

# **PART A**

## **Nora's Bagel Bin**



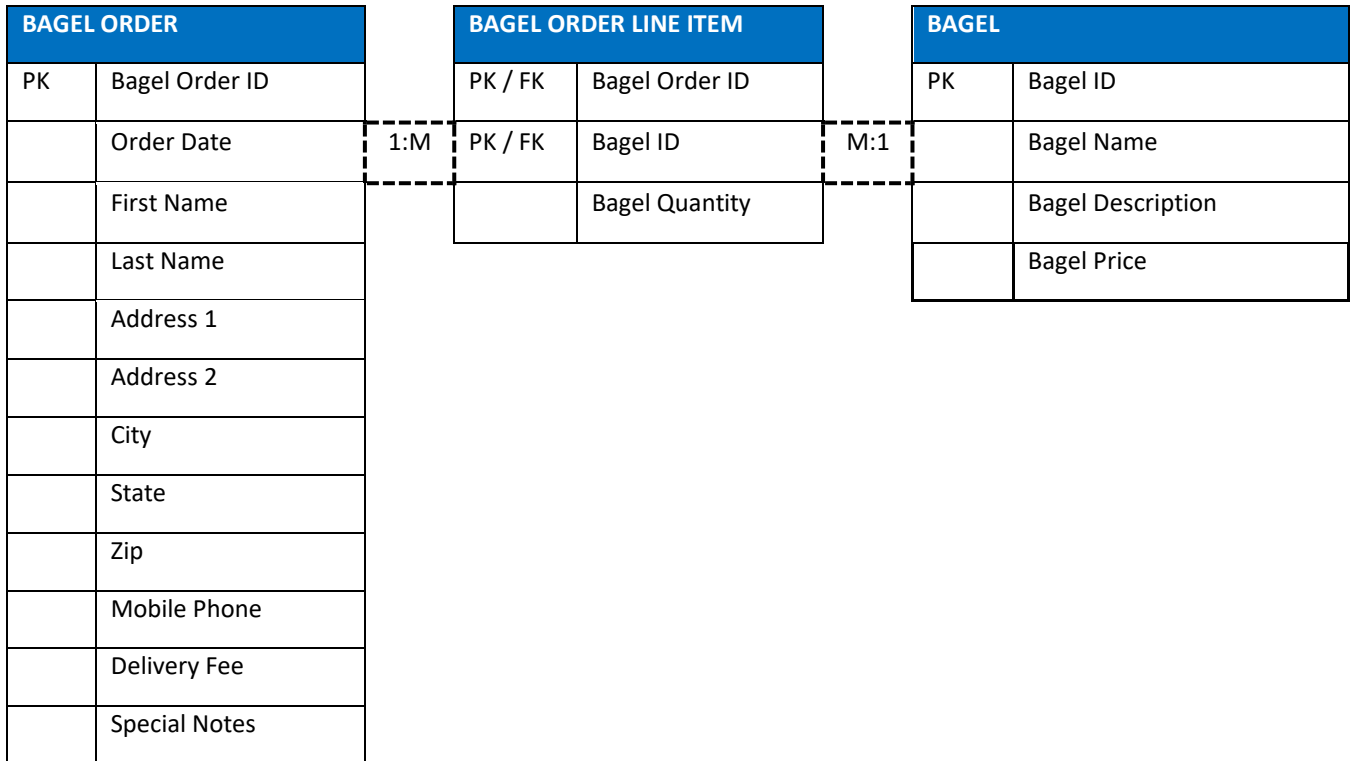
## Nora's Bagel Bin Database Blueprints

### First Normal Form (1NF)

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

# Nora's Bagel Bin Database Blueprints

## Second Normal Form (2NF)



### A.1.c (explanation):

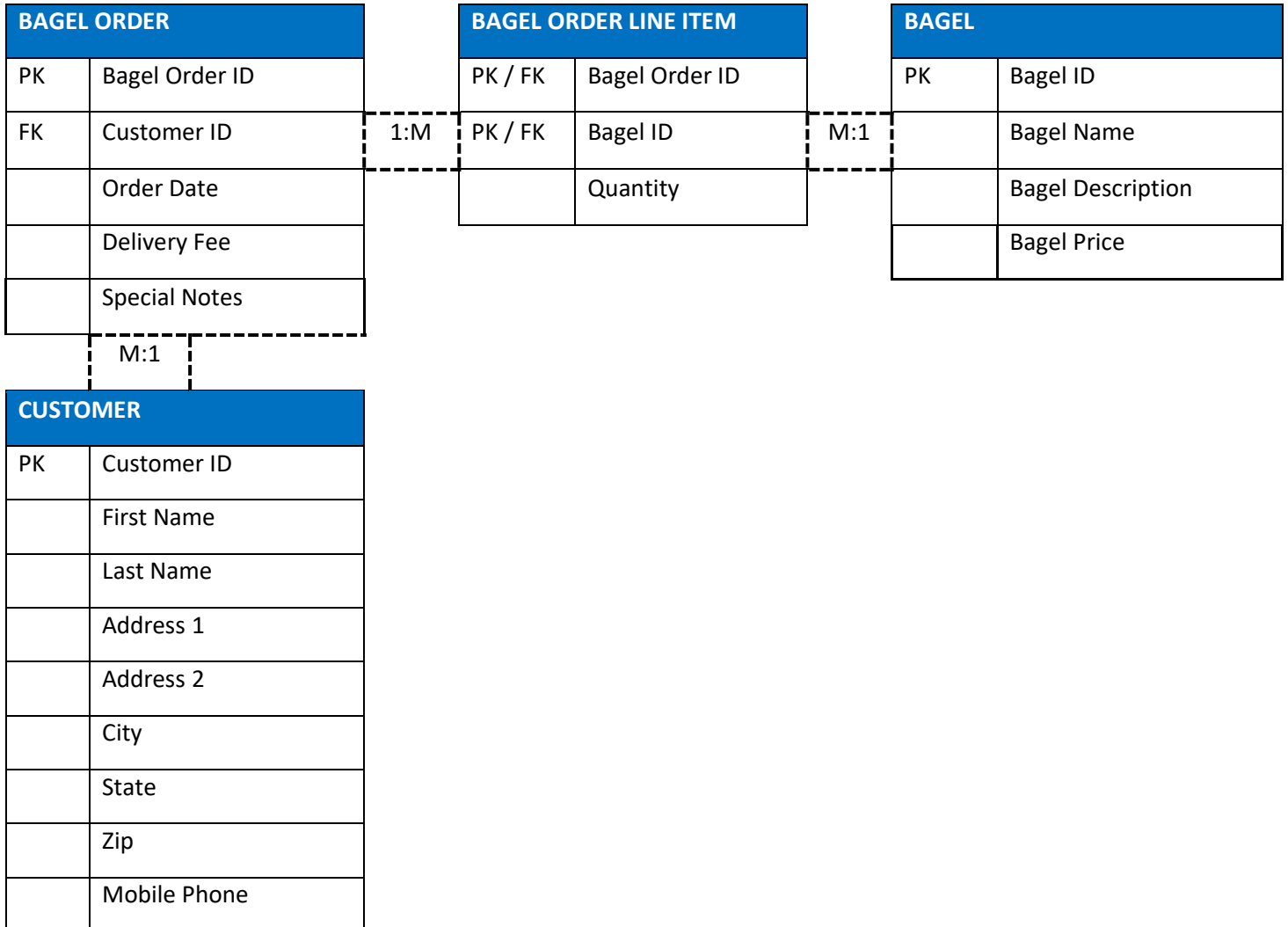
To transform this to 2nd Normal Form we need to make it so that each non-key column is dependent on the whole composite primary key, not just part of it. To be specific, the first table which is in 1st Normal Form includes some non-key columns that are dependent on only part of the composite primary key. For example, Bagel Price (and other bagel information) is dependent solely on Bagel ID and not Bagel Order ID.

To fix this we need to break the table up, separating the Bagel Order information from the Bagel information. However, since "Bagel Order" and "Bagel" have a relationship cardinality of many-to-many, a third linking table needs to be added as well. This linking table was called "Bagel Order Line Item" and it includes Bagel Order ID, Bagel ID, and the quantity of bagels ordered. Bagel Order Line Item has a composite primary key (Bagel Order ID, Bagel ID). The Bagel Order table primary key is Bagel Order ID. The Bagel table primary key is Bagel ID.

The Bagel Order and Bagel tables both have a cardinality of one-to-many with Bagel Order Line Item. This means that each Bagel Order Line Item will relate to one and only one Bagel Order and one and only one Bagel. Bagel Order and Bagel entries can be in many different Bagel Order Line Item entries.

