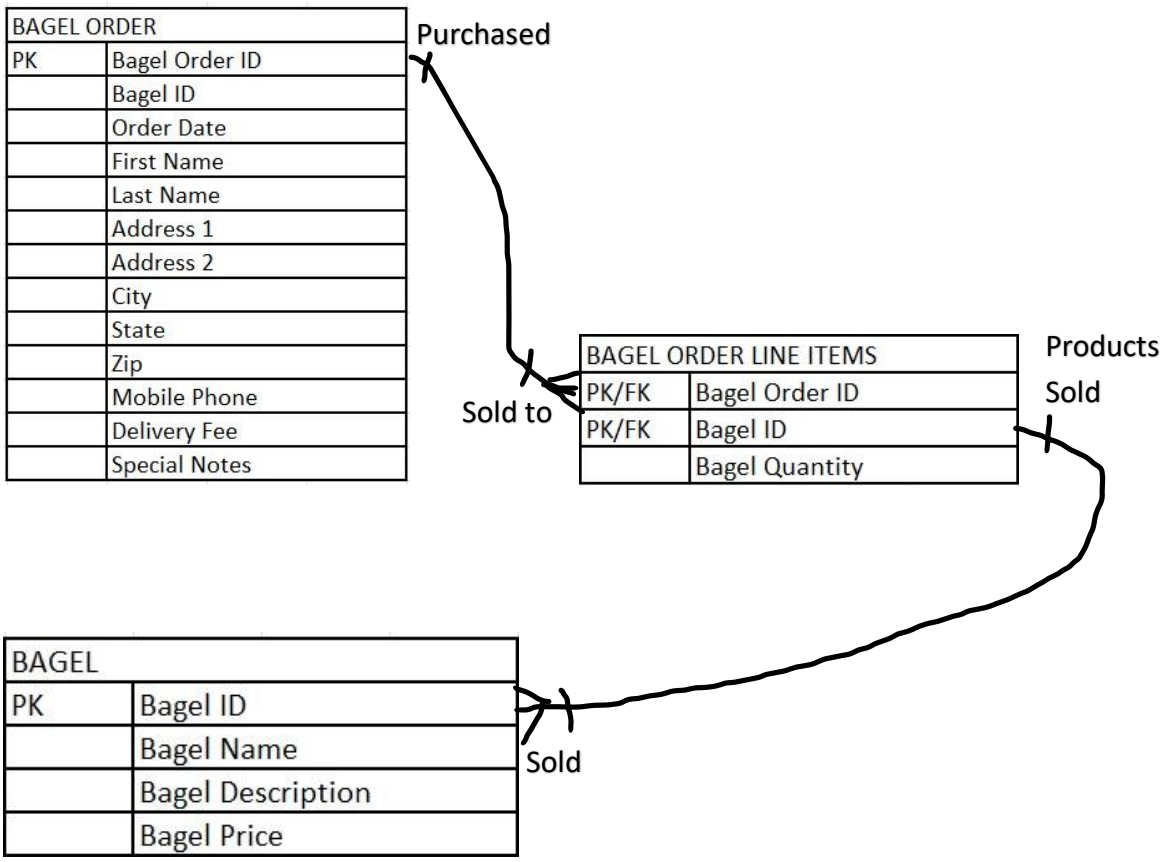
Brandon Davis

C170: Data Management Applications

A1a-A1b.)



