Susan Rico

IST659 Project Deliverable

Part I: Design

Project Narrative 2

Data Dictionary 2

Data Questions 3

Entity Relationship Diagram 3

Logical Model Diagram 4

Part II: Implementation

SQL DDL 5

Data Creation 6

Data Manipulation 7

Data Questions 9

User Interface 11

Reflection 11

**Part I - Design**

**Project Narrative**

The Texas Health and Human Service Commission - Disaster Case Management (HHSC - DCM) section has contracted two separate vendors, which are currently handling Disaster Case Management (DCM) services related to Hurricane Harvey recovery efforts. Bi-monthly, a team of HHSC - DCM members conduct a monitoring on case files for clients being serviced by each specific vendor using the HHSC DCM Monitoring Tool (Appendix A). These are done to provide valuable feedback to the DCM vendor as well as to ensure that the vendors are meeting quality standards. During this process, each staff member assigned to the audit visit reviews approximately 20-25 case files, totaling over 150 monitored case files within a span of 3-4 days. The DCM vendors receive post-monitoring reports including findings as well as a corrective action plan. The findings allow the HHSC team to make suggestion on ways to improve workflow and DCM quality at the vendor level to ensure compliance. The HHSC DCM team would like to establish a streamlined process for entering and storing pertinent data related to the monitoring. This would facilitate the reporting process and ultimately, increase productivity and efficiency.

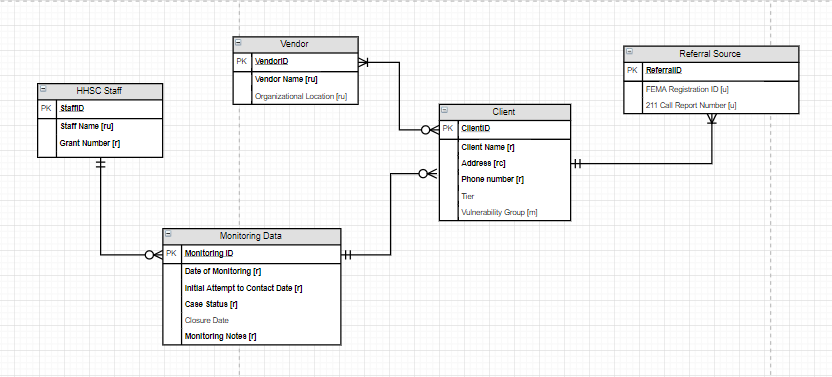
**Data Dictionary** 

**Data Questions**

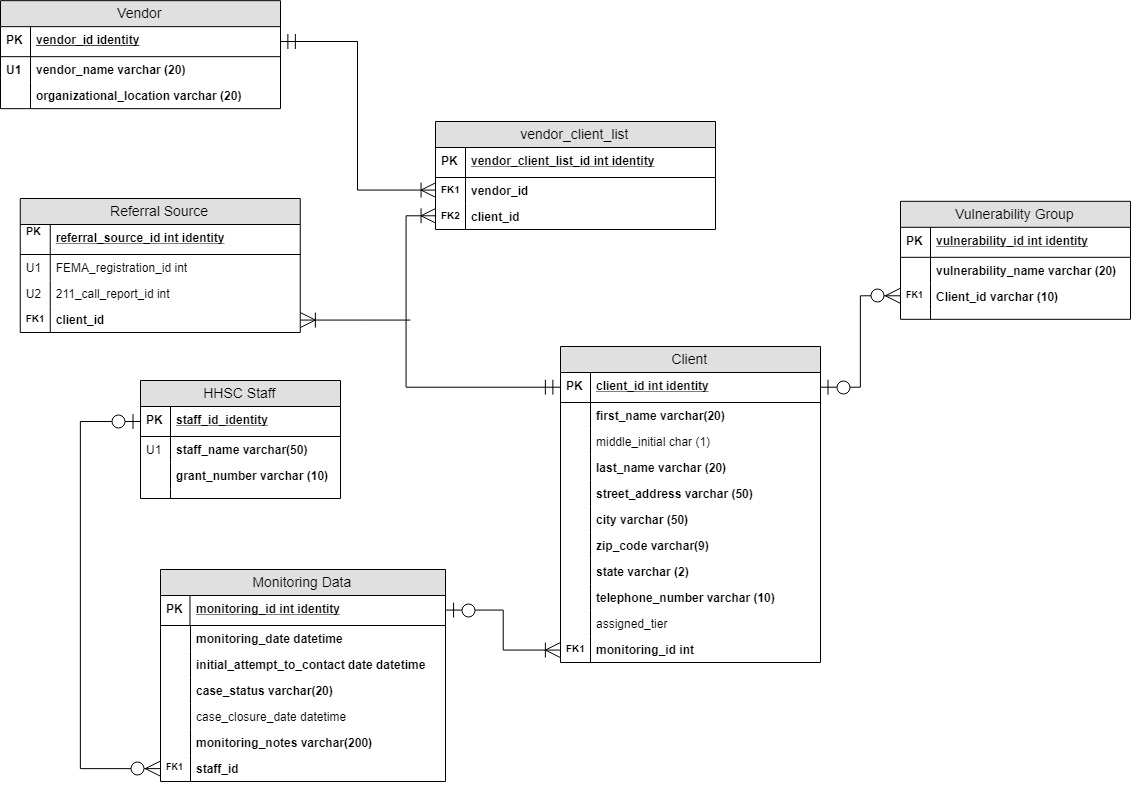
The HHSC DCM data team intends to answer the following questions:

