# 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 *(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

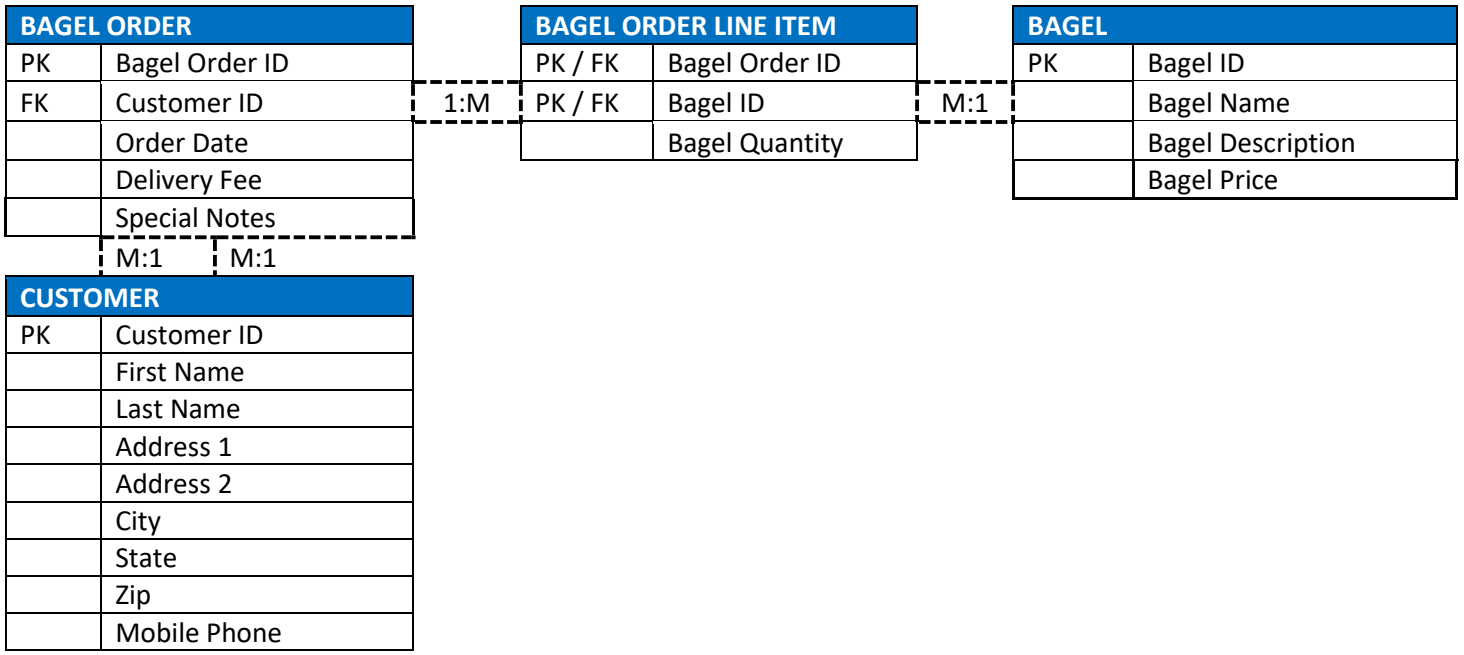
| BAGEL | |
|---|---|
| PK | Bagel ID |
| | Bagel Name |
| | Bagel Description |
| | Bagel Price |

For converting the *attributes* into second normal form I started with the Bagel Order table and assigned all the attributes that would/should be connected to the primary key, Bagel Order ID. This, of course, includes the order date, the customers information, the delivery fee, and any special notes as all this information pertains to the order itself. The Bagel Order Line-Item table only needed the Bagel Quantity attribute to be added, as the Bagel Order will rely on the quantity to determine the price, and the Bagel table includes the price, therefore, the Bagel Quantity Attribute should be on the line-item table. The Bagel Table includes all the bagel-specific information as this is what displays on each line item as shown in the Bagel Order Form provided.

