Project Title

<Student Names> INLS X23 – 00X

2022-04-05

Table of Contents

Table of Contents

[Problem Description 3](#_Toc5028359)

[Requirements Gathering 3](#_Toc5028360)

[Entity Relationship Model Diagram 4](#_Toc5028361)

[Discussion 4](#_Toc5028362)

[Relationship Sentences 4](#_Toc5028363)

[Normalized Relations and Functional Dependencies 5](#_Toc5028364)

[Customer Relation 5](#_Toc5028365)

[SalonService Relation 7](#_Toc5028366)

[Employee Relation 8](#_Toc5028367)

[Appointment Relation 9](#_Toc5028368)

[ServiceRendered Relation 10](#_Toc5028369)

[Relational Model 11](#_Toc5028370)

[Discussion 11](#_Toc5028371)

[SQL DDL 12](#_Toc5028372)

[Entities and Primary Keys: 12](#_Toc5028373)

[Foreign Keys and Other Notable Changes: 14](#_Toc5028374)

[Adding Data to the Tables using SQL INSERT Statements 15](#_Toc5028375)

[Discussion: 17](#_Toc5028376)

[Application Implementation 18](#_Toc5028377)

[Navigation Form 18](#_Toc5028378)

[Customer Data Entry Form 18](#_Toc5028379)

[Customer Appointments Totals Report 19](#_Toc5028380)

[Services and Discounts Report 20](#_Toc5028381)

[Conclusion 21](#_Toc5028382)

# Problem Description

Our company has owned and operated a hair and nail salon in midtown Manhattan for 7 years. We have been using spreadsheets and a paper logbook to keep track of customers, appointments and payments. We would like to replace this manual method of tracking the business with a database.

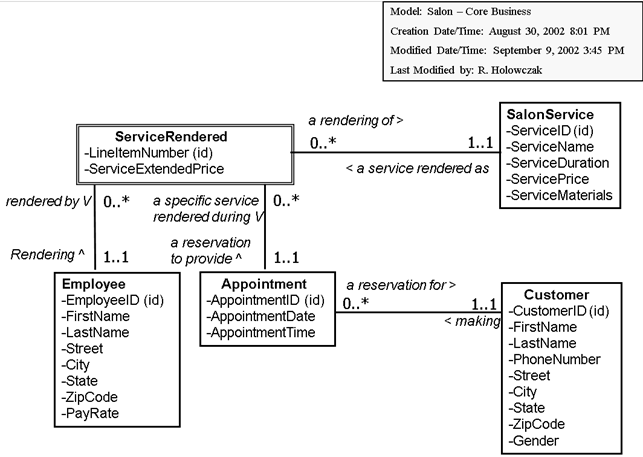
In our business, Customers make Appointments with their favorite hair stylist (folks who cut and style customers’ hair) or other employee and can indulge in a number of Services such as basic hair cut/styling, hair color, highlights, perm, facial, manicure, pedicure, etc. We need to keep track of materials (shampoo, hair color) and the time it will take to complete each service. While each service has a standard price, promotions and other factors may affect the actual price extended to the Customer for the given service.

We also need to keep track of our Employees including their home address, contact information and rate of pay. We need to keep track of each Customer’s contact information as well as their gender. For Appointments we need to know the date and time of the appointment and which customer the appointment is for.

# Requirements Gathering

This section contains methods of requirements gathering such as reviewing current processes, paper notes, and Excel spreadsheets excreta. Additional methods are encourages if others are involved, like shadowing, interviews, and questionnaires.