1. How many case files is each member of the HHSC DCM staff reviewing during each monitoring?
2. What is the breakdown by status of the files monitored?
3. What is the duration between Initial Attempt to Contact date and Closure Date?
4. What is the count according to Vulnerability Group?
5. What are the case statuses by organizational office?

**Entity Relationship Diagram**



**Logical Model Diagram**



**Part II - IMPLEMENTATION**

**SQL DDL**

/\*

Author: Susan Rico

Course: IST659 M400

Term: January, 2019

\*/

--Creating the Staff table

CREATE TABLE Staff(

staff\_id int identity primary key,

staff\_lastname varchar (50) not null,

grant\_number varchar (10) not null

)

--Creating the MonitoringData table

CREATE TABLE MonitoringData(

monitoring\_id int identity primary key,

monitoring\_date datetime not null default GETDATE(),

initial\_attempt\_to\_contact\_date datetime,

case\_status varchar (30) not null,

case\_closure\_date datetime,

monitoring\_notes varchar (500) not null,

staff\_id int not null,

CONSTRAINT FK1\_MonitoringData FOREIGN KEY (staff\_id) REFERENCES Staff (staff\_id)

)

--Creating the Client table

CREATE TABLE Client(

client\_id int identity primary key,

client\_firstname varchar (20) not null,

client\_middle\_initial varchar (1),

client\_lastname varchar (20) not null,

street\_address varchar (50) not null,

client\_city varchar (50) not null,

client\_zipcode varchar (10) not null,

client\_telephone varchar (20) not null,

client\_tier varchar (5),

monitoring\_id int,

vulnerability\_id int,

-- Constraints

CONSTRAINT U1\_Client UNIQUE (client\_telephone),

CONSTRAINT FK1\_Client FOREIGN KEY (monitoring\_id) REFERENCES monitoringdata(monitoring\_id),

CONSTRAINT FK2\_Client FOREIGN KEY (vulnerability\_id) REFERENCES VulnerabilityGroup (vulnerability\_id)

)

--Create Vulnerability Group table

CREATE TABLE VulnerabilityGroup(

vulnerability\_id int identity primary key,

vulnerability\_name varchar (30) null

)

-- Create Vendor table

CREATE TABLE Vendor(

vendor\_id int identity primary key,

vendor\_name varchar(20) not null,

organizational\_location varchar (20) not null

)

--Create VendorClientList

CREATE TABLE VendorClientList(

vendorclientlist\_id int identity primary key,

vendor\_id int not null,

client\_id int not null,

CONSTRAINT FK1\_VendorClientList FOREIGN KEY (vendor\_id) REFERENCES Vendor (vendor\_id),

CONSTRAINT FK2\_VendorClientList FOREIGN KEY (client\_id) REFERENCES Client (client\_id)

)

--Create ReferralSource table

CREATE TABLE ReferralSource(

referralsource\_id int identity primary key,

FEMARegistration\_id int null,

CallReport\_id int null,

client\_id int not null,

CONSTRAINT FK1\_ReferralSource FOREIGN KEY (client\_id) REFERENCES Client (client\_id)

