

University of Oklahoma

# THE BIG BANG

Derreck Landers, Brett Leamont, Munduu Munkhbileg, Nelly Mabicka, Tiandra Forrester

MIS 3353 - Database Management

Professor Clarke Daugherty

3/24/2021



Team Motto: "We will data your bases"

# Executive Summary

We started with making the Entity Relationship Diagram (ERD) to implement the outline for Sonner Tire company. In doing so, we laid out the pieces in order for them to make sense of the outline. Our team, The Big Bang, accepted feedback on behalf of Sonner Tire to better obtain the ERD's to match what they wanted and were looking for, for us to best implement their database. This step of the process took a lot of trial and error in order for our team and Sonner Tire to see our outline of the database. Once the feedback was put back into our Entity RelationShip Diagrams, we were able to start normalizing the data.

Normalizing the data, meant that as a team, we would go through the process used by developers to reduce redundancy and ensure well-structured relationships. This was done by making all columns atomic, eliminating any data duplication that could come up, and making all of them well-structured for our team to move on into the next steps. This was a tedious step in that we needed to go through each of our attributes in the Entity Relationship Diagram. Once done, it was important for us to implement this stage and meet with Sonner Tire once more to ensure that we were on the same page. It helped Sonner Tire with feeling secure that there would not be any confusion on the data they would be receiving when running their queries in the next step in the process. This is the entire point of this process, eliminate any confusion there may be in reading the database and make sure we do it right early on in this step, so that the queries we make in the next step flow with what we have done to this point. To follow, we received feedback on both the ERD and now normalization so that Sonner Tire and our group, The Big Bang, could move into creating the queries used to run their database.

Making the queries, our team used Walton Labs in order to design what we had outlined with the Entity Relationship Diagram and the normalization of the functions. We created queries for each of our Entities in our ERD's in order for Sonner Tire to receive the correct information, ranging from their customers and all the way to their vendors. It was now accessible in a database to be used how Sonner Tire intended it to be used. This was the final step before Sonner Tire would be ready to implement what our group, The Big Bang, had done with it and put it into their everyday workflow. Once completing the design of the queries, We made Sonner Tire aware of the time and pricing of the project.

Our pricing was addressed by the team via a time log throughout the project. Updating Sonner Tire throughout every step of the process in order for them to feel comfortable and understand our pricing as we moved through the project in different stages. After meeting with Sonner Tire once more and making sure all was how they wanted their new database to be, our end pricing was made. The total, for each group member combined in the project that worked in The Big Bang, was \$115,750. We thank Sonner Tire for this amazing opportunity in being part of making their company strive more in the future, and for the memories made along the way.

Contents	Page Numbers
Executive Summary .....	2
Get to Know the Team: (Team Name).....	5
Conceptual Design.....	6
The Client Meeting.....	6
Q&A During the Meeting & Information We Learned.....	7
Significant Assumptions.....	8
What is an ERD? Why is it necessary?.....	8
Changes Made To the ERD.....	9
Business Cycles Used. ....	8
Data Provided by Client.....	9-10
ERD Created.....	11
Query Feasibility and Current ERD.....	12-14
Logical Design.....	14
Normalization .....	15
Normalization of the Data Provided by the Client.....	15
Normalized Relations.....	15-19
Differences between ERD and Normalized Relations.....	20
Database Integrity.....	19
Physical Design and Implementation.....	19
Implemented Physical Design.....	21
Data Dictionary.....	21-22

Denormalization.....	22
Challenges Faced/Addressed During Implementation.....	23
Strengths and Weaknesses Encountered During Implementation.....	23
Specific SQL Statements Requested.....	24-30
Three Additional Queries.....	31-32
User Documentation.....	33-46
What We Learned Throughout This Process.....	47
Team Function/insights.....	48
Appendix.....	49
Team Contract.....	50
Data Dictionary Model.....	51-53
Project Management.....	54

## THE BIG BANG

Name	Major/Year	Background/ Experience	Photo
<b>Brett Learmont</b>	<b>MIS &amp; Finance, Junior</b>	I grew up in Plano, TX and have been living there ever since. I have interned with The Trade Group and NetApp.	
<b>Tiandra Forrester</b>	<b>MIS, Senior</b>	I grew up in Jamaica where I have spent the most of my life. I have interned with Valero Energy Corp and with the Tulsa health Department	
<b>Nelly Mabicka</b>	<b>MIS, Senior</b>	I grew up in Libreville/Gabon, a small country in central/west Africa. I have an upcoming summer internship with alliantgroup.	
<b>Munkhbileg Munkhburen</b>	<b>MIT/MIS, Senior</b>	I grew up in Mongolia Ulaanbaatar. I am doing an internship at Interactive as a machine learning engineer.	
<b>Derrick Landers</b>	<b>MIS, Senior</b>	I grew up in Tulsa and always dreamed of coming to OU. I have internship experience with QT IT and have an upcoming summer internship with BOK Financial	

## Conceptual Design

In the database design process, conceptual design is an organized representation of database concepts and how they relate. Its main outcome is a conceptual data model that outlines the data needed to support business processes, performance measures, events, and history. This model is created without the support of hardware specifications (storage, DBMS technology) and is used by business technology professionals to define varying parts of the database such as data types, names etc. The data model consists mainly of three parts: entities, attributes, and relationships. Entities typically represent real world objects and house the attributes or characteristics of that object. Relationships are the associations and dependencies that exist between different entities.

## The Client Meeting

The client meeting was held on Monday March 15, over zoom. We met as a group the night before to come up with valuable questions to ask the owner of Sooner Tire in order to receive feedback on it. This meeting was the basis for starting our project and moving forward with designing an ERD for the client.

- Meeting Time: 12:15 pm
- Location: Zoom
- Interviewers: Derrick Landers, Brett Learmont, Munduu Munkhbileg, Nelly Mabicka, Tiandra Forrester
- Interviewee: Clarke Daugherty

## **Q&A During the Meeting & Information We Learned**

**Question 1: How do you want us to connect each customer with their sales?**

We want to order by car type and not by the person's name.

**Question 2: I see they receive discounts for larger orders. What kind of discount are you offering and what you want for us to use in your database?**

**Are you wanting the same discounts for each tire manufacturer?**

Nothing is set in stone. Track discounts for purchases over a certain total and define them and apply them to the sales order.

**Question 3: Where did people have issues paying twice and what could we do to fix this?**

In this case, customers do not have a direct relationship with the finished sales order.

There needs to be a connection between the sales order and the vehicle that is being worked on.

**Question 4:Where is our customer base coming from and do we need to create our own customer base?**

We will randomly generate data with names, phone numbers, addresses, gender, etc.

**Question 5: Do we need to display insurance protection as a new sale or just a separate entity?**

Insurance protection will be displayed in the customer tab and in customer payment terms based on whether or not they are using the said insurance.

## Significant Assumptions

- 1) We are going to assume that we want to keep track of both female and male purchasing patterns. We added new entities in the customer on the ERD to track this.
- 2) We are also going to assume that the discount only applies to customers purchasing a whole new set of tires(4). We added this as an attribute in the Payment Terms of the customer.
- 3) We believe that category would be a super type with the given products and services the company has to offer.
- 4) We assume that the same model car will be driven by multiple people and to avoid data duplication we made a reference entity called model.
- 5) We assume that discounts will be given as part of the service that Sonner Tire is giving their customers.

## What is an ERD? Why is it necessary?

An ERD is a map that helps businesses track their inventory and lay out flow of information within their company. The ERD will allow for you to worry less about tracking the data and make it easier for you to find certain aspects that are internally happening within your Company. The goal in creating the Entity Relationship diagram, is for your team and your customer (The Big Bang and Sonner Tire) to find common ground in order to move forward to next steps in Creating the overall database for your client. It ensures comfortability between the two parties So that they understand what the database will do and see an outline of the projects end goal For when Sonner Tire is able to run their database.

## Business Cycles Used

We used the revenue, and the expenditure cycle because Sooner tires provides services to customers and to create revenues. The expenditure cycle was added because the company spends money by purchasing tires from manufacturers. Since Sonner Tire does not produce any tires we took out the production cycle.

## Changes made to ERD generic core

We removed production cycle from integrated cycle

We added Vendor entity

Every customer has cars, we added a car entity.

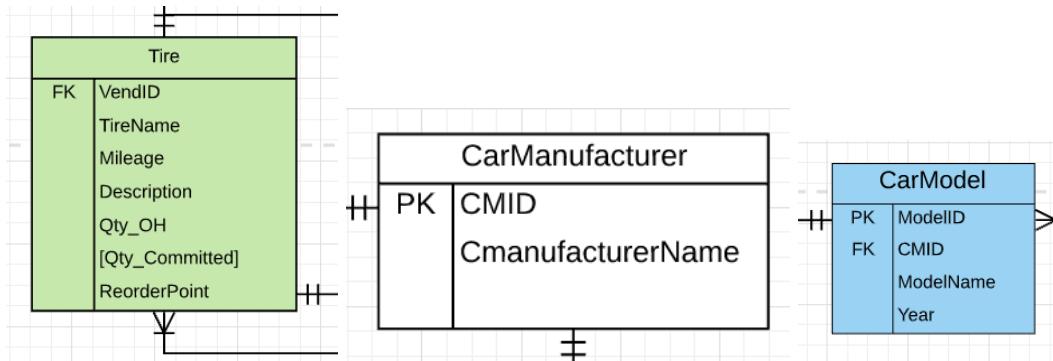
Receipt entity is added for making pick up easier.

Category/Service entity is added

We added subtypes/supertype

## Data Provided by Client

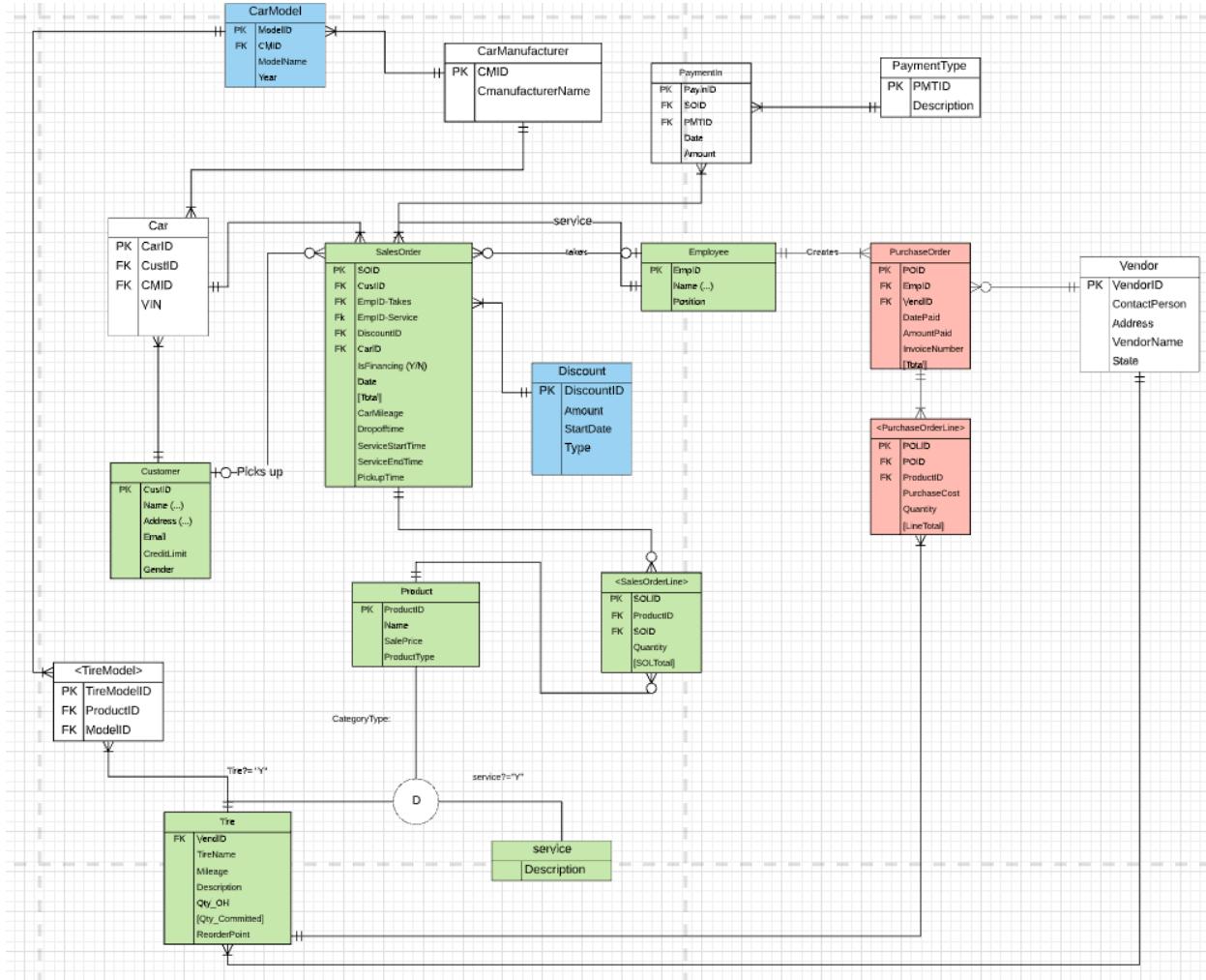
The data provided by Sooner Tire company was a great representation of how they have been keeping their records of business. The data includes the make/model of the car, and the tires on the car as well. Aside from that data, they also have data on each tire they hold in stock. Sorting it by the tire name and then giving details of the tires for which car works best for each car. Along with that they keep track of the mileage of the car and a description of the tires on their condition. This data at the bottom of the page will be kept in our ERD through the three tables below. Each manufacturer will have a CMID. Each model will have a ModelID and each tire will have a ProductID. All the other information will be in the attributes of these tables.



Car Manufacturer	Model	Tire
Ford	Escape 2019, 2018, 2017, 2016, 2015, 2014	235/55R17, 235/50R18, 235/50R19, 235/45R19
Ford	Mustang 2019, 2018, 2017, 2016, 2015, 2014	235/50R18, 255/40R19
Ford	Edge 2019, 2018, 2017, 2016, 2015, 2014	245/50R20, 265/40R21
Ford	Explorer 2019, 2018, 2017, 2016, 2015, 2014	255/50R20, 245/60R18, 235/65R18
Honda	Accord 2018, 2017, 2016, 2015, 2014	235/40R19, 225/50R17, 215/55R17
Honda	Civic 2019, 2018, 2017, 2016, 2015,	215/50R17, 215/55R16, 205/55R16
Honda	Odyssey 2019, 2018, 2017, 2016, 2015, 2014	235/65R17, 235/65R18, 235/60R18
Honda	Pilot 2018, 2017, 2016, 2015, 2014	245/50R20, 245/65R17
Chevrolet	Malibu 2019, 2018, 2017, 2016, 2015	205/65R16, 215/60R16
Chevrolet	Impala 2018, 2017, 2016, 2015, 2014	235/50R18, 245/45R19, 225/60R16
Chevrolet	Tahoe 2017, 2016, 2015, 2014	265/70R17, 265/65R18, 275/55R20
Chevrolet	Silverado 2019, 2018, 2017, 2016, 2015	255/70R17, 265/70R17
Toyota	Camry 2018, 2017, 2016, 2015	205/65R26, 215/55R17
Toyota	Corolla 2017, 2016, 2015, 2014, 2013	195/65R15, 205/55R16
Toyota	4Runner 2018, 2017, 2016, 2015	215/45R17, 245/60R20, 205/55R16
Toyota	Sequoia 2018, 2017, 2016, 2015, 2014	275/65R18, 275/55R20

# ERD Created

This is the Entity-Relationship Diagram we will be using for Sonner Tire.



[https://lucid.app/lucidchart/invitations/accept/inv\\_be704fc-a7a7-4ab8-82df-32b7393ab43c](https://lucid.app/lucidchart/invitations/accept/inv_be704fc-a7a7-4ab8-82df-32b7393ab43c)

## Query Feasibility and Current ERD

#	Query Question	Tables needed to run the query	Projected SQL Statement
1.	Total sales (in dollars) by region for a given tire manufacturer and car manufacturer. It would be great if we can specify the car model and year too (note that we would like to be able to input the month to be calculated).	CarManufacturer, CarModel, Car, SalesOrder TireModel, Tire, Vendor	<pre>select CMID, CModelID, CModelName, CYear, TVendID, Sum(SOLQuantity * PSalesPrice) as TotalSales from TCarModel tcm join TCarManufacturer tcmf on tcm.CCMID = tcmf.CMID join TCar tc on tcmf.CMID = tc.CarCMID join TSalesOrder tso on tso.SOCarID = tc.CarID join TSalesOrderLine tsol on tso.SOID = tsol.SOLOID join TProduct tp on tp.ProdID = tsol.SOLProductID join TTire tt on tp.ProdID = tt.TProductID group by CMID, cmodelid, CModelName, cyear, TVendID</pre>
2.	2. Total sales (in dollars) by a customer in a given year.	Customer, SalesOrder	<pre>SELECT [Total] FROM Customer C JOIN SalesOrder SO ON C.CustID = SO.CustID WHERE Customer = 'name'</pre>
3.	3. The five highest selling tires.	SalesOrder, <SalesOrderLine>, Product, Tire	<pre>SELECT TireName FROM SalesOrder SO JOIN SalesOrderLine SOL ON SO.SOID = SOL.SOID JOIN Product P ON SOL.ProductID = P.ProductID JOIN Tire T ON P.ProductID = T.ProductID WHERE ???</pre>
4.	4. Itemized invoices for jobs for each customer that need to include tires purchased/tire rotation/tire repair/tire protection.	SalesOrder, <SalesOrderLine>, Product, Tire, Service	<pre>SOCustID, PProductType, SDescription FROM SalesOrder SO JOIN SalesOrderLine sol on so.SOID = sol.SOLID join TProduct tp on tp.ProdID = tsol.SOLProductID join TService ts on ts.SProdID = tp.SProdID</pre>
5.	5. The number and type of job performed by each of our employees.	Employee	<pre>SELECT EmpID, Position, COUNT(Position) FROM Employee</pre>
6.	6. Number of times a tire protection has been purchased for a particular tire and number of	Product, SalesOrderLine SalesOrder	<pre>select ProdID, PName, count(prodid) NoPurchases</pre>

	times free service has been applied (free tire damage repair, free replacement).	Discount	from TSalesOrderLine tsol join TProduct tp on tsol.SOLProductID = tp.ProdID where PProductType = 'Service' group by prodid, pname
<b>7.</b>	The following items for Purchase Orders: manufacturer name, number of POs, total cost.	CarManufacturer, CarModel, <TireModel>, Tire, Vendor, PurchaseOrder	select VendorName, count(POID) as totalPO, sum(POTotal) as TotalSales from TPurchaseOrder tpo join TVendor tv on tpo.POVendorID = tv.VendorID group by VendorName
<b>8.</b>	Number of orders and total sales per customer in the past 2 years. This report is particularly important as it shows the number of returning customers.	SalesOrder, Customer	select SOCustID, SOID, sum(SOTotal) as Total, year(SODate) as Year from TSalesOrder group by SOCustID, year(SODate), SOID
<b>9.</b>	List of tires that have not been purchased within the last 6 months (in order to better manage inventory).	SalesOrder, Salesorderline, product	SELECT ProdID, PName, SODate FROM TSalesOrder tso join TSalesOrderLine tsol on tso.SOId = tsol.SOLSId left join TProduct tp on tp.ProdID = tsol.SOLProductID where SODate not between '2020-11-02' and '2021-05-02' and PProductType = 'Tire'
<b>10.</b>	Names of customers who took advantage of the financing option, date purchased, total amount purchased, credit limit, number of payments made, the total amount paid, outstanding amount, is time to pay-off less than 6 months, all displayed from the latest date and then the largest amount owed.	Customer, SalesOrder PaymentIn PaymentType	select CustFName, CustLName, Credit_limit, SOIsFinance, PayDate, PayAmount, PMTDescription from TCustomer tc join TSalesOrder tso on tc.CustID = tso.SOCustID join TPaymentIn tpi on tso.SOId = tpi.PayInID join TPaymentType tpt on tpi.PayInPMTID = tpt.PMTID
<b>11.</b>	Total profit per tire type and manufacturer type in the past 6 months.	CarManufacturer, CarModel, Car, SalesOrder	select TireName, TVendID, sum(SOTotal) as TotalSales, SODate from tsalesorder tso join TSalesOrderLine tsol on tso.SOId = tsol.SOLSId join TProduct tp on tp.ProdID = tsol.SOLProductID join TTire ttr on ttr.TProductID = tp.ProdID where SODate between '2020-10-25' and '2021-04-25' group by TireName, TVendID, SODate
<b>12.</b>	List of all customers that have not made a purchase within the last 12 months from the current date.	Customer, SalesOrder	SELECT C.FirstName, C.LastName FROM Customer C JOIN SalesOrder SO ON C.CustID = SO.CustID WHERE Date BETWEEN('4/26/2020,

