The University of Oklahoma

A picture containing clipart, vector graphics

Description automatically generated Wolf

GANG

Team Members: Matthew Brinkmeyer, Imran Ahmed, Eric Anderson, Brianna Wells, Alec Amend

Class: MIS 3353 – Database Management

Professor Swetha Siripurapu

Five wolves… ONE pack

# **Executive Summary**

The purpose of the project is to outline a database management system for La Cocina de la Abuela Taco Truck to implement for their company. There was a three-step process in creating the database management system – Conceptual Design, Logical Design, and Physical Design. In the first phase, conceptual design, significant assumptions were made while creating the ERD (Entity Relationship Diagram). ERD’s were used to conceptualize and map out what data needs to be modeled and how each object is related to one another. Queries were formed to pull out information created from the ERD’s, and questions relevant to the business needs were used to perform these queries. In the second phase, logical design information taken from the conceptual design stage was normalized. Each of our relations represents 3NF which means that they are atomic and have no data redundancy. Normalized relations were created for each of the entities used in the database. After relations were created, in the third phase, physical design, this is where the information was inserted into the database. Through the Microsoft SQL Server on Walton.Uark.Edu, information can be typed into queries to output data that the client might need. A variety of different information can be out-putted like the number of items ordered in a single month, or the names of all the employees and past customers. The opportunities are endless with this database, providing our client with an accessible way to track and gather data about their company. The project management table represents the pricing of what it would cost to implement this database for La Cocina de la Abuela Taco Truck. Pricing for each phase of the implementation process lists the amount of hours each step took, as well as the cost per hour. The total cost per hour is $25 per hour, with a combined total of 7835 minutes to implement the database. The total cost for the implementation of the database is $3,265. This price covers the cost for each step in developing the database management system.

Contents

[Executive Summary 2](#_Toc89960078)

[Get to Know the Team: (Team Name) 4](#_Toc89960079)

[Conceptual Design 6](#_Toc89960080)

[The Client Meeting 6](#_Toc89960081)

[Q&A During the Meeting & Information We Learned 6](#_Toc89960082)

[Significant Assumptions 7](#_Toc89960083)

[What is an ERD? Why is it necessary? 8](#_Toc89960084)

[Business Cycles Used 8](#_Toc89960085)

[Data Provided by Client 9](#_Toc89960086)

[ERD Created 10](#_Toc89960087)

[Query Feasibility and Current ERD 11](#_Toc89960088)

[Logical Design 16](#_Toc89960089)

[Normalization 16](#_Toc89960090)

[Normalization of the Data Provided by the Client 16](#_Toc89960091)

[Normalized Relations 16](#_Toc89960092)

[Differences between ERD and Normalized Relations 19](#_Toc89960093)

[Database Integrity 19](#_Toc89960094)

[Physical Design and Implementation 21](#_Toc89960095)

[Data Dictionary 21](#_Toc89960096)

[Denormalization 21](#_Toc89960097)

[Implemented Physical Design 22](#_Toc89960098)

[Challenges Faced/Addressed During Implementation 23](#_Toc89960099)

[Strengths and Weaknesses Encountered During Implementation 23](#_Toc89960100)

[Specific SQL Statements Requested 24](#_Toc89960101)

[Three Additional Queries 30](#_Toc89960102)

[User Documentation 33](#_Toc89960103)

[What We Learned Throughout This Process 39](#_Toc89960104)

[Appendix 41](#_Toc89960105)

[Team Contract 41](#_Toc89960106)

[Data Dictionary Model 43](#_Toc89960107)

[Project Management 45](#_Toc89960108)

# **Get to Know the Team: (Team Name)**

A person wearing glasses

Description automatically generated with medium confidence

* Imran Ahmed
* Management Information Systems
* Junior
* Pakistani American born in Oklahoma.
* I like playing and watching sports, stocks and crypto, and movies.

A picture containing person

Description automatically generated

* Matthew Brinkmeyer
* Management Information Systems
* Junior
* From the great state of Dallas, Texas
* I enjoy golfing and bowling in my free time

A person smiling for the camera

Description automatically generated with medium confidence

* Eric Anderson
* Management Information Systems
* Junior
* From Prosper, Texas
* I enjoy playing Rocket League for OU Esports

A picture containing text, person, building, outdoor

Description automatically generated

* Alec Amend
* Management Information Systems
* Senior
* From Edmond, Oklahoma
* I enjoy film and watching sports

A person standing in front of a wall

Description automatically generated with low confidence

* Brianna Wells
* Marketing with Management Information Systems Minor
* Junior
* From Moore, Oklahoma
* I am a mom to a Corgi puppy

# **Conceptual Design**

This is the early stage of creating the database for our client. In this stage we will meet with the client to ask questions and then start to build the database using assumptions and the answers to questions that were asked. This stage is meant for modelling the data in a way that makes sense when looking at the database with a broad perspective. It is not explaining the functional side of the database or giving the minute details that you would need to make a working database. This stage is meant to set the stage for the logical and physical design later in the design process. The first step in creating the conceptual design is using Lucid charts to create entity relationship diagrams to model the data given by the client. These diagrams help picture where the information is being stored and what relationships the data has to each other. The diagram provided in the document is the output of the conceptual design phase of implementation of the database.

## **The Client Meeting**

Wolf Gang was put to the task of creating a Database for La Cocina de la Abuela Taco Truck. As a team, we came together to examine the use case and create questions for our client’s needs. After meeting with the client, we were able to create assumptions and formulate an Entity Relationship Diagram.

* Meeting Time: 10/12 12:30 pm
* Location: Zoom Meeting
* Interviewers: Eric Anderson, Matthew Brinkmeyer, Imran Ahmed, Colby Brabant, Alec Amend
* Interviewee: Alvaro

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

During the meeting we asked our client, Alvaro, questions in order to gather information to create the database. The questions we asked helped us gather information to determine the specifics of what we needed in the ERD. We were trying to gather all the minute details that would help us create the best ERD for our client. This process was extremely helpful, and we learned a lot of things that we no longer had to assume about which makes it more exact for our client.

