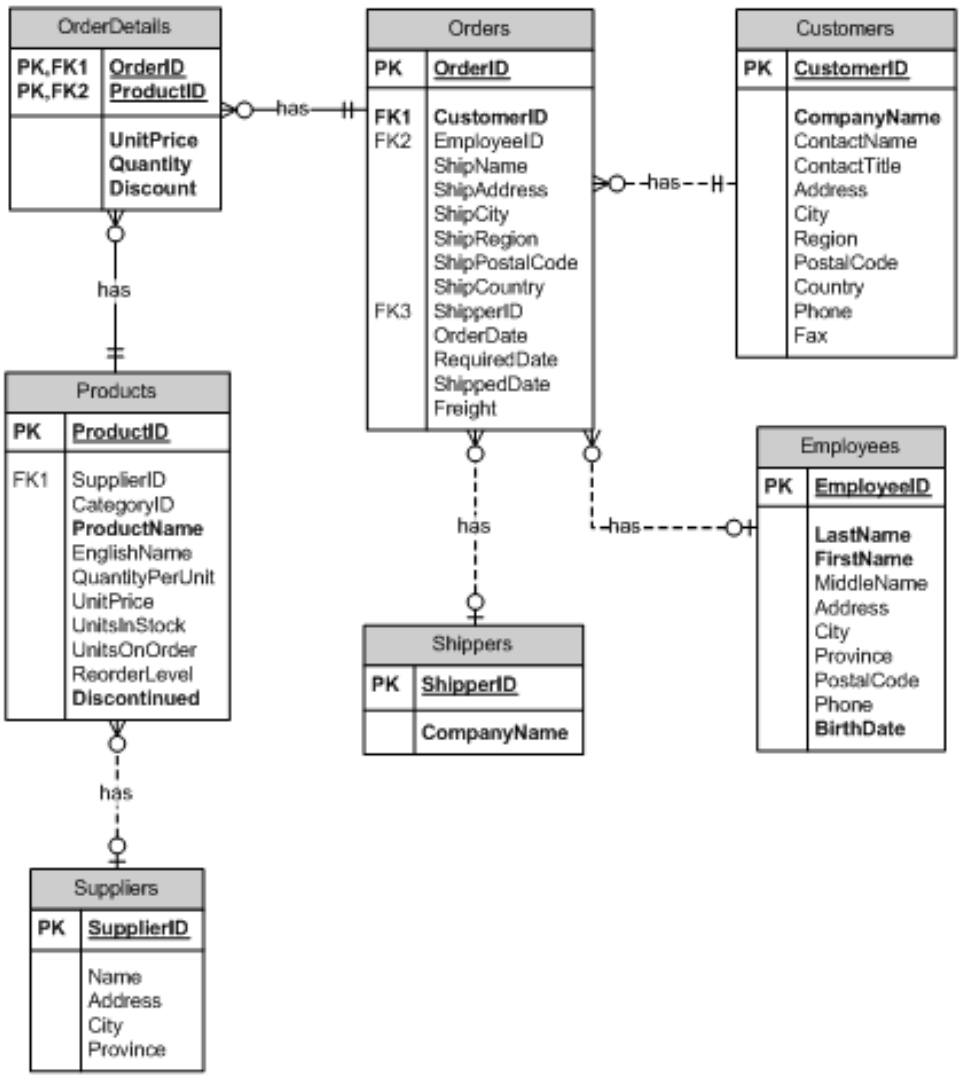
COMP 1630
Relational Database Design and SQL

SQL Project

Using Microsoft SQL Server, write the SQL statements for each question necessary to generate the required result set. Clearly identify your answers to the questions. Save your work in the LearningHub **Assignments** for **SQL Project**.

Entity Relationship Diagram



**Step 1 - Database and Tables**

Run the script **SQLProjectData.sql** to create the database and tables listed below, and to load the data into the tables.

| | |
|---|---|
| Customers | 91 rows |
| Employees | 9 rows |
| Shippers | 3 rows |
| Suppliers | 15 rows |
| Products | 77 rows |
| Orders | 1078 rows |
| OrderDetails | 2820 rows |

**Part A - SQL Statements**

1. List the order detail rows where the quantity is greater than or equal to 90. Display the order id from the OrderDetails table, the product id and product name from the Products table, the supplier id and supplier name from the Suppliers table, and the calculated cost of the order using the formula (OrderDetails.Quantity * Products.UnitPrice). Order the result set by the cost of the order. The query should produce the result set listed below.

