
# Northwind Traders

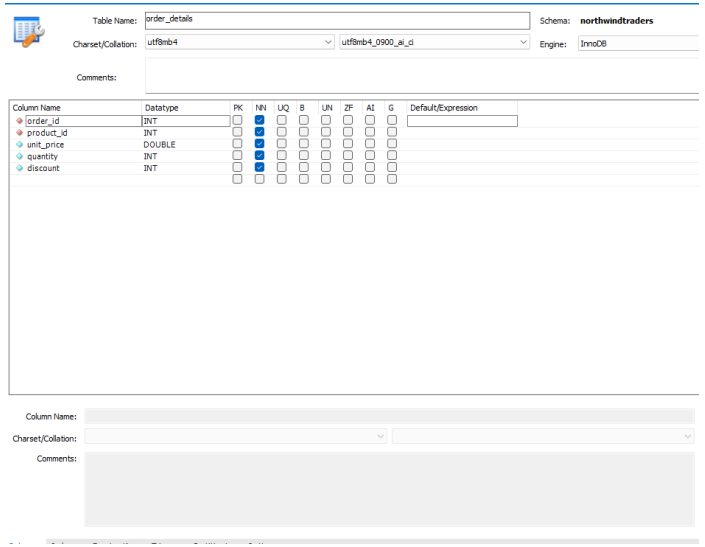
MYSQL EDA PROJECT

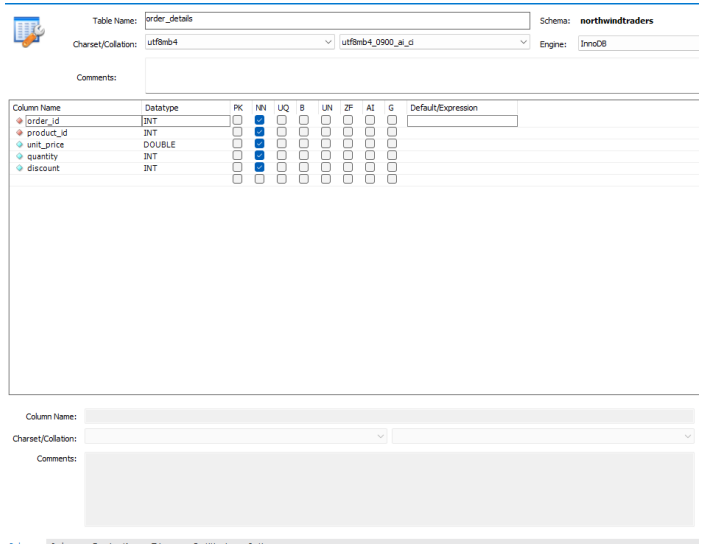
By Vikram Aiyer

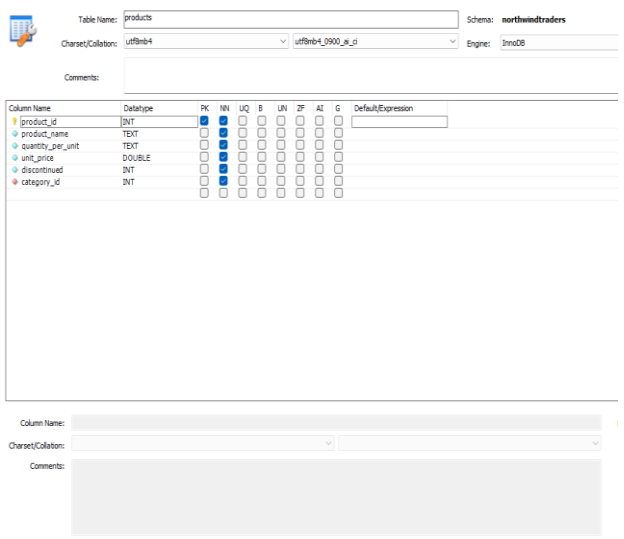
# Extract Transformation & Load (ETL)

```
-- The below code is the process of converting a 'date' string column to a 'date' format in mySql
        ALTER TABLE orders CHANGE COLUMN `orderDate` `order_date` VARCHAR(20);
        UPDATE orders SET order_date = date_format(str_to_date(order_date, '%Y/%m/%d'), '%Y-%m-%d');
        SELECT CAST(order_date AS date) FROM orders;
        ALTER TABLE orders MODIFY COLUMN `order_date` date;

        ALTER TABLE orders CHANGE COLUMN `requiredDate` `required_date` VARCHAR(20);
        UPDATE orders SET required_date = date_format(str_to_date(required_date, '%Y/%m/%d'), '%Y-%m-%d');
        SELECT CAST(required_date AS date) FROM orders;
        ALTER TABLE orders MODIFY COLUMN `required_date` date;

        ALTER TABLE orders CHANGE COLUMN `shippedDate` `shipped_date` VARCHAR(20);
        UPDATE orders SET shipped_date = date_format(str_to_date(shipped_date, '%Y/%m/%d'), '%Y-%m-%d');
        SELECT CAST(shipped_date AS date) FROM orders;
        ALTER TABLE orders MODIFY COLUMN `shipped_date` date;
```