1. Can a customer overlap between veteran and student and long term?
   1. A customer can only get one type of discount.
2. Any there any ingredients needed that aren’t listed on the Menu?
   1. Everything that we are using is on there.
3. Is the discount the same for veterans, students, etc.?
   1. Yes.
4. If you order something less than $10 and have a $10 discount, is it free?
   1. Yes.
5. How many herbs are in your garden?
   1. Is that treated as a vendor?
      1. No.
   2. Is it free?
      1. Yes.
6. Is every customer's visit counted towards the 11th-meal discount?
   1. Has to be an entrée to count
7. What counts as a student?
   1. The customer must be able to show ID card, but they can be any age going to school.
8. Do you have tax on your sales?
   1. There are sales taxes (8.5%) and a credit card $1 fee.
9. Can customers be in more than one category?
   1. yes, but only 1 is needed to count for discount.
10. How long is a long-time customer?
    1. You could track through loyalty program week by week and count, or by the 11th meal
11. Are certain ingredients only bought at certain vendors?
    1. Doesn’t matter
12. Are the only drinks the ones listed?
    1. Yes
13. Can customers opt out of text message or email?
    1. Doesn’t matter
14. Employee task can it be overlapped (can they be more than one?)
    1. Yes
15. Open more food trucks? Sell your herbs? Open actual store?
    1. No other food trucks
16. How long is a time period for a small discount?
    1. No determined length, just 11th meal discount can apply.
17. If a customer wants to modify their food to take out/add ingredients, how do you price that?
    1. If someone adds something you need to add to price
18. Should we model food going bad?
    1. Yes, procedures should be made for the wastage of food.

# **Significant Assumptions**

During the process of making an ERD sometimes you run into questions that are not fully answered and not everything is known, so we must assume what we think is best for our client’s needs. These are some of the more important things that we had to make assumptions on while working on the ERD for you.

* We assumed that La Cocina de la Abuela Taco Truck does not do delivery in or delivery out because the truck is not delivering any food to customers. They are also not taking any deliveries from the vendors. We also assume that we are picking everything up. This affected the ERD by making us delete the delivery in and out on the ERD and having to make sure that someone would go pick up the ingredients when we need them.
* We assumed that there are ingredients that go to waste, this affected the ERD by adding a reference table to ingredients that would let us know when certain ingredients would go bad.
* We assumed that we needed a payment out to track our orders from our vendors since they would typically be bigger than the average consumer. So, we wanted to provide that extra security for our client. This did not impact the ERD too much, we just had to add something that would allow us to track this information.
* We had to assume how they wanted to model all the things on the menu. Like: drinks, sides, combos, and meals. We decided it would be best to make all of those reference tables because it helps keep it clean and more uniform. This changed how we decided to model the ERD vs how others might model it.
* We assumed that the various kinds of customers would be a reference table from customers instead of making it a super-type sub-type relationship. This changed how the ERD looks and how it gets modeled. We have a reference table that has the kind of customer that they are rather than making a table for each kind of customer there could be. We do this because it makes it easier to change in the future if they wish to have more kinds of customers.
* We had to assume the employee type, like the customer type. We decided to make it a reference table rather than making it a super-type sub-type relationship. This changes how the ERD looks and how it gets modeled. There are fewer things for us to model like: having to name the various kinds of employees that they could have. We do this because if they wish to have more kinds of employees it makes it easier to change in the future.
* We had to assume that that there was only one discount that a customer could yes during their purchase. They cannot use the student/military discount and the small discount customers get for not coming in for a while.

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

An ERD is an Entity Relationship Diagram, otherwise known as a Database Schema. ERD’s are used to conceptualize and map out what data needs to be modeled and how each object is related to one another. The three main elements of an ERD are the entities, the attributes, and the relationships. These components allow a person to see the diagram and make sense out of it. In this stage, we are defining what these 3 components will consist of for our client. The diagram we create can be referenced and used to implement a functional database. In this situation we must use and create an ERD to keep track of sales, customer information, raw materials, purchases, and production. For example, we must track the inventory of ingredients and decide if ingredients need to be purchased from vendors. Another example is tracking customer information and their purchase to decide whether a discount is necessary or not.

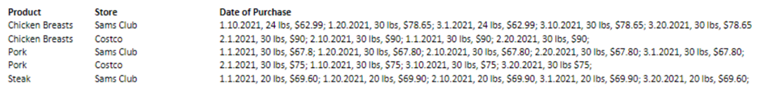
## **Business Cycles Used**

We used the revenue cycle, expenditure cycle, and production cycle. The revenue cycle is used to provide the right product, at the right time, at the right place, for the right price. There is different data that is tracked through the revenue cycle like sales order and cash collections. The taco truck is selling food items to customers which requires the revenue cycle. The purpose for the expenditure cycle is to minimize the total cost of acquiring and maintaining inventories, supplies, and other necessary services needed to run the business. Choosing suppliers, ordering supplies and ingredients, and cash disbursement are a few of the necessary data that a company needs in the expenditure cycle. The taco truck is spending money on things like ingredients and food items which requires the expenditure cycle. The production cycle is needed to efficiently design, plan the production of, and manufacture products. The bill of materials, planning and scheduling of products are all necessary to this cycle. The taco truck is making food items using raw materials which requires the production cycle. All three of these cycles are necessary for the conceptual design phase of creating and implementing a functional database.

## **Data Provided by Client**

We were provided data from the client about all the different ingredients and where they buy them from (the vendors). We were also provided data for the date of purchase, the weight/count, and the price. The product information would be stored in the Ingredient entity, the Store would be stored in the Vendor entity, and the date of purchase would be stored in the Vendor/Ingredient entity.

From this data we will be able to make sure we can record each different attribute (date of purchase, weight/count, and price). We also will make sure that there is historical data that is saved so we know what was purchased 3 months ago, for how much, and what size it was.



