# Introduction

As a team, we created a program for Scott and Grimes Security Consultants to help manage customer relations and related services. This program allows for employees of Scott and Grimes Security Consultants to add new customers and their locations, add estimates and the services estimated, add services performed, and view the sales data generated from the services performed. Additionally, this program also stores database information regarding consultants, consultant’s degrees/skills/experiences, and services available. These database tables have been seeded with data because creating forms to add data to these tables was out of the scope of the project.

This document lays important schema information regarding the program, such as the ERD, data dictionary, and user manual.

## Feature List

* Add new customers to the database system
* Add, update, and delete estimates and estimated services
* Add, update, and delete services performed
* Generate sales reports based on the above tables by date

Contents

[Introduction 1](#_Toc26799491)

[Feature List 1](#_Toc26799492)

[Entity Relationship Diagram 3](#_Toc26799493)

[Tables 4](#_Toc26799494)

[Create Statements 4](#_Toc26799495)

[Insertion Statements 8](#_Toc26799496)

[SQL Statements 16](#_Toc26799497)

[Get All Customers 16](#_Toc26799498)

[Results 16](#_Toc26799499)

[Get Business Consultants 16](#_Toc26799500)

[Results 16](#_Toc26799501)

[Get Services 16](#_Toc26799502)

[Results 16](#_Toc26799503)

[Get Addresses 17](#_Toc26799504)

[Results 17](#_Toc26799505)

[Get Services for an Estimate 17](#_Toc26799506)

[Results 17](#_Toc26799507)

[Get Sales Data 17](#_Toc26799508)

[Results 17](#_Toc26799509)

[Get Sales Summary 17](#_Toc26799510)

[Results 18](#_Toc26799511)

[Get Expected Sales Summary 18](#_Toc26799512)

[Results 18](#_Toc26799513)

[Data Dictionary 19](#_Toc26799514)

[If the column is RED, then it is a foreign key. If the column is YELLOW, then it is a primary key. If the column is PURPLE, then it is both a primary key and a foreign key. 19](#_Toc26799515)

[Customer 19](#_Toc26799516)

[ServicesPerformed 20](#_Toc26799517)

[LocationHasServicesPerformed 20](#_Toc26799518)

[ServicesPerformedHasService 20](#_Toc26799519)

[EstimateHasService 20](#_Toc26799520)

[BusinessConsultant 21](#_Toc26799521)

[TechnicalConsultant 21](#_Toc26799522)

[TechnicalConsultantHasDegree 21](#_Toc26799523)

[Degree 21](#_Toc26799524)

[BusinessExperienceHasDegree 21](#_Toc26799525)

[Location 22](#_Toc26799526)

[Service 22](#_Toc26799527)

[BusinessExperience 22](#_Toc26799528)

[Consultant 23](#_Toc26799529)

[TechnicalSkill 23](#_Toc26799530)

[Estimate 23](#_Toc26799531)

[User Manual 24](#_Toc26799532)

[Sign in Form 24](#_Toc26799533)

[Main Form 24](https://pittsburgstate-my.sharepoint.com/personal/chris_evans_gus_pittstate_edu/Documents/Microsoft%20Teams%20Chat%20Files/Advanced%20Db%20Final%20Security%20Consultants.docx#_Toc26799534)

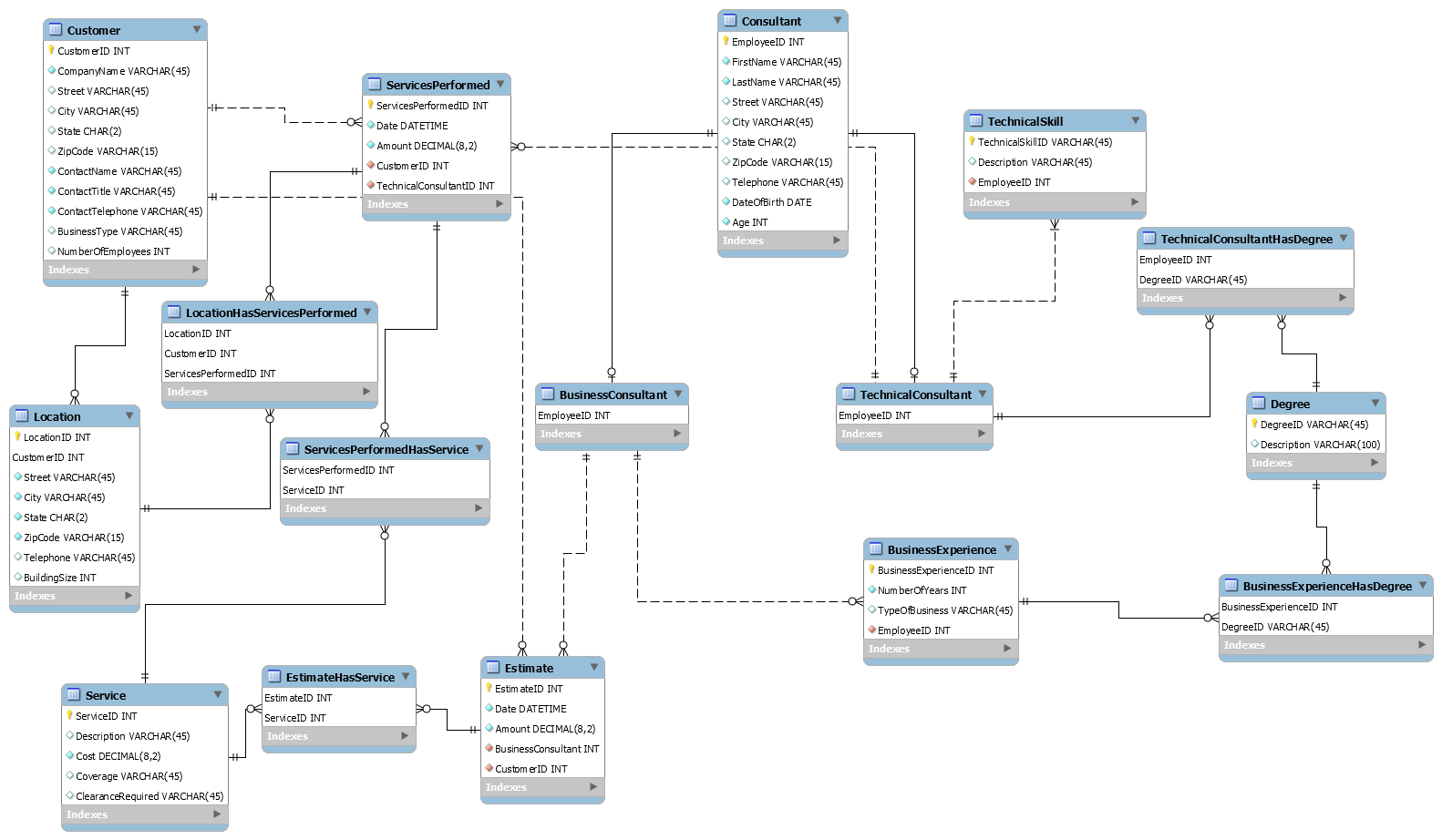
[Add Customer Form 25](#_Toc26799535)

