Timothy Clark
C170: Data Management – Applications (MySQL)

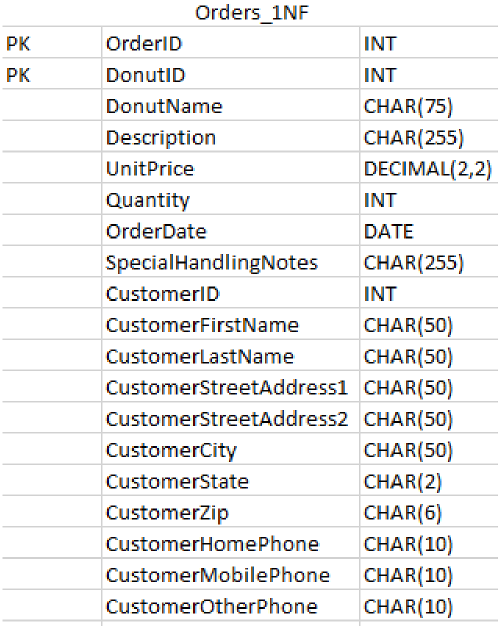
1A) 1NF

RAW MySQL:

```
CREATE TABLE Orders_1NF
(
		OrderID INT NOT NULL AUTO_INCREMENT,
		DonutID INT NOT NULL,
		DonutName CHAR(75) NOT NULL,
		Description CHAR(255) NULL,
		UnitPrice DECIMAL(2,2) NULL,
		Quantity INT NOT NULL,
		OrderDate DATE NOT NULL,
		SpecialHandlingNotes CHAR(255) NULL,
		CustomerID INT NULL,
		CustomerFirstName CHAR(50) NULL,
		CustomerLastName CHAR(50) NULL,
		CustomerStreetAddress1 CHAR(50) NULL,
		CustomerStreetAddress2 CHAR(50) NULL,
		CustomerCity CHAR(50) NULL,
		CustomerState CHAR(2) NULL,
		CustomerZip CHAR(6) NULL,
		CustomerHomePhone CHAR(10) NULL,
		CustomerMobilePhone CHAR(10) NULL,
		CustomerOtherPhone CHAR(10) NULL,
		CONSTRAINT PK_Orders_1NF PRIMARY KEY (OrderID, DonutID)
);
```

Explanation:

The above code was reached after the following considerations. Firstly, I reviewed the Sales Order Form which was provided with this assignment. After reviewing the Order Form, I broke down the sections into individual elements which I then converted into SQL table columns. This will allow orders to populate as Rows into the Orders_1NF table. The primary key selected is derived from OrderID and DonutID creating a composite key with the two data points which enforces complete uniqueness.



Orders_1NF

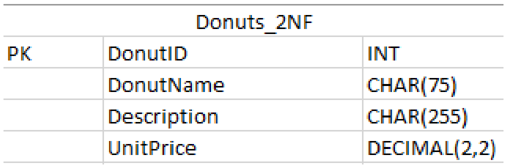
PK	OrderID	INT
PK	DonutID	INT
	DonutName	CHAR(75)
	Description	CHAR(255)
	UnitPrice	DECIMAL(2,2)
	Quantity	INT
	OrderDate	DATE
	SpecialHandlingNotes	CHAR(255)
	CustomerID	INT
	CustomerFirstName	CHAR(50)
	CustomerLastName	CHAR(50)
	CustomerStreetAddress1	CHAR(50)
	CustomerStreetAddress2	CHAR(50)
	CustomerCity	CHAR(50)
	CustomerState	CHAR(2)
	CustomerZip	CHAR(6)
	CustomerHomePhone	CHAR(10)
	CustomerMobilePhone	CHAR(10)
	CustomerOtherPhone	CHAR(10)

Timothy Clark
C170: Data Management – Applications (MySQL)

1B) 2NF - A

RAW MySQL:

```
CREATE TABLE Donuts_2NF
(
		DonutID INT NOT NULL,
		DonutName CHAR(75) NOT NULL,
		Description CHAR(255) NULL,
		UnitPrice DECIMAL(2,2) NULL,
		CONSTRAINT PK_Donuts_2NF PRIMARY KEY (DonutID)
);
```

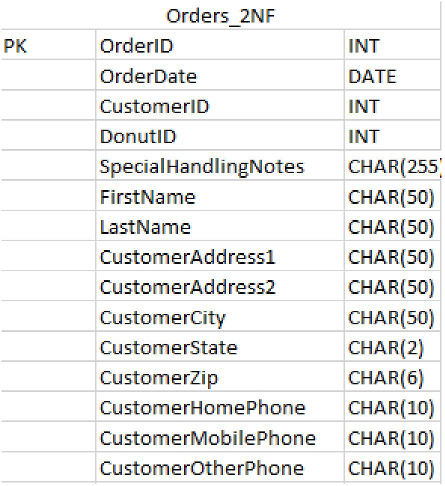


	Donuts_2NF	
PK	DonutID	INT
	DonutName	CHAR(75)
	Description	CHAR(255)
	UnitPrice	DECIMAL(2,2)

1B) 2NF - B

RAW MySQL:

```
CREATE TABLE Orders_2NF
(
		OrderID INT NOT NULL AUTO_INCREMENT,
		OrderDate DATE NOT NULL,
		CustomerID INT NULL,
		DonutID INT NOT NULL,
		SpecialHandlingNotes CHAR(255) NULL,
		CustomerFirstName CHAR(50) NULL,
		CustomerLastName CHAR(50) NULL,
		CustomerStreetAddress1 CHAR(50) NULL,
		CustomerStreetAddress2 CHAR(50) NULL,
		CustomerCity CHAR(50) NULL,
		CustomerState CHAR(2) NULL,
		CustomerZip CHAR(6) NULL,
		CustomerHomePhone CHAR(10) NULL,
		CustomerMobilePhone CHAR(10) NULL,
		CustomerOtherPhone CHAR(10) NULL,
		CONSTRAINT PK_Orders_2NF PRIMARY KEY (OrderID)
);
```



