

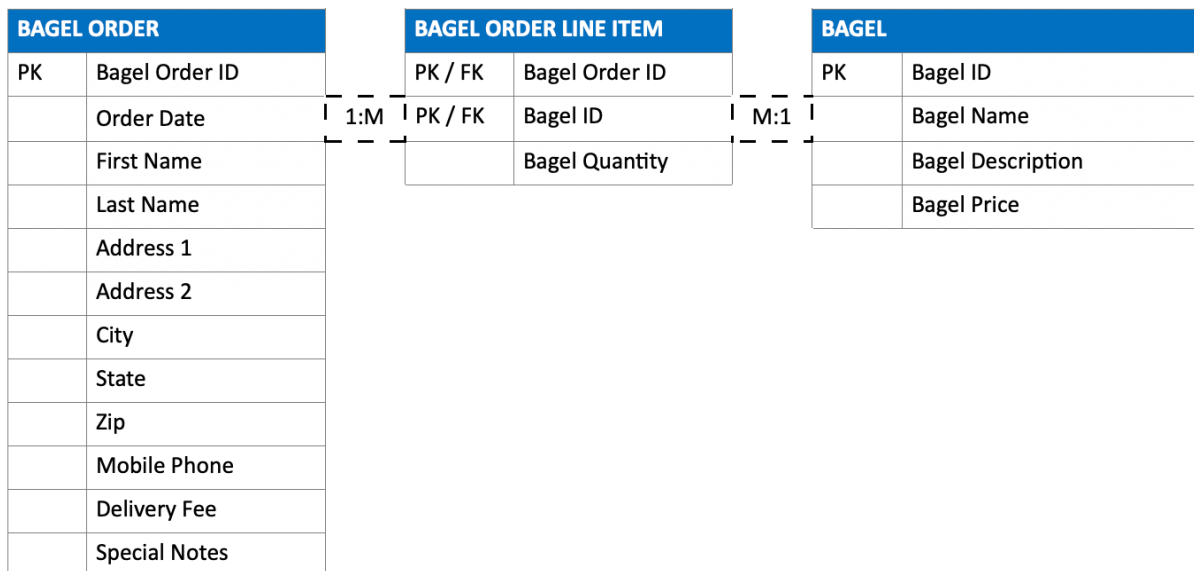
# Task 1: Normalization & Database Design

Xavier Loera Flores

## Part A: Nora's Bagel Bin

### A1: Second Normal Form

#### Second Normal Form (2NF)



#### Attribute Explanation

The Bagel Order table features the Bagel Order Id, Order Date, First Name, Last Name, Address 1, Address 2, City, State, Zip, Mobile Phone, Delivery Fee, and Special Notes attributes since these attributes can all exclusively depend on the Bagel Order ID.

The Bagel Order Line Item table features the Bagel Order ID, Bagel ID, and Bagel Quantity since the Bagel Quantity is dependent on the unique compound attribute consisting of the Bagel Order ID and the Bagel ID.

The Bagel table features the Bagel ID, Bagel Name, Bagel Description, and Bagel Price attributes since these attributes are dependent on and can be referenced by just the Bagel ID without needing a Bagel Order ID.