[Estimates Form 26](https://pittsburgstate-my.sharepoint.com/personal/chris_evans_gus_pittstate_edu/Documents/Microsoft%20Teams%20Chat%20Files/Advanced%20Db%20Final%20Security%20Consultants.docx#_Toc26799536)

[Services Performed Form 29](#_Toc26799537)

[Sales Report Form 31](#_Toc26799538)

# Entity Relationship Diagram



# Tables

The Create SQL script can also be found in the source code

## Create Statements

Create table Customer (

CustomerID int,

CompanyName varchar(45) not null,

Street varchar(45),

City varchar(45),

State char(2),

ZipCode varchar(15),

ContactName varchar(45) not null,

ContactTitle varchar(45) not null,

ContactTelephone varchar(45) not null,

BusinessType varchar(45),

NumberOfEmployees int,

Primary Key(CustomerID));

Create table Location(

LocationID int,

CustomerID int,

Street varchar(45) not null,

City varchar(45) not null,

State char(2) not null,

ZipCode varchar(15),

Telephone varchar(45),

BuildingSize int,

Primary Key(LocationID, CustomerID));

Create table Consultant(

EmployeeID int,

FirstName varchar(45) not null,

LastName varchar(45) not null,

Street varchar(45),

City varchar(45),

State char(2),

ZipCode varchar(15),

Telephone varchar(45),

DateOfBirth date not null,

Age int not null,

Primary Key(EmployeeID));

Create table TechnicalConsultant(

EmployeeID int primary key references Consultant(EmployeeID));

Create table BusinessConsultant(

EmployeeID int primary key references Consultant(EmployeeID));

Create table ServicesPerformed(

ServicesPerformedID int,

Date datetime not null,

Amount decimal(8,2) not null,

Primary Key(ServicesPerformedID),

CustomerID int Foreign Key References Customer(CustomerID) not null,

TechnicalConsultantID int Foreign Key References TechnicalConsultant(EmployeeID) not null);

Create table LocationHasServicesPerformed(

LocationID int,

CustomerID int,

Constraint FK\_Location Foreign Key (LocationID, CustomerID) references Location(LocationID,CustomerID),

ServicesPerformedID int Foreign Key References ServicesPerformed(ServicesPerformedID),

Primary Key(LocationID, CustomerID, ServicesPerformedID));

Create table Estimate(

EstimateID int,

Date datetime not null,

Amount decimal(8,2) not null,

Primary Key(EstimateID),

BusinessConsultant int Foreign Key References BusinessConsultant(EmployeeID) not null,

CustomerID int Foreign Key References Customer(CustomerID) not null);

Create table Service(

ServiceID int,

Description varchar(45),

Cost decimal(8,2) not null,

Coverage varchar(45),

ClearanceRequired varchar(45),

Primary Key(ServiceID));

Create table EstimateHasService(

EstimateID int Foreign Key References Estimate(EstimateID),

ServiceID int Foreign Key References Service(ServiceID),

Primary Key(EstimateID, ServiceID));

Create table ServicesPerformedHasService(

ServicesPerformedID int Foreign Key References ServicesPerformed(ServicesPerformedID),

ServiceID int Foreign Key References Service(ServiceID),

Primary Key(ServicesPerformedID, ServiceID));

Create table TechnicalSkill(

TechnicalSkillID varchar(45),

Description varchar(45),

EmployeeID int Foreign Key References TechnicalConsultant(EmployeeID) not null,

Primary Key(TechnicalSkillID));

Create table BusinessExperience(

BusinessExperienceID int,

NumberOfYears int not null,

TypeOfBusiness varchar(45),

EmployeeID int Foreign Key References BusinessConsultant(EmployeeID) not null,

Primary Key(BusinessExperienceID));

Create table Degree(

DegreeID varchar(45),

Description varchar(100),

Primary Key(DegreeID));

Create table BusinessExperienceHasDegree(

BusinessExperienceID int Foreign Key References BusinessExperience(BusinessExperienceID),

DegreeID varchar(45) Foreign Key References Degree(DegreeID),

Primary Key(BusinessExperienceID, DegreeID));

Create table TechnicalConsultantHasDegree(

EmployeeID int Foreign Key References TechnicalConsultant(EmployeeID),

DegreeID varchar(45) Foreign Key References Degree(DegreeID),

Primary Key(EmployeeID, DegreeID));

## Insertion Statements

insert Service(ServiceId,Description,Cost,Coverage,ClearanceRequired) values (10,'Penetration Testing',1000,'Full','High');

insert Service(ServiceId,Description,Cost,Coverage,ClearanceRequired) values (11,'Physical Security',250,'Equipment','Low');

insert Service(ServiceId,Description,Cost,ClearanceRequired) values (12,'Email Phishing Campaign',100,'Medium');

insert Service(ServiceId,Description,Cost,ClearanceRequired) values (13,'Contingency Plan',2000,'High');

insert Service(ServiceId,Description,Cost,ClearanceRequired) values (14,'Denial of Service Testing',300,'Low');

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (1,'Chris','Evans','423 Square Drive','Fort Scott','KS','66701','667-324-6978','12/1/1988',31);

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (3,'Steve','Shire','542 Maple St.','Pittsburg','KS','66762','417-454-1243','3/30/1960',59);

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (5,'John','Stevens','586 Main Drive','Fort Scott','KS','66701','435-342-6543','1/12/1994',25);

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (2,'John','Snow','654 Game St','Cheyanne','WY','65464','435-342-1355','1/12/1994',25);

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (4,'Catnis','Everdeen','349 Hunger Road','Fort Scott','KS','66701','435-342-6546','3/10/1933',86);

insert Degree(DegreeID, Description) values ('CIS', 'Bachelor of Business Administration Computer Information Systems');

insert Degree(DegreeID, Description) values ('CS', 'Bachelor of Science');

insert Degree(DegreeID, Description) values ('ACCTG', 'Bachelor of Business Administration Accounting');

insert Degree(DegreeID, Description) values ('HRM', 'Bachelor of Psychology Human Resource Management');

insert Degree(DegreeID, Description) values ('MKTG', 'Bachelor of Business Administration Marketing');

insert BusinessConsultant(EmployeeID) values (3);

insert BusinessConsultant(EmployeeID) values (5);

insert BusinessConsultant(EmployeeID) values (1);

insert BusinessConsultant(EmployeeID) values (2);

insert BusinessConsultant(EmployeeID) values (4);

INSERT INTO [dbo].[BusinessExperience]

([BusinessExperienceID]

,[NumberOfYears]

,[TypeOfBusiness]

,[EmployeeID])

VALUES

(1

,3

,'Marketing'

,3);