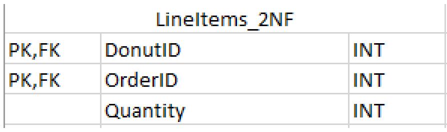
	Orders_2NF	
PK	OrderID	INT
	OrderDate	DATE
	CustomerID	INT
	DonutID	INT
	SpecialHandlingNotes	CHAR(255
	FirstName	CHAR(50)
	LastName	CHAR(50)
	CustomerAddress1	CHAR(50)
	CustomerAddress2	CHAR(50)
	CustomerCity	CHAR(50)
	CustomerState	CHAR(2)
	CustomerZip	CHAR(6)
	CustomerHomePhone	CHAR(10)
	CustomerMobilePhone	CHAR(10)
	CustomerOtherPhone	CHAR(10)

Timothy Clark
C170: Data Management – Applications (MySQL)

1B) 2NF – C

RAW MySQL:

```
CREATE TABLE LineItems_2NF
(
DonutID INT NOT NULL,
OrderID INT NOT NULL,
QUANTITY INT NOT NULL,
CONSTRAINT FK_LineItems_2NF_Order FOREIGN KEY (OrderID) REFERENCES Orders_2NF(OrderID),
CONSTRAINT FK_LineItems_2NF_Donut FOREIGN KEY (DonutID) REFERENCES Donuts_2NF(DonutID),
CONSTRAINT PK_LineItems_2NF PRIMARY KEY (OrderID,DonutID)
);
```



	LineItems_2NF	
PK,FK	DonutID	INT
PK,FK	OrderID	INT
	Quantity	INT

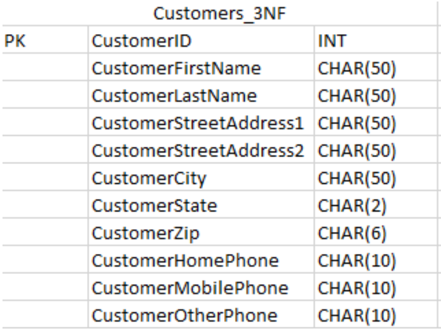
Explanation:

The above code was reached after the following considerations. Firstly, the order data was broken out into three sections. Donut data to be stored separately into a singular table where each row represents an individual item that can be ordered. An Orders table which will hold information specific to each individual sales transaction that occurs as an independent row. Lastly, a line item table that will be utilized to store each specific line item within an order. Two foreign keys are configured to enforce complete uniqueness when a line item is created and linked to a valid order and donut. These two foreign keys combine to create a composite primary key.

1C) 3NF - A

RAW MySQL:

```
CREATE TABLE Customers_3NF
(
        CustomerID INT NULL,
        CustomerFirstName CHAR(50) NULL,
        CustomerLastName CHAR(50) NULL,
        CustomerStreetAddress1 CHAR(50) NULL,
        CustomerStreetAddress2 CHAR(50) NULL,
        CustomerCity CHAR(50) NULL,
        CustomerState CHAR(2) NULL,
        CustomerZip CHAR(6) NULL,
        CustomerHomePhone CHAR(10) NULL,
        CustomerMobilePhone CHAR(10) NULL,
        CustomerOtherPhone CHAR(10) NULL,
        CONSTRAINT PK_Customers_3NF PRIMARY KEY (CustomerID)
);
```



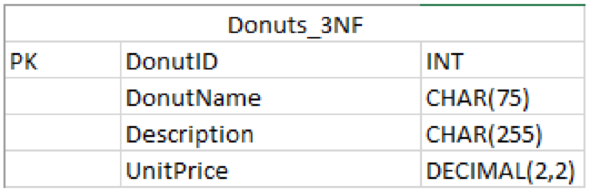
	Customers_3NF	
PK	CustomerID	INT
	CustomerFirstName	CHAR(50)
	CustomerLastName	CHAR(50)
	CustomerStreetAddress1	CHAR(50)
	CustomerStreetAddress2	CHAR(50)
	CustomerCity	CHAR(50)
	CustomerState	CHAR(2)
	CustomerZip	CHAR(6)
	CustomerHomePhone	CHAR(10)
	CustomerMobilePhone	CHAR(10)
	CustomerOtherPhone	CHAR(10)

Timothy Clark
C170: Data Management – Applications (MySQL)

1C) 3NF - B

RAW MySQL:

```
CREATE TABLE Donuts_3NF
(
DonutID INT NOT NULL,
DonutName CHAR(50) NOT NULL,
Description CHAR(50) NOT NULL,
UnitPrice DECIMAL(2,2) NOT NULL,
CONSTRAINT PK_Donuts_3NF PRIMARY KEY (DonutID)
);
```

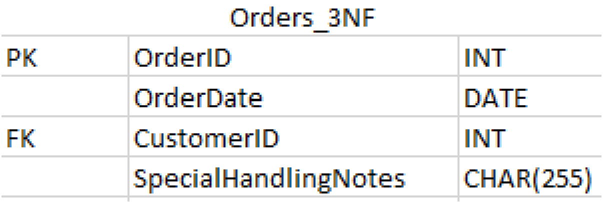


	Donuts_3NF	
PK	DonutID	INT
	DonutName	CHAR(75)
	Description	CHAR(255)
	UnitPrice	DECIMAL(2,2)

1C) 3NF - C

RAW MySQL:

```
CREATE TABLE Orders_3NF
(
        OrderID INT NOT NULL AUTO_INCREMENT,
        OrderDate DATE NOT NULL,
        CustomerID INT NULL,
        SpecialHandlingNotes CHAR(255) NULL,
        CONSTRAINT PK_Orders_3NF PRIMARY KEY (OrderID),
        CONSTRAINT FK_Orders_Customers_3NF FOREIGN KEY (CustomerID) REFERENCES Customers_3NF(CustomerID)
);
```