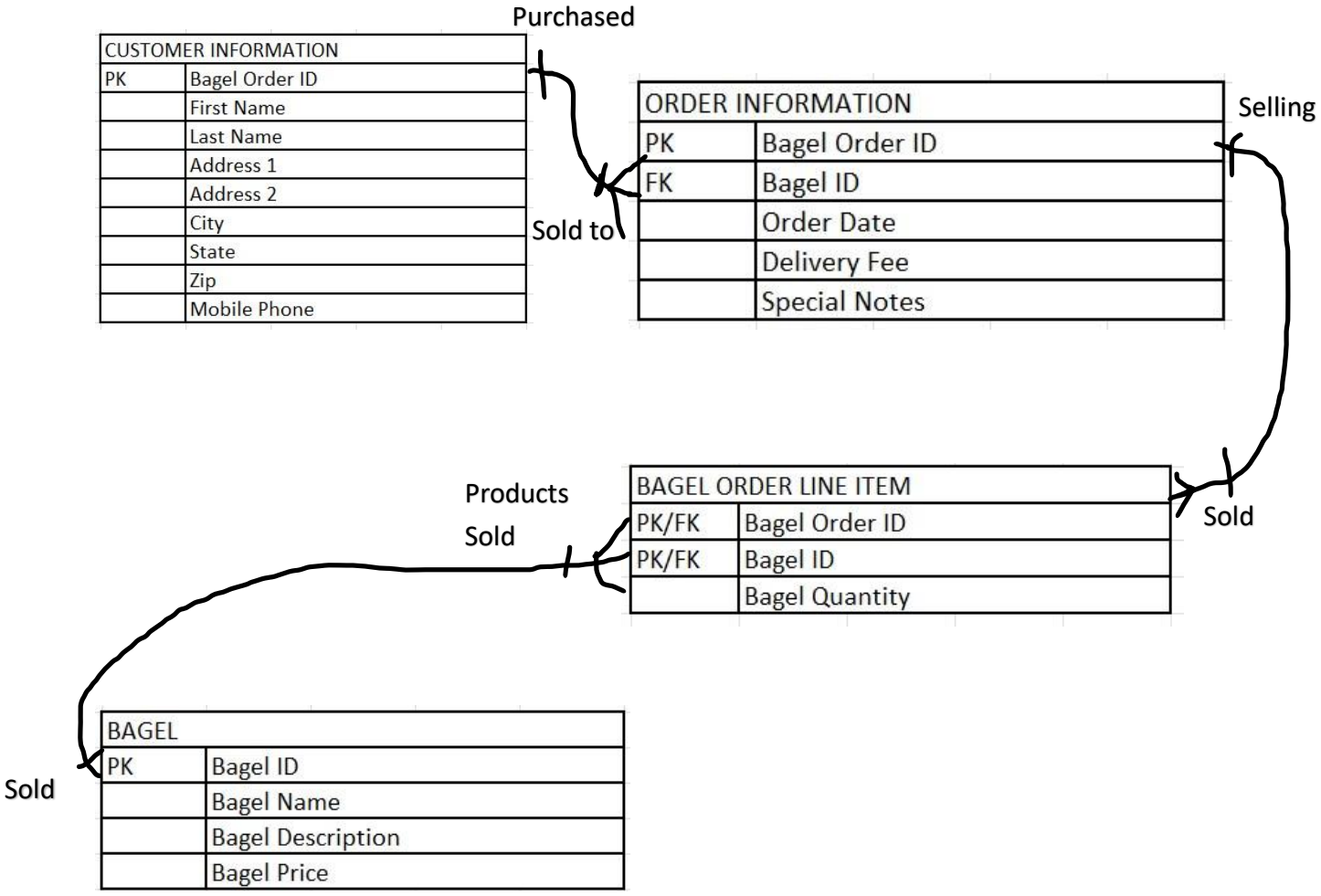
A1c.)

The diagram for Second Normal Form was created after the following considerations. Firstly, reviewing the First Normal Form diagram provided and understanding the data. Lastly, the establishment of simple Primary Keys and Foreign Keys to ensure data uniqueness to maintaining data integrity. The cardinality was determined by ensuring that data model was logical. Meaning that a customer can make multiple orders while having one order ID and that one order ID is given uniquely to all customers who have placed orders for bagels.

Brandon Davis

C170: Data Management Applications

A2a-A2d.)