INSERT INTO [dbo].[BusinessExperience]

([BusinessExperienceID]

,[NumberOfYears]

,[TypeOfBusiness]

,[EmployeeID])

VALUES

(2

,5

,'Security Consulting'

,5);

INSERT INTO [dbo].[BusinessExperience]

([BusinessExperienceID]

,[NumberOfYears]

,[TypeOfBusiness]

,[EmployeeID])

VALUES

(3

,10

,'Software Developer'

,1);

INSERT INTO [dbo].[BusinessExperience]

([BusinessExperienceID]

,[NumberOfYears]

,[TypeOfBusiness]

,[EmployeeID])

VALUES

(4

,7

,'Pentration Testing'

,2);

INSERT INTO [dbo].[BusinessExperience]

([BusinessExperienceID]

,[NumberOfYears]

,[TypeOfBusiness]

,[EmployeeID])

VALUES

(5

,1

,'Physical Security'

,4);

INSERT INTO BusinessExperienceHasDegree(BusinessExperienceID,DegreeID) values (1,'MKTG');

INSERT INTO BusinessExperienceHasDegree(BusinessExperienceID,DegreeID) values (2,'CIS');

INSERT INTO BusinessExperienceHasDegree(BusinessExperienceID,DegreeID) values (3,'CS');

INSERT INTO BusinessExperienceHasDegree(BusinessExperienceID,DegreeID) values (4,'CS');

INSERT INTO BusinessExperienceHasDegree(BusinessExperienceID,DegreeID) values (5,'ACCTG');

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (6,'Luke','Skywalker','445 Force Drive','Nevada','MO','64772','667-324-4568','12/1/1988',31);

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (7,'Harry','Potter','123 Staircase Avenue','Kansas City','KS','64101','417-454-4569','3/30/1960',59);

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (8,'Ron','Weesley','456 Rat Drive','Kansas City','KS','64102','435-342-4543','1/12/1994',25);

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (9,'John','Larrinson','544 Larry Avenue','Ballard','MO','64123','417-654-7998','3/12/1960',59);

insert Consultant(EmployeeID,FirstName,LastName,Street,City,State,ZipCode,Telephone,DateOfBirth,Age)

values (10,'Drako','Mouthful','456 Snake Street','Kansas City','KS','64102','435-342-4543','4/12/1994',25);

insert TechnicalConsultant(EmployeeID) values (6);

insert TechnicalConsultant(EmployeeID) values (7);

insert TechnicalConsultant(EmployeeID) values (8);

insert TechnicalConsultant(EmployeeID) values (9);

insert TechnicalConsultant(EmployeeID) values (10);

insert Customer(CustomerID,CompanyName, Street, City, State,

ZipCode, ContactName, ContactTitle, ContactTelephone,

BusinessType, NumberOfEmployees)

values

(10,'Peerless Products Inc.', '4320 Main St.', 'Fort Scott', 'KS', '66701', 'Steven Burn','IT Manager', '667-123-4654', 'Manufacturing', 400);

insert Customer(CustomerID,CompanyName, Street, City, State,

ZipCode, ContactName, ContactTitle, ContactTelephone,

BusinessType, NumberOfEmployees)

values

(11,'Pittsburg State University', '1701 S Broadway', 'Pittsburg', 'KS', '66762', 'Dr. Sha','CIS Professor', '667-123-3254', 'Education', 312);

insert Customer(CustomerID,CompanyName, Street, City, State,

ZipCode, ContactName, ContactTitle, ContactTelephone,

BusinessType, NumberOfEmployees)

values

(12,'McDonalds', '2342 Maple St.', 'Fort Scott', 'KS', '66701', 'Ronald McDonald','Marketing', '667-123-2345', 'Restaurant', 10);

insert Customer(CustomerID,CompanyName, Street, City, State,

ZipCode, ContactName, ContactTitle, ContactTelephone,

BusinessType, NumberOfEmployees)

values

(13,'Salon 9', '5843 1st Street', 'Pittsburg', 'KS', '66762', 'Sally Salington','Branch Manager', '667-345-3453', 'Cosmetology', 15);

insert Customer(CustomerID,CompanyName, Street, City, State,

ZipCode, ContactName, ContactTitle, ContactTelephone,

BusinessType, NumberOfEmployees)

values

(14,'Watko', '2345 Main St.', 'Pittsburg', 'KS', '66762', 'John Wattson','Plant Manager', '667-123-3453', 'Manufacturing', 345);

insert Location(CustomerID,LocationID, BuildingSize, Street, City, State, ZipCode, Telephone)

values (10,1,123,'4320 Main St.', 'Fort Scott', 'KS', '66701', '667-123-3453');

insert Location(CustomerID,LocationID, BuildingSize, Street, City, State, ZipCode, Telephone)

values (11,1,1245,'1701 S Broadway', 'Pittsburg', 'KS', '66762', '667-123-4567');

insert Location(CustomerID,LocationID, BuildingSize, Street, City, State, ZipCode, Telephone)

values (12,1,45668,'2342 Maple St.', 'Fort Scott', 'KS', '66701', '667-123-7569');

insert Location(CustomerID,LocationID, BuildingSize, Street, City, State, ZipCode, Telephone)

values (13,1,7894,'5843 1st Street', 'Pittsburg', 'KS', '66762', '667-123-7522');

insert Location(CustomerID,LocationID, BuildingSize, Street, City, State, ZipCode, Telephone)

values (14,1,4566,'2345 Main St.', 'Pittsburg', 'KS', '66762', '667-123-4452');

INSERT INTO Estimate(EstimateID,CustomerID,BusinessConsultant,Amount,Date) VALUES (1,10,1,2000,'11/30/2019');

INSERT INTO EstimateHasService(EstimateID,ServiceID) VALUES (1,10);

INSERT INTO EstimateHasService(EstimateID,ServiceID) VALUES (1,12);

INSERT INTO Estimate(EstimateID,CustomerID,BusinessConsultant,Amount,Date) VALUES (2,11,2,300,'11/3/2019');

INSERT INTO EstimateHasService(EstimateID,ServiceID) VALUES (2,11);

INSERT INTO Estimate(EstimateID,CustomerID,BusinessConsultant,Amount,Date) VALUES (3,12,3,3000,'12/6/2019');

INSERT INTO EstimateHasService(EstimateID,ServiceID) VALUES (3,13);

INSERT INTO Estimate(EstimateID,CustomerID,BusinessConsultant,Amount,Date) VALUES (4,13,4,700,'11/15/2019');

INSERT INTO EstimateHasService(EstimateID,ServiceID) VALUES (4,14);

INSERT INTO EstimateHasService(EstimateID,ServiceID) VALUES (4,11);

INSERT INTO Estimate(EstimateID,CustomerID,BusinessConsultant,Amount,Date) VALUES (5,14,5,3000,'12/5/2019');

INSERT INTO EstimateHasService(EstimateID,ServiceID) VALUES (5,13);

