Precious Naff

C170 Database Management Applications

TASK 1

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

Part A: Construct a normalized physical database model to represent the ordering process for Nora's Bagel Bin

Part A.1 Complete the second normal form (2NF) section

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

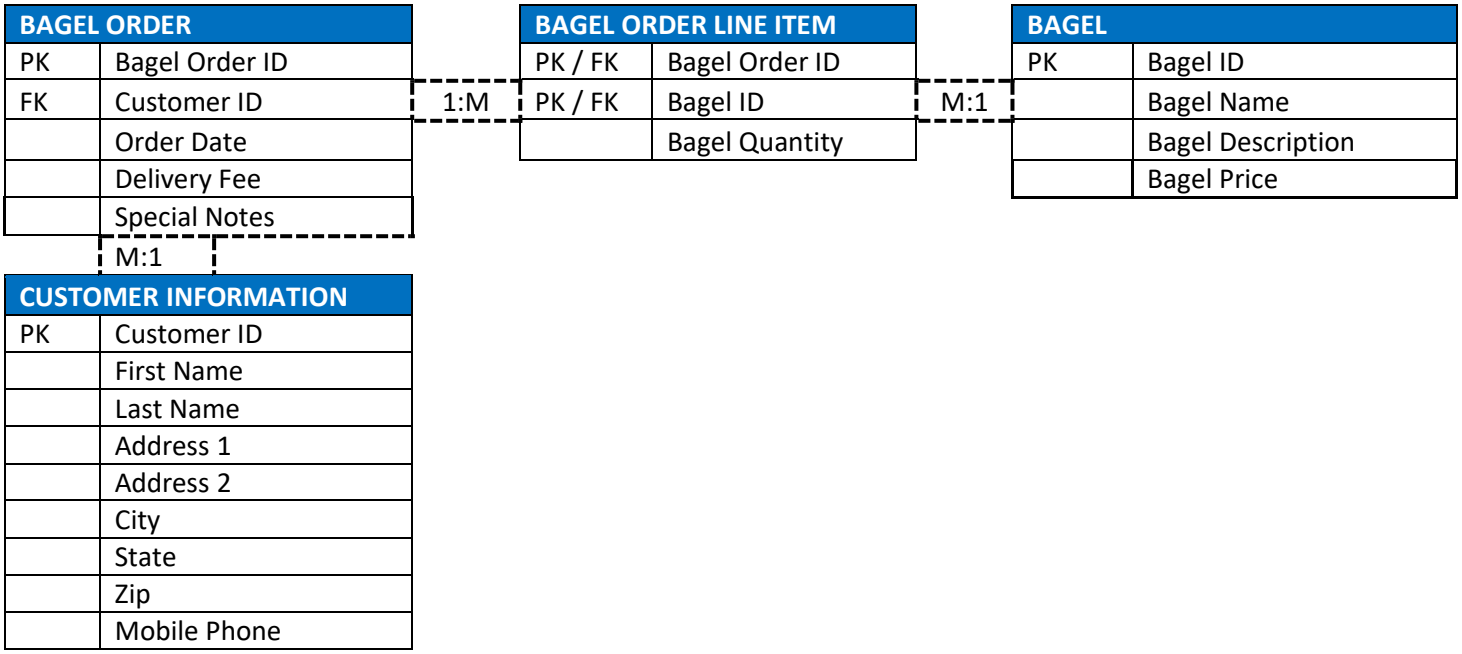
M:1

BAGEL	
PK	Bagel ID
	Bagel Name
	Bagel Description
	Bagel Price

I ordered the "Bagel Order" table by including information pertaining to the order and customer. I then put Bagel Quantity in the "Bagel order line item" table because it is functionally dependent on the bagel order ID and Bagel ID. The "Bagel" table then got all the attributes about the bagel. The cardinality of the relationship between the "Bagel Order" table and the "Bagel order line item" table is one to many because one order can have many bagel order line items, but the bagel order line item can belong to a maximum of one order. The cardinality of the relationship between the "Bagel order line item" is many to one because the Bagel order line item can only have one type of bagel on it, but each bagel type can be on many Bagel orders.

