PART A

The database for Donuts-R-Us has the following attributes in its first normal form with PK denoting the primary key:

First Normal Form:
Table Name: Order
Order ID (PK) First Name Last Name Street Address Apartment Number City State Street Zip Code Home Phone Mobile Phone Other Phone Donut Name Donut Description Unit Price Special Note Donut ID Customer ID Order Time Stamp Quantity

I arrived at the above table by first identifying the attributes that would be needed in a database for the donut shop if it were to be used to track orders and the customers. By including as many details on the customer as possible we retain the ability to target advertising and follow up with the customer in order to improve the business. Using the provided sales order form I cut out the unnecessary items such as subtotal and the total because the total and subtotals can be calculated from the data stored in the table. Because tax is a constant it's not required to be in the database. I then insured that there was atomicity and that there were no repeating groups as well as selecting a primary key.

In my next step in designing the relational database for the donut shop I organized the data attributes into second normal form and introduced relation with the use of foreign keys:

Second Normal Form
Table Name: Customer
Customer ID (PK) First Name Last Name Street Address Apartment Number City State Street Zip Code Home Phone Mobile Phone Other Phone
Table Name: Order
ID (PK) Order ID Customer ID(FK) Donut ID (FK) Quantity Date Special Note
Table Name: Donut
Donut ID (PK) Unit Price Description Name

To bring the database design to 2NF I had to create three separate tables . By creating a table for the orders, donuts, and the customers I was able to eliminate redundant data that had been appearing in the rows while the database design was in 1NF. I used the Order table to associate attributes in the Donut table with the attributes in the Customer table.

I next brought the database design into third normal form by further dividing it into four tables:

Third Normal Form

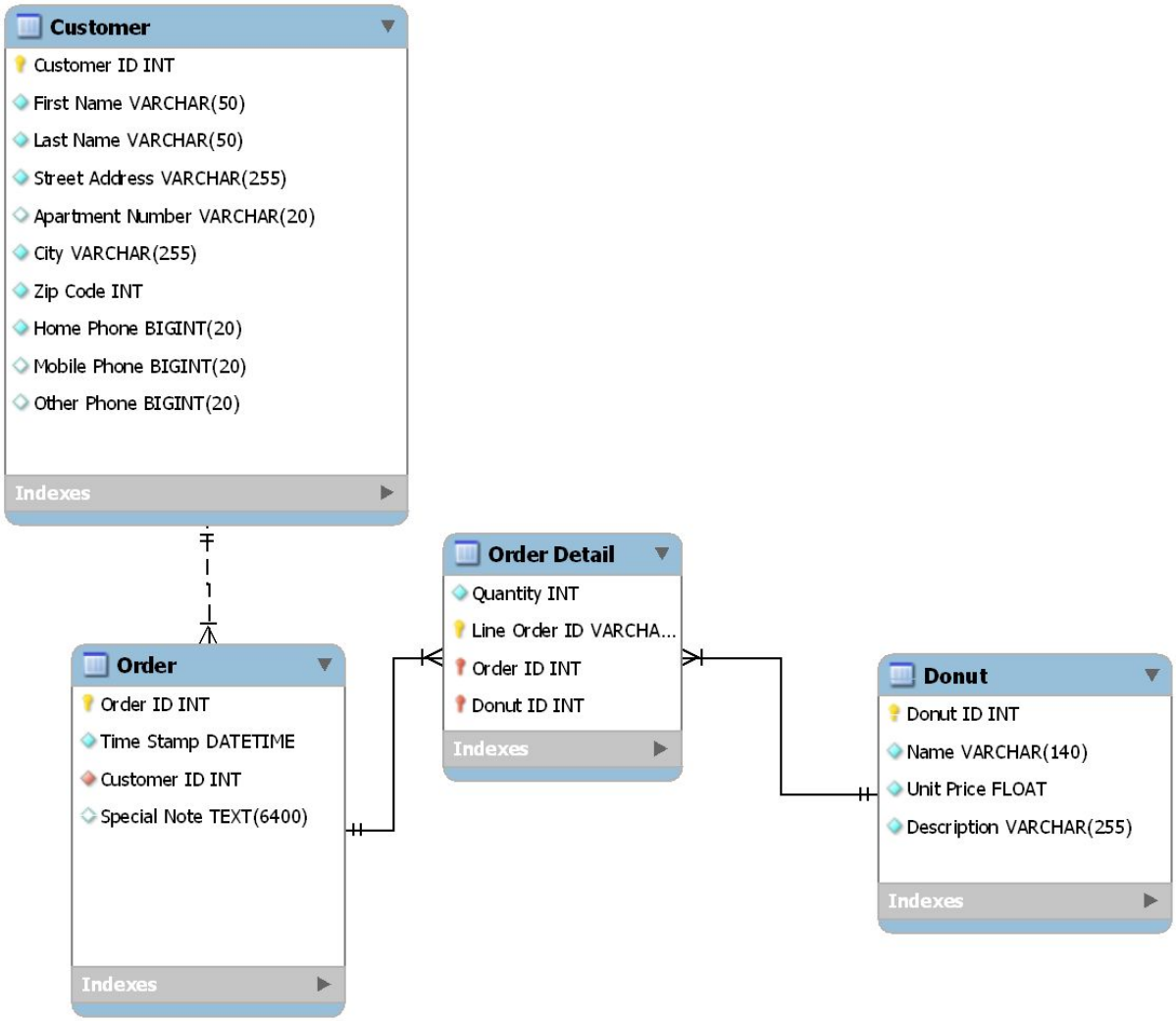
Table Name: Customer
Customer ID (PK) First Name Last Name Street Address Apartment Number City State Street Zip Code Home Phone Mobile Phone Other Phone
Table Name: Order
Order ID (PK) Customer ID(FK) Date Special Note
Table Name: Order Information
Line Order # (PK) Quantity Order ID (FK) Donut ID (FK)
Table Name: Donut
Donut ID (PK) Unit Price Description Name