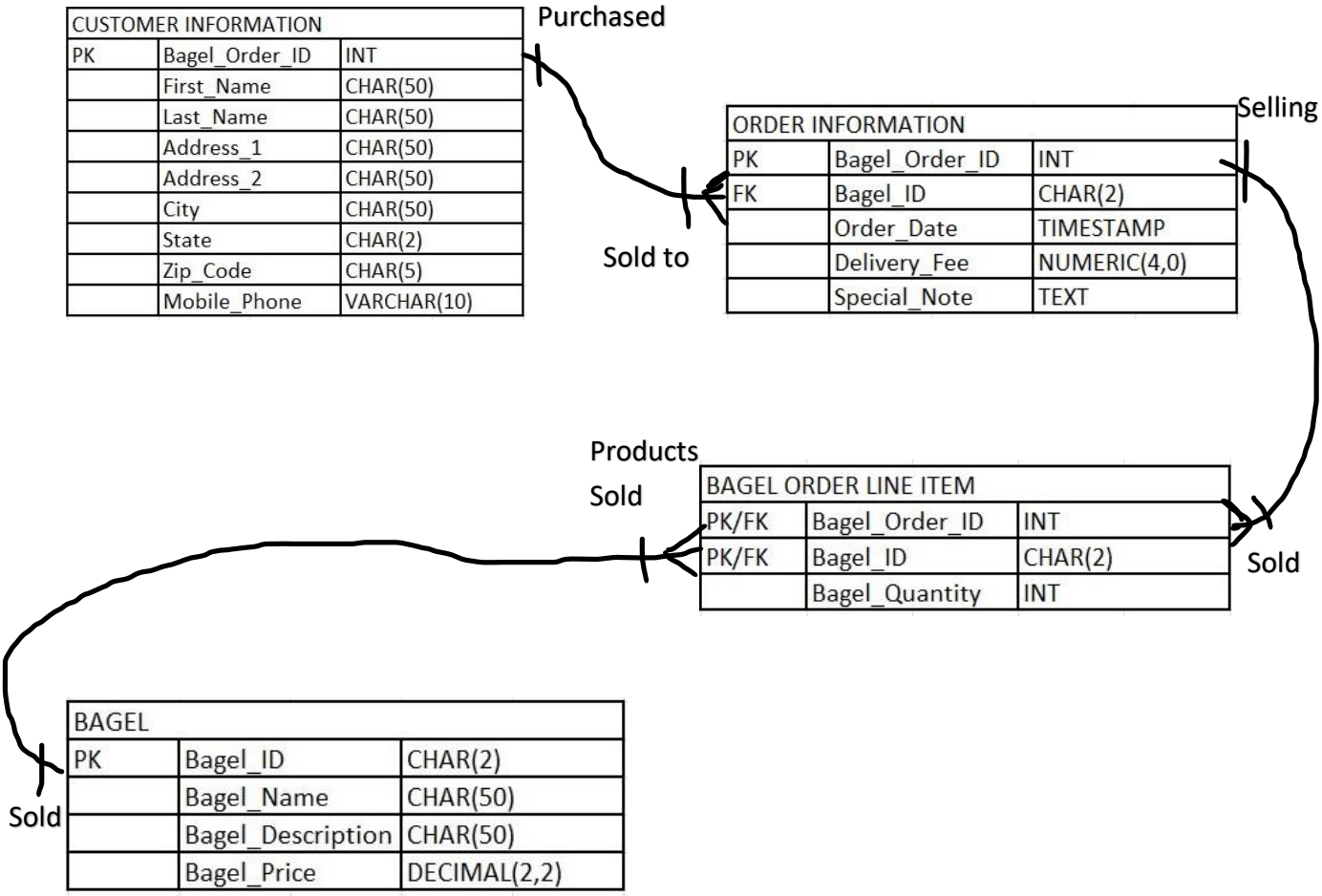
A2e.)

The diagram for Third Normal Form was created after the following considerations. Firstly, reviewing the Second Normal Form diagram provided and understanding the data. Lastly, removing any transitive dependencies to ensure data maintains referential integrity. By doing so I separated the Bagel Order table into Customer Information and Order Information. This allows for the removal of all transitive dependencies allowing this to move into Third Normal Form. The cardinality from the Second Normal Form diagram and the Third Normal Form diagram are the same. The only changes that have been made to move from Second Normal Form to Third Normal Form is the removal of transitive dependencies which would not affect the cardinality of the diagram.

