1)

```
1  SELECT *
2      FROM nwCategories
3  /*
4  Selecting all columns from table nwCategories
5  */
6
```

Result Grid | Filter Rows: | Edit: | Export/Import:

#	CategoryID	CategoryName	Description
1	1	Beverages	Soft drinks, coffees, teas, beers, an…
2	2	Condiments	Sweet and savory sauces, relishes, …
3	3	Confections	Desserts, candies, and sweet breads
4	4	Dairy Products	Cheeses
5	5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	6	Meat/Poultry	Prepared meats
7	7	Produce	Dried fruit and bean curd
8	8	Seafood	Seaweed and fish
*	NULL	NULL	NULL

2)

```sql
SELECT CompanyName, Address, Region, PostalCode, Phone
    FROM nwSuppliers
        WHERE nwSuppliers.Country = 'France' OR nwSuppliers.Country = 'Germany'
/*
Selecting all CompanyName, Address, Region, PostalCode, Phone columns from nwSuppliers
from France or germany
*/
```

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

#	CompanyName	Address	Region	PostalCode	Phone
1	Heli Süßwaren GmbH & Co. KG	Tiergartenstraße 5		10785	(010) 9984510
2	Plutzer Lebensmittelgroßmärkte AG	Bogenallee 51		60439	(069) 992755
3	Nord-Ost-Fisch Handelsgesellschaf…	Frahmredder 112a		27478	(04721) 8713
4	Aux joyeux ecclésiastiques	203 Rue des Francs-Bourgeois		75004	(1) 03.83.00.68
5	Escargots Nouveaux	22 rue H. Voiron		71300	85.57.00.07
6	Gai pâturage	Bat. B 3 rue des Alpes		74000	38.76.98.06

3)

```
1  SELECT ContactName, ContactTitle
2      FROM nwSuppliers
3          WHERE nwSuppliers.SupplierID >= 5 AND nwSuppliers.SupplierID <= 20
4  /*
5  Selecting the contact name and contact title from nwSuppliers
6  having a SupplierID between 5 and 20
7  */
8
```

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

#	CompanyName	Address	Region	PostalCode	Phone
1	Heli Süßwaren GmbH & Co. KG	Tiergartenstraße 5		10785	(010) 9984510
2	Plutzer Lebensmittelgroßmärkte AG	Bogenallee 51		60439	(069) 992755
3	Nord-Ost-Fisch Handelsgesellschaf…	Frahmredder 112a		27478	(04721) 8713
4	Aux joyeux ecclésiastiques	203 Rue des Francs-Bourgeois		75004	(1) 03.83.00.68
5	Escargots Nouveaux	22 rue H. Voiron		71300	85.57.00.07
6	Gai pâturage	Bat. B 3 rue des Alpes		74000	38.76.98.06

4)

```
1  SELECT ProductName, QuantityPerUnit, UnitPrice, UnitsInStock
2      FROM nwProducts
3          WHERE UnitPrice<10;
4  /*
5  Displaying the product name, quantity per unit, unit price and how many units are in stock
6  for all units form the products table with a price greater thank 10
7  */
8
```

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

#	ProductName	QuantityPerUnit	UnitPrice	UnitsInStock
1	Konbu	2 kg box	6.00	24
2	Teatime Chocolate Biscuits	10 boxes x 12 pie…	9.20	25
3	Tunnbröd	12 - 250 g pkgs.	9.00	61
4	Guaraná Fantástica	12 - 355 ml cans	4.50	20
5	Geitost	500 g	2.50	112
6	Jack's New England Clam Cho…	12 - 12 oz cans	9.65	85
7	Rogede sild	1k pkg.	9.50	5
8	Zaanse koeken	10 - 4 oz boxes	9.50	36
9	Filo Mix	16 - 2 kg boxes	7.00	38
10	Tourtière	16 pies	7.45	21
11	Rhönbräu Klosterbier	24 - 0.5 l bottles	7.75	125