# 

# **ERD Created**

# Query Feasibility and Current ERD

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Query Question** | **Tables needed to run the query** | **Projected SQL Statement** |
| 1 | Show how many times and how much money was spent on each wholesaler in the past month. | TVendor/Ingredient  TVendor | Select VName,SUM(VICurrentPrice) as Total  From [TVendor/Ingredient] join TVendor on TVendor.VendorId = [TVendor/Ingredient].VIVendID  Where VIDateBought between '2021/02/01' and '2021/03/31'  Group By VName |
| 2 | How much money was spent on each category of products purchased (e.g., chicken, pork, cheese, plates)? List this for each month | TVendor/Ingredient TIngredient | Select count(VICurrentPrice) numberoftimes, month(VIDateBought) MonthBought, IName, sum(VICurrentPrice) total  From [TVendor/Ingredient] join TIngredient on [TVendor/Ingredient].VIIngID = TIngredient.IngID  Group By month(VIDateBought) , IName |
| 3 | Calculate the number of visits and also the total amount spent for each customer last month | TCustomer  TSalesOrder TSalesOrderLine TMenuItem | SELECT CUSTID, SUM(MISALEPRICE) TOTALSPENT, Count(SOID) TOTALVISITS  FROM TCUSTOMER C JOIN TSALESORDER SO  ON C.CUSTID = SO.SOCUSTID JOIN TSALESORDERLINE SOL  ON SO.SOID = SOL.SOLSOID JOIN TMENUITEM MI  ON SOL.SOLMENUITEMID = MI.MENUITEMID  GROUP BY CUSTID |
| 4 | List the names of customers who received at least one free order (I.e., the $10 discount for their | TCustomer | SELECT CFIRSTNAME, CLASTNAME  FROM TCUSTOMER  WHERE CVISIT = 'YES' |
| 5 | List the names and phone numbers of customers who have not purchased in the last 3 months | TCustomer TSalesOrder | SELECT CFirstName, CLastName, CPhone  FROM TCUSTOMER C JOIN TSALESORDER SO ON C.CUSTID = SO.SOCustID  Where SOVistDate not between '2021-08-11' and '2021-11-21' |
| 6 | What are the 10 most ordered items on the menu? | TSalesOrder TSaleOrderLine TMenuItem | SELECT MIName  FROM TSALESORDER SO JOIN TSALESORDERLINE SOL ON SO.SOID = SOL.SOLSOID JOIN TMENUITEM MI ON SOL.SOLMENUITEMID = MI.MENUITEMID  Group by MIName  ORDER BY COUNT(MenuItemID) DESC |
| 7 | List the number of the times each combo was purchased in the last 3 months? | TCombo TMenuItem TSalesOrderLine TSalesOrder | Select count(MIName) 'Number of Combos Ordered'  From TCombo join TMenuItem on TCombo.[COMenuItem - Kit] = TMenuItem.MenuItemID join TSalesOrderLine on TMenuItem.MenuItemID = TSalesOrderLine.SOLMenuItemID join TSalesOrder on TSalesOrder.SOID = TSalesOrderLine.SOLSOID  Where SOVistDate between '2021/01/01' and '2021/03/30' |
| 8 | List total sales and total cost per month since January 1. | TVendor/Ingredient  TMenuItem | Select SUM(VICurrentPrice)as 'Total Money Spent'  From [TVendor/Ingredient]  Select SUm(MISalePrice) as 'Total Money Earned'  From TMenuItem |
| 9 | List the names of each employee, their phone number, dob, and start date. | TEmployee | SELECT EFIRSTNAME, ELASTNAME, ECELLPHONE, EDOB, EHIREDATE  FROM TEMPLOYEE |
| 10 | List the names of each employee and show how many customers they served every month. | TCustomer  TSaleOrder  TEmployee | SELECT EFIRSTNAME, ELASTNAME, COUNT(SOID) NumberOfCustomer  FROM TCUSTOMER C JOIN TSALESORDER SO ON C.CUSTID = SO.SOCustID JOIN TEMPLOYEE E on E.EmpID = SO.SOEmpID  Group By EFirstName, ELastName |
| 11 | Customers like to tip. Tips are equally split between all employees working on a particular day. Calculate the amount collected in tips for the past 3 months. Display it week by week. | TEmpSchedule  TEmployee | Select ESMyTip, EFirstName,ELastName,DATEPART(Week,ESDate) as Week  From TEmpSchedule join TEmployee on TEmpSchedule.ESEmpID = TEmployee.EmpID  Group By ESMyTip, EFirstName,ELastName, DATEPART(Week,ESDate)  Order By DATEPART(Week,ESDate) |

# **Logical Design**

Now that we have designed and recognized the conceptual design, we are moving into creating the Logical Design to add more details to achieve the user's needs. Logical Design expects the database to have integrity and be efficient, this is achieved by normalization. The relation requires an evaluation of its atomicity, partial-functional-dependency, transitive-dependency to determine the normal form of the relation. Each of our relations represents 3NF which means that they are atomic and have no data redundancy. After assuring normal forms we write out the relations and foreign key constraints. After completing all of these things, we were able to create a Logical Designed database that contains atomicity and no data redundancy.

## **Normalization**

Normalization is the way to organize data within an organization’s database and display the data in a way that is easy to visualize and comprehend. This will allow us to go through all your information and sift out any data redundancies and break them down to make sure we display only the data that is crucial for your restaurant. By implementing normalization through our project, we promise you that we will deliver a swift and effortless system of data that will eliminate wasting unnecessary time trying to sort through the database and will make your workdays more efficient and effective.

## **Normalization of the Data Provided by the Client**

We normalized the data given to us by our client mainly by deconstructing what was given to us under the Date of Purchase column. We recognized that there were three main components of purchase data in this column. There was a date, weight, and cost associated with each ingredient from a certain vendor/store. These three components are going to be tracked in the Vendor/Ingredient associative entity as three separate attributes: DateBought, CurrentPrice, and Weight.

# **Normalized Relations**

TDrink (DrinkID, DName)

TSide (SideID, SName)

TMeal (MealID, MName, MTime )

TWaste (WasteID, WasteDesc)

TVendor (VendID, VName)

TDiscountType (DiscountTypeID, DTType)

TDay (TipID, TDate,TJarTip)

TIngredient (IngID, IngName, IngCost, IngExpirationDate)

TCustomer (CustID, CPhone, CEmail, CStudentVet, CFirstName, CLastName, CVisit)

TEmployee (EmpID, EHireDate, ECellPhone, EFirstName, ELastName)

TEmpSchedule (EmpScheduleID, ESTipID\*, ESEmpID\*, ESDate, ESMyTip, ESTimeIn, ESTimeOut)

Foreign Key ESTipID references TTipType

Null Allowed

On Delete Set Null

Foreign Key ESEmpID references TEmployee

Null Allowed

On Delete Set Null

TSalesOrder (SalesOrderID, SOCustID\*, SOEmpID\*, SODiscountTypeID\*, SOVistDate, SOTipFromCC)

Foreign Key SOCustID references TCustomer

Not Null

On Delete Restrict

Foreign Key SOEmpID references TEmployee

Not Null

On Delete Restrict

Foreign Key SODiscountTypeID references TDiscountType

Not Null

On Delete Restrict

TMenuItem (MenuItemID, MIMealID\*, MIDrinkID\*, MISideID\*, MIName, MISalePrice)

Foreign Key MIMealID references TMenuItem

Null Allowed

On Delete Set Null

Foreign Key MIDrinkID references TMenuItem

Null Allowed

On Delete Set Null

Foreign Key MISideID references TMenuItem

Null Allowed

On Delete Set Null

TSalesOrderLine (SOLID, SOLMenuItemID\*, SOLSOID\*, SOLQuantity, SOLCost, SOLDate)