Part A.2 Complete the third normal form (3NF) section

Third Normal Form (3NF)



I broke down the information having to do with the customer from the “Bagel Order” table and put it in its own table (Customer Information) to get the data to the third normal form since it would be redundant if left in the “Bagel Order” table. Customer ID is now the foreign key in the “Bagel Order” table, which references the primary key in the “Customer Information” table. The cardinality of the relationship between the “Bagel Order” table and the “Customer Information” table is many to one because an order can have a maximum of one customer and a customer can have many orders.

Part A.3 Complete the "Final Physical Database Model" section

Final Physical Database Model

BAGEL ORDER		
PK	bagel_order_id	INT
FK	customer_id	INT
	order_date	TIMESTAMP
	delivery_fee	NUMERIC(2,2)
	special_notes	VARCHAR(120)

1:M

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

M:1

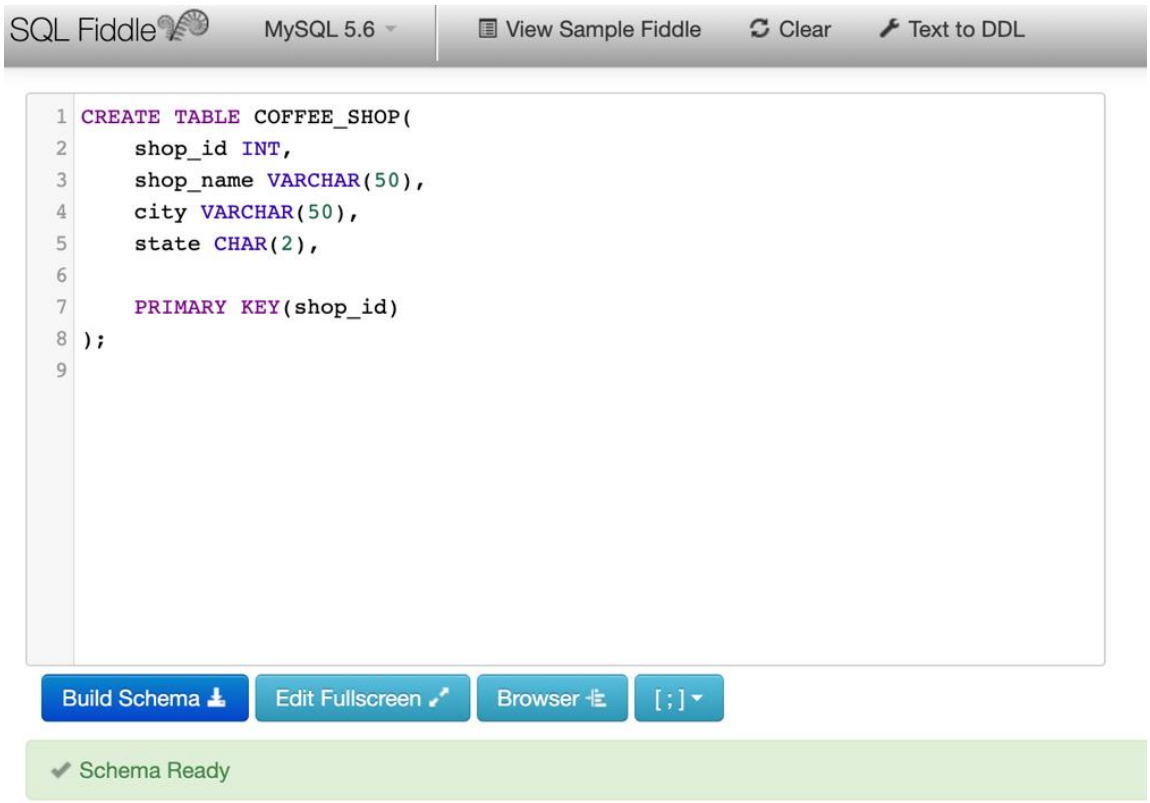
BAGEL		
PK	bagel_id	CHAR(2)
	bagel_name	VARCHAR(40)
	bagel_description	VARCHAR(120)
	bagel_price	NUMERIC(3,2)