| OrderID | Cost | ProductID | ProductName | SupplierID | Name |
|---|---|---|---|---|---|
| 10895 | 495.00 | 24 | Guaraná Fantástica | 10 | Supplier J |
| 10052 | 697.50 | 75 | Rhönbräu Klosterbier | 12 | Supplier L |
| 10894 | 930.00 | 75 | Rhönbräu Klosterbier | 12 | Supplier L |
| ….. | | | | | |
| 11017 | 6050.00 | 59 | Raclette Courdavault | 8 | Supplier H |
| 10776 | 6360.00 | 51 | Manjimup Dried Apples | 14 | Supplier N |
| 10226 | 13616.90 | 29 | Thüringer Rostbratwurst | 12 | Supplier L |

(33 row(s) affected)

2. List the product id, product name, category, and unit price from the Products table, the supplier name from the Suppliers table, and calculate a new price by increasing the unit price by 15% using the formula (unit price * 1.15) where the unit price is greater than or equal to $20.00, and the category id is greater than or equal to 1 and less than or equal to 4. Round the new price to 2 decimal points. Order the result set by the product name. Use a CASE statement to change the category id as follows:

| Category ID value | Change to |
|---|---|
| 1 | Beverages |
| 2 | Sauces & Syrups |
| 3 | Desserts |
| 4 | Cheeses |
| ELSE | Unknown |

The query should produce the result set listed below.

| ProductID | ProductName | Categorys | UnitPrice | NewPrice | SupplierName |
|---|---|---|---|---|---|
| 60 | Camembert Pierrot | Cheeses | 34.00 | 39.10 | Supplier H |
| 4 | Chef Anton’s Cajun Seasoning | Sauces & Syrups | 22.00 | 25.30 | Supplier B |
| 5 | Chef Anton’s Gumbo Mix | Sauces & Syrups | 21.35 | 24.55 | Supplier B |
| ….. | | | | | |
| 61 | Sirop d’érable | Sauces & Syrups | 28.50 | 32.78 | Supplier I |
| 62 | Tarte au sucre | Desserts | 49.30 | 56.70 | Supplier I |
| 63 | Vegie-spread | Sauces & Syrups | 43.90 | 50.49 | Supplier G |

(22 row(s) affected)

3. List the company name, contact name, phone number, and country from the Customers table, and the sum of the freight from the Orders table where the order date is between January 1, 1993 and June 30, 1993. Order the result set by the company name. The query should produce the result set listed below.

| CompanyName | ContactName | Phone | Country | Total_Freight |
|---|---|---|---|---|
| Antonio Moreno Taquería | Antonio Moreno | (5) 555-3932 | Mexico | 147.93 |
| Around the Horn | Thomas Hardy | (71) 555-7788 | UK | 98.33 |
| Berglunds snabbköp | Christina Berglund | 0921-12 34 65 | Sweden | 374.02 |
| ….. | | | | |
| White Clover Markets | Karl Jablonski | (206) 555-4112 | USA | 150.93 |
| Wilman Kala | Matti Karttunen | 90-224 8858 | Finland | 0.75 |
| Wolski Zajazd | Zbyszek Piestrzeniewicz | (26) 642-7012 | Poland | 80.65 |

(73 row(s) affected)

4. List the first name, middle name and last name from the Employees table, and the shipping city and their total number or count of rows from the Orders table. Only display the shipping city if the count of the shipping city is greater than or equal to 7. Display the first name of the employee followed by a space followed by the middle name followed by a space followed by the last name, but do not leave a space if the employee does not have a middle name. Order the result set by the row count. The query should produce the result set listed below.

| EmployeeName | ShipCity | Count |
|---|---|---|
| Laura Callahan | Bräcke | 7 |
| Margaret Elizabeth Peacock | Charleroi | 7 |
| Margaret Elizabeth Peacock | Madrid | 7 |
| ….. | | |
| Nancy Sally Davolio | London | 9 |
| Margaret Elizabeth Peacock | Rio de Janeiro | 10 |
| Nancy Sally Davolio | Boise | 10 |

(12 row(s) affected)