# Nora's Bagel Bin Database Blueprints

## Third Normal Form (3NF)

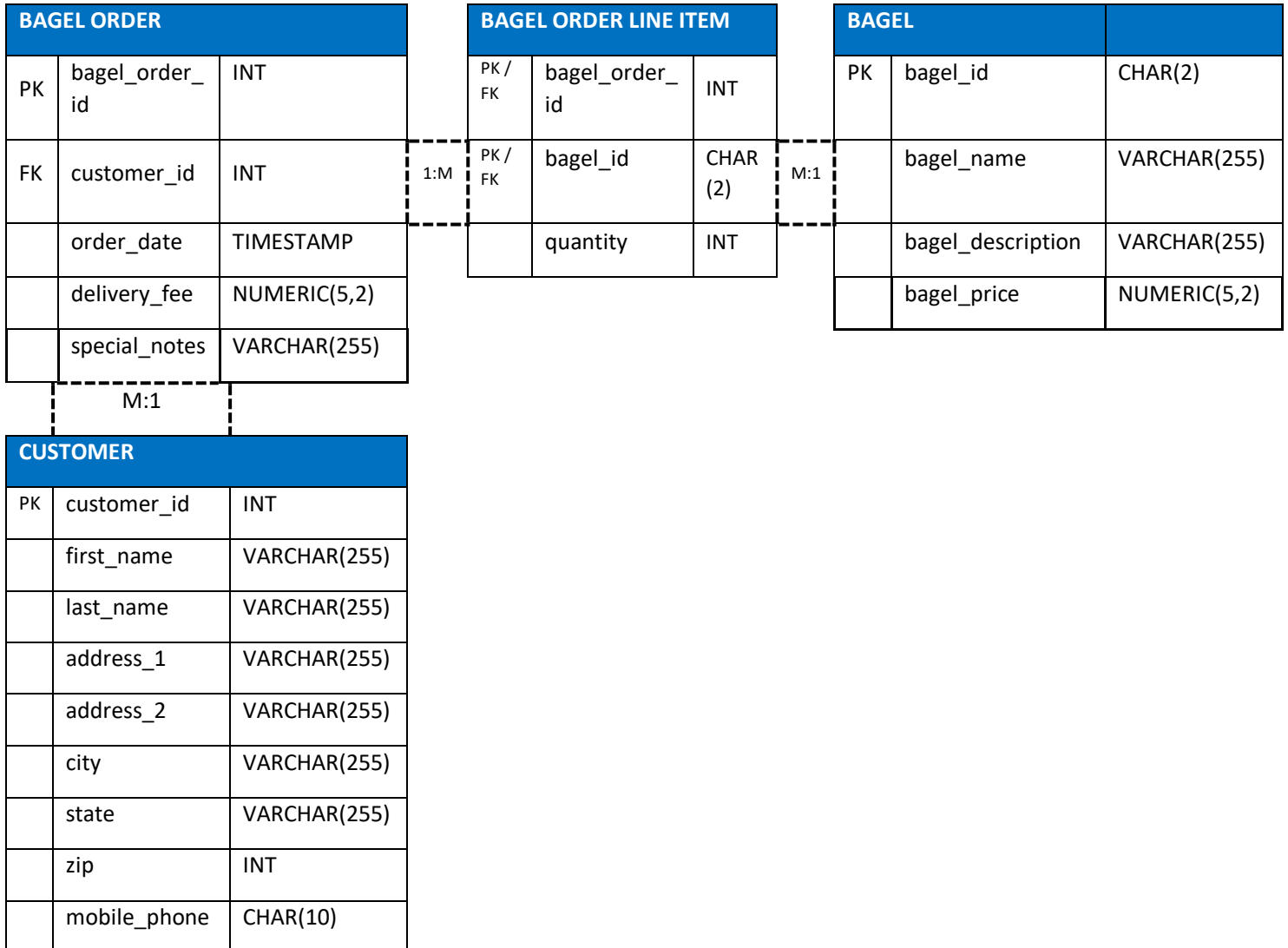


### A.2.e (explanation):

Because there were still redundancies surrounding the customer data in the Bagel Order table, an additional table ("Customer") was added which incorporates all information specific to customers, with 'Customer ID' acting as the primary key (as well as being a foreign key in the updated Bagel Order table). This leaves the Bagel Order table with no redundancies and brings everything to 3rd Normal Form, since none of the non-key columns depend on another non-key column. The relationship between Bagel Order, Bagel Order Line Item, and Bagel remain the same as in the 2nd Normal Form version. The cardinality of Bagel Order and Customer is Many to One, each Customer can correspond with many Bagel Orders, but each Bagel Order can correspond with one and only one Customer.

