

University of Oklahoma

Assistance League of Norman

Dunder Mifflin

April 26, 2020

Swetha Siripurapu

Dunder Mifflin

Elise Nill, Liyuan Wang, Sidiki Ganame, Uyen Tran

MIS 3353 -- Database Management

Swetha Siripurapu

April 26, 2020

Team Motto:

No shortcuts. You must work hard for what you want.



Executive Summary

Every year, many children go without proper clothing to sustain them through the school year. Assistance League is a non-profit organization that gives these children the access to proper clothing and hygiene products they require. This process is called Operation School Bell, and it is impacting nearly 1600 kids from the Norman and Little Axe school districts.

Due to the high volume of data, the Assistance League has asked for us to figure out a way to manage the data. With this goal in mind, we decided to create an ERD to help organize the data. By creating an ERD, we can have a visual representation of how the data will interact with each other. We can also use the ERD to store data which can be processed, as well as keep track of which students are returning and how many students are being impacted by the OSB program.




The whole process started from a client meeting, we learned about customer needs. We determined our significant assumption and decided what kind of business cycle we need to use and composed our ERD through these. By using an ERD, we defined logical design and normalization, then created the list of normalized relations. After that, based on the data we have, we did physical design and implementation by data dictionary innovation and denormalization. After many query experiments and data dictionary modification, we determined that this database is completed and meets customer requirements.


Along the process, our team has spent times and effort to address our customer's needs. Each member has recorded their own time sheets and the combination of the project cost is estimated \$3,440.42. Overall, this has been a great learning opportunity for our team members.

Contents

Executive Summary	3
Get to Know the Team: (Team Name)	5
Conceptual Design	7
The Client Meeting	7
Q&A During the Meeting & Information We Learned	7
Here you will list each question individually, followed by the answer you received during the interview.	7
Significant Assumptions	8
What is an ERD? Why is it necessary?	9
Business Cycles Used	9
ERD Created	9
Logical Design	13
What is Normalization? What purpose does it serve?	13
Define normalization and the importance of including it in this project. Be specific.	13
Normalized Relations	13
What differences are there between the ERD and the normalized relations? How do these differences help in moving from conceptual design to implementation?	18
What is a Referential Integrity Constraint? Why are they necessary?	18
Physical Design and Implementation	18
What is a Data Dictionary?	18
Denormalization	19
Implemented Physical Design	20
Please include a screenshot of your implemented physical ERD.	20
Challenges Faced/Addressed During Implementation	21
Strengths and Weaknesses Encountered During Implementation	21
Specific SQL Statements Requested	22
Three Additional Queries	27
User Documentation	29
What We Learned Throughout This Process	38
Appendix	39
Team Contract	39
Data Dictionary Model	40
Project Management	44

Get to Know the Team: Dunder Mifflin

Name	Year in School	Major/Minor	Internship Experience	Background Info	Contact phone no. & email
Elise Nill 	Junior	MIS	Internship at Cerner doing data analytics	From Kansas City, Missouri, 21 years old	816-589-7805 elise.c.nill-1@ou.edu
Liyuan Wang 	Senior	IB & MIS	Boli Industry: Account Assistant	Chinese, cat lover	405-534-6038 lyw@ou.edu
Uyen Tran 	Junior	MIS	No Internship experience	VietNameese, Transfer student	405-600-8173 uyen.m.tran-1@ou.edu

Sidiki Ganame 	Junior	Economic & MIS	No Internship experience	International student, from Burkina Faso, Davis UWC Scholar	405-314-5811 Sidiki.ganame-1@ou.edu
--	--------	----------------	--------------------------	---	--

Conceptual Design

The Client Meeting

Our team met with Marty Giffin, the chairperson of the Assistance League of Norman, on March 6, 2020. Our team consists of four members: Elise Nill, Liyuan Wang, Sidiki Ganame, and Uyen Tran. The location of our client meeting was Price Hall 3046. During the client meeting, our group asked Mr. Giffin specific questions to clarify our areas of confusion.

- Meeting Time: March 6, 2020
- Location: Price Hall 3046
- Interviewers: Dunder Mifflin Group
- Interviewee: Marty Giffin

Q&A During the Meeting & Information We Learned

Q: Do you care who is specifically referring the children to the organization? Or do you just want to know the positions of the referrer from schools, like the counselors and teachers?

A: They do not care what the referrer's name is, they just would like the school that they are associated with.

Q: Are the grooming kits based on gender and age, or are they already composed?

A: The grooming kits are generic, not gender specific, can be creative with how you want to represent this though.

Q: Do we need to track color, design and brand?

A: It is not required because it changes season to season or year to year.

Q: What if a child comes in and doesn't like to dress as their gender? Do you want to track what gender they are and what gender they like to shop?

A: They don't have to dress based on gender, I would like to know what clothes they take and what size they have.

Q: Is there anything about the vendor that relates to incomplete orders?

A: This information is about whenever a situation like mismatch items happens, we can look back the information about vendors.

Q: For the "season" in the question, does it mean a season or a full semester?

A: The season refers to the school year that the child is visiting during, including the summer after.

Q: There are so many schools, are you looking for a specific school?

A: Probably the same item would be sold by different vendors, what I care is who did I end up buying.

Q: Is the most purchased vendor based on quantity? or is it based on total spent? or how many purchases they made?

A: They want to track the amount of orders that are placed with individual vendors.

Q: In question four, does the grade mean which grade in school they are in, like sixth grade?

A: Yes, it means school grade, not “ABC” grade.

Q: In question eleven, what specific information do you want to know about vendors?

A: The name of vendors and we would really like to have vendor contacts

Q: Before going to Las Vegas, do we get a list of stuff?

A: Yes, there is a list and they track all the information. The nonprofit organization is not trying to make money. Kids don’t pay for the services.

Q: If they want to track which volunteers are most valuable, is it based on how much time total or how many kids they have helped?

A: Time, the policy is everyone can service one kid at a time. So, it is about how much time the volunteer willing to spend with us

Q: The books that they’re giving to kids, do kids care what genre the books are?

A: The genre of books doesn’t have to be specific

Significant Assumptions

In order for our team to create our entity-relationship diagram, there were a few assumptions we had to make along the way. For some, we were able to get the clarification we needed in our client meeting that allowed us to use these assumptions. However, for others, we had to make the assumptions ourselves, in order for our diagram to be logical. Through further discussion with the Assistance League of Norman, we hope to be able to get further information to support these assumptions.

1. Assumption that all children entered into the database are eligible.
 - a. This assumption affected our ERD because we did not have to check whether the child is eligible or not first, which means we don’t have to create another entity.
2. Assumption that the organization always has enough raw materials on hand for the kits.
 - a. This assumption affected our ERD because we did not have to worry about the relationship between the vendor and the raw material entities. We do not have to worry about ordering and keeping track of these quantities.

3. Assumption that the children entered into the database completed an order.
 - a. This assumption affected our ERD because we do not enter in the child's name and information just because they are referred to the organization. In order to be entered into the database they must have actually bought something.
4. Assumption that each gender that an individual child identifies with remains constant throughout their visits.
 - a. This assumption affected our ERD because we did not need to account for a child's gender changing which would affect the count of how many children of each gender preferred a certain type of clothing.
5. Assumption that the kits are specific to gender.
 - a. This assumption affected our ERD because we will track the quantities of each gender for our supply. The boy kit will have items specific to boys and the girl kit will have items specific to girls.

What is an ERD? Why is it necessary?

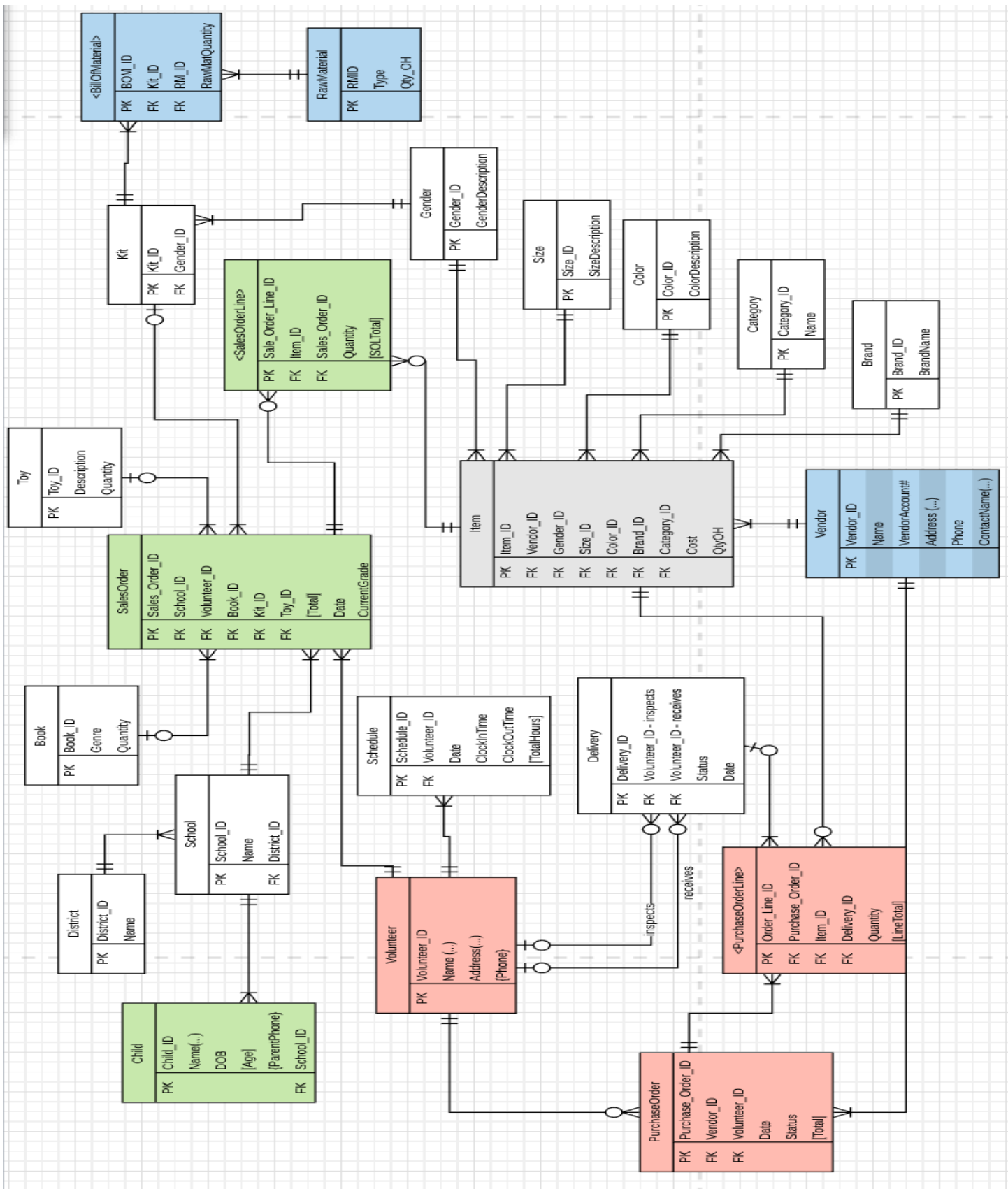
An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure. In Operational School Bell, we used an ERD to record all the information regarding the items selected and store it for later processing. The child can at this time pick a toy from donated items. Also, we can keep track of how many children return each year and benefit from the OSB program, student information and inventory.

Business Cycles Used

When deciding on the Business Cycle, we chose the Revenue Cycle, the Expenditure Cycle, and the Production Cycle. The details provided by our clients show that the OSB program will be serving children who can be considered as our customers and though the OSB program is a non-profit organization, meaning that they do not make any revenues from their activities, however, we needed a Revenue Cycle to keep track of the number of children served, the items distributed. We have included the Expenditure Cycle to track information of purchase. Though the organization does not manufacture any product, the Cycle will enable the organization to keep track of items assembled in a kit.

ERD Created

After carefully reading and understanding the wants and needs of our client, we were able to begin creating our entity-relationship diagram. Creating this diagram took a considerable amount of time in order for our team to ensure that all the expectations were being met. With the help of our client meeting, we were able to further understand what exactly the Assistance League of Norman wanted out of our product. Below, is a model of our entity-relationship diagram.

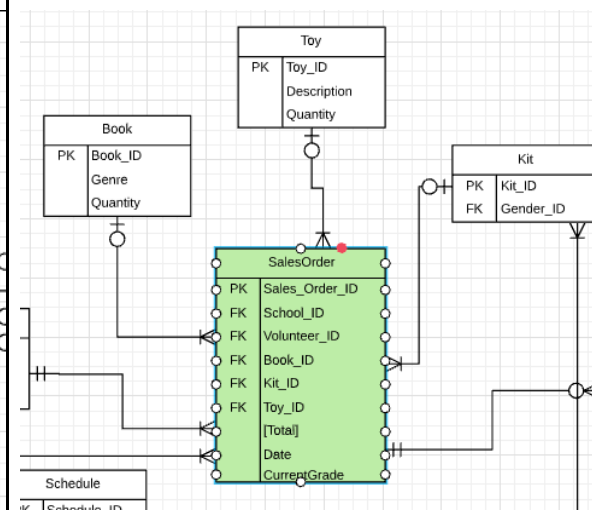
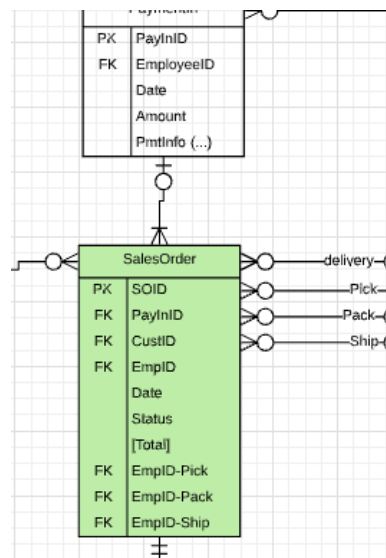


Changes made to generic ERDs

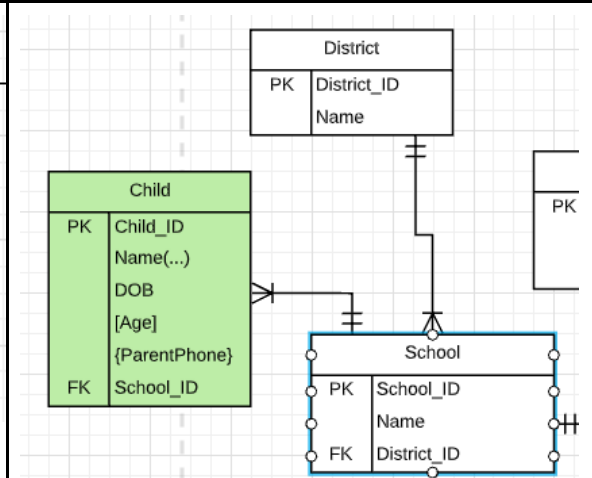
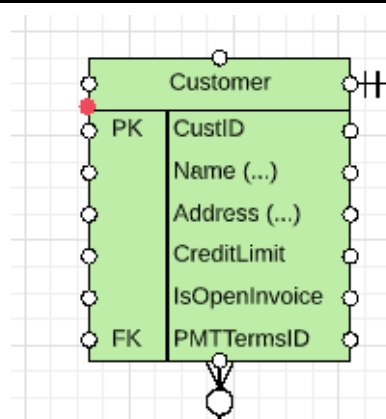
In order to accommodate our client's needs, certain changes were made to the basic expenditure cycle. The expenditure, revenue, and production cycles were used only as a guide for our ERD for the Assistance League of Norman, but as we continued, we added and took away certain aspects of the cycle. While we did make some changes, the overall structure and idea remains constant.

