MODELOS Y BASES DE DATOS

Consultas SQL Básico

2020-1

Laboratorio 1/6

Integrantes:

* Juan Andrés Pico
* Ana Gabriela Silva

**PARTE UNO. Conociendo la organización**

1. **Modelo lógico**

ProductCategory (PK, FK **ProductCategoryID**, *ParentProductCategoryID*)

Product (PK, FK, FK **ProductID**, *ProductModelID*, *ProductCategoryID*)

ProductModel (PK **ProductModelID**)

ProductModelProductDescription (FK, FK, PK *ProductModelID*, *ProductDescriptionID*, **Culture**)

ProductDescription (PK **ProductDescriptionID**)

SalesOrderDetail (FK, PK, FK *SalesOrderID*, **SalesOrderDetail**, *ProductID*)

SalesOrderHeader (PK, FK, FK, FK **SalesOrderID**, *CustomerID*, *BillToAddressID*, *ShipToAddressID*)

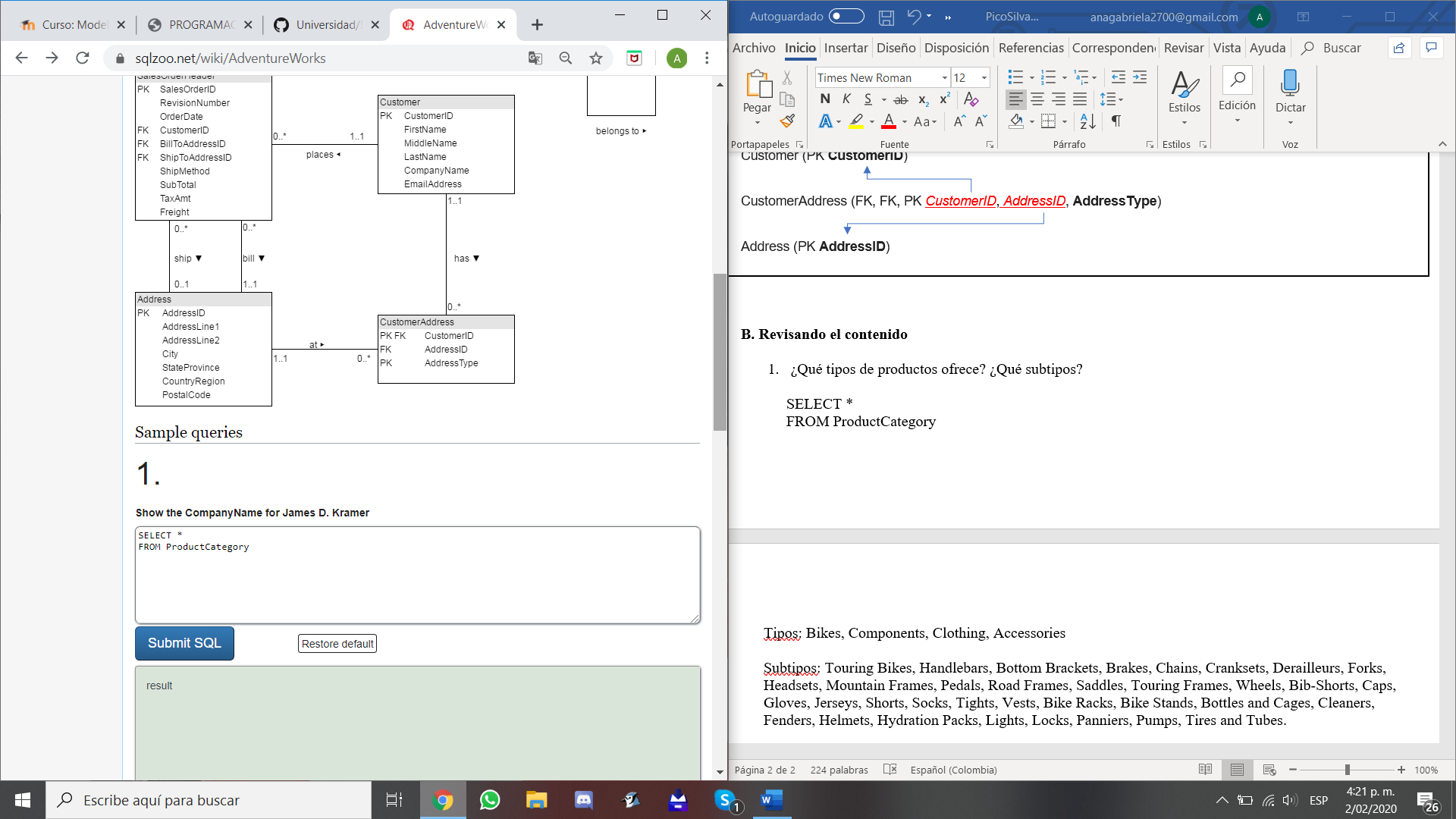
Customer (PK **CustomerID**)

CustomerAddress (FK, FK, PK *CustomerID*, *AddressID*, **AddressType**)

Address (PK **AddressID**)

