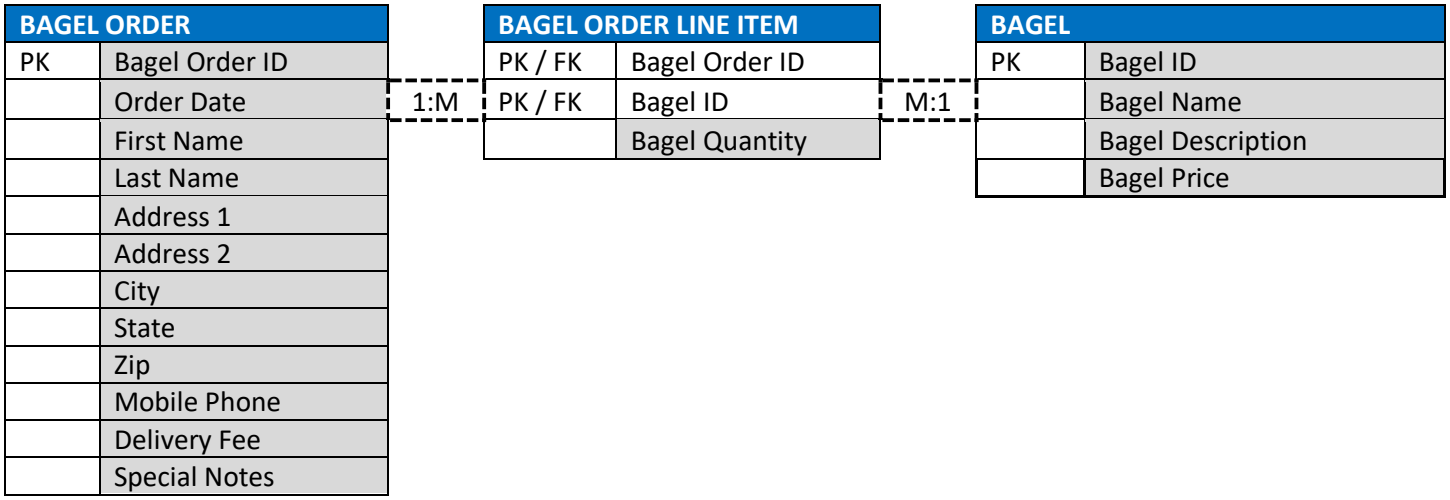
John Salazar
C170 Data Management – Applications
Western Governors University
Student ID: 000650269

PART A

Nora's Bagel Bin Database Blueprints

Second Normal Form (2NF)