			4/26/2021) AND Total = 0
<b>13.</b>	List of customers whose average sales is less than the average of all sales. This will help us to find customers whom we should target to get a higher volume of sales.	Customer, SalesOrder	<pre> select custfname, custlname, avg(SOLQuantity*Psalespriceint) AvgPurchase from TCustomer tc join TSalesOrder tso on tc.CustID = tso.SOCustID join TSalesOrderLine tsol on tsol.SOLSOID = tso.SOID join TProduct tp on tp.ProdID =tsol.SOLProductID group by CustFName, CustLName having avg(SOLQuantity*Psalespriceint) &lt; (select avg(SOLQuantity * PSalesPriceint) from TSalesOrderLine tsol join TProduct tp on tp.ProdID =tsol.SOLProductID) </pre>

## Logical Design

The end goal for logical design is to reduce redundancy and increase efficiency. Logical design is good for new construction and remodeling and it will make it easier to understand as well. Without Logical Design being implemented, there would be confusion on the data redundancy and our model would not operate correctly.

## Normalization

Normalization is the process used by developers to reduce redundancy and ensure well-structured relationships by making all columns atomic, eliminating data duplication, issues, and making them well-structured. This process ensures “data integrity” and efficiency through the report shared by the database. It is important to include normalization in this project to have a better control of product received, to ensure the accurate shipment by vendors and eliminate data duplication issues.

# Normalization of the Data Provided by the Client

Attached Excel Document

TMakelD, MakeName)

TCarModel(ModelID, ModelMakeID, ModelName, ModelYear)

*Foreign Key ModelMakeID references TMakelD*

*NotNull*

*On Delete Restrict*

TTire(TireID, TMakelD, TGoodFor, TModelID, TMilage, TDescription, TCost, TSalePrice)

*Foreign Key TMakelD references TMakelD*

*NotNull*

*On Delete Restrict*

TModelTire(MTID, MTModelID, MTProductID-Tire)

*Foreign Key MTModelID references TCarModel*

*NotNull*

*On Delete Restrict*

*Foreign Key MTProductID-Tire references TTire*

*NotNull*

*On Delete Restrict*

TVendor(VendID, VendVendorName\_VendContactPerson, VendAddress, VendCity, VendZip, VendState)

Normalized means that all values in a given table are atomic. There can be no multivalued columns with more than one item in each and so it eliminates any data duplication. Turning the data into 3NF.

## Normalized Relations

TCarManufacturer(CMid, CManufacturerName)

TCarModel(CModelID, CCMid, CModelName, CYear)

Foreign Key CCMid references TCarManufacturer

Not Null

On Delete Restrict

TState(StateCode, StateName)

TZipCode(ZipCode, ZipCity, ZipStateCode)

Foreign Key ZipStateCode references TState

Not Null

On Delete Restrict

TCustomer(CustID, CustLName, CustFName, CustStreetAdd, CustZipCode)

Foreign Key CustZipCode references TZipCode

Not Null

On Delete Restrict

TProduct(ProductID, PName, PSalePrice, PProductType)

TTire(TProductID, TVendID, TPOLID, TQty\_OH, TLocation, TReorderPoint)

Foreign Key TTireID references TProduct

Not Null

On Delete Cascade

Foreign Key TVendID references TVendor

Not Null

On Delete Restrict

Foreign Key TPOLID references TPurchaseOrderLine

Not Null

On Delete Restrict

TService(SProductID, SType, STime)

Foreign Key SProductID references TProduct

Not Null

On Delete Cascade

TTireModel(TireModelID, TiTireID,TiModelID )

Foreign Key TiTireID references TTire

Not Null

On Delete Restrict

Foreign Key TireModelID references TCarModel

Not Null

On Delete Restrict

TEmployee(EmpID, EmpFName, EmpLName, EmpPosition)

TPaymentIn(PayInID, PayEmpID, PayDate, PayAmount, PayPmtInfo)

Foreign Key PayEmpID references TEmployee

Not Null

On Delete Restrict

TDiscount(DiscountID, DiscountType)

TSalesOrder(SOID, SOPayInID, SOCustID, SOEmpID-Takes, SOEmpID-TechnicianName ,SODiscountID, SOCarID, SODate, SOTechnicianName, SOStatus, SOTotal, SOCARMileage, SODropofftime, SOTimeinService, SOPickupTime)

Foreign Key SOPayInID references TPaymentIn

Not Null

On Delete Restrict

Foreign Key SOCustID references TCustomer

Not Null

On Delete Restrict

Foreign Key SOEmpID-Takes references TEmployee

Not Null

On Delete Restrict

Foreign Key SOEmpID-TechnicianName references TEmployee

Not Null

On Delete Restrict

Foreign Key SODiscountID references TDiscount

Not Null

On Delete Restrict

Foreign Key SOCarID references TCar

Not Null

On Delete Restrict

TSalesOrderLine(SOLID, STireID, SSOID, SQuantity, SSOLStatus, SSaleOrReturn)

Foreign Key STireID references TTire

Not Null

On Delete Restrict

Foreign Key SSOID references TSalesOrder

Not Null

On Delete Restrict

TCar(CarID, CarCustID, CarModelID, CarSOID, CarColor)

Foreign Key CarCustID references TCustomer

Not Null

On Delete Restrict

Foreign Key CarModelID references TCarModel

Not Null

On Delete Restrict

Foreign Key CarSOID references TSalesOrder

Not Null

On Delete Restrict

TRegion(RegionID, RDescription)

TVendor(VendID, VendVendorName, VendZipCode, VendRegionID)

Foreign Key VendZipCode references TZipCode

Not Null

On Delete Restrict

Foreign Key VendRegionID references TRegion

Not Null

On Delete Restrict

TPurchaseOrder(POID, POEmpID, POVendID, PODatePaid, POAmountPaid, POInvoiceNumber)

Foreign Key POEmpID references TEmployee

Not Null

On Delete Restrict

Foreign Key POVendID references TVendor

Not Null

On Delete Restrict

TPurchaseOrderLine(POLID, POLPOID, POLTireID, POLQuantity)

Foreign Key POLPOID references TPurchaseOrder

Not Null

On Delete Restrict

Foreign Key POLProdID references TTireID

Not Null

On Delete Restrict

## Differences between ERD and Normalized Relations

The major differences between an Entity Relationship diagram and normalization is showcasing your logical entries in a diagram as in an ERD vs cleaning up your logical design to take care of any data duplication errors you may run into. The benefits of making your diagram normalized and having normal functions is so when you query your ERD's, there won't be any errors or confusion about which value pertains to which value you are asking for. Normalizing the data breaks up composite and multivalued attributes to allow for an easier creation and implementation of the data.

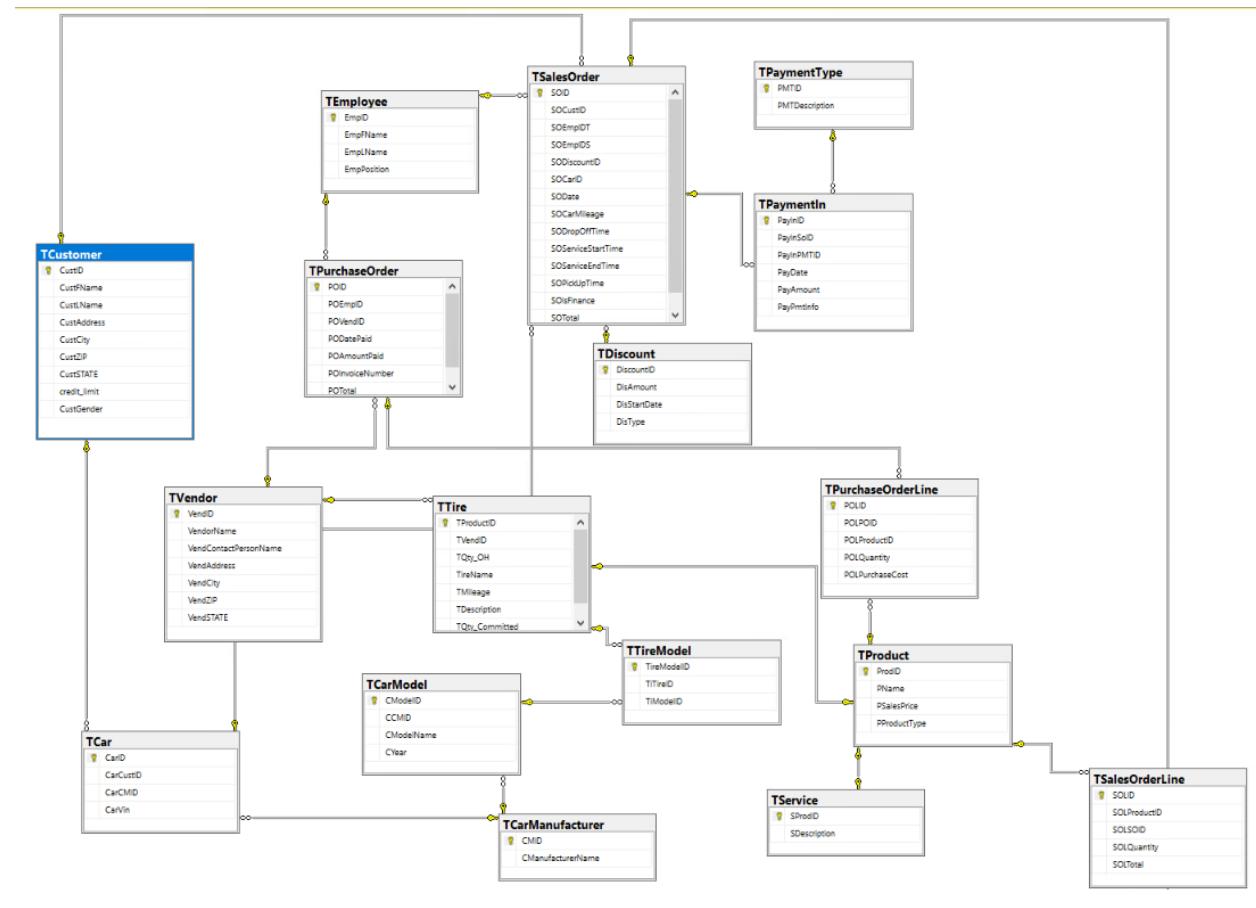
## Database Integrity

The entity integrity rule states that the primary key of a table can't be null, meaning everything in the column must have a value, but the table can contain a value that is null, it just has to be outside of the primary key column. The domain integrity rule states that an attribute must have valid values. For instance, if there is an integer value section for age and you put anything other than a number in that column, it will violate this rule. Lastly, the referential integrity rule is a constraint placed within two tables of data. This can be recognized by if a foreign key is referencing a primary key in another data set. Then you must be able to find the same values in both the foreign key referencing the primary key and the primary key that is being referenced.

# Physical Design and Implementation

First, it is important to mention that the physical design and implementation are platform specific, meaning that they depend on where (what country) and what service (MySQL Vs Oracle) is used. According to the book, “the purpose of the physical design and implementation is to identify the best way to integrate the database design into a particular database management system.” Again, the main goal of the physical design is to create a database that works fast. To create the physical design process for this project, we converted the data collected during the logical design into a physical database. We translate the relations into tables, columns into fields, and rows into records.

## Implemented Physical Design



# Data Dictionary

A data dictionary is one of the most important outputs of the physical design process. It depicts the decision made about how the database is to be implemented and it gives more information to explain how the fields are to be used. It is a useful tool because it serves as a guide during implementation, and after implementation. Creating a data dictionary will provide significant help to the developer who will implement the database and for those who will work on it later. It will also assist those performing maintenance and support work for the database. The data types were configured according to two different types of data. Those two types of data included VarChar and an Int data type. VarChar is a string that will give the data back as words. The Int data type is an integer, allowing for numbered values to be returned. This is how we deciphered what to put under each data type and why we chose one over the other.

	Field Name	PK?	Date Type	Size	Null	References(Foreign Key)	Sample
TCustomer	CustID	Y	Int(Auto Increment)		not null		1-100
	Name	N	VarChar		not null		
	Address	N	VarChar		not null		
	Email	N	VarChar		not null		
	CreditLimit	N	int		not null		
	Gender	N	VarChar		not null		
TCar	CarID	Y	Int(Auto Increment)		not null		1-100
	CustID	N	Int(Auto Increment)		not null	TCustomer	
	CMID	N	Int(Auto Increment)		not null	TCarManufacturer	
	VIN	N	int		not null		
TCarManufacturer	CMID	Y	Int(Auto Increment)		not null		1-100
	CManufacturerName	N	VarChar		not null		
TProduct	ProductID	Y	Int(Auto Increment)		not null		1-106
	PName	N	varchar		not null		
	PSalePrice	N	int		not null		
	PProductType	N	varchar		not null		
TTire	TProductID	Y	Int(Auto Increment)		not null	TProduct	5-106
	TVendID	N	int		not null	TVendor	
	TQty_OH	N	int		not null		
	TQty_committed	N	int		not null		
	TireName	N	varchar		not null		
	TMileage	N	int		not null		
	TDescription	N	varchar		not null		
TService	SProductID	Y	Int(Auto Increment)		not null	TProduct	2-5
	SDescription	N	varchar		not null		
TTireModel	TireModelID	Y	Int(Auto Increment)		not null		1000-1100
	TiTireID	N	int		not null	TTire	
	TiModellID	N	int		not null	TCarModel	
TVendor	VendorID	Y	int(Auto Increment)		not null		1-16
	VendorName	N	VarChar		not null		

## Denormalization

Contrary to normalization, denormalization is simply a reduction of normalization. It is the process of allowing data duplication. Usually, denormalization is done to allow a database to process the queries faster. We normalized the address when we normalized the data; however, in our database we denormalized addresses by eliminating the city and only keeping state and zip in order to cut down on the time spent searching for specific customer data.

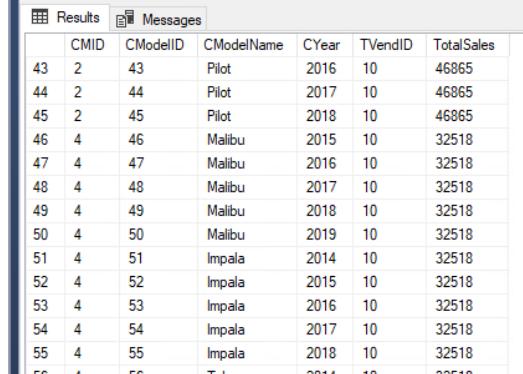
## Challenges Faced/Addressed During Implementation

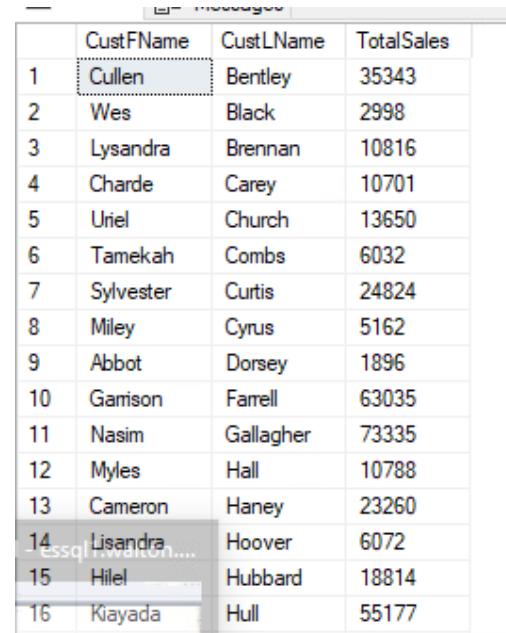
During the physical design and implementation process, there were three main challenges that we faced. First, we experienced difficulties in learning how to use the database system given for this project. This was the result of user inexperience when working with Walton's DBMS and prior to this course our team had little experience with working with databases in general. To address this problem, we practice working Walton DBMS by creating and executing inquiries, using the student implementation guide given in class. The second challenge that we faced was fixing and editing our ERD in preparation for physical design and implementation. This portion was particularly challenging because we realized that there were several parts missing from our ERD. To address this problem, we presented our ERD to our instructor, received his feedback and implemented the needed changes. Lastly, we had some challenges with creating the data dictionary for the project. More specifically we had trouble with identifying what information we needed and how it should be presented. For direction on creating the data dictionary, we referred to the example given to us in our class resources and during class lectures. Using this guide, we were able to successfully create our own data dictionary.

## Strengths and Weaknesses Encountered During Implementation

Overall, as a team we all complimented each other well in regard to our strengths and weaknesses. Some of us really excelled in areas that others were not quite up to par and that lead to a very smooth implementation. The area that we would consider a weakness was the building of our ERD as it seemed like a guessing game at times; however, after putting our minds together, it was finished properly. Our strengths definitely involved the normalization of the data and putting our information into SQLServer. As a team we worked very well in helping others weaknesses with our own individual strengths.