# Entity Relationship Model Diagram



## User Stories (3)

*Daphne and Fred was getting ready for a big mystery and wanted to look their best. Each called the salon to book an appointment. Fred scheduled a 10-minute wash (shampoo & condition) and dry with Bubbles Puff that costs $15 and 20 minute trim with Edward Scissorhands for $30.00 on April 1, 2021 @ 11:15am. Daphne also booked a 10-minute wash and dry with Bubbles Puff, highlights which take 120 minutes for $120, and a Brazilian blowout by Aang costing $300 and takes another hour for the same date, but at 8am.*

## Discussion

What we like about this model:

* All relationship lines are going in a horizontal or vertical position. No lines are crossed.
* Attribute names are spelled out with no spaces in the names. No abbreviations are used.
* Each relationship has two very phrases from which we can make relationship sentences (see next section).
* The diagram also has a “legend” in the upper–right corner so we can tell what the diagram represents and who last modified the diagram.

## Relationship Sentences

* One **Customer**   may be   making one or more **Appointments**
* One **Appointment**  must be  a reservation for one and only one **Customer**
* One **SalonService**  may be  a service rendered as one or more **ServiceRendered**
* One **ServiceRendered**  must be  a rendering of one and only one **SalonService**
* One **Employee**  may be  rendering one or more **ServiceRendered**
* One **ServiceRendered**  must be  rendered by one and only one **Employee**
* One **Appointment**  may be  a reservation to provide one or more **ServiceRendered**
* One **ServiceRendered**  must be  a specific service rendered during one and only one **Appointment**

# Normalized Relations and Functional Dependencies

SAMPLE DATA IS OPTIONAL

The next step is to [Normalize the Relations](https://holowczak.com/database-normalization/).

The steps to follow for each relation are:

1. Write out the relation including all attribute names. Indicate keys and foreign keys.
2. Provide some sample data for the relation.
3. State the **Key** for the relation and write down all **Functional Dependencies**.
4. Go through the definitions of each normal form starting with 1NF and going up to BCNF (or 3NF depending on your project requirements).
5. If a relation meets the definition of a normal form, move up to the next higher normal form.
6. If a relation fails to meet the definition of a normal form (e.g., it contains a partial-key dependency or it contains a transitive dependency), then split the relation into two new relations.  
   Begin the normalization process from the beginning with each of these two new relations.

### Customer Relation

**Customer ( CustomerID (key) , FirstName, LastName, CustPhone, Street, City, State, ZipCode, Gender )**

*Sample Data*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CustomerID** | **FirstName** | **LastName** | **PhoneNumber** | **Street** | **City** | **State** | **ZipCode** | **Gender** |
| C101 | Elia | Fawcett | 201-222-2222 | 8989 Smith Rd | Garfield | NJ | 07026 | F |
| C102 | Ishwarya | Roberts | 201-222-3333 | 65 Hope Rd | Garfield | NJ | 07026 | M |
| C103 | Frederic | Fawcett | 201-222-2222 | 8989 Smith Rd | Garfield | NJ | 07026 | M |
| C104 | Goldie | Montand | 201-222-4321 | 5235 Ironwood Ln | Garfield | NJ | 07026 | F |
| C105 | Dheeraj | Alexander | 201-222-4545 | 666 22nd Ave | Bergenfield | NJ | 07621 | M |
| C106 | Josie | Davis | 201-333-6789 | 4200 Bluejay Ave | Bergenfield | NJ | 07621 | F |
| C107 | Faye | Glenn | 201-333-4242 | 1522 Main St | Cliffside Park | NJ | 07010 | F |
| C108 | Lauren | Hershey | 201-444-1313 | 2360 Maxon Rd | Englewood | NJ | 07631 | F |

Key: CustomerID

FD1: CustomerID -> FirstName, LastName, PhoneNumber, Street, City, State, ZipCode, Gender

FD2: ZipCode -> City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency exists: CustomerID -> ZipCode and ZipCode -> City, State

Solution: Split Customer relation into two new relations named CustomerData and ZipCodes:

CustomerData (CustomerID (key), FirstName, LastName, CustPhone, Street, ZipCode (fk), Gender )

Key: CustomerID

FD1: CustomerID -> FirstName, LastName, PhoneNumber, Street, ZipCode (fk), Gender

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

ZipCodes( ZipCode (key), City, State)