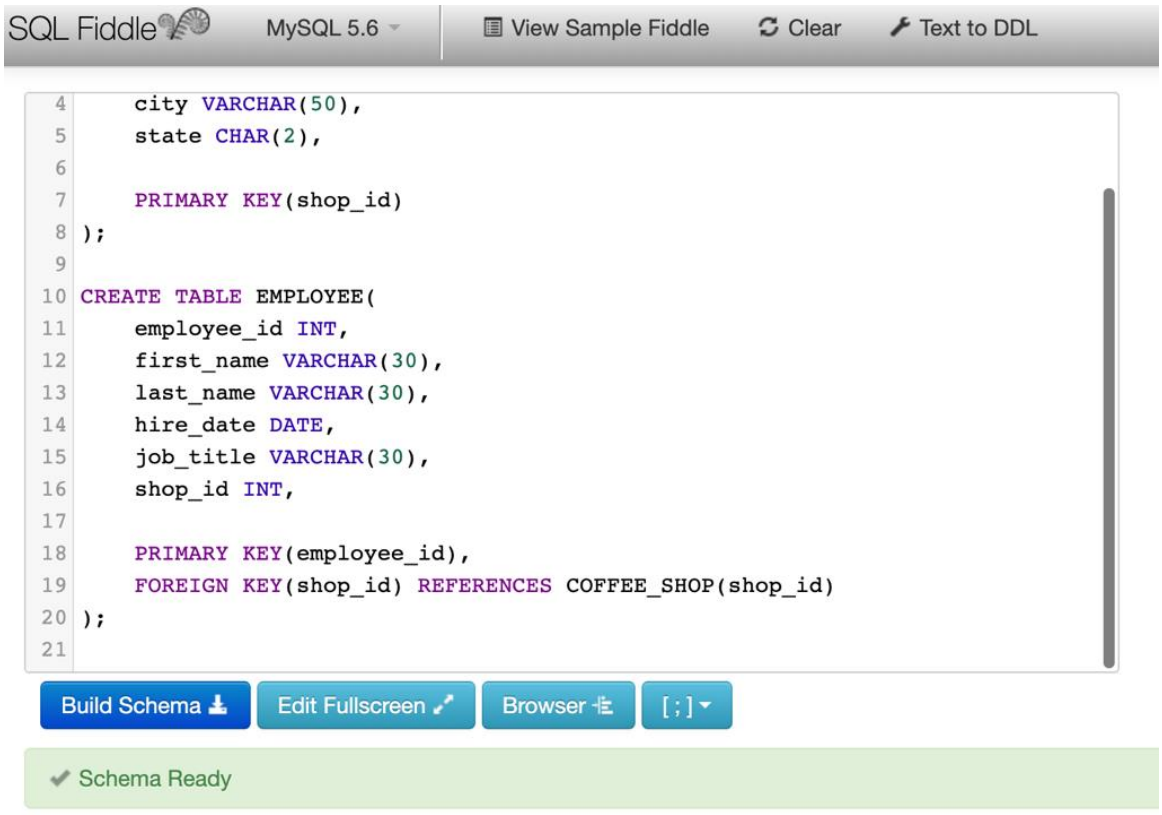
M:1

PK	customer_id	INT
	first_name	VARCHAR(40)
	last_name	VARCHAR(40)
	address_1	VARCHAR(100)
	address_2	VARCHAR(100)
	city	VARCHAR(40)
	state	CHAR(2)
	zip	VARCHAR(5)
	mobile_phone	VARCHAR(10)

Part B Create a database using the attached "Jaunty Coffee Co. ERD"

Part B.1 Develop SQL code to create *each* table





SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL

```
    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)
);

```

Build Schema Edit Fullscreen Browser [;]

Schema Ready

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL

```
    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 [;]

Schema Ready

Part B.2 Develop SQL code to populate *each* table in the database design document

SQL Fiddle | MySQL 5.6 | View Sample Fiddle | Clear | Text to DDL

```
);