## Specific SQL Statements Requested

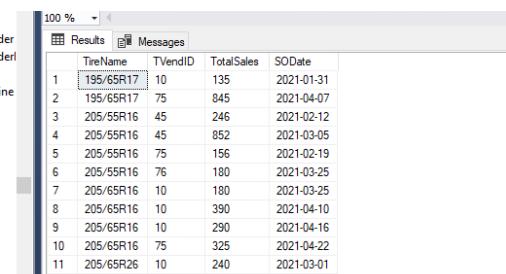
Query #	Question	SQL	Partial Output																																																																																										
1	Total sales (in dollars) by region for a given tire manufacturer and car manufacturer. It would be great if we can specify the car model and year too (note that we would like to be able to input the month to be calculated).	<pre>select CMID, CModelID, CModelName, CYear, TVendID, Sum(SOLQuantity * PSalesPriceint) as TotalSales from TCarModel tcm join TCarManufacturer tcmf on tcm.CCMID = tcmf.CMID join TCar tc on tcmf.CMID = tc.CarCMID join TSalesOrder tso on tso.SOCarID = tc.CarID</pre>	 <table border="1"> <thead> <tr> <th>CMID</th> <th>CModelID</th> <th>CModelName</th> <th>CYear</th> <th>TVendID</th> <th>TotalSales</th> </tr> </thead> <tbody> <tr><td>43</td><td>2</td><td>Pilot</td><td>2016</td><td>10</td><td>46865</td></tr> <tr><td>44</td><td>2</td><td>Pilot</td><td>2017</td><td>10</td><td>46865</td></tr> <tr><td>45</td><td>2</td><td>Pilot</td><td>2018</td><td>10</td><td>46865</td></tr> <tr><td>46</td><td>4</td><td>Malibu</td><td>2015</td><td>10</td><td>32518</td></tr> <tr><td>47</td><td>4</td><td>Malibu</td><td>2016</td><td>10</td><td>32518</td></tr> <tr><td>48</td><td>4</td><td>Malibu</td><td>2017</td><td>10</td><td>32518</td></tr> <tr><td>49</td><td>4</td><td>Malibu</td><td>2018</td><td>10</td><td>32518</td></tr> <tr><td>50</td><td>4</td><td>Malibu</td><td>2019</td><td>10</td><td>32518</td></tr> <tr><td>51</td><td>4</td><td>Impala</td><td>2014</td><td>10</td><td>32518</td></tr> <tr><td>52</td><td>4</td><td>Impala</td><td>2015</td><td>10</td><td>32518</td></tr> <tr><td>53</td><td>4</td><td>Impala</td><td>2016</td><td>10</td><td>32518</td></tr> <tr><td>54</td><td>4</td><td>Impala</td><td>2017</td><td>10</td><td>32518</td></tr> <tr><td>55</td><td>4</td><td>Impala</td><td>2018</td><td>10</td><td>32518</td></tr> <tr><td>56</td><td>4</td><td>Tahoe</td><td>2014</td><td>10</td><td>32518</td></tr> </tbody> </table>	CMID	CModelID	CModelName	CYear	TVendID	TotalSales	43	2	Pilot	2016	10	46865	44	2	Pilot	2017	10	46865	45	2	Pilot	2018	10	46865	46	4	Malibu	2015	10	32518	47	4	Malibu	2016	10	32518	48	4	Malibu	2017	10	32518	49	4	Malibu	2018	10	32518	50	4	Malibu	2019	10	32518	51	4	Impala	2014	10	32518	52	4	Impala	2015	10	32518	53	4	Impala	2016	10	32518	54	4	Impala	2017	10	32518	55	4	Impala	2018	10	32518	56	4	Tahoe	2014	10	32518
CMID	CModelID	CModelName	CYear	TVendID	TotalSales																																																																																								
43	2	Pilot	2016	10	46865																																																																																								
44	2	Pilot	2017	10	46865																																																																																								
45	2	Pilot	2018	10	46865																																																																																								
46	4	Malibu	2015	10	32518																																																																																								
47	4	Malibu	2016	10	32518																																																																																								
48	4	Malibu	2017	10	32518																																																																																								
49	4	Malibu	2018	10	32518																																																																																								
50	4	Malibu	2019	10	32518																																																																																								
51	4	Impala	2014	10	32518																																																																																								
52	4	Impala	2015	10	32518																																																																																								
53	4	Impala	2016	10	32518																																																																																								
54	4	Impala	2017	10	32518																																																																																								
55	4	Impala	2018	10	32518																																																																																								
56	4	Tahoe	2014	10	32518																																																																																								

		<pre> join TSalesOrderLine tsol on tso.SOID = tsol.SOLSOID join TProduct tp on tp.ProdID = tsol.SOLProductID join TTire tt on tp.ProdID = tt.TProductID group by CMID, cmodelid, CModelName, cyear, TVendID </pre>																																																																					
2	Total sales (in dollars) by a customer in a given year.	<pre> Select CustFName, CustLName, sum(SOLQuantity * psalespriceint) as TotalSales from TCustomer tcr join TCar tc on tcr.CustID = tc.CarCustID join TSalesOrder tso on tso.SOCarID = tc.CarID join TSalesOrderLine tsol on tso.SOID = tsol.SOLSOID join TProduct tp on tp.ProdID = tsol.SOLProductID join TTire tt on tp.ProdID = tt.TProductID group by CustFName, CustLName </pre>	 <table border="1"> <thead> <tr> <th></th> <th>CustFName</th> <th>CustLName</th> <th>TotalSales</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Cullen</td> <td>Bentley</td> <td>35343</td> </tr> <tr> <td>2</td> <td>Wes</td> <td>Black</td> <td>2998</td> </tr> <tr> <td>3</td> <td>Lysandra</td> <td>Brennan</td> <td>10816</td> </tr> <tr> <td>4</td> <td>Charde</td> <td>Carey</td> <td>10701</td> </tr> <tr> <td>5</td> <td>Uriel</td> <td>Church</td> <td>13650</td> </tr> <tr> <td>6</td> <td>Tamekah</td> <td>Combs</td> <td>6032</td> </tr> <tr> <td>7</td> <td>Sylvester</td> <td>Curtis</td> <td>24824</td> </tr> <tr> <td>8</td> <td>Miley</td> <td>Cyrus</td> <td>5162</td> </tr> <tr> <td>9</td> <td>Abbot</td> <td>Dorsey</td> <td>1896</td> </tr> <tr> <td>10</td> <td>Garrison</td> <td>Farrell</td> <td>63035</td> </tr> <tr> <td>11</td> <td>Nasim</td> <td>Gallagher</td> <td>73335</td> </tr> <tr> <td>12</td> <td>Myles</td> <td>Hall</td> <td>10788</td> </tr> <tr> <td>13</td> <td>Cameron</td> <td>Haney</td> <td>23260</td> </tr> <tr> <td>14</td> <td>Lisandra</td> <td>Hoover</td> <td>6072</td> </tr> <tr> <td>15</td> <td>Hilel</td> <td>Hubbard</td> <td>18814</td> </tr> <tr> <td>16</td> <td>Kiayada</td> <td>Hull</td> <td>55177</td> </tr> </tbody> </table>		CustFName	CustLName	TotalSales	1	Cullen	Bentley	35343	2	Wes	Black	2998	3	Lysandra	Brennan	10816	4	Charde	Carey	10701	5	Uriel	Church	13650	6	Tamekah	Combs	6032	7	Sylvester	Curtis	24824	8	Miley	Cyrus	5162	9	Abbot	Dorsey	1896	10	Garrison	Farrell	63035	11	Nasim	Gallagher	73335	12	Myles	Hall	10788	13	Cameron	Haney	23260	14	Lisandra	Hoover	6072	15	Hilel	Hubbard	18814	16	Kiayada	Hull	55177
	CustFName	CustLName	TotalSales																																																																				
1	Cullen	Bentley	35343																																																																				
2	Wes	Black	2998																																																																				
3	Lysandra	Brennan	10816																																																																				
4	Charde	Carey	10701																																																																				
5	Uriel	Church	13650																																																																				
6	Tamekah	Combs	6032																																																																				
7	Sylvester	Curtis	24824																																																																				
8	Miley	Cyrus	5162																																																																				
9	Abbot	Dorsey	1896																																																																				
10	Garrison	Farrell	63035																																																																				
11	Nasim	Gallagher	73335																																																																				
12	Myles	Hall	10788																																																																				
13	Cameron	Haney	23260																																																																				
14	Lisandra	Hoover	6072																																																																				
15	Hilel	Hubbard	18814																																																																				
16	Kiayada	Hull	55177																																																																				

3	The five highest selling tires.	<pre>select top 5 ProdID, PName, count(solid) TimesSold from TSalesOrderLine tsol join TProduct tp on tp.ProdID = tsol.SOLProductID group by ProdID, PName order by count(solid) desc</pre>	<table border="1"> <thead> <tr> <th></th><th>ProdID</th><th>PName</th><th>TimesSold</th></tr> </thead> <tbody> <tr> <td>1</td><td>69</td><td>235/40R19</td><td>4</td></tr> <tr> <td>2</td><td>90</td><td>245/60R20</td><td>4</td></tr> <tr> <td>3</td><td>75</td><td>235/45R19</td><td>3</td></tr> <tr> <td>4</td><td>64</td><td>215/60R16</td><td>3</td></tr> <tr> <td>5</td><td>96</td><td>275/65R18</td><td>3</td></tr> </tbody> </table>		ProdID	PName	TimesSold	1	69	235/40R19	4	2	90	245/60R20	4	3	75	235/45R19	3	4	64	215/60R16	3	5	96	275/65R18	3																																																																																																																
	ProdID	PName	TimesSold																																																																																																																																								
1	69	235/40R19	4																																																																																																																																								
2	90	245/60R20	4																																																																																																																																								
3	75	235/45R19	3																																																																																																																																								
4	64	215/60R16	3																																																																																																																																								
5	96	275/65R18	3																																																																																																																																								
4	Itemized invoices for jobs for each customer that need to include tires purchased/tire rotation/tire repair/tire protection.	<pre>select CustID, SOID, CarID, SODate, TQty_Committed, ProdID, PName, PSalesPrice, (SOLQuantity * psalespriceint) as LineTotal from TCustomer tcr join TCar tc on tcr.CustID = tc.CarCustID join TSalesOrder tso on tso.SOCarID = tc.CarID join TSalesOrderLine tsol on tso.SOID = tsol.SOLSOID join TProduct tp on tp.ProdID = tsol.SOLProductID</pre>	<table border="1"> <thead> <tr> <th>CustID</th><th>SOID</th><th>CarID</th><th>SODate</th><th>ProdID</th><th>PName</th><th>PSalesPrice</th><th>LineTotal</th></tr> </thead> <tbody> <tr> <td>1</td><td>30</td><td>6</td><td>1035</td><td>2020-12-30</td><td>90</td><td>245/60R20</td><td>104.5</td></tr> <tr> <td>2</td><td>21</td><td>18</td><td>1011</td><td>2021-01-02</td><td>2</td><td>Install Tire</td><td>20</td></tr> <tr> <td>3</td><td>64</td><td>23</td><td>1007</td><td>2020-12-26</td><td>100</td><td>255/70R17</td><td>91.5</td></tr> <tr> <td>4</td><td>25</td><td>14</td><td>1065</td><td>2021-03-10</td><td>69</td><td>235/40R19</td><td>104.5</td></tr> <tr> <td>5</td><td>1</td><td>76</td><td>1081</td><td>2021-04-22</td><td>77</td><td>235/45R19</td><td>89.5</td></tr> <tr> <td>6</td><td>69</td><td>27</td><td>1068</td><td>2021-04-11</td><td>63</td><td>215/60R16</td><td>83.5</td></tr> <tr> <td>7</td><td>55</td><td>70</td><td>1012</td><td>2021-01-17</td><td>69</td><td>235/40R19</td><td>104.5</td></tr> <tr> <td>8</td><td>31</td><td>95</td><td>1003</td><td>2021-02-23</td><td>97</td><td>275/65R18</td><td>86.25</td></tr> <tr> <td>9</td><td>82</td><td>40</td><td>1019</td><td>2021-02-16</td><td>18</td><td>245/50R20</td><td>89.25</td></tr> <tr> <td>10</td><td>75</td><td>20</td><td>1008</td><td>2021-04-15</td><td>79</td><td>235/50R18</td><td>81.5</td></tr> <tr> <td>11</td><td>99</td><td>100</td><td>1032</td><td>2020-12-25</td><td>100</td><td>255/70R17</td><td>91.5</td></tr> <tr> <td>12</td><td>95</td><td>37</td><td>1086</td><td>2021-04-10</td><td>48</td><td>205/65R16</td><td>82.5</td></tr> <tr> <td>13</td><td>75</td><td>19</td><td>1008</td><td>2021-04-17</td><td>7</td><td>235/50R19</td><td>107.50</td></tr> <tr> <td>14</td><td>75</td><td>73</td><td>1008</td><td>2021-02-19</td><td>71</td><td>235/40R19</td><td>85.5</td></tr> <tr> <td>15</td><td>99</td><td>56</td><td>1032</td><td>2021-03-25</td><td>39</td><td>275/65R18</td><td>80.25</td></tr> <tr> <td>16</td><td>71</td><td>36</td><td>1023</td><td>2020-12-25</td><td>61</td><td>255/40R19</td><td>86.25</td></tr> </tbody> </table>	CustID	SOID	CarID	SODate	ProdID	PName	PSalesPrice	LineTotal	1	30	6	1035	2020-12-30	90	245/60R20	104.5	2	21	18	1011	2021-01-02	2	Install Tire	20	3	64	23	1007	2020-12-26	100	255/70R17	91.5	4	25	14	1065	2021-03-10	69	235/40R19	104.5	5	1	76	1081	2021-04-22	77	235/45R19	89.5	6	69	27	1068	2021-04-11	63	215/60R16	83.5	7	55	70	1012	2021-01-17	69	235/40R19	104.5	8	31	95	1003	2021-02-23	97	275/65R18	86.25	9	82	40	1019	2021-02-16	18	245/50R20	89.25	10	75	20	1008	2021-04-15	79	235/50R18	81.5	11	99	100	1032	2020-12-25	100	255/70R17	91.5	12	95	37	1086	2021-04-10	48	205/65R16	82.5	13	75	19	1008	2021-04-17	7	235/50R19	107.50	14	75	73	1008	2021-02-19	71	235/40R19	85.5	15	99	56	1032	2021-03-25	39	275/65R18	80.25	16	71	36	1023	2020-12-25	61	255/40R19	86.25
CustID	SOID	CarID	SODate	ProdID	PName	PSalesPrice	LineTotal																																																																																																																																				
1	30	6	1035	2020-12-30	90	245/60R20	104.5																																																																																																																																				
2	21	18	1011	2021-01-02	2	Install Tire	20																																																																																																																																				
3	64	23	1007	2020-12-26	100	255/70R17	91.5																																																																																																																																				
4	25	14	1065	2021-03-10	69	235/40R19	104.5																																																																																																																																				
5	1	76	1081	2021-04-22	77	235/45R19	89.5																																																																																																																																				
6	69	27	1068	2021-04-11	63	215/60R16	83.5																																																																																																																																				
7	55	70	1012	2021-01-17	69	235/40R19	104.5																																																																																																																																				
8	31	95	1003	2021-02-23	97	275/65R18	86.25																																																																																																																																				
9	82	40	1019	2021-02-16	18	245/50R20	89.25																																																																																																																																				
10	75	20	1008	2021-04-15	79	235/50R18	81.5																																																																																																																																				
11	99	100	1032	2020-12-25	100	255/70R17	91.5																																																																																																																																				
12	95	37	1086	2021-04-10	48	205/65R16	82.5																																																																																																																																				
13	75	19	1008	2021-04-17	7	235/50R19	107.50																																																																																																																																				
14	75	73	1008	2021-02-19	71	235/40R19	85.5																																																																																																																																				
15	99	56	1032	2021-03-25	39	275/65R18	80.25																																																																																																																																				
16	71	36	1023	2020-12-25	61	255/40R19	86.25																																																																																																																																				
5	The number and type of job performed by each of our employees.	<pre>select EmpID, EmpFName, EmpLName, count(SOEmpIDS) as Manager, count(soempidt) as Technician from TEmployee te join TSalesOrder tso</pre>	<table border="1"> <thead> <tr> <th>EmpID</th><th>EmpFName</th><th>EmpLName</th><th>Manager</th><th>Technician</th></tr> </thead> <tbody> <tr> <td>1</td><td>Greg</td><td>Fracker</td><td>12</td><td>12</td></tr> <tr> <td>2</td><td>Wesley</td><td>Heiney</td><td>9</td><td>9</td></tr> <tr> <td>3</td><td>Simon</td><td>Happy</td><td>17</td><td>17</td></tr> <tr> <td>4</td><td>Foster</td><td>Roy</td><td>11</td><td>11</td></tr> <tr> <td>5</td><td>Billy</td><td>Thekid</td><td>8</td><td>8</td></tr> <tr> <td>6</td><td>Lionel</td><td>Richey</td><td>6</td><td>6</td></tr> <tr> <td>7</td><td>Rickey</td><td>Smalls</td><td>8</td><td>8</td></tr> <tr> <td>8</td><td>Phillip</td><td>Edwards</td><td>8</td><td>8</td></tr> <tr> <td>9</td><td>Smitty</td><td>Smith</td><td>12</td><td>12</td></tr> <tr> <td>10</td><td>Rich</td><td>Wells</td><td>9</td><td>9</td></tr> </tbody> </table>	EmpID	EmpFName	EmpLName	Manager	Technician	1	Greg	Fracker	12	12	2	Wesley	Heiney	9	9	3	Simon	Happy	17	17	4	Foster	Roy	11	11	5	Billy	Thekid	8	8	6	Lionel	Richey	6	6	7	Rickey	Smalls	8	8	8	Phillip	Edwards	8	8	9	Smitty	Smith	12	12	10	Rich	Wells	9	9																																																																																	
EmpID	EmpFName	EmpLName	Manager	Technician																																																																																																																																							
1	Greg	Fracker	12	12																																																																																																																																							
2	Wesley	Heiney	9	9																																																																																																																																							
3	Simon	Happy	17	17																																																																																																																																							
4	Foster	Roy	11	11																																																																																																																																							
5	Billy	Thekid	8	8																																																																																																																																							
6	Lionel	Richey	6	6																																																																																																																																							
7	Rickey	Smalls	8	8																																																																																																																																							
8	Phillip	Edwards	8	8																																																																																																																																							
9	Smitty	Smith	12	12																																																																																																																																							
10	Rich	Wells	9	9																																																																																																																																							