**B. Revisando el contenido**

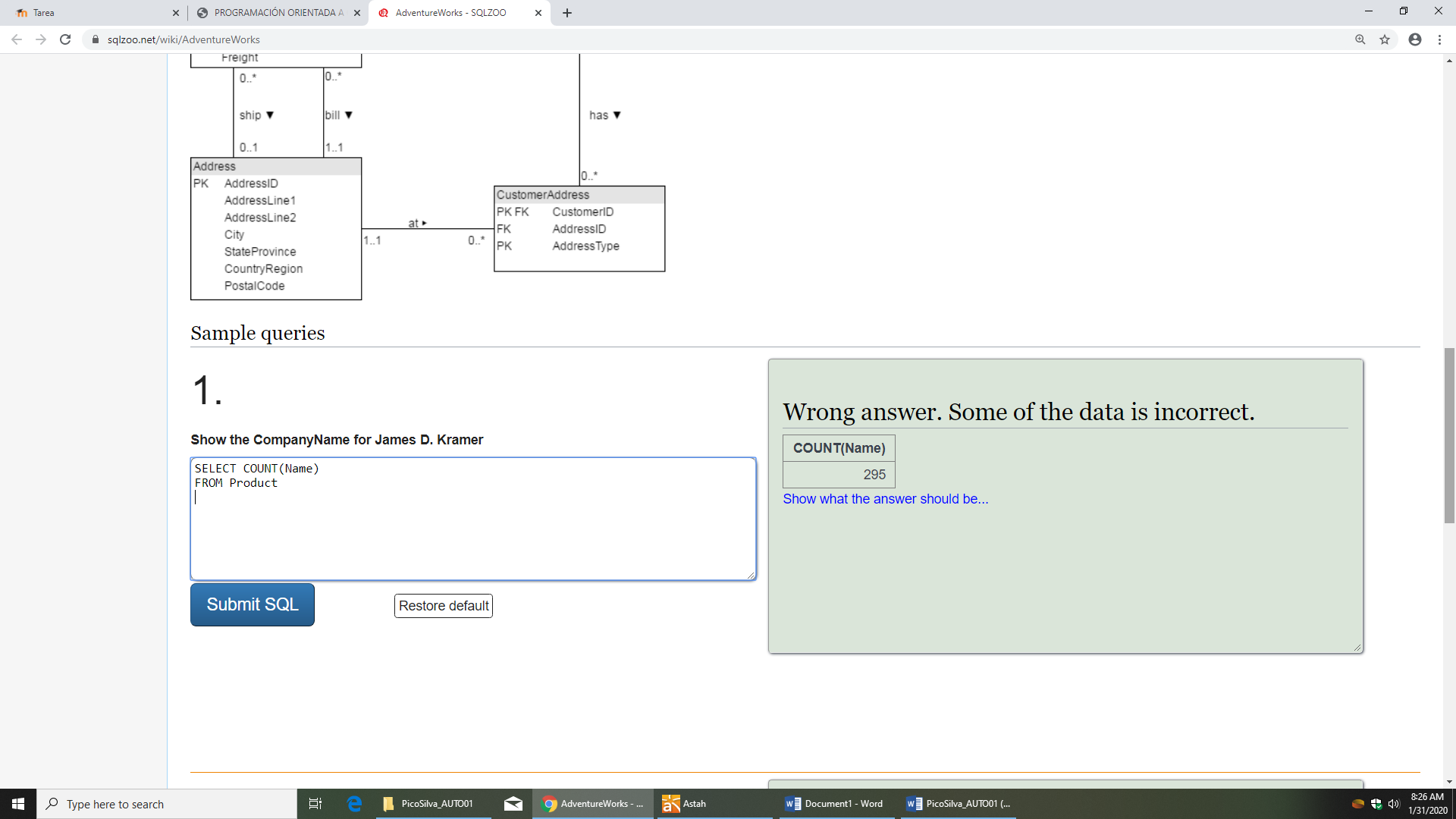
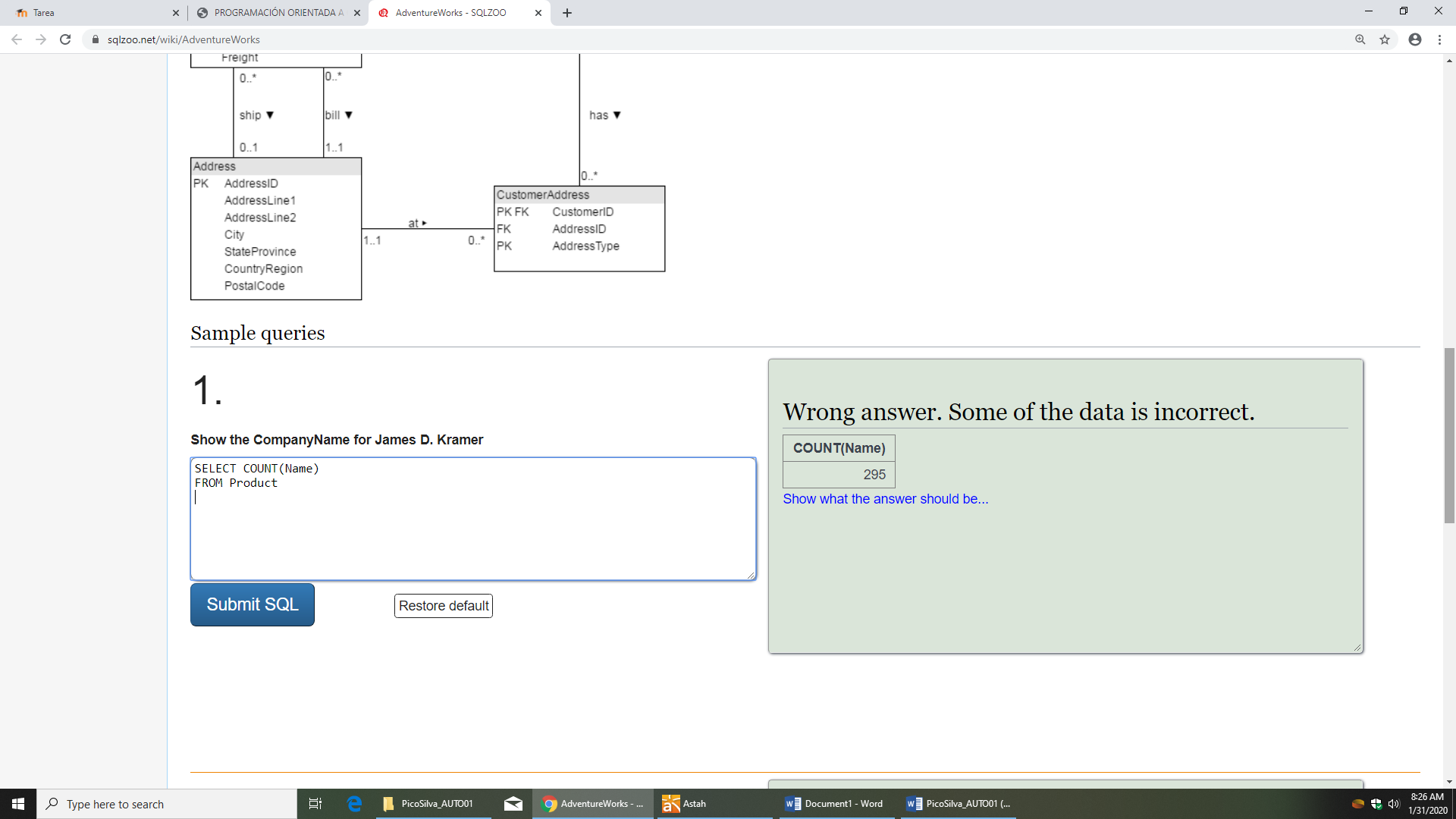
1. ¿Qué tipos de productos ofrece? ¿Qué subtipos?