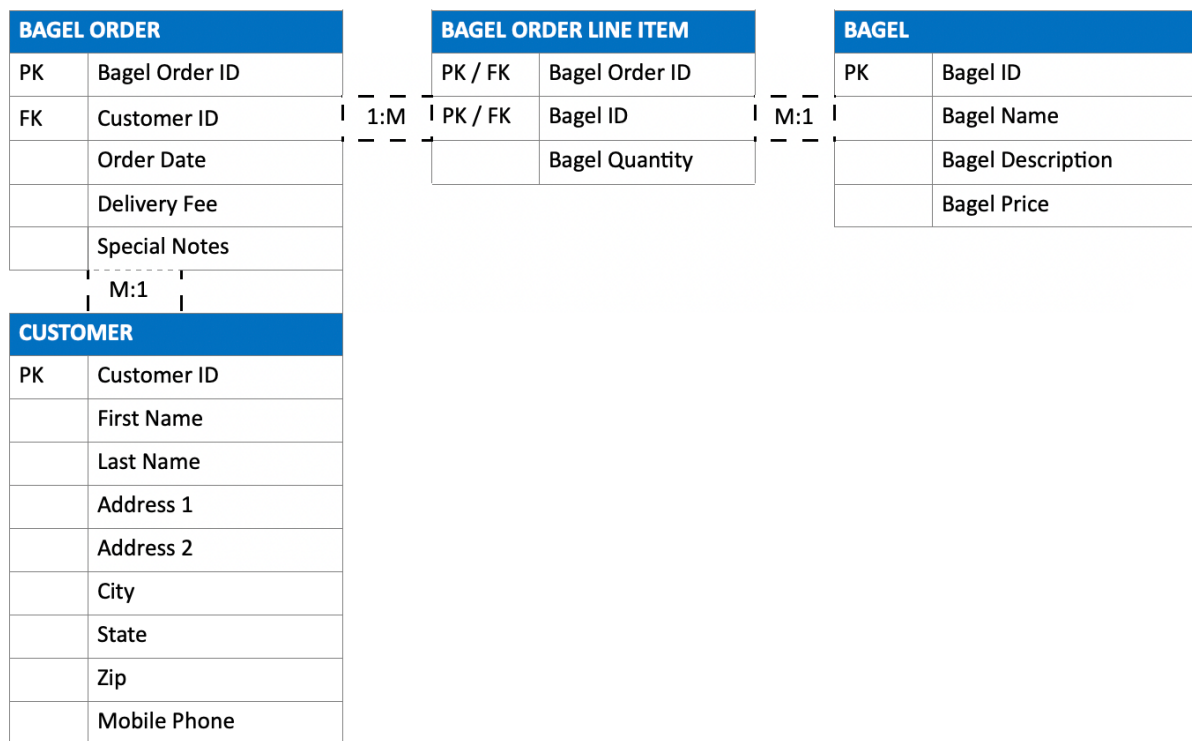
## Cardinality of Relationships

Every single Bagel Order can have many Bagel Order Line Items but each Bagel Order Line Item can only be assigned to one Bagel Order. The Bagel Order Table has a one-to-many relationship with the Bagel Order Line Item table.

Every single Bagel Order Line Item must have one Bagel but any single Bagel can be in many Bagel Order Line Items. The Bagel order Line Item table has a many-to-one relationship with the Bagel table.

## A2: Third Normal Form

### Third Normal Form (3NF)



## Attribute Explanation

The Bagel Order table features the Bagel Order Id, Customer ID, Order Date, Delivery Fee, and Special Notes attributes since these attributes are all dependent on the Bagel Order ID.

The Customer table features the Customer ID, First Name, Last Name, Address 1, Address 2, City, State, Zip, and Mobile Phone attributes since these attributes are all dependent on and can be referenced by just the Customer ID without needing a Bagel Order ID.

The Bagel Order Line Item table features the Bagel Order ID, Bagel ID, and Bagel Quantity since the Bagel Quantity is dependent on the unique compound attribute consisting of the Bagel Order ID and the Bagel ID.

The Bagel table features the Bagel ID, Bagel Name, Bagel Description, and Bagel Price attributes since these attributes are dependent on and can be referenced using just the Bagel ID.