1. Each attribute from the provided 1NF table has been assigned to a 2NF table.

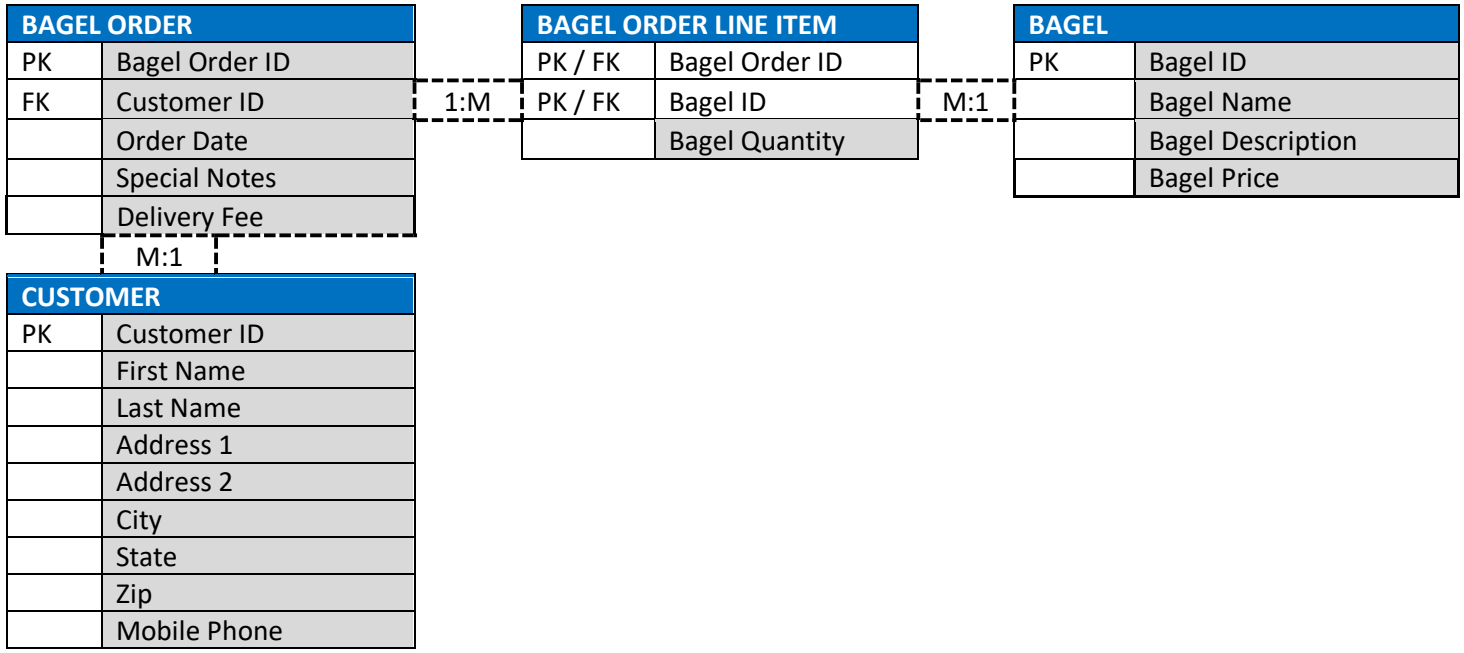
The table **BAGEL ORDER** has a relation of **one to many** with the new **BAGEL ORDER LINE ITEM** table, because a single bagel order can contain many line items. The **BAGEL ORDER LINE ITEM** table then has a **many to one** relation to the **BAGEL** table, as there can be many line items on an order, but each line item will only contain one bagel.

The relationship cardinality explained above was established by determining that each order will have a **Bagel Order ID**, so this has been designated as a **primary key** for **BAGEL ORDER** and is combined with the **Bagel ID** attribute to form a **composite primary key** in the **BAGEL ORDER LINE ITEM** table. **Bagel ID** is then used as a **primary key** to identify the specific bagel in the **BAGEL** table.

The **BAGEL ORDER LINE ITEM** table is also known as an **associative** or **intersection** table, as it helps to normalize repeated values while being associated with the other data from the entity.

Nora's Bagel Bin Database Blueprints

Third Normal Form (3NF)



2. The customer data within the **BAGEL ORDER** table is transitively dependent on **non-unique attributes** or **non-primary keys.** In order to normalize the **2NF** database to a **3NF** database, the **CUSTOMER** table is established with a **primary key 'Customer ID'**. **Customer ID** is then inserted into the **BAGEL ORDER** table and designated as a **foreign key** for association in the **CUSTOMER** table. The rest of the **CUSTOMER** table is then filled with the other attributes pertaining to the customer.

The cardinality remains the same between the original tables **BAGEL ORDER, BAGEL ORDER LINE ITEM,** and **BAGEL**. The new table is established having a **many to one** relation from **BAGEL ORDER** to **CUSTOMER,** as many bagel orders can be made by a single customer and a single customer can make many orders.

