**ISQS 6338 (D01) – Fall 2018**

**Module 07 Learning Exercise**

1. SELECT UnitsInStock, ReorderLevel, ProductName

FROM products

WHERE UnitsInStock <= ReorderLevel

A screenshot of a cell phone

Description automatically generated

1. SELECT CategoryID, SUM(UnitsInStock) AS Total

FROM products

GROUP BY CategoryID

ORDER BY SUM(UnitsInStock) DESC

A screenshot of a cell phone

Description automatically generated

1. SELECT Country, COUNT(Country) AS Total\_Country

FROM customers

GROUP BY Country

ORDER BY Total\_Country Asc

A close up of a white background

Description automatically generated

SELECT City, COUNT(City) AS Total\_City

FROM customers

GROUP BY City

ORDER BY Total\_City Asc

A screenshot of a cell phone

Description automatically generated

1. SELECT ProductID, ProductName, SupplierID

FROM products

GROUP BY ProductID

A screenshot of a cell phone

Description automatically generated

1. SELECT OrderID, DATE(OrderDate), ShipName

FROM orders

WHERE OrderID < 10300

A close up of a person

Description automatically generated

1. SELECT UnitsInStock, UnitsOnOrder, ReorderLevel, ProductName, Discontinued

FROM products

WHERE (UnitsOnOrder+ReorderLevel)<=UnitsInStock AND Discontinued = 0

A screenshot of a cell phone

Description automatically generated

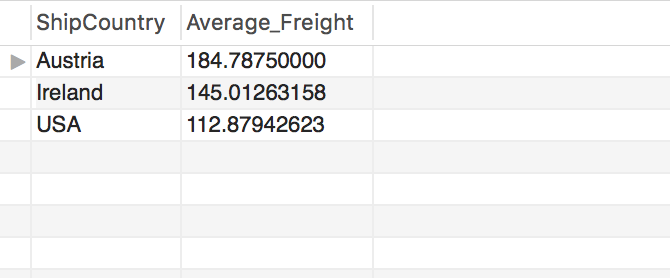
1. SELECT ShipCountry, AVG(Freight) AS Average\_Freight

FROM orders

GROUP BY ShipCountry

HAVING Average\_Freight > 100

ORDER BY Average\_Freight desc



1. SELECT ShipCountry, AVG(Freight) AS Average\_Freight

FROM orders

WHERE Date(Orderdate) between date('2015-01-01') and date('2015-12-31')

GROUP BY ShipCountry

HAVING Average\_Freight > 110

ORDER BY Average\_Freight desc

A screenshot of a social media post

Description automatically generated

1. select shipcountry, avg(freight) as average\_freight, date(min(orderdate)) as orderdate

from orders

where orderdate>=date\_add(curdate(),interval -1 year)

and orderdate<date\_add(curdate(),interval 1 day)

group by shipcountry

order by average\_freight desc

1. SELECT o.EmployeeID, e.LastName, o.OrderID, od.Quantity, p.ProductName

FROM employees e JOIN orders o ON o.EmployeeID = e.EmployeeID JOIN orderdetails od ON o.OrderID = od.OrderID, products p

1. SELECT CustomerID, CompanyName, Region

FROM customers

ORDER BY case when Region is NULL then 1 else 0 end, Region, CustomerID

1. SELECT OrderID, LAST\_DAY('2016-05-01'), EmployeeID

FROM orders

ORDER BY OrderID

A screenshot of a cell phone

Description automatically generated

1. SELECT \*

FROM orders

ORDER BY case when ShipRegion is NULL then 1 else 0 end, ShipRegion

LIMIT 10

1. SELECT CustomerID, EmployeeID

FROM orders

WHERE EmployeeID != 4

1. SELECT OrderID, CustomerID

FROM orders

WHERE CustomerID IN (SELECT CustomerID

FROM orderdetails

WHERE OrderID = 0)

1. select orderdetails.OrderID, orders.CustomerID, SUM(UnitPrice) AS Total\_UnitPrice

FROM orderdetails, orders, customers

WHERE orderdetails.OrderID = orders.OrderID

GROUP BY OrderID

HAVING Total\_UnitPrice >= 15.000

1. SELECT orderdetails.OrderID, orders.CustomerID, SUM(UnitPrice) AS Total\_UnitPrice, (UnitPrice\*Quantity\*0.30) AS Discount

FROM orderdetails, orders, customers

WHERE orderdetails.OrderID = orders.OrderID

GROUP BY OrderID

HAVING Total\_UnitPrice >= 15.000