Change #	Original ERD	Updated ERD
From the generic cycle there is usually nothing branching off the product entity, but in our updated version we add a bunch of reference tables connected to the Item table. This will help us to more efficiently track these items.		
From the generic cycle, there is usually an employee entity, but we don't have employees, so we changed it to volunteer. This volunteer will receive and inspect the packages. Also, they will have a schedule entity that tracks how much they work, allowing us to reward the most valuable volunteer.		

In the generic revenue cycle, there are things like payment and delivery off the entity SalesOrder. In our ERD we removed the payment entity because they are not paying for the items and we added the entities book, toy and kit. These are reference tables for these objects.



We divided the Customer entity into few parts which contains the child entity, district entity and school entity. For customer entities, there will be more specific information needed in the future.



Logical Design

Logical design is the process of deciding how to arrange the attributes of the entities in a given business environment into database structures, such as the tables of relational databases. The goal of logical design is to create well-structured tables that properly reflect the company's business environment. Logical design is applicable to our project because we are making a new construction which will break down the logical relationship, enabling our client to understand it. Through the logical design process, we will be applying the five properties of relational models to our ERD. Those properties are as follows: unique name for each relation, atomic attributes, unique name for each non-key column, no derived attributes, insignificance for column sequence.

Normalization

Normalization is a database design and remodel technique that helps organize the tables within a database in a way that reduces data redundancy and increases the dependency of the data. This process takes larger tables within the database and divides them into smaller ones and then linking them with relationships. There are four main different types of "normal forms". The first is 0NF, which means that all the columns are not atomic. The second is 1NF, which means the columns are atomic and there is a partial functional dependency. The third is 2NF, which means that the columns are atomic, no partial functional dependencies, but has at least one transitive dependency. Finally, there is 3NF, which means all the columns are atomic, no partial functional dependencies, and no transitive dependencies. The normal form 3NF is the ideal form you want your database in. A partial functional dependency happens when a non-key attribute is predicted by less than the "full" primary key. A transitive dependency happens when a non-key value in a relation predicts another non-key value in that same relation. It is important to normalize databases to eliminate useless repeated data and to ensure that the data is stored logically. Within this project for our client, Operation School Bell, it is also important that we utilize this technique. We will need to use this process in areas of our database like the "Clothing" table since there are many different types of clothing that the children can receive including coats, pants, shirts, shoes, etc. Another area we will utilize normalization is for any table that includes a "Name" attribute. We will need to normalize this in order to perform certain tasks that before would not have been accurate.

Normalized Relations

TDistrict(District_ID, DistName)

TSchool(School_ID, SchoolName, SchoolDistrict_ID)

Foreign Key SchoolDistrict_ID references TDistrict

Not Null

On delete Restrict

TChild(Child_ID, ChildDOB, ChildAge, ChildFName, ChildName, ChildParentHomePhone, ChildParentCellPhone, ChildSchool_ID)

Foreign Key ChildSchool_ID references TSchool

Not Null

On delete Restrict

TVolunteer(Volunteer_ID, VolFirstName, VolLastName, VolStreet, VolCity, VolState, VolZipCode, VolCellPhone, VolHomePhone)

TSchedule(Schedule_ID, ScheVolunteer_ID, ScheDate, ScheClockInTime, ScheClockOutTime, ScheTotalHours)

Foreign Key ScheVolunteer_ID references TVolunteer

Not Null

On delete Restrict

TBook(Book_ID, BookGenre, BookQuantity)

TToy(Toy_ID, ToyDescription, ToyQuantity)

TGender(Gender_ID, GenderDescription)

TKit(Kit_ID, KitGender_ID)

Foreign Key KitGender_ID references TGender

Not Null

On delete restrict

TSalesOrder(SalesOrder_ID, SOVolunteer_ID, SOSchool_ID, SOBook_ID, SOKit_ID, SOToy_ID, SOTotal, SODate, SOCurrentGrade)

Foreign Key SOSchool_ID references TSchool

Not Null

On delete Restrict

Foreign Key SOVolunteer_ID references TVolunteer

Not Null

On delete Restrict

Foreign Key SOBook_ID references TBook

Null Allowed

On delete Set Null

Foreign Key SOKit_ID references TKit

Null Allowed

On delete Set Null

Foreign Key SOToy_ID references TToy

Null Allowed

On delete Set Null

TDelivery(DeliveryID, DelVolunteer_ID - inspects, DelVolunteer_ID - receives, DelStatus, DelDate)

Foreign Key DelVolunteer_ID-inspects references TVolunteer

Null Allowed

On delete Set Null

Foreign Key DelVolunteer_ID-receives references TVolunteer

Null Allowed

On delete Set Null

TVendor (Vendor_ID, VName, VVendorAccount#, VStreet, VCity, VState, VZipCode, VPhone, VContactFirstName, VContactLastName)

TSize(Size_ID, SizeSizeDescription)

TColor(Color_ID, ColorColorDescription)

TCategory(Category_ID, CategoryName)

TBrand(Brand_ID, BrandBrandName)

TItem(Item_ID, ItemVendor_ID, ItemGender_ID, ItemSize_ID, ItemColor_ID, ItemBrand_ID, ItemCategory_ID, ItemCost, ItemQtyOH)

Foreign Key ItemVendor_ID references TVendor

Not Null

On delete Restrict

Foreign Key ItemGender_ID references TGender

Not Null

On delete Restrict

Foreign Key ItemSize_ID references TSize

Not Null

On delete Restrict

Foreign Key ItemColor_ID references TColor

Not Null

On delete Restrict

Foreign Key ItemBrand_ID references TBrand

Not Null

On delete Restrict

Foreign Key ItemCategory_ID references TCategory

Not Null

On delete Restrict

TSalesOrderLine(Sale_Order_Line_ID, SOLItem_ID, SOLSales_Order_ID, SOLQuantity, SOLTotal)

Foreign Key SOLItem_ID references TItem

Not Null

On delete Restrict

Foreign Key SOLSales_Order_ID references TSalesOrder

Not Null

On delete Restrict

TPurchaseOrder(Purchase_Order_ID, POVendor_ID, POVolunteer_ID, PODate, POStatus, POTotal)

Foreign Key POVendor_ID references TVendor

Not Null

On Delete Restrict

Foreign Key POVolunteer_ID references TVolunteer

Not Null

On delete restrict

TPurchaseOrderLine(Order_Line_ID, POLPurchase_Order_ID, POLItem_ID, POLDelivery_ID, POLQuantity, POLLineTotal)

Foreign Key POLPurchase_Order_ID references TPurchaseOrder

Not Null

Delete Restrict

Foreign Key POLItem_ID references TItem

Not Null

Delete Restrict

Foreign Key POLDelivery_ID Reference TDelivery

Null Allowed

On Delete Set Null

TRawMaterial(RM_ID, RMTType, RMQty_OH)

TBillOfMaterial(BOM_ID, BOMItem_ID, BOMKit_ID, BOMRM_ID, BOMRawMatQuantity)

Foreign Key BOMItem_ID references TItem

Not Null

On delete Restrict

Foreign Key BOMKit_ID references TKit

Not Null

On delete Restrict

Foreign Key BOMRM_ID references TRawMaterial

Not Null

On delete Restrict

Differences between ERD and Normalized Relations

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities have attributes that define its properties. Normalization is a process through which developers can help ensure that a database is trustworthy and that you can trust the reports generated through the database to reflect reality and efficiently. That it doesn't take too much time to receive information from the database). Normalization requires you to do two things: ensure atomicity of columns and eliminate data duplication issues. Normalization helps reduce redundancy and ensure well-structured relations by: Making all columns atomic, eliminating data duplication issues and making them well-structured.

Referential Integrity

There are three types of data integrity: domain integrity, referential integrity, entity integrity. Domain integrity stands for all the values in the same column must be from the same "domain". This integrity will split all the attributes of all component combinations and create an atomic attribute. Entity integrity means that the entity initially defines each row as the only row in the table. The rows cannot be the same. Therefore, a primary key can be defined and there would be a unique identifier that exists in the primary key field. Referential integrity represents the association of entities based on the relationship between entities. The foreign key has to match a valid primary key, otherwise it could be null as long as the mandatory is optional.

Physical Design and Implementation

Physical design of a database is about categorizing the data into tables and attributes and providing information on how they are related to each other. Physical design allows us to better integrate a database design, and the goal is to make sure that the database will work efficiently. The implementation of the database refers to the process of importing tables created through the physical design into the database, then creating diagrams and SQL queries. In physical design, implementation, and the completed database, certain terms become predominant: "entities" and "relations" are referred to as "tables"; "attributes" and "columns" are referred to as "fields", and "instances" and "rows" are referred to as "records".

Data Dictionary

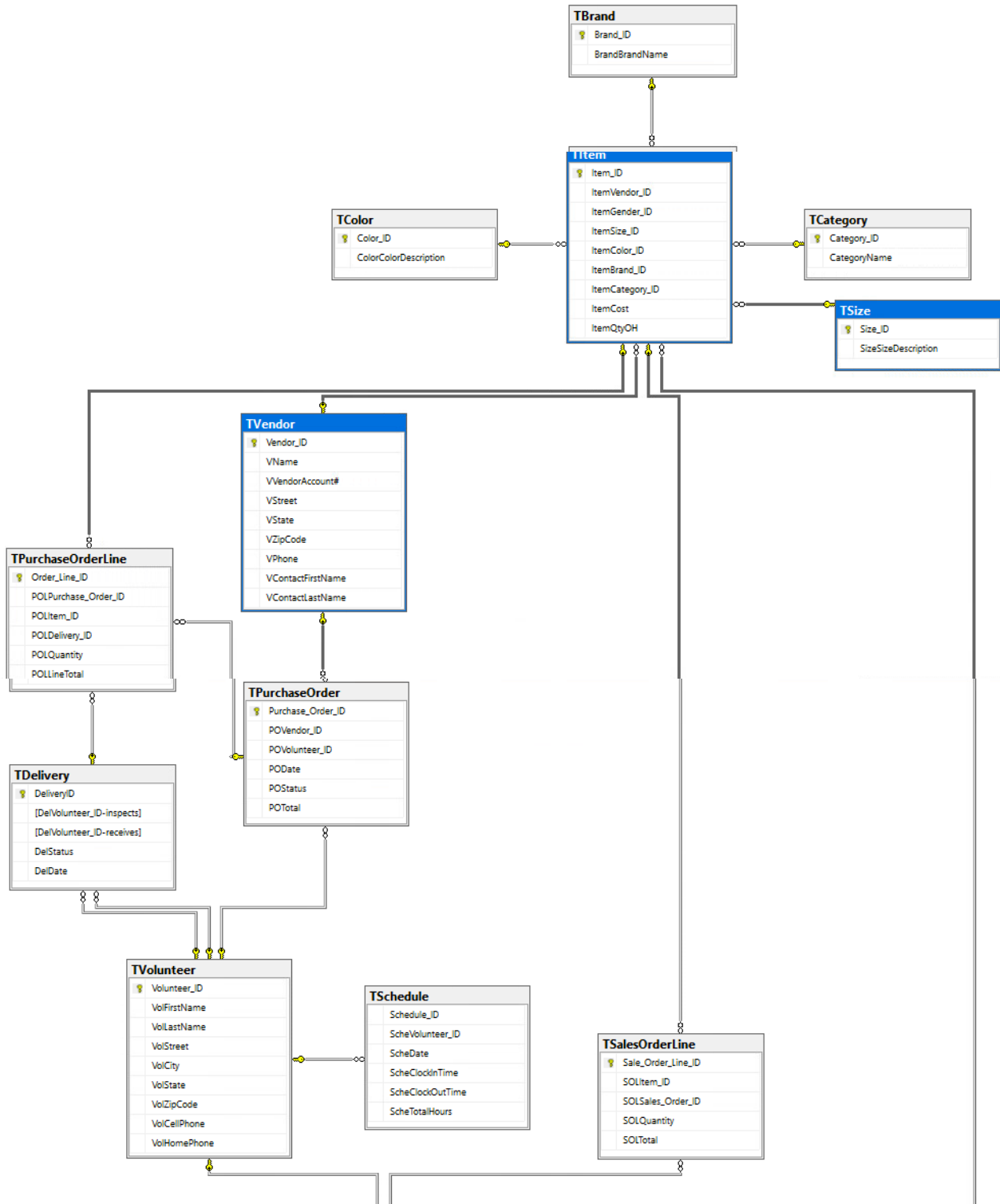
A data dictionary could be defined as a set of information that shows contents, format and structures of a database and the relationships between the entities. A data dictionary is important because it shows the output of the physical design process and reflects the decisions made about how the database is implemented while providing more information on how various fields are used. For our project, we have used: field names, key, the data type, requirement, default value and description. The table below is a sample of a data dictionary from our project.