Key: ZipCode

FD1: ZipCode -> City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

### SalonService Relation

**SalonService ( ServiceID (key), ServiceName, ServiceDuration, ServicePrice, ServiceMaterials )**

Sample Data:

|  |  |  |  |
| --- | --- | --- | --- |
| **ServiceID** | **ServiceDuration** | **ServicePrice** | **ServiceMaterials** |
| SV101 | Men’s Haircut | 20 | 22.00 | None |
| SV102 | Women’s Haircut | 30 | 32.00 | None |
| SV103 | Child Haircut | 20 | 15.00 | None |
| SV104 | Women’s Hair Color | 60 | 76.00 | Color, Reagent, Gloves, Reagent Brush, Foil |
| SV105 | Women’s Hair Style | 45 | 56.00 | Shampoo, Conditioner |
| SV106 | Men’s Hair Style | 45 | 46.00 | Shampoo, Conditioner |

Key: ServiceID

FD1: ServiceID -> ServiceName, ServiceDuration, ServicePrice, ServiceMaterials

1NF: ServiceMaterials may be treated as a multi-valued attribute. In this case SalonService is not in 1NF.

Solution: Split out ServiceMaterials into a separate relation.

For this example, however we will keep ServiceMaterials as an attribute of the SalonService relation.

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

### Employee Relation

**Employee( EmployeeID (key), FirstName, LastName, Street, City, State, ZipCode, PayRate )**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EmployeeID** | **FirstName** | **LastName** | **Street** | **City** | **State** | **ZipCode** | **PayRate** |
| E300 | Joy | Aveda | 46 Easton Ave. | Garfield | NJ | 07026 | 18.00 |
| E400 | Geraldo | Geraldo | 12 Fortis Blvd. Apt. 2A | Garfield | NJ | 07026 | 22.00 |
| E500 | Koy | Petruzzio | 70 Wilard St. | Garfield | NJ | 07026 | 20.00 |
| E600 | Landry | Monroe | 73 Holly Terrace | Cliffside Park | NJ | 07010 | 18.00 |
| E700 | Pat | Reese | 2 Lincoln Place | Cliffside Park | NJ | 07010 | 23.00 |
| E800 | Winter | Tanner | 215 Elm Ave | Teaneck | NJ | 07665 | 23.00 |

Key: EmployeeID

FD1: EmployeeID -> FirstName, LastName, Street, City, State, ZipCode, PayRate

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency exists: EmployeeID -> ZipCode and ZipCode -> City, State

Solution: Split Customer relation into two new relations named EmployeeData and ZipCodes:

EmployeeData(EmployeeID (key), FirstName, LastName, Street, ZipCode (fk), PayRate )

Key: EmployeeID

FD1: EmployeeID -> FirstName, LastName, Street, ZipCode (fk), PayRate

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependencies

BCNF: All determinants are candidate keys

Note: We already have a ZipCodes relation from when the Customer relation was split up. So we re-use that ZipCodes relation. There is no need to create a second ZipCodes relation.

### Appointment Relation

**Appointment ( AppointmentID (key), AppointmentDate, AppotinmentTime, CustomerID (fk) )**

Sample Data:

|  |  |  |  |
| --- | --- | --- | --- |
| **AppointmentID** | **AppointmentDate** | **AppointmentTime** | **CustomerID** |
| A400 | 10/22/2017 | 11:00:00 AM | C101 |
| A401 | 11/6/2017 | 12:45:00 PM | C102 |
| A402 | 12/7/2017 | 2:00:00 PM | C106 |
| A403 | 12/18/2017 | 3:30:00 PM | C106 |
| A404 | 12/21/2017 | 11:30:00 AM | C108 |
| A405 | 12/31/2017 | 10:00:00 AM | C107 |
| A406 | 1/11/2018 | 3:15:00 PM | C103 |
| A407 | 1/12/2018 | 2:30:00 PM | C104 |
| A408 | 1/22/2018 | 4:00:00 PM | C105 |