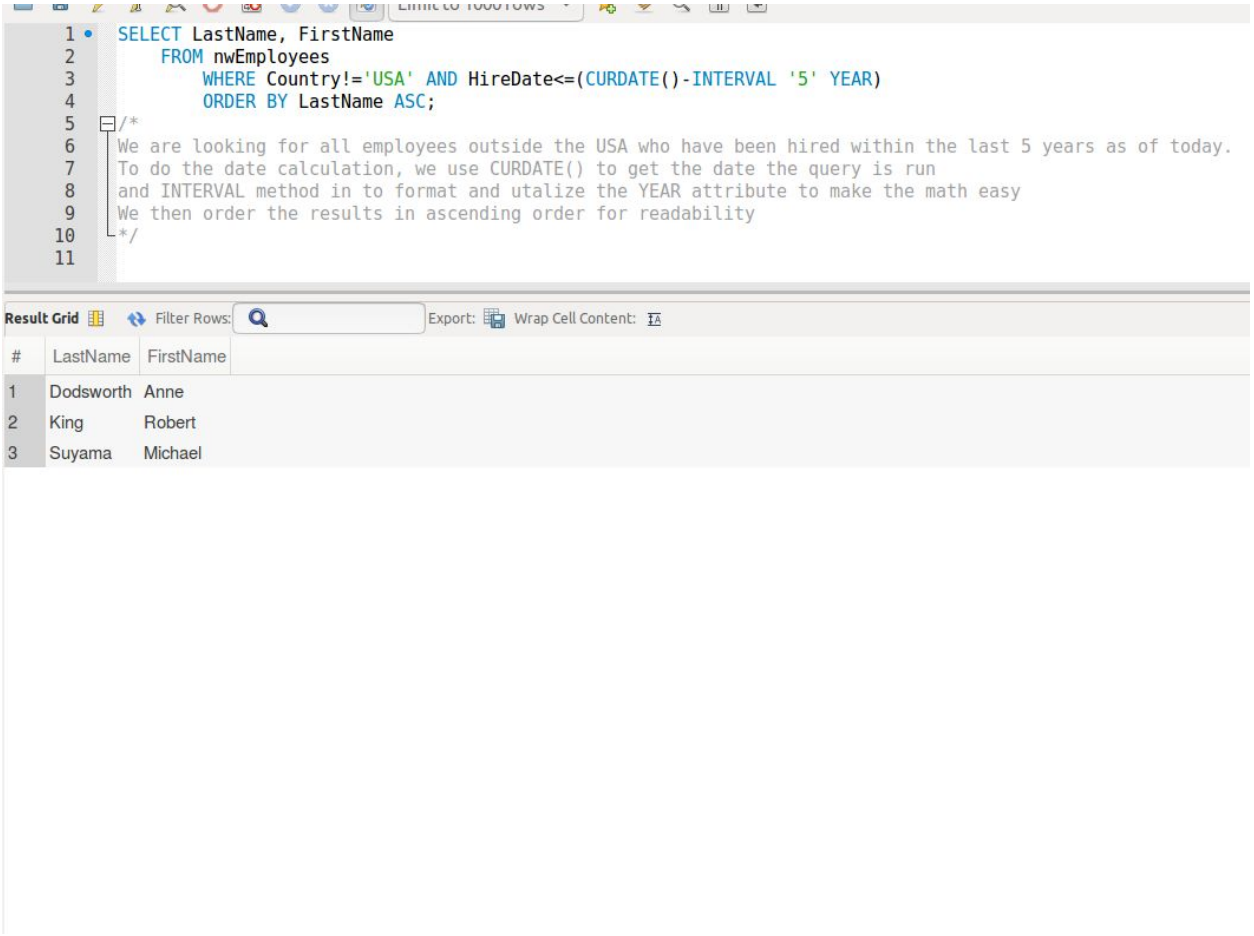
5)

```sql
SELECT ProductID,ProductName, UnitPrice, UnitsInStock
    FROM nwProducts
        WHERE UnitsInStock>=1 AND UnitsInStock <= ReorderLevel;
/*
Displaying the product name, product ID, unit price and how many units are in stock
for all units form the products table where there are more then 1 in stock
and less then the reorder level for that product
*/
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

#	ProductID	ProductName	UnitPrice	UnitsInStock
1	2	Chang	19.00	17
2	3	Aniseed Syrup	10.00	13
3	11	Queso Cabrales	21.00	22
4	21	Sir Rodney's Scones	10.00	3
5	30	Nord-Ost Matjeshering	25.89	10
6	32	Mascarpone Fabioli	32.00	9
7	37	Gravad lax	26.00	11
8	43	Ipoh Coffee	46.00	17
9	45	Rogede sild	9.50	5
10	48	Chocolade	12.75	15
11	49	Maxilaku	20.00	10
12	56	Gnocchi di nonna Alice	38.00	21
13	64	Wimmers gute Semmelknödel	33.25	22
14	66	Louisiana Hot Spiced Okra	17.00	4
15	68	Scottish Longbreads	12.50	6
16	70	Outback Lager	15.00	15
17	74	Longlife Tofu	10.00	4
*	NULL	NULL	NULL	NULL

6)

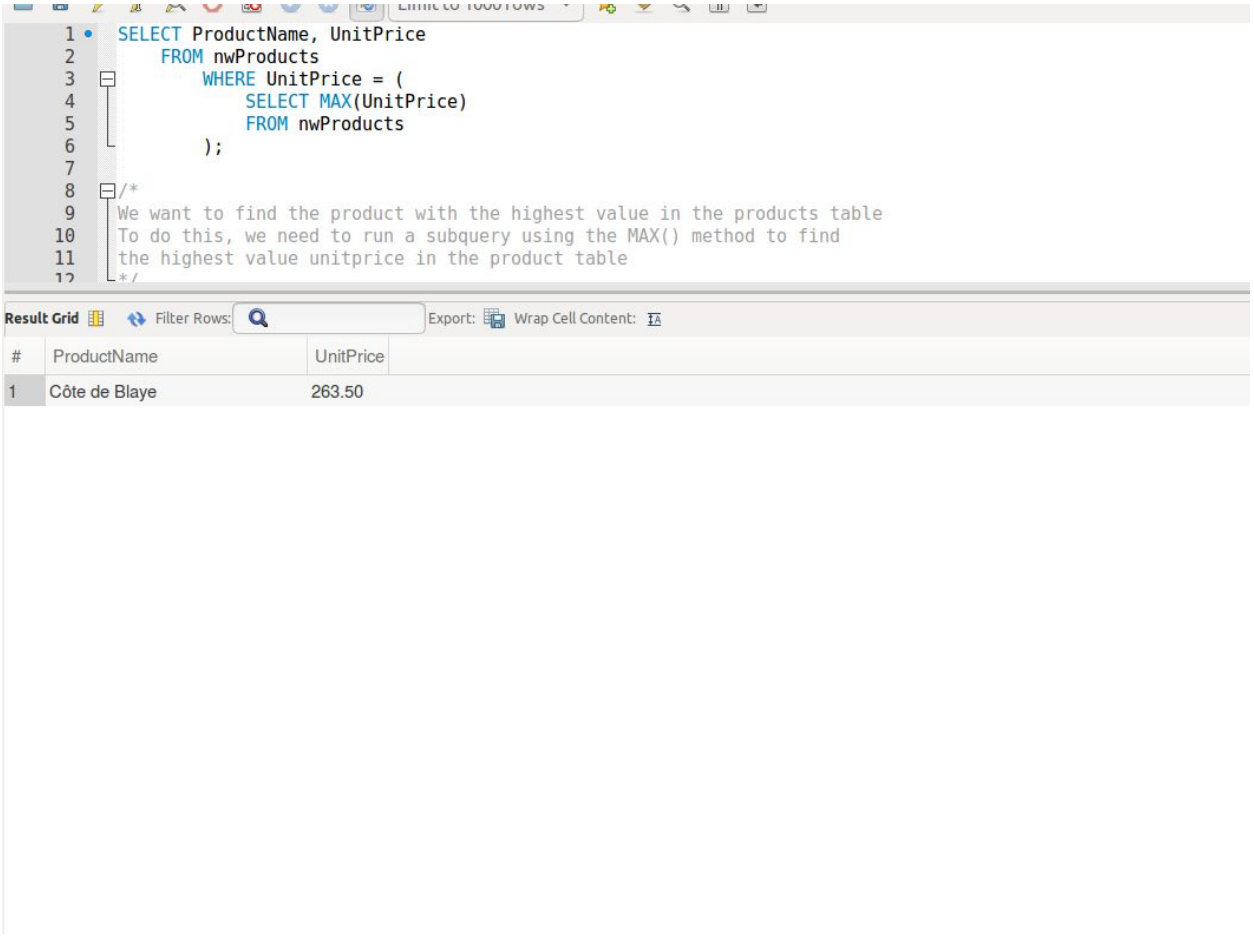