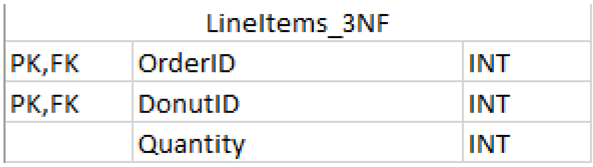


	Orders_3NF	
PK	OrderID	INT
	OrderDate	DATE
FK	CustomerID	INT
	SpecialHandlingNotes	CHAR(255)

1C) 3NF - D

RAW MySQL:

```
CREATE TABLE LineItems_3NF
(
OrderID INT NOT NULL,
DonutID INT NOT NULL,
QUANTITY INT NOT NULL,
CONSTRAINT PK_LineItems_3NF PRIMARY KEY (DonutID, OrderID),
CONSTRAINT FK_LineItems_Donut_3NF FOREIGN KEY (DonutID) REFERENCES Donuts_3NF (DonutID),
CONSTRAINT FK_LineItems_Order_3NF FOREIGN KEY (OrderID) REFERENCES Orders_3NF (OrderID)
);
```



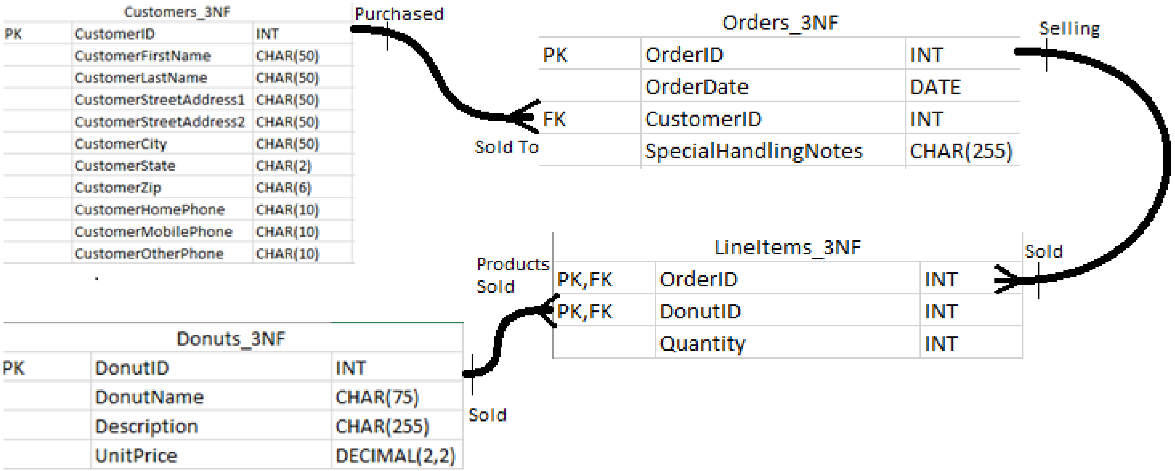
	LineItems_3NF	
PK,FK	OrderID	INT
PK,FK	DonutID	INT
	Quantity	INT

Timothy Clark
C170: Data Management – Applications (MySQL)

Explanation:

The above code was reached after the following considerations. The tables within the third Normalization Form closely mimic the table examples from the Second Normalization Form. However, with the Third Normalization Form a new table named Customers has been added. The Customers table allows for the same customer to be utilized via Foreign Key reference without the need of repeating information being stored for use across multiple orders. As this data resides in another separate table a Foreign Key is being utilized to enforce valid data for look-ups within the Orders table. The Line Items table contains a Foreign Key constraint against the Orders table thus only allowing valid order records to be linked back to their correlating line items within the Line Items table. The Line Item table also contains a composite primary key utilizing the OrderID and DonutID to enforce unique records. Each table is using a named primary key for easy identification rather than using auto numeric named keys.

1C – Part B) Diagram



Customers_3NF

PK	CustomerID	INT
	CustomerFirstName	CHAR(50)
	CustomerLastName	CHAR(50)
	CustomerStreetAddress1	CHAR(50)
	CustomerStreetAddress2	CHAR(50)
	CustomerCity	CHAR(50)
	CustomerState	CHAR(2)
	CustomerZip	CHAR(6)
	CustomerHomePhone	CHAR(10)
	CustomerMobilePhone	CHAR(10)
	CustomerOtherPhone	CHAR(10)

Orders_3NF

PK	OrderID	INT
	OrderDate	DATE
FK	CustomerID	INT
	SpecialHandlingNotes	CHAR(255)

LineItems_3NF

PK,FK	OrderID	INT
PK,FK	DonutID	INT
	Quantity	INT

Donuts_3NF

PK	DonutID	INT
	DonutName	CHAR(75)
	Description	CHAR(255)
	UnitPrice	DECIMAL(2,2)

Diagram Explanation:

The above diagram was reached after the following considerations. There are four tables in Third Normalization form. Thus, all four of these tables have been included and combined into a single connected record system. Removing any singular table from this data set would create an incomplete data set causing issues in the system and order.

Customers is linked to the sales table as it stores the data of whom the order was sold too.

Orders is linked to whom placed the order to obtain the data held in the Customers table as well as to the Line Items table as it needs to obtain the Items that were placed in the actual Order of relating ID.