)

**Through this process, I had to alter and eventually drop my tables after attempting to load my mock data from Excel. I learned that first I needed to drop the constraints in order to drop the table since it had a relationship with other tables. See below for example.**

ALTER TABLE MonitoringData DROP CONSTRAINT FK1\_staffID

DROP TABLE MonitoringData

**DATA CREATION**

**Below is a portion of the INSERT statements I used to load my data from Excel on to my database.**

--Transform and load

INSERT INTO Staff (staff\_lastname, grant\_number)

SELECT staff\_lastname, grant\_number FROM staff$

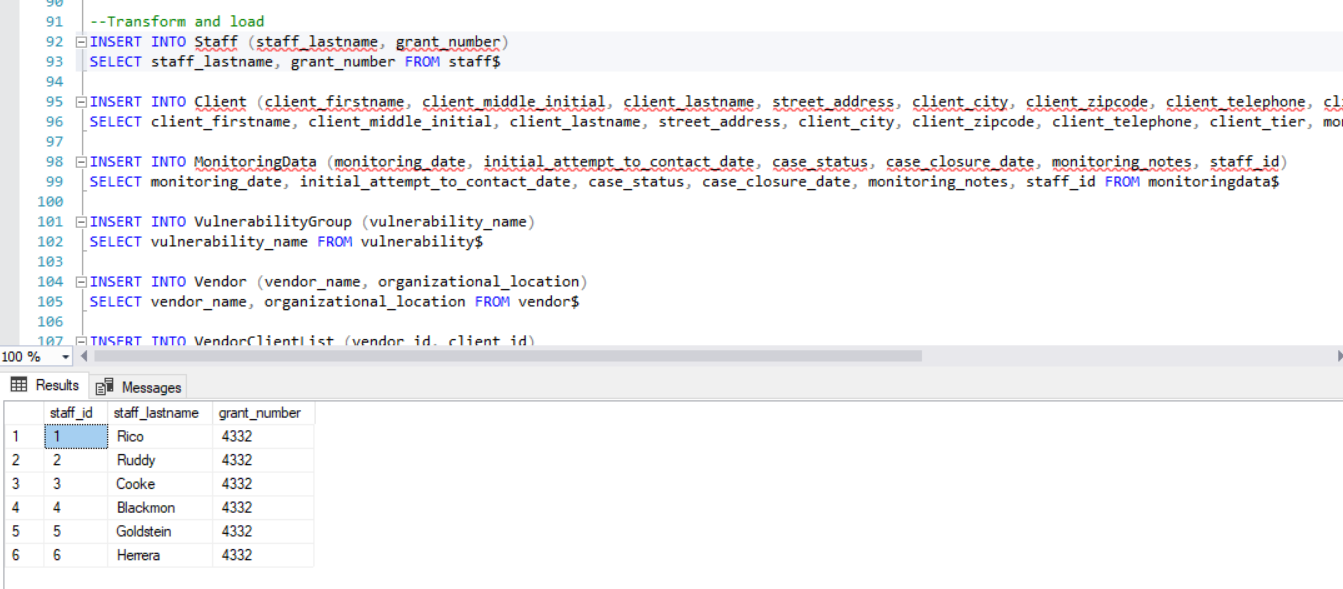
INSERT INTO Client (client\_firstname, client\_middle\_initial, client\_lastname, street\_address, client\_city, client\_zipcode, client\_telephone, client\_tier, monitoring\_id, vulnerability\_id)

SELECT client\_firstname, client\_middle\_initial, client\_lastname, street\_address, client\_city, client\_zipcode, client\_telephone, client\_tier, monitoring\_id, vulnerability\_id FROM client$

INSERT INTO MonitoringData (monitoring\_date, initial\_attempt\_to\_contact\_date, case\_status, case\_closure\_date, monitoring\_notes, staff\_id)

SELECT monitoring\_date, initial\_attempt\_to\_contact\_date, case\_status, case\_closure\_date, monitoring\_notes, staff\_id FROM monitoringdata$

**Loading Excel Data**



**The figure above displays some of the INSERT statements used to load data from Excel.**



**Once loaded, I used “SELECT” statements to confirm that the Excel data was properly added on to my SQL Server Database.**

**DATA MANIPULATION**

**In case of the addition of staff members to the monitoring process, an INSERT statement as seen below would add the new staff member into the database.**