```
SELECT LastName, FirstName
    FROM nwEmployees
        WHERE Country!='USA' AND HireDate<=(CURDATE()-INTERVAL '5' YEAR)
        ORDER BY LastName ASC;
/*
We are looking for all employees outside the USA who have been hired within the last 5 years as of today.
To do the date calculation, we use CURDATE() to get the date the query is run
and INTERVAL method in to format and utalize the YEAR attribute to make the math easy
We then order the results in ascending order for readability
*/
```

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

#	LastName	FirstName
1	Dodsworth	Anne
2	King	Robert
3	Suyama	Michael

7)



```
SELECT ProductName, UnitPrice
    FROM nwProducts
        WHERE UnitPrice = (
            SELECT MAX(UnitPrice)
            FROM nwProducts
        );

/*
We want to find the product with the highest value in the products table
To do this, we need to run a subquery using the MAX() method to find
the highest value unitprice in the product table
*/
```

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

#	ProductName	UnitPrice
1	Côte de Blaye	263.50

8)

```sql
SELECT ProductID, ProductName, (UnitsInStock * UnitPrice) AS TotalInventoryValue
    FROM nwProducts
        HAVING TotalInventoryValue > 2000.00
        ORDER BY TotalInventoryValue DESC;

/*
We create an ALIAS Column for the total inventory value by doing the calcualtion
in the SELECT statement
We order the values in descending order so we can see the products
with the highest total value first
*/
```

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

#	ProductID	ProductName	TotalInventoryValue
1	38	Côte de Blaye	4479.50
2	59	Raclette Courdavault	4345.00
3	12	Queso Manchego La Pastora	3268.00
4	20	Sir Rodney's Marmalade	3240.00
5	61	Sirop d'érable	3220.50
6	6	Grandma's Boysenberry Spread	3000.00
7	9	Mishi Kobe Niku	2813.00
8	55	Pâté chinois	2760.00
9	18	Carnarvon Tigers	2625.00
10	40	Boston Crab Meat	2263.20
11	22	Gustaf's Knäckebröd	2184.00
12	27	Schoggi Schokolade	2151.10
13	36	Inlagd Sill	2128.00

9)

```
1  SELECT *
2      FROM nwProducts
3          WHERE (Discontinued = 1 AND QuantityPerUnit LIKE '%can%');
4
5  /*
6  To look for discontinued items, we need to know that nwProducts.Discontinued is a bool
7  value with 0 meaning NOT DISCONTINUED and 1 meaning that YES DISCONTINUED
8
9  We can search for a string literal in QuantityPerUnit column by using LIKE
10 Side note, why are we putting text in QuantityPerUnit?
11 */
12
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

#	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinu
1	24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.50	20	0	0	1
2	28	Rössle Sauerkraut	12	7	25 - 825 g cans	45.60	26	0	0	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