INSERT INTO ServicesPerformed(ServicesPerformedID,CustomerID,TechnicalConsultantID,Amount,Date) VALUES (1,10,6,2100,'11/30/2019');

INSERT INTO ServicesPerformedHasService(ServicesPerformedID,ServiceID) VALUES (1,10);

INSERT INTO ServicesPerformedHasService(ServicesPerformedID,ServiceID) VALUES (1,12);

INSERT INTO ServicesPerformed(ServicesPerformedID,CustomerID,TechnicalConsultantID,Amount,Date) VALUES (2,11,7,300,'11/5/2019');

INSERT INTO ServicesPerformedHasService(ServicesPerformedID,ServiceID) VALUES (2,11);

INSERT INTO ServicesPerformed(ServicesPerformedID,CustomerID,TechnicalConsultantID,Amount,Date) VALUES (3,12,8,3150.89,'12/8/2019');

INSERT INTO ServicesPerformedHasService(ServicesPerformedID,ServiceID) VALUES (3,13);

INSERT INTO ServicesPerformed(ServicesPerformedID,CustomerID,TechnicalConsultantID,Amount,Date) VALUES (4,13,9,700,'11/20/2019');

INSERT INTO ServicesPerformedHasService(ServicesPerformedID,ServiceID) VALUES (4,14);

INSERT INTO ServicesPerformedHasService(ServicesPerformedID,ServiceID) VALUES (4,11);

INSERT INTO ServicesPerformed(ServicesPerformedID,CustomerID,TechnicalConsultantID,Amount,Date) VALUES (5,14,10,3500,'12/10/2019');

INSERT INTO ServicesPerformedHasService(ServicesPerformedID,ServiceID) VALUES (5,13);

INSERT INTO TechnicalConsultantHasDegree(DegreeID,EmployeeID) VALUES ('ACCTG',6);

INSERT INTO TechnicalConsultantHasDegree(DegreeID,EmployeeID) VALUES ('CS',7);

INSERT INTO TechnicalConsultantHasDegree(DegreeID,EmployeeID) VALUES ('HRM',8);

INSERT INTO TechnicalConsultantHasDegree(DegreeID,EmployeeID) VALUES ('CIS',9);

INSERT INTO TechnicalConsultantHasDegree(DegreeID,EmployeeID) VALUES ('MKTG',10);

INSERT INTO TechnicalSkill(TechnicalSkillID,EmployeeID,Description) VALUES ('PEN TEST',6,'Pentration Testing');

INSERT INTO TechnicalSkill(TechnicalSkillID,EmployeeID,Description) VALUES ('CODE',7,'Software Development');

INSERT INTO TechnicalSkill(TechnicalSkillID,EmployeeID,Description) VALUES ('SALES',8,'Sales & Marketing');

INSERT INTO TechnicalSkill(TechnicalSkillID,EmployeeID,Description) VALUES ('PHYSC',9,'Physical Security');

INSERT INTO TechnicalSkill(TechnicalSkillID,EmployeeID,Description) VALUES ('BACKUP',10,'Contingency Planning & Backup');

INSERT INTO LocationHasServicesPerformed(ServicesPerformedID,CustomerID,LocationID) VALUES (1,10,1);

INSERT INTO LocationHasServicesPerformed(ServicesPerformedID,CustomerID,LocationID) VALUES (2,11,1);

INSERT INTO LocationHasServicesPerformed(ServicesPerformedID,CustomerID,LocationID) VALUES (3,12,1);

INSERT INTO LocationHasServicesPerformed(ServicesPerformedID,CustomerID,LocationID) VALUES (4,13,1);

INSERT INTO LocationHasServicesPerformed(ServicesPerformedID,CustomerID,LocationID) VALUES (5,14,1);

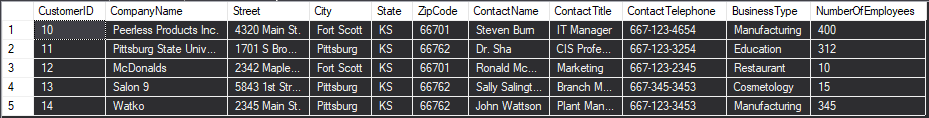
# SQL Statements

Below are some examples of the sql queries we are doing in the program.

## Get All Customers

select \* from Customer

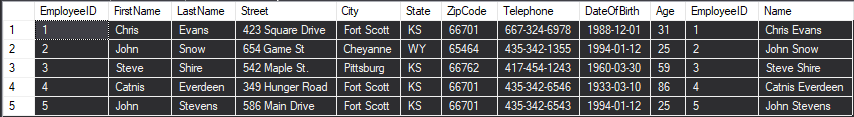
### Results



## Get Business Consultants

select \*,(FirstName + ' ' + LastName) as Name from Consultant c, BusinessConsultant bc where bc.EmployeeID = c.EmployeeID

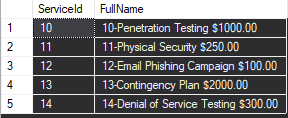
### Results



## Get Services

select ServiceId, (CONVERT(varchar(10), ServiceId) + '-' + Description + ' $' + Convert(varchar(10),Cost)) as FullName from Service

### Results



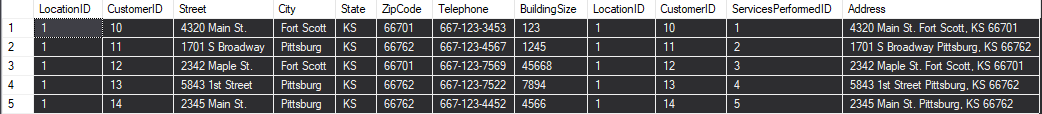
## Get Addresses

select \*,(Street + ' ' + City + ', ' + State + ' ' + ZipCode) as Address

from Location l, LocationHasServicesPerformed lsp

where lsp.LocationID = l.LocationID and l.CustomerID = lsp.CustomerID

### Results



## Get Services for an Estimate

select s.ServiceId, (CONVERT(varchar(10), s.ServiceId) + '-' + Description + ' $' + Convert(varchar(10),Cost)) as FullName

from Service s, EstimateHasService ehs

where s.ServiceId = ehs.ServiceID and ehs.EstimateID = @EstimateID

### Results

declare @estimateId as int = 1;



The parameters used in these queries is below:

declare @From as datetime = '12/5/2019';

declare @To as datetime = '3/5/2020';

## Get Sales Data

select c.CompanyName, (co.FirstName + ' ' + co.LastName) as TechnicalConsultant, sp.Date, Sum(sp.Amount) as Sales, Sum(s.Cost) as Cost, Sum(sp.Amount - s.Cost) as Profit

from ServicesPerformed sp left outer join

Customer c on sp.CustomerID = c.CustomerID left outer join

Consultant co on sp.TechnicalConsultantID = co.EmployeeID left outer join

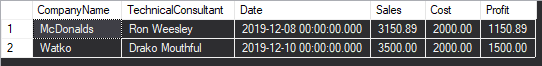
ServicesPerformedHasService sps on sps.ServicesPerformedID = sp.ServicesPerformedID left outer join