Getting to third normal form I examined all the fields to ensure that there were no non-key fields dependent on any other non-key fields. In addition I created another table, separating out quantity and donut ID. In the 2NF tables I noticed that date was repeating and so was order ID and customer ID. By creating two separate tables, one for the details of an order and one for the order and dates, this repetition is eliminated, thus saving storage space. Additionally, it is not anticipated that the donut shop would need to know the details of every order for every customer. By offloading those details to a separate table queries can be made

by customer or date quickly, and an additional query would be made to find the exact details of a specific order.

PART B

I next constructed an entity relationship model to visualize the tables I had previously normalized. I first created a Customer entity which is a table that exists on its own. This has a parental relationship with the Order table since an Order may not exist without a Customer, but the Customer can have many Orders. That being said, the Cardinality of the order is 1 to N because the person in the Customer table would not be present in the Customer table unless they had made a purchase. Each order can have many donuts and many donuts can belong to many orders. To deal with this I split the Order table down into Order and Order Detail. Each Order must have 1 to N Order Detail but each Order Detail can belong to 1 and only 1 Order. Order detail ties the Donut and Order entities together . A Donut entity can belong to many Order Detail entities but each Order Detail entity can only have one Donut entity.



PART C

In part C I am to develope MySQL code for creating each of the tables I designed in A and B. Below is said code:

```
CREATE TABLE IF NOT EXISTS `Customer` (
  `Customer ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `First Name` VARCHAR(50) NOT NULL,
  `Last Name` VARCHAR(50) NOT NULL ,
  `Street Address` VARCHAR(255) NOT NULL ,
  `Apartment Number` VARCHAR(20) NULL ,
  `City` VARCHAR(255) NOT NULL ,
  `State` VARCHAR(255) NOT NULL ,
  `Zip Code` INT UNSIGNED NOT NULL ,
  `Home Phone` BIGINT UNSIGNED NOT NULL ,
  `Mobile Phone` BIGINT UNSIGNED NULL  ,
  `Other Phone` BIGINT UNSIGNED NULL ,
  PRIMARY KEY (`Customer ID`) )
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `Order` (
  `Order ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Customer ID` INT UNSIGNED NOT NULL,
  `Time Stamp` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `Special Note` TEXT,
   PRIMARY KEY (`Order ID`))
  ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `Order Detail` (
  `Order ID` INT UNSIGNED NOT NULL,
  `Order Line ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Quantity` INT UNSIGNED NOT NULL,
  `Donut_Donut ID` INT UNSIGNED NOT NULL ,
  PRIMARY KEY (`Order Line ID`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `Donut` (
  `Donut ID` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `Name` VARCHAR(140) NOT NULL,
  `Unit Price` FLOAT UNSIGNED NOT NULL,
  `Description` VARCHAR(255) NOT NULL,
```

```
  PRIMARY KEY (`Donut ID`) )
ENGINE = InnoDB;


ALTER TABLE `Order Detail` ADD
  CONSTRAINT `fk_Order Detail_Donut1`
        FOREIGN KEY (`Donut_Donut ID`)
        REFERENCES `Donut` (`Donut ID`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION;

 ALTER TABLE `Order Detail` ADD
        FOREIGN KEY (`Order ID`)
        REFERENCES `Order` (`Order ID`);

CREATE INDEX d_name ON Donut(`Name`);

DESCRIBE `Customer`;
DESCRIBE `Order`;
DESCRIBE `Order Detail`;
DESCRIBE `Donut`;
```

Below are the screenshots demonstrating that the code above tested successfully.

✔ Record Count: 6; Execution Time: 1ms ✚ View Execution Plan ➦ link

COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	COLUMN_KEY	COLUMN_DEFAULT	EXTRA
Customer ID	int(10) unsigned	NO	PRI	(null)	auto_increment
First Name	varchar(50)	NO		(null)	
Last Name	varchar(50)	NO		(null)	
Street Address	varchar(255)	NO		(null)	
Apartment Number	varchar(20)	YES		(null)	
City	varchar(255)	NO		(null)	
State	varchar(255)	NO		(null)	
Zip Code	int(10) unsigned	NO		(null)	
Home Phone	bigint(20) unsigned	NO		(null)	
Mobile Phone	bigint(20) unsigned	YES		(null)	
Other Phone	bigint(20) unsigned	YES		(null)	

✔ Record Count: 11; Execution Time: 1ms ➦ link

COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	COLUMN_KEY	COLUMN_DEFAULT	EXTRA
Order ID	int(10) unsigned	NO	PRI	(null)	auto_increment
Customer ID	int(10) unsigned	NO		(null)	
Time Stamp	timestamp	NO		CURRENT_TIMESTAMP	
Special Note	text	YES		(null)	

✔ Record Count: 4; Execution Time: 1ms ➦ link

COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	COLUMN_KEY	COLUMN_DEFAULT	EXTRA
Order ID	int(10) unsigned	NO	MUL	(null)	
Order Line ID	int(10) unsigned	NO	PRI	(null)	auto_increment
Quantity	int(10) unsigned	NO		(null)	
Donut_Donut ID	int(10) unsigned	NO	MUL	(null)	

✔ Record Count: 4; Execution Time: 1ms ➦ link

COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	COLUMN_KEY	COLUMN_DEFAULT	EXTRA
Donut ID	int(10) unsigned	NO	PRI	(null)	auto_increment
Name	varchar(140)	NO	MUL	(null)	
Unit Price	float unsigned	NO		(null)	
Description	varchar(255)	NO		(null)	