Foreign Key SOLMenuItemID references TMenuItem

Not Null

On Delete Restrict

Foreign Key SOLSOID references TSalesOrder

Not Null

On Delete Restrict

TCombo (ComboID, ComboMenuItem/Kit\*, ComboMenuItem/Type\*, ComboName, ComboDesc)

Foreign Key ComboMenuItem/Kit references TMenuItem

Null Allowed

On Delete Set Null

Foreign Key ComboMenuItem/Type references TMenuItem

Null Allowed

On Delete Set Null

TIngredient/Waste (IngTypeID, IWIngID\*, IWWasteID\*, IWDate)

Foreign Key IWIngID references TIngredient

Not Null

On Delete Restrict

Foreign Key IWWasteID references TWaste

Not Null

On Delete Restrict

TVendor/Ingredient (VendorIngredientID, VIVendID\*, VIIngID\*, VIDateBought, VICurrentPrice)

Foreign Key VIVendID references TVendor

Not Null

On Delete Restrict

Foreign Key VIIngID references TIngredient

Not Null

On Delete Restrict

TPurchaseOrder (POID, POVendID\*, PODate)

Foreign Key POVendID references TVendor

Not Null

On Delete Restirct

TPurchaseOrderLine (POLID, POLPOID\*, POLIngID\*, POLQuantity)

Foreign Key POLPOID references TPurchaseOrder

Not Null

On Delete Restrict

Foreign Key POLIngID references TIngredient

Not Null

On Delete Restrict

\* = Foreign Key

## **Differences between ERD and Normalized Relations**

The differences between ERD and Normalized Relations are that we got rid of all of the composite and derived attributes. We do this because we want to make the data less complicated and redundant. ERDs are diagrams that show the logical entries and relationships between them. Normalization is what helps create the database. You try to remove the redundancy from all the data also known as the one face, one place rule. You do this so that if you have the same information stored in many places it can become out of sync and make the queries almost impossible to do. In effect, normalization is used to help keep our database more efficient and intuitive.

## **Database Integrity**

A referential integrity restraint states that if there is a relationship between entities, the foreign key must match a valid primary key, or be null. An entity integrity restraint states that a primary key must exist and cannot be null. A domain integrity restraint states that all values in a column must be from the same domain. As a group, to ensure that these three database integrity rules are enforced, we have normalized the ERD to follow these rules. All the entities have valid primary keys, and their foreign keys match these primary keys. The primary keys that are in the ERD are not null and contain data within these keys. All the values in the columns are from the same domain.

# **Physical Design and Implementation**

Now that we have Normalized and ensured data integrity, we moved on to implementing our database in the Physical Design phase. The main goal of the Physical Design phase is to ensure that our database works efficiently. Implementation requires that all data attributes be specified in the data dictionary. The Data Dictionary gives additional information including primary keys, foreign keys, description, and the data type. Additionally, to make the database run efficiently we denormalized the tables to reduce the number of tables to get information faster. We are using Microsoft SQL Server to implement our design into a database.

## **Data Dictionary**

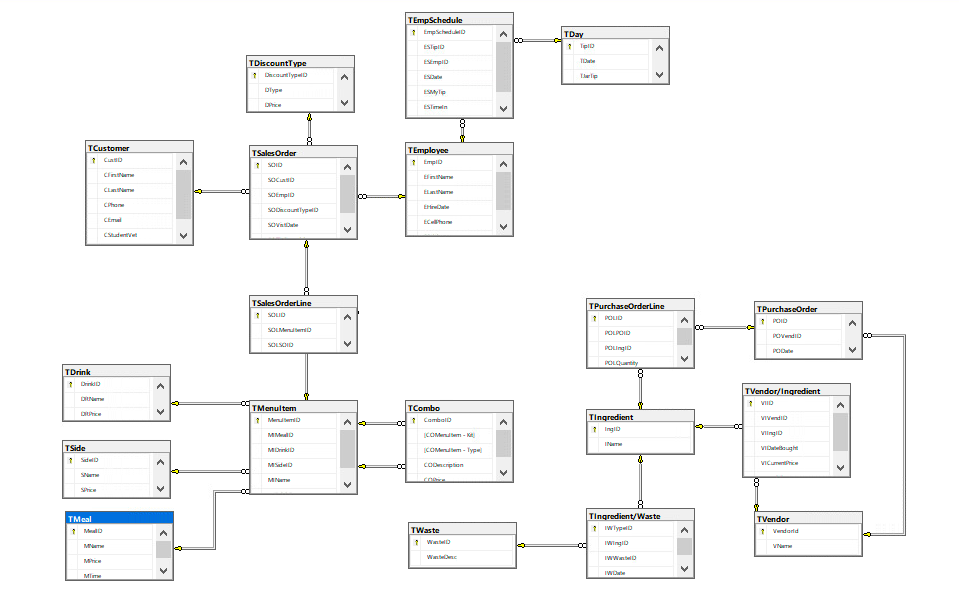
A data dictionary is used during the Physical Design phase of the database creation process. The data dictionary gives additional information to explain the data fields and give further information to the developer. For example, the data dictionary gives details of the data type for each entity and its attributes. Furthermore, it describes if it is a primary or foreign key, and the description of the data comprised in the attribute.

## **Denormalization**

Denormalization is defined as limiting the number of tables within an ERD, by adding redundant information into a database to increase the reading speed of the database. This is necessary for our ERD due to our read and join operations being slow and not as efficient as it would be with denormalizing the database. With the amount of data that we have within our database - without denormalizing it - will get confused when sorting through and checking all the different tables and will decrease the process time and may even crash the entire database.

## 

## **Implemented Physical Design**



## **Challenges Faced/Addressed During Implementation**

One of the biggest challenges when creating the database was limited accessibility to the walton.uark.edu database as a group. The information was limited to who could access it at a given time, which made it difficult as a team to enter information into the database. As a result of this challenge, as a group we decided it was best to get together and work on it as a team to accomplish entering all the information into the database.

Another challenge that we had when creating the database was not having enough information in general to complete some of the information needed to complete each table. Overall, a lot of assumptions were made as a group so that we could complete each database. The assumptions that we made throughout implementation were made based on prior experience from other team members in the group on what a food truck business should have in their database.

## **Strengths and Weaknesses Encountered During Implementation**

Some of the strengths when building ERD’s and writing queries were that as a group, we were able to accomplish a lot during implementation. An assignment like building a database takes a lot of work and is a team effort, and it should not be done individually. As a group we all met regularly to work together and divided up tasks to complete each step of implementation. A step that we took extra to make the process of implementation easier was creating many varieties of diagrams in Lucid Charts, the varieties of these charts made it easier for us as a group to make sure everything was included in each step of the implementation process.