## Cardinality of Relationships

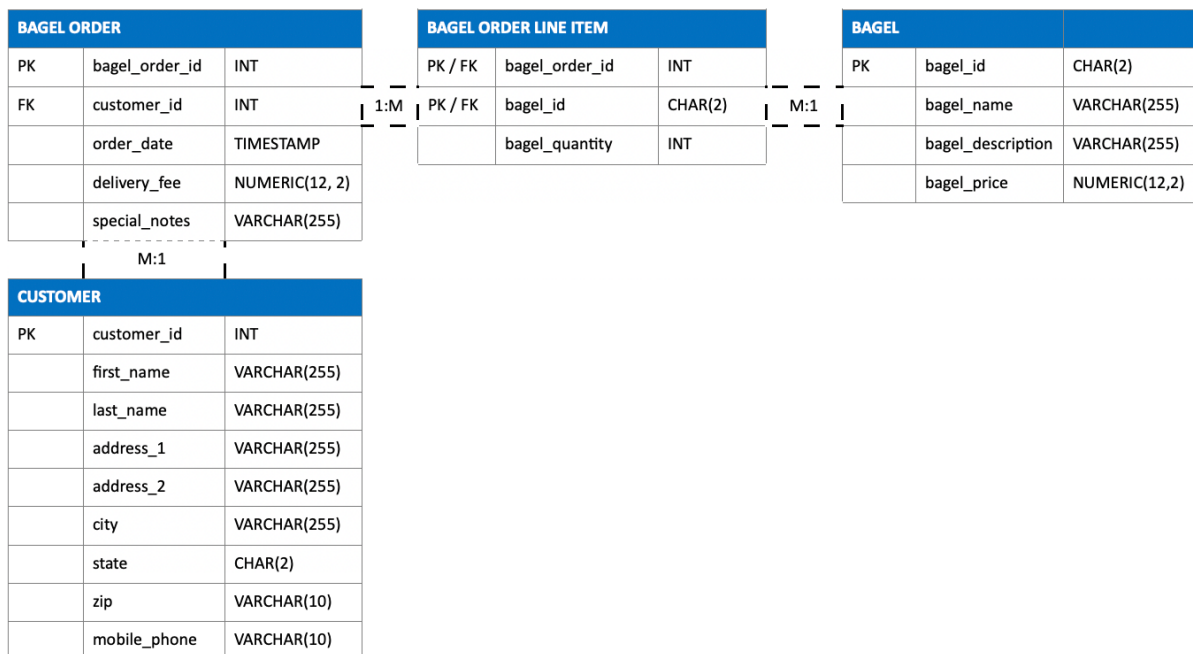
Every single Bagel Order can have many Bagel Order Line Items but each Bagel Order Line Item can only be assigned to one Bagel Order. The Bagel Order Table has a one-to-many relationship with the Bagel Order Line Item table.

Every single Bagel Order Line Item must have one Bagel but any single Bagel can be in many Bagel Order Line Items. The Bagel order Line Item table has a many-to-one relationship with the Bagel table.

Every single Bagel Order must have one Customer but any single Customer can have many Bagel Orders. The Bagel Order Table has a many-to-one relationship with the Customer table.

## A3: Final Physical Database Model

### Final Physical Database Model



# Part B:Jaunty Coffee Co

B1:

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
1 CREATE TABLE Coffee_Shop (
2   shop_id          INT PRIMARY KEY,
3   shop_name        VARCHAR(50),
4   city             VARCHAR(50),
5   state            CHAR(2)
6 );
7
8 CREATE TABLE Employee (
9   employee_id      INT PRIMARY KEY,
10  first_name        VARCHAR(30),
11  last_name         VARCHAR(30),
12  hire_date         DATE,
13  job_title         VARCHAR(30),
14  shop_id           INT,
15  FOREIGN KEY (shop_id) REFERENCES Coffee_Shop(shop_id)
16 );
17
18 CREATE TABLE Supplier (
19   supplier_id      INT PRIMARY KEY,
20   company_name     VARCHAR(50),
21   country          VARCHAR(30),
22   sales_contact_name VARCHAR(60),
23   email            VARCHAR(50) NOT NULL
24 );
25
26 CREATE TABLE Coffee (
27   coffee_id        INT PRIMARY KEY,
```

Build Schema Edit Fullscreen Browser [;] Run SQL Edit Fullscreen [;]

Schema Ready

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

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

CREATE TABLE Supplier (
    supplier_id      INT PRIMARY KEY,
    company_name     VARCHAR(50),
    country          VARCHAR(30),
```

```

    sales_contact_name    VARCHAR(60),
    email                 VARCHAR(50) NOT NULL
);

CREATE TABLE Coffee (
    coffee_id             INT PRIMARY KEY,
    shop_id               INT,
    supplier_id           INT,
    coffee_name           VARCHAR(30),
    price_per_pound       NUMERIC(5,2),
    FOREIGN KEY (shop_id) REFERENCES Coffee_Shop(shop_id),
    FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id)
);