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

## Final Physical Database Model



# **PART A**

**Jaunty Coffee Co.**

### B.1.a - Creating Tables - SQL Code

SQL Fiddle 


```
1
2 CREATE TABLE COFFEE_SHOP (
3     shop_id INT,
4     shop_name VARCHAR(50),
5     city VARCHAR(50),
6     state CHAR(2),
7     PRIMARY KEY (shop_id)
8 );
9
10 CREATE TABLE EMPLOYEE (
11     employee_id INT,
12     first_name VARCHAR(30),
13     last_name VARCHAR(30),
14     hire_date DATE,
15     job_title VARCHAR(30),
16     shop_id INT,
17     PRIMARY KEY (employee_id),
18     FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP (shop_id)
19 );
20
21 CREATE TABLE SUPPLIER (
22     supplier_id INT,
23     company_name VARCHAR(50),
24     country VARCHAR(50),
25     sales_contact_name VARCHAR(50),
26     email VARCHAR(50),
27     PRIMARY KEY (supplier_id)
28 );
29
30
31 CREATE TABLE COFFEE (
32     coffee_id INT,
33     shop_id INT,
34     supplier_id INT,
35     coffee_name VARCHAR(30),
36     price_per_pound NUMERIC(5,2),
37     PRIMARY KEY (coffee_id),
38     FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id),
39     FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
40 );
41
```

### B.1.b: Creating Tables - Database Server's Response

SQL Fiddle 

MySQL 5.6 ▾

 View Sample Fiddle

 Clear

 Text to DDL

## Schema Browser

- **coffee** (TABLE)
  - coffee\_id INT(10)
  - shop\_id INT(10)
  - supplier\_id INT(10)
  - coffee\_name VARCHAR(30)
  - price\_per\_pound DECIMAL(5)
- **coffee\_shop** (TABLE)
  - shop\_id INT(10)
  - shop\_name VARCHAR(50)
  - city VARCHAR(50)
  - state CHAR(2)
- **employee** (TABLE)
  - employee\_id INT(10)
  - first\_name VARCHAR(30)
  - last\_name VARCHAR(30)
  - hire\_date DATE(10)
  - job\_title VARCHAR(30)
  - shop\_id INT(10)
- **supplier** (TABLE)
  - supplier\_id INT(10)
  - company\_name VARCHAR(50)
  - country VARCHAR(50)
  - sales\_contact\_name VARCHAR(50)
  - email VARCHAR(50)

DDL Editor 


✓ Schema Ready



### B.2.a: Populating Database - SQL Code

```
42 #B.2 Populating the Database
43 INSERT INTO COFFEE_SHOP
44 VALUES ('1', 'Stella Brew', 'Lee Vining', 'CA'),
45          ('2', 'Cartel', 'Tempe', 'AZ'),
46          ('3', 'Looney Bean', 'Bishop', 'CA');
47
48 INSERT INTO EMPLOYEE
49 VALUES ('1', 'Tyler', 'Meester', '2022-07-01', 'Server', '2'),
50          ('2', 'Saxon', 'Richardson', '2022-05-16', 'Server', '3'),
51          ('3', 'Hilary', 'Lempit', '2022-04-01', 'Barista', '1');
52
53 INSERT INTO SUPPLIER
54 VALUES ('1', 'Cofee Inc.', 'United States', 'Jeremy Cofe', 'cofefanatic@gmail.com'),
55          ('2', 'Grounded', 'Columbia', 'Pablo Ground', 'groundedcoffee@gmail.com'),
56          ('3', 'Brewtiful', 'Ecuador', 'Maria Brews', 'brewtiful@gmail.com');
57
58 INSERT INTO COFFEE
59 VALUES ('1', '3', '2', 'Ground Columbian', '5'),
60          ('2', '2', '1', 'Fanatical Light Roast', '7'),
61          ('3', '1', '3', 'Ecuadorian Brew', '9');
62
```

### B.2.b: Populating Database - SQL Query



```
1 SELECT *
2 FROM COFFEE_SHOP;
3
4 SELECT *
5 FROM EMPLOYEE;
6
7 SELECT *
8 FROM SUPPLIER;
9
10 SELECT *
11 FROM COFFEE;
12
```