5. List the orders where the shipped date is greater than or equal to January 1, 1994 and less than or equal to January 31, 1994, and calculate the length in days from the order date and the shipped date. Display the order id, and the shipped date from the Orders table, the company name from the Customers table, the calculated order cost by using the formula (Quantity * UnitPrice). Display the shipped date in the format MMM DD YYYY. Order the result set by the shipped date. The query should produce the result set listed below.

| OrderID | CompanyName | Order_Cost | ShippedDate | Days |
|---|---|---|---|---|
| 10864 | Around the Horn | 72.00 | Jan 03 1994 | 7 |
| 10864 | Around the Horn | 210.00 | Jan 03 1994 | 7 |
| 10869 | Seven Seas Imports | 720.00 | Jan 03 1994 | 5 |
| ….. | | | | |
| 10916 | Rancho grande | 104.70 | Jan 31 1994 | 10 |
| 10916 | Rancho grande | 192.00 | Jan 31 1994 | 10 |

| | | | | |
|---|---|---|---|---|
| 10916 | Rancho grande | 390.00 | Jan 31 1994 | 10 |

(140 row(s) affected)

6. List all the orders where the order date is from 1994, and the cost of the order is greater than or equal to $2,500.00. Display the order id and a new shipped date calculated by adding 10 days to the shipped date from the Orders table, the product id from the Products table, the company name from the Customer table, and the cost of the order. Use the formula (OrderDetails.Quantity * Products.UnitPrice) to calculate the cost of the order. Format the calculated shipped date as MMM DD YYYY. Order the result set by the company name. The query should produce the result set listed below.

| OrderID | ProductID | CompanyName | OrderCost | ShippedDate |
|---|---|---|---|---|
| 10953 | 20 | Around the Horn | 4050.00 | Feb 26 1994 |
| 10949 | 62 | Bottom-Dollar Markets | 2958.00 | Feb 18 1994 |
| 10895 | 60 | Ernst Handel | 3400.00 | Jan 27 1994 |
| ….. | | | | |
| 11030 | 29 | Save-a-lot Markets | 7427.40 | Mar 31 1994 |
| 11030 | 59 | Save-a-lot Markets | 5500.00 | Mar 31 1994 |
| 11032 | 38 | White Clover Markets | 6587.50 | Mar 27 1994 |

(16 row(s) affected)

7. List the suppliers with a supplier id greater than or equal to 1 and less than or equal to 5, and the customers with a country of Canada or Italy. Display the supplier name from the Suppliers table, and company name, contact name and phone number from the Customers table. Display ‘Suppliers’ for the rows from the Suppliers table, and ‘Customers’ for rows from the Customers table. Order the result set by supplier name. The query should produce the result set listed below.

| TableName | Name | ContactName | Phone |
|---|---|---|---|
| Customers | Bottom-Dollar Markets | Elizabeth Lincoln | (604) 555-4729 |
| Customers | Franchi S.p.A. | Paolo Accorti | 011-4988260 |
| Customers | Laughing Bacchus Wine Cellars | Yoshi Tannamuri | (604) 555-3392 |
| ….. | | | |
| Suppliers | Supplier B | | |
| Suppliers | Supplier C | | |
| Suppliers | Supplier D | | |

(11 row(s) affected)

8. List all the orders that have a shipped date of NULL and an employee that has a city of New Westminster in the Employees table. Display the customer id and phone number from the Customers table, the first name and last name from the Employees table, and the order id and order date from the Orders table. Format the employee's name as the last name followed by a comma and a space followed by the first name. Format the order date as YYYY-MM-DD. Order the result set by the company name and order date. The query should produce the result set listed below.

| CustomerID | Phone | Name | OrderID | OrderDate |
|---|---|---|---|---|
| BLAUS | 0621-08460 | Dodsworth, Anne | 11058 | 1994-03-23 |
| BOTTM | (604) 555-4729 | Suyama, Michael | 11045 | 1994-03-17 |
| ERNSH | 7675-3425 | King, Robert | 11008 | 1994-03-02 |
| LAMAI | 61.77.61.10 | King, Robert | 11051 | 1994-03-21 |
| RANCH | (1) 123-5555 | Suyama, Michael | 11019 | 1994-03-07 |
| SIMOB | 31 12 34 56 | King, Robert | 11074 | 1994-03-30 |

(6 row(s) affected)