For *cardinality*, each order can contain many line items, but a line item will only be attached to one order; so, Bagel Order to Bagel Order Line Item has a cardinality of one-to-many. While *technically* each line-item can contain multiple bagels, the quantity is specified in the Bagel Order Line-Item table. Therefore, a line-item will only feature one type of bagel, but a bagel could belong to many line-items, leading to a cardinality of many-to-one.

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

**Third Normal Form (3NF)**




To convert the database into third normal form, the non-key columns should depend on the key and nothing else, therefore the customer information should be moved away from the Bagel Order table and into its own customer table. On top of the increased clarity and modularity, this would also make it easier for the business to hold customer information and their previous orders in the system for future use. To make this change, I created a new primary key (Customer ID) for the Customer table and moved all the customer information attributes to the new table. Since the order needs to have a reference to the customer placing the order, the customer tables primary key is referenced as a foreign key on the Bagel Order table. Assuming the business holds some customer information, a customer could have many Bagel Orders but based on the order form provided, a Bagel Order will only have one customer. Therefore, Bagel Order and Customer have a cardinality of many-to-one.

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

**Final Physical Database Model**

| BAGEL ORDER | | |
|---|---|---|
| PK | bagel_order_id | INT |
| FK | customer_id | INT |
| | order_date | TIMESTAMP |
| | delivery_fee | INT |
| | special_notes | VARCHAR(40) |

1:M

| BAGEL ORDER LINE ITEM | | |
|---|---|---|
| PK / FK | bagel_order_id | INT |
| PK / FK | bagel_id | CHAR(2) |
| | bagel_quantity | INT |

M:1

| BAGEL | | |
|---|---|---|
| PK | bagel_id | CHAR(2) |
| | bagel_name | VARCHAR(15) |
| | bagel_desc | VARCHAR(30) |
| | bagel_price | NUMERIC(2,2) |

M:1 M:1

| CUSTOMERS | | |
|---|---|---|
| PK | customer_id | INT |
| | first_name | VARCHAR(20) |
| | last_name | VARCHAR(20) |
| | address_1 | VARCHAR(40) |
| | address_2 | VARCHAR(30) |
| | city | VARCHAR(30) |
| | state | VARCHAR(30) |
| | zip | CHAR(5) |
| | mobile_phone | CHAR(9) |

# Jaunty Coffee Co. ERD Database

**B.**

## 1. Develop SQL Code to Create Each Table

```
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));
```

Build Schema | Edit Fullscreen | Browser | [ ; ]

```
SHOW TABLES;
```

Run SQL | Edit Fullscreen | [ ; ]

| Tables_in_db_9_6842b4 |
| --- |
| coffee |
| coffee_shop |
| employee |
| employee_view |
| supplier |

Record Count: 5; Execution Time: 7ms link

## 2. Develop SQL Code to Populate Each Table

Insert Statements:

```sql
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));

-- POPULATING TABLES --
INSERT INTO Coffee_Shop
VALUES (0, 'Wake Up Call Coffee', 'Post Falls', 'ID'),
       (1, 'Starbucks', 'Seattle', 'WA'),
       (2, 'Jacobs Java', 'Spokane Valley', 'WA');

INSERT INTO Employee
VALUES (0, 'James', 'Dunaway', '2019-11-13', 'Assistant Manager', 1),
       (1, 'Steven', 'Jones', '2019-10-04', 'Manager', 1),
       (2, 'Allison', 'Garcia', '2021-12-30', 'Manager', 0),
       (3, 'Jordan', 'Clark', '2020-03-19', 'Barista', 0),
       (4, 'Zakry', 'Wilcox', '2021-02-10', 'Barista', 1),
       (5, 'Olivia', 'Wilson', '2019-11-21', 'Barista', 2),
       (6, 'Wilson', 'Owens', '2022-04-01', 'Manager', 2);

INSERT INTO Supplier
VALUES (0, 'Coffee-R-Us', 'USA', 'Steve Smith', 'stevesmith@coffeerus.com'),
       (1, 'Roasters INC', 'Brazil', 'Bob Chair', 'bobchair@roastersinc.com'),
       (2, 'Alligator Coffee Co', 'Kenya', 'Starlon Stones', 'starlon@alligatorcoffeeco.com');

INSERT INTO Coffee
VALUES (0, 0, 0, 'Arabica Boralis', 2.50),
       (1, 1, 1, 'Coffee Robusta', 2.10),
       (2, 2, 2, 'Beans of Liberica', 2.19);
```