INSERT INTO coffee_shop
  VALUES (1, 'Kaladi Brothers', 'Anchorage', 'AK'),
         (2, 'Starbucks', 'Chicago', 'IL'),
         (3, 'Super Beans', 'San Diego', 'CA');
INSERT INTO employee
  Values (197, 'Layla', 'Jones', '2016-03-18', 'Manager', 1),
         (238, 'Amber', 'Matthews', '2019-11-04', 'Barista',2),
         (270, 'Dave', 'Coleman', '2020-01-20', 'Barista',3);
INSERT INTO supplier
  VALUES (12, 'Quick Coffee', 'Columbia', 'Mark Smith', 'msmith@qcoffee.com')
         (17, 'Coffee 2 you', 'United States', 'Luca Miller', 'lucam@coffeeyo
         (21, 'Bean Farm', 'Brazil', 'Ivan Garcia', 'Ivang@beanfarm.com');
INSERT INTO coffee
  Values (1, 3, 12, 'Caffe Latte', 10.99),
         (2, 2, 21, 'Iced Coffe', 12.99),
         (3, 1, 17, 'Americano', 10.99);
```

Build Schema | Edit Fullscreen | Browser | [;]

Schema Ready

Part B.3 Develop SQL code to create a view

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

```
  VALUES (12, 'Quick Coffee', 'Columbia', 'Mark Smith', 'msmith@qcoffee.com')
         (17, 'Coffee 2 you', 'United States', 'Luca Miller', 'lucam@coffeeyo
         (21, 'Bean Farm', 'Brazil', 'Ivan Garcia', 'Ivang@beanfarm.com');
INSERT INTO coffee
  Values (1, 3, 12, 'Caffe Latte', 10.99),
         (2, 2, 21, 'Iced Coffe', 12.99),
         (3, 1, 17, 'Americano', 10.99);

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 | [;]

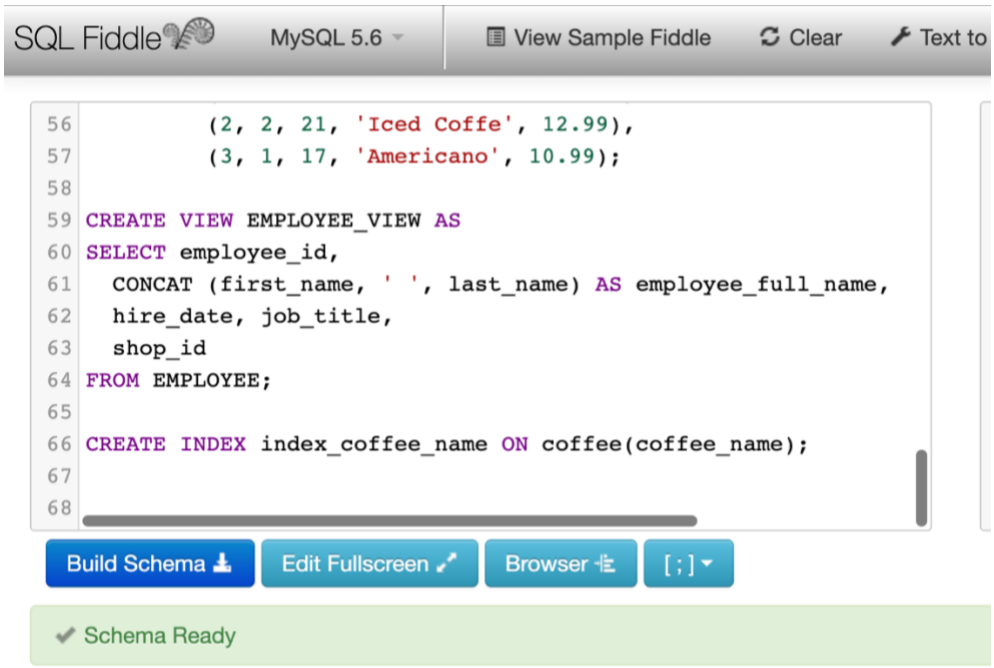
```
SELECT employee_id,
  CONCAT (first_name, ' ', last_name) AS employee_full_name,
  hire_date, job_title,
  shop_id
FROM EMPLOYEE;
```

Run SQL | Edit Fullscreen | [;]