**Part B - INSERT, UPDATE, DELETE and VIEWS Statements**

1. **Create a view called suppliers_products_vw listing the products and their suppliers where the units on order is greater than 0. Display the product id, quantity per unit, units in stock, and units on order from the Products table, and the supplier name from the Suppliers table. Use the following query to test your view to produce the result set listed below.**

```
SELECT *
FROM suppliers_products_vw
ORDER BY ProductID;
```

| ProductID | QuantityPerUnit | UnitsInStock | UnitsOnOrder | SupplierName |
|---|---|---|---|---|
| 2 | 24 - 12 oz bottles | 17 | 40 | Supplier A |
| 3 | 12 - 550 ml bottles | 13 | 70 | Supplier A |
| 11 | 1 kg pkg. | 22 | 30 | Supplier |
| ….. | | | | |
| 68 | 10 boxes x 8 pieces | 6 | 10 | Supplier H |
| 70 | 24 - 355 ml bottles | 15 | 10 | Supplier G |
| 74 | 5 kg pkg. | 4 | 20 | Supplier D |

**(17 row(s) affected)**

2. **Using the UPDATE statement, change the fax value to 'Unknown' in the Customers table where the current fax value is NULL and the Shipping Country in the Orders table is Portugal. There should be 1 row affected.**

3. **Using the INSERT statement, add two rows to the Employees table. The first row should have an employee id of 10, last name of Stevenson, first name of Susan, and birth date of May 13, 1990. The second row should have an employee id of 11, last name of Thompson, first name of Darlene, and birth date of September 10, 1995.**

4. **Create a view called employee_inform_vw to list the employees in the Employee table. Display the employee id, last name, first name, phone number, and birth date. Format the name as first name followed by a space followed by the last name. Display the phone number as opening bracket followed by the first 3 digits of the phone number followed by the closing bracket followed by the next 3 digits of the phone number followed by a dash followed by the last 4 digits of the phone number. Display spaces if the employee does not have a phone number. Display the birth date as MMM DD YYYY. Use the following query to test your view to produce the result set listed below.**

```
SELECT *
FROM employee_inform_vw
WHERE EmployeeID IN ( 3, 9, 11 );
```

| EmployeeID | Name | PhoneNumber | BirthDate |
|---|---|---|---|
| 3 | Janet Leverling | (604)555-3412 | Aug 30 1963 |
| 9 | Anne Dodsworth | (604)555-4444 | Jan 27 1966 |
| 11 | Darlene Thompson | | Sep 10 1995 |

(3 row(s) affected)

5. **Using a subquery, list all the orders that have a shipped date of NULL and an employee that has a city of New Westminster in the Employees table. Display the customer id, contact name and phone number from the Customers table, and the order id and order date from the Orders table. Format the order date as YYYY-MM-DD. Order the result set by the company name and order date. The query should produce the result set listed below.**

| CustomerID | ContactName | Phone | OrderID | OrderDate |
|---|---|---|---|---|
| BLAUS | Hanna Moos | 0621-08460 | 11058 | 1994-03-23 |
| BOTTM | Elizabeth Lincoln | (604) 555-4729 | 11045 | 1994-03-17 |
| ERNSH | Roland Mendel | 7675-3425 | 11008 | 1994-03-02 |
| LAMAI | Annette Roulet | 61.77.61.10 | 11051 | 1994-03-21 |
| RANCH | Sergio Gutiérrez | (1) 123-5555 | 11019 | 1994-03-07 |
| SIMOB | Jytte Petersen | 31 12 34 56 | 11074 | 1994-03-30 |

(6 row(s) affected)

6. **Using the UPDATE statement, add the phone number 6042537581 to the Employees table for the employee id of 10.**

7. **Create a view called order_shipped _vw to list the orders where the difference between the shipped date and the order date is greater than 10 days, and with an order date year greater than or equal to 1993. Display the order id, order date and shipped date from the Orders table, the first name and last name from the Employees table, and the shipper name from the Shippers table, and the difference of the days between the shipped date and the order date. Format the employee name as the first name followed by a space followed by the last name. Display the order and shipped date in the format as YYYY-MM-DD. Use the following query to test your view to produce the result set listed below.**

```
SELECT *
FROM order_shipped _vw
ORDER BY OrderDate;
```

| OrderID | OrderDate | ShippedDate | EmployeeName | ShipperName | DayDifference |
|---|---|---|---|---|---|
| 10440 | 1993-01-04 | 1993-01-22 | Margaret Peacock | United Package | 18 |
| 10441 | 1993-01-04 | 1993-02-05 | Janet Leverling | United Package | 32 |
| 10447 | 1993-01-08 | 1993-01-29 | Margaret Peacock | United Package | 21 |
| ….. | | | | | |
| 11022 | 1994-03-08 | 1994-03-28 | Anne Dodsworth | United Package | 20 |

| | | | | | |
|---|---|---|---|---|---|
| 11026 | 1994-03-09 | 1994-03-22 | Margaret Peacock | Speedy Express | 13 |
| 11029 | 1994-03-10 | 1994-03-21 | Margaret Peacock | Speedy Express | 11 |