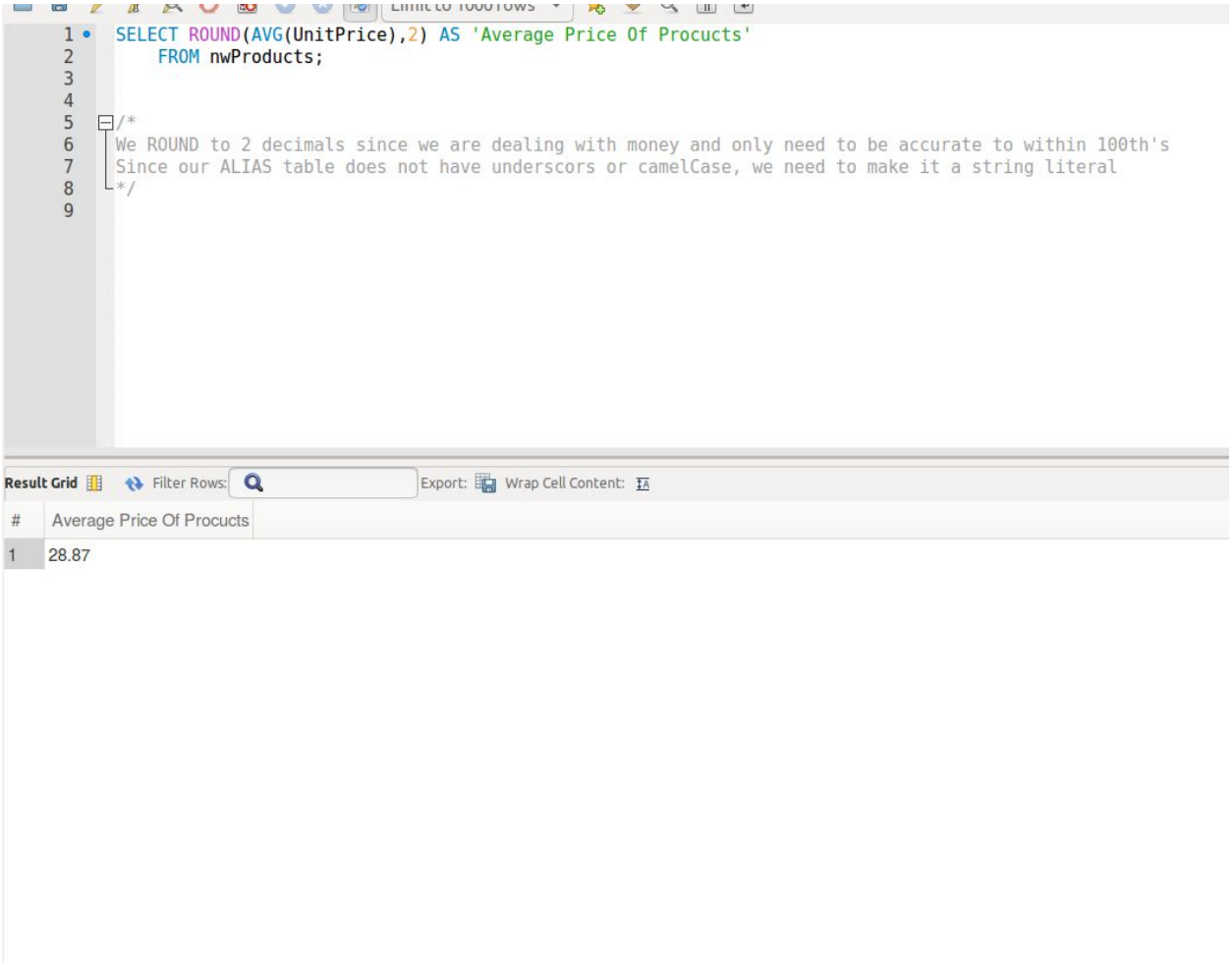
10)

```sql
SELECT DISTINCT(ShipCountry), COUNT(OrderID) AS Orders
    FROM nwOrders
        WHERE ShipCountry!='USA' AND ShippedDate BETWEEN '2013-09-01'AND '2013-09-31'
        GROUP BY ShipCountry ASC;

/*
We filter all duplicate countries by using the DISTINCT() method
and use a simple BETWEEN (string value date 1) AND (string value date 2)
to find all queries between that date range
*/
```

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

#	ShipCountry	Orders
1	Brazil	4
2	France	3
3	Germany	1
4	Ireland	1
5	Italy	2
6	Mexico	3
7	Spain	2
8	Sweden	1
9	Venezuela	1

11)

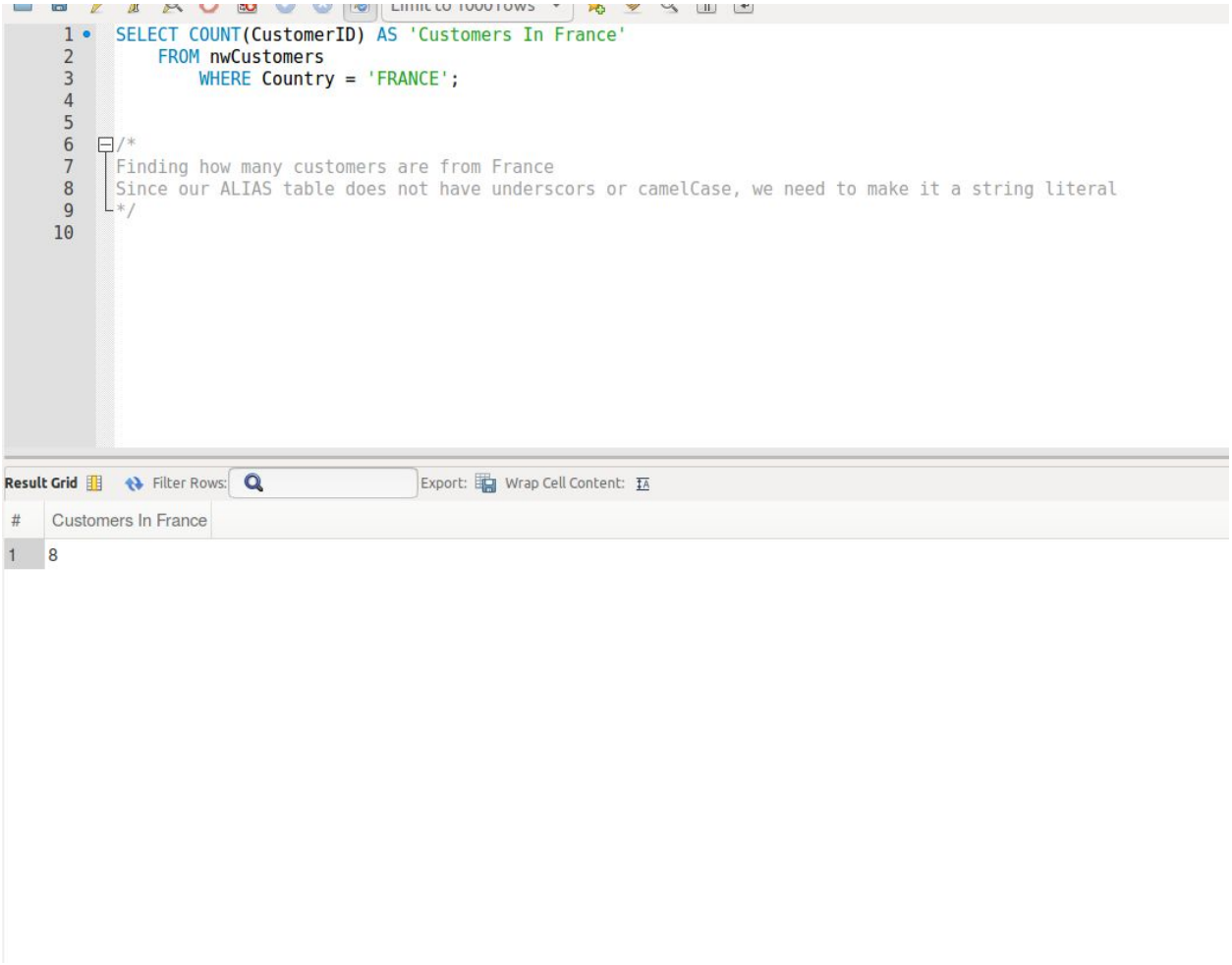


```
SELECT ROUND(AVG(UnitPrice),2) AS 'Average Price Of Procucts'
    FROM nwProducts;


/*
We ROUND to 2 decimals since we are dealing with money and only need to be accurate to within 100th's
Since our ALIAS table does not have underscors or camelCase, we need to make it a string literal
*/
```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	Average Price Of Procucts
1	28.87