		on te.EmpID = tso.SOEmplDS group by empid, EmpFName, EmpLName																																																													
6	Number of times a tire protection has been purchased for a particular tire and number of times free service has been applied (free tire damage repair, free replacement).	select ProdID, PName, count(prodid) NoPurchases from TSalesOrderLine tsol join TProduct tp on tsol.SOLProductID = tp.ProdID where PProductType = 'Service' group by prodid, pname	<table border="1"> <thead> <tr> <th></th> <th>ProdID</th> <th>PName</th> <th>NoPurchases</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>Install Tire</td> <td>1</td> </tr> <tr> <td>2</td> <td>3</td> <td>Repair Tire</td> <td>1</td> </tr> <tr> <td>3</td> <td>4</td> <td>Rotate Tire</td> <td>1</td> </tr> </tbody> </table>		ProdID	PName	NoPurchases	1	2	Install Tire	1	2	3	Repair Tire	1	3	4	Rotate Tire	1																																												
	ProdID	PName	NoPurchases																																																												
1	2	Install Tire	1																																																												
2	3	Repair Tire	1																																																												
3	4	Rotate Tire	1																																																												
7	The following items for Purchase Orders: manufacturer name, number of POs, total cost.	select VendID, VendorName, Count(POID) Count, sum(POLQuantity * POLPurchaseCostfloa t) Total from TVendor tv join TPurchaseOrder tp on tv.VendID = tp.POVendorID join TPurchaseOrderLine tpo on tpo.POLPOID = tp.POVID join TTire tt on tpo.POLProductID = tt.TProductID group by vendid, VendorName	<table border="1"> <thead> <tr> <th></th> <th>VendID</th> <th>VendorName</th> <th>Count</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>13</td> <td>Michelin</td> <td>1</td> <td>6680.58</td> </tr> <tr> <td>2</td> <td>44</td> <td>Premier A/S</td> <td>7</td> <td>34435.54</td> </tr> <tr> <td>3</td> <td>45</td> <td>Defender</td> <td>4</td> <td>35123.09</td> </tr> <tr> <td>4</td> <td>49</td> <td>Advantage T/A</td> <td>14</td> <td>46632.43</td> </tr> <tr> <td>5</td> <td>50</td> <td>ProContact GX</td> <td>16</td> <td>53011.78</td> </tr> <tr> <td>6</td> <td>51</td> <td>ALL-Terrain</td> <td>12</td> <td>52345.21</td> </tr> <tr> <td>7</td> <td>52</td> <td>Pilot Sport 4</td> <td>14</td> <td>44861.28</td> </tr> <tr> <td>8</td> <td>54</td> <td>ContiTrac TR</td> <td>10</td> <td>50821.72</td> </tr> <tr> <td>9</td> <td>55</td> <td>Ecopia EP20</td> <td>3</td> <td>3537.51</td> </tr> <tr> <td>10</td> <td>60</td> <td>Dueler H/L Alenza</td> <td>7</td> <td>43039.01</td> </tr> <tr> <td>11</td> <td>72</td> <td>G-ForceRival</td> <td>12</td> <td>57792.13</td> </tr> </tbody> </table>		VendID	VendorName	Count	Total	1	13	Michelin	1	6680.58	2	44	Premier A/S	7	34435.54	3	45	Defender	4	35123.09	4	49	Advantage T/A	14	46632.43	5	50	ProContact GX	16	53011.78	6	51	ALL-Terrain	12	52345.21	7	52	Pilot Sport 4	14	44861.28	8	54	ContiTrac TR	10	50821.72	9	55	Ecopia EP20	3	3537.51	10	60	Dueler H/L Alenza	7	43039.01	11	72	G-ForceRival	12	57792.13
	VendID	VendorName	Count	Total																																																											
1	13	Michelin	1	6680.58																																																											
2	44	Premier A/S	7	34435.54																																																											
3	45	Defender	4	35123.09																																																											
4	49	Advantage T/A	14	46632.43																																																											
5	50	ProContact GX	16	53011.78																																																											
6	51	ALL-Terrain	12	52345.21																																																											
7	52	Pilot Sport 4	14	44861.28																																																											
8	54	ContiTrac TR	10	50821.72																																																											
9	55	Ecopia EP20	3	3537.51																																																											
10	60	Dueler H/L Alenza	7	43039.01																																																											
11	72	G-ForceRival	12	57792.13																																																											

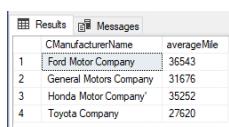
8	Number of orders and total sales per customer in the past 2 years. This report is particularly important as it shows the number of returning customers.	<pre>select CustID, CustFName, CustLName, Count(SOID) as Count from TCustomer tcr join TCar tc on tcr.CustID = tc.CarCustID join TSalesOrder tso on tso.SOCarID = tc.CarID where year(SODate) &gt;= 2 group by custid, custfname, custlname</pre>	<table border="1"> <thead> <tr> <th></th> <th>CustID</th> <th>CustFName</th> <th>CustLName</th> <th>Count</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>Frank</td><td>Ocean</td><td>3</td></tr> <tr><td>2</td><td>3</td><td>Miley</td><td>Cyrus</td><td>1</td></tr> <tr><td>3</td><td>4</td><td>Wes</td><td>Black</td><td>1</td></tr> <tr><td>4</td><td>7</td><td>Colette</td><td>Marquez</td><td>1</td></tr> <tr><td>5</td><td>16</td><td>Myles</td><td>Hall</td><td>1</td></tr> <tr><td>6</td><td>21</td><td>Sylvester</td><td>Curtis</td><td>2</td></tr> <tr><td>7</td><td>23</td><td>Karly</td><td>Joseph</td><td>1</td></tr> <tr><td>8</td><td>24</td><td>Oscar</td><td>Raymond</td><td>1</td></tr> <tr><td>9</td><td>25</td><td>Cameron</td><td>Haney</td><td>1</td></tr> <tr><td>10</td><td>26</td><td>Charde</td><td>Carey</td><td>1</td></tr> <tr><td>11</td><td>28</td><td>Uriel</td><td>Church</td><td>2</td></tr> <tr><td>12</td><td>29</td><td>Denise</td><td>Evans</td><td>1</td></tr> <tr><td>13</td><td>30</td><td>Cullen</td><td>Bentley</td><td>4</td></tr> <tr><td>14</td><td>31</td><td>Gareth</td><td>Trevino</td><td>2</td></tr> <tr><td>15</td><td>36</td><td>Hop</td><td>Norman</td><td>1</td></tr> <tr><td>16</td><td>37</td><td>Abigail</td><td>Vance</td><td>1</td></tr> </tbody> </table>		CustID	CustFName	CustLName	Count	1	1	Frank	Ocean	3	2	3	Miley	Cyrus	1	3	4	Wes	Black	1	4	7	Colette	Marquez	1	5	16	Myles	Hall	1	6	21	Sylvester	Curtis	2	7	23	Karly	Joseph	1	8	24	Oscar	Raymond	1	9	25	Cameron	Haney	1	10	26	Charde	Carey	1	11	28	Uriel	Church	2	12	29	Denise	Evans	1	13	30	Cullen	Bentley	4	14	31	Gareth	Trevino	2	15	36	Hop	Norman	1	16	37	Abigail	Vance	1																																																			
	CustID	CustFName	CustLName	Count																																																																																																																																							
1	1	Frank	Ocean	3																																																																																																																																							
2	3	Miley	Cyrus	1																																																																																																																																							
3	4	Wes	Black	1																																																																																																																																							
4	7	Colette	Marquez	1																																																																																																																																							
5	16	Myles	Hall	1																																																																																																																																							
6	21	Sylvester	Curtis	2																																																																																																																																							
7	23	Karly	Joseph	1																																																																																																																																							
8	24	Oscar	Raymond	1																																																																																																																																							
9	25	Cameron	Haney	1																																																																																																																																							
10	26	Charde	Carey	1																																																																																																																																							
11	28	Uriel	Church	2																																																																																																																																							
12	29	Denise	Evans	1																																																																																																																																							
13	30	Cullen	Bentley	4																																																																																																																																							
14	31	Gareth	Trevino	2																																																																																																																																							
15	36	Hop	Norman	1																																																																																																																																							
16	37	Abigail	Vance	1																																																																																																																																							
9	List of tires that have not been purchased within the last 6 months (in order to better manage inventory).	<pre>SELECT Prodid, PName, SODate FROM TSalesOrder tso join TSalesOrderLine tsol on tso.SOID = tsol.SOLSOID left join TProduct tp on tp.Prodid= tsol.SOLProductID where SODate not between '2020-11-02' and '2021-05-02' and PProductType = 'Tire'</pre>	<table border="1"> <thead> <tr> <th></th> <th>Prodid</th> <th>PName</th> <th>SODate</th> </tr> </thead> <tbody> <tr><td>1</td><td>43</td><td>205/65R16</td><td>2020-01-10</td></tr> <tr><td>2</td><td>7</td><td>235/50r19</td><td>2020-03-22</td></tr> <tr><td>3</td><td>69</td><td>235/40R19</td><td>2020-02-23</td></tr> </tbody> </table>		Prodid	PName	SODate	1	43	205/65R16	2020-01-10	2	7	235/50r19	2020-03-22	3	69	235/40R19	2020-02-23																																																																																																																								
	Prodid	PName	SODate																																																																																																																																								
1	43	205/65R16	2020-01-10																																																																																																																																								
2	7	235/50r19	2020-03-22																																																																																																																																								
3	69	235/40R19	2020-02-23																																																																																																																																								
10	Names of customers who took advantage of the financing option, date purchased, total amount purchased, credit limit, number of payments made, the total amount paid, outstanding amount, is time to pay-off less than 6	<pre>select CustFName, CustLName, Credit_limit, SOIsFinance,PayDate , PayAmount, PMTDescription from TCustomer tc join TSalesOrder tso on tc.CustID = tso.SOCustID join TPaymentIn tpi on tso.SOID = tpi.PayInID</pre>	<table border="1"> <thead> <tr> <th></th> <th>CustFName</th> <th>CustLName</th> <th>Credit_limit</th> <th>SOIsFinance</th> <th>PayDate</th> <th>PayAmount</th> <th>PMTDescription</th> </tr> </thead> <tbody> <tr><td>1</td><td>Idola</td><td>Riggs</td><td>60</td><td>No</td><td>2021-04-24</td><td>67</td><td>Debit</td></tr> <tr><td>2</td><td>Damon</td><td>Mcclain</td><td>10</td><td>Yes</td><td>2021-04-02</td><td>24</td><td>credit</td></tr> <tr><td>3</td><td>Steven</td><td>Santos</td><td>50</td><td>Yes</td><td>2021-04-13</td><td>34</td><td>credit</td></tr> <tr><td>4</td><td>Karly</td><td>Joseph</td><td>10</td><td>No</td><td>2021-04-02</td><td>12</td><td>credit</td></tr> <tr><td>5</td><td>Abigail</td><td>Vance</td><td>50</td><td>Yes</td><td>2021-04-07</td><td>26</td><td>moneyorder</td></tr> <tr><td>6</td><td>Katell</td><td>Waren</td><td>10</td><td>Yes</td><td>2021-04-15</td><td>89</td><td>moneyorder</td></tr> <tr><td>7</td><td>Lillian</td><td>Mendoza</td><td>200</td><td>No</td><td>2021-04-16</td><td>76</td><td>cash</td></tr> <tr><td>8</td><td>Hyatt</td><td>Allen</td><td>50</td><td>No</td><td>2021-04-06</td><td>89</td><td>cash</td></tr> <tr><td>9</td><td>Addison</td><td>Mays</td><td>60</td><td>Yes</td><td>2021-04-10</td><td>59</td><td>credit</td></tr> <tr><td>10</td><td>Roary</td><td>Valenzuela</td><td>30</td><td>Yes</td><td>2021-04-07</td><td>30</td><td>moneyorder</td></tr> <tr><td>11</td><td>Gl</td><td>Simpson</td><td>50</td><td>No</td><td>2021-04-24</td><td>96</td><td>moneyorder</td></tr> <tr><td>12</td><td>Nora</td><td>Wells</td><td>10</td><td>No</td><td>2021-04-17</td><td>16</td><td>Debit</td></tr> <tr><td>13</td><td>Uriel</td><td>Church</td><td>30</td><td>Yes</td><td>2021-04-17</td><td>51</td><td>Debit</td></tr> <tr><td>14</td><td>Roary</td><td>Valenzuela</td><td>30</td><td>No</td><td>2021-04-16</td><td>9</td><td>Debit</td></tr> <tr><td>15</td><td>Karly</td><td>Joseph</td><td>10</td><td>Yes</td><td>2021-04-16</td><td>52</td><td>Debit</td></tr> <tr><td>16</td><td>Ita</td><td>Duke</td><td>70</td><td>No</td><td>2021-04-07</td><td>93</td><td>moneyorder</td></tr> </tbody> </table>		CustFName	CustLName	Credit_limit	SOIsFinance	PayDate	PayAmount	PMTDescription	1	Idola	Riggs	60	No	2021-04-24	67	Debit	2	Damon	Mcclain	10	Yes	2021-04-02	24	credit	3	Steven	Santos	50	Yes	2021-04-13	34	credit	4	Karly	Joseph	10	No	2021-04-02	12	credit	5	Abigail	Vance	50	Yes	2021-04-07	26	moneyorder	6	Katell	Waren	10	Yes	2021-04-15	89	moneyorder	7	Lillian	Mendoza	200	No	2021-04-16	76	cash	8	Hyatt	Allen	50	No	2021-04-06	89	cash	9	Addison	Mays	60	Yes	2021-04-10	59	credit	10	Roary	Valenzuela	30	Yes	2021-04-07	30	moneyorder	11	Gl	Simpson	50	No	2021-04-24	96	moneyorder	12	Nora	Wells	10	No	2021-04-17	16	Debit	13	Uriel	Church	30	Yes	2021-04-17	51	Debit	14	Roary	Valenzuela	30	No	2021-04-16	9	Debit	15	Karly	Joseph	10	Yes	2021-04-16	52	Debit	16	Ita	Duke	70	No	2021-04-07	93	moneyorder
	CustFName	CustLName	Credit_limit	SOIsFinance	PayDate	PayAmount	PMTDescription																																																																																																																																				
1	Idola	Riggs	60	No	2021-04-24	67	Debit																																																																																																																																				
2	Damon	Mcclain	10	Yes	2021-04-02	24	credit																																																																																																																																				
3	Steven	Santos	50	Yes	2021-04-13	34	credit																																																																																																																																				
4	Karly	Joseph	10	No	2021-04-02	12	credit																																																																																																																																				
5	Abigail	Vance	50	Yes	2021-04-07	26	moneyorder																																																																																																																																				
6	Katell	Waren	10	Yes	2021-04-15	89	moneyorder																																																																																																																																				
7	Lillian	Mendoza	200	No	2021-04-16	76	cash																																																																																																																																				
8	Hyatt	Allen	50	No	2021-04-06	89	cash																																																																																																																																				
9	Addison	Mays	60	Yes	2021-04-10	59	credit																																																																																																																																				
10	Roary	Valenzuela	30	Yes	2021-04-07	30	moneyorder																																																																																																																																				
11	Gl	Simpson	50	No	2021-04-24	96	moneyorder																																																																																																																																				
12	Nora	Wells	10	No	2021-04-17	16	Debit																																																																																																																																				
13	Uriel	Church	30	Yes	2021-04-17	51	Debit																																																																																																																																				
14	Roary	Valenzuela	30	No	2021-04-16	9	Debit																																																																																																																																				
15	Karly	Joseph	10	Yes	2021-04-16	52	Debit																																																																																																																																				
16	Ita	Duke	70	No	2021-04-07	93	moneyorder																																																																																																																																				

	months, all displayed from the latest date and then the largest amount owed.	join TPaymentType tpt on tpi.PayInPMTID = tpt.PMTID																																																					
11	Total profit per tire type and manufacturer type in the past 6 months	select TireName, TVendID, sum(SOTotal) as TotalSales, SODate from tsalesorder tso join TSalesOrderLine tsol on tso.SOID = tsol.SOLSOID join TProduct tp on tp.ProdID = tsol.SOLProductID join TTire ttr on ttr.TProductID = tp.ProdID where SODate between '2020-10-25' and '2021-04-25' group by TireName, TVendID, SODate	 <table border="1"> <thead> <tr> <th>TireName</th> <th>TVendID</th> <th>TotalSales</th> <th>SODate</th> </tr> </thead> <tbody> <tr><td>195/65R17</td><td>10</td><td>135</td><td>2021-01-31</td></tr> <tr><td>195/65R17</td><td>75</td><td>845</td><td>2021-04-07</td></tr> <tr><td>205/55R16</td><td>45</td><td>246</td><td>2021-02-12</td></tr> <tr><td>205/55R16</td><td>45</td><td>852</td><td>2021-03-05</td></tr> <tr><td>205/55R16</td><td>75</td><td>156</td><td>2021-02-19</td></tr> <tr><td>205/55R16</td><td>76</td><td>180</td><td>2021-03-25</td></tr> <tr><td>205/65R16</td><td>10</td><td>180</td><td>2021-03-25</td></tr> <tr><td>205/65R16</td><td>10</td><td>390</td><td>2021-04-10</td></tr> <tr><td>205/65R16</td><td>10</td><td>290</td><td>2021-04-16</td></tr> <tr><td>205/65R16</td><td>75</td><td>325</td><td>2021-04-22</td></tr> <tr><td>205/65R26</td><td>10</td><td>240</td><td>2021-03-01</td></tr> <tr><td>205/65R26</td><td>10</td><td>180</td><td>2021-04-03</td></tr> </tbody> </table>	TireName	TVendID	TotalSales	SODate	195/65R17	10	135	2021-01-31	195/65R17	75	845	2021-04-07	205/55R16	45	246	2021-02-12	205/55R16	45	852	2021-03-05	205/55R16	75	156	2021-02-19	205/55R16	76	180	2021-03-25	205/65R16	10	180	2021-03-25	205/65R16	10	390	2021-04-10	205/65R16	10	290	2021-04-16	205/65R16	75	325	2021-04-22	205/65R26	10	240	2021-03-01	205/65R26	10	180	2021-04-03
TireName	TVendID	TotalSales	SODate																																																				
195/65R17	10	135	2021-01-31																																																				
195/65R17	75	845	2021-04-07																																																				
205/55R16	45	246	2021-02-12																																																				
205/55R16	45	852	2021-03-05																																																				
205/55R16	75	156	2021-02-19																																																				
205/55R16	76	180	2021-03-25																																																				
205/65R16	10	180	2021-03-25																																																				
205/65R16	10	390	2021-04-10																																																				
205/65R16	10	290	2021-04-16																																																				
205/65R16	75	325	2021-04-22																																																				
205/65R26	10	240	2021-03-01																																																				
205/65R26	10	180	2021-04-03																																																				
12	List of all customers that have not made a purchase within the last 12 months from the current date.	select Custid, CustFName, CustLName from TCustomer left join TSalesOrder on TCustomer.CustID = TSalesOrder.SOCustID where SOID is null	 <table border="1"> <thead> <tr> <th>CustID</th> <th>CustFName</th> <th>CustLname</th> </tr> </thead> <tbody> <tr><td>1</td><td>Frank</td><td>Ocean</td></tr> <tr><td>2</td><td>Eric</td><td>Andre</td></tr> <tr><td>3</td><td>Miley</td><td>Cyrus</td></tr> <tr><td>4</td><td>Collette</td><td>Marquez</td></tr> <tr><td>5</td><td>Kathleen</td><td>Wilkins</td></tr> <tr><td>6</td><td>Shaleigh</td><td>Kline</td></tr> <tr><td>7</td><td>Bernard</td><td>Marks</td></tr> <tr><td>8</td><td>Iola</td><td>Marks</td></tr> <tr><td>9</td><td>Rebecca</td><td>Levine</td></tr> <tr><td>10</td><td>Wesley</td><td>Day</td></tr> <tr><td>11</td><td>Christian</td><td>Owen</td></tr> <tr><td>12</td><td>Oscar</td><td>Raymond</td></tr> <tr><td>13</td><td>Cameron</td><td>Haney</td></tr> <tr><td>14</td><td>Brittany</td><td>Pitts</td></tr> <tr><td>15</td><td>Cullen</td><td>Bentley</td></tr> <tr><td>16</td><td>Gareth</td><td>Trevino</td></tr> </tbody> </table>	CustID	CustFName	CustLname	1	Frank	Ocean	2	Eric	Andre	3	Miley	Cyrus	4	Collette	Marquez	5	Kathleen	Wilkins	6	Shaleigh	Kline	7	Bernard	Marks	8	Iola	Marks	9	Rebecca	Levine	10	Wesley	Day	11	Christian	Owen	12	Oscar	Raymond	13	Cameron	Haney	14	Brittany	Pitts	15	Cullen	Bentley	16	Gareth	Trevino	
CustID	CustFName	CustLname																																																					
1	Frank	Ocean																																																					
2	Eric	Andre																																																					
3	Miley	Cyrus																																																					
4	Collette	Marquez																																																					
5	Kathleen	Wilkins																																																					
6	Shaleigh	Kline																																																					
7	Bernard	Marks																																																					
8	Iola	Marks																																																					
9	Rebecca	Levine																																																					
10	Wesley	Day																																																					
11	Christian	Owen																																																					
12	Oscar	Raymond																																																					
13	Cameron	Haney																																																					
14	Brittany	Pitts																																																					
15	Cullen	Bentley																																																					
16	Gareth	Trevino																																																					