(108 row(s) affected)

8. **Using the DELETE statement, delete employees with an employee id of 10 or 11 from the Employees table.**

Part C - Stored Procedures, Triggers, and Functions

1. Create a stored procedure called orders_by_dates_sp displaying the orders shipped between specific dates. The start and end dates will be input parameters for the stored procedure. Display the order id, order date, and shipped date from the Orders table, the product id from the OrderDetails table, and the company name from the Customer table. Display the order and shipped date in the format MMM DD YYYY. Print the message 'Please enter valid dates' if either the start or end date is not supplied. Order the result set by the shipped date. Use the following query to test your stored procedure to produce the result set listed below.

```
EXECUTE orders_by_dates_sp;                          -- Both dates missing. Print message

EXECUTE orders_by_dates_sp '1994-03-01';             -- One date is missing. Print message

EXECUTE orders_by_dates_sp '1994-03-01', '1994-03-31';    --Print results below.
```

| OrderID | ProductID | CustomerCompany | OrderDate | ShippedDate |
|---|---|---|---|---|
| 10951 | 33 | Richter Supermarkt | Feb 07 1994 | Mar 01 1994 |
| 10951 | 41 | Richter Supermarkt | Feb 07 1994 | Mar 01 1994 |
| 10951 | 75 | Richter Supermarkt | Feb 07 1994 | Mar 01 1994 |
| ….. | | | | |
| 11063 | 34 | Hungry Owl All-Night Grocers | Mar 24 1994 | Mar 30 1994 |
| 11063 | 40 | Hungry Owl All-Night Grocers | Mar 24 1994 | Mar 30 1994 |
| 11063 | 41 | Hungry Owl All-Night Grocers | Mar 24 1994 | Mar 30 1994 |

(195 row(s) affected)

2. Create an INSERT trigger called insert_employee_tr on the Employees table to prevent the adding of a row with a phone number that is blank or NULL. Print the message 'Phone number is incorrect' when the phone number is not valid. Use the following query to test your trigger.

```
INSERT Employees                -- Trigger should prevent the insert and print message.
VALUES( 20, 'Doe', 'Jane', 'Sally', '15 Pine Street', 'Vancouver', 'BC', 'V6X 4T6', NULL, '1975-05-23' );

INSERT Employees                -- Trigger should prevent the insert and print message.
VALUES( 20, 'Doe', 'Jane', 'Sally', '15 Pine Street', 'Vancouver', 'BC', 'V6X 4T6', '', '1975-05-23' );

INSERT Employees                --Trigger should allow the insert.
VALUES( 20, 'Doe', 'Jane', 'Sally', '15 Pine Street', 'Vancouver', 'BC', 'V6X 4T6', '6045552581', '1975-05-23' );
```

3. Create an UPDATE trigger called check_shippeddate_tr on the Orders table to check that the shipped date is less than the required date of the order. Print the message 'Order was shipped after the required date' when the shipped date is greater than the required date, else print the message ‘Order was shipped successfully’. Allow all the update commands to process. Use the following queries to test your trigger.

```
UPDATE Orders                          -- Trigger should print message that shipped
SET ShippedDate = '1994-04-20'         -- date late, and row will update.
WHERE OrderID = 11051
  AND CustomerID = 'LAMAI'
  AND EmployeeID = 7;

UPDATE Orders                          -- Trigger should print message that order
SET ShippedDate = '1994-04-10'         -- shipped on time, and row will update.
WHERE OrderID = 11039
  AND CustomerID = 'LINOD'
  AND EmployeeID = 1;
```