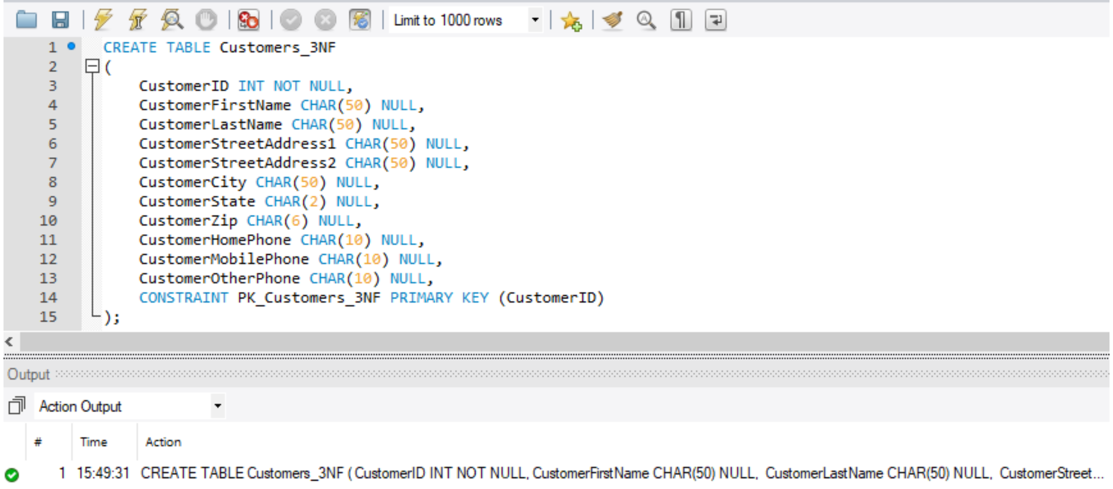
Line Items is storing the individual line items of donuts ordered in any singular order across all orders. As a single Order (held in the Orders table) can include multiple Line Items however it will only remain a singular Order line item in the Orders table as it all stems from a singular OrderID.

Donuts is the table holding information on all products available for order. There is only one row for each individual donut available for purchase as the information only needs to be stored once and then referenced when accessed via the foreign keys now available.

Foreign Keys/Primary Keys are being utilized to enforce cardinality through restrictive constraints assigned utilizing foreign key constraints. FK_Orders_Customers_3 requires a valid customer to be utilized within the Orders table. FK_LineItems_Order_3NF requires that any line item is linked to a valid order record. FK_LineItems_Donuts_3NF requires that any donut listed in the line item table exist within the Donuts table. Lastly, adding a composite key including the OrderID and DonutID enforces a constraint of complete uniqueness not allowing any singular duplicate donut on the same order.

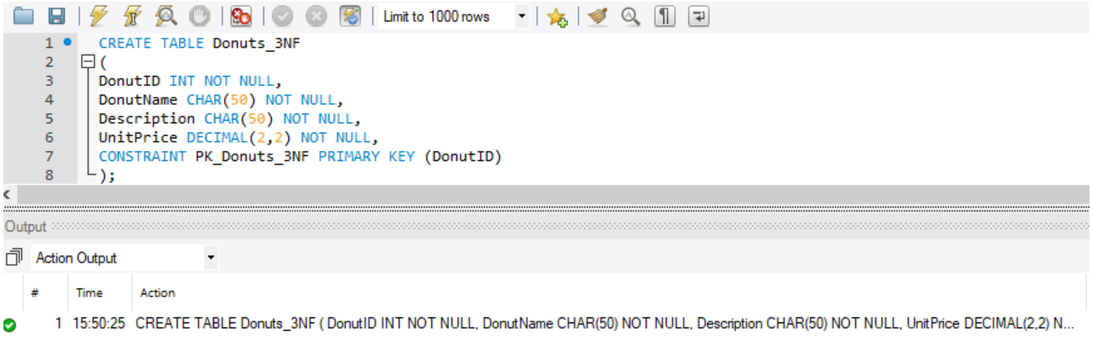
Timothy Clark
C170: Data Management – Applications (MySQL)

Proof of Execution Screenshots



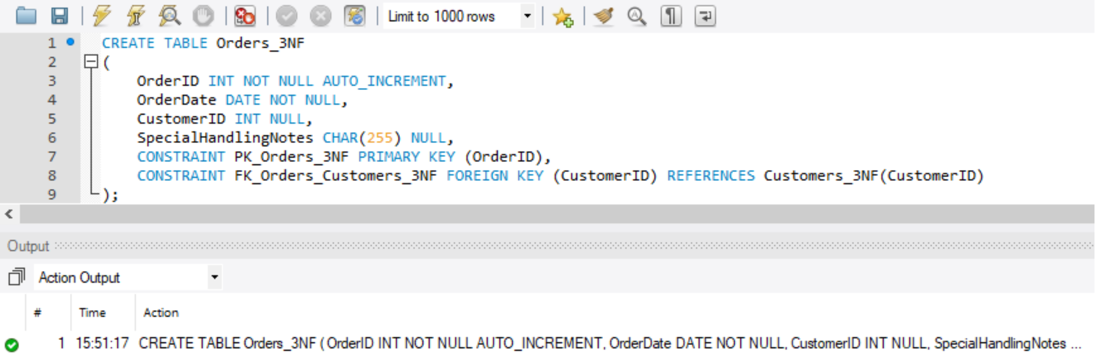
```
CREATE TABLE Customers_3NF
(
    CustomerID INT NOT NULL,
    CustomerFirstName CHAR(50) NULL,
    CustomerLastName CHAR(50) NULL,
    CustomerStreetAddress1 CHAR(50) NULL,
    CustomerStreetAddress2 CHAR(50) NULL,
    CustomerCity CHAR(50) NULL,
    CustomerState CHAR(2) NULL,
    CustomerZip CHAR(6) NULL,
    CustomerHomePhone CHAR(10) NULL,
    CustomerMobilePhone CHAR(10) NULL,
    CustomerOtherPhone CHAR(10) NULL,
    CONSTRAINT PK_Customers_3NF PRIMARY KEY (CustomerID)
);
```

```
#   Time      Action
1   15:49:31  CREATE TABLE Customers_3NF ( CustomerID INT NOT NULL, CustomerFirstName CHAR(50) NULL,  CustomerLastName CHAR(50) NULL,  CustomerStreet...
```