✔ Record Count: 4; Execution Time: 1ms link

PART D

In part D a view which includes all details from customer with the first and last names concatenated. Below is the code I used and the screen shots:

```
CREATE VIEW `Customer View` AS
SELECT
CONCAT(`First Name`, " ",`Last Name`) AS `Customer Name`,
  `Street Address` ,
  `Apartment Number`,
  `City`,
  `State`,
  `Zip Code`,
  `Home Phone`,
  `Mobile Phone`,
  `Other Phone`
FROM `Customer`;
```

```
SELECT * FROM `Customer View`;
```

✔ Record Count: 6; Execution Time: 1ms ✚ View Execution Plan link

Customer Name	Street Address	Apartment Number	City	State	Zip Code	Home Phone	Mobile Phone	Other Phone
Andrew Hart	3670 SE Knott Street	A6	Portland	Oregon	97202	4012848821	3264556488	(null)
Lolita Cherry	12 Hupmbert Street	6C9	Ithica	New York	97212	1028855579	(null)	(null)
Turtle Smith	9000 Powerman Blvd	99A	Portland	Oregon	97252	9151683360	(null)	(null)
Manny Killjoy	10 Faceplant Street	N2	Portland	Oregon	97252	5032581956	(null)	(null)

PART E

In part E I am to create an index on the donut name column. Below is that code:

```
CREATE INDEX d_name ON Donut(`Name`);
```

COLUMN_NAME	COLUMN_TYPE	IS_NULLABLE	COLUMN_KEY	COLUMN_DEFAULT	EXTRA
Donut ID	int(10) unsigned	NO	PRI	(null)	auto_increment
Name	varchar(140)	NO	MUL	(null)	
Unit Price	float unsigned	NO		(null)	
Description	varchar(255)	NO		(null)	

Record Count: 4; Execution Time: 1ms link

PART F

In part F I am to populate the tables with some data. I invented some customers and used the order form to create some order data and populate the donut table. All of the Order Detail and Order table inserts will be placed within transactions to ensure atomicity. Below is the code and test results:

```
    INSERT INTO `Customer` (
 `First Name`,
 `Last Name` ,
 `Street Address` ,
 `Apartment Number`,
 `City`,
 `State`,
 `Zip Code`,
 `Home Phone`,
 `Mobile Phone`

)
VALUES (
 "Andrew",
 "Hart",
 "3670 SE Knott Street",
 "A6",
 "Portland",
 "Oregon",
 "97202",
 "4012848821",
```

```
  "3264556488"

);
          INSERT INTO `Customer` (
  `First Name`,
  `Last Name` ,
  `Street Address` ,
  `Apartment Number`,
  `City`,