One of the different weaknesses that we had as a group was that not everything could be accessed together as a group. The database provided by walton.uark.edu was not group friendly and made it limited to who in the group could access it at a given time. At times we utilized Microsoft Excel and it was also limited to how many people could access it as a group. There were a lot of assumptions that we had to make as a group, which was a weakness because it made it difficult at times to write a certain table.

# **Specific SQL Statements Requested**

These are saved in our database under Programmability, Stored Procedures as Query1, Query2, Query3, Query4, Query5, Query6, Query7, Query8, Query9, Query10 and Query11. To see them right click on them and click “Execute Stored Procedure” then click “OK” and then on the bottom click “Results”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Query #** | **Question** | **SQL** | **Partial Output** |
| 1 | Show how many times and how much money was spent on each wholesaler in the past month. | Select VName,SUM(VICurrentPrice) as Total  From [TVendor/Ingredient] join TVendor on TVendor.VendorId = [TVendor/Ingredient].VIVendID  Where VIDateBought between '2021/02/01' and '2021/03/31'  Group by VName |  |
| 2 | How much money was spent on each category of products purchased (e.g., chicken, pork,  cheese, plates)? List this for each month. | Select DISTINCT IName, VICurrentPrice, Count(MONTH(VIDateBought)) MonthBought  From [TVendor/Ingredient] join TIngredient on [TVendor/Ingredient].VIIngID = TIngredient.IngID  Group By IName, VICurrentPrice  Order by Count(month(VIDateBought)) |  |
| 3 | Calculate the number of visits and also the total amount spent for each customer last month | SELECT CUSTID, SUM(MISALEPRICE) TOTALSPENT, Count(SOID) TOTALVISITS  FROM TCUSTOMER C JOIN TSALESORDER SO  ON C.CUSTID = SO.SOCUSTID JOIN TSALESORDERLINE SOL  ON SO.SOID = SOL.SOLSOID JOIN TMENUITEM MI  ON SOL.SOLMENUITEMID = MI.MENUITEMID  GROUP BY CUSTID |  |
| 4 | List the names of customers who received at least one free order (I.e., the $10 discount for their meal) | SELECT CFIRSTNAME, CLASTNAME  FROM TCUSTOMER  WHERE CVISIT = 'YES' |  |
| 5 | List the names and phone numbers of customers who have not purchased in the last 3 months | SELECT CFirstName, CLastName, CPhone  FROM TCUSTOMER C JOIN TSALESORDER SO ON C.CUSTID = SO.SOCustID  Where SOVistDate not between '2021-08-11' and '2021-11-21' |  |
| 6 | What are the 10 most ordered items on the menu? | SELECT MIName  FROM TSALESORDER SO JOIN TSALESORDERLINE SOL ON SO.SOID = SOL.SOLSOID JOIN TMENUITEM MI ON SOL.SOLMENUITEMID = MI.MENUITEMID  Group by MIName  ORDER BY COUNT(MenuItemID) DESC |  |
| 7 | List the number of times each combo was purchased in the last 3 months. | Select count(MIName) 'Number of Combos Ordered'  From TCombo join TMenuItem on TCombo.[COMenuItem - Kit] = TMenuItem.MenuItemID join TSalesOrderLine on TMenuItem.MenuItemID = TSalesOrderLine.SOLMenuItemID join TSalesOrder on TSalesOrder.SOID = TSalesOrderLine.SOLSOID  Where SOVistDate between '2021/01/01' and '2021/03/30' |  |
| 8 | List Total Sales and Total Cost per month since January 1. | Select SUM(VICurrentPrice)as 'Total Money Spent'  From [TVendor/Ingredient]  Select SUm(MISalePrice) as 'Total Money Earned'  From TMenuItem |  |
| 9 | List the names of each employee, their phone number, DOB, and start date. | SELECT EFIRSTNAME, ELASTNAME, ECELLPHONE, EDOB, EHIREDATE  FROM TEMPLOYEE |  |
| 10 | List the names of employees and show how many customers they served every month. | SELECT EFIRSTNAME, ELASTNAME, COUNT(SOID) NumberOfCustomer  FROM TCUSTOMER C JOIN TSALESORDER SO ON C.CUSTID = SO.SOCustID JOIN TEMPLOYEE E on E.EmpID = SO.SOEmpID  Group By EFirstName, ELastName | Inserting image... |
| 11 | Customers like to tip. Tips are equally split between all employees working on a particular day. Calculate the amount collected in tips for the past 3 months. Display it week by week. | Select ESMyTip, EFirstName,ELastName,DATEPART(Week,ESDate) as Week  From TEmpSchedule join TEmployee on TEmpSchedule.ESEmpID = TEmployee.EmpID  Group By ESMyTip, EFirstName,ELastName, DATEPART(Week,ESDate)  Order By DATEPART(Week,ESDate) |  |

## 

## **Three Additional Queries**

These are saved in our database under Programmability, Stored Procedures as Extra1, Extra2, and Extra3. To see them right click on them and click “Execute Stored Procedure” then click “OK” and then on the bottom click “Results”.

