A1.

Nora's Bagel Bin Database Blueprints *(continued)*

Second Normal Form (2NF)

BAGEL ORDER	
PK	Bagel Order ID
	Order Date
	First Name
	Last Name
	Address 1
	Address 2
	City
	State
	Zip
	Mobile Phone
	Delivery Fee
	Special Notes

1:M

BAGEL ORDER LINE ITEM	
PK / FK	Bagel Order ID
PK / FK	Bagel ID
	Bagel Quantity

M:1

BAGEL	
PK	Bagel ID
	Bagel Name
	Bagel Description
	Bagel Price

C. The three tables made going into second normal form are the Bagel Order Table, Bagel Order Line Item Table, and Bagel Table. Regarding the Bagel Order table this table refers to everything that deals with the actual order itself. This of course would include the Order ID, Order Date, and Customer information. The Bagel table refers to all things regarding the actual bagel and not the order, this includes the Bagel ID, name, description, and price. The only attribute that refers to both primary keys is Bagel Quantity. Bagel Quantity relies on the Bagel Order ID to see how many bagels are ordered in each order and also relies on the Bagel ID primary key to refer to how many of which kind of bagel. The cardinality I found is that a single Bagel Order can order many bagel order line items. Then the next one is many bagel order line items that refer to a single bagel.

A2.

Nora's Bagel Bin Database Blueprints *(continued)*

Third Normal Form (3NF)

Bagel Order	
PK	Bagel Order ID
FK	Customer ID
	Order Date
	Delivery Fee
	Special Notes

M:1

Customer	
PK	Customer ID
	First Name
	Last Name
	Address 1
	Address 2
	City
	State
	Zip
	Mobile Phone

1:M

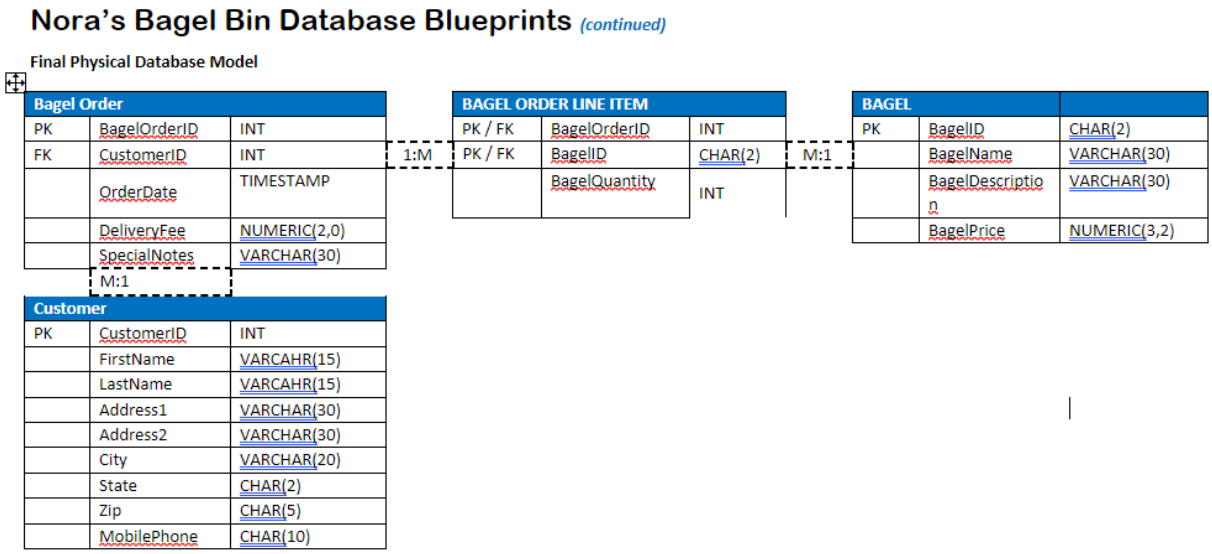
BAGEL ORDER LINE ITEM	
PK / FK	Bagel Order ID
PK / FK	Bagel ID
	Bagel Quantity

M:1

BAGEL	
PK	Bagel ID
	Bagel Name
	Bagel Description
	Bagel Price

E. Referring to the new table I made in third normal form. I made an additional table named Customer because customer information is something that can be repeated many times. I decided to make all the customer information such as name, address, zip, etc. into its own table to limit repeated data. The new cardinality I made between the two new tables is that many bagel orders can be placed by a single customer. This is true because a single customer with a single customer ID can place many bagel orders under the same customer ID.

A3.