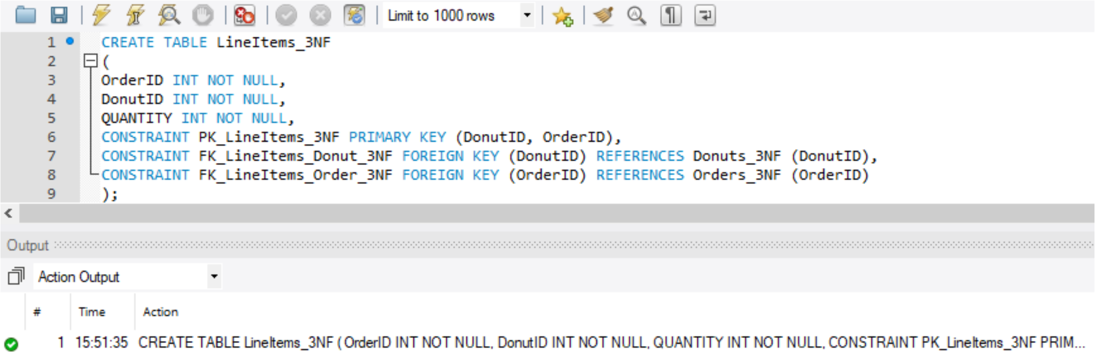
```
CREATE TABLE Donuts_3NF
(
DonutID INT NOT NULL,
DonutName CHAR(50) NOT NULL,
Description CHAR(50) NOT NULL,
UnitPrice DECIMAL(2,2) NOT NULL,
CONSTRAINT PK_Donuts_3NF PRIMARY KEY (DonutID)
);
```

```
#   Time      Action
1   15:50:25  CREATE TABLE Donuts_3NF ( DonutID INT NOT NULL, DonutName CHAR(50) NOT NULL, Description CHAR(50) NOT NULL, UnitPrice DECIMAL(2,2) N...
```



```
CREATE TABLE Orders_3NF
(
    OrderID INT NOT NULL AUTO_INCREMENT,
    OrderDate DATE NOT NULL,
    CustomerID INT NULL,
    SpecialHandlingNotes CHAR(255) NULL,
    CONSTRAINT PK_Orders_3NF PRIMARY KEY (OrderID),
    CONSTRAINT FK_Orders_Customers_3NF FOREIGN KEY (CustomerID) REFERENCES Customers_3NF(CustomerID)
);
```

```
#   Time      Action
1   15:51:17  CREATE TABLE Orders_3NF ( OrderID INT NOT NULL AUTO_INCREMENT, OrderDate DATE NOT NULL, CustomerID INT NULL, SpecialHandlingNotes ...
```



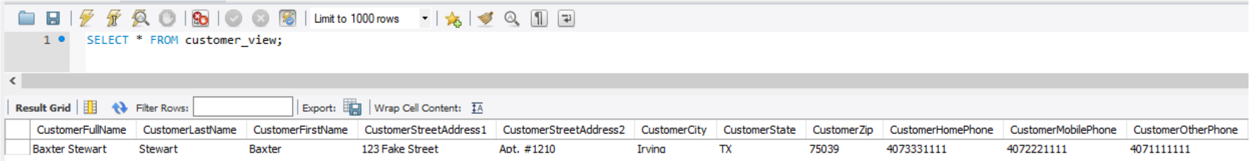
```
CREATE TABLE LineItems_3NF
(
OrderID INT NOT NULL,
DonutID INT NOT NULL,
QUANTITY INT NOT NULL,
CONSTRAINT PK_LineItems_3NF PRIMARY KEY (DonutID, OrderID),
CONSTRAINT FK_LineItems_Donut_3NF FOREIGN KEY (DonutID) REFERENCES Donuts_3NF (DonutID),
CONSTRAINT FK_LineItems_Order_3NF FOREIGN KEY (OrderID) REFERENCES Orders_3NF (OrderID)
);
```

```
#   Time      Action
1   15:51:35  CREATE TABLE LineItems_3NF ( OrderID INT NOT NULL, DonutID INT NOT NULL, QUANTITY INT NOT NULL, CONSTRAINT PK_LineItems_3NF PRIM...
```

Timothy Clark
C170: Data Management – Applications (MySQL)

D) Customer View

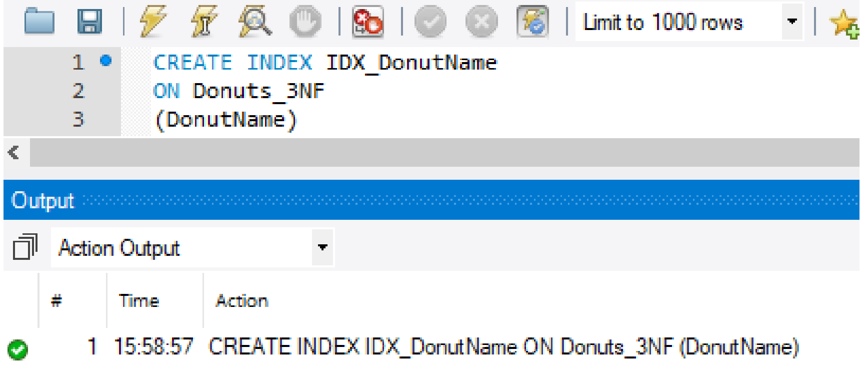
CREATE VIEW Customer_View AS SELECT
CONCAT(CustomerFirstName, ' ', CustomerLastName) AS CustomerFullName, CustomerLastName, CustomerFirstName,
CustomerStreetAddress1, CustomerStreetAddress2, CustomerCity, CustomerState, CustomerZip,
CustomerHomePhone, CustomerMobilePhone, CustomerOtherPhone
FROM Customers_3NF



```
SELECT * FROM customer_view;
```