12)



```
SELECT COUNT(CustomerID) AS 'Customers In France'
    FROM nwCustomers
        WHERE Country = 'FRANCE';


/*
Finding how many customers are from France
Since our ALIAS table does not have underscors or camelCase, we need to make it a string literal
*/
```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	Customers In France
1	8

13)

```
1  SELECT nwCustomers.CustomerID, nwCustomers.CompanyName
2      FROM nwCustomers JOIN nwOrders
3          ON nwCustomers.CustomerID = nwOrders.CustomerID
4      GROUP BY nwOrders.CustomerID
5      HAVING COUNT(nwOrders.CustomerID)>20;
6
7
8  /*
9  Since we are doing an explicit JOIN, we need to distinguish which talb each attribute
10 came from by writing the full nwTablename.DesiredAttribute
11 */
12
```

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

#	CustomerID	CompanyName
1	ERNSH	Ernst Handel
2	QUICK	QUICK-Stop
3	SAVEA	Save-a-lot Markets

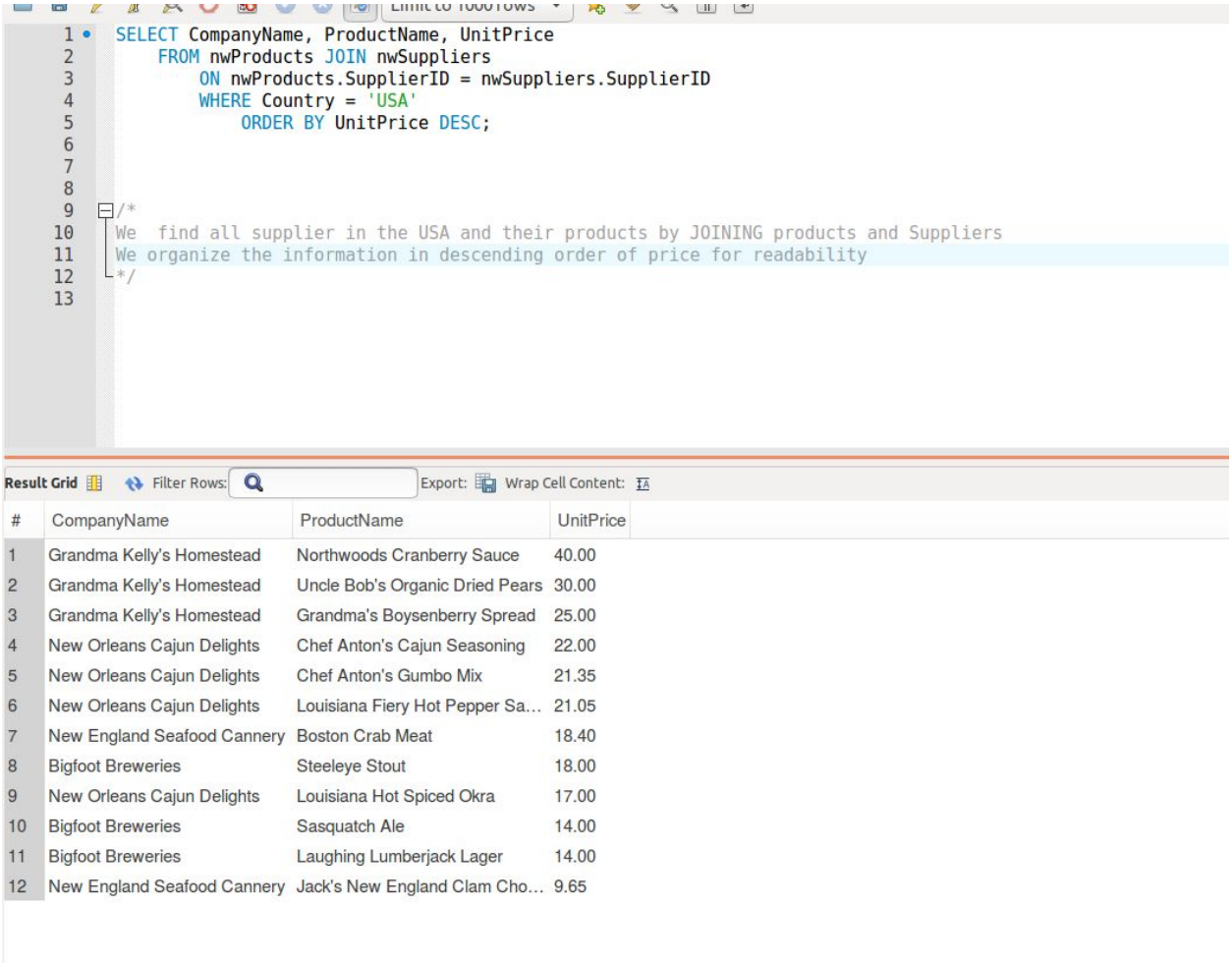
14)

```sql
SELECT SupplierID, SUM(UnitsInStock*UnitPrice) AS ValueOfInventory
    FROM nwProducts
        GROUP BY SupplierID
        HAVING COUNT(SupplierID)>3;


/*
We can calculate the inventory value by multiplying how many units are
in stock times the unit price and sum all those columns together in a single step
*/