Key: AppointmentID

FD1: AppointmentID -> AppointmentDate, AppotinmentTime, CustomerID (fk)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

### ServiceRendered Relation

**ServiceRendered ( AppointmentID (fk)(key), LineItemNumber(key), ServiceID (fk), ServiceExtendedPrice, EmployeeID(fk) )**

Sample Data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AppointmentID** | **LineItemNumber** | **ServiceID** | **ServiceExtendedPrice** | **EmployeeID** |
| A400 | 1 | SV104 | 75.00 | E400 |
| A400 | 2 | SV102 | 25.00 | E400 |
| A401 | 1 | SV101 | 22.00 | E500 |
| A402 | 1 | SV104 | 75.00 | E600 |
| A402 | 2 | SV102 | 30.00 | E800 |
| A403 | 1 | SV105 | 50.00 | E300 |
| A404 | 1 | SV105 | 55.00 | E300 |
| A405 | 1 | SV102 | 30.00 | E700 |
| A405 | 2 | SV104 | 70.00 | E700 |
| A405 | 3 | SV105 | 50.00 | E700 |

Key: AppointmentID, LineItemNumber

FD1: AppointmentID, LineItemNumber -> ServiceID (fk), ServiceExtendedPrice, EmployeeID(fk)

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

## Relational Model

**Customer** ( CustomerID (key) , FirstName, LastName, PhoneNumber, Street, ZipCode (fk), Gender )

**ZipCodes** ( ZipCode (key), City, State)

**SalonService** ( ServiceID (key), ServiceName, ServiceDuration, ServicePrice, ServiceMaterials )

**Employee** ( EmployeeID (key), FirstName, LastName, Street, ZipCode (fk), PayRate )

**Appointment** ( AppointmentID (key), AppointmentDate, AppotinmentTime, CustomerID (fk) )

**ServiceRendered** ( AppointmentID (fk)(key), LineItemNumber(key), ServiceID (fk), ServiceExtendedPrice, EmployeeID(fk))

## Discussion

Talk about the normalization steps and challenges.

# SQL DDL

## Entities and Primary Keys:

Create a table in the database for each of the relations in the final set of relations.

The following SQL code creates the six tables and adds the PRIMARY KEY constraint to each one:

CREATE TABLE ZipCodes

(

zipcode VARCHAR(12) NOT NULL,

city VARCHAR(36),

state VARCHAR(4),

CONSTRAINT pk\_zipcodes

PRIMARY KEY (zipcode)

)

CREATE TABLE Customer

(

CustomerID VARCHAR(10) NOT NULL,

FirstName VARCHAR(35),

LastName VARCHAR(35),

PhoneNumber VARCHAR(15),

Street VARCHAR(35),

ZipCode VARCHAR(12),

Gender VARCHAR(2),

CONSTRAINT pk\_customer

PRIMARY KEY (CustomerID)

)

CREATE TABLE Appointment

(

AppointmentID VARCHAR(10) NOT NULL,

AppointmentDateTime DATE,

CustomerID VARCHAR(10) NOT NULL,

CONSTRAINT pk\_appointment

PRIMARY KEY (AppointmentID)

)

CREATE TABLE SalonService

(

ServiceID VARCHAR(10) NOT NULL,

ServiceName VARCHAR(35),

ServiceDuration INTEGER,

ServicePrice NUMBER,

ServiceMaterials VARCHAR(255),

CONSTRAINT pk\_salonservice

PRIMARY KEY (ServiceID)

)

CREATE TABLE Employee

(

EmployeeID VARCHAR(10) NOT NULL,

FirstName VARCHAR(35),

LastName VARCHAR(25),

Street VARCHAR(45),

ZipCode VARCHAR(12),

PayRate NUMBER,

CONSTRAINT pk\_employee

PRIMARY KEY (EmployeeID)

)