# Identifying Cardinality

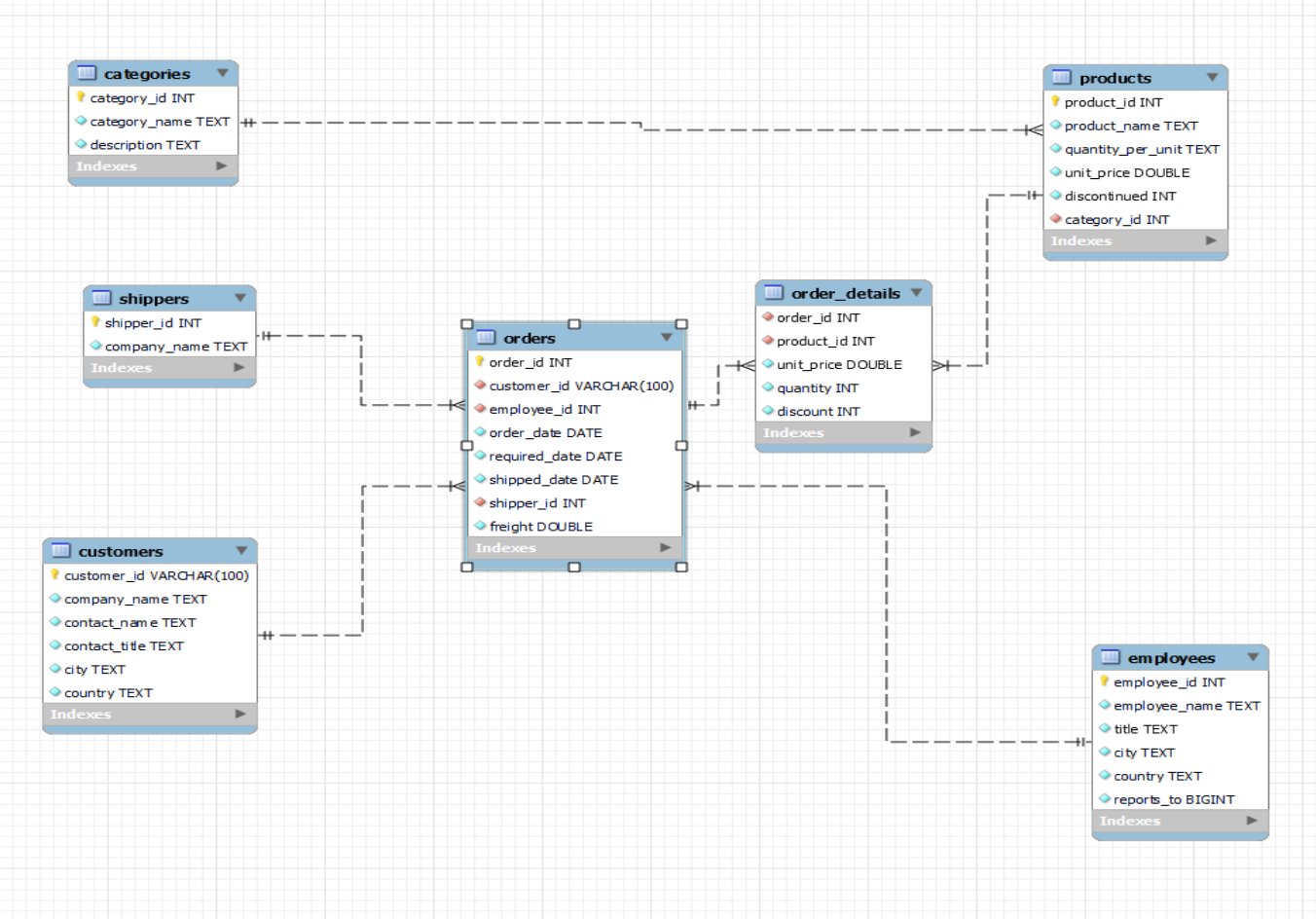



By Vikram Aiyer

# Data Model


categories
category_id INT
category_name TEXT
description TEXT
Indexes
products
product_id INT
product_name TEXT
quantity_per_unit TEXT
unit_price DOUBLE
discontinued INT
category_id INT
Indexes
shippers
shipper_id INT
company_name TEXT
Indexes
orders
order_id INT
customer_id VARCHAR(100)
employee_id INT
order_date DATE
required_date DATE
shipped_date DATE
shipper_id INT
freight DOUBLE
Indexes
order_details
order_id INT
product_id INT
unit_price DOUBLE
quantity INT
discount INT
Indexes
customers
customer_id VARCHAR(100)
company_name TEXT
contact_name TEXT
contact_title TEXT
city TEXT
country TEXT
Indexes
employees
employee_id INT
employee_name TEXT
title TEXT
city TEXT
country TEXT
reports_to BIGINT
Indexes

# Q1 Calculate Total Orders By year

```sql
-- Q1 Calculate Total Orders and sort it by year
SELECT
YEAR(order_date) AS yr,
COUNT(orders.order_id) AS total_orders
-- customer_id AS customers
FROM
orders
INNER JOIN
order_details
ON
order_details.order_id = orders.order_id
GROUP BY 1
ORDER BY 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| yr | total_orders |