Nora's Bagel Bin Database Blueprints

Final Physical Database Model

BAGEL ORDER		
PK	BagelOrderID	INT
FK	CustomerID	INT
	OrderDate	TIMESTAMP
	SpecialNotes	VARCHAR(150)
	DeliveryFee	NUMERIC (2,2)

1:M

BAGEL ORDER LINE ITEM		
PK / FK	BagelOrderID	INT
PK / FK	BagelID	CHAR(2)
	BagelQuantity	INT

M:1

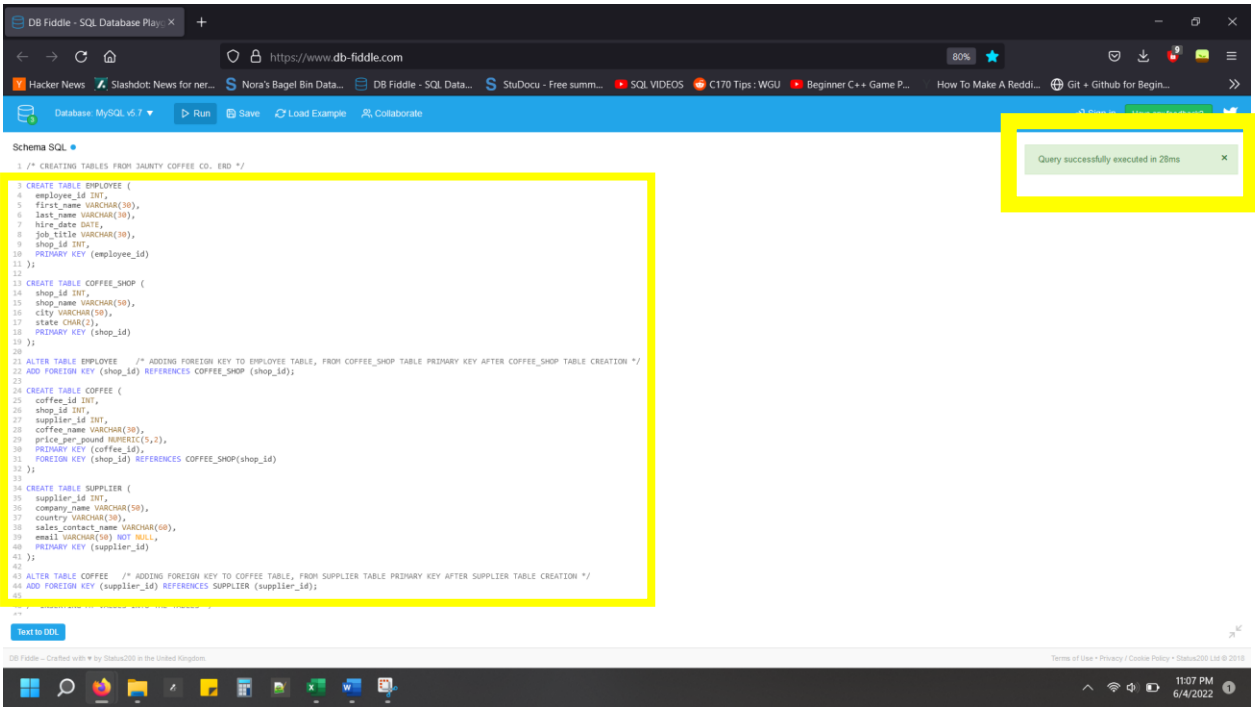
BAGEL		
PK	BagelID	CHAR(2)
	BagelName	VARCHAR(30)
	BagelDescription	VARCHAR(45)
	BagelPrice	NUMERIC(2,2)