Build Schema | Edit Fullscreen | Browser | [ ; ]

```sql
SELECT * FROM Coffee_Shop;

SELECT * FROM Employee;

SELECT * FROM Supplier;

SELECT * FROM Coffee;
```

Run SQL | Edit Fullscreen | [ ; ]

## Select Query Result:

| shop_id | shop_name | city | state |
|---|---|---|---|
| 0 | Wake Up Call Coffee | Post Falls | ID |
| 1 | Starbucks | Seattle | WA |
| 2 | Jacobs Java | Spokane Valley | WA |

Record Count: 3; Execution Time: 4ms  View Execution Plan  link

| employee_id | first_name | last_name | hire_date | job_title | shop_id |
|---|---|---|---|---|---|
| 0 | James | Dunaway | 2019-11-13 | Assistant Manager | 1 |
| 1 | Steven | Jones | 2019-10-04 | Manager | 1 |
| 2 | Allison | Garcia | 2021-12-30 | Manager | 0 |
| 3 | Jordan | Clark | 2020-03-19 | Barista | 0 |
| 4 | Zakry | Wilcox | 2021-02-10 | Barista | 1 |
| 5 | Olivia | Wilson | 2019-11-21 | Barista | 2 |
| 6 | Wilson | Owens | 2022-04-01 | Manager | 2 |

Record Count: 7; Execution Time: 1ms  View Execution Plan  link

| supplier_id | company_name | country | sales_contact_name | email |
|---|---|---|---|---|
| 0 | Coffee-R-Us | USA | Steve Smith | stevesmith@coffeerus.com |
| 1 | Roasters INC | Brazil | Bob Chair | bobchair@roastersinc.com |
| 2 | Alligator Coffee Co | Kenya | Starlon Stones | starlon@alligatorcoffeeco.com |

Record Count: 3; Execution Time: 0ms  View Execution Plan  link

| coffee_id | shop_id | supplier_id | coffee_name | price_per_pound |
|---|---|---|---|---|
| 0 | 0 | 0 | Arabica Boralis | 2.5 |
| 1 | 1 | 1 | Coffee Robusta | 2.1 |
| 2 | 2 | 2 | Beans of Liberica | 2.19 |

Record Count: 3; Execution Time: 1ms  View Execution Plan  link

## 3. Develop SQL Code to Create a View

```
  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));

-- POPULATING TABLES --
INSERT INTO Coffee_Shop
VALUES (0, 'Wake Up Call Coffee', 'Post Falls', 'ID'),
       (1, 'Starbucks', 'Seattle', 'WA'),
       (2, 'Jacobs Java', 'Spokane Valley', 'WA');

INSERT INTO Employee
VALUES (0, 'James', 'Dunaway', '2019-11-13', 'Assistant Manager', 1),
       (1, 'Steven', 'Jones', '2019-10-04', 'Manager', 1),
       (2, 'Allison', 'Garcia', '2021-12-30', 'Manager', 0),
       (3, 'Jordan', 'Clark', '2020-03-19', 'Barista', 0),
       (4, 'Zakry', 'Wilcox', '2021-02-10', 'Barista', 1),
       (5, 'Olivia', 'Wilson', '2019-11-21', 'Barista', 2),
       (6, 'Wilson', 'Owens', '2022-04-01', 'Manager', 2);

INSERT INTO Supplier
VALUES (0, 'Coffee-R-Us', 'USA', 'Steve Smith', 'stevesmith@coffeerus.com'),
       (1, 'Roasters INC', 'Brazil', 'Bob Chair', 'bobchair@roastersinc.com'),
       (2, 'Alligator Coffee Co', 'Kenya', 'Starlon Stones', 'starlon@alligatorcoffeeco.com');

INSERT INTO Coffee
VALUES (0, 0, 0, 'Arabica Boralis', 2.50),
       (1, 1, 1, 'Coffee Robusta', 2.10),
       (2, 2, 2, 'Beans of Liberica', 2.19);

CREATE VIEW Employee_View AS
SELECT employee_id, CONCAT(first_name, ' ', last_name) AS employee_full_name, hire_date, job_title, shop_id
FROM Employee;
```