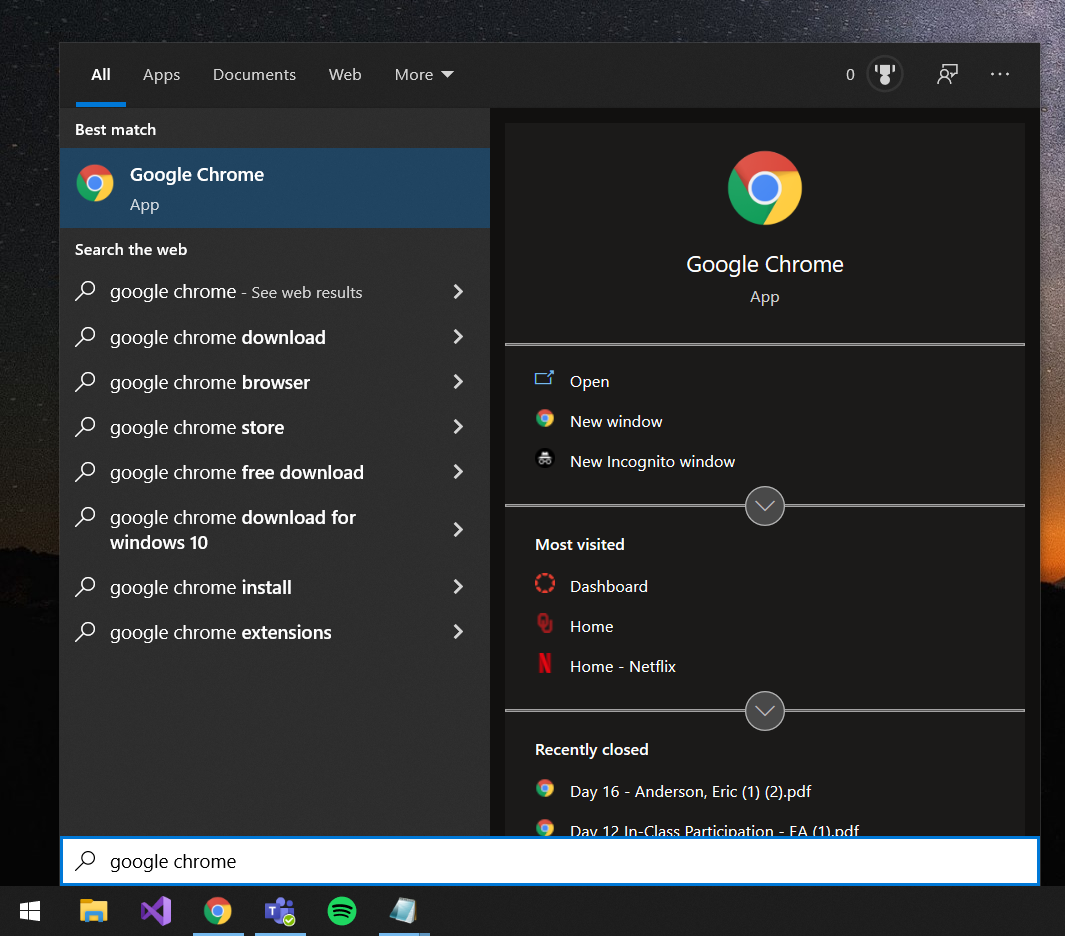
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Query # | Question | Why is this important | SQL | Partial Output | Recap of Findings |
| 1 | What ingredients are in waste from last month | So the company can see what went bad from last month to be able to buy more ingredients. | Select I.IName, IW.IWDATE AS DateGoneBad  From TIngredient I JOIN [TIngredient/Waste] IW on I.IngID = IW.IWIngID  Where I.IName not like ('%Plastic%') and I.IName not like ('%Plates%') and I.IName not like ('%Napkins%') |  | This shows what items went bad and on what date they went bad. |
| 2 | When did the employees clock in and clock out | This is important to the company so they may supervise the punctuality of each employee and if they were late or not. | Select ESDate, ESTimein, ESTimeout, EFirstname, Elastname  From TEmpSchedule ES join TEmployee E on ES.ESEmpID = E.EmpID |  | This shows the employees name and then what time they clocked in at and what time they clocked out at. |
| 3 | What is the total discount amount from month of march | To see an estimate of what the company should expect for discounts for a month. | Select COUNT(SODiscountTypeID) \*5 as 'Student Vet Total Discount Amount'  From TSalesOrder  Select COUNT(SODIscountTypeID) \* 10 as 'Tenth Visit Total Discount Amount'  From TSalesOrder  Where SODiscountTypeID = 2002 and SOVistDate between '2021/03/01' and '2021/03/31' |  | This is to show the total discount amount for the month of March |

# **User Documentation**

Now that we have explained our process of implementing our database in SQL Server, we will now include a step-by-step process of how to access the database and query it for information. This section will include the steps to go from a fresh-booted computer to the database queries on the Walton Virtual Environment.

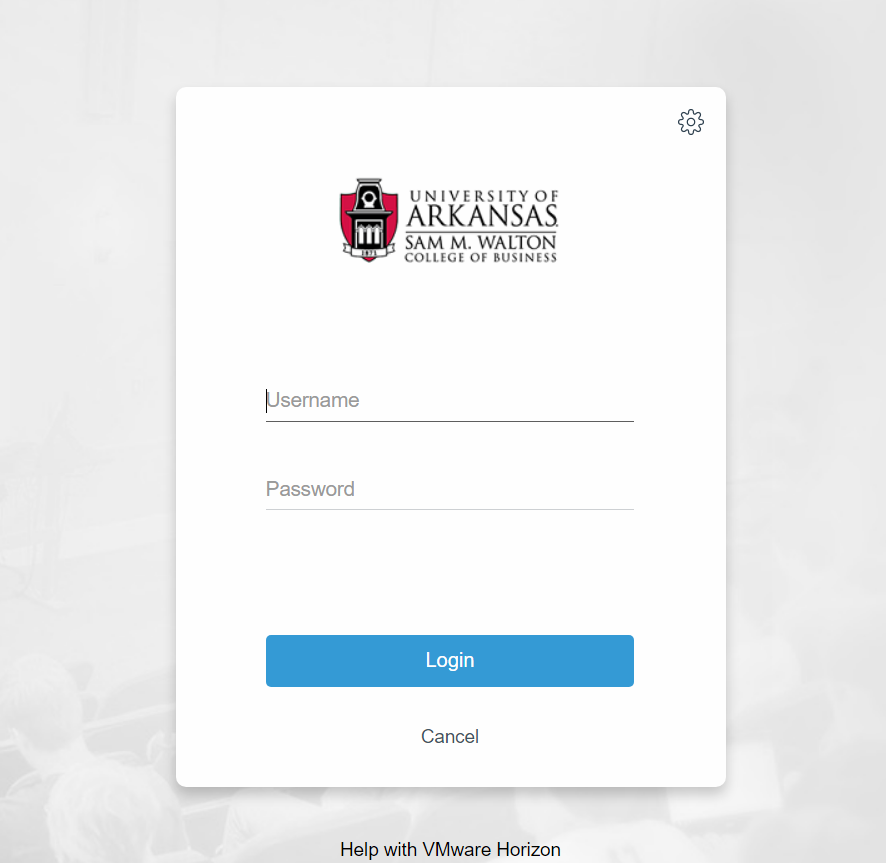
STEP 1: Google Chrome

* Launch Google Chrome by pressing the Windows Key and typing “google chrome”
* Once the Application pops up you can press it and launch Google Chrome