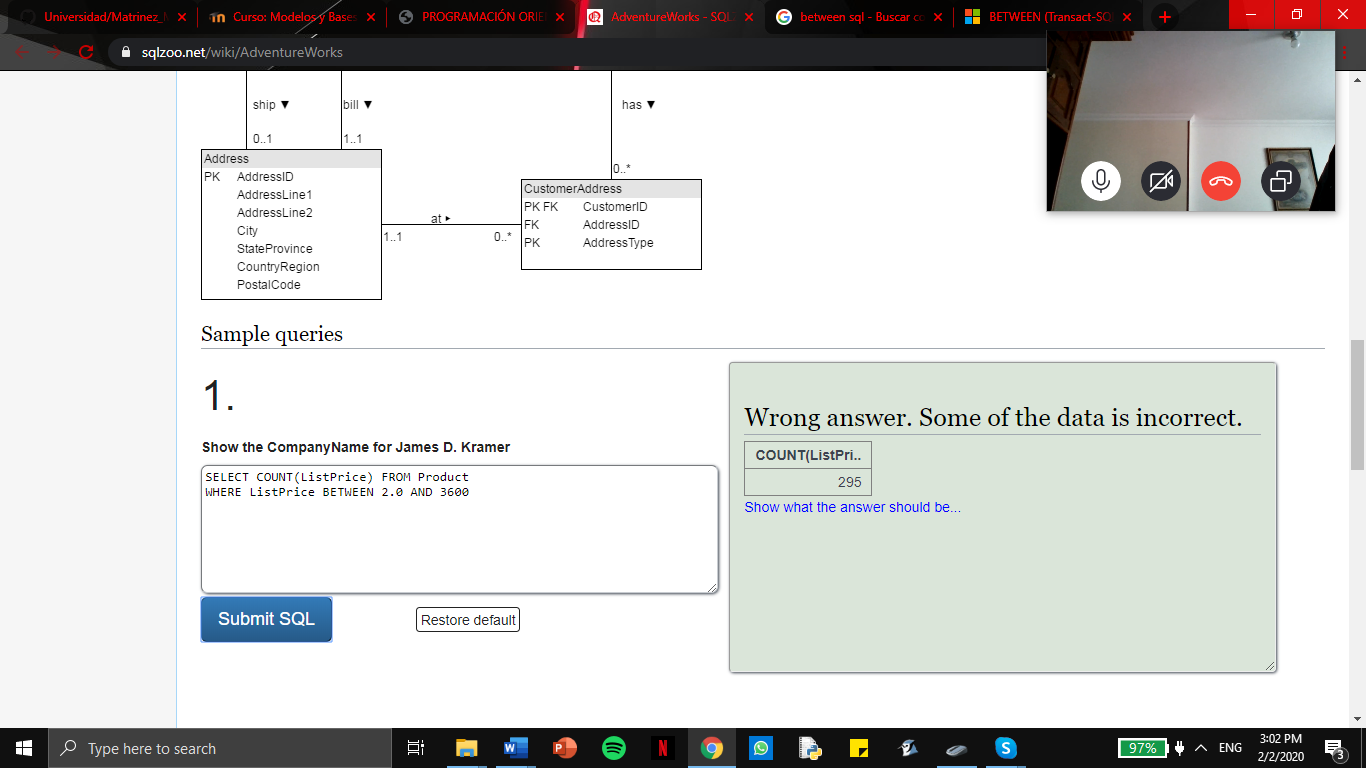
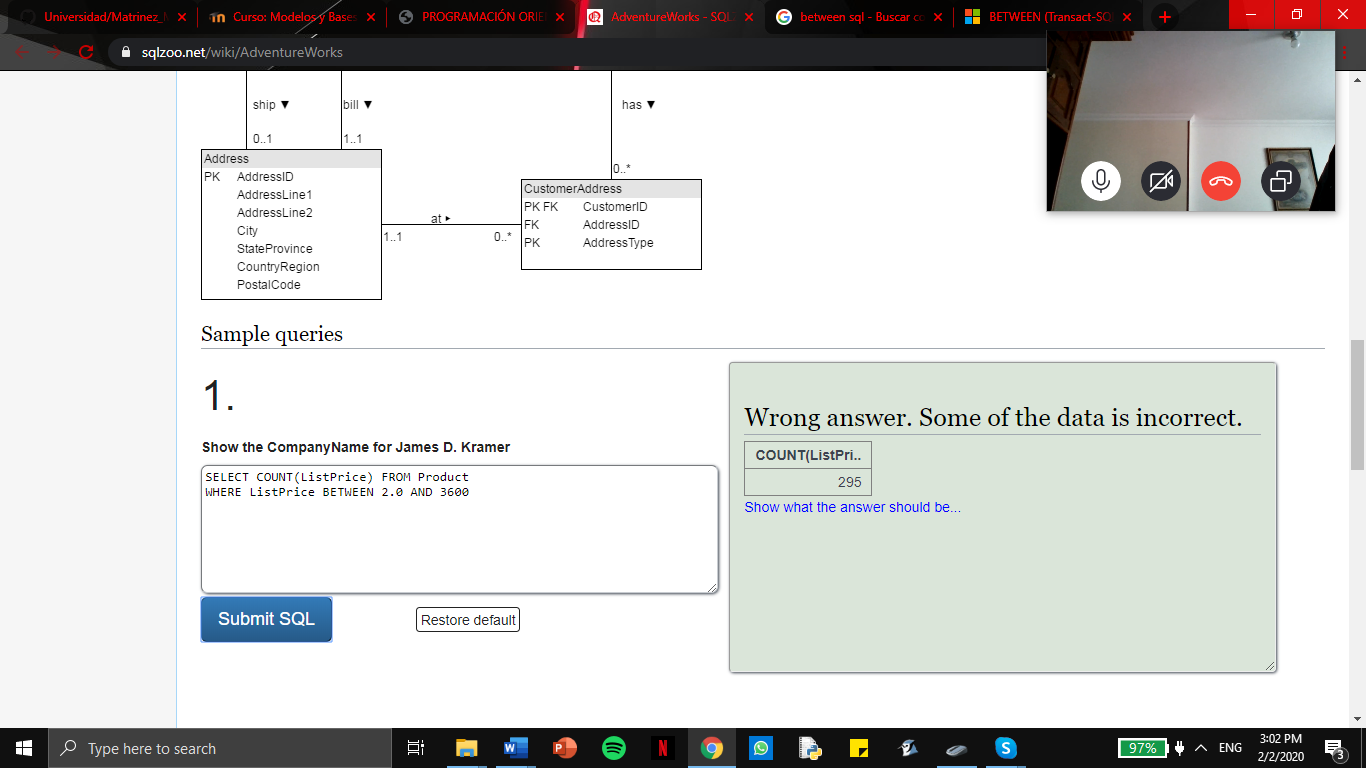
Tipos: Bikes, Components, Clothing, Accessories