Service s on s.ServiceID = sps.ServiceID

where Date >= @From and Date < @To

group by c.CompanyName, co.FirstName, co.LastName, sp.Date

### Results



## Get Sales Summary

select Sum(sp.Amount) as Sales, Sum(s.Cost) as Cost, Sum(sp.Amount - s.Cost) as Profit

from ServicesPerformed sp left outer join

ServicesPerformedHasService sps on sps.ServicesPerformedID = sp.ServicesPerformedID left outer join

Service s on s.ServiceID = sps.ServiceID

where Date >= @From and Date < @To

### Results



## Get Expected Sales Summary

select Sum(sp.Amount) as Sales, Sum(s.Cost) as Cost, Sum(sp.Amount - s.Cost) as Profit

from Estimate sp left outer join

EstimateHasService sps on sps.EstimateID = sp.EstimateID left outer join

Service s on s.ServiceID = sps.ServiceID

where Date >= @From and Date < @To

### Results



# Data Dictionary

If the column is RED, then it is a foreign key. If the column is YELLOW, then it is a primary key. If the column is PURPLE, then it is both a primary key and a foreign key.

## Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Description** | **Constraints** | **Example** |
| Customer \_ID | INT | Primary Key | Unique, Required | 1 |
| CompanyName | VARCHAR (45) | Company Name | Required | McDonalds |
| Street | VARCHAR (45) | Street Address |  | 471 Main St. |
| City | VARCHAR (45) | City Name |  | Pittsburg |
| State | CHAR (2) | State Name |  | KS |
| ZipCode | VARCHAR (45) | Place Zip Code |  | 66701 |
| ContactName | VARCHAR (45) | Customer Contact Name | Required | Steve Shives |
| ContactTelephone | VARCHAR (45) | Customer Telephone Number | Required | 417-234-1234 |
| ContactTitle | VARCHAR (45) | Customer Title | Required | Developer |
| BusinessType | VARCHAR (45) | Company Business Type |  | Restaurant |
| NumberOfEmployees | INT | Company Number of Employees | Number > 0 | 10 |

## ServicesPerformed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data type** | **Description** | **Constraints** | **Example** |
| ServicePerformedID | INT | Primary Key | Unique, Required | 1 |
| Date | DATETIME | Service Performed date | Required | 12/4/19 |
| Amount | DECIMAL(8,2) | Cost for the Service | Required, Amount > 0 | 1000 |
| CutomerID | INT | Customer ID from Customer table | Required | 12 |
| TechnicalConsultantID | INT | Employee ID for Technical Consultant | Required | 13 |

## LocationHasServicesPerformed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| Location | INT | Location where the services performed | Required | 12 |
| CustomerID | INT | Customer Unique Number | Unique, Required | 1 |
| ServicePerformedID | INT | Service Performed Unique Number | Unique, Required | 12 |

## ServicesPerformedHasService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| ServicePerformedID | INT | Service Performed Unique Number | Unique, Required | 12 |
| ServiceID | INT | Service Unique Number | Unique, Required | 1 |

## EstimateHasService

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| EstimateID | INT | Estimate of service Unique Number | Unique, Required | 12 |
| ServiceID | INT | Service Unique Number | Unique, Required | 1 |

## BusinessConsultant

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| EmployeeID | INT | Employee Unique Number | Unique, Required | 1 |
|  |  |  |  |  |

## TechnicalConsultant

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| EmployeeID | INT | Employee Unique Number | Unique, Required | 1 |

## TechnicalConsultantHasDegree

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| EmployeeID | INT | Employee Unique Number | Unique, Required | 1 |
| DegreeID | VARCHAR (45) | Qualification for the Technical Consultant | Unique, Required | 12 |

## Degree

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| DegreeID | VARCHAR (45) | Degree Unique Number | Unique, Required | 1 |
| Description | VARCHAR (100) | Brief description for academic degree. | Required | Bachelors |

## BusinessExperienceHasDegree

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| BusinessExperienceID | INT | Unique identifier for employees business experience | Required | 3 |
| DegreeID | VARCHAR (45) | Unique abbreviation for academic degree | Unique, Required | CIS |

## Location

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data type | Description | Constraints | Example |
| LocationID | INT | Primary Key | Unique, Required | 12 |
| CustomerID | INT | Foreign Key | Unique, Required | 1 |
| Street | VARCHAR(45) | Street Name | Required | S Broadway St. |
| City | VARCHAR(45) | City Name | Required | Pittsburg |
| State | CHAR(2) | State Abbreviation | Required | Kansas |
| ZipCode | VARCHAR(15) | Zip Code Number | Required | 66762 |
| Telephone | VARCHAR(45) | Telephone Number | Required | 417-232-231 |
| BuildingSize | INT | Square footage of the building | Required | 1000 sqft |

## Service

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Description | Constraints | Example |
| ServiceID | INT | Primary Key | Unique, Required | 1 |
| Description | VARCHAR(45) | Service Description | Required | Penetration Testing |
| Cost | DECIMAL(8,2) | Cost of the Service | Required | 1200.00 |
| Coverage | VARCHAR(45) | Assets which are covered by the service | Required | Equipment |
| ClearanceRequired | VARCHAR(45) | Clearance required for Service | Required | High |

## BusinessExperience

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Description | Constraints | Example |
| BusinessExperienceID | INT | Primary Key for employee’s business experience | Unique, Required | 1 |
| NumberOfYears | INT | Number of years Experience | Required | 1 |
| TypeOfBusiness | VARCHAR(45) | Type of Business | Required | Restaurant |
| EmployeeID | INT | Employee ID | Unique, Required | 1 |

## Consultant

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Description | Constraints | Example |
| EmployeeID | INT | Primary Key | Unique, Required | 1 |
| FirstName | VARCHAR(45) | Person First Name | Required | Harry |
| LastName | VARCHAR(45) | Person Last Name | Required | Potter |
| Street | VARCHAR(45) | Street Name | Required | English St. |
| City | VARCHAR(45) | City Name | Required | Pittsburg |
| State | CHAR(2) | State Abbreviation | Required | Kansas |
| ZipCode | VARCHAR(45) | Zip Code Number | Required | 66762 |
| Telephone | VARCHAR(45) | Telephone Number | Required | 417-342-2145 |
| DateOfBirth | Date | Person’s Date of Birth | Required | 12/2/2019 |
| Age | INT | Age of Person | Required | 22 |

## TechnicalSkill

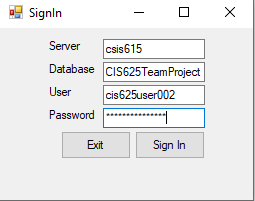
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Description | Constraints | Example |
| TechnicalSkillID | VARCHAR(45) | Primary Key Technical Skill ID Number | Unique, Required | 1 |
| Description | VARCHAR(45) | Description of technical skill | Required | Software Development |
| EmployeeID | INT | Employee ID Number | Unique, Required | 1 |