M:1

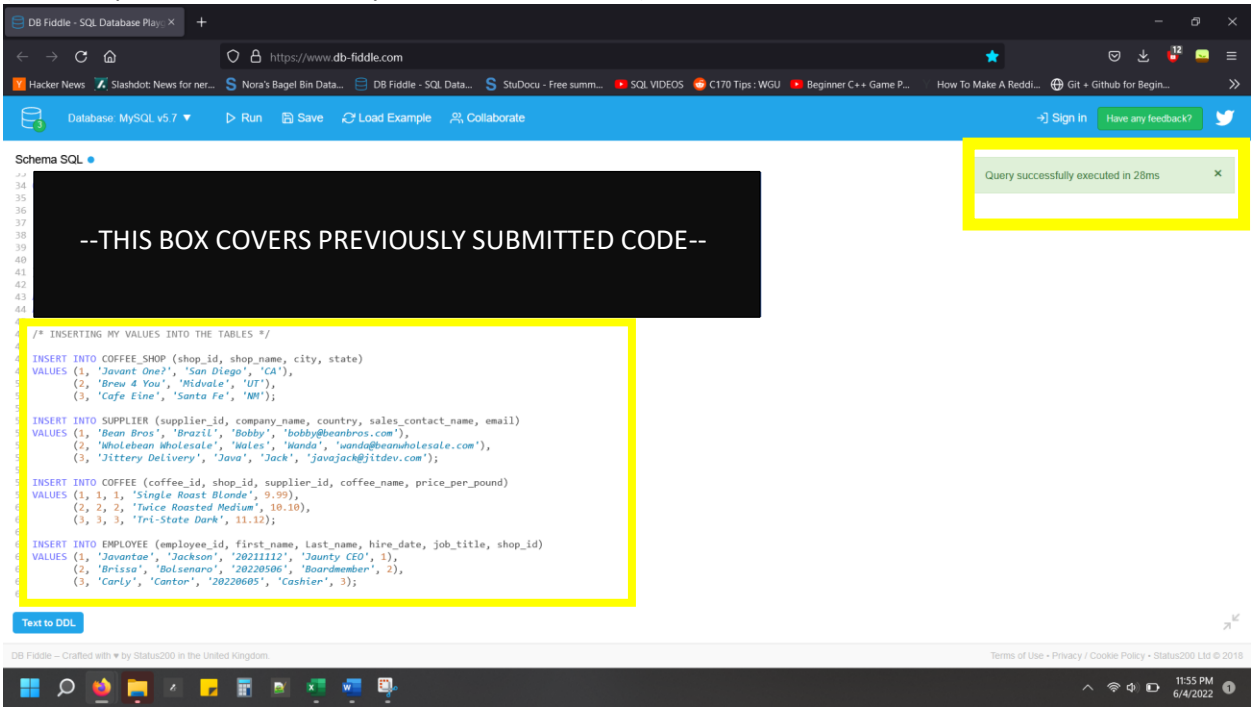
CUSTOMER		
PK	CustomerID	INT
	FirstName	VARCHAR(20)
	LastName	VARCHAR(30)
	Address1	VARCHAR(30)
	Address2	VARCHAR(30)
	City	VARCHAR(30)
	State	VARCHAR(30)
	Zip	INT
	MobilePhone	VARCHAR(12)

PART B (Entire Markdown Following Sectional ScreenShots)

1. CREATING A DATABASE USING JAUNTY COFFEE CO. ERD
(Table Creation – Successfully executed screenshot)



2. POPULATING TABLE WITH MY CHOSEN DATA
(Table Population – Successfully Executed Screenshot)



(SELECT * QUERY FOR EACH TABLE)

SELECT *
FROM EMPLOYEE;

Query #1 Execution time: 0ms

employee_id	first_name	last_name	hire_date	job_title	shop_id
1	Javantae	Jackson	2021-11-12	Jaunty CEO	1
2	Brissa	Bolsenaro	2022-05-06	Boardmember	2
3	Carly	Cantor	2022-06-05	Cashier	3

SELECT *
FROM COFFEE_SHOP;

Query #1 Execution time: 1ms

shop_id	shop_name	city	state
1	Javant One?	San Diego	CA
2	Brew 4 You	Midvale	UT
3	Cafe Eine	Santa Fe	NM

SELECT *
FROM COFFEE;