CREATE TABLE ServiceRendered

(

AppointmentID VARCHAR(10) NOT NULL,

LineItemNumber INTEGER NOT NULL,

ServiceID VARCHAR(10) NOT NULL,

ServiceExtendedPrice NUMBER,

EmployeeID VARCHAR(10) NOT NULL,

CONSTRAINT pk\_ServiceRendered

PRIMARY KEY (AppointmentID, LineItemNumber)

)

## Foreign Keys and Other Notable Changes:

The following SQL codes add FOREIGN KEY constraints to link the tables together:

ALTER TABLE Customer

ADD CONSTRAINT fk\_customer\_zipcodes

FOREIGN KEY (ZipCode)

REFERENCES ZipCodes (ZipCode)

ALTER TABLE Employee

ADD CONSTRAINT fk\_employee\_zipcodes

FOREIGN KEY (ZipCode)

REFERENCES ZipCodes (ZipCode)

ALTER TABLE Appointment

ADD CONSTRAINT fk\_customer\_appointment

FOREIGN KEY (CustomerID)

REFERENCES Customer (CustomerID)

ALTER TABLE ServiceRendered

ADD CONSTRAINT fk\_ServiceRendered\_Service

FOREIGN KEY (ServiceID)

REFERENCES SalonService (ServiceID)

ALTER TABLE ServiceRendered

ADD CONSTRAINT fk\_ServiceRendered\_Employee

FOREIGN KEY (EmployeeID)

REFERENCES Employee (EmployeeID)

ALTER TABLE ServiceRendered

ADD CONSTRAINT fk\_ServiceRendered\_Appointment

FOREIGN KEY (AppointmentID)

REFERENCES Appointment (AppointmentID)

## Adding Data to the Tables using SQL INSERT Statements

INSERT INTO ZipCodes VALUES ('07026', 'Garfield', 'NJ');

INSERT INTO ZipCodes VALUES ('07621', 'Bergenfield', 'NJ');

INSERT INTO ZipCodes VALUES ('07010', 'Cliffside Park', 'NJ');

INSERT INTO ZipCodes VALUES ('07631', 'Englewood', 'NJ');

INSERT INTO ZipCodes VALUES ('07665', 'Teaneck', 'NJ');

INSERT INTO Customer VALUES ('C101', 'Elia', 'Fawcett', '201-222-2222', '8989 Smith Rd', '07026', 'F');

INSERT INTO Customer VALUES ('C102', 'Ishwarya', 'Roberts', '201-222-3333', '65 Hope Rd', '07026', 'M');

INSERT INTO Customer VALUES ('C103', 'Frederic', 'Fawcett', '201-222-2222', '8989 Smith Rd', '07026', 'M');

INSERT INTO Customer VALUES ('C104', 'Goldie', 'Montand', '201-222-4321', '5235 Ironwood Ln', '07026', 'F');

INSERT INTO Customer VALUES ('C105', 'Dheeraj', 'Alexander', '201-222-4545', '666 22nd Ave', '07621', 'M');

INSERT INTO Customer VALUES ('C106', 'Josie', 'Davis', '201-333-6789', '4200 Bluejay Ave', '07621', 'F');

INSERT INTO Customer VALUES ('C107', 'Faye', 'Glenn', '201-333-4242', '1522 Main St', '07010', 'F');

INSERT INTO Customer VALUES ('C108', 'Lauren', 'Hershey', '201-444-1313', '2360 Maxon Rd', '07631', 'F');

INSERT INTO SalonService VALUES ('SV101', 'Men''s Haircut', 20, 22, 'None');

INSERT INTO SalonService VALUES ('SV102', 'Women''s Haircut', 30, 32, 'None');

INSERT INTO SalonService VALUES ('SV103', 'Child Haircut', 20, 15, 'None');

INSERT INTO SalonService VALUES ('SV104', 'Women''s Hair Color', 60, 76, 'Color, Reagent, Gloves, Reagent Brush, Foil');