```

B2:

SQL Fiddle MySQL 5.6 - View Sample Fiddle Clear Text to DDL Donate About

```

33 FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id)
34 );
35
36
37 INSERT INTO Coffee_Shop
38 VALUES
39 (1, 'Starbucks Coffee', 'San Jose', 'CA'),
40 (2, 'Pete Coffee', 'Los Angeles', 'CA'),
41 (3, 'Phil Coffee', 'Brooklyn', 'NY');
42
43 INSERT INTO Employee
44 VALUES
45 (1, 'John', 'Wick', '2023-09-10', 'Barista', 1),
46 (2, 'Indiana', 'Jones', '2023-10-09', 'Manager', 2),
47 (3, 'Sherlock', 'Holmes', '2023-07-12', 'Driver', 3);
48
49 INSERT INTO Supplier
50 VALUES
51 (1, 'CoffeeBeans', 'United States', 'Michael West', 'michael@coffeebeans.
52 (2, 'ValueBeans', 'United States', 'John Simmons', 'john@valuebeans.com')
53 (3, 'LuxBeans', 'United States', 'Mary Strong', 'mary@luxbeans.com');
54
55 INSERT INTO Coffee
56 VALUES
57 (1, 1, 1, 'Mocha', 7.99),
58 (2, 2, 2, 'Frap', 5.99),
59 (3, 3, 3, 'Black', 1.50);

```

```

1 SELECT * FROM Coffee_shop;
2 SELECT * FROM Employee;
3 SELECT * FROM Supplier;
4 SELECT * FROM Coffee;

```

Build Schema Edit Fullscreen Browser [;] Run SQL Edit Fullscreen [;]

shop_id	shop_name	city	state
1	Starbucks Coffee	San Jose	CA
2	Pete Coffee	Los Angeles	CA
3	Phil Coffee	Brooklyn	NY

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

employee_id	first_name	last_name	hire_date	job_title	shop_id
1	John	Wick	2023-09-10	Barista	1
2	Indiana	Jones	2023-10-09	Manager	2
3	Sherlock	Holmes	2023-07-12	Driver	3

59

(3, 3, 3, 'Black', 1.50);

Build Schema

Edit Fullscreen

Browser

[:]

Run SQL

Edit Fullscreen

[:]

shop_id	shop_name	city	state
1	Starbucks Coffee	San Jose	CA
2	Pete Coffee	Los Angeles	CA
3	Phil Coffee	Brooklyn	NY

Record Count: 3; Execution Time: 4ms

View Execution Plan

link

employee_id	first_name	last_name	hire_date	job_title	shop_id
1	John	Wick	2023-09-10	Barista	1
2	Indiana	Jones	2023-10-09	Manager	2
3	Sherlock	Holmes	2023-07-12	Driver	3

Record Count: 3; Execution Time: 1ms

View Execution Plan

link

supplier_id	company_name	country	sales_contact_name	email
1	CoffeeBeans	United States	Michael West	michael@coffeebeans.com
2	ValueBeans	United States	John Simmons	john@valuebeans.com
3	LuxBeans	United States	Mary Strong	mary@luxbeans.com

Record Count: 3; Execution Time: 1ms

View Execution Plan

link

coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
1	1	1	Mocha	7.99
2	2	2	Frap	5.99
3	3	3	Black	1.5