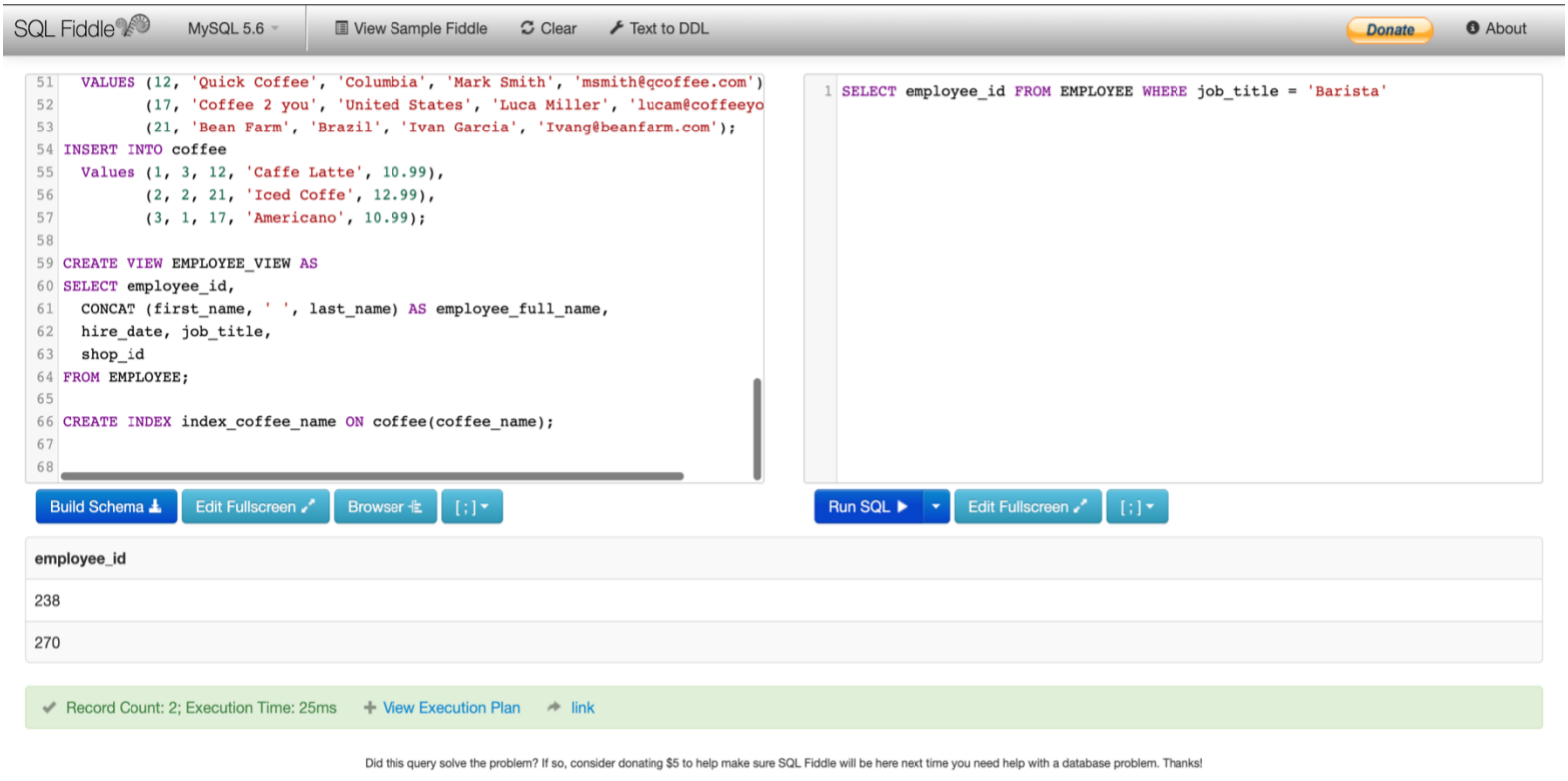
employee_id	employee_full_name	hire_date	job_title	shop_id
197	Layla Jones	2016-03-18	Manager	1
238	Amber Matthews	2019-11-04	Barista	2
270	Dave Coleman	2020-01-20	Barista	3