Query #1 Execution time: 0ms

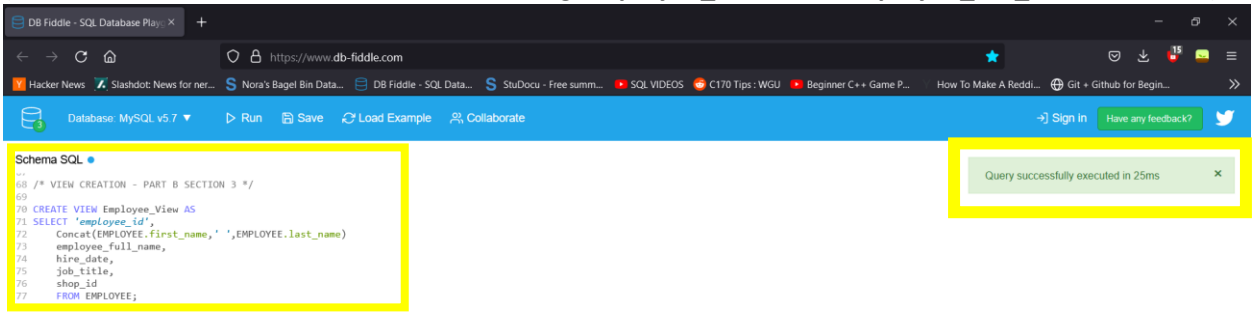
coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
1	1	1	Single Roast Blonde	9.99
2	2	2	Twice Roasted Medium	10.10
3	3	3	Tri-State Dark	11.12

SELECT *
FROM SUPPLIER;

Query #1 Execution time: 1ms

supplier_id	company_name	country	sales_contact_name	email
1	Bean Bros	Brazil	Bobby	bobby@beanbros.com
2	Wholebean Wholesale	Wales	Wanda	wanda@beanwholesale.com
3	Jittery Delivery	Java	Jack	javajack@jitdev.com

3. DEVELOP SQL CODE TO CREATE A VIEW
(View Creation – Successful screenshot showing **employee_view** with **employee_full_name** attribute)



View Table Result:

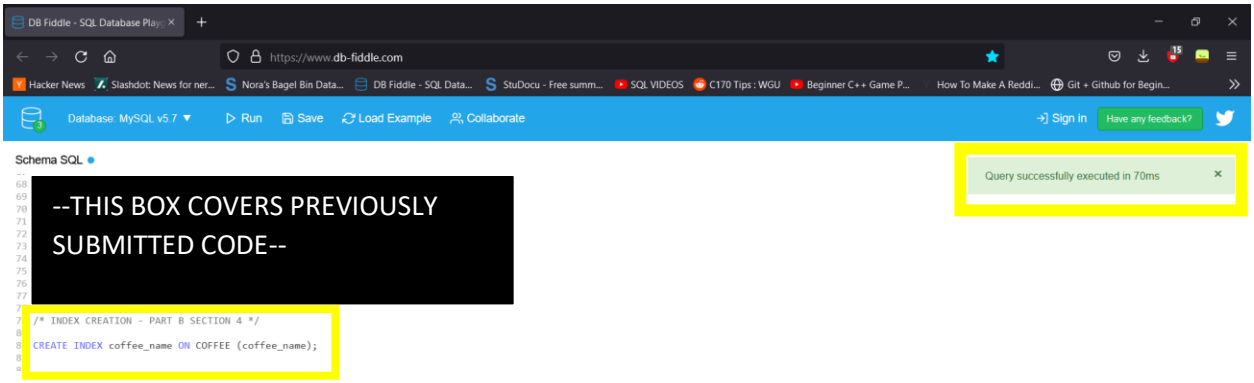
SELECT *

FROM employee_view;