13	List of customers whose average sales is less than the average of all sales. This will help us to find customers whom we should target to get a higher volume of sales.	<pre> select custfname,        custlname,        avg(SOLQuantity*Psal             espriceint) AvgPurchase from TCustomer tc join TSalesOrder tso on tc.CustID = tso.SOCustID join TSalesOrderLine tsol on tsol.SOLSOID = tso.SOIID join TProduct tp on tp.ProdID =tsol.SOLProductID group by CustFName, CustLName having avg(SOLQuantity*Psal             espriceint) &lt; (select avg(SOLQuantity * PSalesPriceint) from TSalesOrderLine tsol join TProduct tp on tp.ProdID =tsol.SOLProductID) </pre>	<table border="1"> <thead> <tr> <th></th><th>custfname</th><th>custlname</th><th>AvgPurchase</th></tr> </thead> <tbody> <tr><td>1</td><td>Wes</td><td>Black</td><td>1456</td></tr> <tr><td>2</td><td>Kamal</td><td>Bowers</td><td>7233</td></tr> <tr><td>3</td><td>Tamekah</td><td>Combs</td><td>8150</td></tr> <tr><td>4</td><td>Ila</td><td>Duke</td><td>4150</td></tr> <tr><td>5</td><td>Garrison</td><td>Farrell</td><td>6072</td></tr> <tr><td>6</td><td>Lillian</td><td>Farrell</td><td>1499</td></tr> <tr><td>7</td><td>Kane</td><td>Freeman</td><td>5829</td></tr> <tr><td>8</td><td>Nasim</td><td>Gallagher</td><td>1287</td></tr> <tr><td>9</td><td>Yardley</td><td>Glenn</td><td>6032</td></tr> <tr><td>10</td><td>Ciara</td><td>Gutierrez</td><td>5538</td></tr> <tr><td>11</td><td>Lisandra</td><td>Hoover</td><td>8318</td></tr> <tr><td>12</td><td>Kiayada</td><td>Hull</td><td>4326</td></tr> <tr><td>13</td><td>Phelan</td><td>James</td><td>6120</td></tr> <tr><td>14</td><td>Eugenia</td><td>Johnston</td><td>7488</td></tr> <tr><td>15</td><td>Karly</td><td>Joseph</td><td>8523</td></tr> <tr><td>16</td><td>Laura</td><td>Kirby</td><td>7645</td></tr> </tbody> </table>		custfname	custlname	AvgPurchase	1	Wes	Black	1456	2	Kamal	Bowers	7233	3	Tamekah	Combs	8150	4	Ila	Duke	4150	5	Garrison	Farrell	6072	6	Lillian	Farrell	1499	7	Kane	Freeman	5829	8	Nasim	Gallagher	1287	9	Yardley	Glenn	6032	10	Ciara	Gutierrez	5538	11	Lisandra	Hoover	8318	12	Kiayada	Hull	4326	13	Phelan	James	6120	14	Eugenia	Johnston	7488	15	Karly	Joseph	8523	16	Laura	Kirby	7645
	custfname	custlname	AvgPurchase																																																																				
1	Wes	Black	1456																																																																				
2	Kamal	Bowers	7233																																																																				
3	Tamekah	Combs	8150																																																																				
4	Ila	Duke	4150																																																																				
5	Garrison	Farrell	6072																																																																				
6	Lillian	Farrell	1499																																																																				
7	Kane	Freeman	5829																																																																				
8	Nasim	Gallagher	1287																																																																				
9	Yardley	Glenn	6032																																																																				
10	Ciara	Gutierrez	5538																																																																				
11	Lisandra	Hoover	8318																																																																				
12	Kiayada	Hull	4326																																																																				
13	Phelan	James	6120																																																																				
14	Eugenia	Johnston	7488																																																																				
15	Karly	Joseph	8523																																																																				
16	Laura	Kirby	7645																																																																				

## Three Additional Queries

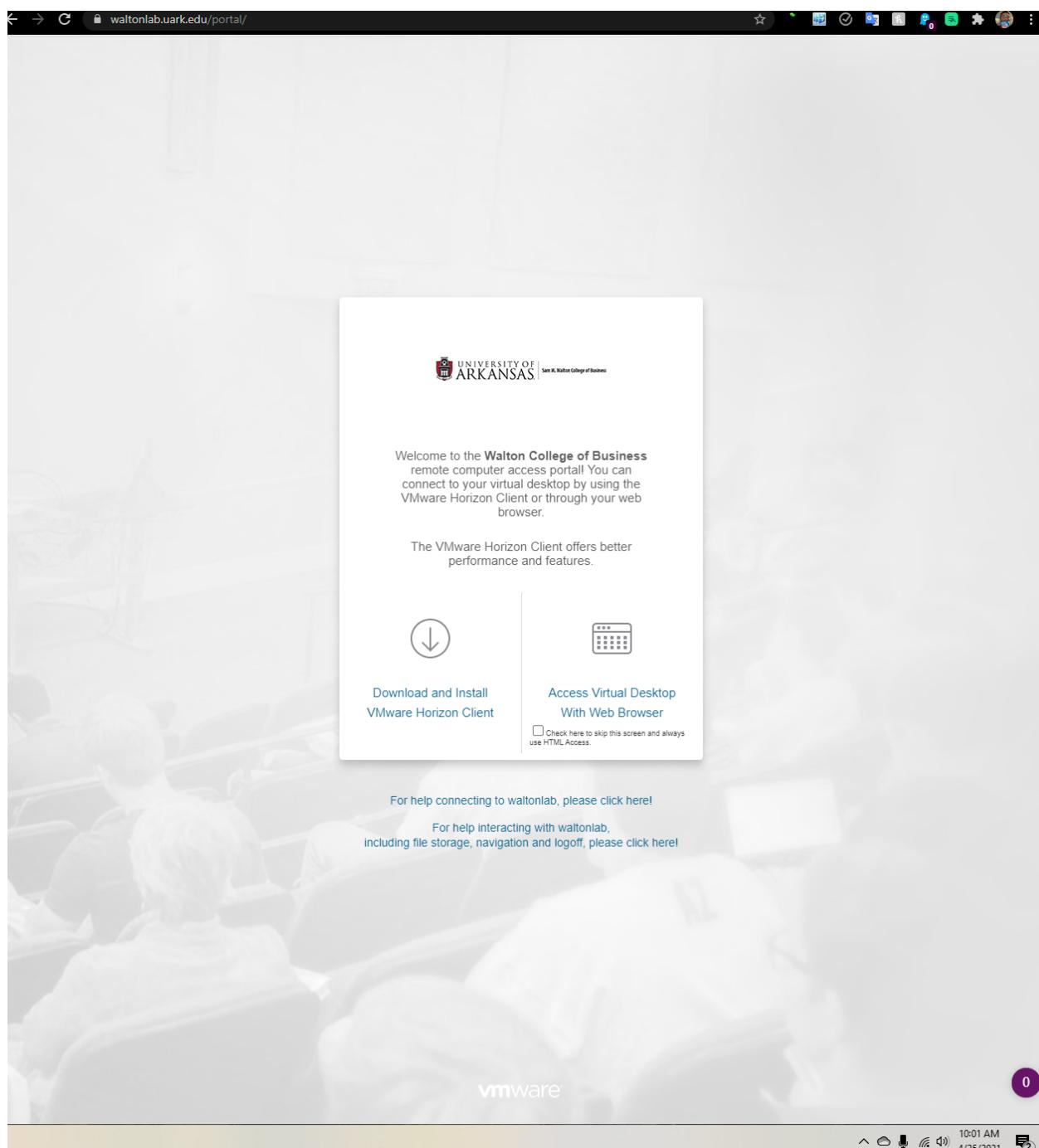
Query #	Question	Why is this important	SQL	Partial Output	Recap of Findings									
1	What percentage of customers are male and female	We can efficiently target marketing	<pre> select CustGender, Count(soid) from TCustomer join TSalesOrder </pre>	<table border="1"> <thead> <tr> <th></th> <th>CustGender</th> <th>(No column name)</th> </tr> </thead> <tbody> <tr><td>1</td><td>Femal</td><td>36</td></tr> <tr><td>2</td><td>Male</td><td>64</td></tr> </tbody> </table>		CustGender	(No column name)	1	Femal	36	2	Male	64	There is more male customer than female
	CustGender	(No column name)												
1	Femal	36												
2	Male	64												

			on TCustomer.Cu stID = TSalesOrder.S OCustID group by CustGender		
2	Which customer had what amount of discount	We can check whether we should reduce discount or increase discount	select CustID, CustLName, CustFName, DisAmount from TCustomer join TSalesOrder on TCustomer.Cu stID = TSalesOrder.S OCustID join TDiscount on TDiscount.Dis countID = TSalesOrder.S ODiscountID		
3	What is average car mileage by car manufacturers	If certain car manufacturer have higher mileage there is more chance to sell more of tire that fits in specific car manufacturer	select CManufacturer Name, avg(SOCarMile age) as averageMile from TCar join TSalesOrder tso on TCar.CarID = tso.SOCarID join TCarManufact urer tcm on tcm.CMID = TCar.CarCMID		Ford has most mileage ford drivers drive a lot and have higher chance to change their tires

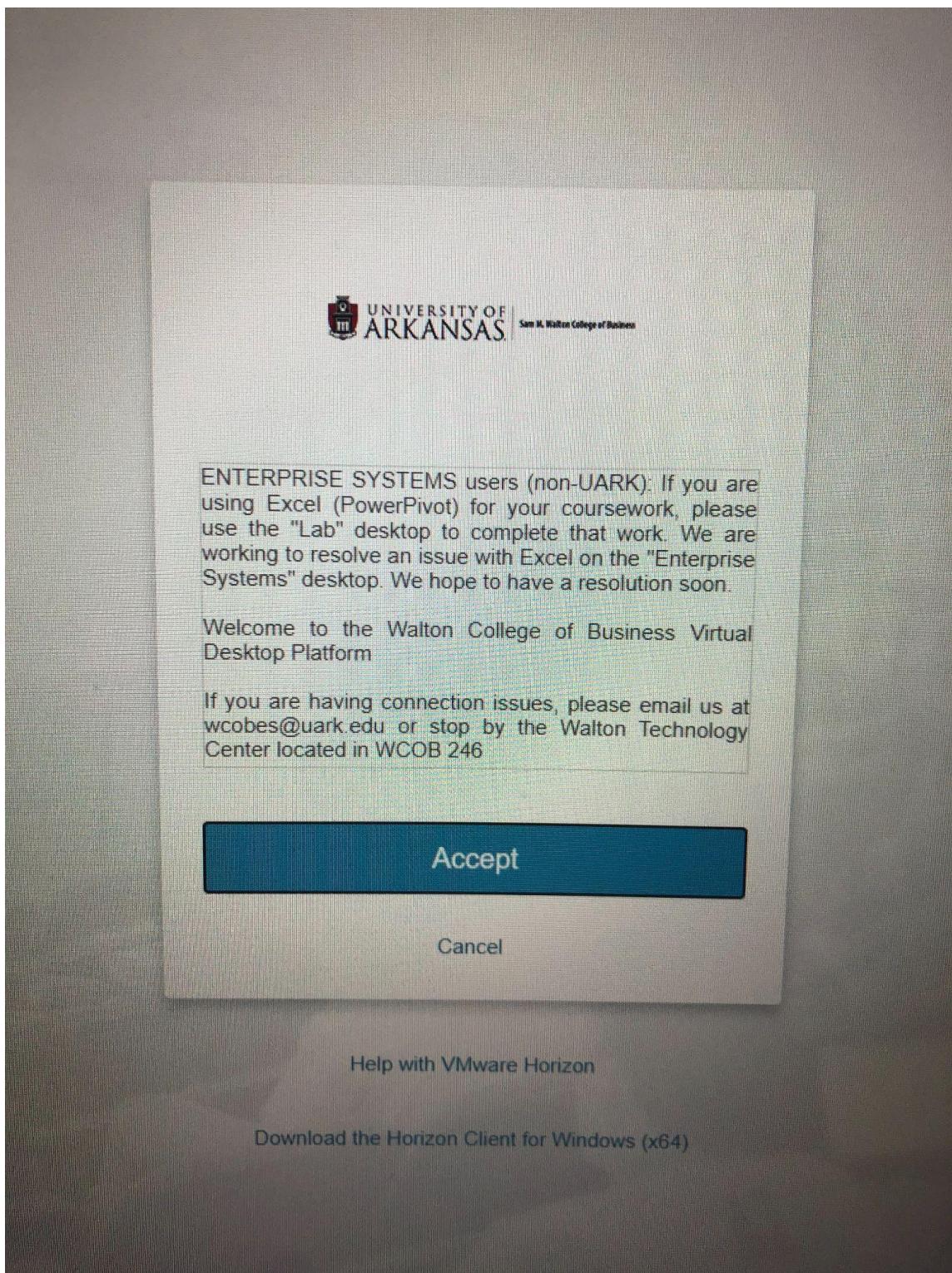
			group by CManufacturer Name		
--	--	--	-----------------------------------	--	--

## User Documentation

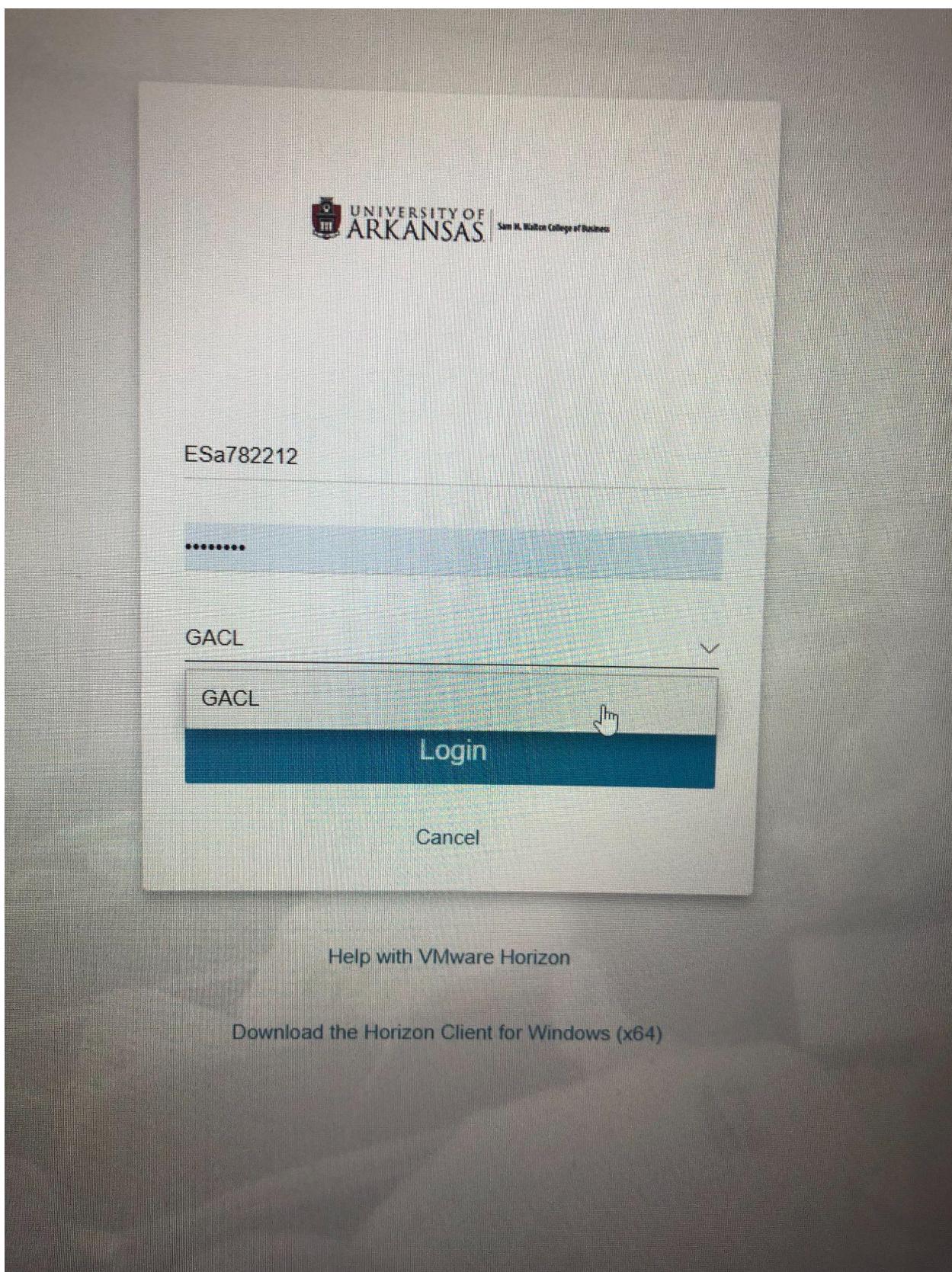
User Documentation is essential for anyone trying to login and use Walton Labs. Below, you will be able to follow the steps needed to access and use your database that you have been granted access too. There will be one database for you as an individual and another for your group. Follow the steps below if you need any help in the process.