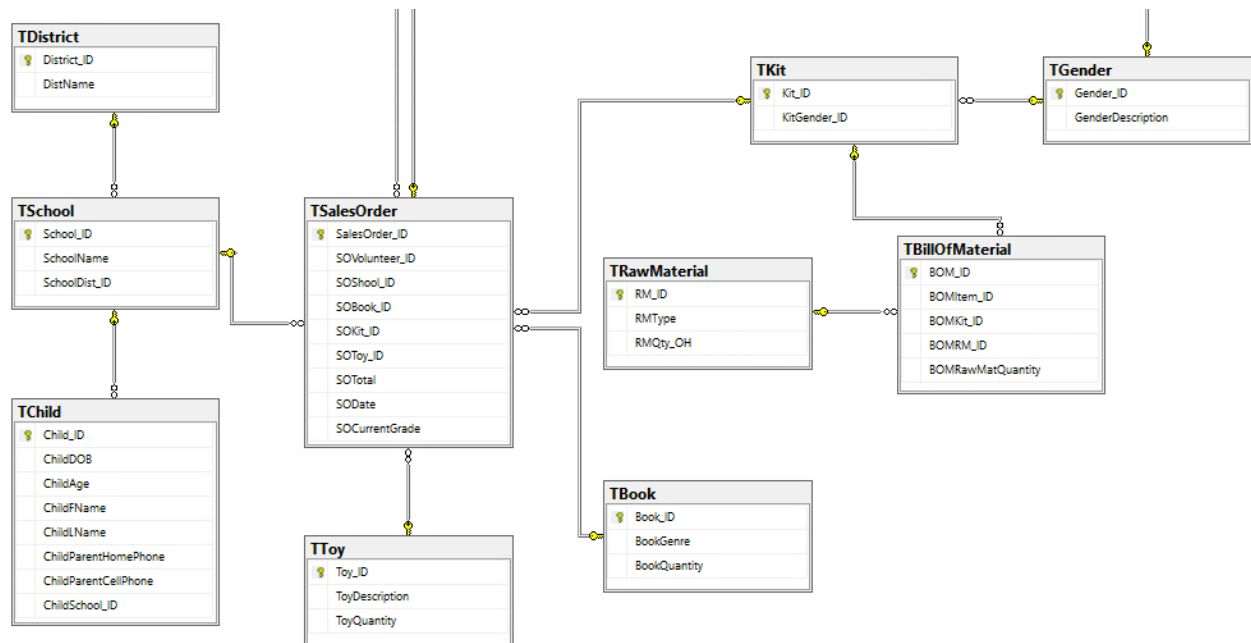
District					
Field Name	Key	Data Type	Reqd	Default Value	Description
District_ID	PK	AutoNumber	y		Random unique random
DistName		Text	y		name of the school district
School					
Field Name	Key	Data Type	Reqd	Default Value	Description
School_ID	PK	AutoNumber	y		Random unique number
SchoolName		Text	y		name of the school
SchoolDistrict_ID	FK references District	Number	y		
Child					
Field Name	Key	Data Type	Reqd	Default Value	Description
Child_ID	PK	AutoNumber	y		Random unique number
ChildDOB		Date			birth day of the child
ChildAge		Number			derive the age of the child
ChildFName		Name			first name of the child
ChildLName		Name			last name of the child
ChildParentHomePhone		Phone			child's parent's home phone number
ChildParentCellPhone		Phone			child's parent's cell phone number
ChildSchool_ID	FK references School	Number	y		

Denormalization

Denormalization is a strategy that database managers use to increase the performance of a database infrastructure. It involves adding redundant data to a normalized database to reduce certain types of problems with database queries that combine data from various tables into a single table. In our ERD, denormalization refers to the reduction of normalization of a table or tables within a database usually by allowing data duplication to occur. For example, composite attributes such as Name and Address have not been automated to their components in order to improve database efficiency. It also refers to reducing atomicity in a field or fields, but this is rarely done. Traditionally, denormalization has been a way to allow a database to process queries more quickly. Though, Denormalization has a few advantages such as the fewer the tables that need to be referenced, the faster the query, Denormalization introduce data duplication issues that could cause inefficiency of the database and occupy unnecessary storage.

Implemented Physical Design





Challenges Faced/Addressed During Implementation

One challenge we faced was that after creating the data dictionary, we had implemented nine tables and we lost everything. We had to backtrack and recreate them which wasted about five hours of work. By using our recorded video from the previous day while others continued with the other tables, we were able to recreate the lost table while still gaining ground. Another challenge was that we encountered several errors when importing some of our tables into the database. Sometimes, the table was well imported, but the data shown on the database was not. To solve those challenges, we made sure to change the primary keys that have a combination of letters and numbers and then we copied the data of each file into a new file.

Strengths and Weaknesses Encountered During Implementation

One strength our group had was we were very familiar with the functions and formulas, so we did very well on the excel sheet portion. Also, when we imported the data on the database, most of them ran very perfectly. Another strength our group displayed was our proficiency with writing SQL code. Some of our weaknesses mainly included us having trouble with creating the primary keys. At first, we used a letter and number for the primary key, but it wouldn't run correctly. It took us a lot of time to change those primary keys, but we were able to complete this. Also, sometimes when we would import the table, there would be some extra fields that we did not put in the table. In order to fix this problem, we had to copy all the data on that table and paste on the different excel sheet. We were unsure as to why this was happening or why our solution worked, but it did.

Specific SQL Statements Requested

Below, the twelve specific reports Operation School Bell required us to perform are listed. The reports of these queries they have asked us to perform are imperative to their organization. After a long process of creating and editing our database, we were able to carry out these queries. As well as the questions, the SQL code we wrote with its output is also listed below.

Query #	Question	SQL	Partial Output																																																																						
1	How many children were helped by each volunteer during a season?	Select count Child ID as Number_of_Children, TV. Volunteer_ID, (TV.VolFirstName + + TV.VolLastName) as Volunteer_Name from TChild TC join TSchool TS on TC. ChildSchool_ID = TS.School_ID join TSalesorder TSO on TS.School_ID = TSO.soschool_ID Join TVolunteer TV on TSO.SOVolunteer_ID = TV. Volunteer_ID where SODate between '2019-1-1' and '2019-5-1' group by TV. Volunteer_ID, (TV.VolFirstName + ' ' + TV.VollastName)	<div>Results Messages</div> <table><thead><tr><th></th><th>Number_of_Children</th><th>Volunteer_ID</th><th>Volunteer_Name</th></tr></thead><tbody><tr><td>1</td><td>7</td><td>3037</td><td>Abbie Whaley</td></tr><tr><td>2</td><td>3</td><td>3018</td><td>Ardisj MacMech...</td></tr><tr><td>3</td><td>5</td><td>3074</td><td>Berri Colwill</td></tr><tr><td>4</td><td>7</td><td>3079</td><td>Claresta Anning</td></tr><tr><td>5</td><td>5</td><td>3093</td><td>Colet Guile</td></tr><tr><td>6</td><td>2</td><td>3097</td><td>Costa Longford</td></tr><tr><td>7</td><td>11</td><td>3034</td><td>Ebony Guerrieri</td></tr><tr><td>8</td><td>2</td><td>3025</td><td>Egan Carlo</td></tr><tr><td>9</td><td>8</td><td>3056</td><td>Farah Porkiss</td></tr><tr><td>10</td><td>4</td><td>3091</td><td>Heywood Baulch</td></tr><tr><td>11</td><td>2</td><td>3069</td><td>Hugo Roggers</td></tr><tr><td>12</td><td>2</td><td>3099</td><td>Joannes Carluc...</td></tr><tr><td>13</td><td>2</td><td>3065</td><td>Korrie Heiner</td></tr></tbody></table>		Number_of_Children	Volunteer_ID	Volunteer_Name	1	7	3037	Abbie Whaley	2	3	3018	Ardisj MacMech...	3	5	3074	Berri Colwill	4	7	3079	Claresta Anning	5	5	3093	Colet Guile	6	2	3097	Costa Longford	7	11	3034	Ebony Guerrieri	8	2	3025	Egan Carlo	9	8	3056	Farah Porkiss	10	4	3091	Heywood Baulch	11	2	3069	Hugo Roggers	12	2	3099	Joannes Carluc...	13	2	3065	Korrie Heiner														
	Number_of_Children	Volunteer_ID	Volunteer_Name																																																																						
1	7	3037	Abbie Whaley																																																																						
2	3	3018	Ardisj MacMech...																																																																						
3	5	3074	Berri Colwill																																																																						
4	7	3079	Claresta Anning																																																																						
5	5	3093	Colet Guile																																																																						
6	2	3097	Costa Longford																																																																						
7	11	3034	Ebony Guerrieri																																																																						
8	2	3025	Egan Carlo																																																																						
9	8	3056	Farah Porkiss																																																																						
10	4	3091	Heywood Baulch																																																																						
11	2	3069	Hugo Roggers																																																																						
12	2	3099	Joannes Carluc...																																																																						
13	2	3065	Korrie Heiner																																																																						
2	How many of each item were distributed to the children from different school districts?	Select count (Item ID) as Number_of_Items, CategoryName, SchoolDist_ID, DistName from TDistrict TD join TSchool TS on TD.District_ID = TS.SchoolDist_ID join TSalesOrder TSO on TS.School_ID = TSO.SOSchool_ID join TSalesOrderLine TSOL on TSO.SalesOrder_ID = TSOL.SOLSales_Order_ID join TItem TI on TSOL.SOLItem_ID = TI.Item_ID join TCategory TC on TI.ItemCategory_ID = TC.Category_ID group by	<div>Results Messages</div> <table><thead><tr><th></th><th>Number_of_Items</th><th>CategoryName</th><th>SchoolDist...</th><th>DistName</th></tr></thead><tbody><tr><td>1</td><td>10</td><td>coat</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>2</td><td>3</td><td>corduroys</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>3</td><td>8</td><td>jeans</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>4</td><td>7</td><td>khakis</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>5</td><td>2</td><td>long sleeve shirt</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>6</td><td>9</td><td>shoes</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>7</td><td>2</td><td>short sleeve s...</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>8</td><td>8</td><td>socks</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>9</td><td>5</td><td>t-shirt</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>10</td><td>10</td><td>underwear</td><td>8000</td><td>Little Axe Public Sch...</td></tr><tr><td>11</td><td>6</td><td>coat</td><td>8001</td><td>Norman School Distr...</td></tr><tr><td>12</td><td>8</td><td>corduroys</td><td>8001</td><td>Norman School Distr...</td></tr><tr><td>13</td><td>2</td><td>jeans</td><td>8001</td><td>Norman School Distr...</td></tr></tbody></table>		Number_of_Items	CategoryName	SchoolDist...	DistName	1	10	coat	8000	Little Axe Public Sch...	2	3	corduroys	8000	Little Axe Public Sch...	3	8	jeans	8000	Little Axe Public Sch...	4	7	khakis	8000	Little Axe Public Sch...	5	2	long sleeve shirt	8000	Little Axe Public Sch...	6	9	shoes	8000	Little Axe Public Sch...	7	2	short sleeve s...	8000	Little Axe Public Sch...	8	8	socks	8000	Little Axe Public Sch...	9	5	t-shirt	8000	Little Axe Public Sch...	10	10	underwear	8000	Little Axe Public Sch...	11	6	coat	8001	Norman School Distr...	12	8	corduroys	8001	Norman School Distr...	13	2	jeans	8001	Norman School Distr...
	Number_of_Items	CategoryName	SchoolDist...	DistName																																																																					
1	10	coat	8000	Little Axe Public Sch...																																																																					
2	3	corduroys	8000	Little Axe Public Sch...																																																																					
3	8	jeans	8000	Little Axe Public Sch...																																																																					
4	7	khakis	8000	Little Axe Public Sch...																																																																					
5	2	long sleeve shirt	8000	Little Axe Public Sch...																																																																					
6	9	shoes	8000	Little Axe Public Sch...																																																																					
7	2	short sleeve s...	8000	Little Axe Public Sch...																																																																					
8	8	socks	8000	Little Axe Public Sch...																																																																					
9	5	t-shirt	8000	Little Axe Public Sch...																																																																					
10	10	underwear	8000	Little Axe Public Sch...																																																																					
11	6	coat	8001	Norman School Distr...																																																																					
12	8	corduroys	8001	Norman School Distr...																																																																					
13	2	jeans	8001	Norman School Distr...																																																																					