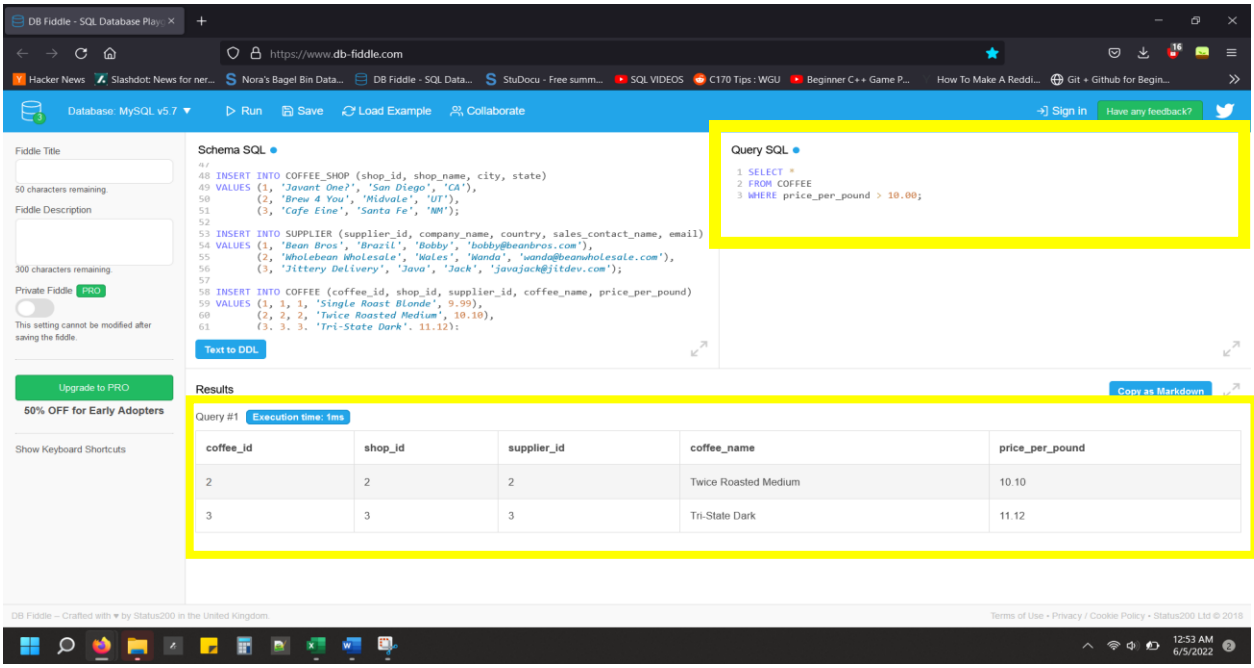
Query #1 **Execution time: 1ms**

employee_id	employee_full_name	hire_date	job_title	shop_id
employee_id	Javantae Jackson	2021-11-12	Jaunty CEO	1
employee_id	Brissa Bolsenaro	2022-05-06	Boardmember	2
employee_id	Carly Cantor	2022-06-05	Cashier	3