B1.

```sql
CREATE TABLE COFFEE_SHOP (
  shop_id INT,
  shop_name VARCHAR(50),
  city VARCHAR(50),
  state CHAR(2),
  PRIMARY KEY(shop_id)
);

CREATE TABLE EMPLOYEE (
  employee_id INT,
  first_name VARCHAR(30),
  last_name VARCHAR(30),
  hire_date DATE,
  job_title VARCHAR(30),
  shop_id INT,
  PRIMARY KEY(employee_id),
  FOREIGN KEY(shop_id) REFERENCES COFFEE_SHOP(shop_id)
);

CREATE TABLE SUPPLIER (
  supplier_id INT,
  company_name VARCHAR(50),
  country VARCHAR(30),
  sales_contact_name VARCHAR(60),
  email VARCHAR(50) NOT NULL,
  PRIMARY KEY(supplier_id)
);

CREATE TABLE COFFEE (
  coffee_id INT,
  shop_id INT,
  supplier_id INT,
  coffee_name VARCHAR(30),
  price_per_pound NUMERIC(5,2),
  PRIMARY KEY(coffee_id),
  FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id),
  FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
);
```

B1.

B2.

B2.

```
SELECT *
FROM COFFEE_SHOP, EMPLOYEE, SUPPLIER, COFFEE;
```

Run SQL ▶ | Edit Fullscreen | [;]

country	sales_contact_name	email	coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
United States	Gerald	lionmanecoffee@gmail.com	301	1	201	Machiatto	25.99
United States	Gerald	lionmanecoffee@gmail.com	301	1	201	Machiatto	25.99
United States	Gerald	lionmanecoffee@gmail.com	301	1	201	Machiatto	25.99
United States	Gerald	lionmanecoffee@gmail.com	302	2	202	Frapuccino	36.99
United States	Gerald	lionmanecoffee@gmail.com	302	2	202	Frapuccino	36.99
United States	Gerald	lionmanecoffee@gmail.com	302	2	202	Frapuccino	36.99
United States	Gerald	lionmanecoffee@gmail.com	303	3	203	Medium Roast	19.99
United States	Gerald	lionmanecoffee@gmail.com	303	3	203	Medium Roast	19.99
United States	Gerald	lionmanecoffee@gmail.com	303	3	203	Medium Roast	19.99
United States	Gerald	lionmanecoffee@gmail.com	301	1	201	Machiatto	25.99
Brazil	Tom	baristalovers@yahoo.com	301	1	201	Machiatto	25.99
Brazil	Tom	baristalovers@yahoo.com	301	1	201	Machiatto	25.99

B3.

```
CREATE VIEW EmployeeFullName
   AS SELECT employee_id, CONCAT(first_name, " ", last_name) AS employee_full_name, hire_date, job_title,shop_id
    from EMPLOYEE;
```

B3.

employee_id	employee_full_name	hire_date	job_title	shop_id
101	John Daily	2005-01-01	Cashier	1
102	Bobby Johnson	2008-05-15	Barista	2
103	Frank Smith	2009-06-12	Manager	3

B4.

```
CREATE INDEX CoffeeName
ON COFFEE (coffee_name);
```

Build Schema | Edit Fullscreen | Browser | [;]

Run SQL | Edit Fullscreen | [;]

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Com
coffee	0	PRIMARY	1	coffee_id	A	3	(null)	(null)		BTREE	
coffee	1	shop_id	1	shop_id	A	3	(null)	(null)	YES	BTREE	
coffee	1	supplier_id	1	supplier_id	A	3	(null)	(null)	YES	BTREE	
coffee	1	CoffeeName	1	coffee_name	A	3	(null)	(null)	YES	BTREE	

B5.

```
SELECT supplier_id, company_name
FROM SUPPLIER
WHERE country = 'United States';
```

B5.

supplier_id	company_name
201	Lion Mane
203	McDonalds

B6.

```
SELECT COFFEE_SHOP.shop_id, COFFEE_SHOP.shop_name, COFFEE_SHOP.city, EMPLOYEE.employee_id, EMPLOYEE.first_name, EMPLOYEE.last_name
FROM EMPLOYEE
JOIN COFFEE_SHOP
ON EMPLOYEE.shop_id = COFFEE_SHOP.shop_id
JOIN COFFEE
ON COFFEE.shop_id = COFFEE_SHOP.shop_id
```

B6.

shop_id	shop_name	city	employee_id	first_name	last_n
1	Starbucks	Hickory	101	John	Daily
2	Dunkin Donuts	Conover	102	Bobby	Johns
3	Coffee By The Bay	Statesville	103	Frank	Smith