|---|---|
| 2013 | 405 |
| 2014 | 1059 |
| 2015 | 691 |

# Q2 Calculate highest Orders By customer ID

```sql
-- Q2 Calculate highest Orders By customer ID
SELECT DISTINCT
COUNT(orders.order_id) AS total_orders,
(customer_id) AS customers
FROM
orders
LEFT JOIN
order_details
ON
order_details.order_id = orders.order_id
GROUP BY 2
ORDER BY 1 DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| total_orders | customers |
|---|---|
| 116 | SAVEA |
| 102 | ERNSH |
| 86 | QUICK |
| 71 | RATTC |
| 55 | HUNGO |
| 52 | BERGS |
| 48 | FRANK |
| 45 | FOLKO |

# Q3 Calculate total products in each categories

```sql
-- Q3 Calculate total products in each categories
SELECT
(category_name) As cat,
COUNT(products.category_id) AS total_products
FROM
categories
RIGHT OUTER JOIN
products
ON
products.category_id = categories.category_id
GROUP BY 1
ORDER BY 2 DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| cat | total_products |
|---|---|
| Confections | 13 |
| Beverages | 12 |
| Condiments | 12 |
| Seafood | 12 |
| Dairy Products | 10 |
| Grains & Cereals | 7 |
| Meat & Poultry | 6 |
| Produce | 5 |

Result 29

# Q4 What was the overall freight costs incurred by each company