Subtipos: Touring Bikes, Handlebars, Bottom Brackets, Brakes, Chains, Cranksets, Derailleurs, Forks, Headsets, Mountain Frames, Pedals, Road Frames, Saddles, Touring Frames, Wheels, Bib-Shorts, Caps, Gloves, Jerseys, Shorts, Socks, Tights, Vests, Bike Racks, Bike Stands, Bottles and Cages, Cleaners, Fenders, Helmets, Hydration Packs, Lights, Locks, Panniers, Pumps, Tires and Tubes.

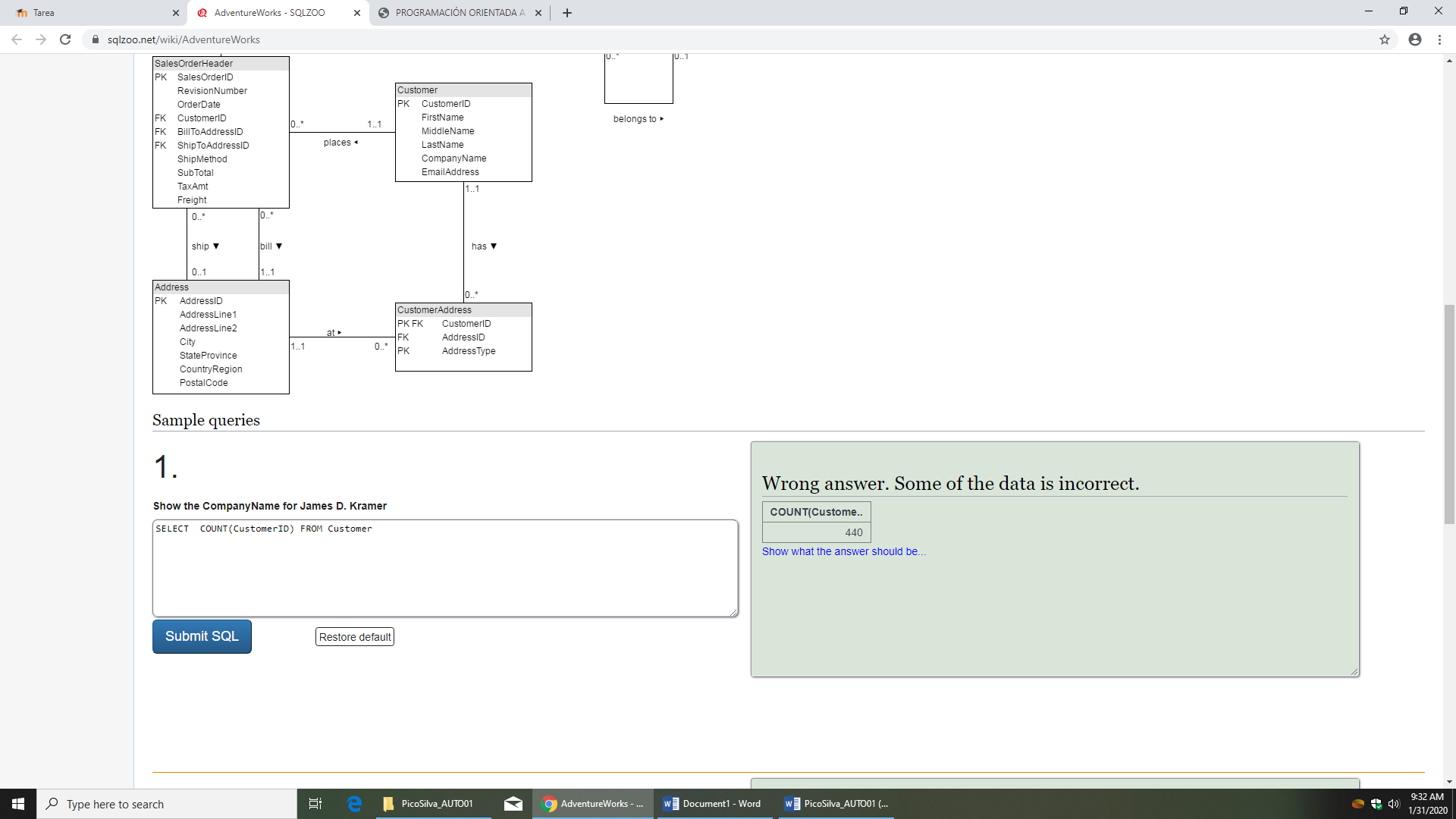
1. ¿Cuántos productos ofrece?