		DistName, CategoryName, SchoolDist_ID																																																																																	
3	How many items of each size were purchased by children? (this will help with ordering for the next season)	Select count(distinct Item_ID) as Number_Of_Item, Size_ID, SizeSizeDescription from TSalesOrder TSO join TSalesOrderLine TSOL on TSO.SalesOrder_ID = TSOL.SOLSales_Order_ID join TItem TI on TSOL.SOLItem_ID = TI.Item_ID join TSize TS on TI.ItemSize_ID = TS.Size_ID group by Size_ID, SizeSizeDescription	<div><div>Results</div><div>Messages</div><table><tr><th></th><th>Number_Of_It...</th><th>Size_ID</th><th>SizeSizeDescription</th></tr><tr><td>1</td><td>6</td><td>9000</td><td>shoe size 1</td></tr><tr><td>2</td><td>5</td><td>9001</td><td>shoe size 2</td></tr><tr><td>3</td><td>2</td><td>9002</td><td>shoe size 3</td></tr><tr><td>4</td><td>4</td><td>9003</td><td>shoe size 4</td></tr><tr><td>5</td><td>3</td><td>9004</td><td>shoe size 5</td></tr><tr><td>6</td><td>7</td><td>9005</td><td>shoe size 6</td></tr><tr><td>7</td><td>6</td><td>9006</td><td>shoe size 7</td></tr><tr><td>8</td><td>9</td><td>9007</td><td>shoe size 8</td></tr><tr><td>9</td><td>5</td><td>9008</td><td>shoe size 9</td></tr><tr><td>10</td><td>4</td><td>9009</td><td>shoe size 10</td></tr><tr><td>11</td><td>3</td><td>9010</td><td>shoe size 11</td></tr><tr><td>12</td><td>5</td><td>9011</td><td>shoe size 12</td></tr><tr><td>13</td><td>1</td><td>9012</td><td>small</td></tr><tr><td>14</td><td>4</td><td>9013</td><td>medium</td></tr><tr><td>15</td><td>3</td><td>9014</td><td>large</td></tr></table></div>		Number_Of_It...	Size_ID	SizeSizeDescription	1	6	9000	shoe size 1	2	5	9001	shoe size 2	3	2	9002	shoe size 3	4	4	9003	shoe size 4	5	3	9004	shoe size 5	6	7	9005	shoe size 6	7	6	9006	shoe size 7	8	9	9007	shoe size 8	9	5	9008	shoe size 9	10	4	9009	shoe size 10	11	3	9010	shoe size 11	12	5	9011	shoe size 12	13	1	9012	small	14	4	9013	medium	15	3	9014	large																
	Number_Of_It...	Size_ID	SizeSizeDescription																																																																																
1	6	9000	shoe size 1																																																																																
2	5	9001	shoe size 2																																																																																
3	2	9002	shoe size 3																																																																																
4	4	9003	shoe size 4																																																																																
5	3	9004	shoe size 5																																																																																
6	7	9005	shoe size 6																																																																																
7	6	9006	shoe size 7																																																																																
8	9	9007	shoe size 8																																																																																
9	5	9008	shoe size 9																																																																																
10	4	9009	shoe size 10																																																																																
11	3	9010	shoe size 11																																																																																
12	5	9011	shoe size 12																																																																																
13	1	9012	small																																																																																
14	4	9013	medium																																																																																
15	3	9014	large																																																																																
4	How many children from each school were helped and what was their class grade? We want this information to be in one report.	Select count(distinct Child_ID) as Number_Of_Children, School_ID, SchoolName, TSO. SOCurrentGrade From TChild TC join TSchool TS on TC.ChildSchool_ID =TS.School_ID join TSalesOrder TSO on TS.School ID = TSO. SOShoool_ID Group by School_ID, SOCurrentGrade, SchoolName	<div><div>Results</div><div>Messages</div><table><tr><th></th><th>Number_Of_Childr...</th><th>School_...</th><th>SchoolName</th><th>SOCurrentGra...</th></tr><tr><td>1</td><td>4</td><td>1000</td><td>Little Axe Elementary</td><td>1</td></tr><tr><td>2</td><td>1</td><td>1004</td><td>Cleveland Elementary School</td><td>1</td></tr><tr><td>3</td><td>3</td><td>1005</td><td>Eisenhower Elementary School</td><td>1</td></tr><tr><td>4</td><td>1</td><td>1008</td><td>Kennedy Elementary School</td><td>1</td></tr><tr><td>5</td><td>4</td><td>1009</td><td>Lakeview Elementary School</td><td>1</td></tr><tr><td>6</td><td>5</td><td>1012</td><td>McKinley Elementary School</td><td>1</td></tr><tr><td>7</td><td>5</td><td>1013</td><td>Monroe Elementary School</td><td>1</td></tr><tr><td>8</td><td>3</td><td>1015</td><td>Roosevelt Elementary School</td><td>1</td></tr><tr><td>9</td><td>2</td><td>1016</td><td>Truman Elementary School (grades 3-5)</td><td>1</td></tr><tr><td>10</td><td>3</td><td>1017</td><td>Truman Primary Elementary School (preK-2nd gra...</td><td>1</td></tr><tr><td>11</td><td>4</td><td>1000</td><td>Little Axe Elementary</td><td>2</td></tr><tr><td>12</td><td>3</td><td>1005</td><td>Eisenhower Elementary School</td><td>2</td></tr><tr><td>13</td><td>2</td><td>1006</td><td>Jackson Elementary School</td><td>2</td></tr><tr><td>14</td><td>2</td><td>1007</td><td>Jefferson Elementary School</td><td>2</td></tr><tr><td>15</td><td>4</td><td>1009</td><td>Lakeview Elementary School</td><td>2</td></tr></table></div>		Number_Of_Childr...	School_...	SchoolName	SOCurrentGra...	1	4	1000	Little Axe Elementary	1	2	1	1004	Cleveland Elementary School	1	3	3	1005	Eisenhower Elementary School	1	4	1	1008	Kennedy Elementary School	1	5	4	1009	Lakeview Elementary School	1	6	5	1012	McKinley Elementary School	1	7	5	1013	Monroe Elementary School	1	8	3	1015	Roosevelt Elementary School	1	9	2	1016	Truman Elementary School (grades 3-5)	1	10	3	1017	Truman Primary Elementary School (preK-2nd gra...	1	11	4	1000	Little Axe Elementary	2	12	3	1005	Eisenhower Elementary School	2	13	2	1006	Jackson Elementary School	2	14	2	1007	Jefferson Elementary School	2	15	4	1009	Lakeview Elementary School	2
	Number_Of_Childr...	School_...	SchoolName	SOCurrentGra...																																																																															
1	4	1000	Little Axe Elementary	1																																																																															
2	1	1004	Cleveland Elementary School	1																																																																															
3	3	1005	Eisenhower Elementary School	1																																																																															
4	1	1008	Kennedy Elementary School	1																																																																															
5	4	1009	Lakeview Elementary School	1																																																																															
6	5	1012	McKinley Elementary School	1																																																																															
7	5	1013	Monroe Elementary School	1																																																																															
8	3	1015	Roosevelt Elementary School	1																																																																															
9	2	1016	Truman Elementary School (grades 3-5)	1																																																																															
10	3	1017	Truman Primary Elementary School (preK-2nd gra...	1																																																																															
11	4	1000	Little Axe Elementary	2																																																																															
12	3	1005	Eisenhower Elementary School	2																																																																															
13	2	1006	Jackson Elementary School	2																																																																															
14	2	1007	Jefferson Elementary School	2																																																																															
15	4	1009	Lakeview Elementary School	2																																																																															
5	How many children from each school were helped and how many should we still expect to come?	Select count(distinct TC.Child_ID) as Number_of_Children_Visited, School_ID, SchoolName From TChild TC join TSchool TS on TC.ChildSchool_ID = TS.School_ID join TSalesOrder TSO on TS.School_ID = TSO.SOShoool_ID where YEAR(TSO. SODate) = 2019 group by School_ID, SchoolName	<div><div>Results</div><div>Messages</div><table><tr><th></th><th>Number_of_Children_Visit...</th><th>School_...</th><th>SchoolName</th></tr><tr><td>1</td><td>4</td><td>1000</td><td>Little Axe Elementary</td></tr><tr><td>2</td><td>10</td><td>1001</td><td>Little Axe Middle</td></tr><tr><td>3</td><td>3</td><td>1002</td><td>Little Axe High</td></tr><tr><td>4</td><td>1</td><td>1004</td><td>Cleveland Elementary School</td></tr><tr><td>5</td><td>3</td><td>1005</td><td>Eisenhower Elementary School</td></tr><tr><td>6</td><td>2</td><td>1006</td><td>Jackson Elementary School</td></tr><tr><td>7</td><td>2</td><td>1007</td><td>Jefferson Elementary School</td></tr><tr><td>8</td><td>1</td><td>1008</td><td>Kennedy Elementary School</td></tr><tr><td>9</td><td>4</td><td>1009</td><td>Lakeview Elementary School</td></tr><tr><td>10</td><td>5</td><td>1010</td><td>Lincoln Elementary School</td></tr><tr><td>11</td><td>8</td><td>1011</td><td>Madison Elementary School</td></tr><tr><td>12</td><td>5</td><td>1012</td><td>McKinley Elementary School</td></tr><tr><td>13</td><td>5</td><td>1013</td><td>Monroe Elementary School</td></tr><tr><td>14</td><td>2</td><td>1016</td><td>Truman Elementary School (grades 3-5)</td></tr><tr><td>15</td><td>2</td><td>1017</td><td>Truman Primary Elementary School (preK-2nd gra...</td></tr></table></div>		Number_of_Children_Visit...	School_...	SchoolName	1	4	1000	Little Axe Elementary	2	10	1001	Little Axe Middle	3	3	1002	Little Axe High	4	1	1004	Cleveland Elementary School	5	3	1005	Eisenhower Elementary School	6	2	1006	Jackson Elementary School	7	2	1007	Jefferson Elementary School	8	1	1008	Kennedy Elementary School	9	4	1009	Lakeview Elementary School	10	5	1010	Lincoln Elementary School	11	8	1011	Madison Elementary School	12	5	1012	McKinley Elementary School	13	5	1013	Monroe Elementary School	14	2	1016	Truman Elementary School (grades 3-5)	15	2	1017	Truman Primary Elementary School (preK-2nd gra...																
	Number_of_Children_Visit...	School_...	SchoolName																																																																																
1	4	1000	Little Axe Elementary																																																																																
2	10	1001	Little Axe Middle																																																																																
3	3	1002	Little Axe High																																																																																
4	1	1004	Cleveland Elementary School																																																																																
5	3	1005	Eisenhower Elementary School																																																																																
6	2	1006	Jackson Elementary School																																																																																
7	2	1007	Jefferson Elementary School																																																																																
8	1	1008	Kennedy Elementary School																																																																																
9	4	1009	Lakeview Elementary School																																																																																
10	5	1010	Lincoln Elementary School																																																																																
11	8	1011	Madison Elementary School																																																																																
12	5	1012	McKinley Elementary School																																																																																
13	5	1013	Monroe Elementary School																																																																																
14	2	1016	Truman Elementary School (grades 3-5)																																																																																
15	2	1017	Truman Primary Elementary School (preK-2nd gra...																																																																																

6	An itemized invoice for each purchase.	Select SalesOrder_ID, SOTotal, SODate, SOLITEM_ID, SOLQuantity, SOLTotal From TSalesOrder TSO join TSalesOrderLine TSOL on TSO.SalesOrder_ID = TSOL.SOLSales_Order_ID join TItem TI on TSOL.SOLItem_ID = TI.Item_ID	<div><div>Results</div><div>Messages</div><table><tr><th></th><th>SalesOrder_ID</th><th>SOTo...</th><th>SODate</th><th>SOLItem_ID</th><th>SOLQuant...</th><th>SOLTo...</th></tr><tr><td>1</td><td>10156</td><td>49</td><td>2019-05-...</td><td>389</td><td>7</td><td>77</td></tr><tr><td>2</td><td>10131</td><td>63</td><td>2019-03-...</td><td>308</td><td>10</td><td>86</td></tr><tr><td>3</td><td>10195</td><td>151</td><td>2019-11-...</td><td>319</td><td>4</td><td>64</td></tr><tr><td>4</td><td>10119</td><td>156</td><td>2020-10-...</td><td>310</td><td>5</td><td>60</td></tr><tr><td>5</td><td>10112</td><td>81</td><td>2019-10-...</td><td>358</td><td>3</td><td>70</td></tr><tr><td>6</td><td>10102</td><td>187</td><td>2020-06-...</td><td>346</td><td>4</td><td>53</td></tr><tr><td>7</td><td>10189</td><td>81</td><td>2019-05-...</td><td>358</td><td>10</td><td>60</td></tr><tr><td>8</td><td>10109</td><td>85</td><td>2019-12-...</td><td>324</td><td>10</td><td>78</td></tr><tr><td>9</td><td>10102</td><td>187</td><td>2020-06-...</td><td>340</td><td>2</td><td>83</td></tr><tr><td>10</td><td>10175</td><td>179</td><td>2020-05-...</td><td>363</td><td>7</td><td>92</td></tr><tr><td>11</td><td>10130</td><td>120</td><td>2019-12-...</td><td>399</td><td>8</td><td>96</td></tr><tr><td>12</td><td>10138</td><td>68</td><td>2020-11-...</td><td>307</td><td>10</td><td>82</td></tr><tr><td>13</td><td>10137</td><td>77</td><td>2019-12-...</td><td>315</td><td>5</td><td>93</td></tr><tr><td>14</td><td>10164</td><td>22</td><td>2019-03-...</td><td>318</td><td>4</td><td>61</td></tr><tr><td>15</td><td>10110</td><td>134</td><td>2019-03-...</td><td>374</td><td>3</td><td>64</td></tr></table></div>		SalesOrder_ID	SOTo...	SODate	SOLItem_ID	SOLQuant...	SOLTo...	1	10156	49	2019-05-...	389	7	77	2	10131	63	2019-03-...	308	10	86	3	10195	151	2019-11-...	319	4	64	4	10119	156	2020-10-...	310	5	60	5	10112	81	2019-10-...	358	3	70	6	10102	187	2020-06-...	346	4	53	7	10189	81	2019-05-...	358	10	60	8	10109	85	2019-12-...	324	10	78	9	10102	187	2020-06-...	340	2	83	10	10175	179	2020-05-...	363	7	92	11	10130	120	2019-12-...	399	8	96	12	10138	68	2020-11-...	307	10	82	13	10137	77	2019-12-...	315	5	93	14	10164	22	2019-03-...	318	4	61	15	10110	134	2019-03-...	374	3	64
	SalesOrder_ID	SOTo...	SODate	SOLItem_ID	SOLQuant...	SOLTo...																																																																																																													
1	10156	49	2019-05-...	389	7	77																																																																																																													
2	10131	63	2019-03-...	308	10	86																																																																																																													
3	10195	151	2019-11-...	319	4	64																																																																																																													
4	10119	156	2020-10-...	310	5	60																																																																																																													
5	10112	81	2019-10-...	358	3	70																																																																																																													
6	10102	187	2020-06-...	346	4	53																																																																																																													
7	10189	81	2019-05-...	358	10	60																																																																																																													
8	10109	85	2019-12-...	324	10	78																																																																																																													
9	10102	187	2020-06-...	340	2	83																																																																																																													
10	10175	179	2020-05-...	363	7	92																																																																																																													
11	10130	120	2019-12-...	399	8	96																																																																																																													
12	10138	68	2020-11-...	307	10	82																																																																																																													
13	10137	77	2019-12-...	315	5	93																																																																																																													
14	10164	22	2019-03-...	318	4	61																																																																																																													
15	10110	134	2019-03-...	374	3	64																																																																																																													
7	Who were our suppliers in a particular year? Provide the total paid for each supplier and the number of POs.	Select TV.VName, sum(distinct POTotal) as Grand_Total, count(distinct Purchase_Order_ID) as Number_of_POS from TVendor TV join TPurchaseOrder TPO on TV.Vendor_ID = TPO.POVendor_ID join TPurchaseOrderLine TPOL on TPO.Purchase_Order_ID = TPOL.POLPurchase_Order_ID where TPO.PODate between '2019-1-1" and "2020-1-1' group by TV. VName	<div><div>Results</div><div>Messages</div><table><tr><th></th><th>VName</th><th>Grand_Total</th><th>Number_of_P...</th></tr><tr><td>1</td><td>IDACORP</td><td>9016</td><td>6</td></tr><tr><td>2</td><td>Idaho Community</td><td>20660</td><td>13</td></tr><tr><td>3</td><td>IPA</td><td>22153</td><td>14</td></tr><tr><td>4</td><td>Nagel</td><td>19260</td><td>12</td></tr><tr><td>5</td><td>Sunwest Bank</td><td>6966</td><td>5</td></tr><tr><td>6</td><td>The Julius C Jek...</td><td>14853</td><td>10</td></tr></table></div>		VName	Grand_Total	Number_of_P...	1	IDACORP	9016	6	2	Idaho Community	20660	13	3	IPA	22153	14	4	Nagel	19260	12	5	Sunwest Bank	6966	5	6	The Julius C Jek...	14853	10																																																																																				
	VName	Grand_Total	Number_of_P...																																																																																																																
1	IDACORP	9016	6																																																																																																																
2	Idaho Community	20660	13																																																																																																																
3	IPA	22153	14																																																																																																																
4	Nagel	19260	12																																																																																																																
5	Sunwest Bank	6966	5																																																																																																																
6	The Julius C Jek...	14853	10																																																																																																																
8	How many children were served for more than one season?	select COUNT (Child_ID) AS Number_Of Children from TChild TC join TSchool TS ON TC.ChildSchool_ID = TS.School_ID join TSalesorder TSO on TS.School_ID= TSO.SOSchool_ID HAVING COUNT (SODate)>1	<div><div>Results</div><div>Messages</div><table><tr><th></th><th>Number_Of_Childr...</th></tr><tr><td>1</td><td>397</td></tr></table></div>		Number_Of_Childr...	1	397																																																																																																												
	Number_Of_Childr...																																																																																																																		
1	397																																																																																																																		