```
75    -- Q4 What was the overall freight costs incurred by each company
76    SELECT
77    (shippers.company_name) AS shipping_co,
78    ROUND(SUM(freight),0) AS total_freight
79    FROM
80    orders
81    INNER JOIN shippers
82    ON
83    orders.shipper_id = shippers.shipper_id
84    GROUP BY 1
85    ORDER BY 2 DESC;
86
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| shipping_co | total_freight |
|---|---|
| United Package | 28245 |
| Federal Shipping | 20513 |
| Speedy Express | 16185 |

# Q5 What was the total sales by each product after discount?

```
87
88      -- Q5 What was the total sales by each product after discount?
89 •    SELECT
90      (products.product_name) AS prod_name,
91      ROUND(SUM(order_details.unit_price),2) - ROUND(SUM(order_details.discount),2) AS sales
92      FROM
93      order_details
94      INNER JOIN products
95      ON
96      order_details.product_id = products.product_id
97      GROUP BY 1
98      ORDER BY 2 DESC;
```

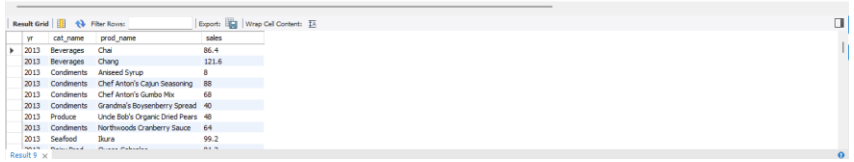
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| prod_name | sales |
|---|---|
| Côte de Blaye | 5902.4 |
| Thüringer Rostbratwurst | 3713.38 |
| Raclette Courdavault | 2761 |
| Tarte au sucre | 2227.8 |
| Manjimup Dried Apples | 1971.6 |
| Gnocchi di nonna Alice | 1770.8 |
| Camembert Pierrot | 1638.8 |
| Carnarvon Tigers | 1612.5 |

Result 9

# Q6 Use the same context and break in down by Year and categories

```sql
-- Q6 Use the same context and break in down by Year and categories
SELECT
    YEAR(o.order_date) AS yr,
    c.category_name AS cat_name,
    -- p.product_id AS p,
    p.product_name AS prod_name,
    ROUND(SUM(od.unit_price - od.discount), 2) AS sales
FROM
    order_details od
INNER JOIN
    products p ON od.product_id = p.product_id
LEFT JOIN
    orders o ON od.order_id = o.order_id
INNER JOIN
    categories c ON p.category_id = c.category_id
GROUP BY
    p.product_id,
    YEAR(o.order_date)
ORDER BY
    p.product_id,
    yr;
```



| yr | cat_name | prod_name | sales |
|---|---|---|---|
| 2013 | Beverages | Chai | 86.4 |
| 2013 | Beverages | Chang | 121.6 |
| 2013 | Condiments | Aniseed Syrup | 8 |
| 2013 | Condiments | Chef Anton's Cajun Seasoning | 88 |
| 2013 | Condiments | Chef Anton's Gumbo Mix | 68 |
| 2013 | Condiments | Grandma's Boysenberry Spread | 40 |
| 2013 | Produce | Uncle Bob's Organic Dried Pears | 48 |
| 2013 | Condiments | Northwoods Cranberry Sauce | 64 |
| 2013 | Seafood | Ikura | 99.2 |

```sql
-- In this step we will use unpivot method to show year as separate columns
-- Strangely MySql does not support PIVOT/UNPIVOT functions, thanks CHATgpt for saving my time!
    -- SELECT
--     total_sales_cat.yr,
--     total_sales_cat.cat_name,
--     total_sales_cat.prod_name,
--     total_sales_cat.sales
--     FROM
--     total_sales_cat
--     UNPIVOT
-- (
-- SUM(total_sales_cat.sales) FOR total_sales_cat.yr IN (total_sales_cat.cat_name, total_sales_cat.prod_name )
-- )AS PivotTable;
```

```sql
-- Q6 Use the same context and break in down by Year and categories

CREATE TEMPORARY TABLE total_sales_cat
SELECT
    YEAR(o.order_date) AS yr,
    c.category_name AS cat_name,
    -- p.product_id AS p,
    p.product_name AS prod_name,
    ROUND(SUM(od.unit_price - od.discount), 2) AS sales
FROM
    order_details od
INNER JOIN
    products p ON od.product_id = p.product_id
LEFT JOIN
    orders o ON od.order_id = o.order_id
INNER JOIN
    categories c ON p.category_id = c.category_id
GROUP BY
    p.product_id,
    YEAR(o.order_date)
ORDER BY
    p.product_id,
    yr;
```

```sql
    -- Chatgpt version
    SELECT
    cat_name,
    prod_name,
    SUM(CASE WHEN yr = 2013 THEN sales ELSE 0 END) AS `2013`,
    SUM(CASE WHEN yr = 2014 THEN sales ELSE 0 END) AS `2014`,
    SUM(CASE WHEN yr = 2015 THEN sales ELSE 0 END) AS `2015`
FROM
    total_sales_cat
GROUP BY
    cat_name,
    prod_name
ORDER BY
    cat_name,
    prod_name;
```