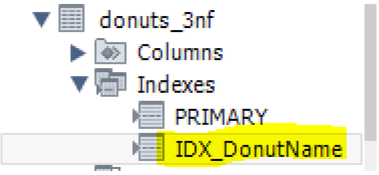
CustomerFullName	CustomerLastName	CustomerFirstName	CustomerStreetAddress1	CustomerStreetAddress2	CustomerCity	CustomerState	CustomerZip	CustomerHomePhone	CustomerMobilePhone	CustomerOtherPhone
Baxter Stewart	Stewart	Baxter	123 Fake Street	Apt. #1210	Irving	TX	75039	4073331111	4072221111	4071111111

E) Donut Name Index

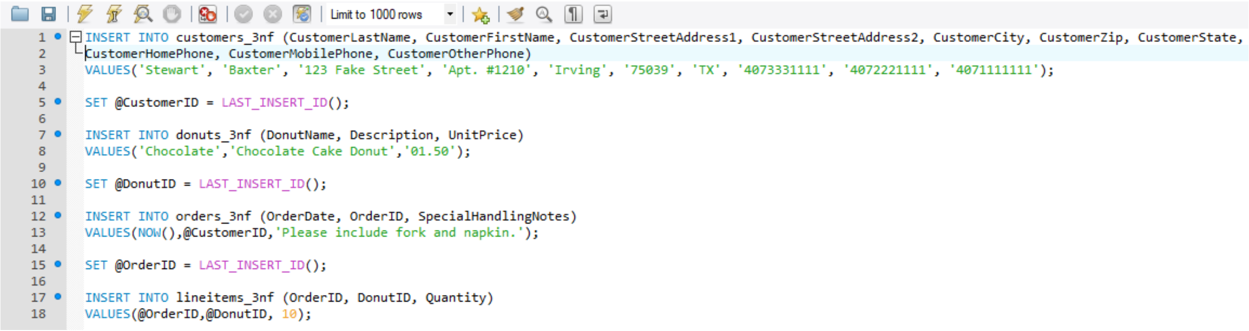


```
CREATE INDEX IDX_DonutName
ON Donuts_3NF
(DonutName)
```

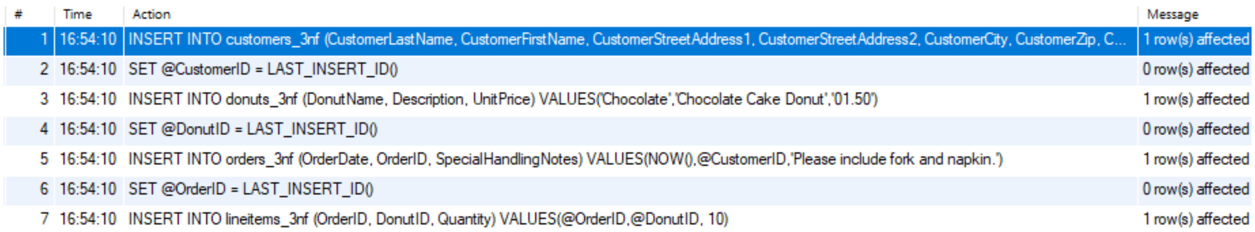
```
#   Time      Action
1   15:58:57  CREATE INDEX IDX_DonutName ON Donuts_3NF (DonutName)
```



F) Populate Tables



```
INSERT INTO customers_3nf (CustomerLastName, CustomerFirstName, CustomerStreetAddress1, CustomerStreetAddress2, CustomerCity, CustomerZip, CustomerState,
CustomerHomePhone, CustomerMobilePhone, CustomerOtherPhone)
VALUES('Stewart', 'Baxter', '123 Fake Street', 'Apt. #1210', 'Irving', '75039', 'TX', '4073331111', '4072221111', '4071111111');

SET @CustomerID = LAST_INSERT_ID();

INSERT INTO donuts_3nf (DonutName, Description, UnitPrice)
VALUES('Chocolate','Chocolate Cake Donut','01.50');

SET @DonutID = LAST_INSERT_ID();

INSERT INTO orders_3nf (OrderDate, OrderID, SpecialHandlingNotes)
VALUES(NOW(),@CustomerID,'Please include fork and napkin.');

SET @OrderID = LAST_INSERT_ID();

INSERT INTO lineitems_3nf (OrderID, DonutID, Quantity)
VALUES(@OrderID,@DonutID, 10);
```

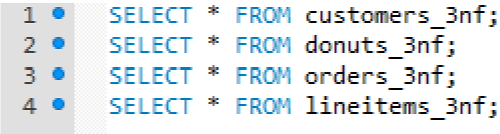


```
#  Time      Action                                                                                                                      Message
1  16:54:10  INSERT INTO customers_3nf (CustomerLastName, CustomerFirstName, CustomerStreetAddress1, CustomerStreetAddress2, CustomerCity, CustomerZip, C...  1 row(s) affected
2  16:54:10  SET @CustomerID = LAST_INSERT_ID()                                                                                          0 row(s) affected
3  16:54:10  INSERT INTO donuts_3nf (DonutName, Description, UnitPrice) VALUES('Chocolate','Chocolate Cake Donut','01.50')                1 row(s) affected
4  16:54:10  SET @DonutID = LAST_INSERT_ID()                                                                                             0 row(s) affected
5  16:54:10  INSERT INTO orders_3nf (OrderDate, OrderID, SpecialHandlingNotes) VALUES(NOW(),@CustomerID,'Please include fork and napkin.')  1 row(s) affected
6  16:54:10  SET @OrderID = LAST_INSERT_ID()                                                                                             0 row(s) affected
7  16:54:10  INSERT INTO lineitems_3nf (OrderID, DonutID, Quantity) VALUES(@OrderID,@DonutID, 10)                                        1 row(s) affected
```