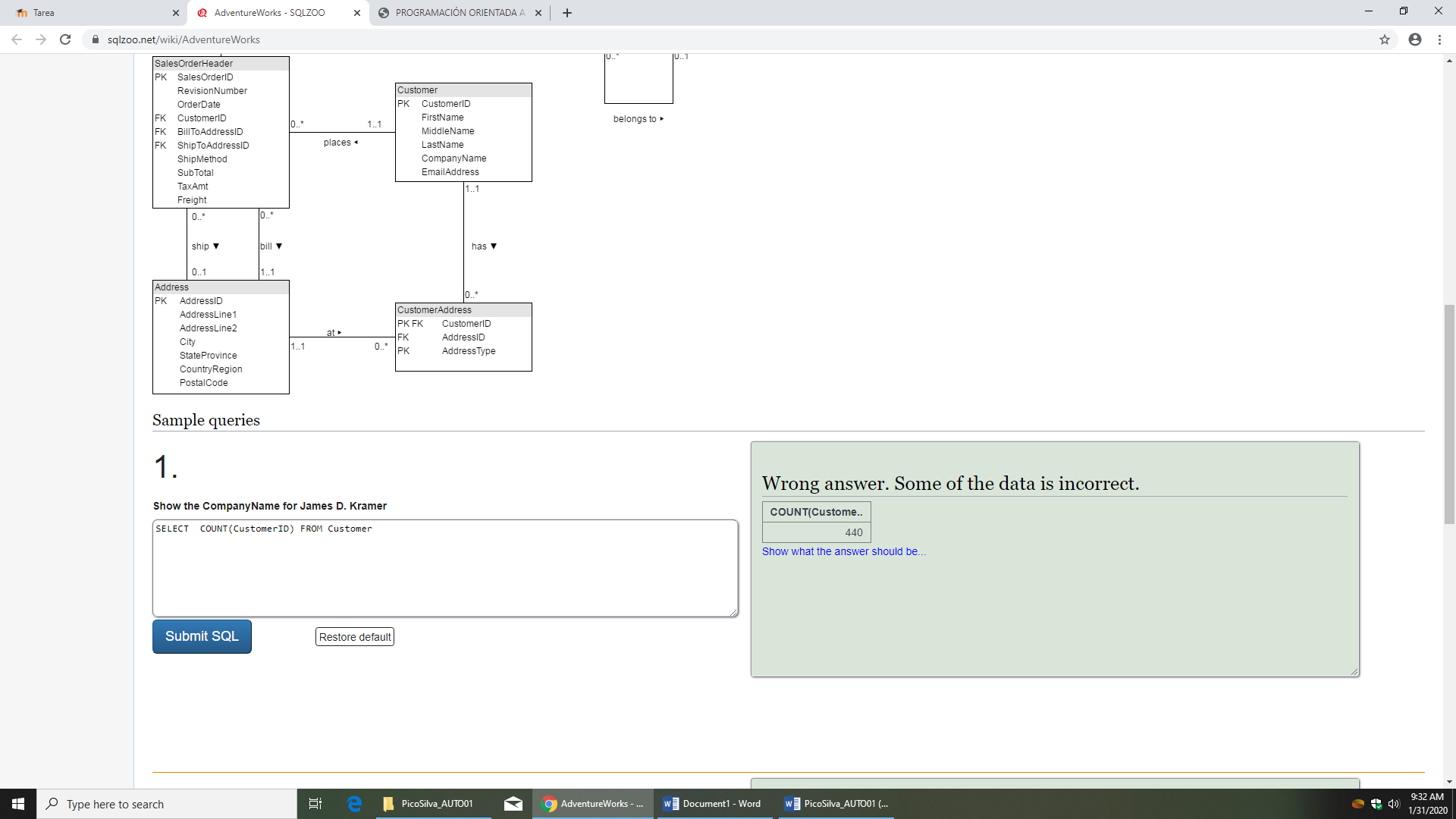


¿En qué rango de precios?