4. Create a stored procedure called shipping_date_sp where the shipped date is equal to an input parameter of the shipped date for the stored procedure. Display the order id, order date, required date, and shipped date from the Orders table, and the company name and phone number from the Customers table. Display all the dates in the format MMM DD YYYY. Order the result set by the order date. If the shipped date is missing, display the message ‘Please enter a valid shipped date’. Use the following query to test your stored procedure to produce the result set listed below.

```
EXECUTE shipping_date_sp;                  -- Date missing. Print message.

EXECUTE shipping_date_sp '1994-03-01';     -- Print result set below.
```

| OrderID | CompanyName | CustomerPhone | OrderDate | RequiredDate | ShippedDate |
|---|---|---|---|---|---|
| 10951 | Richter Supermarkt | 0897-034214 | Feb 07 1994 | Mar 21 1994 | Mar 01 1994 |
| 10990 | Ernst Handel | 7675-3425 | Feb 23 1994 | Apr 06 1994 | Mar 01 1994 |
| 10991 | QUICK-Stop | 0372-035188 | Feb 23 1994 | Mar 23 1994 | Mar 01 1994 |

(3 row(s) affected)

5. Create a stored procedure called sales_by_employees_sp to list the cost of invoices by employees. To determine the order cost, the formula will be (OrderDetails.Quantity * OrderDetails.UnitPrice). Display the employee id, first name, last name from the Employees table, and the sum of the order cost. Format the employee's name as the first name followed by a space followed by the last name. Order the result set by the last name of the employee. Use the following query to test your stored procedure to produce the result set listed below.

   EXECUTE sales_by_employees_sp;

| EmployeeID | Name | InvoiceCost |
|---|---|---|
| 5 | Steven Buchanan | 90467.55 |
| 8 | Laura Callahan | 174341.83 |
| 1 | Nancy Davolio | 237508.81 |
| ….. | | |
| 3 | Janet Leverling | 264607.00 |
| 4 | Margaret Peacock | 308128.65 |
| 6 | Michael Suyama | 92182.80 |

   (9 row(s) affected)

6. Create a scalar user defined function called OrderCost to calculate the cost of an order. The formula to calculate the cost of the order is ((1.0 – Discount) * (UnitPrice * Quantity) + Freight).

7. Create a stored procedure called total_cost_sp to list the order id from the OrderDetails table, the English name from the Products table, and the total cost of each order. The two input parameters for the stored procedure are the start and end order id values. To calculate the cost of the orders, use your user defined function OrderCost created in Part C question 6. Order the result set by the order id. Use the following query to test your stored procedure to produce the result set listed below.

   EXECUTE total_cost_sp 10800, 10850;

| OrderID | EnglishName | TotalCost |
|---|---|---|
| 10800 | Cabrales Cheese | 1082.44 |
| 10800 | Manjimup Dried Apples | 614.44 |
| 10800 | Pork Pie 184.38 | |
| ….. | | |
| 10850 | NuNuCa Chocolate-Nut Spread | 287.19 |
| 10850 | Goat Cheese | 57.69 |
| 10850 | Outback Lager | 431.69 |

   (139 row(s) affected)

8. Create a multi-table valued function called determine_discount which has two input parameters; one parameter for the quantity value and one parameter for the discount value. List the quantity, discount, and new discount for customers, and the product where the product is not discontinued (Product. Discontinued value is zero) and the quantity is equal to the input parameter of the quantity value. The new discount value is calculated by adding the discount in the OrderDetails table to the input parameter of the discount value. Display the customer id and company name from the Customers table, the product name for the Products table, the quantity and discount from the OrderDetails table, and the calculated new discount value. Use the following query to test your function to produce the result set listed below.

```
SELECT *
FROM determine_discount( 80, 0.05 );
```

| CustomerID | CompanyName | ProductName | Quantity | Discount | NewDiscount |
|---|---|---|---|---|---|
| ERNSH | Ernst Handel | Louisiana Fiery Hot Pepper Sauce | 80 | 0.05 | 0.10 |
| SAVEA | Save-a-lot Markets | Queso Manchego La Pastora | 80 | 0.00 | 0.05 |
| FRANK | Frankenversand | Chef Anton's Cajun Seasoning | 80 | 0.00 | 0.05 |
| ….. | | | | | |
| QUICK | QUICK-Stop | Chartreuse verte | 80 | 0.05 | 0.10 |
| SAVEA | Save-a-lot Markets | Scottish Longbreads | 80 | 0.25 | 0.30 |
| SAVEA | Save-a-lot Markets | Konbu | 80 | 0.00 | 0.05 |

(16 row(s) affected)

100 Marks