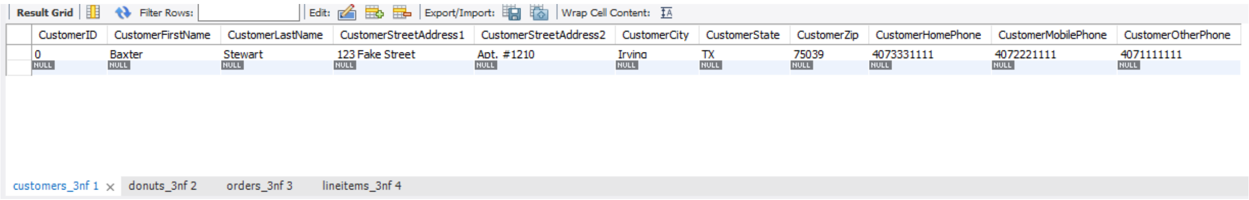
Timothy Clark
C170: Data Management – Applications (MySQL)

G) Values & Complex Join Statement

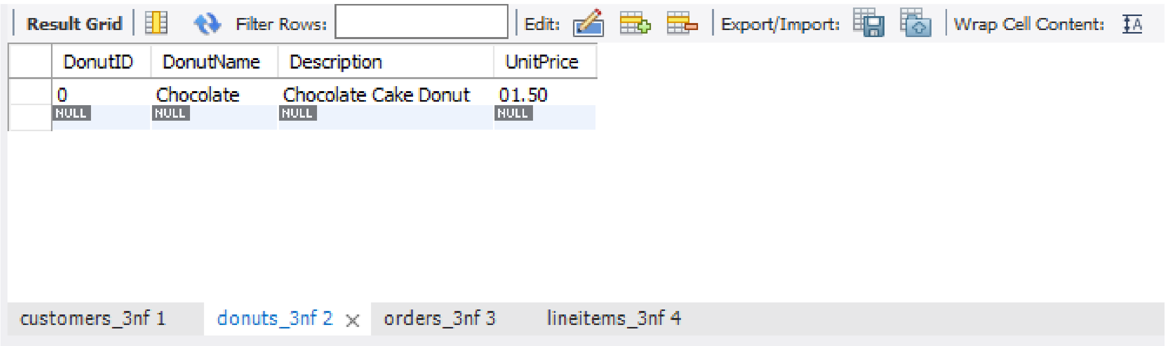
Basic Queries



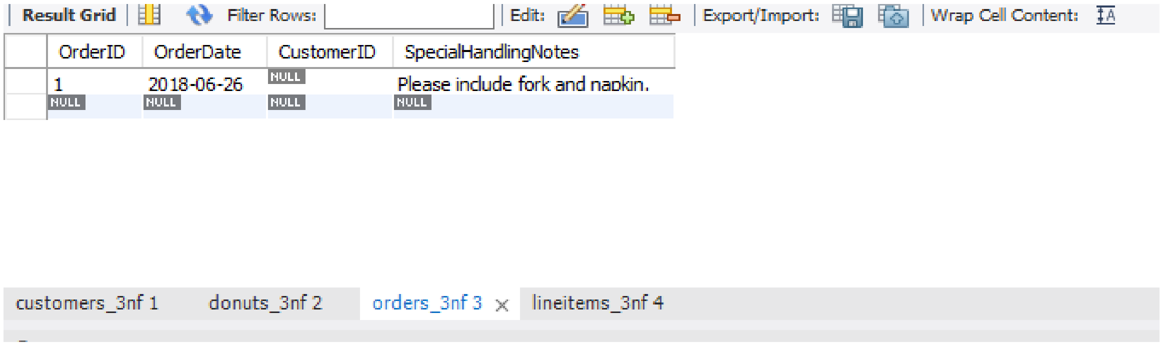
```
SELECT * FROM customers_3nf;
SELECT * FROM donuts_3nf;
SELECT * FROM orders_3nf;
SELECT * FROM lineitems_3nf;
```



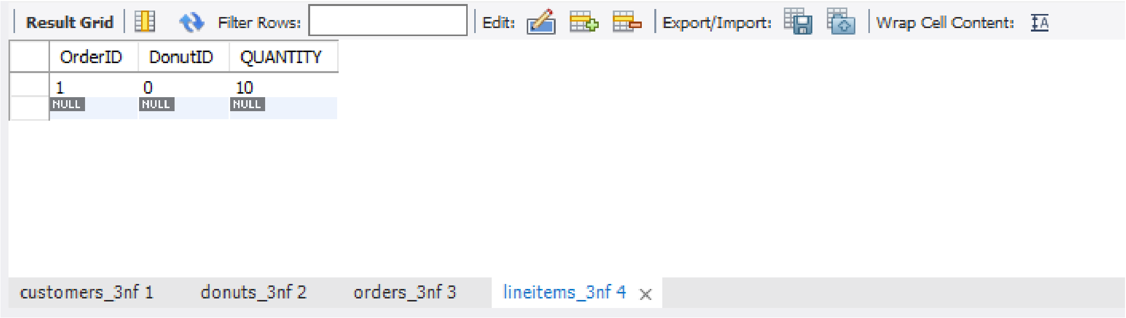
CustomerID	CustomerFirstName	CustomerLastName	CustomerStreetAddress1	CustomerStreetAddress2	CustomerCity	CustomerState	CustomerZip	CustomerHomePhone	CustomerMobilePhone	CustomerOtherPhone
0	Baxter	Stewart	123 Fake Street	Apt. #1210	Irving	TX	75039	4073331111	4072221111	4071111111
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



DonutID	DonutName	Description	UnitPrice
0	Chocolate	Chocolate Cake Donut	01.50
NULL	NULL	NULL	NULL



OrderID	OrderDate	CustomerID	SpecialHandlingNotes
1	2018-06-26	NULL	Please include fork and napkin.
NULL	NULL	NULL	NULL