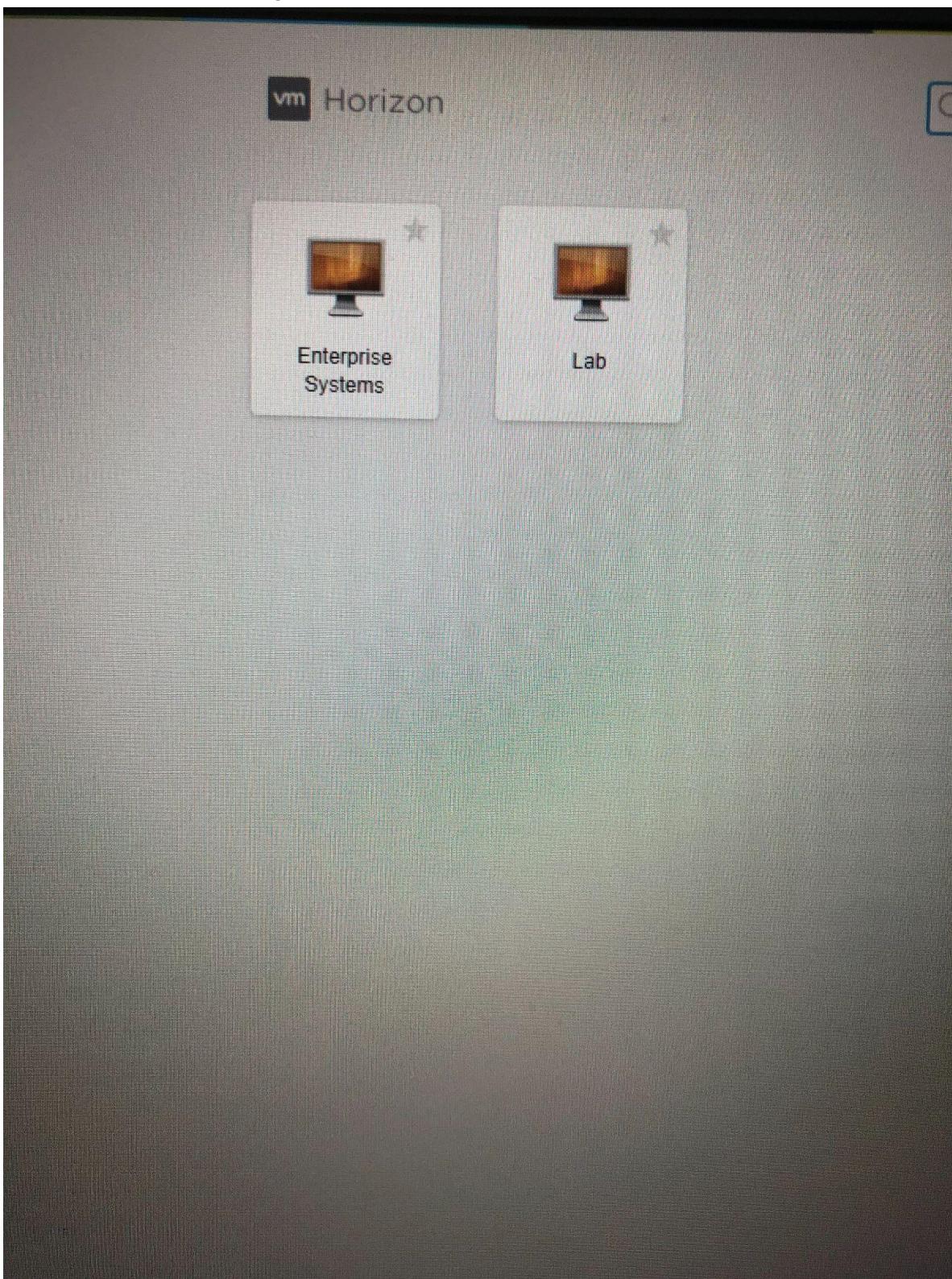
Type in your browser of choice [waltonlab.uark.edu](http://waltonlab.uark.edu) and this screen will display. When this screen displays click on the right option, Access virtual desktop with web browser



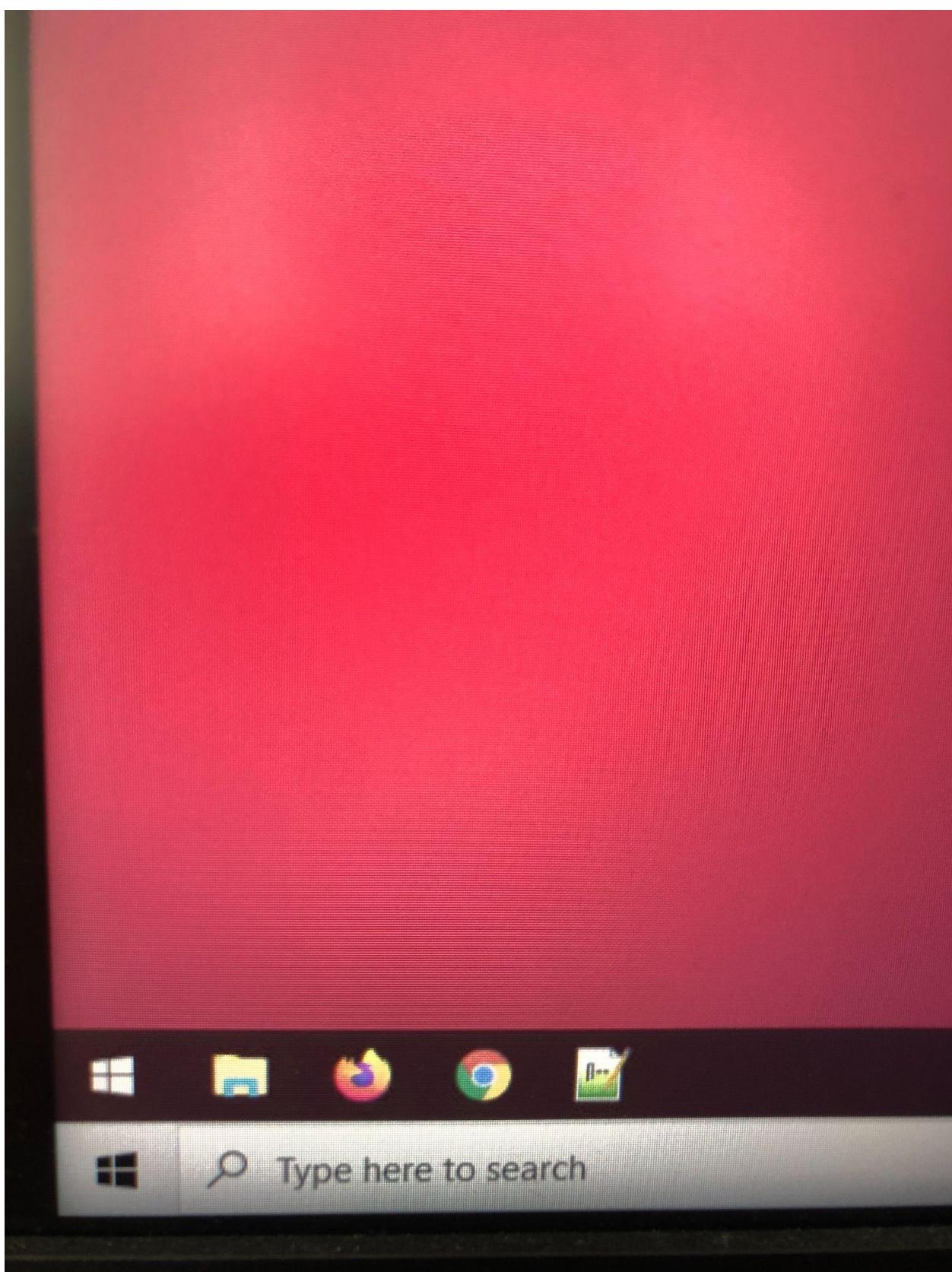
You will then want to accept this user agreement.



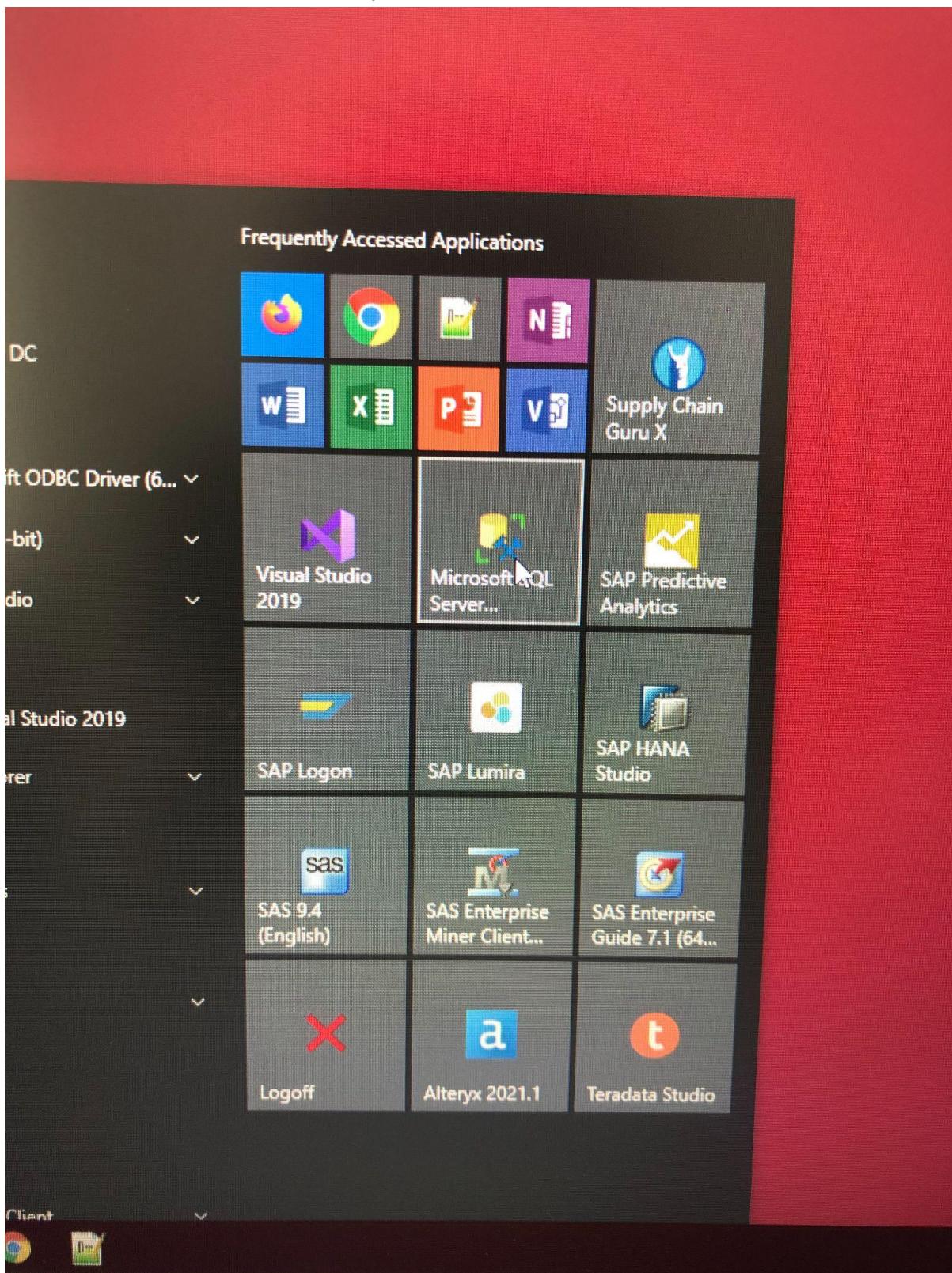
Once you accept the user agreement, type in your username and password and make sure it is on GACL and then click login.



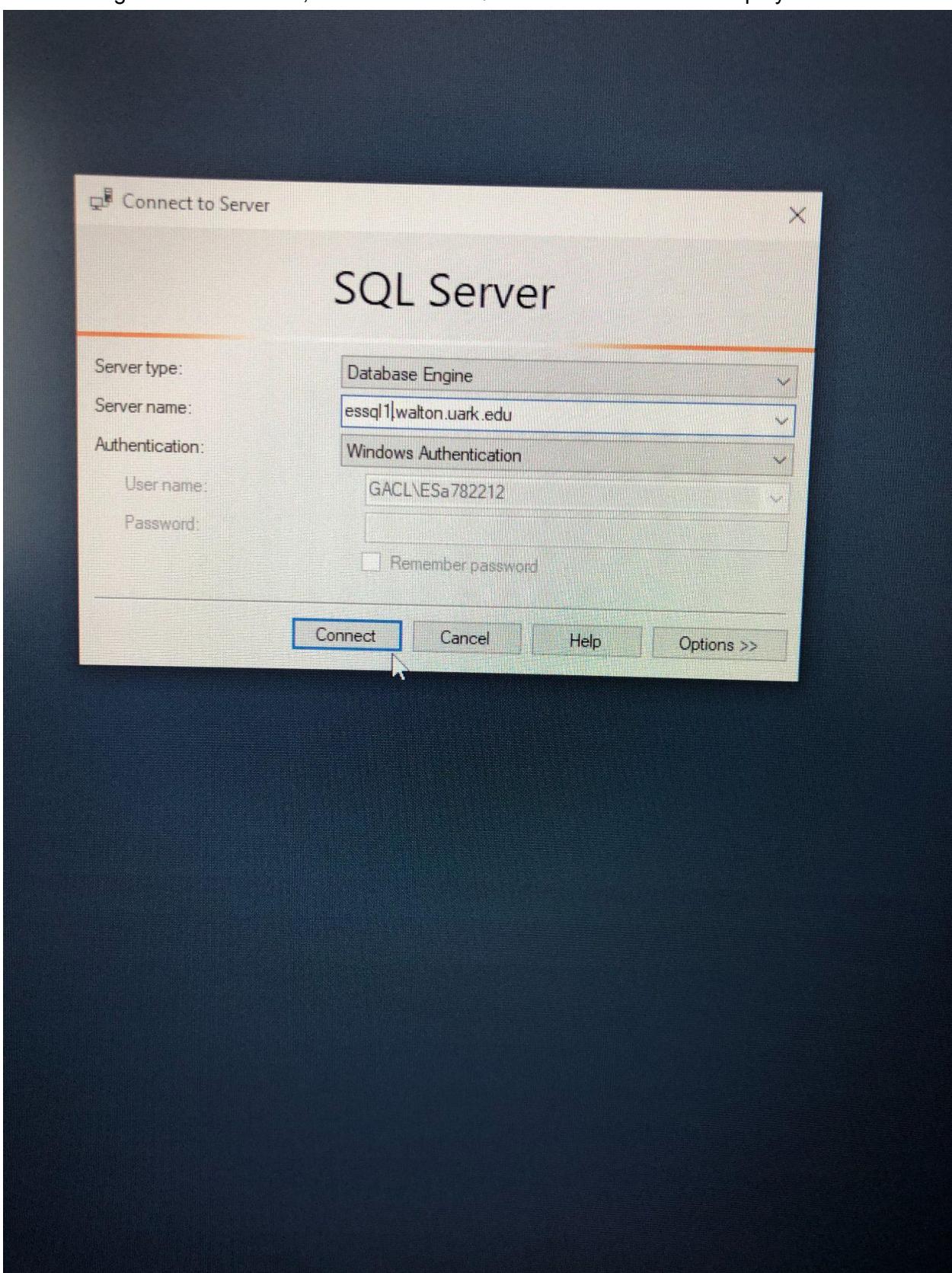
Next, you will want to click on enterprise systems



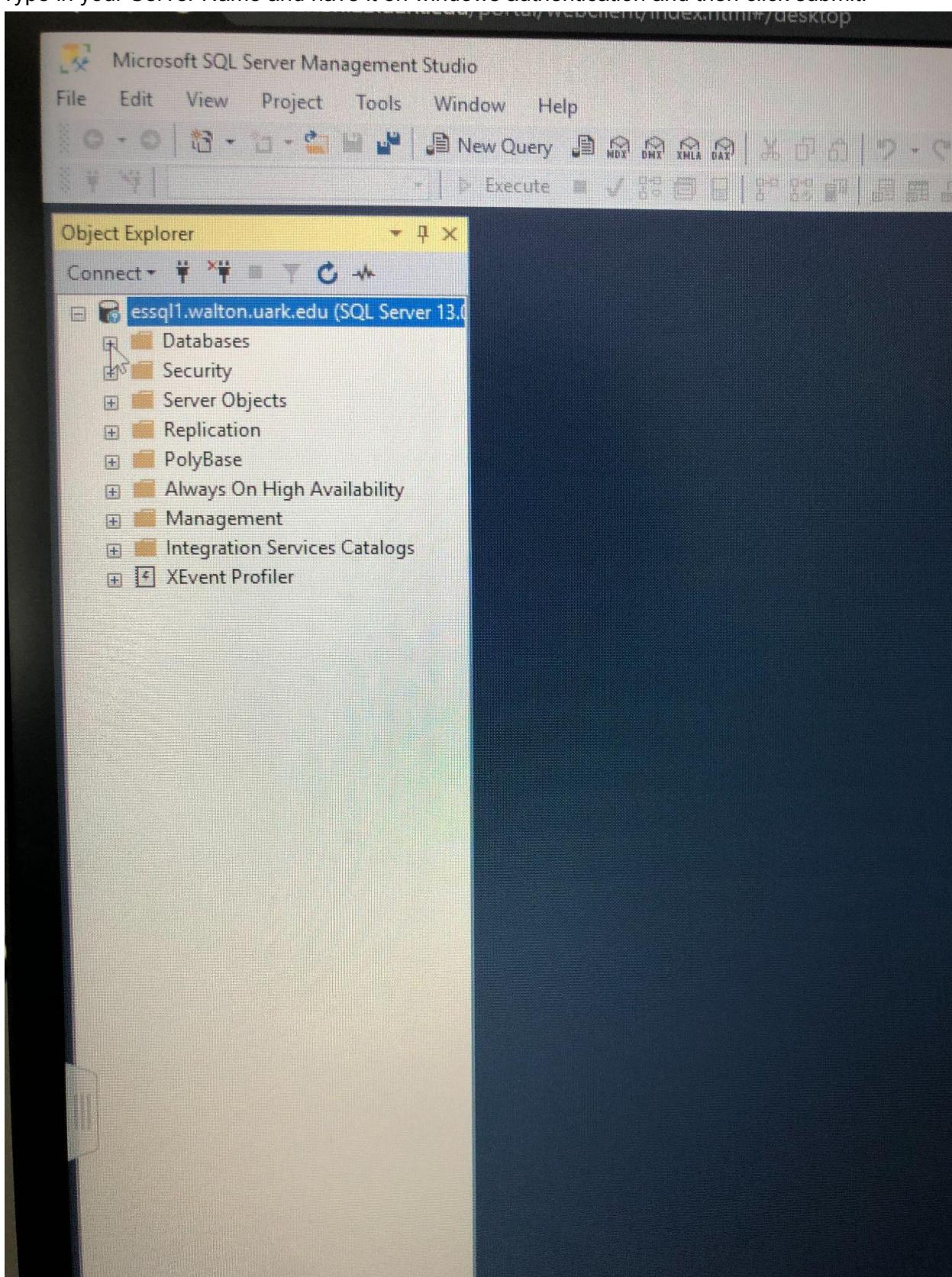
Now you can see that you will have a windows button for your own desktop and then a windows button for Walton Labs. \*Make sure you click the windows in Walton Labs which is the top one\*