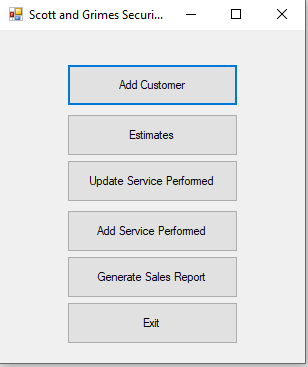
## Estimate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Description | Constraints | Example |
| EstimateID | INT | Primary Key Estimate ID Number | Unique, Required | 1 |
| Date | DATETIME | Date of the estimate | Required | 11/30/2019 |
| Amount | DECIMAL(8,2) | Amount of the estimate in USD | Required | 1200.00 |
| BusinessConsultant | INT | Business consultant ID which conducted the estimate | Required | 1 |
| CustomerID | INT | Unique identifier for which customer estimate was for | Required | 10 |

# User Manual

## Sign in Form

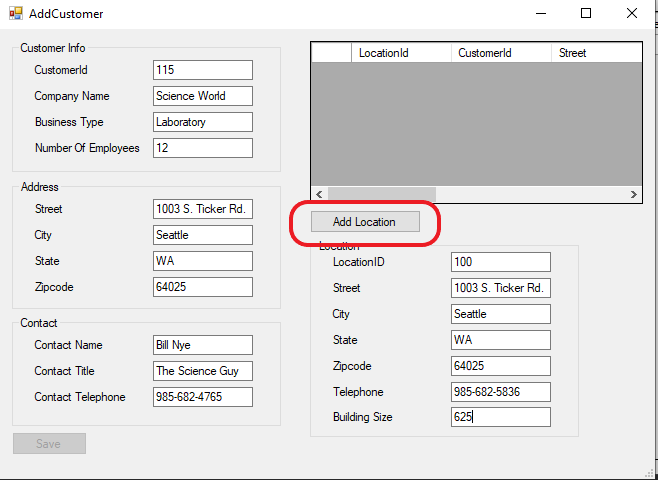
User enters their information to sign in. Once the user enters their information, they click the button to Sign In. If the user wants to exit the application click the button Exit.

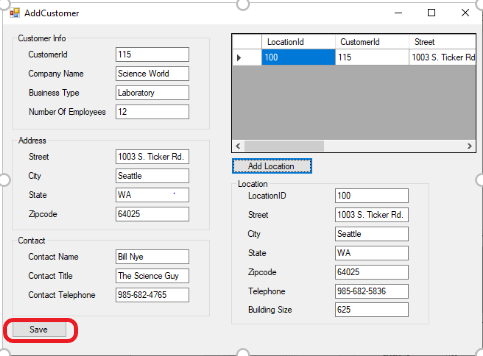


## Main Form

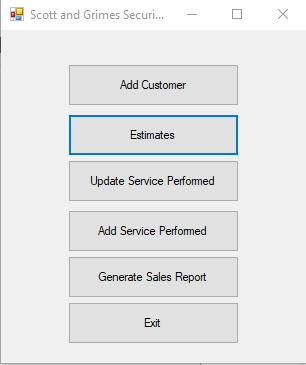
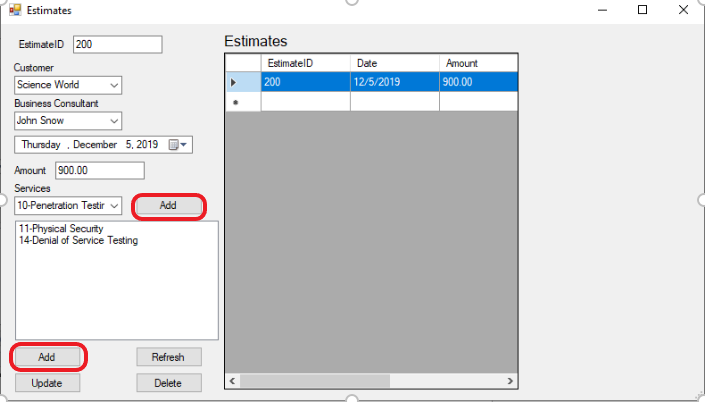
This is the main form. The main form allows employees to perform various tasks like adding a customer, view estimates, update and add services performed, generate sales report, and exit the application.

## Add Customer Form

The user will add Customer Information, Address Information, Contact Information, and Location Information. Once, the information is typed in the user will press the button Add Location.



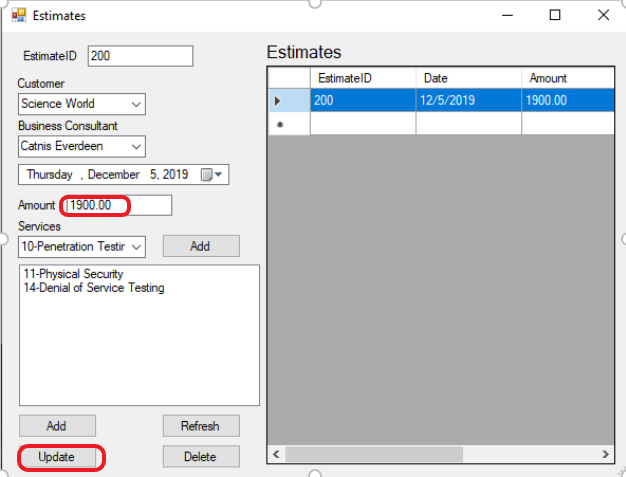
After the location is added, the user will be able to see the Location Information in the view. Then the user will press the button Save to add a customer.

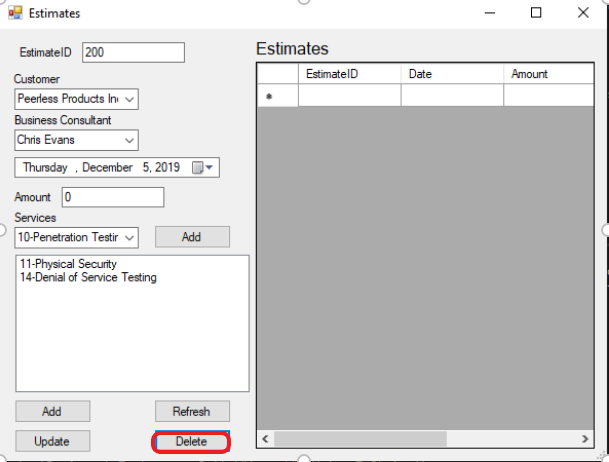


## Estimates Form

To add an estimate the user will need to enter the Estimate information which includes: EstimateID, selecting Customer and Business Consultant, Amount, and selecting the Services that were performed. To select Services the user will need to select the add button next to the Services drop down for each service found in the Services drop down. Once the information is given the user will press the Add button. The user will be able to view the estimate that was just added in the view on the right.