### B.2.b: Populating Database - Database Server's Response

shop_id	shop_name	city	state
1	Stella Brew	Lee Vining	CA
2	Cartel	Tempe	AZ
3	Looney Bean	Bishop	CA

✔ Record Count: 3; Execution Time: 2ms + [View Execution Plan](#) ➔ [link](#)

employee_id	first_name	last_name	hire_date	job_title	shop_id
1	Tyler	Meester	2022-07-01	Server	2
2	Saxon	Richardson	2022-05-16	Server	3
3	Hilary	Lempit	2022-04-01	Barista	1

✔ Record Count: 3; Execution Time: 1ms + [View Execution Plan](#) ➔ [link](#)

supplier_id	company_name	country	sales_contact_name	email
1	Cofee Inc.	United States	Jeremy Cofe	cofefanatic@gmail.com
2	Grounded	Columbia	Pablo Ground	groundedcoffee@gmail.com
3	Brewtiful	Ecuador	Maria Brews	brewtiful@gmail.com

✔ Record Count: 3; Execution Time: 3ms + [View Execution Plan](#) ➔ [link](#)

coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
1	3	2	Ground Columbian	5
2	2	1	Fanatical Light Roast	7
3	1	3	Ecuadorian Brew	9

✔ Record Count: 3; Execution Time: 2ms + [View Execution Plan](#) ➔ [link](#)

### B.3.a: Creating View Table - SQL Code

```
64 #B3.a Creating a View Table
65 CREATE VIEW employee_view_table AS
66 SELECT
67     employee_id,
68     first_name,
69     last_name,
70     hire_date,
71     job_title,
72     shop_id,
73     CONCAT(first_name, ' ', last_name) AS employee_full_name
74 FROM EMPLOYEE;
75
76
```

### B.3.b: Creating View Table - SQL Query

```
1 SELECT *
2 FROM employee_view_table;
3
4
```

### B.3.b: Creating View Table - Database Server's Response

employee_id	first_name	last_name	hire_date	job_title	shop_id	employee_full_name
1	Tyler	Meester	2022-07-01	Server	2	Tyler Meester
2	Saxon	Richardson	2022-05-16	Server	3	Saxon Richardson
3	Hilary	Lempit	2022-04-01	Barista	1	Hilary Lempit

✓ Record Count: 3; Execution Time: 6ms + [View Execution Plan](#) ➔ [link](#)

#### B.4.a: Creating Index - SQL Code

```
76 #B4.a
77 CREATE INDEX idx_coffee_name
78 ON COFFEE(coffee_name);
79
```

#### B.4.b Creating Index - SQL Query

```
1 SHOW INDEX
2 FROM COFFEE;
```

#### B.4.b Creating Index - Database Server's Response

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
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	idx_coffee_name	1	coffee_name	A	3	(null)	(null)	YES	BTREE		

✓ Record Count: 4; Execution Time: 15ms ➔ [link](#)

#### B.5.a: SFW (SELECT-FROM-WHERE) - SQL Query

```
1 SELECT coffee_id, coffee_name, price_per_pound
2 FROM COFFEE
3 WHERE supplier_id = 2;
4
5
```

#### B.5.b: SFW (SELECT-FROM-WHERE) - Database Server's Response

coffee_id	coffee_name	price_per_pound
1	Ground Columbian	5

✓ Record Count: 1; Execution Time: 17ms + [View Execution Plan](#) ➔ [link](#)

#### B.6.a: Table Join - SQL Query

```
1 SELECT COFFEE_SHOP.shop_id,
2         COFFEE_SHOP.shop_name,
3         COFFEE.coffee_name,
4         SUPPLIER.country as coffee_source
5 FROM COFFEE_SHOP
6 JOIN COFFEE ON COFFEE_SHOP.shop_id=COFFEE.shop_id
7 JOIN SUPPLIER ON COFFEE.supplier_id=SUPPLIER.supplier_id;
8
```

#### B.6.b - Table Join - Database Server's Response

shop_id	shop_name	coffee_name	coffee_source
1	Stella Brew	Ecuadorian Brew	Ecuador
2	Cartel	Fanatical Light Roast	United States
3	Looney Bean	Ground Columbian	Columbia

✓ Record Count: 3; Execution Time: 6ms + [View Execution Plan](#) ➔ [link](#)