9	What is the preferred color of girls and boys? How many girls and boys did we serve? It would be great if this can be done in 1 report/query.	select sum(SOLQuantity) as Number_of_Items, ColorColorDescription, GenderDescription from TSalesOrder TSO join TSalesOrderLine TSOL on TSO.SalesOrder_ID = TSOL.SOLSales_Order_ID join TItem TI on TSOL.SOLItem_ID = TI.Item_ID join TGender TG on TI.ItemGender_ID = TG.Gender_ID join TColor TC on TI.ItemColor_ID = TC.Color_ID group by ColorColorDescription, GenderDescription order by GenderDescription, Number_of_Items Desc	<div><div>Results</div><div>Messages</div></div> <table><thead><tr><th></th><th>Number_of_Items</th><th>ColorColorDescription</th><th>GenderDescription</th></tr></thead><tbody><tr><td>1</td><td>70</td><td>blue</td><td>Female</td></tr><tr><td>2</td><td>46</td><td>yellow</td><td>Female</td></tr><tr><td>3</td><td>34</td><td>white</td><td>Female</td></tr><tr><td>4</td><td>33</td><td>green</td><td>Female</td></tr><tr><td>5</td><td>28</td><td>pink</td><td>Female</td></tr><tr><td>6</td><td>27</td><td>orange</td><td>Female</td></tr><tr><td>7</td><td>20</td><td>black</td><td>Female</td></tr><tr><td>8</td><td>5</td><td>purple</td><td>Female</td></tr><tr><td>9</td><td>5</td><td>red</td><td>Female</td></tr><tr><td>10</td><td>1</td><td>grey</td><td>Female</td></tr><tr><td>11</td><td>57</td><td>white</td><td>Male</td></tr><tr><td>12</td><td>34</td><td>yellow</td><td>Male</td></tr></tbody></table>		Number_of_Items	ColorColorDescription	GenderDescription	1	70	blue	Female	2	46	yellow	Female	3	34	white	Female	4	33	green	Female	5	28	pink	Female	6	27	orange	Female	7	20	black	Female	8	5	purple	Female	9	5	red	Female	10	1	grey	Female	11	57	white	Male	12	34	yellow	Male
	Number_of_Items	ColorColorDescription	GenderDescription																																																				
1	70	blue	Female																																																				
2	46	yellow	Female																																																				
3	34	white	Female																																																				
4	33	green	Female																																																				
5	28	pink	Female																																																				
6	27	orange	Female																																																				
7	20	black	Female																																																				
8	5	purple	Female																																																				
9	5	red	Female																																																				
10	1	grey	Female																																																				
11	57	white	Male																																																				
12	34	yellow	Male																																																				
10	What is the most purchased vendor by Little Axe Elementary students?	Select TVe.Vendor_ID, VName, POTotal, count(distinct Purchase_Order_ID) as Number_of_Orders, SchoolName from TSchool TC join TSalesOrder TSO on TC.School_ID = TSO SOSchool_ID join TVolunteer TV on TSO. SOVolunteer ID - TV.Volunteer_ID join TPurchaseOrder TPO on TV.Volunteer_ID = TPO. POVolunteer_ID join TVendor TVe on TPO.POVendor ID = TVe. Vendor_ID where SchoolName like 'XLittle Axe Elementary%' Group by TVe.Vendor_ID, VName, POTotal, SchoolName order by POTotal desc	<div><div>Results</div><div>Messages</div></div> <table><thead><tr><th></th><th>Vendor_ID</th><th>VName</th><th>POTotal</th><th>Number_of_Orders</th><th>SchoolName</th></tr></thead><tbody><tr><td>1</td><td>99</td><td>Idaho Commun...</td><td>2000</td><td>1</td><td>Little Axe Element...</td></tr><tr><td>2</td><td>53</td><td>Nagel</td><td>1908</td><td>1</td><td>Little Axe Element...</td></tr><tr><td>3</td><td>99</td><td>Idaho Commun...</td><td>1761</td><td>1</td><td>Little Axe Element...</td></tr><tr><td>4</td><td>57</td><td>Sunwest Bank</td><td>1490</td><td>1</td><td>Little Axe Element...</td></tr><tr><td>5</td><td>16</td><td>Idaho Commun...</td><td>1459</td><td>1</td><td>Little Axe Element...</td></tr><tr><td>6</td><td>69</td><td>Sunwest Bank</td><td>1124</td><td>1</td><td>Little Axe Element...</td></tr></tbody></table>		Vendor_ID	VName	POTotal	Number_of_Orders	SchoolName	1	99	Idaho Commun...	2000	1	Little Axe Element...	2	53	Nagel	1908	1	Little Axe Element...	3	99	Idaho Commun...	1761	1	Little Axe Element...	4	57	Sunwest Bank	1490	1	Little Axe Element...	5	16	Idaho Commun...	1459	1	Little Axe Element...	6	69	Sunwest Bank	1124	1	Little Axe Element...										
	Vendor_ID	VName	POTotal	Number_of_Orders	SchoolName																																																		
1	99	Idaho Commun...	2000	1	Little Axe Element...																																																		
2	53	Nagel	1908	1	Little Axe Element...																																																		
3	99	Idaho Commun...	1761	1	Little Axe Element...																																																		
4	57	Sunwest Bank	1490	1	Little Axe Element...																																																		
5	16	Idaho Commun...	1459	1	Little Axe Element...																																																		
6	69	Sunwest Bank	1124	1	Little Axe Element...																																																		

11	For each PO that was placed in July of this year, show each line, including the PO number, name of vendor, name of product, item price, quantity, line total.	Select Purchase_Order_ID PurchaseOrderNumber, VName NameOfVendor, CategoryName NameOfProduct, ItemCost, POLQuantity, POLLineTotal from TPurchaseOrder TPO join TPurchaseOrderLine TPOL on TPO.Purchase_Order_ID = TPOL. POLPurchase_Order_ID join TVendor TV on TPO. POVendor_ID = TV. Vendor_ID join TItem TI on TV.Vendor_ID = TI.ItemVendor_ID join TCategory TC on TI.ItemCategory_ID = TC.Category_ID where month(PODate) = 5 and year(PODate) = 2019	<div>Results Messages</div> <table><thead><tr><th></th><th>PurchaseOrderNumber</th><th>NameOfVendor</th><th>NameOfProduct</th><th>ItemCost</th><th>POLQuant...</th><th>POLLineTo...</th></tr></thead><tbody><tr><td>1</td><td>20112</td><td>IPA</td><td>shoes</td><td>95</td><td>10</td><td>274</td></tr><tr><td>2</td><td>20112</td><td>IPA</td><td>shoes</td><td>95</td><td>5</td><td>576</td></tr><tr><td>3</td><td>20126</td><td>The Julius C Jek...</td><td>long sleeve shirt</td><td>29</td><td>8</td><td>25</td></tr><tr><td>4</td><td>20161</td><td>Sunwest Bank</td><td>long sleeve shirt</td><td>33</td><td>8</td><td>404</td></tr><tr><td>5</td><td>20161</td><td>Sunwest Bank</td><td>long sleeve shirt</td><td>33</td><td>5</td><td>473</td></tr><tr><td>6</td><td>20112</td><td>IPA</td><td>shoes</td><td>95</td><td>8</td><td>603</td></tr></tbody></table>		PurchaseOrderNumber	NameOfVendor	NameOfProduct	ItemCost	POLQuant...	POLLineTo...	1	20112	IPA	shoes	95	10	274	2	20112	IPA	shoes	95	5	576	3	20126	The Julius C Jek...	long sleeve shirt	29	8	25	4	20161	Sunwest Bank	long sleeve shirt	33	8	404	5	20161	Sunwest Bank	long sleeve shirt	33	5	473	6	20112	IPA	shoes	95	8	603
	PurchaseOrderNumber	NameOfVendor	NameOfProduct	ItemCost	POLQuant...	POLLineTo...																																														
1	20112	IPA	shoes	95	10	274																																														
2	20112	IPA	shoes	95	5	576																																														
3	20126	The Julius C Jek...	long sleeve shirt	29	8	25																																														
4	20161	Sunwest Bank	long sleeve shirt	33	8	404																																														
5	20161	Sunwest Bank	long sleeve shirt	33	5	473																																														
6	20112	IPA	shoes	95	8	603																																														
12	List all schools that did not send a child in a particular season? a child in a particular season.	select TS.School_ID, TS.SchoolName From TSchool TS where TS.School_ID not in (Select distinct School_ID From TSchool TS join TSalesOrder TSO on TS.School_ID = TSO.SOSchool_ID where SODate between '2019-1-1' and '2019-5-1').	<div>Results Messages</div> <table><thead><tr><th></th><th>School_ID</th><th>SchoolName</th></tr></thead><tbody><tr><td>1</td><td>1000</td><td>Little Axe Elementary</td></tr><tr><td>2</td><td>1001</td><td>Little Axe Middle</td></tr><tr><td>3</td><td>1003</td><td>Adams Elementary School</td></tr><tr><td>4</td><td>1005</td><td>Eisenhower Elementary School</td></tr><tr><td>5</td><td>1006</td><td>Jackson Elementary School</td></tr><tr><td>6</td><td>1008</td><td>Kennedy Elementary School</td></tr><tr><td>7</td><td>1014</td><td>Reagan Elementary School</td></tr><tr><td>8</td><td>1015</td><td>Roosevelt Elementary School</td></tr><tr><td>9</td><td>1017</td><td>Truman Primary Elementary School (preK-2nd gra...</td></tr><tr><td>10</td><td>1020</td><td>Alcott Middle School</td></tr><tr><td>11</td><td>1021</td><td>Irving Middle School</td></tr><tr><td>12</td><td>1022</td><td>Longfellow Middle School</td></tr></tbody></table>		School_ID	SchoolName	1	1000	Little Axe Elementary	2	1001	Little Axe Middle	3	1003	Adams Elementary School	4	1005	Eisenhower Elementary School	5	1006	Jackson Elementary School	6	1008	Kennedy Elementary School	7	1014	Reagan Elementary School	8	1015	Roosevelt Elementary School	9	1017	Truman Primary Elementary School (preK-2nd gra...	10	1020	Alcott Middle School	11	1021	Irving Middle School	12	1022	Longfellow Middle School										
	School_ID	SchoolName																																																		
1	1000	Little Axe Elementary																																																		
2	1001	Little Axe Middle																																																		
3	1003	Adams Elementary School																																																		
4	1005	Eisenhower Elementary School																																																		
5	1006	Jackson Elementary School																																																		
6	1008	Kennedy Elementary School																																																		
7	1014	Reagan Elementary School																																																		
8	1015	Roosevelt Elementary School																																																		
9	1017	Truman Primary Elementary School (preK-2nd gra...																																																		
10	1020	Alcott Middle School																																																		
11	1021	Irving Middle School																																																		
12	1022	Longfellow Middle School																																																		

Three Additional Queries

After completing the twelve queries requested by Operation School Bell, we thought of a few others that might be beneficial to their company. The following queries are ones that we thought could offer more information about their customers and his/her orders and information about each of the volunteers. Some of our ideas were prompted by the information packet we received at the beginning of this process. In this packet, Operation School Bell mentioned a few things that they were having trouble with or that they would potentially like to see a report of. Below are the three additional queries we created.

Query #	Question	Why is this important	SQL	Partial Output	Recap of Findings																																																				
1	List all volunteers and their volunteer time for a given.	This was one thing the company wanted improvements on. They want to reward the volunteer that volunteers the most time.	Select Volunteer_ID, (VolFirstName + ' ' + VolLastName) as Volunteer_Name, round (sum(ScheTotalHours),2) as Total_Hours From TVolunteer TV join TSchedule TS on TV.Volunteer_ID = TS.ScheVolunteer_ID where ScheDate between '2019-1-1' and '2019-5-1' Group by Volunteer_ID, (VolFirstName + ' ' + VolLastName) order by Total_Hours Desc	<div>Results Messages</div> <table><thead><tr><th></th><th>Volunteer_ID</th><th>Volunteer_Name</th><th>Total_Hours</th></tr></thead><tbody><tr><td>1</td><td>3087</td><td>Zonda Bassil</td><td>18.92</td></tr><tr><td>2</td><td>3018</td><td>Ardijs MacMechan</td><td>14.96</td></tr><tr><td>3</td><td>3053</td><td>Jordana Coom</td><td>12.76</td></tr><tr><td>4</td><td>3015</td><td>Ricky Piotrkowski</td><td>11.25</td></tr><tr><td>5</td><td>3019</td><td>Dareen Artiss</td><td>9.24</td></tr><tr><td>6</td><td>3016</td><td>Maureen Laugharan</td><td>8.73</td></tr><tr><td>7</td><td>3020</td><td>Egbert Plinck</td><td>8.7</td></tr><tr><td>8</td><td>3091</td><td>Heywood Baulch</td><td>8.61</td></tr><tr><td>9</td><td>3071</td><td>Diann Sager</td><td>8.4</td></tr><tr><td>10</td><td>3017</td><td>Preston Keedwell</td><td>8.4</td></tr><tr><td>11</td><td>3095</td><td>Marion Tench</td><td>8.19</td></tr><tr><td>12</td><td>3037</td><td>Abbie Whaley</td><td>7.9</td></tr></tbody></table>		Volunteer_ID	Volunteer_Name	Total_Hours	1	3087	Zonda Bassil	18.92	2	3018	Ardijs MacMechan	14.96	3	3053	Jordana Coom	12.76	4	3015	Ricky Piotrkowski	11.25	5	3019	Dareen Artiss	9.24	6	3016	Maureen Laugharan	8.73	7	3020	Egbert Plinck	8.7	8	3091	Heywood Baulch	8.61	9	3071	Diann Sager	8.4	10	3017	Preston Keedwell	8.4	11	3095	Marion Tench	8.19	12	3037	Abbie Whaley	7.9	We were able to find out the hours worked by each volunteer. Zonda Bassil had the most hours at 19.8 hours.
	Volunteer_ID	Volunteer_Name	Total_Hours																																																						
1	3087	Zonda Bassil	18.92																																																						
2	3018	Ardijs MacMechan	14.96																																																						
3	3053	Jordana Coom	12.76																																																						
4	3015	Ricky Piotrkowski	11.25																																																						
5	3019	Dareen Artiss	9.24																																																						
6	3016	Maureen Laugharan	8.73																																																						
7	3020	Egbert Plinck	8.7																																																						
8	3091	Heywood Baulch	8.61																																																						
9	3071	Diann Sager	8.4																																																						
10	3017	Preston Keedwell	8.4																																																						
11	3095	Marion Tench	8.19																																																						
12	3037	Abbie Whaley	7.9																																																						
2	How many items from each category were purchased in a given season?	This was also something they wanted to improve upon. Helps know what they'll need for next season.	select count(distinct SOLQuantity) as Number_of_Items, Category_ID, CategoryName from TSalesorder TSO join TSalesOrderLine TSSOL on TSO.SalesOrder_ID = TSSOL.SalesOrder_ID join TItem TI on TSSOL.SOLItem_ID = TI.Item_ID join TCategory TC on TI.ItemCategory_ID = TC.Category_ID where SODate between '2019-1-1' and '2019-5-1' group by Category_ID, CategoryName	<div>Results Messages</div> <table><thead><tr><th></th><th>Number_of_Items</th><th>Category_ID</th><th>CategoryName</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>701</td><td>corduroys</td></tr><tr><td>2</td><td>3</td><td>702</td><td>khakis</td></tr><tr><td>3</td><td>1</td><td>704</td><td>long sleeve shirt</td></tr><tr><td>4</td><td>1</td><td>705</td><td>short sleeve s...</td></tr><tr><td>5</td><td>2</td><td>706</td><td>coat</td></tr><tr><td>6</td><td>2</td><td>707</td><td>shoes</td></tr><tr><td>7</td><td>2</td><td>708</td><td>socks</td></tr><tr><td>8</td><td>1</td><td>709</td><td>underwear</td></tr></tbody></table>		Number_of_Items	Category_ID	CategoryName	1	2	701	corduroys	2	3	702	khakis	3	1	704	long sleeve shirt	4	1	705	short sleeve s...	5	2	706	coat	6	2	707	shoes	7	2	708	socks	8	1	709	underwear	We found the number of items purchased grouped by the category.																
	Number_of_Items	Category_ID	CategoryName																																																						
1	2	701	corduroys																																																						
2	3	702	khakis																																																						
3	1	704	long sleeve shirt																																																						
4	1	705	short sleeve s...																																																						
5	2	706	coat																																																						
6	2	707	shoes																																																						
7	2	708	socks																																																						
8	1	709	underwear																																																						