Record Count: 3; Execution Time: 1ms

View Execution Plan

link

```

INSERT INTO Coffee_Shop
VALUES
    (1, 'Starbucks Coffee', 'San Jose', 'CA'),
    (2, 'Pete Coffee', 'Los Angeles', 'CA'),
    (3, 'Phil Coffee', 'Brooklyn', 'NY');

INSERT INTO Employee
VALUES
    (1, 'John', 'Wick', '2023-09-10', 'Barista', 1),
    (2, 'Indiana', 'Jones', '2023-10-09', 'Manager', 2),
    (3, 'Sherlock', 'Holmes', '2023-07-12', 'Driver', 3);

INSERT INTO Supplier
VALUES
    (1, 'CoffeeBeans', 'United States', 'Michael West', 'michael@coffeebeans.com'),
    (2, 'ValueBeans', 'United States', 'John Simmons', 'john@valuebeans.com'),
    (3, 'LuxBeans', 'United States', 'Mary Strong', 'mary@luxbeans.com');

INSERT INTO Coffee
VALUES

```

```
(1, 1, 1, 'Mocha', 7.99),
(2, 2, 2, 'Frap', 5.99),
(3, 3, 3, 'Black', 1.50);
```

B3:

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
43 INSERT INTO Employee
44 VALUES
45 (1, 'John', 'Wick', '2023-09-10', 'Barista', 1),
46 (2, 'Indiana', 'Jones', '2023-10-09', 'Manager', 2),
47 (3, 'Sherlock', 'Holmes', '2023-07-12', 'Driver', 3);
48
49 INSERT INTO Supplier
50 VALUES
51 (1, 'CoffeeBeans', 'United States', 'Michael West', 'michael@coffeebeans.
52 (2, 'ValueBeans', 'United States', 'John Simmons', 'john@valuebeans.com')
53 (3, 'LuxBeans', 'United States', 'Mary Strong', 'mary@luxbeans.com');
54
55 INSERT INTO Coffee
56 VALUES
57 (1, 1, 1, 'Mocha', 7.99),
58 (2, 2, 2, 'Frap', 5.99),
59 (3, 3, 3, 'Black', 1.50);
60
61 CREATE VIEW employee_info AS SELECT
62 employee_id,
63 CONCAT(first_name, ' ', last_name) AS employee_full_name,
64 hire_date,
65 job_title,
66 shop_id
67 FROM Employee;
68
69
```

```
1 SELECT * FROM employee_info
```

Build Schema Edit Fullscreen Browser [;]

Run SQL Edit Fullscreen [;]

employee_id	employee_full_name	hire_date	job_title	shop_id
1	John Wick	2023-09-10	Barista	1
2	Indiana Jones	2023-10-09	Manager	2
3	Sherlock Holmes	2023-07-12	Driver	3

Record Count: 3; Execution Time: 5ms View Execution Plan link

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

B4:

SQL FiddleMySQL 5.6View Sample FiddleClearText to DDLDonateAbout

47 (3, 'Sherlock', 'Holmes', '2023-07-12', 'Driver', 3);  
48  
49 INSERT INTO Supplier  
50 VALUES  
51 (1, 'CoffeeBeans', 'United States', 'Michael West', 'michael@coffeebeans.  
52 (2, 'ValueBeans', 'United States', 'John Simmons', 'john@valuebeans.com')  
53 (3, 'LuxBeans', 'United States', 'Mary Strong', 'mary@luxbeans.com');  
54  
55 INSERT INTO Coffee  
56 VALUES  
57 (1, 1, 1, 'Mocha', 7.99),  
58 (2, 2, 2, 'Frap', 5.99),  
59 (3, 3, 3, 'Black', 1.50);  
60  
61  
62 CREATE VIEW employee\_info AS SELECT  
63 employee\_id,  
64 CONCAT(first\_name, ' ', last\_name) AS employee\_full\_name,  
65 hire\_date,  
66 job\_title,  
67 shop\_id  
68 FROM Employee;  
69  
70  
71 CREATE INDEX idx\_coffee ON Coffee(coffee\_name);  
72  
73

1 SELECT \*FROM Coffee USE INDEX(idx\_coffee)

Build SchemaEdit FullscreenBrowser[;]Run SQLEdit Fullscreen[;]

coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
1	1	1	Mocha	7.99
2	2	2	Frap	5.99
3	3	3	Black	1.5

Record Count: 3; Execution Time: 9msView Execution Planlink

CREATE INDEX idx\_coffee ON Coffee(coffee\_name);



B5:

SQL FiddleMySQL 5.6View Sample FiddleClearText to DDLDonateAbout

