders placed for customers who have more than the average number of orders per customer. Hint - find the average number of orders per customer first, and use that as part of a `GROUP BY` and `HAVING` clause. Show the top ordering customers first and limit your output to the top 5 countries.**

```sql
SELECT c.customerid, COUNT(*) num_orders
FROM customers c
         JOIN orders o ON c.customerid = o.customerid
GROUP BY 1
HAVING COUNT(*) > (
    -- subquery that returns the average number of orders per customer
    SELECT AVG(num_orders)
    FROM (
             SELECT c.customerid, COUNT(*) num_orders
             FROM customers c
                      JOIN orders o ON c.customerid = o.customerid
             GROUP BY 1) num_orders_per_customer)
```

```
ORDER BY 2 DESC
LIMIT 5
```

Sample Output:

customerid	num_orders
SAVEA	31
ERNSH	30
QUICK	28
HUNGO	19
FOLKO	19

5. ***(1 point)*** **Northwind always wants to ensure that the best selling products are in stock. There is a Tableau dashboard that is connected to the enterprise data warehouse which displays the average `unitsinstock` for top products. Write the backend SQL query that will compute the average `unitsinstock` for the top 5 best selling products. We define best selling as the total order value of a product as computed using `unitprice * quantity`.**

```
SELECT AVG(unitsinstock) unitsinstock_for_best_selling_products
FROM products
WHERE productid IN (
    -- subquery to get the top 5 best selling products by total order value.
    SELECT productid
    FROM order_details
    GROUP BY 1
    ORDER BY SUM(unitprice * quantity) DESC
    LIMIT 5
)
```

Sample Output:

unitsinstock_for_best_selling_products
26.4

6. ***(1 point)*** **The HR onboarding team wants to use the top employees' most recent order transactions as exemplars to show the newest batch of Northwind employees. Write a query that shows the order ID, customer ID, freight, and order date for the 10 most orders processed by employees who have processed over 100,000 in total value (`unitprice * quantity`) for orders.**

```
SELECT orderid, o.customerid, o.freight, orderdate
FROM orders o
WHERE employeeid IN (
-- subquery to get the employee IDs of employees who have processed over
-- 100,000 in total order value
    SELECT e.employeeid
    FROM employees e
             JOIN orders o ON e.employeeid = o.employeeid
             JOIN order_details od ON o.orderid = od.orderid
    GROUP BY 1
    HAVING SUM(od.quantity * od.unitprice) > 100000)
ORDER BY orderdate DESC
LIMIT 10;
```

Sample Output:

orderid	customerid	freight	orderdate
11076	BONAP	38.28	1998-05-06
11075	RICSU	6.19	1998-05-06
11077	RATTC	8.53	1998-05-06
11074	SIMOB	18.44	1998-05-06
11070	LEHMS	136.00	1998-05-05
11072	ERNSH	258.64	1998-05-05
11073	PERIC	24.95	1998-05-05
11071	LILAS	0.93	1998-05-05
11068	QUEEN	81.75	1998-05-04
11067	DRACD	7.98	1998-05-04

7. ***(1 point)*** **A heavy order at Northwind is defined as an order that is larger than 200 in freight requirements. Show a report that lists the number of heavy freight orders per employee (listing their name). Only show employees that have had more than 10 heavy freight orders.**

```
SELECT e.firstname || ' ' || e.lastname AS employee_name,
       COUNT(DISTINCT o.orderid)        AS total_heavy_orders
FROM employees e
         JOIN orders o ON e.employeeid = o.employeeid
WHERE o.freight > 200
GROUP BY 1
HAVING COUNT(DISTINCT o.orderid) > 10;
```

Sample Output:

employee_name	total_heavy_orders
Andrew Fuller	11
Margaret Peacock	13

8. ***(1 point)*** **A heavy order at Northwind is defined as an order that is larger than 200 in freight requirements. Show a report that lists the number of heavy freight orders per employee (listing their name). Only show employees that have had more heavy freight orders than the average number of orders per customer.**

```
SELECT e.firstname || ' ' || e.lastname AS employee_name,
       COUNT(DISTINCT o.orderid)        AS total_heavy_orders
FROM employees e
         JOIN orders o ON e.employeeid = o.employeeid
WHERE o.freight > 200
GROUP BY 1
HAVING COUNT(DISTINCT o.orderid) > (
    -- subquery to find the average number of orders per customer
    SELECT AVG(num_orders_per_customer) avg_orders_per_customer
    FROM (SELECT customerid, COUNT(orderid) num_orders_per_customer
          FROM orders
          GROUP BY 1) AS orders_per_customer)
```

Sample Output:

employee_name	total_heavy_orders
Andrew Fuller	11
Janet Leverling	10
Laura Callahan	10
Margaret Peacock	13

9. ***(1 point)*** **Northwind wants to improve on its fulfillment times of orders. List all employee names, shipper names, company names, and number of late order shipments (a shipment is late if `shippeddate` is after `requireddate`). Only show counts originating from orders that were supposed to be fulfilled by Federal Shipping or United Package.**

```sql
SELECT e.firstname || ' ' || e.lastname AS employee_name,

       s.companyname                    AS shipper_name,
       c.companyname                    AS customer_name,
       COUNT(DISTINCT orderid)          AS late_order_shipments
FROM orders o
         JOIN shippers s ON o.shipvia = s.shipperid
         JOIN customers c ON o.customerid = c.customerid
         JOIN employees e ON o.employeeid = e.employeeid
WHERE shippeddate > requireddate
  AND s.companyname IN ('Federal Shipping', 'United Package')
GROUP BY 1, 2, 3
```