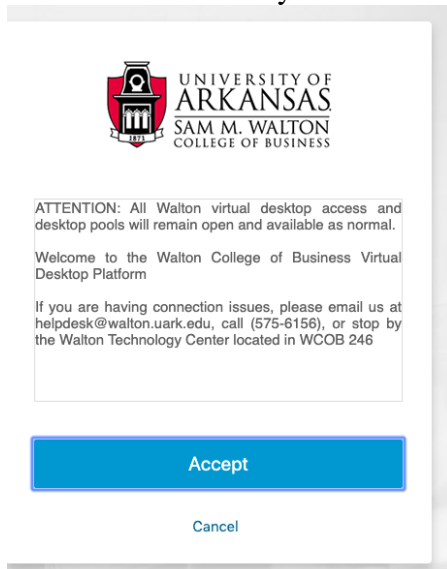
3	How many kits of each gender were purchased in a given season?	Helps them keep track of how many kits they have ready.	<pre>select count(TSO.SOKit_ID) as Number_of_Kits, TG. GenderDescription from TSalesOrder TSO join TKit TK on TSO.SOKit_ID = TK.Kit ID join TGender TG on TK.KitGender_ ID = TG.Gender_ID where SODate between '2019-1-1' and '2019-5-1' and TSO.SOKit ID is not null group by TG.GenderDescription</pre>	<div>Results Messages</div> <table><thead><tr><th></th><th>Number_of_K...</th><th>GenderDescription</th></tr></thead><tbody><tr><td>1</td><td>6</td><td>Female</td></tr><tr><td>2</td><td>10</td><td>Male</td></tr></tbody></table>		Number_of_K...	GenderDescription	1	6	Female	2	10	Male	We found that in that given time, 6 female kits and 10 male kits were purchased.
	Number_of_K...	GenderDescription												
1	6	Female												
2	10	Male												

User Documentation

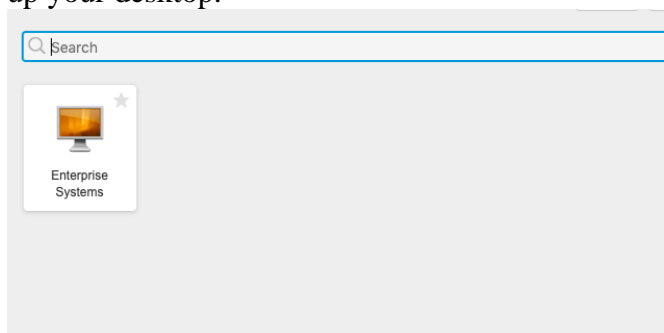
In order for our client, Operation School Bell, to actually log into and access the database we have created, the steps below should be followed. We have included a very detailed step-by-step procedure of how to exactly complete this. This process can be very tedious, so be sure to follow the written direction as closely as possible.

Steps to login to the Database

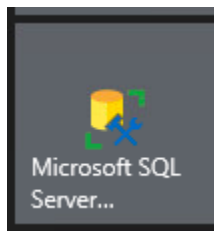
1. Open the email with your credentials for the database.
2. Click on the link <https://waltonlab.uark.edu/>
3. The link should take you to the following screen. Click the blue button “Accept”



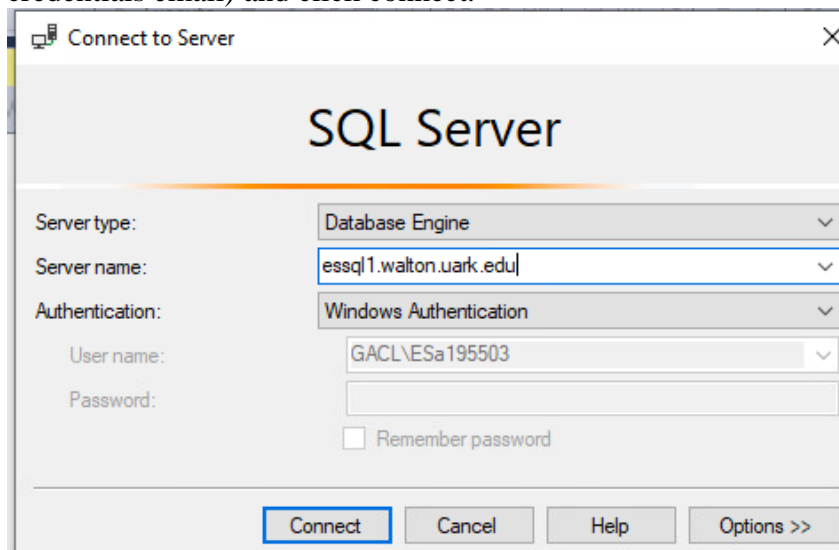
4. The next screen will ask you to input your credentials from the email. (From step 1)
5. Once you have logged in click “Enterprise Systems”. It may take a few minutes to open up your desktop.



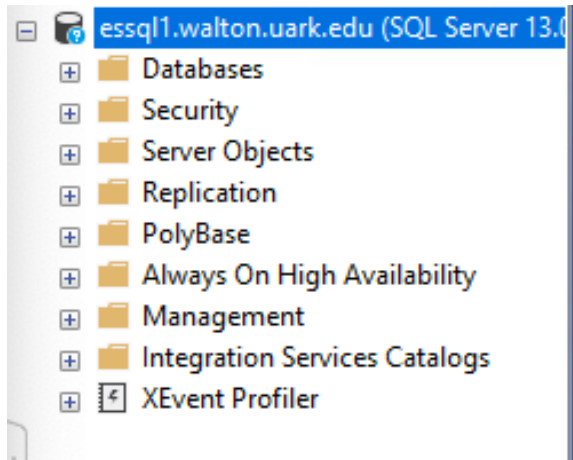
6. Your screen should open up to the following. Once this screen is up, you should click the windows icon in the bottom left hand corner of the screen and search “Microsoft SQL Server”. The icon is pictured below.



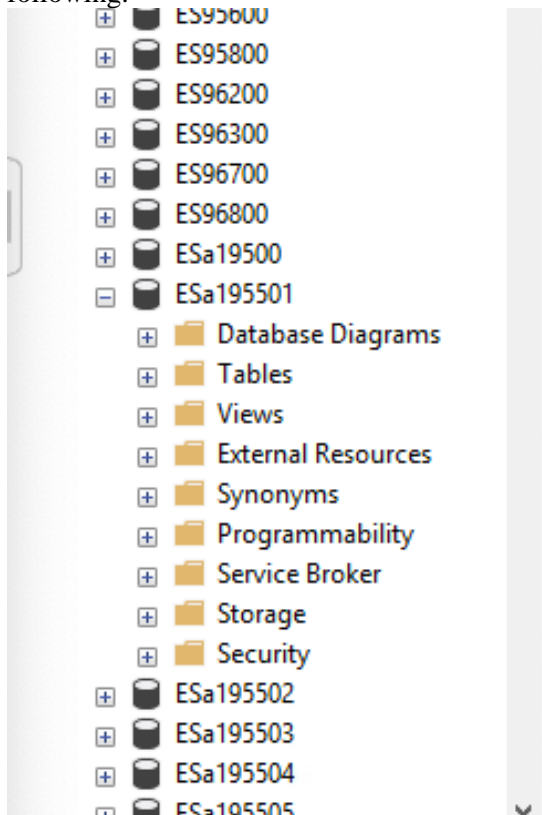
7. When Microsoft SQL server is open, you should input the server name (given on your credentials email) and click connect.



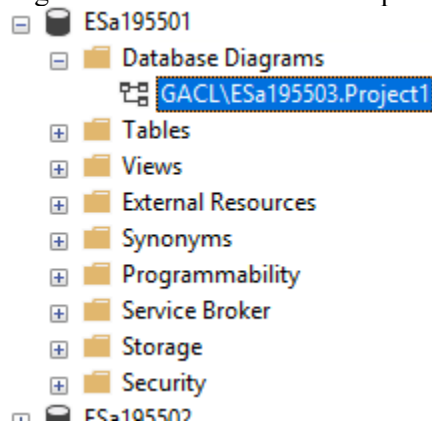
8. To access our database, you will want to click on the “plus-sign” next to the Databases folder



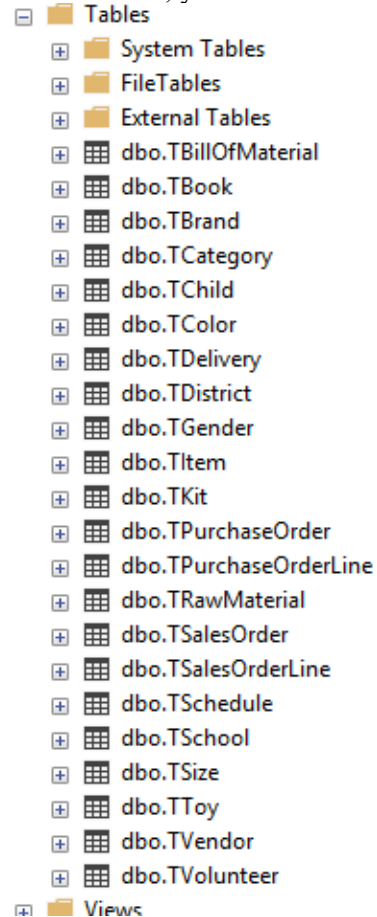
9. Once this is open, scroll down to “ESa195501” and click the same “plus-sign” bow next to this number. This is where our database is located. When this is open, it should look like the following.



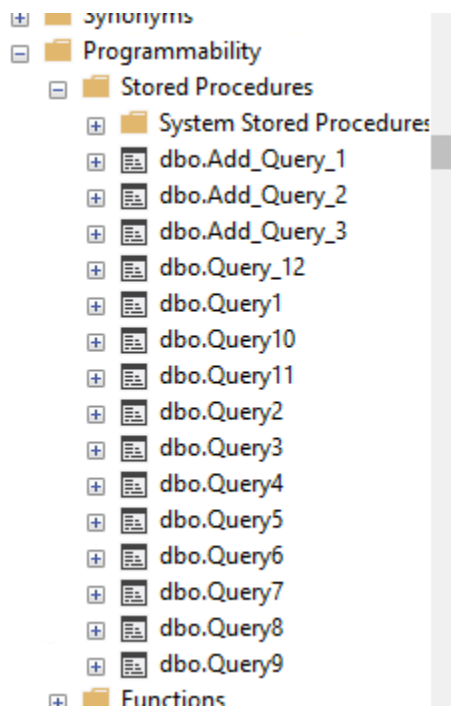
10. Within this menu, you can toggle between things to see what we have done. Under “Database Diagrams” is our database we implemented.



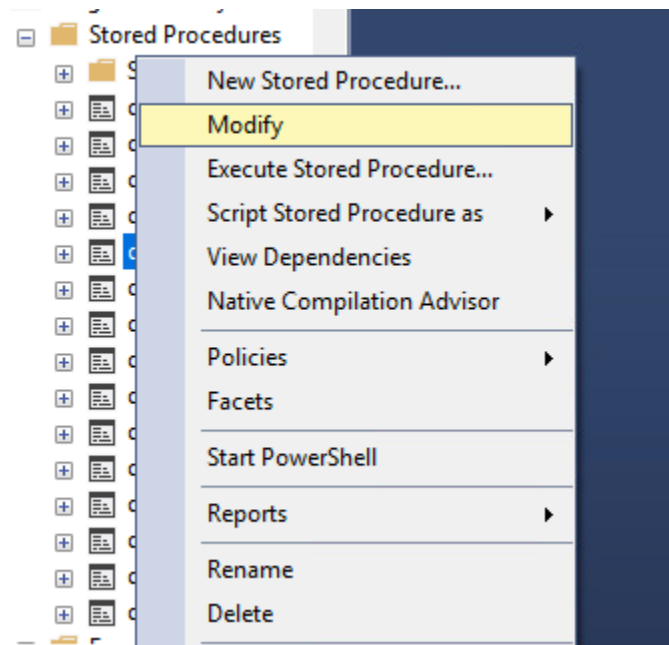
11. Under “Table”, you will be able to look at all the individual tables we implemented.