  `State`,
  `Zip Code`,
  `Home Phone`

)
VALUES (
 "Lolita",
  "Cherry",
  "12 Hupmbert Street",
  "6C9",
  "Ithica",
  "New York",
  "97212",
  "1028855579"

);
          INSERT INTO `Customer` (
  `First Name`,
  `Last Name` ,
  `Street Address` ,
  `Apartment Number`,
  `City`,
  `State`,
  `Zip Code`,
  `Home Phone`

)
VALUES (
 "Turtle",
  "Smith",
  "9000 Powerman Blvd",
  "99A",
  "Portland",
  "Oregon",
```

```
  "97252",
  "9151683360"

);
          INSERT INTO `Customer` (
  `First Name`,
  `Last Name` ,
  `Street Address` ,
  `Apartment Number`,
  `City`,
  `State`,
  `Zip Code`,
  `Home Phone`

)
VALUES (
 "Manny",
  "Killjoy",
  "10 Faceplant Street",
  "N2",
  "Portland",
  "Oregon",
  "97252",
  "5032581956"

);


INSERT INTO `Donut` (
`Name`,
`Unit Price`,
`Description`
)
VALUES(
"Plain",
"1.50",
"Plain Donut"
);

INSERT INTO `Donut` (
`Name`,
`Unit Price`,
`Description`
```

```
)
VALUES(
"Glazed",
"1.75",
"Glazed Donut"
);

INSERT INTO `Donut` (
`Name`,
`Unit Price`,
`Description`
)
VALUES(
"Cinnamon",
"1.75",
"Cinnamon Donut"
);

INSERT INTO `Donut` (
`Name`,
`Unit Price`,
`Description`
)
VALUES(
"Chocolate",
"1.75",
"Chocolate Donut"
);
INSERT INTO `Donut` (
`Name`,
`Unit Price`,
`Description`
)
VALUES(
"Sprinkle",
"1.75",
"Sprinkle Donut"
);

INSERT INTO `Donut` (
`Name`,
`Unit Price`,
`Description`
```

```
)
VALUES(
"Gluten-Free",
"2.00",
"Gluten-Free Donut"
);
START TRANSACTION;
INSERT INTO `Order`
(
  `Customer ID`,
  `Special Note`
)
VALUES(
"2",
"Please include plates and napkins."
);

INSERT INTO `Order Detail` (
`Order ID` ,
`Quantity`,
`Donut_Donut ID`
)
VALUES (
"1",
"1",
"1"
);

INSERT INTO `Order Detail` (
`Order ID` ,
`Quantity`,
`Donut_Donut ID`
)
VALUES (
"1",
"5",
"2"
);
INSERT INTO `Order Detail` (
`Order ID` ,
`Quantity`,
`Donut_Donut ID`
)
```

```
VALUES (
"1",
"12",
"3"
);
INSERT INTO `Order Detail` (
`Order ID` ,
`Quantity`,
`Donut_Donut ID`
)
VALUES (
"1",
"3",
"4"
);
INSERT INTO `Order Detail` (
`Order ID` ,
`Quantity`,
`Donut_Donut ID`
)
VALUES (
"1",
"4",
"5"
);
INSERT INTO `Order Detail` (
`Order ID` ,
`Quantity`,
`Donut_Donut ID`
)
VALUES (
"1",
"5",
"6"
);
COMMIT;
```