INSERT INTO SalonService VALUES ('SV105', 'Women''s Hair Style', 45, 56, 'Shampoo, Conditioner');

INSERT INTO SalonService VALUES ('SV106', 'Men''s Hair Style', 45, 46, 'Shampoo, Conditioner');

INSERT INTO Employee VALUES ('E300', 'Joy', 'Aveda', '46 Easton Ave.', '07026', 18);

INSERT INTO Employee VALUES ('E400', 'Geraldo', 'Geraldo', '12 Fortis Blvd. Apt. 2A', '07026', 22);

INSERT INTO Employee VALUES ('E500', 'Koy', 'Petruzzio', '70 Wilard St. ', '07026', 20);

INSERT INTO Employee VALUES ('E600', 'Landry', 'Monroe', '73 Holly Terrace', '07010', 18);

INSERT INTO Employee VALUES ('E700', 'Pat', 'Reese', '2 Lincoln Place', '07010', 23);

INSERT INTO Employee VALUES ('E800', 'Winter', 'Tanner', '215 Elm Ave', '07665', 23);

INSERT INTO Appointment VALUES ('A400', '10/22/2017 11:00:00 AM', 'C101');

INSERT INTO Appointment VALUES ('A401', '11/06/2017 12:45:00 PM', 'C102');

INSERT INTO Appointment VALUES ('A402', '12/07/2017 02:00:00 PM', 'C106');

INSERT INTO Appointment VALUES ('A403', '12/18/2017 03:30:00 PM', 'C106');

INSERT INTO Appointment VALUES ('A404', '12/21/2017 11:30:00 AM', 'C108');

INSERT INTO Appointment VALUES ('A405', '12/31/2017 10:00:00 AM', 'C107');

INSERT INTO Appointment VALUES ('A406', '01/11/2018 03:15:00 PM', 'C103');

INSERT INTO Appointment VALUES ('A407', '01/12/2018 02:30:00 PM', 'C104');

INSERT INTO Appointment VALUES ('A408', '01/22/2018 04:00:00 PM', 'C105');

INSERT INTO ServiceRendered VALUES ('A400', 1, 'SV104', 75, 'E400');

INSERT INTO ServiceRendered VALUES ('A400', 2, 'SV102', 25, 'E400');

INSERT INTO ServiceRendered VALUES ('A401', 1, 'SV101', 22, 'E500');

INSERT INTO ServiceRendered VALUES ('A402', 1, 'SV104', 75, 'E600');

INSERT INTO ServiceRendered VALUES ('A402', 2, 'SV102', 30, 'E800');

INSERT INTO ServiceRendered VALUES ('A403', 1, 'SV105', 50, 'E300');

INSERT INTO ServiceRendered VALUES ('A404', 1, 'SV105', 55, 'E300');

INSERT INTO ServiceRendered VALUES ('A405', 1, 'SV102', 30, 'E700');

INSERT INTO ServiceRendered VALUES ('A405', 2, 'SV104', 70, 'E700');

INSERT INTO ServiceRendered VALUES ('A405', 3, 'SV105', 50, 'E700');

## Discussion:

* We add just enough data to be able to test out the relationships between the tables and to give the application developers something to work with.
* Be careful of the order in which data is added. For example, all ZipCodes need to be inserted first, before either Customer or Employee (which both use ZipCode as a foreign key) can be inserted.
* When adding VARCHAR data with embedded quotes, use two quotes together. e.g., ‘Women”s Hair Style’
* At this point the database schema is ready for the application developers to get to work designing forms, reports and queries.

Now that the database schema has been created and is populated with some sample data, the database application can be developed.