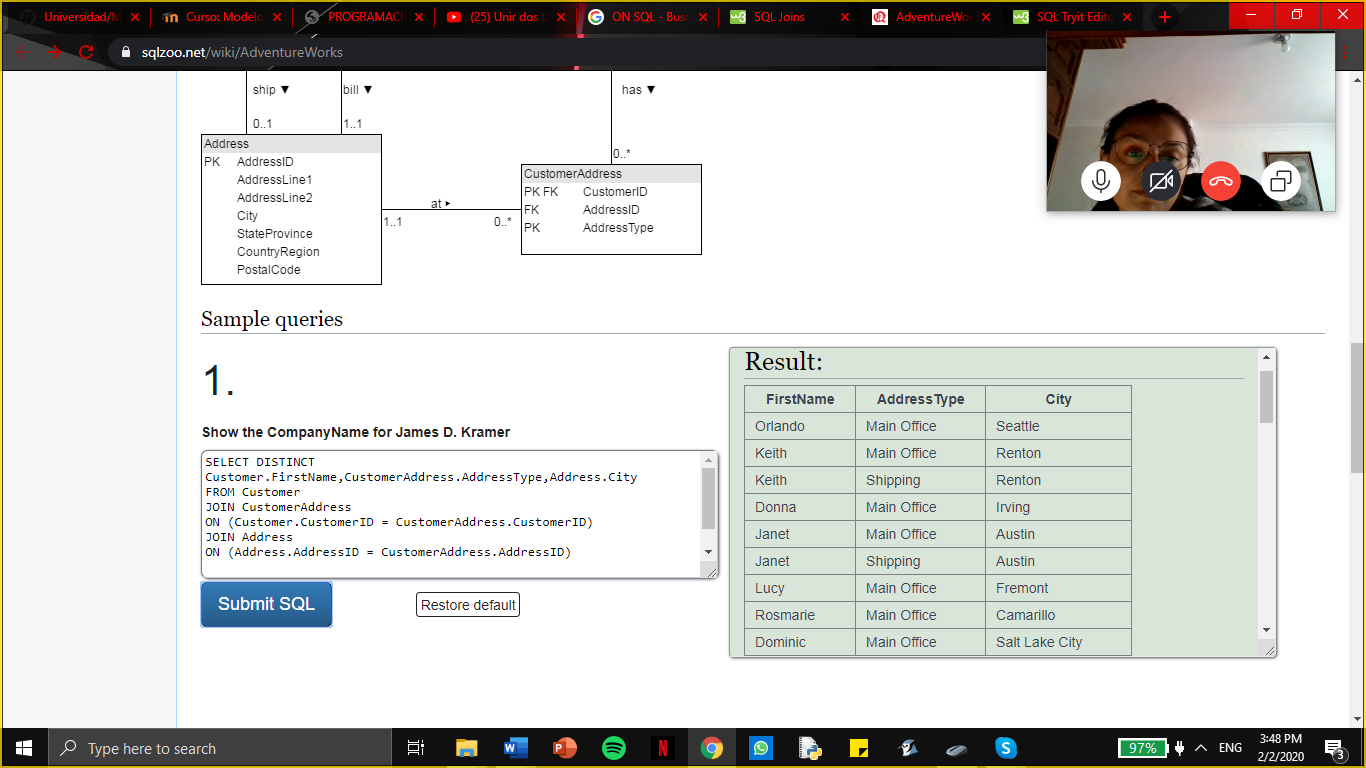


1. ¿Cuántos clientes tiene?

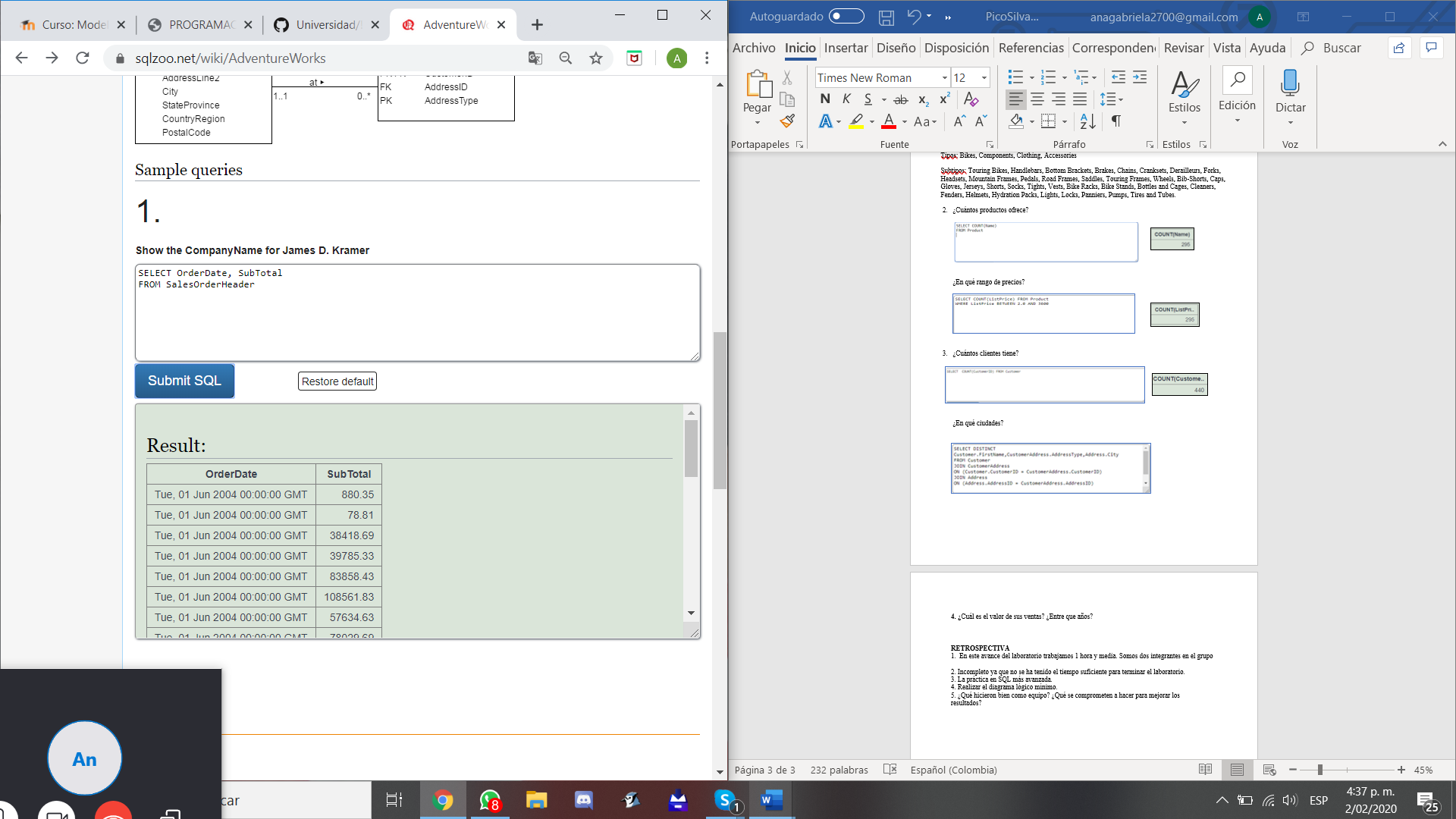




¿En qué ciudades?