-- Add HHSC Staff member to Staff table

INSERT INTO Staff

(staff\_lastname, grant\_number)

VALUES

('Gore', 4332)

SELECT \* FROM Staff

**It is valuable to know how many cases are reviewed by each of the HHSC Staff. I created a function to count these.**

--Create function to yield total count of monitorings completed by each member of

--HHSC Staff

GO

CREATE FUNCTION dbo.StaffMonitoringCount(@staffid int)

RETURNS int AS

BEGIN

DECLARE @returnValue int

SELECT @returnValue = COUNT (staff\_id) FROM MonitoringData

WHERE MonitoringData.staff\_id = @staffid

RETURN @returnValue

END

GO

--Use dbo.StaffMonitoringCount function to return counts of monitoring data

--by each member of HHSC staff

SELECT staff\_id

, dbo.StaffMonitoringCount(Staff\_id) as StaffMonitoringCount

FROM MonitoringData

GROUP BY staff\_id

ORDER BY StaffMonitoringCount DESC

**After completing a monitoring review, HHSC staff reviews the length between Initial Case Contact and Closure day. I created a function to return the duration in days followed by a SELECT statement. Since closure dates only exist on cases that are closed, I included a WHERE statement to only include ‘Closed’ cases.**

--Create function to return days between Initial Contact and Closure

GO

CREATE FUNCTION dbo.DurationBetweenInitialContactandClosure (@monitoringid int)

RETURNS int AS

BEGIN

DECLARE @returnValue int

SELECT @returnValue = SUM (DATEDIFF(D, initial\_attempt\_to\_contact\_date, case\_closure\_date)) FROM

MonitoringData

WHERE MonitoringData.monitoring\_id = @monitoringid

RETURN @RETURNVALUE

END

GO

-- Return duration days

SELECT \*

, dbo.DurationBetweenInitialContactandClosure (monitoring\_id) as TotalDurationDays

FROM MonitoringData

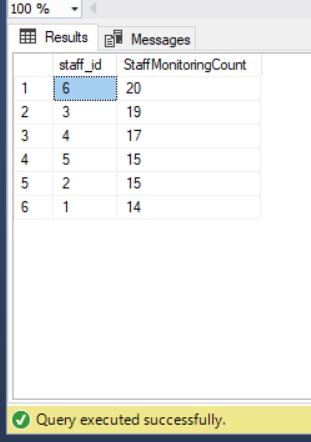
WHERE case\_status = 'closed'

--Select all cases that with a Case Status of "Closed"

SELECT \* FROM MonitoringData WHERE case\_status = 'Closed'