# Application Implementation

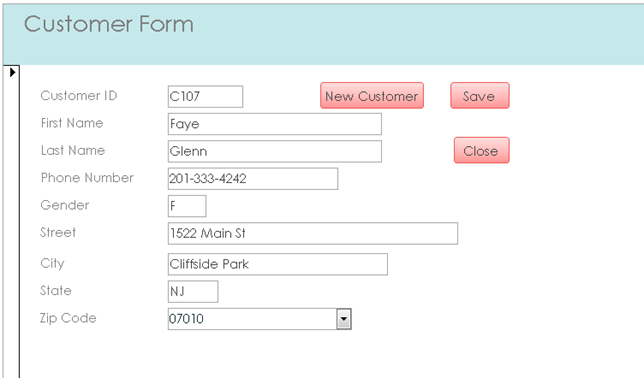
See more examples <https://holowczak.com/the-hair-salon-database-project/6/>

This section contains data entry forms (at least two, and should account for all needed data) and the reports and the SQL the reports generate.

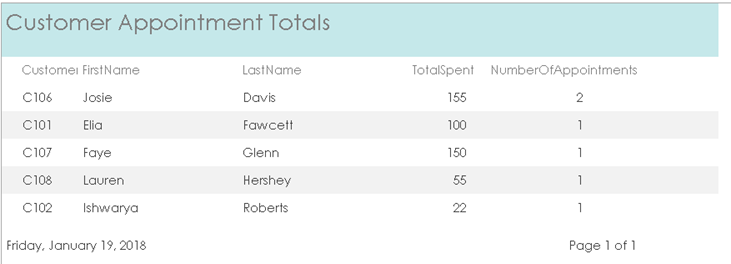
## Navigation Form



## Customer Data Entry Form



## Customer Appointments Totals Report



This report provides a summary of the number of appointments and total amount spent by each customer.

Based on query:

SELECT Customer.CustomerID, FirstName, LastName,

SUM(q.TotalSpent) AS TotalSpent,

COUNT(q.AppointmentID) AS NumberOfAppointments

FROM Customer, Appointment, Query\_Total\_Spent\_Each\_Appointment AS q

WHERE Customer.CustomerID = Appointment.CustomerID

AND Appointment.AppointmentID = q.AppointmentID

GROUP BY Customer.CustomerID, FirstName, LastName

ORDER BY LastName, FirstName;

And Query Total\_Spent\_Each\_Appointment

SELECT Appointment.AppointmentID,

SUM(ServiceExtendedPrice) AS TotalSpent

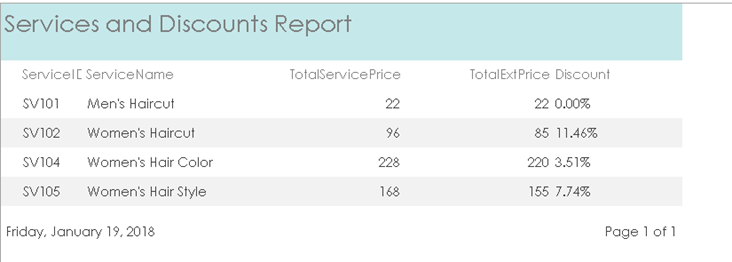
FROM Appointment, ServiceRendered

WHERE Appointment.AppointmentID = ServiceRendered.AppointmentID

GROUP BY Appointment.AppointmentID

ORDER BY Appointment.AppointmentID;

## Services and Discounts Report



This report shows each of the services with totals of the regular service price, the extended price and an indication of the amount of discount applied to the services rendered in the past.

Based on query:

SELECT SalonService.ServiceID, ServiceName,

SUM(ServicePrice) AS TotalServicePrice,

SUM(ServiceExtendedPrice) AS TotalExtPrice,

FORMAT(1.0 - (SUM(ServiceExtendedPrice) / SUM(ServicePrice)), "0.00%")

AS Discount

FROM SalonService, ServiceRendered

WHERE SalonService.ServiceID = ServiceRendered.ServiceID

GROUP BY SalonService.ServiceID, ServiceName

ORDER BY SalonService.ServiceID, ServiceName;

# Conclusion

Conclude with a synopsis of your project, challenges and lessons learned. Overall, the conclusion should be 1 page single spaced.