Sample Output:

employee_name	shipper_name	customer_name	late_order_shipments
Andrew Fuller	United Package	Bon app'	1
Anne Dodsworth	United Package	HILARION-Abastos	1
Anne Dodsworth	United Package	Hungry Owl All-Night Grocers	1
Janet Leverling	Federal Shipping	Princesa Isabel Vinhos	1
Janet Leverling	United Package	Berglunds snabbköp	1
Janet Leverling	United Package	Morgenstern Gesundkost	1
Laura Callahan	Federal Shipping	Hungry Owl All-Night Grocers	1
Laura Callahan	United Package	Lazy K Kountry Store	1
Margaret Peacock	Federal Shipping	B's Beverages	1
Margaret Peacock	Federal Shipping	QUICK-Stop	1
Margaret Peacock	Federal Shipping	Save-a-lot Markets	1
Margaret Peacock	United Package	Great Lakes Food Market	1
Margaret Peacock	United Package	Island Trading	1
Margaret Peacock	United Package	Piccolo und mehr	1
Margaret Peacock	United Package	Suprêmes délices	1
Michael Suyama	Federal Shipping	Folk och fä HB	1
Michael Suyama	Federal Shipping	Gourmet Lanchonetes	1
Michael Suyama	United Package	Split Rail Beer & Ale	1
Nancy Davolio	Federal Shipping	Gourmet Lanchonetes	1
Nancy Davolio	United Package	Bon app'	1
Robert King	United Package	Gourmet Lanchonetes	1
Robert King	United Package	Lehmanns Marktstand	1
Robert King	United Package	Seven Seas Imports	1
Robert King	United Package	White Clover Markets	1
Steven Buchanan	Federal Shipping	Wartian Herkku	1

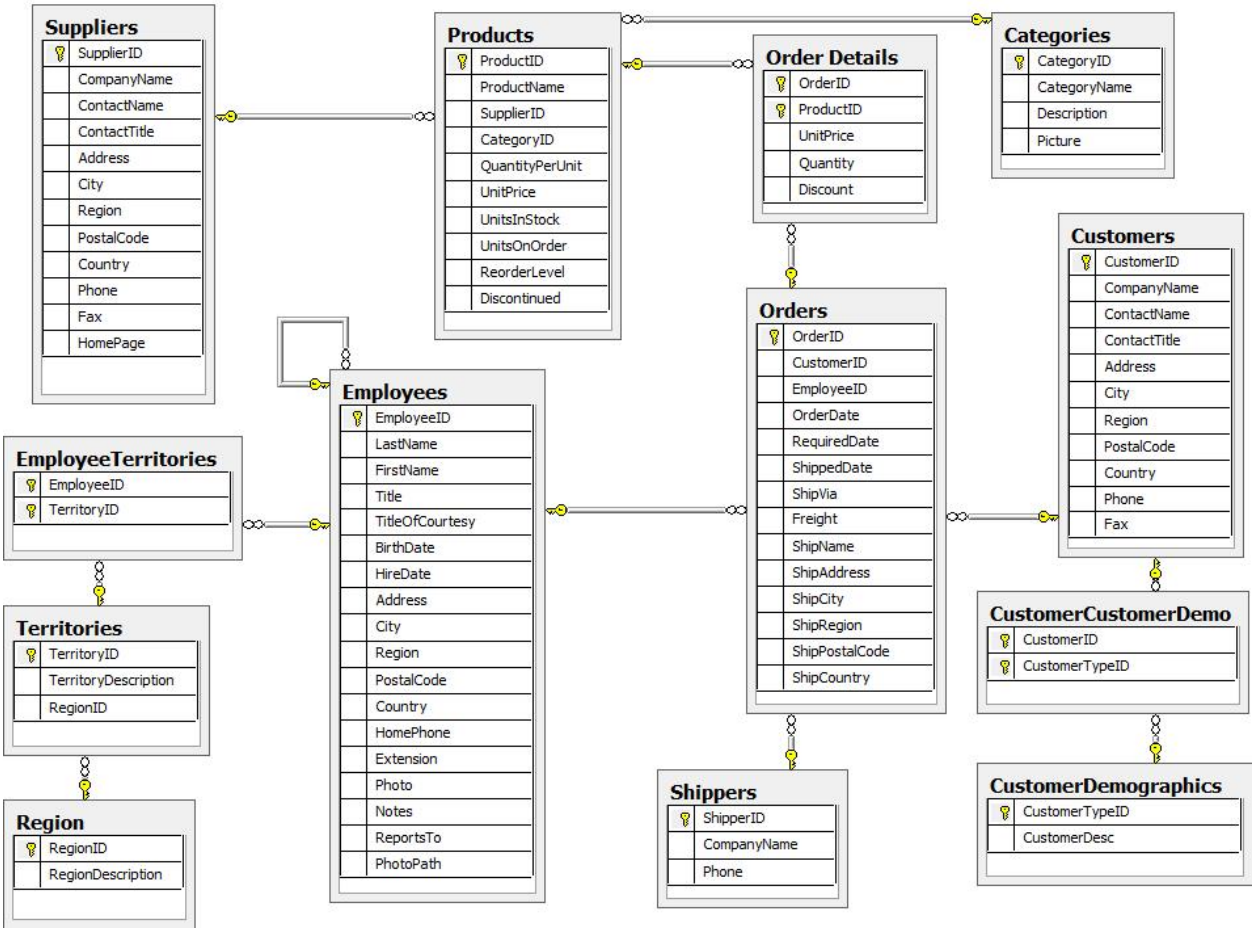

Figure 1: Northwind ERD