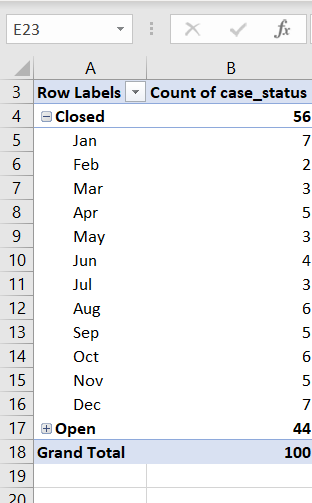
**Data Questions**

1. How many case files is each member of the HHSC DCM staff reviewing during each monitoring?



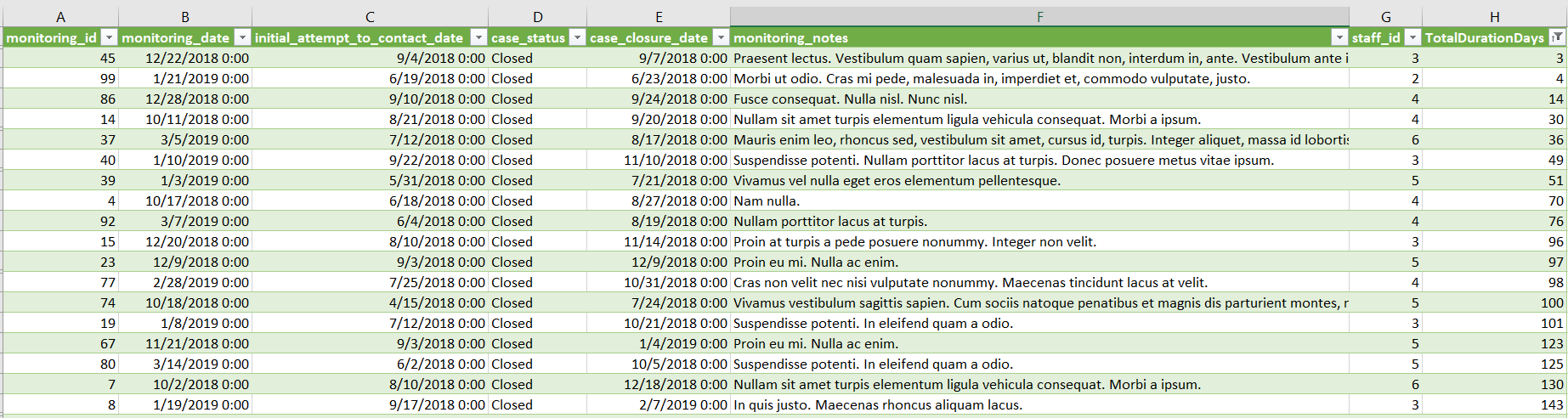
* In a real-world situation, I would also be able to provide reliable “Monitoring Dates” but due to using generated mock data, this was a challenge.