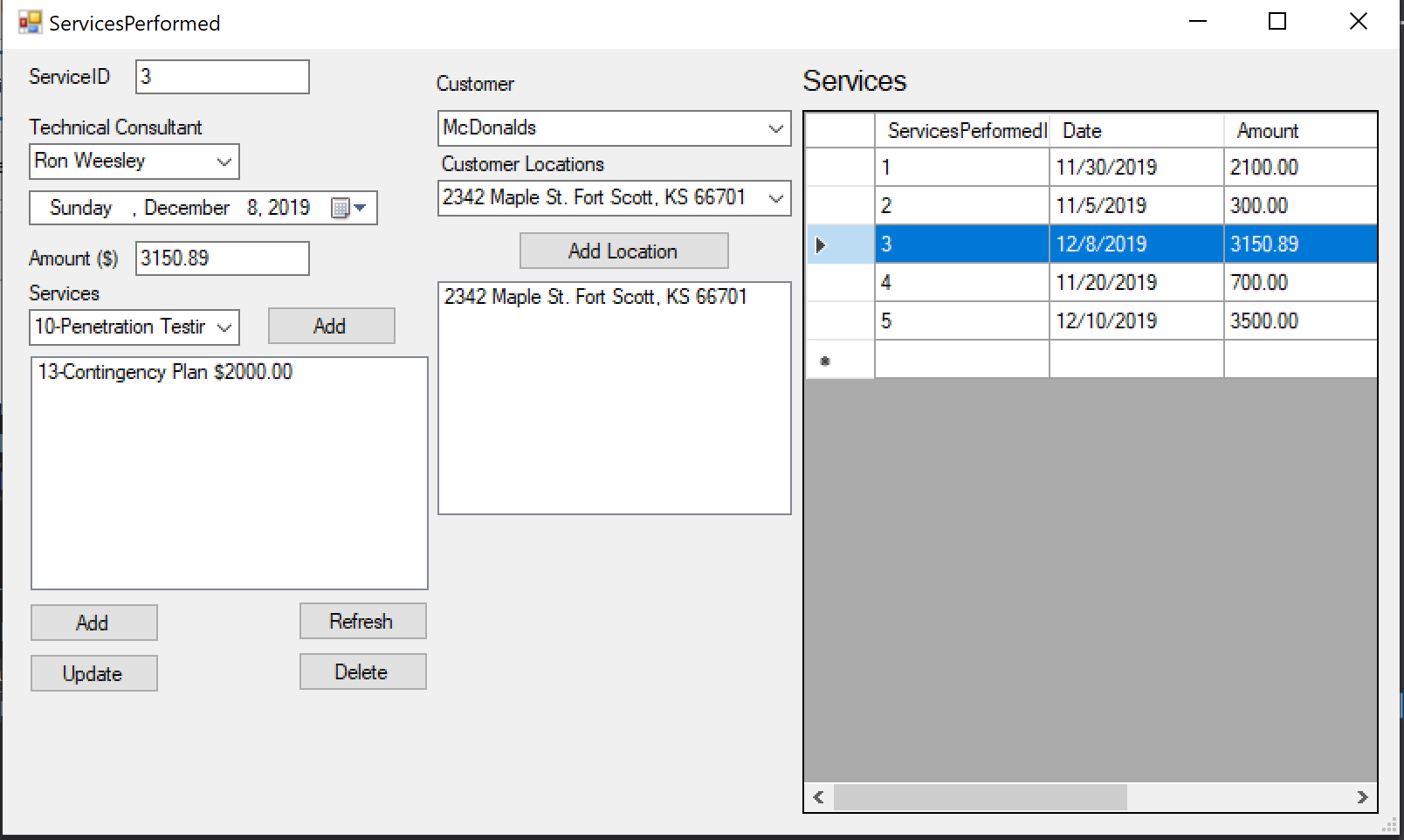
To view, add, update, and delete estimates press the button Estimates on the homepage.

If the user needs to update the Amount for the estimate the user will select the estimate that they would like to update. The information will populate on the right. The user then adds the new Amount for the estimate. After, the user will press the Update button. The new updated amount will show in the view on the right.

If an estimate needs to be deleted. The user will select the estimate in the view that is on the right. Once, the information populates on the left the user will select the Delete button. The deleted estimate then will not be seen in the view on the right.