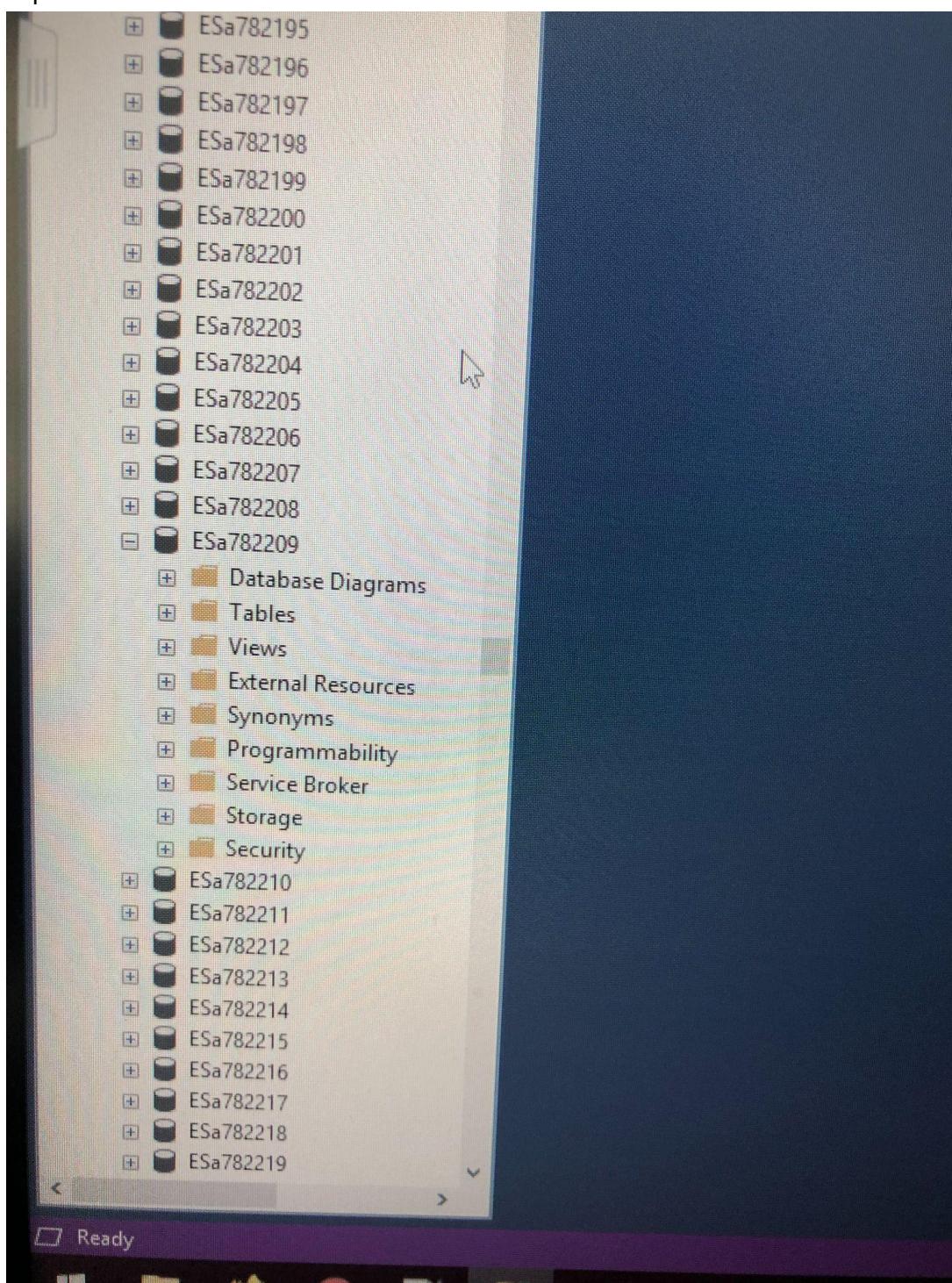
After clicking the windows icon, find Microsoft SQL Server and click it as displayed above.



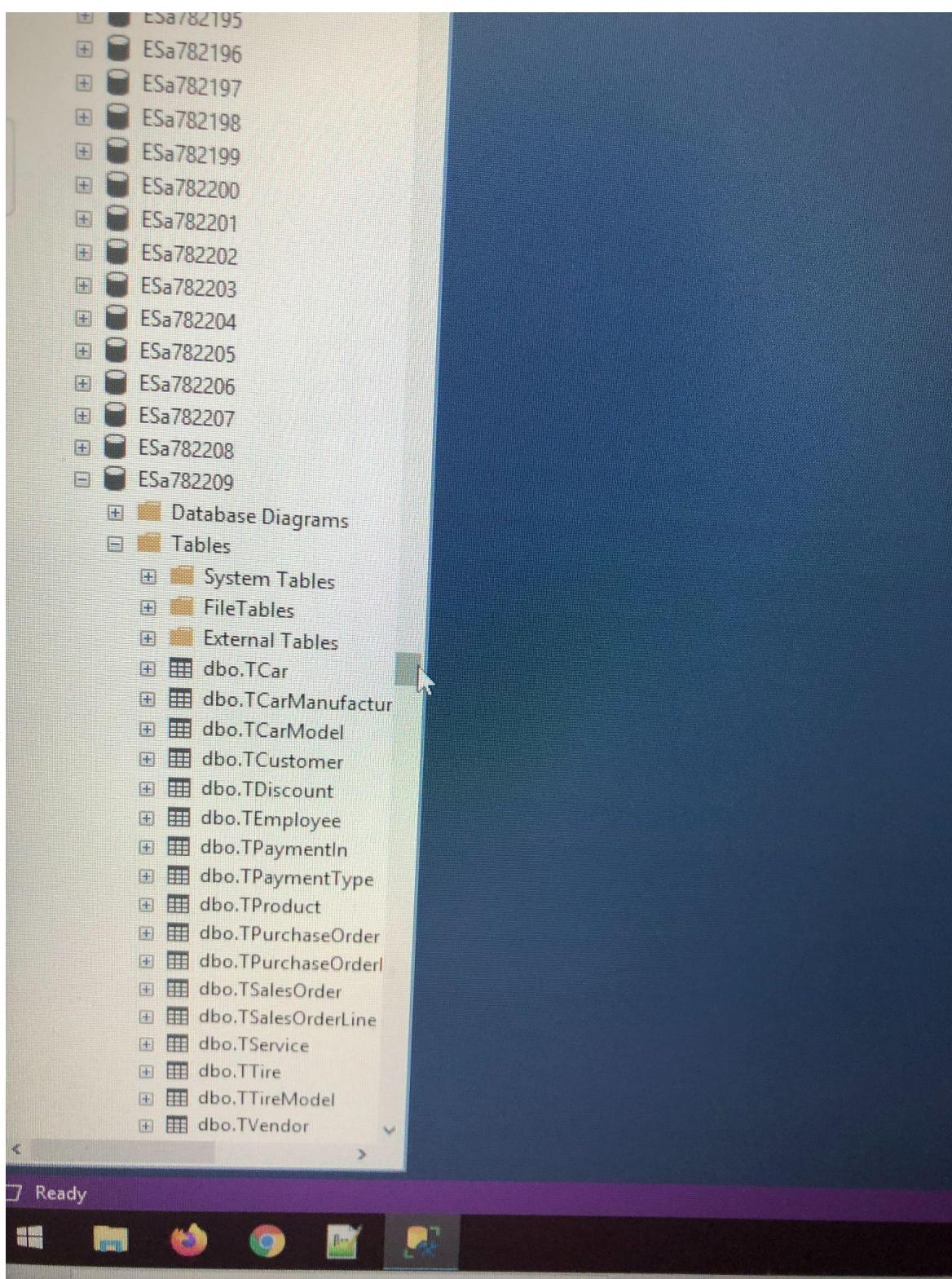
Type in your Server Name and have it on windows authentication and then click submit.



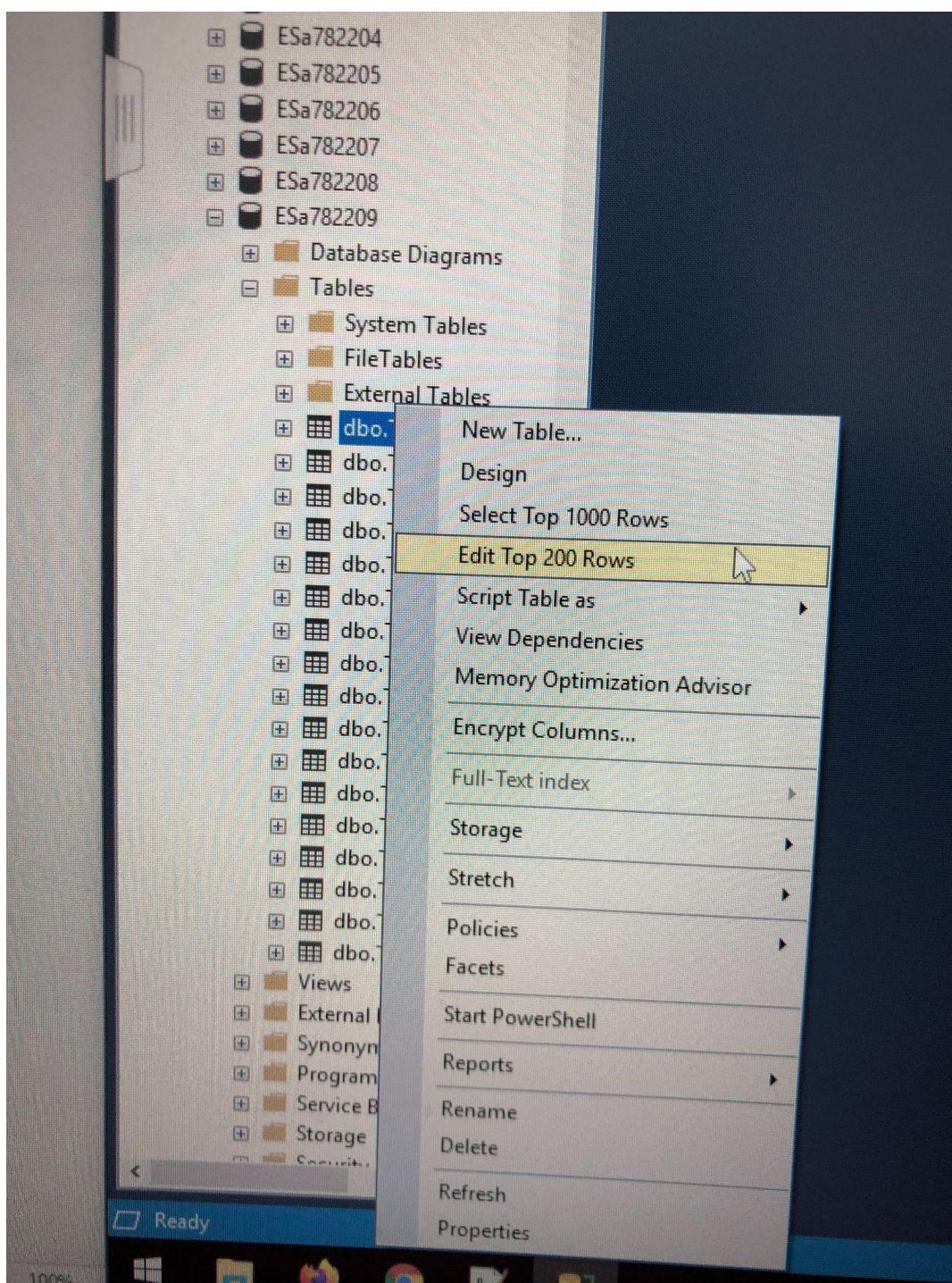
Expand the databases tab on the far left



This is where you can find your database that you have access too. You should have access to two, your individual database, and your group database. You can also type ESA really quickly to help find yours quicker after expanding the databases on the previous step.



From here you can expand your table tab to see what you have been working on. If this is your first time, simply right click tables and click add new.

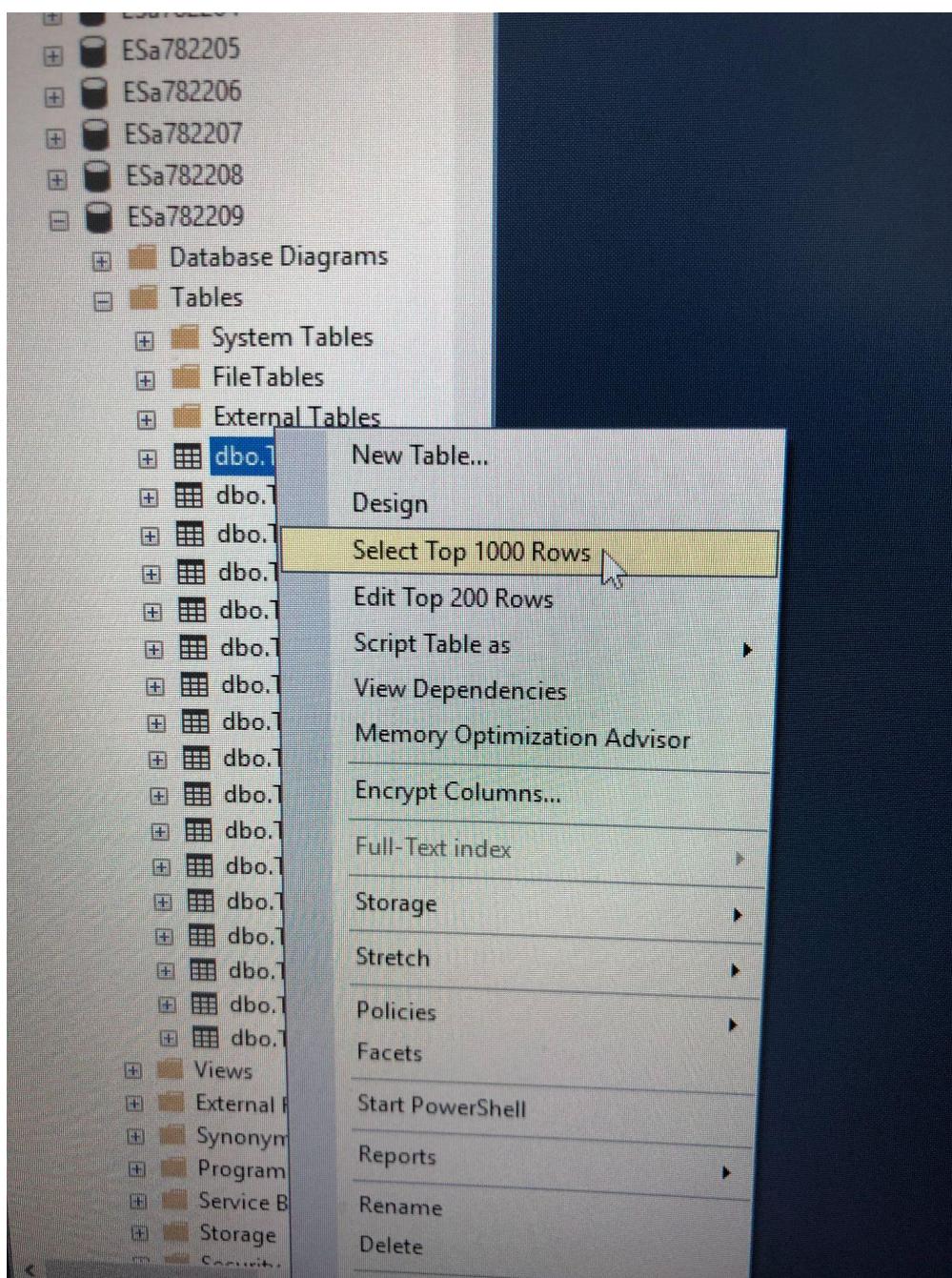


You can right click any of your table and select the edit top 200 rows in order to bring up the screen depicted in the next picture.

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) window titled "ESSQL1.ESa782209 - dbo.TCar - Microsoft SQL Server Management Studio". The window is running in a VMware Horizon session, as indicated by the title bar. The "Object Explorer" pane on the left lists various database objects, including tables like ESa782179 through ESa782209, Database Diagrams, Tables, and System Tables. The main pane displays a table named "dbo.TCar" with three columns: "CarID", "CarCustID", and "CarCMID". The table has 1027 rows of data. The "New Query" button in the toolbar is highlighted with a yellow box. The URL in the browser tab is "waltonlab.uark.edu/portal/webclient/index.html#/desktop".

CarID	CarCustID	CarCMID
1000	2	2
1001	1	3
1002	64	4
1003	31	1
1004	101	4
1005	68	1
1006	58	4
1007	64	2
1008	75	3
1009	102	3
1010	36	3
1011	21	4
1012	55	4
1013	37	1
1014	29	1
1015	77	3
1016	70	1
1017	66	4
1018	35	1
1019	82	3
1020	48	2
1021	81	1
1022	76	2
1023	71	4
1024	62	3
1025	87	1
1026	76	3
1027	3	4

Once this screen displays, you have the ability to edit any of the rows of data you wish as well you are given the option to create a new query from this point on.



You can also click on select top 1000 rows which will display a screen with the query you have already written. In the next picture we will display how to run your query.