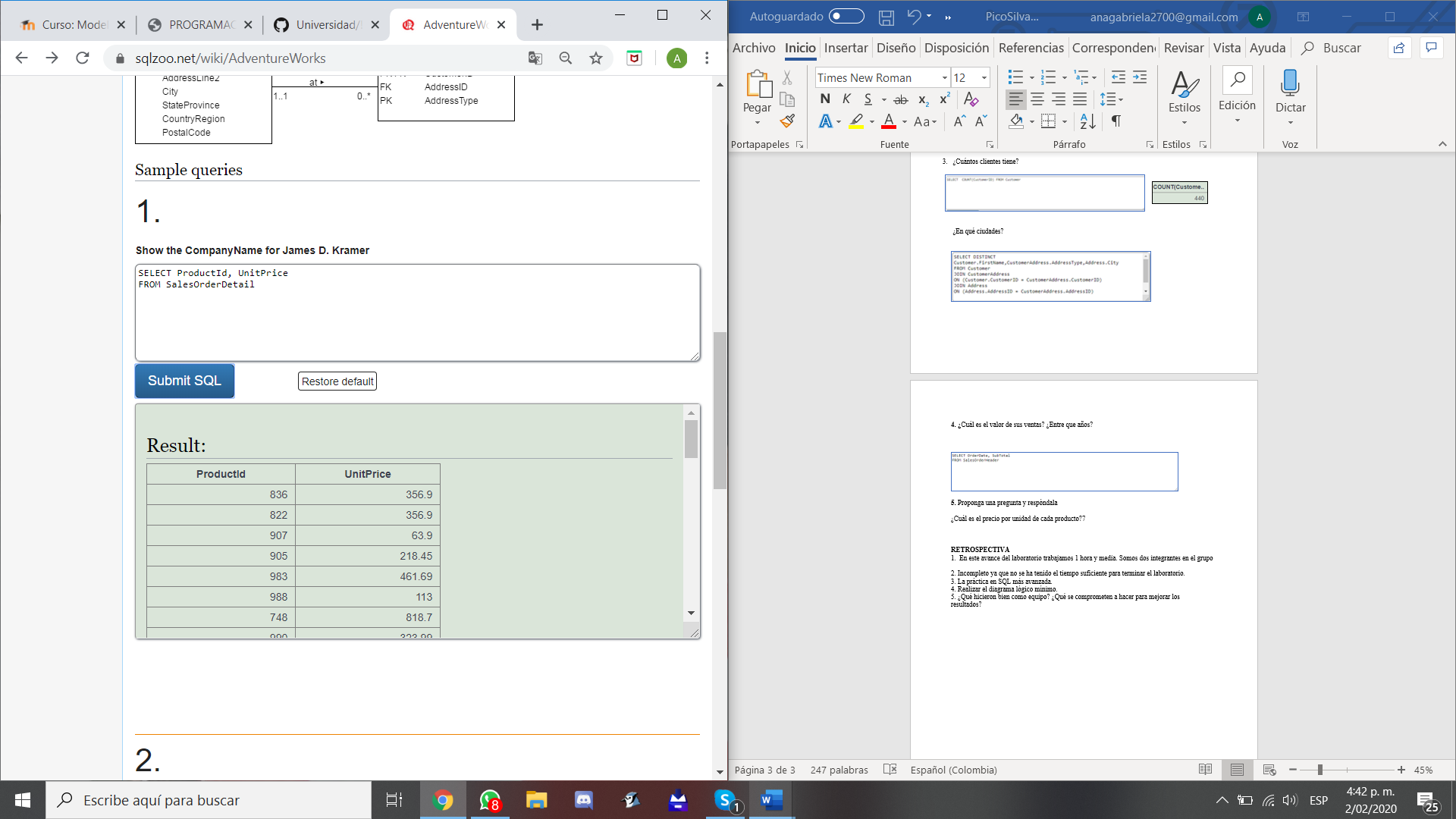


**4. ¿**Cuál es el valor de sus ventas? ¿Entre que años?



**5.** Proponga una pregunta y respóndala

¿Cuál es el precio por unidad de cada producto?



**C. Contexto**

La misión de la organización de AdventureWorld es dar registro ordenado de los productos que se ofrecen, así mismo el control de ventas al por mayor y al detal teniendo en cuenta la información de sus clientes.

Ofrece la posibilidad de conocer la información de ventas de una empresa.

**D. Usuarios**

* Clientes: son los compradores de los productos de la compañía.

**PARTE DOS. Implementando.**

1. **Easy Questions**

* **SQL:**

SELECT FirstName, EmailAddress

FROM Customer

WHERE CompanyName = "Bike World"

**Cálculo:**

**Álgebra:**

* **SQL**

SELECT CompanyName FROM Customer

JOIN CustomerAddress ON Customer.CustomerID = CustomerAddress.CustomerID

JOIN Address ON CustomerAddress.AddressID = Address.AddressID

WHERE Address.City = "Dallas"

**Cálculo:**

**Álgebra:**

* **SQL**

SELECT COUNT(\*) AS Total

FROM SalesOrderDetail

JOIN Product ON SalesOrderDetail.ProductID = Product.ProductID

WHERE Product.ListPrice > 1000

**Cálculo:**

**Álgebra:**

* **SQL**

SELECT CustomerID, CompanyName

FROM Customer

JOIN SalesOrderHeader ON (Customer.CustomerID = SalesOrderHeader.CustomerID )

WHERE SalesOrderHeader.Subtotal + SalesOrderHeader.TaxAmt + SalesOrderHeader.Freight > 100000

**Cálculo:**

**Álgebra:**

* **SQL**

SELECT SUM (SalesOrderDetail.OrderQty) As Total

FROM SalesOrderDetail

JOIN Product (ON SalesOrderDetail.ProductID = Product.ProductID)

JOIN SalesOrderHeader ON (SalesOrderDetail.SalesOrderID = SalesOrderHeader.SalesOrderID)

JOIN Customer ON (SalesOrderHeader.CustomerID = Customer.CustomerID)

WHERE Product.Name = 'Racing Socks, L' AND Customer.CompanyName = 'Riding Cycles';

**Cálculo:**

**Álgebra:**

1. **Medium Questions**

* **SQL**

SELECT SalesOrderID, UnitPrice

FROM SalesOrderDetail

WHERE OrderQty = 1

**Cálculo**

* **SQL**

SELECT Product.name, Customer.CompanyName

FROM ProductModel

JOIN Product ON ProductModel.ProductModelID = Product.ProductModelID

JOIN SalesOrderDetail

ON (SalesOrderDetail.ProductID = Product.ProductID)

JOIN SalesOrderHeader

ON (SalesOrderDetail.SalesOrderID = SalesOrderHeader.SalesOrderID)