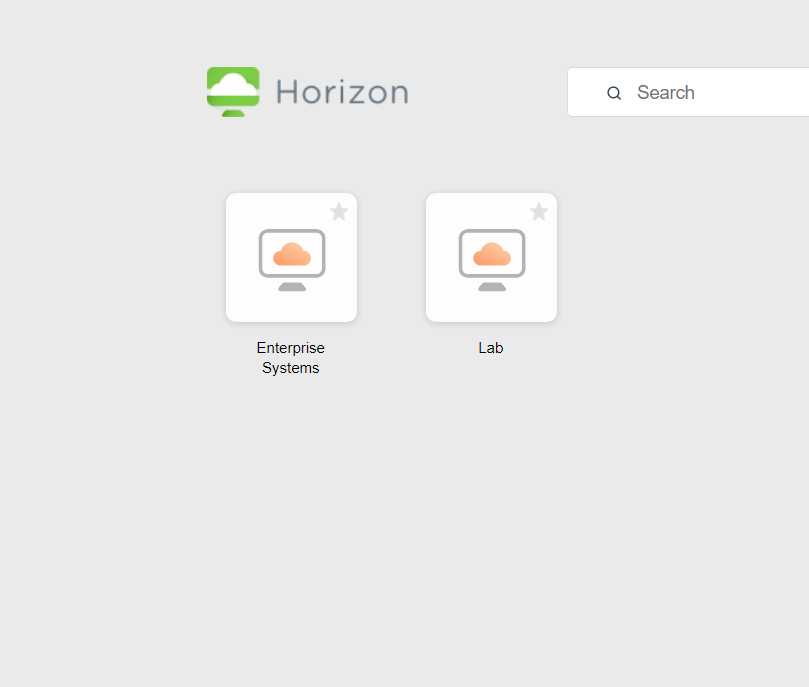


STEP 2: Walton Labs

* Type in the Google Chrome browser search bar <https://waltonlab.uark.edu/>
* If there is an option for “Access Virtual Desktop With Web Browser,” make sure you press that
* Press Accept
* When prompted, enter your given username and password

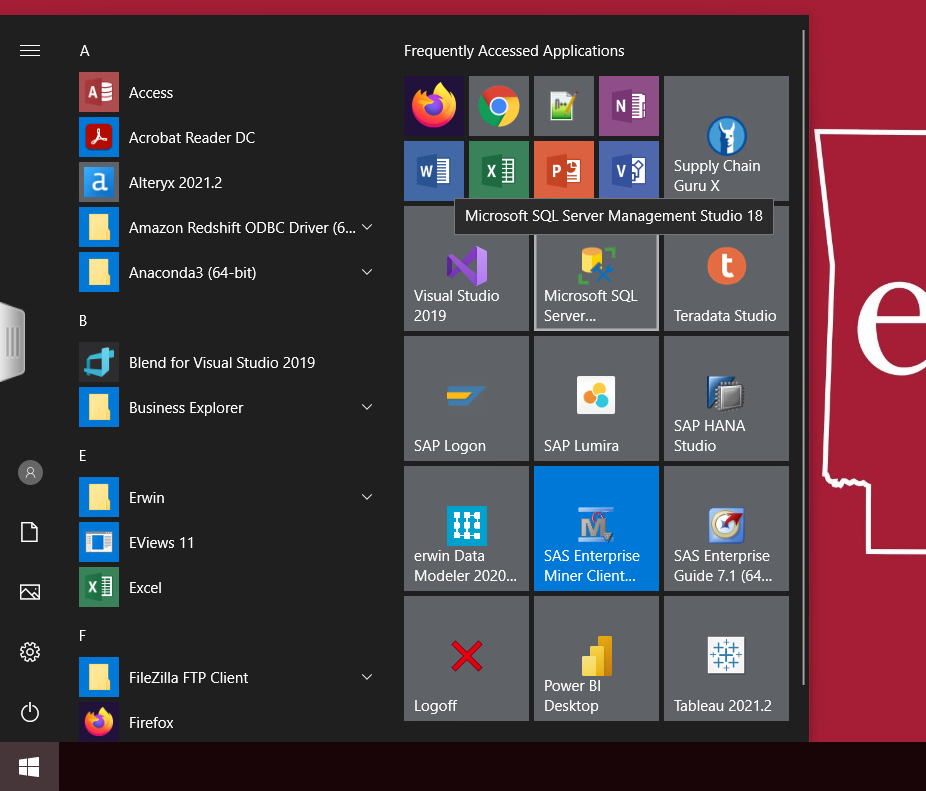


* press icon for Enterprise Systems

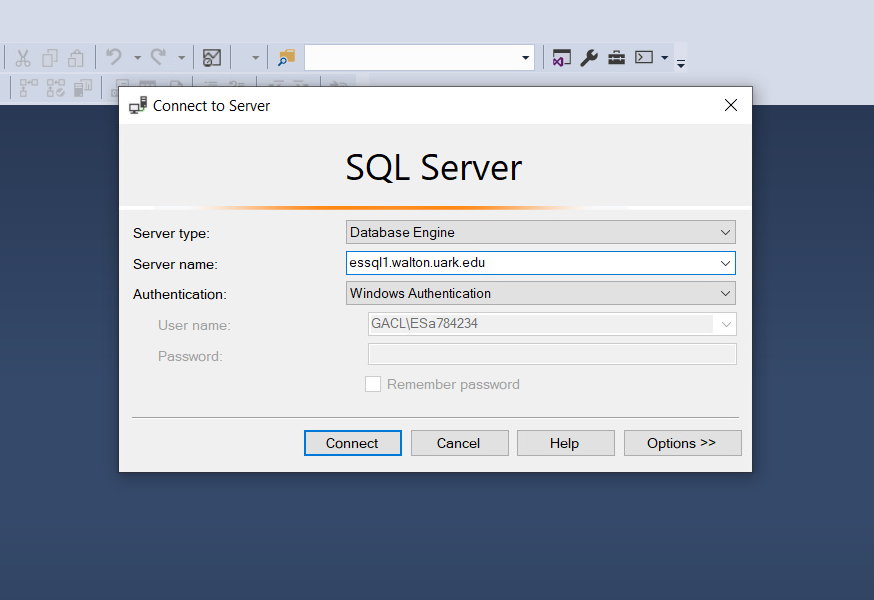


STEP 3: Virtual Desktop

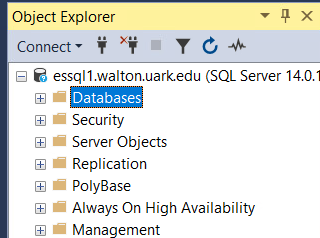
* Hit Windows Key on virtual desktop, then press icon for Microsoft SQL Server Management Studio 18 (if no icon then search in start menu)