Record Count: 3; Execution Time: 6ms | View Execution Plan | link

Part B.4 Develop SQL code to create an index on the coffee_name field



Part B.5 Develop SQL code to create an SFW (SELECT–FROM–WHERE) query



Part B.6 Develop SQL code to create a query

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

```
57          (3, 1, 17, 'Americano', 10.99);
58
59  CREATE VIEW EMPLOYEE_VIEW AS
60  SELECT employee_id,
61    CONCAT (first_name, ' ', last_name) AS employee_full_name,
62    hire_date, job_title,
63    shop_id
64  FROM EMPLOYEE;
65
66  CREATE INDEX index_coffee_name ON coffee(coffee_name);
67
68  SELECT COFFEE_SHOP.shop_name AS 'Shop', COFFEE.coffee_name AS 'Coffee Name',
69      FROM ((COFFEE
70      INNER JOIN COFFEE_SHOP ON  COFFEE.shop_id = COFFEE_SHOP.shop_id)
71      INNER JOIN SUPPLIER ON COFFEE.supplier_id = SUPPLIER.supplier_id);
72
73
74
```

Build Schema | Edit Fullscreen | Browser | [;]

```
1  SELECT COFFEE_SHOP.shop_name AS 'Shop', COFFEE.coffee_name AS 'Coffee Name',
2      FROM ((COFFEE
3      INNER JOIN COFFEE_SHOP ON  COFFEE.shop_id = COFFEE_SHOP.shop_id)
4      INNER JOIN SUPPLIER ON COFFEE.supplier_id = SUPPLIER.supplier_id);
5
```

Run SQL | Edit Fullscreen | [;]

Shop	Coffee Name	Price Per Pound	Supplier
Super Beans	Caffe Latte	10.99	Quick Coffee
Starbucks	Iced Coffe	12.99	Bean Farm
Kaladi Brothers	Americano	10.99	Coffee 2 you

Record Count: 3; Execution Time: 23ms | View Execution Plan | link

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!

Complete SQL Code

SQL Fiddle — Exit Fullscreen Schema Editor

```
CREATE DATABASE jaunty_coffee_co;

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),
```

SQL Fiddle — Exit Fullscreen Schema Editor

```
    PRIMARY KEY(coffee_id),
    FOREIGN KEY(shop_id) REFERENCES COFFEE_SHOP(shop_id),
    FOREIGN KEY(supplier_id) REFERENCES SUPPLIER(supplier_id)
);

INSERT INTO coffee_shop
  VALUES (1, 'Kaladi Brothers', 'Anchorage', 'AK'),
         (2, 'Starbucks', 'Chicago', 'IL'),
         (3, 'Super Beans', 'San Diego', 'CA');
INSERT INTO employee
  Values (197, 'Layla', 'Jones', '2016-03-18', 'Manager', 1),
         (238, 'Amber', 'Matthews', '2019-11-04', 'Barista',2),
         (270, 'Dave', 'Coleman', '2020-01-20', 'Barista',3);
INSERT INTO supplier
  VALUES (12, 'Quick Coffee', 'Columbia', 'Mark Smith', 'msmith@qcoffee.com'),
         (17, 'Coffee 2 you', 'United States', 'Luca Miller', 'lucam@coffeeyou.com'),
         (21, 'Bean Farm', 'Brazil', 'Ivan Garcia', 'Ivang@beanfarm.com');
INSERT INTO coffee
  Values (1, 3, 12, 'Caffe Latte', 10.99),
         (2, 2, 21, 'Iced Coffe', 12.99),
         (3, 1, 17, 'Americano', 10.99);

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;

CREATE INDEX index_coffee_name ON coffee(coffee_name);

SELECT employee_id FROM EMPLOYEE WHERE job_title = 'Barista'

SELECT COFFEE_SHOP.shop_name AS 'Shop', COFFEE.coffee_name AS 'Coffee Name', COFFEE.price_per_pound AS 'Price Per Pound', SUPPLIER.company_name AS 'Supplier'
    FROM ((COFFEE
    INNER JOIN COFFEE_SHOP ON  COFFEE.shop_id = COFFEE_SHOP.shop_id)
```