Build Schema | Edit Fullscreen | Browser | [ ; ]

```
SELECT * FROM Employee_View;
```

Run SQL | Edit Fullscreen | [ ; ]

| employee_id | employee_full_name | hire_date | job_title | shop_id |
|---|---|---|---|---|
| 0 | James Dunaway | 2019-11-13 | Assistant Manager | 1 |
| 1 | Steven Jones | 2019-10-04 | Manager | 1 |
| 2 | Allison Garcia | 2021-12-30 | Manager | 0 |
| 3 | Jordan Clark | 2020-03-19 | Barista | 0 |
| 4 | Zakry Wilcox | 2021-02-10 | Barista | 1 |
| 5 | Olivia Wilson | 2019-11-21 | Barista | 2 |
| 6 | Wilson Owens | 2022-04-01 | Manager | 2 |

Record Count: 7; Execution Time: 6ms | View Execution Plan | link

## 4. Develop SQL Code to Create an Index on the ‘coffee_name’ Field

```
  FOREIGN KEY (shop_id) REFERENCES Coffee_Shop(shop_id),
  FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id));

-- POPULATING TABLES --
INSERT INTO Coffee_Shop
VALUES (0, 'Wake Up Call Coffee', 'Post Falls', 'ID'),
       (1, 'Starbucks', 'Seattle', 'WA'),
       (2, 'Jacobs Java', 'Spokane Valley', 'WA');

INSERT INTO Employee
VALUES (0, 'James', 'Dunaway', '2019-11-13', 'Assistant Manager', 1),
       (1, 'Steven', 'Jones', '2019-10-04', 'Manager', 1),
       (2, 'Allison', 'Garcia', '2021-12-30', 'Manager', 0),
       (3, 'Jordan', 'Clark', '2020-03-19', 'Barista', 0),
       (4, 'Zakry', 'Wilcox', '2021-02-10', 'Barista', 1),
       (5, 'Olivia', 'Wilson', '2019-11-21', 'Barista', 2),
       (6, 'Wilson', 'Owens', '2022-04-01', 'Manager', 2);

INSERT INTO Supplier
VALUES (0, 'Coffee-R-Us', 'USA', 'Steve Smith', 'stevesmith@coffeerus.com'),
       (1, 'Roasters INC', 'Brazil', 'Bob Chair', 'bobchair@roastersinc.com'),
       (2, 'Alligator Coffee Co', 'Kenya', 'Starlon Stones', 'starlon@alligatorcoffeeco.com');

INSERT INTO Coffee
VALUES (0, 0, 0, 'Arabica Boralis', 2.50),
       (1, 1, 1, 'Coffee Robusta', 2.10),
       (2, 2, 2, 'Beans of Liberica', 2.19);

CREATE VIEW Employee_View AS
SELECT employee_id, CONCAT(first_name, ' ', last_name) AS employee_full_name, hire_date, job_title, shop_id
FROM Employee;

ALTER TABLE Coffee ADD INDEX coffee_index (coffee_name);
```

Build Schema | Edit Fullscreen | Browser | [ , ]

Run SQL | Edit Fullscreen | [ , ]

**Query Panel**

Use this panel to try to solve the problem with other SQL statements (SELECTs, etc...). Results will be displayed below. Share your queries by copying and pasting the URL that is generated after each run.