| cat_name | prod_name | 2013 | 2014 | 2015 |
|---|---|---|---|---|
| Beverages | Chai | 86.4 | 277.2 | 288 |
| Beverages | Chang | 121.6 | 323 | 342 |
| Beverages | Chartreuse verte | 115.2 | 223.2 | 162 |
| Beverages | Côte de Blaye | 1054 | 2476.9 | 2371.5 |
| Beverages | Guarana Fantastica | 39.6 | 81.9 | 94.5 |

By Vikram Aiyer

# Q9 what about employees? Can we use Q7 and pull out records of employees who processed those orders?

```sql
-- Q9 what about employees? Can we use Q7 and pull out records of employees who processed those orders?
SELECT
  -- (orders.order_id) i,
  (customers.country) customer_country,
  (orders.customer_id) cust,
  (employees.employee_name) employee_nme,
  order_date,
  required_date,
  shipped_date,
  DATEDIFF(shipped_date, order_date) AS days_to_ship, -- days difference between order date and shipped date
  DATEDIFF(required_date, shipped_date) AS days_to_delivery, -- days difference between required date and shipped date
  DATEDIFF(shipped_date, order_date) / DATEDIFF(required_date, shipped_date) * 100 AS closure_rt
  FROM
  orders
LEFT JOIN
    order_details ON orders.order_id = order_details.order_id
INNER JOIN
    customers ON customers.customer_id = orders.customer_id
LEFT JOIN
    employees ON employees.employee_id = orders.employee_id
GROUP BY customer_country
ORDER BY closure_rt;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| customer_country | cust | employee_nme | order_date | required_date | shipped_date | days_to_ship | days_to_delivery | closure_rt |
|---|---|---|---|---|---|---|---|---|
| Belgium | MAISD | Steven Buchanan | 2014-05-07 | 2014-06-04 | 2014-05-09 | 2 | 26 | 7.6923 |
| Norway | SANTG | Nancy Davolio | 2013-12-18 | 2014-01-15 | 2013-12-20 | 2 | 26 | 7.6923 |
| Argentina | CACTU | Laura Callahan | 2014-04-29 | 2014-05-27 | 2014-05-02 | 3 | 25 | 12.0000 |
| Portugal | FURIB | Margaret Peacock | 2013-10-14 | 2013-11-11 | 2013-10-17 | 3 | 25 | 12.0000 |
| Venezuela | GROSR | Laura Callahan | 2013-07-30 | 2013-08-27 | 2013-08-02 | 3 | 25 | 12.0000 |
| Finland | WARTH | Janet Leverling | 2013-07-26 | 2013-09-06 | 2013-07-31 | 5 | 37 | 13.5135 |
| Sweden | BERGS | Laura Callahan | 2013-08-12 | 2013-09-09 | 2013-08-16 | 4 | 24 | 16.6667 |
| Spain | BOLID | Margaret Peacock | 2013-10-10 | 2013-11-07 | 2013-10-14 | 4 | 24 | 16.6667 |
| Canada | BOTTM | Margaret Peacock | 2013-12-20 | 2014-01-17 | 2013-12-24 | 4 | 24 | 16.6667 |
| Poland | WOLZA | Nancy Davolio | 2013-12-05 | 2014-01-02 | 2013-12-09 | 4 | 24 | 16.6667 |
| UK | AROUT | Michael Suyama | 2013-11-15 | 2013-12-13 | 2013-11-20 | 5 | 23 | 21.7391 |
| Mexico | ANATR | Robert King | 2013-09-18 | 2013-10-16 | 2013-09-24 | 6 | 22 | 27.2727 |

Result 7

# Q7 Do some Gap Analysis of orders to find out the TAT period and to which country the order got most delayed?