Brandon Davis

C170: Data Management Applications

A3a-A3b.)



Brandon Davis

C170: Data Management Applications

B1a.) Raw Code to Create Tables (MySQL):

Employee Table:

```
CREATE TABLE Employee (
  EmployeeID int NOT NULL,
  FirstName varchar(30) DEFAULT NULL,
  LastName varchar(30) DEFAULT NULL,
  HireDate date DEFAULT NULL,
  JobTitle varchar(30) DEFAULT NULL,
  ShopID int DEFAULT NULL,
  PRIMARY KEY (EmployeeID),
  FOREIGN KEY (ShopID) REFERENCES CoffeeShop (ShopID)
);
```

CoffeeShop Table:

```
CREATE TABLE CoffeeShop (
  ShopID int NOT NULL,
  ShopName varchar(50) DEFAULT NULL,
  City varchar(50) DEFAULT NULL,
  State char(2) DEFAULT NULL,
  PRIMARY KEY (ShopID)
);
```

Coffee Table:

```
CREATE TABLE Coffee (
  CoffeeID int NOT NULL,
  ShopID int DEFAULT NULL,
  SupplierID int DEFAULT NULL,
  CoffeeName varchar(30) DEFAULT NULL,
  PricePerPound decimal(5,2) DEFAULT NULL,
```

Brandon Davis

C170: Data Management Applications

```
  PRIMARY KEY (CoffeeID),
  FOREIGN KEY (ShopID) REFERENCES coffeeshop (ShopID),
  FOREIGN KEY (SupplierID) REFERENCES supplier (SupplierID)
);
```

Supplier Table:

```
CREATE TABLE Supplier (
  SupplierID int NOT NULL,
  CompanyName varchar(50) DEFAULT NULL,
  Country varchar(30) DEFAULT NULL,
  SalesContactName varchar(60) DEFAULT NULL,
  Email varchar(50) NOT NULL,
  PRIMARY KEY (SupplierID)
);
```

B1b.) Screenshots from database

Employee Table:

```
1  CREATE TABLE Employee
2  (EmployeeID INT,
3  FirstName VARCHAR(30),
4  LastName VARCHAR(30),
5  HireDate DATE,
6  JobTitle VARCHAR(30),
7  ShopID INT,
8  PRIMARY KEY (EmployeeID),
9  FOREIGN KEY (ShopID) REFERENCES CoffeeShop(ShopID)
10 );
```

Brandon Davis

C170: Data Management Applications

Coffee Shop Table:

```
CREATE TABLE CoffeeShop(
ShopID INT,
ShopName VARCHAR(50),
City VARCHAR(50),
State CHAR(2),
PRIMARY KEY (ShopID)
);
```

```
1  17:12:19  CREATE TABLE CoffeeShop( ShopID INT, ShopName VARCHAR(50), City VARCHAR(50), State C...  0 row(s) affected  0.016 sec
```

Coffee Table:

```
CREATE TABLE Coffee(
CoffeeID INT,
ShopID INT,
SupplierID INT,
CoffeeName VARCHAR(30),
PricePerPound NUMERIC(5,2),
PRIMARY KEY (CoffeeID),
FOREIGN KEY (ShopID) REFERENCES CoffeeShop(ShopID),
FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID)
);
```

```
1  17:35:50  CREATE TABLE Coffee( CoffeeID INT, ...  0 row(s) affected  0.015 sec
```

Supplier Table:

```
CREATE TABLE Supplier(
SupplierID INT,
CompanyName VARCHAR(50),
Country VARCHAR(30),
SalesContactName VARCHAR(60),
Email VARCHAR(50) NOT NULL,
PRIMARY KEY (SupplierID)
);
```

```
1  17:31:52  CREATE TABLE Supplier( SupplierID IN...  0 row(s) affected  0.016 sec
```

Brandon Davis

C170: Data Management Applications

B2a.) Row Insertion Code (MySQL):

Employee Table:

INSERT INTO Employee (EmployeeID, FirstName, LastName, HireDate, JobTitle, ShopID) VALUES (101, 'Luke', 'Diamond', '2022-08-21', 'Store Owner', 007), (102, 'Jessica', 'Yates', '2022-8-21', 'Store Manager', 007), (103, 'Chris', 'Ramsey', '2022-8-25', 'Assistant Manager', 007);

Coffee Shop Table:

INSERT INTO CoffeeShop (ShopID, ShopName, City, State) VALUES (007, 'Anamaniac Coffee Co.', 'Greensboro', 'NC'), (666, 'Sin City Coffee Co.', 'Las Vegas', 'NV'), (101, 'Back 2 Basics Coffee Co.', 'New York City', 'NY');

Coffee Table:

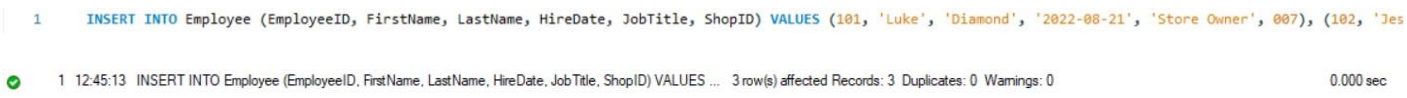
INSERT INTO Coffee (CoffeeID, ShopID, SupplierID, CoffeeName, PricePerPound) VALUES (200, '007', '002', 'Blackbeards Blend', '15.00'), (201, '666', '003', 'Revival Blend', '16.00'), (202, '101', '001', 'The Don Blend', '15.50');

Supplier Table:

INSERT INTO Supplier (SupplierID, CompanyName, Country, SalesContactName, Email) VALUES (001, 'Frank Fresh Beans', 'Italy', 'Gio Delmonico', 'Gio@freshbeans.com'), (002, 'Screamin Eagle Beans', 'USA', 'Ricky Bobby', 'Ricky@freedom.us'), (003, 'Free Range Beans', 'Columbia', 'Jesus Montenegro', 'JesusM@freerange.com');

B2b.) Screenshots from Database

Employee Table:



Brandon Davis

C170: Data Management Applications

Coffee Shop Table:

```
1 • INSERT INTO CoffeeShop (ShopID, ShopName, City, State) VALUES (007, 'Anamaniac Coffee Co.', 'Greensboro', 'NC'), (666, 'Sin City Coffee Co.', 'Las Vegas', 'NV'),
```

```
1  12:12:45  INSERT INTO CoffeeShop (ShopID, ShopName, City, State) VALUES (007, 'Anamaniac Coffee Co.',...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
```

Coffee Table:

```
1 • INSERT INTO Coffee (CoffeeID, ShopID, SupplierID, CoffeeName, PricePerPound) VALUES (200, '007', '002', 'Blackbeards Blend', '15.00'), (201, '666', '003', 'Revi\
```

```
1  12:33:08  INSERT INTO Coffee (CoffeeID, ShopID, SupplierID, CoffeeName, PricePerPound) VALUES (200, '...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
```

Supplier Table:

```
1 • INSERT INTO Supplier (SupplierID, CompanyName, Country, SalesContactName, Email) VALUES (001, 'Frank Fresh Beans', 'Italy', 'Gio Delmonico', 'Gio@freshbeans.com'
```

```
1  12:22:28  INSERT INTO Supplier (SupplierID, CompanyName, Country, SalesContactName, Email) VALUES (0...  3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0  0.000 sec
```

B3a.) Create View Code for Employee (MySQL):

Create View Employee_table as

Select EmployeeID, Concat(FirstName, ' ', LastName) AS 'EmployeeFullName', HireDate, JobTitle, ShopID

From Employee

Where EmployeeID > 100;

B3b.) Screenshots from Database

```
1   Create View Employee_table as
2   Select EmployeeID, Concat(FirstName, ' ', LastName) AS 'EmployeeFullName', HireDate, JobTitle, ShopID
3   From Employee
4   Where EmployeeID > 100;
```