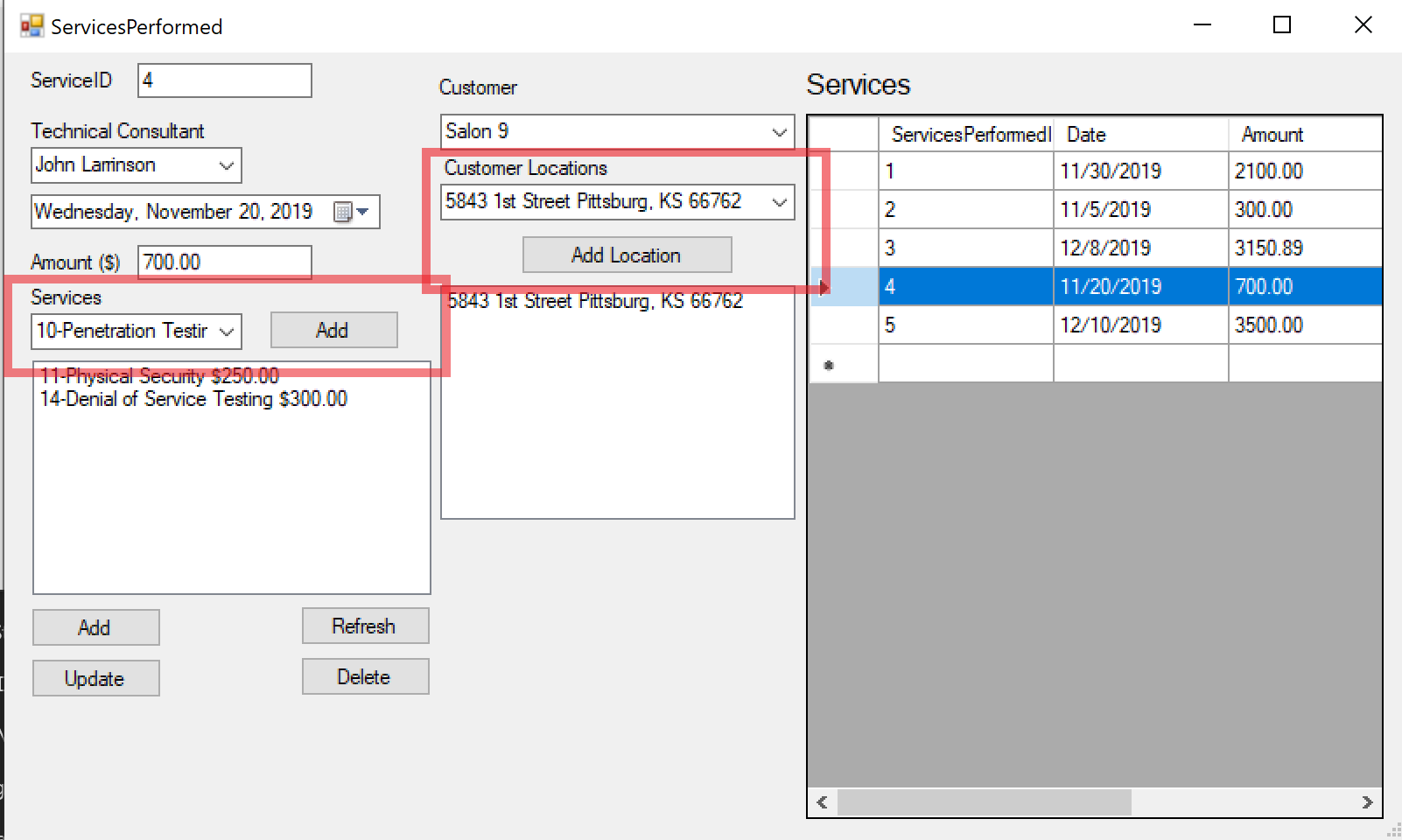
## Services Performed Form

The services performed form allows you to enter in services that have been performed by consultants. Services performed allow you to put in the service id, the services performed, the consultant that did the services, the cost of the services performed, the customer, and the locations at which these services were performed.



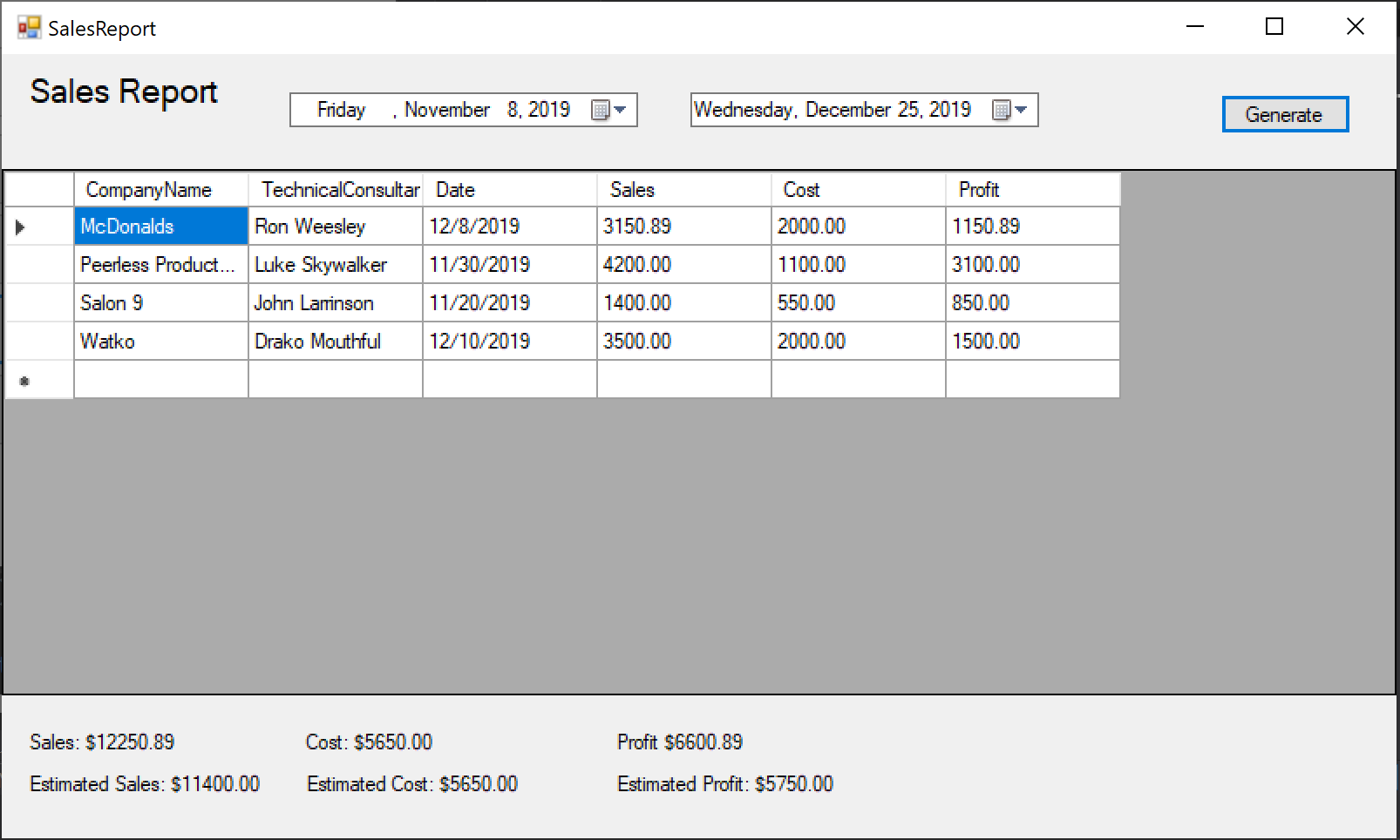
The services performed form functions like the estimates form. To delete or update a services performed entry, simple click to the left of the row you want to change in the services grid. The information on the right will update accordingly. You can then click the update/delete button.

To add services and locations to the services performed entry, you can select a service/location from the drop downs and then click add. In most cases, customers will have only one location.

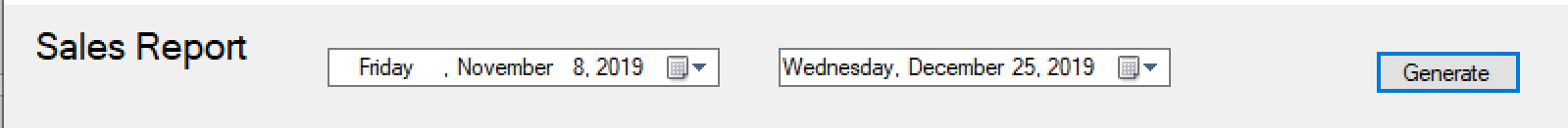


## Sales Report Form

Using the data we entered in the above forms, we are able to calculate total sales between two dates, the cost of the services performed, and the profit made. Additionally, we can view what the expected sales, cost, and profit was going to be based on the estimates we created.



By default, when you open the report it will show the last 30 days of services performed. To change the range, just change the dates above the report. This will get all the sales from the first date to the second date. To update the report once you change the dates, click on the generate button.



Below the report you can see summary information about the report. This gives you sales, cost, and profit for actual and estimated. From this report you can see that profit was $6600.89, but we expected it to be $5750.00. This means that we did better than what we expected. Notice that if estimates and services performed have the same services, the actual and estimated cost will be the same.