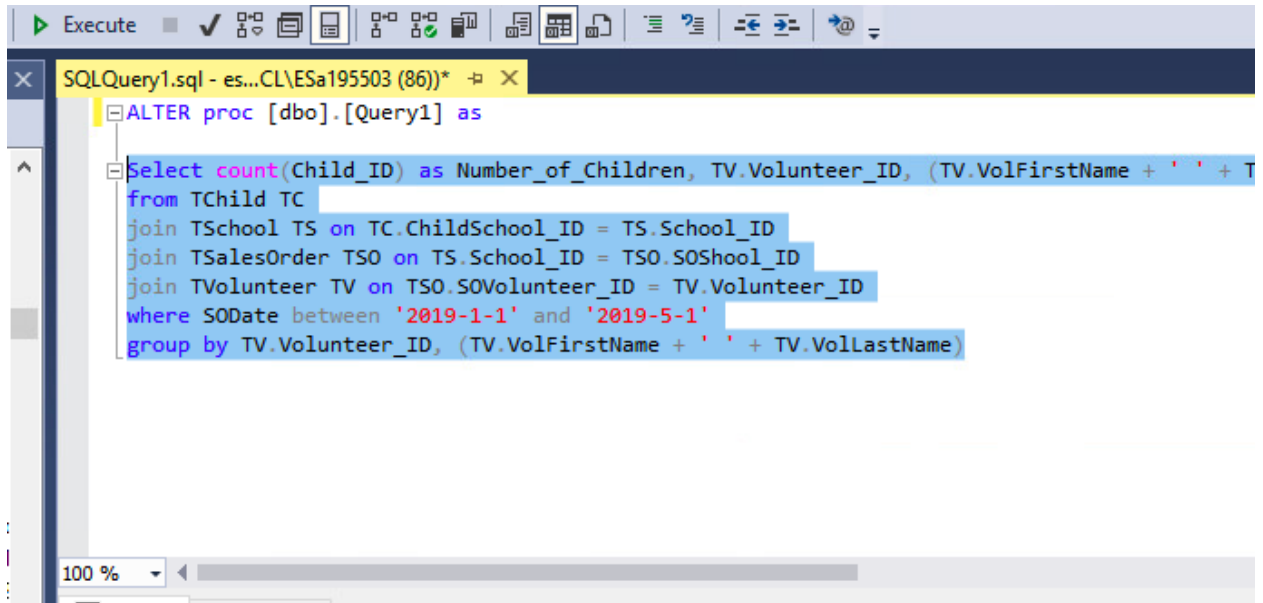
12. Under “Programmability” and then “Stored Procedures”, you can see all the queries we created.



13. To look at/run the queries we created, right click on the query you want and click “modify”.



14. The SQL code will appear. To run the query, select all of it and then click execute in the top left.



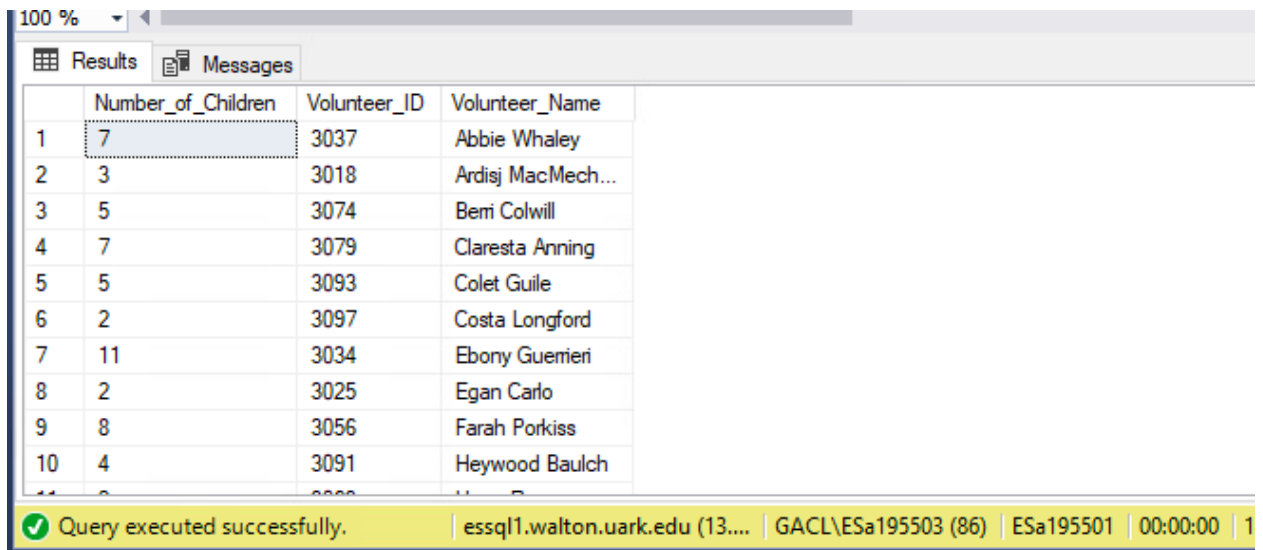
```

ALTER proc [dbo].[Query1] as

Select count(Child_ID) as Number_of_Children, TV.Volunteer_ID, (TV.VolFirstName + ' ' + TV.VolLastName) as Volunteer_Name
from TChild TC
join TSchool TS on TC.ChildSchool_ID = TS.School_ID
join TSalesOrder TSO on TS.School_ID = TSO.SOSchool_ID
join TVolunteer TV on TSO.SOVolunteer_ID = TV.Volunteer_ID
where SOSDate between '2019-1-1' and '2019-5-1'
group by TV.Volunteer_ID, (TV.VolFirstName + ' ' + TV.VolLastName)

```

15. Once you have run the query, the results will display at the bottom of the screen.

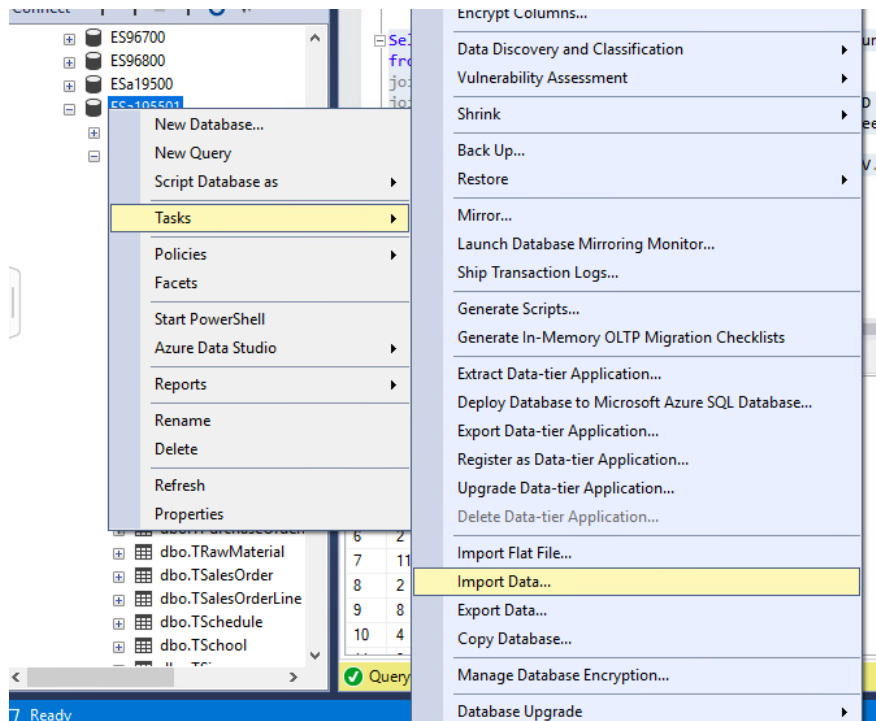


	Number_of_Children	Volunteer_ID	Volunteer_Name
1	7	3037	Abbie Whaley
2	3	3018	Ardisj MacMech...
3	5	3074	Berni Colwill
4	7	3079	Claresta Anning
5	5	3093	Colet Guile
6	2	3097	Costa Longford
7	11	3034	Ebony Guemieri
8	2	3025	Egan Carlo
9	8	3056	Farah Porkiss
10	4	3091	Heywood Baulch

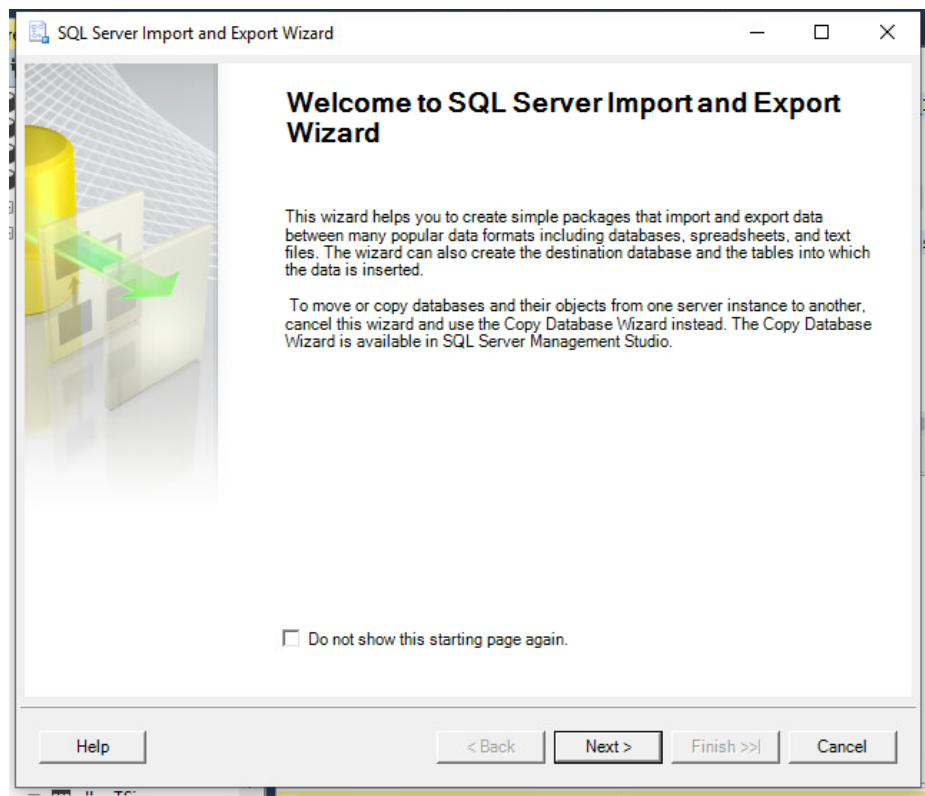
Query executed successfully. | essql1.walton.uark.edu (13.... | GACL\ESa195503 (86) | ESa195501 | 00:00:00 | 1

16. To run another query, repeat steps 13 through 15 for the desired query.

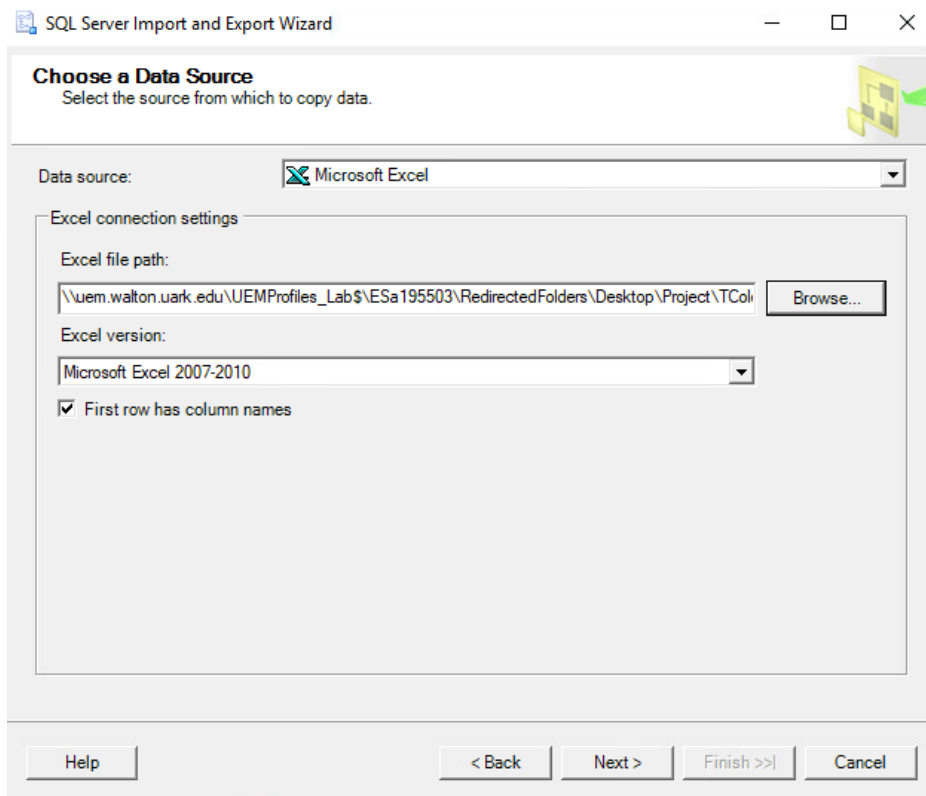
17. To enter a new data table or data, you will take the table you likely created in Excel and save that to your computer. Then go into the database and right click on the folder name. (In our case ESa195501) You will want to hover over “tasks” and then find “import data” and click that.



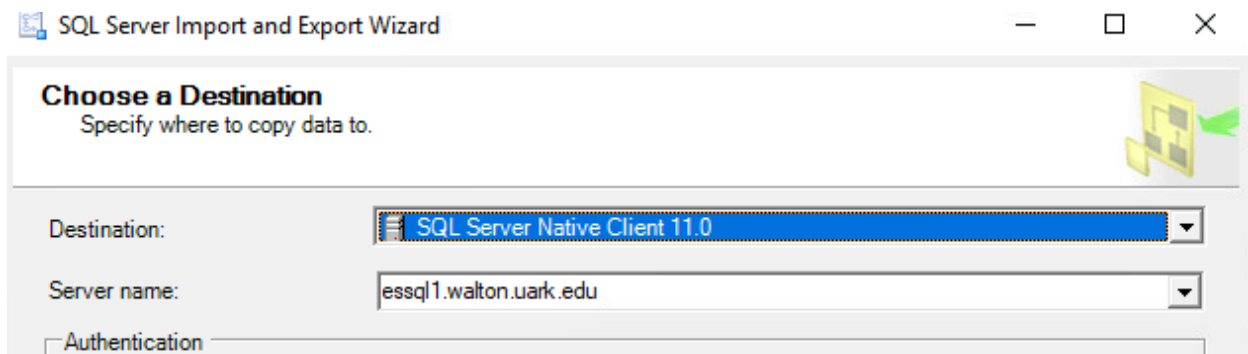
18. A box will appear that looks like the following. Click Next.



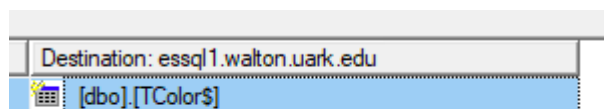
19. You will need to select Microsoft Excel and then choose the file you would like to import and click Next.