JOIN Customer

ON (SalesOrderHeader.CustomerID = Customer.CustomerID)

WHERE ProductModel.Name = 'Racing Socks';

**Cálculo**

* **SQL**

SELECT ProductDescription.Description

FROM ProductDescription

JOIN ProductModelProductDescription

ON ProductDescription.ProductDescriptionID = ProductModelProductDescription.ProductDescriptionID

JOIN ProductModel

ON ProductModelProductDescription.ProductModelID = ProductModel.ProductModelID

JOIN Product

ON ProductModel.ProductModelID = Product.ProductModelID

WHERE ProductModelProductDescription.culture = 'fr' AND Product.ProductID = '736'

**Cálculo**

* **SQL**

SELECT Customer.CompanyName,

SalesOrderHeader.SubTotal,

SUM(SalesOrderDetail.OrderQty \* Product.weight)

FROM Product

JOIN SalesOrderDetail ON Product.ProductID = SalesOrderDetail.ProductID

JOIN SalesOrderHeader

ON SalesOrderDetail.SalesOrderID = SalesOrderHeader.SalesorderID

JOIN Customer

ON SalesOrderHeader.CustomerID = Customer.CustomerID

GROUP BY SalesOrderHeader.SalesOrderID, SalesOrderHeader.SubTotal, Customer.CompanyName

ORDER BY SalesOrderHeader.SubTotal DESC

**Cálculo**

**Group by ¿Calculo?**

* **SQL**

SELECT SUM (SalesOrderDetail.OrderQty)

FROM ProductCategory

JOIN Product

ON ProductCategory.ProductCategoryID = Product.ProductCategoryID

JOIN SalesOrderDetail

ON Product.ProductID = SalesOrderDetail.ProductID

JOIN SalesOrderHeader

ON SalesOrderDetail.SalesOrderID = SalesOrderHeader.SalesorderID

JOIN Address

ON SalesOrderHeader.ShipToAddressID = Address.AddressID

WHERE Address.City = 'London' AND ProductCategory.Name = 'Cranksets'

**Cálculo**

1. **Hard Questions**

* **SQL**

SELEC SalesOrderHeader.SalesOrderID, SalesOrderHeader.SubTotal,

SUM(SalesOrderDetail.OrderQty \* SalesOrderDetail.UnitPrice),

SUM(SalesOrderDetail.OrderQty \* Product.ListPrice)

FROM SalesOrderHeader

JOIN SalesOrderDetail

ON SalesOrderHeader.SalesOrderID = SalesOrderDetail.SalesOrderID

JOIN Product

ON SalesOrderDetail.ProductID = Product.ProductID

GROUP BY SalesOrderHeader.SalesOrderID, SalesOrderHeader.SubTotal

* **SQL**

SELECT Product.Name,

SUM (SalesOrderDetail.OrderQty \* SalesOrderDetail.UnitPrice) AS Total\_Sale\_Value

FROM Product

JOIN SalesOrderDetail

ON Product.ProductID = SalesOrderDetail.ProductID

GROUP BY Product.Name

ORDER BY Total\_Sale\_Value DESC;

* **14**: **No se logró hacer**
* **15: No se logró hacer**

1. **Resit Questions**

* **SQL**

SELECT CompanyName

FROM Customer

JOIN SalesOrderHeader

ON Customer.CustomerID = SalesOrderHeader.CustomerID

WHERE CompanyName LIKE '%Good Toys%' OR CompanyName LIKE '%Bike World%'

* **SQL**

SELECT Product.Name,

SalesOrderDetail.OrderQty

FROM Customer JOIN SalesOrderHeader

ON Customer.CustomerID = SalesOrderHeader.CustomerID

JOIN SalesOrderDetail

ON SalesOrderHeader.SalesOrderID = SalesOrderDetail.SalesOrderID

JOIN Product

ON SalesOrderDetail.ProductID = Product.ProductID

WHERE Customer.CompanyName = 'Futuristic Bikes'

* **3. No se pudo realizar**
* **SQL**

SELECT Address.CountyRegion,

SUM(SubTotal)

FROM SalesOrderHeader

JOIN Address

ON SalesOrderHeader.ShipToAddressID = Address.AddressID

GROUP BY Address.CountyRegion

ORDER BY SUM(SubTotal) DESC

**PARTE TRES. Definiendo e implementando consultas gerenciales.**



SELECT Product.Name, Product.ListPrice, SalesOrderDetail.UnitPrice,

SalesOrderDetail.UnitPriceDiscount,SUM(SalesOrderHeader.Subtotal + SalesOrderHeader.TaxAmt+ SalesOrderHeader.Freight) AS WithTaxes,Customer.CustomerID

FROM Product

JOIN SalesOrderDetail

ON (Product.ProductID LIKE SalesOrderDetail.ProductID)

JOIN SalesOrderHeader

ON ( SalesOrderDetail.SalesOrderID LIKE SalesOrderHeader.SalesOrderID)

JOIN Customer

ON (Customer.CustomerID LIKE SalesOrderHeader.CustomerID)

1. Es la mejor consulta que se puede hacer ya que se le está pidiendo a la base de datos el nombre, su precio de venta, los detalles por unidad de cada producto, su total sumando el subtotal y los impuestos y respectivos y además se incluye información del cliente.

Proponemos ¿Qué tipo de producto se vende? ya que no fue incluida en la misión y es de gran importancia para el usuario, el cambio que se le daría al modelo sería el de incluir a la tabla ProductCategory con Product y su respectiva llave foránea la cual es ProductCategoryID.

1. SELECT Name,Color,ListPrice, Sze,Weight

FROM Product

**RETROSPECTIVA**

1. El tiempo total invertido en el laboratorio fue de 15 horas cada integrante del laboratorio.

2. Incompleto ya que algunas consultas no pudieron ser realizadas.

3. La práctica en SQL más avanzada.

4. Realizar el diagrama lógico mínimo y algunas consultas con términos no tan conocidos.

5. Trabajar en equipo, recolectar ideas. Nos comprometemos a ser más eficientes en el trabajo durante un tiempo determinado.