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the connection is to `- essql1.walton.uark.edu.ESa782209 (GACL\ESa782212 (158)) - Microsoft SQL Server Management Studio`. The toolbar has several icons, and the status bar shows "09". The main window has a tab labeled "SQL Execute (F5) ...CL\ESa782212 (158)" with a yellow border. Below it is a script pane containing a SELECT statement:

```
***** Script for SelectTopNRows command from SSMS *****
SELECT TOP (1000) [CarID]
    ,[CarCustID]
    ,[CarCMID]
    ,[CarVin]
FROM [ESa782209].[dbo].[TCar]
```

To the left of the script pane is a vertical list of numbers from 782179 to 782209. At the bottom of the screen, there is a "Results" tab showing a table with four columns: CarID, CarCustID, CarCMID, and CarVin. The data is as follows:

	CarID	CarCustID	CarCMID	CarVin
1	1000	2	2	RFG40VGL6EP6918E
2	1001	1	3	ZZW46NUF1AF8668E
3	1002	64	4	RJD65DSF4KD02070
4	1003	31	1	ILL93APM0RR2426D
5	1004	101	4	XSI15AFA7VG6211B
6	1005	68	1	BPT27RTK0HZ0650E
7	1006	58	4	TYG45VS87HS5778W

Once you write your query, highlight over it and then press execute. Your data will funnel into the bottom of the screen as you can see here and this is the same data that pops up when you right click and click edit top 200 rows. From this screen you are also more than welcome to add a new query as you can see above the execute button in this picture. You now have all the details needed to master walton labs and go create your queries.

## What We Learned Throughout This Process

Derrick Landers	I learned the importance of the steps which need to be taken to make a working database. From creating and designing the ERD to using that “map” to input data into it, every single part of this project was a great lesson learned. I did not expect a database to be this much work and enjoyed learning about the process from start to finish.
Tiandra Forrester	In this project, I learned the importance of effective time management. Inorder to successfully construct a database that meets the needs and requirements of the business, it important to allocate adequate amount time to account for revisions, editing, consultations, implementation etc.
Brett Learmont	The importance of receiving feedback from your customer throughout a project is essential to doing a good job. It allows for your team to stay on track, stay on schedule, and feel comfortable about heading down the right path during each stage of your project.
Munkhbileg Munkhburen	I learned how to draw erd. Also, I understood the importance of correct ERD.
Nelly Mabicka	Throughout this process, I have learned the importance of communication with our client and between my team members. It was important to have a correct ERD from the start to allow us to provide our client with efficient work. I also learned that the Physical Design and implementation of different components are really useful to achieve an accurate database and to help those performing maintenance work on the database.

## Team Function/Insights

Since the beginning of this project, our team had one main goal in mind. That was, to create an efficient and effective database that would fulfill Sonner Tire's needs. With each milestone of our project, we learned new things about our dynamic as a team and how each member influenced our effectiveness as a team. In milestone one, we learned that we enjoyed the process of ideation and thrived on new suggestions and ideas. More specifically, when we were presented with a new idea, we explored it to the best of our abilities by asking the necessary questions and by attempting to implement several trial-and-error sessions. In milestone two, we discovered one of our strengths was our ability to remain accountable. Throughout the project, each person insured that they completed their portion and tasks before the deadline. Lastly, we learned that we possess excellent communication skills. Each member shared any major updates or changes concerning the project, consistently and timely. We also shared any major events in our personal lives that could potentially impact the progress of our project. Lastly, our team was very supportive of each member. If any member was struggling to understand a concept or task, we all worked to provide assistance and clarify any confusion. In addition to the positives, we encounter one major challenge, time conflicts. Occasionally we found it difficult to create set meeting times that everyone was able to attend. In order to overcome this, we had to make sacrifices in our personal lives to accommodate the needs of our project. For some of us, that meant either going to bed later or waking up much earlier. Needless to say, overcoming this challenge strengthened our commitment and bond to the team.

# Appendix

## Team Contract

Our group team, The Big Bang, was given an amazing opportunity to work alongside Sonner Tire Company to instill a new database system for improved efficiency and confidence. The project, consisting of Brett, Nelly, Derrick, Tiandra, and Munkhbileg was broken up into three sections. The sections allow for us to receive feedback on each step of the process so that Sonner Tire is given the database they ask for. Our contract is the our agreement and dedication to this project

Team Name: \_\_\_\_\_ Big Bang \_\_\_\_\_ Logo: (*Provide later + upload to your Teams account*)

Team Motto: \_\_\_\_\_ Lets Bang out this code\_\_\_\_\_

---

 **Team Members**

Name	Email	Phone	Strengths	Availability to Meet
Brett Learmont	Brett.G.Learmont@ou.edu	(214) 784-6062	Teamwork and focusing as a group	Tuesday Thursday Friday
Derrick Landers	John.d.landers-1@ou.edu	9189061410	Excel and planning	
Nelly Mabicka	gnellymabicka@ou.edu	4052190836	Teamwork Communication	Sunday Monday Tuesday
Munkhbileg Munkhburen	Mmunkhbileg9@ou.edu			In Mongolia for semester
Tiandra Forrester	Tiandra.s.Forrester-1@ou.edu	918- 282-1677	Team Building and communication	Tuesday Thurday

## Data Dictionary Model

Here is our Data Dictionary composed of tables (Tcustomer, Tcar, TCarManufacturer, TProduct, TTire, TService, TTireModel, TVendor, TDiscount, TEmployee, TPaymentType, TPaymentIN, TPurchaseOrder, TPurchaseOrderLine, TSaleOrder), fields and records.

A	B	C	D	E	F	G	H
Table Name	Field Name	PK?	Date Type	Size	Null	References(Foreign Key)	Sample
TCustomer	CustID	Y	Int(Auto Increment)		not null		1-100
	Name	N	VarChar		not null		
	Address	N	VarChar		not null		
	Email	N	VarChar		not null		
	CreditLimit	N	int		not null		
	Gender	N	VarChar		not null		
TCar	CarID	Y	Int(Auto Increment)		not null		1-100
	CustID	N	Int(Auto Increment)		not null	TCustomer	
	CMID	N	Int(Auto Increment)		not null	TCarManufacturer	
	VIN	N	int		not null		
TCarManufacturer	CMID	Y	Int(Auto Increment)		not null		1-100
	CManufacturerName	N	VarChar		not null		
TProduct	ProductID	Y	Int(Auto Increment)		not null		1-106
	PName	N	varchar		not null		
	PSalePrice	N	int		not null		
	PProductType	N	varchar		not null		
TTire	TProductID	Y	Int(Auto Increment)		not null	TProduct	5-106
	TVendID	N	int		not null	TVendor	
	TQty_OH	N	int		not null		
	TQty_committed	N	int		not null		
	TireName	N	varchar		not null		
	TMileage	N	int		not null		
	TDescription	N	varchar		not null		
TService	SProductID	Y	Int(Auto Increment)		not null	TProduct	2-5
	SDescription	N	varchar		not null		
TTireModel	TireModelID	Y	Int(Auto Increment)		not null		1000-1100
	TiTireID	N	int		not null	TTire	
	TiModelID	N	int		not null	TCarModel	

# Sonner Tire, The Big Bang

51

	<u>CDescription</u>	N	VarChar		not null		
TTireModel	<u>TireModelID</u>	Y	Int(Auto Increment)		not null	TTire	1000-1100
	<u>TiTireID</u>	N	int		not null		
	<u>TiModelID</u>	N	int		not null	TCarModel	
TVendor	<u>VenderID</u>	Y	int(Auto Increment)		not null	1-16	
	VenderName	N	VarChar		not null		
	VenderContactPers	N	VarChar		not null		
	VendAddress	N	VarChar		not null		
	VendCity	N	VarChar		not null		
	VendZip	N	int		not null		
TDiscount	<u>DicountID</u>	Y	int(Auto Increment)		not null	1-10	
	DisAmount	N	int		not null		
	DisStartDate	N	VarChar		not null		
	DisType	N	VarChar		not null		
TEmployee	<u>EmpID</u>	Y	int(Auto Increment)		not null	1-10	
	EmpFName	N	VarChar		not null		
	EmpLName	N	VarChar		not null		
	EmpPosition	N	VarChar		not null		
TPaymenttype	<u>PMTID</u>	Y	int(Auto Increment)		not null	1-5	
	PMTDescription	N	VarChar		Not null		
TPaymentIn	<u>PayInID</u>	Y	int(Auto Increment)		not null	1-100	
	PayInSoID	N	int		not null		
	PayInPMTID	N	int		not null		
	PayDate	N	VarChar		not null		
TVendor	<u>VenderID</u>	Y	int(Auto Increment)		not null	1-16	
	VenderName	N	VarChar		not null		
	VenderContactPers	N	VarChar		not null		
	VendAddress	N	VarChar		not null		
	VendCity	N	VarChar		not null		
	VendZip	N	int		not null		
TDiscount	<u>DicountID</u>	Y	int(Auto Increment)		not null	1-10	
	DisAmount	N	int		not null		
	DisStartDate	N	VarChar		not null		
	DisType	N	VarChar		not null		
TEmployee	<u>EmpID</u>	Y	int(Auto Increment)		not null	1-10	
	EmpFName	N	VarChar		not null		
	EmpLName	N	VarChar		not null		
	EmpPosition	N	VarChar		not null		
TPaymenttype	<u>PMTID</u>	Y	int(Auto Increment)		not null	1-5	
	PMTDescription	N	VarChar		Not null		
TPaymentIn	<u>PayInID</u>	Y	int(Auto Increment)		not null	1-100	
	PayInSoID	N	int		not null		
	PayInPMTID	N	int		not null		
	PayDate	N	VarChar		not null		
	PayAmount	N	int		not null		
TPurchaseOrder	<u>POID</u>	Y	int(Auto Increment)		not null	1-102	
	POEmpID	N	int		not null		
	POVendID	N	int		not null		
	PODtae	N	VarChar		not null		

TPurchaseOrder	<u>POID</u>	Y	int(Auto Increment)		not null		1-102
	POEmpID	N	int		not null		
	POVendID	N	int		not null		
	PODtae	N	VarChar		not null		
	POAmount	N	int		not null		
	POInvoice	N	int		not null		
	POTotal	N	int		not null		
TPurchaseOrderLine	<u>POLID</u>	Y	int(Auto increment)		not null		1-100
	POLPOID	N	int		not null		
	POLProductID	N	int		not null		
	POLQuantity	N	int		not null		
	POLPurchaseCost	N	int		not null		
TSalesOrder	<u>SOID</u>	Y	int(Auto increment)		not null		1-100
	SOCustID	N	int		not null		
	SOEmpIDT	N	int		not null		
	SOSEmpIDS	N	int		not null		
	SOSDiscountID	N	int		not null		
	SOCarID	N	int		not null		
	SOSDate	N	int		not null		
	SOSCarMileage	N	int		not null		
	SOSDropTime	N	int		not null		
	SOServiceStartTime	N	int		not null		
	SOServiceEndTime	N	int		not null		
	SOSPickUpTime	N	int		not null		
	SOisFinance	N	int		not null		
	SOSTotal	N	int		not null		

# Project Management

Project Start Date	3/14/21	Project End Date		Cost (per 60 min)	\$25			
	Student Name	Duration (Min)	% Complete	Planned Minutes	Actual Minutes	Difference Minutes	Subtotal Minutes	Subtotal Cost
<b>Milestone 1</b>								
Read Case + Prepare Questions for client	Brett Learmont	35	100	30		30		
Client Meeting	Brett Learmont	15	100	15	15	0		
ERD Design	Brett Learmont	90	100	60	90	-30		
Assumptions	Brett Learmont	20	100	15	20	-5		
Write-up preparation	Brett Learmont	45	100	30	45	-15		
<b>Sub Total</b>						-20	170	\$71
<b>Milestone 2</b>								
Sonner Tire company sample data	Brett Learmont	70	100	60	70	-10		
Google Doc for our project	Brett Learmont	55	100	60	55	5		
Fixing up the ERD	Brett Learmont	50	100	60	50	10		
<b>Sub Total</b>						5	175	\$73
<b>Milestone 3</b>								
Walton Labs	Brett Learmont	110	100	120	110	10		
Google Doc added detail	Brett Learmont	50	100	60	50	10		
Data Dictionary	Brett Learmont	45	100	60	45	15		
complete ERD	Brett Learmont	30	100	60	30	30		
<b>Sub Total</b>						65	235	\$98
<b>Final Submission</b>								
Fix up walton lab	Brett Learmont	60	100	60	60	0		
fix up the google doc	Brett Learmont	60	100	60	60	0		
write executive summary	Brett Learmont	55	100	60	55	5		
<b>Sub Total</b>						0	175	\$73
							<b>Total</b>	<b>755 \$18,875</b>
Project Start Date	3/14/21	Project End Date		Cost (per 60 min)	\$25			
	Student Name	Duration (Min)	% Complete	Planned Minute: Actual Minutes	Difference Minu	Subtotal Minute	Subtotal Cost	
<b>Milestone 1</b>								
Read Case + Prepare Questions for client	Derrick Landers	40	100	20	15	5		
Client Meeting	Derrick Landers	15	100	25	15	10		
ERD Design	Derrick Landers	70	100	120	130	-10		
Assumptions	Derrick Landers	10	100	15	10	5		
Write-up preparation	Derrick Landers	30	100	60	50	10		
<b>Sub Total</b>						20	220	\$92
<b>Milestone 2</b>								
Normalizing Data	Derrick Landers	45	100	60	45	15		
Fixing ERD	Derrick Landers	60	100	60	60	0		
Project Doc Work	Derrick Landers	35	100	60	35	25		
<b>Sub Total</b>						40	140	\$58
<b>Milestone 3</b>								
Finishing ERD	Derrick Landers	60	100	60	60	0		
Input Data on SQLServer	Derrick Landers	180	100	180	180	0		
Data Dictionary	Derrick Landers	20	100	30	20	10		
Google Doc	Derrick Landers	60	100	60	60	0		
<b>Sub Total</b>						10	320	\$133
<b>Final Submission</b>								
Fix Walton Labs	Derrick Landers	40	100	60	40	20		
Fix ERD	Derrick Landers	10	100	60	10	50		
Fix Google Doc	Derrick Landers	20	100	60	20	40		
<b>Sub Total</b>						0	70	\$29
							<b>Total</b>	<b>750 \$18,750</b>

## Sonner Tire, The Big Bang

54

Project Start Date	3/14/21	Project End Date		Cost (per 60 min)	\$25		
Student Name	Duration (Min)	% Complete	Planned Minutes	Actual Minutes	Difference Min	Subtotal Minute	Subtotal Cost
<b>Milestone 1</b>							
Read Case + Prepare Questions for client	Munduu Munkhbileg	40	100	30		30	
Client Meeting	Munduu Munkhbileg	15	100	15		0	
ERD Design	Munduu Munkhbileg	90	100	60		0	
Assumptions	Munduu Munkhbileg	10	100	10		0	
Write-up preparation	Munduu Munkhbileg	35	100	35		0	
<b>Sub Total</b>					30	120	\$50
<b>Milestone 2</b>							
Sonner Tire company sample data	Munduu Munkhbileg	70	100	70		0	
Google Doc for our project	Munduu Munkhbileg	90	100	90		0	
Fixing up the ERD	Munduu Munkhbileg	50	100	50		0	
<b>Sub Total</b>					0	210	\$88
<b>Milestone 3</b>							
Walton Labs	Munduu Munkhbileg	600	100	200		-300	
Google Doc added detail	Munduu Munkhbileg	60	100	60		0	
Data Dictionary	Munduu Munkhbileg	40	100	40		0	
complete ERD	Munduu Munkhbileg	50	80	50		0	
<b>Sub Total</b>					-300	650	\$271
<b>Final Submission</b>							
Fix up walton lab	Munduu Munkhbileg	150	100	50	150	-100	
fix up the google doc	Munduu Munkhbileg	60	100	60	60	0	
fix sql	Munduu Munkhbileg	150	100	150	150	0	
fix datas	Munduu Munkhbileg	80	100	80	80	0	
<b>Sub Total</b>					-100	440	\$183
						<b>Total</b>	1420 \$35,500

# Sonner Tire, The Big Bang

55

Project Start Date	3/14/21	Project End Date		Cost (per 60 min)	\$25
	Student Name	Duration (Min)	% Complete	Planned Minute: Actual Minutes	Difference Minu Subtotal Minute Subtotal Cost
<b>Milestone 1</b>					
ad Case + Prepare Questions for client	Nelly Mabicka	40	100	30	30
Client Meeting	Nelly Mabicka	15	100	15	0
ERD Design	Nelly Mabicka	60	100	30	-15
Assumptions	Nelly Mabicka	20	100	30	5
Write-up preparation	Nelly Mabicka	30	100	40	5
<b>Sub Total</b>				25	120 \$50
<b>Milestone 2</b>					
Sonner Tire company sample data	Nelly Mabicka	90	100	90	0
Google Doc for our project	Nelly Mabicka	60	100	60	0
Fixing up the ERD	Nelly Mabicka	50	100	50	0
<b>Sub Total</b>				0	200 \$83
<b>Milestone 3</b>					
Walton Labs	Nelly Mabicka	100	100	130	30
Google Doc	Nelly Mabicka	100	100	120	-20
Data Dictionary	Nelly Mabicka	140	100	60	-80
Complete ERD	Nelly Mabicka	30	100	30	0
<b>Sub Total</b>				-70	410 \$171
<b>Final Submission</b>					
fix walton Lab	Nelly Mabicka	30	100	30	0
fix up google doc	Nelly Mabicka	60	100	60	0
Fix Up data dico	Nelly Mabicka	40	100	40	0
<b>Sub Total</b>				40	130 \$54
				<b>Total</b>	860 \$21,500
Project Start Date	3/14/21	Project End Date		Cost (per 60 min)	\$25
	Student Name	Duration (Min)	% Complete	Planned Minute: Actual Minutes	Difference Minu Subtotal Minute Subtotal Cost
<b>Milestone 1</b>					
Read Case + Prepare Questions for client	Tiandra Forrester	40	100	30	30
Client Meeting	Tiandra Forrester	15	100	15	0
ERD Design	Tiandra Forrester	90	100	60	0
Assumptions	Tiandra Forrester	30	100	25	-5
Write-up preparation	Tiandra Forrester	40	100	35	-10
<b>Sub Total</b>				15	150 \$63
<b>Milestone 2</b>					
Sonner Tire company sample data	Tiandra Forrester	70	100	70	0
Google Doc for our project	Tiandra Forrester	50	100	45	-5
Fixing up the ERD	Tiandra Forrester	75	100	70	-5
<b>Sub Total</b>				-10	195 \$81
<b>Milestone 3</b>					
Walton Labs	Tiandra Forrester	100	100	110	10
Google Doc added detail	Tiandra Forrester	130	100	100	30
Data Dictionary	Tiandra Forrester	45	100	60	15
complete ERD	Tiandra Forrester	30	100	30	0
<b>Sub Total</b>				25	305 \$127
<b>Final Submission</b>					
Fixed walton lab	Tiandra Forrester	40	100	40	0
Add header and edited doc (fonts, layout etc.)	Tiandra Forrester	30	100	30	-5
write intro to conceptual design	Tiandra Forrester	60	100	60	0
write challenges faced and address	Tiandra Forrester	60	100	60	0
write team insights and interctions	Tiandra Forrester	60	100	60	0
fixed table of contents	Tiandra Forrester	45	100	45	0
<b>Sub Total</b>				-5	195 \$81
				<b>Total</b>	845 \$21,125