```

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

#	CustomerID	CompanyName
1	ERNSH	Ernst Handel
2	QUICK	QUICK-Stop
3	SAVEA	Save-a-lot Markets

15)



```
SELECT CompanyName, ProductName, UnitPrice
    FROM nwProducts JOIN nwSuppliers
        ON nwProducts.SupplierID = nwSuppliers.SupplierID
        WHERE Country = 'USA'
            ORDER BY UnitPrice DESC;



/*
We  find all supplier in the USA and their products by JOINING products and Suppliers
We organize the information in descending order of price for readability
*/
```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	CompanyName	ProductName	UnitPrice
1	Grandma Kelly's Homestead	Northwoods Cranberry Sauce	40.00
2	Grandma Kelly's Homestead	Uncle Bob's Organic Dried Pears	30.00
3	Grandma Kelly's Homestead	Grandma's Boysenberry Spread	25.00
4	New Orleans Cajun Delights	Chef Anton's Cajun Seasoning	22.00
5	New Orleans Cajun Delights	Chef Anton's Gumbo Mix	21.35
6	New Orleans Cajun Delights	Louisiana Fiery Hot Pepper Sa…	21.05
7	New England Seafood Cannery	Boston Crab Meat	18.40
8	Bigfoot Breweries	Steeleye Stout	18.00
9	New Orleans Cajun Delights	Louisiana Hot Spiced Okra	17.00
10	Bigfoot Breweries	Sasquatch Ale	14.00
11	Bigfoot Breweries	Laughing Lumberjack Lager	14.00
12	New England Seafood Cannery	Jack's New England Clam Cho…	9.65

16)

```sql
SELECT LastName, FirstName, Title, Extension, COUNT(nwOrders.EmployeeID) AS 'Number of Orders'
    FROM nwEmployees JOIN nwOrders
        ON nwEmployees.EmployeeID = nwOrders.EmployeeID
        GROUP BY nwEmployees.EmployeeID
        HAVING COUNT(nwOrders.EmployeeID)>100;



/*
To find all employees plus their information, who have more then 100 orders
we join our employee and orders tables and distinguish which table the information
is coming from, we prefix it with its destination table name
*/

```

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

#	CompanyName	ProductName	UnitPrice
1	Grandma Kelly's Homestead	Northwoods Cranberry Sauce	40.00
2	Grandma Kelly's Homestead	Uncle Bob's Organic Dried Pears	30.00
3	Grandma Kelly's Homestead	Grandma's Boysenberry Spread	25.00
4	New Orleans Cajun Delights	Chef Anton's Cajun Seasoning	22.00
5	New Orleans Cajun Delights	Chef Anton's Gumbo Mix	21.35
6	New Orleans Cajun Delights	Louisiana Fiery Hot Pepper Sa…	21.05
7	New England Seafood Cannery	Boston Crab Meat	18.40
8	Bigfoot Breweries	Steeleye Stout	18.00
9	New Orleans Cajun Delights	Louisiana Hot Spiced Okra	17.00
10	Bigfoot Breweries	Sasquatch Ale	14.00
11	Bigfoot Breweries	Laughing Lumberjack Lager	14.00
12	New England Seafood Cannery	Jack's New England Clam Cho…	9.65

17)

```
SELECT DISTINCT nwCustomers.CustomerID, nwCustomers.CompanyName
    FROM nwOrders RIGHT OUTER JOIN nwCustomers
        ON nwOrders.CustomerID = nwCustomers.CustomerID
        WHERE nwOrders.CustomerID IS NULL;



/*
Since we do not need ALL the information from both tables, we
can do a RIGHT OUTER JOIN on the Orders and Customer tables to get
the information we need
*/
```

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

#	CustomerID	CompanyName
1	FISSA	FISSA Fabrica Inter. Salchichas S.A.
2	PARIS	Paris specialites

Result 14

Action Output

18)

```sql
/*Create an OUT OF STOCK LIST showing the Supplier CompanyName, Supplier ContactName,
Product CategoryName, CategoryDescription, ProductName and UnitsOnOrder
for all products that are out of stock (UnitsInStock = 0). (5)*/
SELECT  nwSuppliers.CompanyName, nwSuppliers.ContactName, nwCategories.CategoryName, nwCategories.Description,
        nwProducts.ProductName, nwProducts.UnitsOnOrder
    FROM nwProducts, nwSuppliers, nwCategories
        WHERE (nwProducts.CategoryID = nwCategories.CategoryID)
            AND (nwSuppliers.SupplierID = nwProducts.SupplierID)
```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	CompanyName	ContactName	CategoryName	Description	ProductName	UnitsOnOrder
1	New Orleans Cajun Delights	Shelley Burke	Condiments	Sweet and savory sauces, relishes, ...	Chef Anton's Gumbo Mix	0
2	Pavlova Ltd.	Ian Devling	Meat/Poultry	Prepared meats	Alice Mutton	0
3	Plutzer Lebensmittelgroßmärkte AG	Martin Bein	Meat/Poultry	Prepared meats	Thüringer Rostbratwurst	0
4	Formaggi Fortini s.r.l.	Elio Rossi	Dairy Products	Cheeses	Gorgonzola Telino	70
5	G'day Mate	Wendy Mackenzie	Meat/Poultry	Prepared meats	Perth Pasties	0