Schema Ready

## 5. Develop SQL Code to Create an SFW Query

```
  PRIMARY KEY (coffee_id),
  FOREIGN KEY (shop_id) REFERENCES Coffee_Shop(shop_id),
  FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id));

-- POPULATING TABLES --
INSERT INTO Coffee_Shop
VALUES (0, 'Wake Up Call Coffee', 'Post Falls', 'ID'),
       (1, 'Starbucks', 'Seattle', 'WA'),
       (2, 'Jacobs Java', 'Spokane Valley', 'WA');

INSERT INTO Employee
VALUES (0, 'James', 'Dunaway', '2019-11-13', 'Assistant Manager', 1),
       (1, 'Steven', 'Jones', '2019-10-04', 'Manager', 1),
       (2, 'Allison', 'Garcia', '2021-12-30', 'Manager', 0),
       (3, 'Jordan', 'Clark', '2020-03-19', 'Barista', 0),
       (4, 'Zakry', 'Wilcox', '2021-02-10', 'Barista', 1),
       (5, 'Olivia', 'Wilson', '2019-11-21', 'Barista', 2),
       (6, 'Wilson', 'Owens', '2022-04-01', 'Manager', 2);

INSERT INTO Supplier
VALUES (0, 'Coffee-R-Us', 'USA', 'Steve Smith', 'stevesmith@coffeerus.com'),
       (1, 'Roasters INC', 'Brazil', 'Bob Chair', 'bobchair@roastersinc.com'),
       (2, 'Alligator Coffee Co', 'Kenya', 'Starlon Stones', 'starlon@alligatorcoffeeco.com');

INSERT INTO Coffee
VALUES (0, 0, 0, 'Arabica Boralis', 2.50),
       (1, 1, 1, 'Coffee Robusta', 2.10),
       (2, 2, 2, 'Beans of Liberica', 2.19);

CREATE VIEW Employee_View AS
SELECT employee_id, CONCAT(first_name, ' ', last_name) AS employee_full_name, hire_date, job_title, shop_id
FROM Employee;

ALTER TABLE Coffee ADD INDEX coffee_index (coffee_name);
```

Build Schema | Edit Fullscreen | Browser | [ ; ]

```
SELECT employee_id AS ID,
       employee_full_name AS Name,
       hire_date AS Hired,
       job_title AS Title,
       shop_id AS Shop
FROM Employee_View
WHERE shop_id != 2;
```

Run SQL | Edit Fullscreen | [ ; ]

| ID | Name | Hired | Title | Shop |
|---|---|---|---|---|
| 0 | James Dunaway | 2019-11-13 | Assistant Manager | 1 |
| 1 | Steven Jones | 2019-10-04 | Manager | 1 |
| 2 | Allison Garcia | 2021-12-30 | Manager | 0 |
| 3 | Jordan Clark | 2020-03-19 | Barista | 0 |
| 4 | Zakry Wilcox | 2021-02-10 | Barista | 1 |

Record Count: 5; Execution Time: 4ms | View Execution Plan | link

## 6. Develop SQL Code to Create a Query by Doing the Following

```
  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));

-- POPULATING TABLES --
INSERT INTO Coffee_Shop
VALUES (0, 'Wake Up Call Coffee', 'Post Falls', 'ID'),
       (1, 'Starbucks', 'Seattle', 'WA'),
       (2, 'Jacobs Java', 'Spokane Valley', 'WA');
```

Build Schema | Edit Fullscreen | Browser | [ . ]

```
SELECT cs.shop_name AS 'Coffee Shop',
       c.coffee_name AS 'Coffee Beans',
       s.company_name AS 'Bean Supplier'
FROM Coffee c
INNER JOIN Coffee_Shop cs ON cs.shop_id = c.shop_id
INNER JOIN Supplier s ON s.supplier_id = c.supplier_id
ORDER BY cs.shop_name, c.coffee_name, s.company_name;
```

Run SQL ▶ | Edit Fullscreen | [ . ]

| Coffee Shop | Coffee Beans | Bean Supplier |
|---|---|---|
| Jacobs Java | Beans of Liberica | Alligator Coffee Co |
| Starbucks | Coffee Robusta | Roasters INC |
| Wake Up Call Coffee | Arabica Boralis | Coffee-R-Us |

Record Count: 3, Execution Time: 20ms + View Execution Plan link