```
47 (3, 'Sherlock', 'Holmes', '2023-07-12', 'Driver', 3);
48
49 INSERT INTO Supplier
50 VALUES
51 (1, 'CoffeeBeans', 'United States', 'Michael West', 'michael@coffeebeans.
52 (2, 'ValueBeans', 'United States', 'John Simmons', 'john@valuebeans.com')
53 (3, 'LuxBeans', 'United States', 'Mary Strong', 'mary@luxbeans.com');
54
55 INSERT INTO Coffee
56 VALUES
57 (1, 1, 1, 'Mocha', 7.99),
58 (2, 2, 2, 'Frap', 5.99),
59 (3, 3, 3, 'Black', 1.50);
60
61
62 CREATE VIEW employee_info AS SELECT
63 employee_id,
64 CONCAT(first_name, ' ', last_name) AS employee_full_name,
65 hire_date,
66 job_title,
67 shop_id
68 FROM Employee;
69
70
71 CREATE INDEX idx_coffee ON Coffee(coffee_name);
72
73
```

```
1 SELECT * FROM Employee where employee_id = 1
```

Build SchemaEdit FullscreenBrowser[:]

Run SQLEdit Fullscreen[:]

employee_id	first_name	last_name	hire_date	job_title	shop_id
1	John	Wick	2023-09-10	Barista	1

Record Count: 1; Execution Time: 15msView Execution Planlink

Did this query solve the problem? If so, consider donating \$5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!

```
SELECT * FROM Employee where employee_id = 1
```

B6:

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
11 last_name VARCHAR(30),
12 hire_date DATE,
13 job_title VARCHAR(30),
14 shop_id INT,
15 FOREIGN KEY (shop_id) REFERENCES Coffee_Shop(shop_id)
16 );
17
18 CREATE TABLE Supplier (
19     supplier_id INT PRIMARY KEY,
20     company_name VARCHAR(50),
21     country VARCHAR(30),
22     sales_contact_name VARCHAR(60),
23     email VARCHAR(50) NOT NULL
24 );
25
26 CREATE TABLE Coffee (
27     coffee_id INT PRIMARY KEY,
28     shop_id INT,
29     supplier_id INT,
30     coffee_name VARCHAR(30),
31     price_per_pound NUMERIC(5,2),
32     FOREIGN KEY (shop_id) REFERENCES Coffee_Shop(shop_id),
33     FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id)
34 );
35
36
37 INSERT INTO Coffee_Shop
38 VALUES
```

```
1 SELECT
2 Supplier.sales_contact_name,
3 Supplier.email,
4 Coffee_Shop.shop_name,
5 Coffee_Shop.city,
6 Coffee_Shop.state,
7 Coffee.coffee_name,
8 Coffee.price_per_pound
9 FROM Coffee
10 INNER JOIN Coffee_Shop ON Coffee_Shop.shop_id = Coffee.shop_id
11 INNER JOIN Supplier ON Coffee.supplier_id = Supplier.supplier_id;
```

Build Schema Edit Fullscreen Browser [;] Run SQL Edit Fullscreen [;]

sales_contact_name	email	shop_name	city	state	coffee_name	price_per_pound
Michael West	michael@coffeebeans.com	Starbucks Coffee	San Jose	CA	Mocha	7.99
John Simmons	john@valuebeans.com	Pete Coffee	Los Angeles	CA	Frap	5.99
Mary Strong	mary@luxbeans.com	Phil Coffee	Brooklyn	NY	Black	1.5

Record Count: 3; Execution Time: 18ms View Execution Plan link

```
SELECT
Supplier.sales_contact_name,
Supplier.email,
Coffee_Shop.shop_name,
Coffee_Shop.city,
Coffee_Shop.state,
Coffee.coffee_name,
Coffee.price_per_pound
FROM Coffee
INNER JOIN Coffee_Shop ON Coffee_Shop.shop_id = Coffee.shop_id
INNER JOIN Supplier ON Coffee.supplier_id = Supplier.supplier_id;
```