PART G1

Below is the code and screenshots used for testing part F and Part G1:

```
SELECT * FROM `Customer`;
SELECT * FROM `Order`;
SELECT * FROM `Order Detail`;
SELECT * FROM `Donut`;
```

Sugar Tax
0.1

✔ Record Count: 1; Execution Time: 0ms ✚ View Execution Plan ➦ link

Customer ID	First Name	Last Name	Street Address	Apartment Number	City	State	Zip Code	Home Phone	Mobile Phone	Other Phone
1	Andrew	Hart	3670 SE Knott Street	A6	Portland	Oregon	97202	4012848821	3264556488	(null)
2	Lolita	Cherry	12 Hupmbert Street	6C9	Ithica	New York	97212	1028855579	(null)	(null)
3	Turtle	Smith	9000 Powerman Blvd	99A	Portland	Oregon	97252	9151683360	(null)	(null)
4	Manny	Killjoy	10 Faceplant Street	N2	Portland	Oregon	97252	5032581956	(null)	(null)

✔ Record Count: 4; Execution Time: 1ms ✚ View Execution Plan ➦ link

Order ID	Customer ID	Time Stamp	Special Note
1	2	June, 29 2015 23:38:41	Please include plates and napkins.

✔ Record Count: 1; Execution Time: 1ms ✚ View Execution Plan ➦ link

Order ID	Order Line ID	Quantity	Donut_Donut ID
1	1	1	1
1	2	5	2
1	3	12	3
1	4	3	4
1	5	4	5
1	6	5	6

✔ Record Count: 6; Execution Time: 1ms ✚ View Execution Plan ➦ link

Donut ID	Name	Unit Price	Description
1	Plain	1.5	Plain Donut
2	Glazed	1.75	Glazed Donut
3	Cinnamon	1.75	Cinnamon Donut
4	Chocolate	1.75	Chocolate Donut
5	Sprinkle	1.75	Sprinkle Donut
6	Gluten-Free	2	Gluten-Free Donut