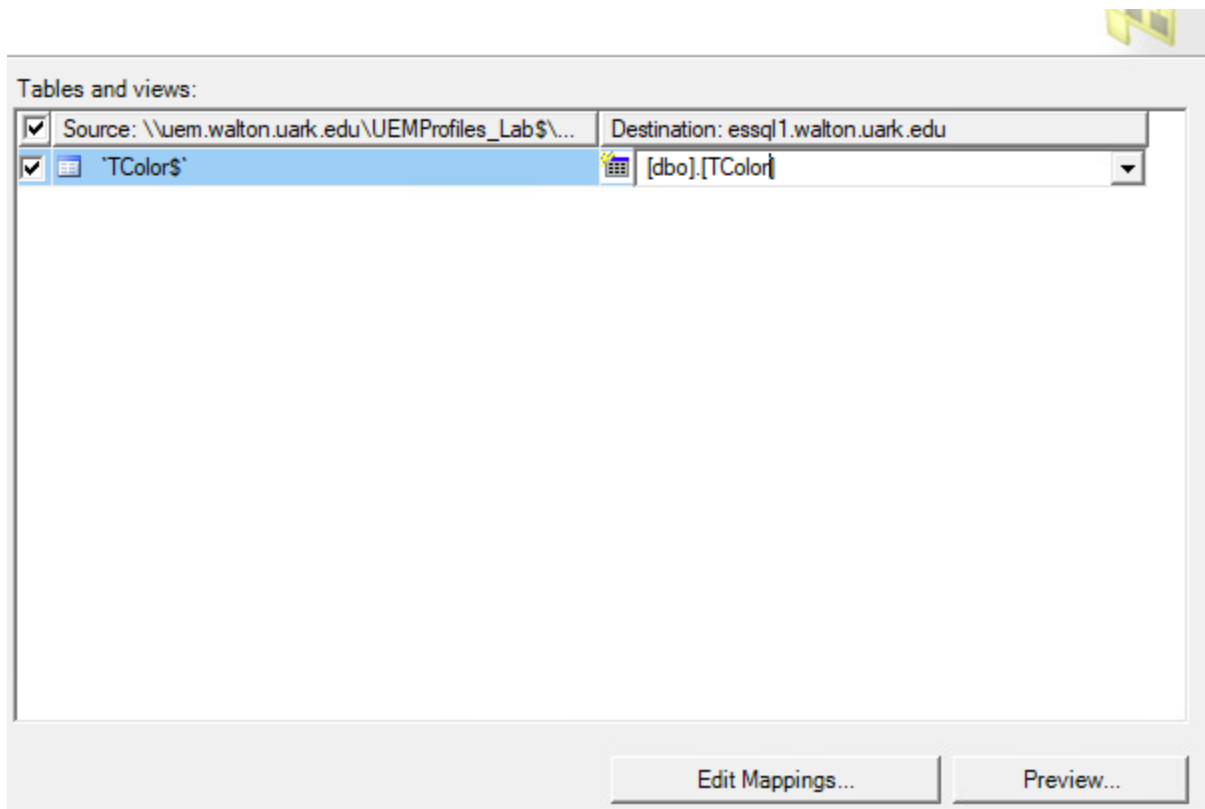


20. Select SQL Server Native Client 11.0 and click Next and Next again.



21. You must remove the \$ sign from the table name and the click Edit Mappings.





22. Choose the proper data types for your data and click ok. Then click Next three times and then finish. Your new table will be under the Tables folder.

☐ Create destination table Edit SQL...

☐ Delete rows in destination table ☐ Drop and re-create destination table

☒ Append rows to the destination table ☐ Enable identity insert

Mappings:

Source	Destination	Type	Nullable	Size	Precision	Scale
Color_ID	Color_ID	int	<input type="checkbox"/>			
ColorColorDescr...	ColorColorDescr...	nvarchar	<input type="checkbox"/>	50		

What We Learned Throughout This Process

The following table will show the problems and experiences encountered by each of our team members at various stages of the project. Based on the different division of labor of each member, the problems encountered by each person are also different. However, through the much effort that we put into this project and by the support of our team, we learned a lot from this project.

Member Name:	What you learned:
Uyen Tran	Over this semester, I know how to create the ERD and can provide the relationship for each entity, how to create normalization design and implement physical design. Also, I learned a lot of experience about teamwork.
Sidiki Ganame	Throughout this project, I have learned and improved my teamwork skills, communication, and collaboration. I learned how to start from creating an ERD from scratch to normalizing. I also learned the physical design and how to implement a database. My favorite part was the creation of the data dictionary, the tables and importing them to the database. I also got frustrated every time it didn't work, and we were not able to figure out the errors.
Elise Nill	I learned from this project the amount of thought and work that goes into creating and implementing a database. I learned the importance of creating an efficient ERD. Everything starts with your ERD, so it is important that all your relationships are correct, and all of your attributes are necessary. I also learned how to normalize an ERD and the importance of that. I also learned that this is a very tedious but rewarding process when everything finally works out in the end.
Liyuan Wang	Before this project, I didn't know anything about how to create a database. There were also many things I didn't understand in the process of doing this project. In the guidance of the teacher through classes and with the help and discussions of the group members, I learned the whole process of building the database and many small details that must be aware of in this process. I believe I will gain a lot of experiences after going through this class and finishing this project.

Appendix

Before beginning our project with Operation School Bell, our team met before to sign a contract. This contract explains what we expect from one another while completing this project. We also have a screenshot of our data dictionary we created for our database. This contains all the information for our tables we created. Finally, under the project management section, there are screenshots of our cost tracking excel sheets. These excel sheets reflect the amount of time each team member spent on specific tasks throughout the project.

Team Contract

Team Name: Dunder Mifflin Logo:

Team Motto: Work Hard

Team Members

Name	Email	Phone	Strengths	Availability to Meet
Elise Nill	Elise.c.nill-1@ou.edu	816-589-7804	- organized - prompt	Fridays
Uyen Tran	Uyen.M.Tran-1@ou.edu	405-600-8173	-crazy ideas	Friday, Saturday, and Sunday
Liyuan Wang	Liyuan.Wang-1@ou.edu	405-534-6038	-focused on details	Friday, Wednesday
Sidiki Ganame	Sidiki.Ganame-1@ou.edu	4053145811	-planning -creativity	Friday and Sunday

Unique Capabilities:

Proficient with computers, worked with SQL before, creative, problem-solvers, critical thinkers

Team Expectations (for Peer Evaluation):

Thoughtful evaluations, good work ethic, timeliness, quality

Presentation Date Preferences (Rank Order Available Dates; make sure you list dates that absolutely don't work for your team):

1. April 2nd
2. April 16th
3. March 10th

Data Dictionary Model

Below is our data dictionary. In this model we created all the twenty-two tables within our database. For each table we have our field names listed and for each of the field names we have if they are a key, its data type, if it is required, its default value, and a short description. We created our data dictionary before constructing our tables in the database, in order to get everything laid out and organized.

District					
Field Name	Key	Data Type	Reqd	Default Value	Description
District_ID	PK	AutoNumber	y		Random unique random
DistName		Text	y		name of the school district
School					
Field Name	Key	Data Type	Reqd	Default Value	Description
School_ID	PK	AutoNumber	y		Random unique number
SchoolName		Text	y		name of the school
SchoolDistrict_ID	FK references District	Number	y		
Child					
Field Name	Key	Data Type	Reqd	Default Value	Description
Child_ID	PK	AutoNumber	y		Random unique number
ChildDOB		Date			birth day of the child
ChildAge		Number			derive the age of the child
ChildFName		Name			first name of the child
ChildLName		Name			last name of the child
ChildParentHomePhone		Phone			child's parent's home phone number
ChildParentCellPhone		Phone			child's parent's cell phone number
ChildSchool_ID	FK references School	Number	y		
Volunteer					
Field Name	Key	Data Type	Reqd	Default Value	Description
Volunteer_ID	PK	AutoNumber	y		Random unique number
VolFirstName		Name			first name of volunteer
VolLastName		Name			last name of volunteer
VolStreet		Street			street address of volunteer
VolCity		City			city of address of volunteer
VolState		State			state of the volunteer
VolZipCode		Zip			zip code of the volunteer
VolCellPhone		Phone			volunteer cell phone
VolHomePhone		Phone			volunteer home phone
Schedule					
Field Name	Key	Data Type	Reqd	Default Value	Description
Schedule_ID	PK	AutoNumber	y		Random unique number
ScheVolunteer_ID	FK references Volunteer	Number	y		
ScheDate		Date			Date the volunteer worked
ScheClockInTime		Time			Volunteer Clock in time
ScheClockOutTime		Time			Volunteer Clock out time
ScheTotalHours		Number			Total hours worked derived from clock-in and clock-out time

Book					
Field Name	Key	Data Type	Reqd	Default Value	Description
Book_ID	PK	AutoNumber	y		Random unique number
BookGenre		text			genre of the book
BookQuantity		Number			how many books the child took
Toy					
Field Name	Key	Data Type	Reqd	Default Value	Description
Toy_ID	PK	AutoNumber	y		Random unique number
ToyDescription		text			description of the toy the child took
ToyQuantity		Number			quantity of toys the child took
Gender					
Field Name	Key	Data Type	Reqd	Default Value	Description
Gender_ID	PK	AutoNumber	y		Random unique number
GenderDescription		text			either male or female
Kit					
Field Name	Key	Data Type	Reqd	Default Value	Description
Kit_ID	PK	AutoNumber	y		Random unique number
KitGender_ID	FK references Gender	Number	y		
SaleOrder					
Field Name	Key	Data Type	Reqd	Default Value	Description
SalesOrder_ID	PK	Autonumber	y		Random unique random
SOVolunteer_ID	FK references Volunteer	number	y		ID of the Volunteer
SOSchool_ID	FK references School	number	y		ID of school
SOBook_ID	FK references Book	number	y		ID of the Books
SOKit_ID	FK references Kit	number	y		ID of kit
SOToy_ID	FK references Toy	number	y		ID of Toy
SOTotal		number			The total number of the sale order
SODate		date			The date of the order
SOCurrentGrade		number			current grade of child when place order
TDelivery					
Field Name	Key	Data Type	Reqd	Default Value	Description
DeliveryID	PK	Autonumber	y		Random unique random
DelVolunteer_ID - inspects	FK references Volunteer	number	y		the volunteer who inspect the order
DelVolunteer_ID - receives	FK references Volunteer	number	y		The volunteer who receives the order
DelStatus		text			status of the package
DelDate		date			date the package was delivered

TVendor					
Field Name	Key	Data Type	Reqd	Default Value	Description
Vendor_ID	PK	Autonumber	y		Random unique random
VName		Name			Name of Vendor
VVendorAccount#		Number			The Vendor account number
VStreet		Street			Vendors street address
Vstate		State			Vendor's state
VZipCode		Zip			Vendor's zipcode
VPhone		Phone			vendor phone number
VContactFirstName		Name			vendor contact first name
VContactLastName		Name			Vendor contact last name
Size					
Field Name	Key	Data Type	Reqd	Default Value	Description
Size_ID	PK	AutoNumber	y		item's size
SizeSizeDescription		Text			
Color					
Field Name	Key	Data Type	Reqd	Default Value	Description
Color_ID	PK	AutoNumber	y		item's color
ColorColorDescription		Text			
Category					
Field Name	Key	Data Type	Reqd	Default Value	Description
Category_ID	PK	AutoNumber	y		ID of category
CategoryName		Text			name of category
Brand					
Field Name	Key	Data Type	Reqd	Default Value	Description
Brand_ID	Pk	Autonumber	y		ID of Brand
BrandBrandName		Text			name of the brand
Item					
Field Name	Key	Data Type	Reqd	Default Value	Description
Item_ID	PK	AutoNumber	y		ID of Item
ItemVendor_ID	FK references Vendor	Number	y		vendor ID of Item
ItemGender_ID	FK references Gender	Number	y		Gender of child who receives the item
ItemSize_ID	FK references Size	Number	y		item's size
ItemColor_ID	FK references Color	Number	y		the color of the item
ItemBrand_ID	FK references Brand	Number	y		item brand
ItemCategory_ID	FK references Category	Number	y		category of item
ItemCost		Number			cost of items
ItemQtyOH		Number			item quantity

SalesOrderLine					
Field Name	Key	Data Type	Reqd	Default Value	Description
Sale_Order_Line_ID	PK	Autonumber	y		ID of sale order line
SOLItem_ID	FK references Item	number	y		Item's ID when order
SOLSales_Order_ID	FK references SalesOrder	number	y		ID of the sale order
SOLQuantity		number			quantity of the sale order
SOLTotal		number			total number of the sale order
TPurchaseOrder					
Field Name	Key	Data Type	Reqd	Default Value	Description
Purchase_Order_ID	PK	AutoNumber	y		ID of purchase order
POVendor_ID	FK references Vendor	AutoNumber	y		ID of vendor
POVolunteer_ID	FK references Volunteer	AutoNumber	y		ID of volunteer
PODate		Date			date of purchase order
POStatus		Boolean			status of the purchase order - Completed (Y/N)?
POTotal		Number			total number of the purchase order
TPurchase OrderLine					
Field Name	Key	Data Type	Reqd	Default Value	Description
Order_Line_ID	PK	AutoNumber	y		Random unique number
POLPurchase_Order_ID	FK references PurchaseOrder	AutoNumber	y		ID of Purchase Order
POLItem_ID	FK references Item	AutoNumber	y		ID of Item
POLDelivery_ID	FK references Delivery	AutoNumber	y		ID of Delivery
POLQuantity		Number	ChC		Quantity of POL
POLLlineTotal		Number			Total quantity order
TRawMaterial					
Field Name	Key	Data Type	Reqd	Default Value	Description
RM_ID	PK	AutoNumber	y		Random unique number
RMTType		text			
RMQty_OH		Number			Quantity of Raw Material on hand
TBillOfMaterial					
Field Name	Key	Data Type	Reqd	Default Value	Description
BOM_ID	PK	AutoNumber	y		Random unique number
BOMItem_ID	FK references Item	AutoNumber	y		ID of Item
BOMKit_ID	FK references Kit	AutoNumber	y		ID of Kit
BOMRM_ID	FK references RawMaterial	AutoNumber	y		ID of Raw Material
BOMRawMatQuantity		Number			Total quantity of Raw Material acquired

Project Management

Project Start Date	3/6/20			Project End Date	4/26/20		Cost (per 60 min)	\$25
	Student Name	Duration (Min)	% Complete	Planned Minutes	Actual Minutes	Difference Minutes	Subtotal Minutes	Subtotal Cost
Milestone 1								
Read Case + Prepare Questions for client	Elise Nill, Sidiki Ganame, Uyen Tran, Liyuan Wang	225		120	225	-105		
Client Meeting		227		160	227	-67		
ERD Design		330		480	330	150		
Assumptions		165		280	165	115		
Write-up preparation		425		400	425	-25		
Sub Total							68	1372 \$572
Milestone 2								
ERD Design		645		720	645	75		
Normalization		505		320	505	-185		
Write-up preparation		295		180	295	-115		
Sub Total							-225	1445 \$602
Milestone 3								
Implementation/SQL		2400		1680	2400	-720		
Milestone 2 Rework		300		280	300	-20		
Data dictionary		890		720	890	-170		
Write-up preparation		280		360	280	80		
Sub Total							-830	3870 \$1,613
Final Submission								
Queries		900	100%	960	900	60		
Write-up preparation		580	100%	360	580	-220		
Formatting and Review		90	100%	90	90	0		
							0	
Sub Total							-160	1570 \$654
						Total	8257	\$3,440.42