#	Time	Action	Message	Duration / Fetch
1	13:05:07	Create View Employee_table as Select EmployeeID, Concat(FirstName, ' ', LastName) AS 'Employe...	0 row(s) affected	0.000 sec

Brandon Davis

C170: Data Management Applications

B4a.) Create Index Code for Coffee_name (MySQL):

Create Index CIndex ON Coffee(CoffeeName);

B4b.) Screenshots from Database

```
1    Create Index CIndex ON Coffee(CoffeeName);
```

```
1  13:10:04  Create Index CIndex ON Coffee(CoffeeName)    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0    0.031 sec
```

B5a.) SFW (Select-from-where) query code (MySQL):

Select * From employee

Where EmployeeID > 100;

B5b.) Screenshots from Database

```
1    Select * From employee
2    Where EmployeeID > 100;
```

EmployeeID	FirstName	LastName	HireDate	JobTitle	ShopID
101	Luke	Diamond	2022-08-21	Store Owner	7
102	Jessica	Yates	2022-08-21	Store Manager	7
103	Chris	Ramsey	2022-08-25	Assistant Manager	7
NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types

employee 3 | Apply | Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	13:12:47	Select * From employee Where EmployeeID > 100 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

B6a.) Join Tables code:

SELECT * FROM Coffee c

inner join CoffeeShop s ON s.ShopID = c.ShopID

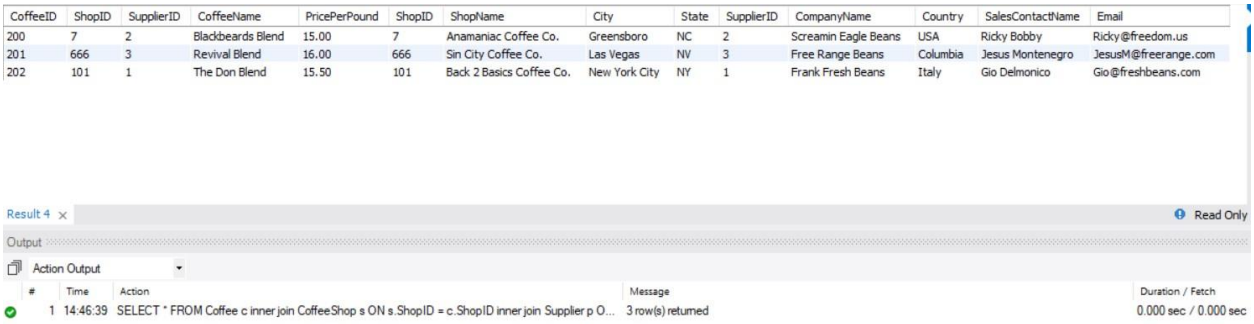
inner join Supplier p ON p.SupplierID = c.SupplierID

Brandon Davis

C170: Data Management Applications

B6b.) Screenshots from Database

```
1 •  SELECT * FROM Coffee c
2    inner join CoffeeShop s ON s.ShopID = c.ShopID
3    inner join Supplier p ON p.SupplierID = c.SupplierID
```



CoffeeID	ShopID	SupplierID	CoffeeName	PricePerPound	ShopID	ShopName	City	State	SupplierID	CompanyName	Country	SalesContactName	Email
200	7	2	Blackbeards Blend	15.00	7	Anamaniac Coffee Co.	Greensboro	NC	2	Screamin Eagle Beans	USA	Ricky Bobby	Ricky@freedom.us
201	666	3	Revival Blend	16.00	666	Sin City Coffee Co.	Las Vegas	NV	3	Free Range Beans	Columbia	Jesus Montenegro	JesusM@freerange.com
202	101	1	The Don Blend	15.50	101	Back 2 Basics Coffee Co.	New York City	NY	1	Frank Fresh Beans	Italy	Gio Delmonico	Gio@freshbeans.com

Result 4 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	14:46:39	SELECT * FROM Coffee c inner join CoffeeShop s ON s.ShopID = c.ShopID inner join Supplier p O...	3 row(s) returned	0.000 sec / 0.000 sec