PART G2

In part G2 I created a complex join to display all of the information that was displayed on the customer order form. In addition to this I also include a view that calculates the total for an order.

```
  CREATE VIEW `tom` AS
SELECT
`Order`.`Order ID`,
CONCAT("$", FORMAT(SUM((FORMAT(`Order Detail`.`Quantity` *`Donut`.`Unit Price` , 2))),2))
AS `Sub Total`,
CONCAT ("%", FORMAT(10, 0)) AS `Tax Applied`,
CONCAT("$", FORMAT(SUM((FORMAT(`Order Detail`.`Quantity` *`Donut`.`Unit Price` * 1.1 ,
2))),2)) AS `Total`
FROM `Order` JOIN `Order Detail` ON `Order`.`Order ID` = `Order Detail`.`Order ID` JOIN
`Donut` ON `Order Detail`.`Donut_Donut ID` = `Donut`.`Donut ID`;
```

```
CREATE VIEW `Order_Form` AS
SELECT
  `Order`.`Order ID`,
  `Special Note`,
  `Order Detail`.`Quantity`,
  `Order Detail`.`Donut_Donut ID`,
  `Donut`.`Name`,
  CONCAT("$",`Donut`.`Unit Price`) AS `Donut Price`,

  `Donut`.`Description`,
  `Customer`.`Customer ID`,
  `Customer`.`First Name`,
  `Customer`.`Last Name` ,
  `Customer`.`Street Address` ,
  `Customer`.`Apartment Number`,
  `Customer`.`City`,
  `Customer`.`State`,
  `Customer`.`Zip Code`,
  `Customer`.`Home Phone`,
  CONCAT("$",(FORMAT(`Order Detail`.`Quantity` *`Donut`.`Unit Price`, 2))) AS `Line Total`
  FROM
  `Order` JOIN `Customer` ON `Order`.`Order ID` =`Customer`.`Customer ID`
```

```
  JOIN `Order Detail` ON `Order`.`Order ID` = `Order Detail`.`Order ID` JOIN `Donut` ON
`Order Detail`.`Donut_Donut ID` = `Donut`.`Donut ID`
  ORDER BY `Order`.`Order ID`;
```

```
SELECT * FROM `Order_Form` ;
SELECT * FROM `tom` WHERE `Order ID` = 1;
```

Order ID	Special Note	Quantity	Donut_Donut ID	Name	Donut Price	Description	Customer ID	First Name	Last Name	Street Address	Apartment Number	City	State	Zip Code	Home Phone	Line Total
1	Please include plates and napkins.	1	1	Plain	$1.5	Plain Donut	1	Andrew	Hart	3670 SE Knott Street	A6	Portland	Oregon	97202	4012848821	$1.50
1	Please include plates and napkins.	5	2	Glazed	$1.75	Glazed Donut	1	Andrew	Hart	3670 SE Knott Street	A6	Portland	Oregon	97202	4012848821	$8.75
1	Please include plates and napkins.	12	3	Cinnamon	$1.75	Cinnamon Donut	1	Andrew	Hart	3670 SE Knott Street	A6	Portland	Oregon	97202	4012848821	$21.00
1	Please include plates and napkins.	3	4	Chocolate	$1.75	Chocolate Donut	1	Andrew	Hart	3670 SE Knott Street	A6	Portland	Oregon	97202	4012848821	$5.25
1	Please include plates and napkins.	4	5	Sprinkle	$1.75	Sprinkle Donut	1	Andrew	Hart	3670 SE Knott Street	A6	Portland	Oregon	97202	4012848821	$7.00
1	Please include plates and napkins.	5	6	Gluten-Free	$2	Gluten-Free Donut	1	Andrew	Hart	3670 SE Knott Street	A6	Portland	Oregon	97202	4012848821	$10.00

✔ Record Count: 6; Execution Time: 1ms ✚ View Execution Plan ➦ link

Order ID	Sub Total	Tax Applied	Total
1	$53.50	%10	$58.85

✔ Record Count: 1; Execution Time: 1ms ✚ View Execution Plan ➦ link