1. **What is the breakdown by status of the files monitored?**

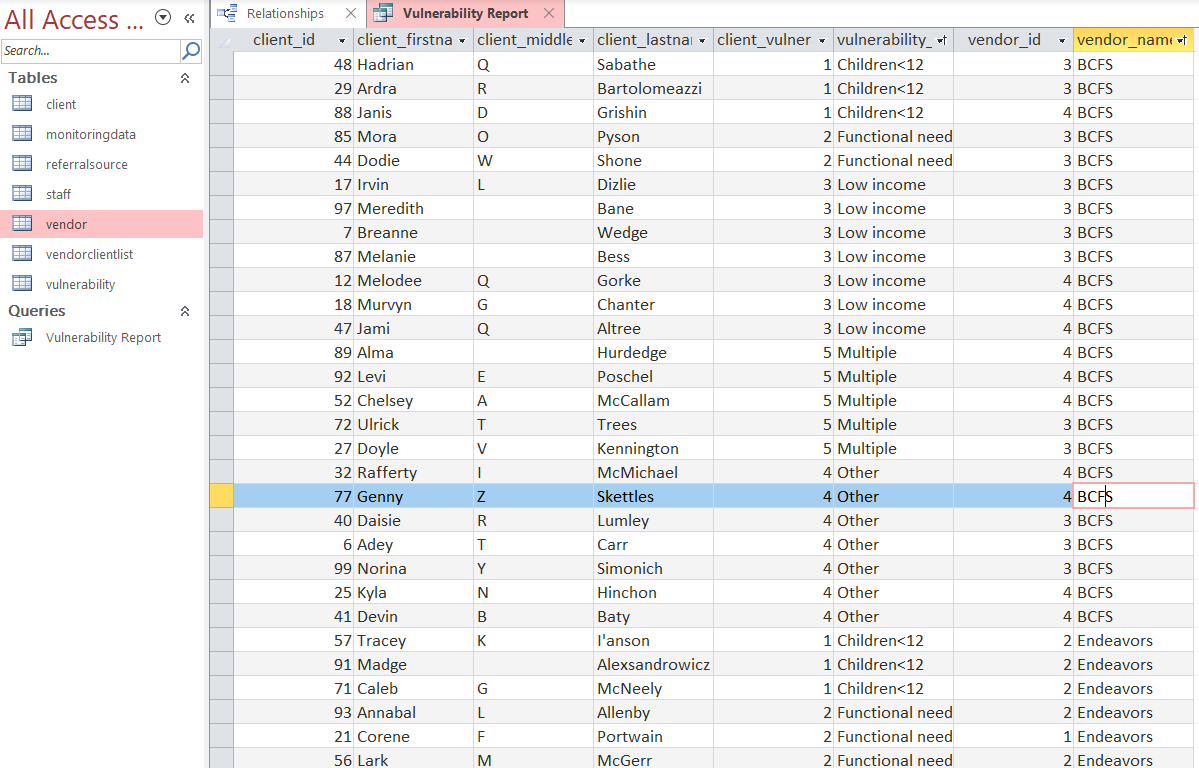


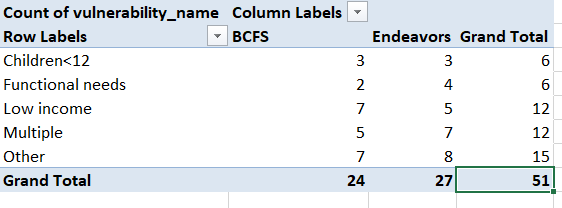
* I was able to insert my data from SQL server on to Excel and used the Monitoring Data table to insert a pivot table to provide this view.
* Although basic, it provides quick facts regarding the outcome of our monitoring reviews that are included in the Corrective Action Plan for our two vendors.
* This pivot table breaks down the monitoring review by case status as well as adds an element of Closure Month.

1. **What is the duration between Initial Attempt to Contact date and Closure Date?**



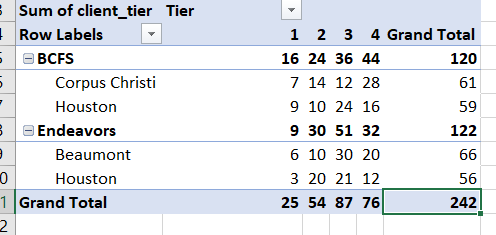
1. **What is the count according to Vulnerability Group?**





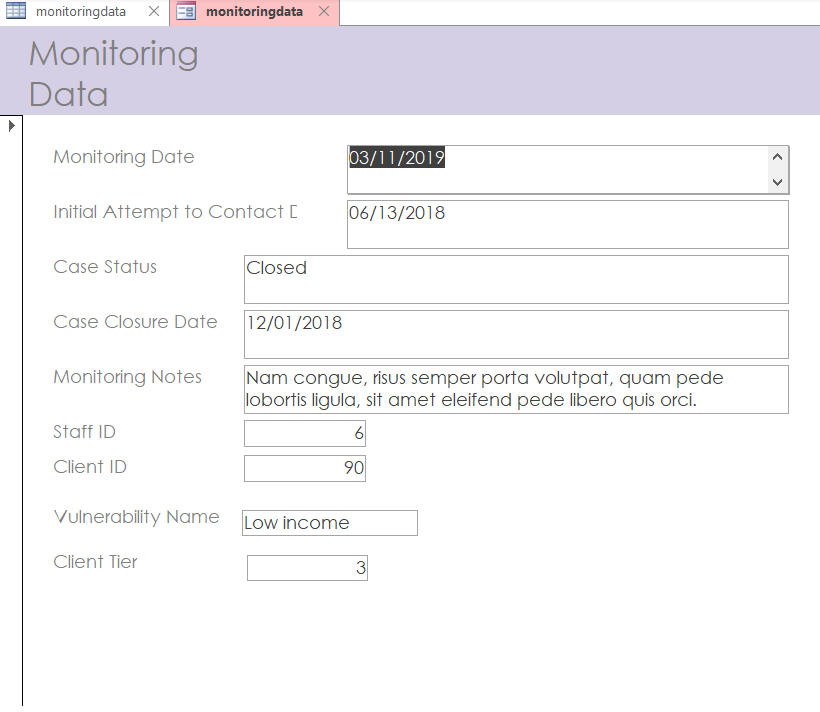
Due to resource limitations, our real-world database would likely run off Access. However, Access can be more user friendly as it does not require much coding experience. I have provided the examples of what a Vulnerability Report would look like in Access. Within my query, I also pulled the Vendor Name from my Vendor table. We can then export it into Excel and create a very user-friendly pivot tables for quick reading.

1. **What are the Case Tier levels by organizational office?**



* I created a query on Access to derive the organizational location for each vendor and their respective counts for case tier.
* This can help establish any trends for populations needs and risks based on geographical location.

**USER INTERFACE**



**Above is an example of a monitoring data form created in Access.**

**REFFLECTION**

As part of this process, I have learned that establishing a solid and accurate ERD and Logical Model Diagram is very important. It was during the loading of the data that I learned that some of my relationships established were faulty. I have also learned that there are a multitude of resources to continue my self-learning process.

I hope to continue to fine-tune this process and eventually establish this database as part of my job. It would facilitate and streamline our current process in place. I’ve learned that it is very important to establish rules for our data and really capture the goal of what we are trying to achieve. I’m looking forward to continuing to learn about databases.