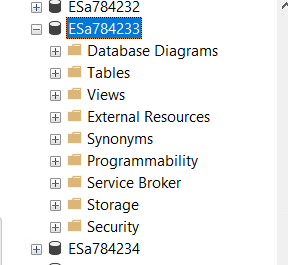
* Type “essql1.walton.uark.edu” in the Server Name field and press Connect



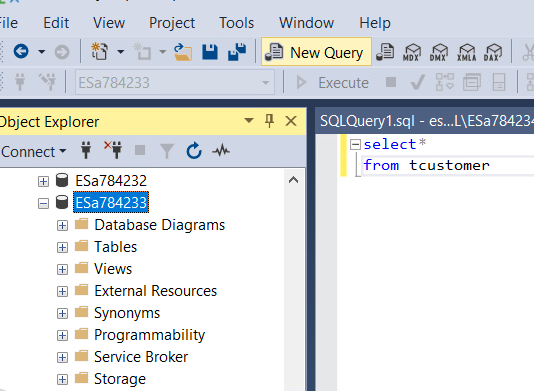
* Expand the Databases folder



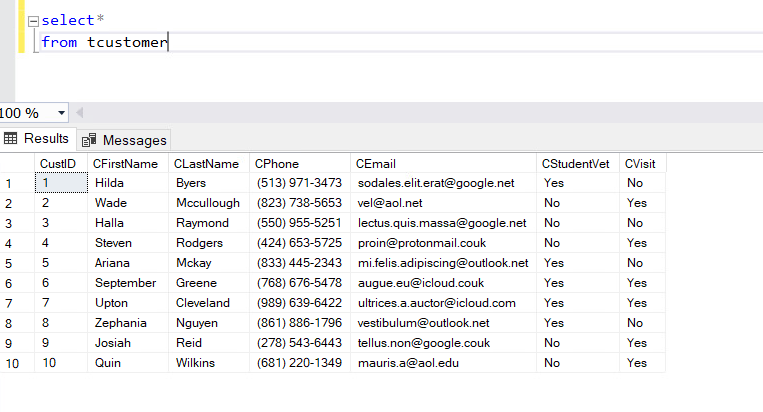
* Find the Username that corresponds with the desired database. Once you have located your database, then select it to open it.



* After selecting your desired database, press the New Query button to start a query.



* You can write a query like this one and select it and press execute to see the data. In the example, we have selected all the information contained into the TCustomer table



# **What We Learned Throughout This Process**

In this section, we will be outlining the group's overall accomplishments throughout implementing the database. As a group, our overall accomplishments have varied from person to person, but in the end, we have all accomplished a shared goal of conceptualizing, normalizing, and implementing a functional database.

As a group, collectively we have all learned how to implement and create a database. Towards the beginning of the project, we all learned how to create entities, and create a diagram in Lucid Charts. Our diagrams changed throughout the process as we learned how to normalize our information and incorporate it within our diagrams. After we learned how to normalize our information, our charts went from 2NF to 3NF. We then all learned how to access Walton.Uark.edu, and although it was a learning process, we all learned how to insert data into the database. In this database, we learned how to create different queries and how to execute them. From start to finish, we learned how to utilize the three different stages of the design process – Conceptual Design, Logical Design, and Physical Design. After completing the final steps, as a group we can all say that we understand how to create and implement a database.

As a team, we collaborated effectively all adhering to the contract created at the start of the project. Further, our communication was effective where we utilized various ways to communicate like Discord, GroupMe, and in-person communication. Utilization of these various communication methods allowed us to work together in many ways and work efficiently. All group members were present at all meetings, were up to date, and collaborated effectively on the creation of the database. Wolf Gang worked efficiently and learned along the way; our client will be very satisfied with the exceptional work we have provided.

|  |  |
| --- | --- |
| **Member Name:** | **What you learned:** |
| Eric Anderson | Through the different milestones of the project, I have gained proficiency in skills such as teamwork, time management, and focusing on small tasks. In the conceptual stage, I learned a lot about how to think about a real-life scenario in terms of entities and their relationships. In the logical stage, I learned how to normalize tables and attributes to be able to plan for a better database. In the final stage, I learned how the normalized ERD would look in terms of the physical data that we needed to add and query for. I gained a better understanding of the development process, as well as how to work in a more professional group setting. |
| Alec Amend | While working in my group for this project, I have learned a lot about working in a team regarding coding tools and database implementation. I realized that I am not the best coder nor the best at creating databases. However, I learned that you don’t have to be the best at successfully completing the tasks but being a good teammate and willing to do anything to help complete the project prevails. Additionally, I learned a lot of new skills like creating Entity Relationship Diagrams and creating Normalized Relations and helping the group in researching and writing the paragraphs needed within this document. The challenges I had within the project were not being as proficient in all the tools necessary for the project and using Walton as I did not fully understand it in class nor did I in the project. |
| Imran Ahmed | Throughout the journey of creating this database I learned how to conceptually design a database by creating an entity relationship diagram. Furthermore, I learned how to ensure its integrity and eliminate any data redundancy by normalizing the relations in the logical design. I then learned how to implement the database and insert data into the database during the physical design process. I gained a better understanding of how databases work and the steps to go through when creating one. Furthermore, I learned how to communicate and work within a group. |
| Briana Wells | Throughout the project, I have learned how to completely implement a database, and design it from start to finish. This was a challenge going in because there were topics that I was not familiar with yet, and it was a learning process throughout the semester. When it came to creating ERD’s and implementing the database in Walton.Uark.edu, these were some of the biggest struggles I had, because specifically because the Walton database in my opinion was not very user friendly, and if you made a mistake, it was very difficult to correct your mistake. I also had lots of connection issues with the database. |
| Matthew Brinkmeyer | I quite enjoyed this project. I think it was pretty fun to create a database. Or at least I did until milestone 3. I never realized how hard it is to actually create one. There are a lot of things that go into all of this and even when you think that you have finished there is always something wrong that throws a wrench into the project. I have a lot of respect for all the people who create and maintain databases now. I think having a team for this project made it a lot easier because I don’t think that I could’ve finished this project by myself without going insane. I even went a little crazy doing this even though we had a full team. But I really did enjoy this thing because it really is hands on with databases and I like that we had the change to mess around with it. I kind of wish that we had started this earlier on and the final submission could be our final since we worked on it all semester. |

# 

# **Appendix**

## **Team Contract**

Text

Description automatically generatedTable

Description automatically generated

Table, timeline

Description automatically generated

## **Data Dictionary Model**

**Table

Description automatically generated**

**A picture containing text, building

Description automatically generated**

## **Project Management**

Calendar

Description automatically generated with low confidence