```sql
-- Q7  Do some Gap Analysis of orders to find out the TAT period and to which country the order got most delayed?
SELECT
  -- order_id,
  (customers.country) c,
  (orders.customer_id) cust,
  order_date,
  required_date,
  shipped_date,
  DATEDIFF(shipped_date, order_date) AS days_to_ship, -- days difference between order date and shipped date
  DATEDIFF(required_date, shipped_date) AS days_to_delivery, -- days difference between required date and shipped date
  DATEDIFF(shipped_date, order_date) / DATEDIFF(required_date, shipped_date) * 100 AS closure_rt
  FROM
  orders
LEFT JOIN
    order_details ON orders.order_id = order_details.order_id
INNER JOIN
    customers ON customers.customer_id = orders.customer_id
GROUP BY c
ORDER BY closure_rt;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| c | cust | order_date | required_date | shipped_date | days_to_ship | days_to_delivery | closure_rt |
|---|---|---|---|---|---|---|---|
| USA | GREAL | 2014-05-06 | 2014-05-20 | 2014-05-09 | 3 | 11 | 27.2727 |
| Ireland | HUNGO | 2013-09-05 | 2013-10-03 | 2013-09-11 | 6 | 22 | 27.2727 |
| Brazil | COMMI | 2013-08-27 | 2013-09-24 | 2013-09-03 | 7 | 21 | 33.3333 |
| Denmark | SIMOB | 2013-10-29 | 2013-11-26 | 2013-11-05 | 7 | 21 | 33.3333 |
| Germany | ALFKI | 2014-08-25 | 2014-09-22 | 2014-09-02 | 8 | 20 | 40.0000 |
| Italy | FRANS | 2014-01-22 | 2014-02-19 | 2014-01-31 | 9 | 19 | 47.3684 |
| Switzerland | CHOPS | 2013-07-11 | 2013-08-08 | 2013-07-23 | 12 | 16 | 75.0000 |
| France | BLONP | 2013-07-25 | 2013-08-22 | 2013-08-12 | 18 | 10 | 180.0000 |

# Q8 Let us do some deep dive on products which are discontinued but has some potential among the customers by orders

```sql
-- Q8 Let us do some deep dive on products which are discontinued but has some potential among the customers by orders
SELECT
-- product_id p,
(categories.category_name) AS cat_name,
product_name,
-- category_id,
COUNT(order_details.order_id)  AS total_orders,
ROUND(SUM(order_details.unit_price) - SUM(order_details.discount)) AS total_sales,
-- discontinued AS NA
CASE WHEN products.discontinued = 1 THEN 'Yes' ELSE NULL END AS discontinued_products
FROM
products
LEFT JOIN order_details ON order_details.product_id = products.product_id
RIGHT OUTER JOIN categories ON categories.category_id = products.category_id
INNER JOIN orders ON orders.order_id = order_details.order_id
WHERE discontinued = 1
GROUP BY product_name
ORDER BY total_sales DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| cat_name | product_name | total_orders | total_sales | discontinued_products |
|---|---|---|---|---|
| Meat & Poultry | Thüringer Rostbratwurst | 32 | 3713 | Yes |
| Produce | Rössle Sauerkraut | 33 | 1385 | Yes |
| Meat & Poultry | Alice Mutton | 37 | 1349 | Yes |
| Meat & Poultry | Perth Pasties | 30 | 905 | Yes |
| Meat & Poultry | Mishi Kobe Niku | 5 | 466 | Yes |
| Grains & Cereals | Singaporean Hokkien Fried Mee | 30 | 396 | Yes |
| Beverages | Guarana Fantastica | 51 | 216 | Yes |
| Condiments | Chef Anton's Gumbo Mix | 10 | 196 | Yes |

Result 29

# Q 10 Let us find which categories got us the most revenue and from which country?