19)

```sql
SELECT nwProducts.ProductName, nwSuppliers.CompanyName AS 'Supplier Name', nwSuppliers.Country, nwProducts.UnitsInStock
    FROM nwProducts, nwSuppliers
        WHERE (nwProducts.QuantityPerUnit LIKE('%bottle%'))
        AND nwProducts.SupplierID = nwSuppliers.SupplierID;



/*
Since we are trying to locate items based on a string again, we need to use the LIKE method again
Here, we are doing an implicit JOIN by simply stating the two tables we are selecting from without
actually explicitly joining them on their key
*/
```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	CustomerID	CompanyName
1	FISSA	FISSA Fabrica Inter. Salchichas S.A.
2	PARIS	Paris specialites

Result 14

Action Output

20)

```sql
SELECT nwCustomers.CompanyName, nwCustomers.Country, CAST(SUM((nwOrderDetails.UnitPrice * nwOrderDetails.Quantity) -
       nwOrderDetails.Discount*nwOrderDetails.UnitPrice*nwOrderDetails.Quantity) AS DECIMAL(9,2)) AS Value
    FROM nwCustomers JOIN nwOrders
        ON nwOrders.CustomerID = nwCustomers.CustomerID
        JOIN nwOrderDetails
            ON nwOrderDetails.OrderID = nwOrders.OrderID
            GROUP BY(nwCustomers.CustomerID)
            HAVING(Value)>30000
            ORDER BY(Value) DESC;


/*
Here we need to be very carerful since there is a lot going on
We ahve a double JOIN (indented for readability)
and we calcualte the full price of the products in our select statement by
summing the unitprice times the quantity of the order less the FULL numeric discount
value. We then sort on a dollar value constraint and order in descending order for readability
```

Result Grid Filter Rows: Export: Wrap Cell Content:

#	CompanyName	Country	Value
1	QUICK-Stop	Germany	110277.31
2	Ernst Handel	Austria	104874.98
3	Save-a-lot Markets	USA	104361.95
4	Rattlesnake Canyon Grocery	USA	51097.80
5	Hungry Owl All-Night Grocers	Ireland	49979.91
6	Hanari Carnes	Brazil	32841.37
7	Koniglich Essen	Germany	30908.38

Result 15

Action Output

21)

```sql
/*Create an "Employees Orders" VIEW listing employee lastname, firstname, and the total
count of each employee's orders. (No answer set needed.)

Run a query against the Employee Orders view listing the employees and their order counts in
order from largest to smallest number of orders. (9 rows, one for each employee.)*/
CREATE OR REPLACE VIEW Employees_Orders AS
SELECT nwEmployees.LastName, nwEmployees.FirstName, COUNT(nwOrders.EmployeeID) AS 'Number Of Orders'
    FROM nwEmployees JOIN nwOrders
    ON nwEmployees.EmployeeID = nwOrders.EmployeeID
    GROUP BY nwEmployees.EmployeeID;
```

22)

```sql
SELECT LastName, FirstName, Number_Of_Orders
    FROM Employees_Orders
    ORDER BY(Number_Of_Orders)DESC;

/*
find the number of orders each employee has placed and order in descending order
*/
```

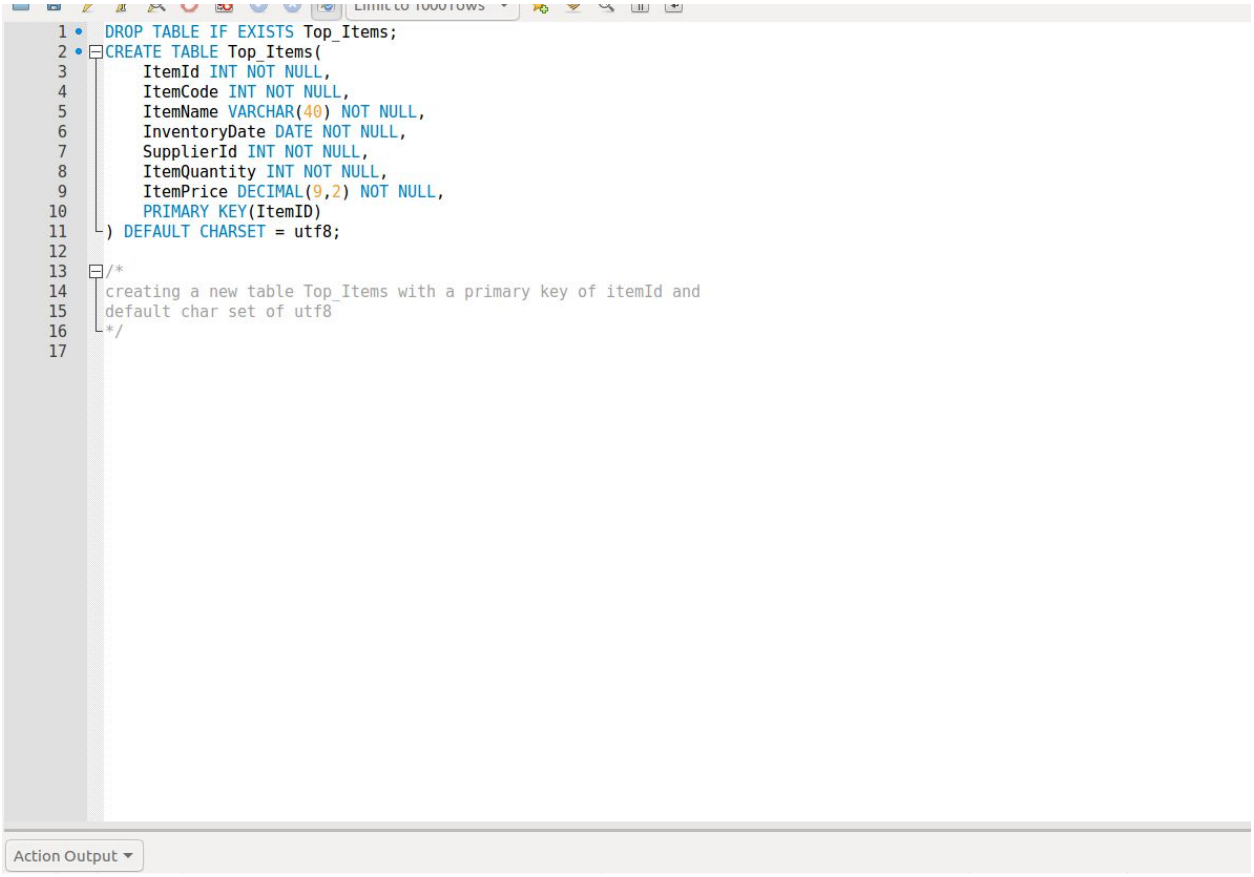
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	LastName	FirstName	Number_Of_Orders
1	Peacock	Margaret	156
2	Leverling	Janet	127
3	Davolio	Nancy	123
4	Callahan	Laura	104
5	Fuller	Andrew	96
6	King	Robert	72
7	Suyama	Michael	67
8	Dodsworth	Anne	43
9	Buchanan	Steven	42