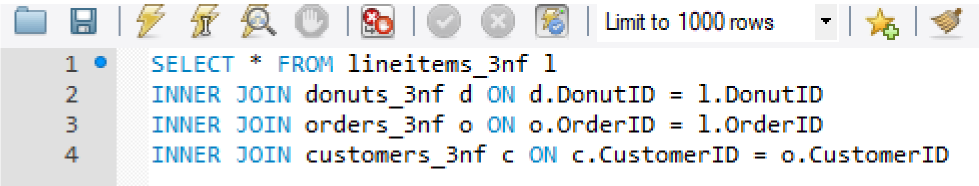


OrderID	DonutID	QUANTITY
1	0	10
NULL	NULL	NULL

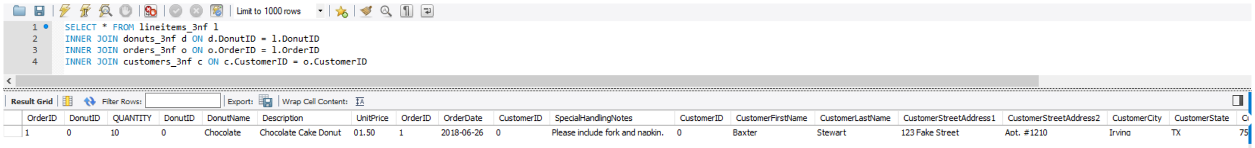
Timothy Clark
C170: Data Management – Applications (MySQL)

Complex Join Statements

Simple



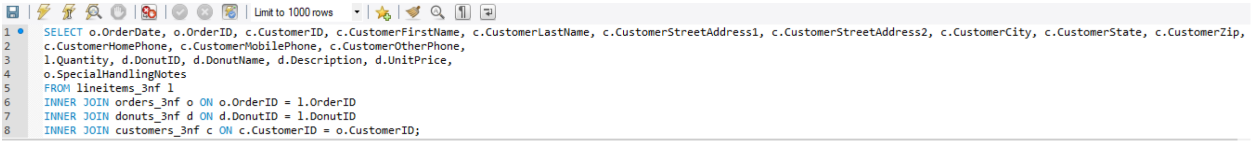
```
SELECT * FROM lineitems_3nf l
INNER JOIN donuts_3nf d ON d.DonutID = l.DonutID
INNER JOIN orders_3nf o ON o.OrderID = l.OrderID
INNER JOIN customers_3nf c ON c.CustomerID = o.CustomerID
```



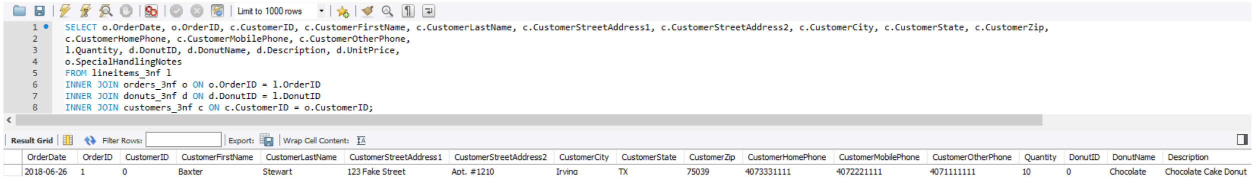
```
SELECT * FROM lineitems_3nf l
INNER JOIN donuts_3nf d ON d.DonutID = l.DonutID
INNER JOIN orders_3nf o ON o.OrderID = l.OrderID
INNER JOIN customers_3nf c ON c.CustomerID = o.CustomerID
```

OrderID	DonutID	QUANTITY	DonutID	DonutName	Description	UnitPrice	OrderID	OrderDate	CustomerID	SpecialHandlingNotes	CustomerID	CustomerFirstName	CustomerLastName	CustomerStreetAddress1	CustomerStreetAddress2	CustomerCity	CustomerState	C
1	0	10	0	Chocolate	Chocolate Cake Donut	01.50	1	2018-06-26	0	Please include fork and napkin.	0	Baxter	Stewart	123 Fake Street	Apt. #1210	Irving	TX	75

Proper



```
SELECT o.OrderDate, o.OrderID, c.CustomerID, c.CustomerFirstName, c.CustomerLastName, c.CustomerStreetAddress1, c.CustomerStreetAddress2, c.CustomerCity, c.CustomerState, c.CustomerZip,
c.CustomerHomePhone, c.CustomerMobilePhone, c.CustomerOtherPhone,
l.Quantity, d.DonutID, d.DonutName, d.Description, d.UnitPrice,
o.SpecialHandlingNotes
FROM lineitems_3nf l
INNER JOIN orders_3nf o ON o.OrderID = l.OrderID
INNER JOIN donuts_3nf d ON d.DonutID = l.DonutID
INNER JOIN customers_3nf c ON c.CustomerID = o.CustomerID;
```



```
SELECT o.OrderDate, o.OrderID, c.CustomerID, c.CustomerFirstName, c.CustomerLastName, c.CustomerStreetAddress1, c.CustomerStreetAddress2, c.CustomerCity, c.CustomerState, c.CustomerZip,
c.CustomerHomePhone, c.CustomerMobilePhone, c.CustomerOtherPhone,
l.Quantity, d.DonutID, d.DonutName, d.Description, d.UnitPrice,
o.SpecialHandlingNotes
FROM lineitems_3nf l
INNER JOIN orders_3nf o ON o.OrderID = l.OrderID
INNER JOIN donuts_3nf d ON d.DonutID = l.DonutID
INNER JOIN customers_3nf c ON c.CustomerID = o.CustomerID;
```

OrderDate	OrderID	CustomerID	CustomerFirstName	CustomerLastName	CustomerStreetAddress1	CustomerStreetAddress2	CustomerCity	CustomerState	CustomerZip	CustomerHomePhone	CustomerMobilePhone	CustomerOtherPhone	Quantity	DonutID	DonutName	Description
2018-06-26	1	0	Baxter	Stewart	123 Fake Street	Apt. #1210	Irving	TX	75039	4073331111	4072221111	4071111111	10	0	Chocolate	Chocolate Cake Donut