```sql
-- Q 10 Let us find which categories got us the most revenue and from which country?
SELECT DISTINCT
-- (o.customer_id) AS x,
(c.country) Customer_country,
(p.product_name) Name_of_product,
(ct.category_name) Category_Product,
COUNT(o.order_id) AS Total_orders,
ROUND(SUM(od.unit_price) - SUM(od.discount), 2) As Total_sales_After_Discount
FROM
orders o
INNER JOIN customers c ON o.customer_id = c.customer_id
INNER JOIN order_details od ON od.order_id = o.order_id
INNER JOIN products p ON od.product_id = p.product_id
INNER JOIN categories ct ON ct.category_id = p.category_id

GROUP BY Customer_country
ORDER BY Total_sales_After_Discount DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| Customer_country | Name_of_product | Category_Product | Total_orders | Total_sales_After_Discount |
|---|---|---|---|---|
| USA | Chai | Beverages | 352 | 10462.91 |
| Germany | Chai | Beverages | 328 | 8544.84 |
| Brazil | Chai | Beverages | 203 | 5324.64 |
| France | Chai | Beverages | 184 | 4839.46 |
| Austria | Chang | Beverages | 125 | 3469.95 |

Result 23

# Q 11 Create a stored procedure on the basis of Q9

```
229    -- Q 11 Create a stored procedure on the basis of Q9
230    DELIMITER //
231 •  CREATE PROCEDURE order_processed()
232    BEGIN
233        SELECT
234      -- (orders.order_id) i,
235      (customers.country) customer_country,
236      (orders.customer_id) cust,
237      (employees.employee_name) employee_nme,
238      order_date,
239      required_date,
240      shipped_date,
241      DATEDIFF(shipped_date, order_date) AS days_to_ship, -- days difference between order date and shipped date
242      DATEDIFF(required_date, shipped_date) AS days_to_delivery, -- days difference between required date and shipped date
243      DATEDIFF(shipped_date, order_date) / DATEDIFF(required_date, shipped_date) * 100 AS closure_rt
244      FROM
245      orders
246    LEFT JOIN
247        order_details ON orders.order_id = order_details.order_id
248    INNER JOIN
249        customers ON customers.customer_id = orders.customer_id
250    LEFT JOIN
251        employees ON employees.employee_id = orders.employee_id
252    GROUP BY customer_country
253    ORDER BY closure_rt;
254        END //
255    DELIMITER ;
256 •  CALL order_processed;
257
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| customer_country | cust | employee_nme | order_date | required_date | shipped_date | days_to_ship | days_to_delivery | closure_rt |
|---|---|---|---|---|---|---|---|---|
| Belgium | MAISD | Steven Buchanan | 2014-05-07 | 2014-06-04 | 2014-05-09 | 2 | 26 | 7.6923 |
| Norway | SANTG | Nancy Davolio | 2013-12-18 | 2014-01-15 | 2013-12-20 | 2 | 26 | 7.6923 |
| Argentina | CACTU | Laura Callahan | 2014-04-29 | 2014-05-27 | 2014-05-02 | 3 | 25 | 12.0000 |
| Portugal | FURIB | Margaret Peacock | 2013-10-14 | 2013-11-11 | 2013-10-17 | 3 | 25 | 12.0000 |
| Venezuela | GROSR | Laura Callahan | 2013-07-30 | 2013-08-27 | 2013-08-02 | 3 | 25 | 12.0000 |

Result 1