Employees_Orders 16

Action Output

23)


DROP TABLE IF EXISTS Top_Items;
CREATE TABLE Top_Items(
ItemId INT NOT NULL,
ItemCode INT NOT NULL,
ItemName VARCHAR(40) NOT NULL,
InventoryDate DATE NOT NULL,
SupplierId INT NOT NULL,
ItemQuantity INT NOT NULL,
ItemPrice DECIMAL(9,2) NOT NULL,
PRIMARY KEY(ItemID)
) DEFAULT CHARSET = utf8;
/*
creating a new table Top_Items with a primary key of itemId and
default char set of utf8
*/
Action Output

24)

```
INSERT INTO Top_Items(ItemID, ItemCode, ItemName, InventoryDate, ItemQuantity, ItemPrice, SupplierId)
    SELECT nwProducts.ProductID, nwProducts.ProductID, nwProducts.ProductName, CURDATE(), nwProducts.UnitsInStock,
           nwProducts.UnitPrice, nwProducts.SupplierID
    FROM nwProducts
        HAVING(nwProducts.UnitPrice*nwProducts.UnitsInStock)>2500;

    SELECT *
        FROM Top_Items;

/*
Populating the new Top_Items table with items their
corresponding information from our Products table with an inventory value (Unitprice * units in stock)
more then $2500
*/
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

#	ItemId	ItemCode	ItemName	InventoryDate	SupplierId	ItemQuantity	ItemPrice
1	6	6	Grandma's Boysenberry Spread	2018-03-11	3	120	25.00
2	9	9	Mishi Kobe Niku	2018-03-11	4	29	97.00
3	12	12	Queso Manchego La Pastora	2018-03-11	5	86	38.00
4	18	18	Carnarvon Tigers	2018-03-11	7	42	62.50
5	20	20	Sir Rodney's Marmalade	2018-03-11	8	40	81.00
6	38	38	Côte de Blaye	2018-03-11	18	17	263.50
7	55	55	Pâté chinois	2018-03-11	25	115	24.00
8	59	59	Raclette Courdavault	2018-03-11	28	79	55.00
9	61	61	Sirop d'érable	2018-03-11	29	113	28.50
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Top_Items 17

Action Output

25)

```sql
SET SQL_SAFE_UPDATES = 0;
DELETE FROM Top_Items
    WHERE Top_Items.SupplierId = ANY(
        SELECT nwSuppliers.SupplierID
            FROM nwSuppliers
            WHERE nwSuppliers.Country = 'Canada'
    );
/*
SET SQL_SAFE_UPDATES = 0; is used to allow me to delete from a table without querying on a primary
key
other then that, we perform a subquery to find suppliers from Canada and remove them from our Top_Items table
based on their SupplierId
*/
```

Action Output

26)

```sql
/* Add a new column to the Top_Items table called InventoryValue ((decimal (9,2))) after the
inventory date. No answer set needed.
*/
ALTER TABLE Top_Items
    ADD COLUMN InventoryValue decimal (9,2) AFTER InventoryDate;


SELECT *
    FROM Top_Items;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

#	ItemID	ItemCode	ItemName	InventoryDate	InventoryValue	SupplierID	ItemQuantity	ItemPrice
1	6	6	Grandma's Boysenberry Spread	2018-03-03	NULL	3	120	25.00

Top_Items 24

27)

```sql
SET SQL_SAFE_UPDATES = 0;
UPDATE Top_Items
    SET InventoryValue = (ItemPrice*ItemQuantity);

SELECT *
    FROM Top_Items;
/*
SET SQL_SAFE_UPDATES = 0; is used to allow me to update from a table without querying on a primary
key
*/
```

Action Output

28)

```sql
SELECT *
    FROM Top_Items;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

#	ItemID	ItemCode	ItemName	InventoryDate	InventoryValue	SupplierID	ItemQuantity	ItemPrice
1	6	6	Grandma's Boysenberry Spread	2018-03-03	3000.00	3	120	25.00
2	9	9	Mishi Kobe Niku	2018-03-03	2813.00	4	29	97.00
3	12	12	Queso Manchego La Pastora	2018-03-03	3268.00	5	86	38.00
4	18	18	Carnarvon Tigers	2018-03-03	2625.00	7	42	62.50
5	20	20	Sir Rodney's Marmalade	2018-03-03	3240.00	8	40	81.00
6	38	38	Côte de Blaye	2018-03-03	4479.50	18	17	263.50
7	59	59	Raclette Courdavault	2018-03-03	4345.00	28	79	55.00
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

29)

```sql
DROP TABLE Top_items;
```