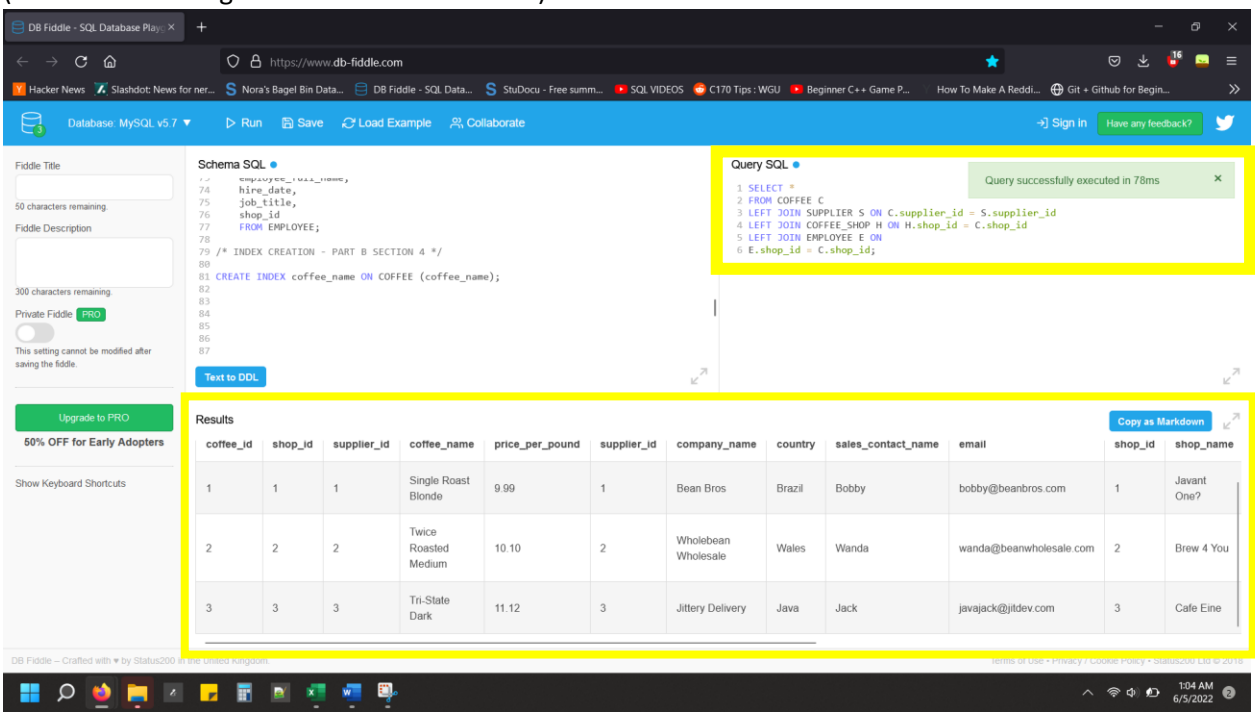
4. DEVELOP SQL CODE TO CREATE AN INDEX ON THE **coffee_name** FIELD
(Index Creation – Successful screenshot)



5. DEVELOP SQL CODE TO CREATE AN SFW QUERY
(SFW query and result)



6. DEVELOP SQL CODE TO CREATE A QUERY
(Table results using at least 3 different tables)



MARKDOWN OF DATABASE CREATION:

Schema (MySQL v5.7)

```
/* CREATING TABLES FROM JAUNTY COFFEE CO. ERD */

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)
);
```

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

ALTER TABLE EMPLOYEE    /* ADDING FOREIGN KEY TO EMPLOYEE TABLE, FROM COFFEE_SHOP TABLE PRIMARY KEY AFTER COFFEE_SHOP TABLE CREATION */
ADD FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP (shop_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)
);

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)
```

```
);

ALTER TABLE COFFEE   /* ADDING FOREIGN KEY TO COFFEE TABLE, FROM SUPPLIER TABLE PRIMARY KEY AFTER SUPPLIER TABLE CREATION */
ADD FOREIGN KEY (supplier_id) REFERENCES SUPPLIER (supplier_id);

/* INSERTING MY VALUES INTO THE TABLES */

INSERT INTO COFFEE_SHOP (shop_id, shop_name, city, state)
VALUES (1, 'Javant One?', 'San Diego', 'CA'),
    (2, 'Brew 4 You', 'Midvale', 'UT'),
        (3, 'Cafe Eine', 'Santa Fe', 'NM');

INSERT INTO SUPPLIER (supplier_id, company_name, country, sales_contact_name, email)
VALUES (1, 'Bean Bros', 'Brazil', 'Bobby', 'bobby@beanbros.com'),
        (2, 'Wholebean Wholesale', 'Wales', 'Wanda', 'wanda@beanwholesale.com'),
    (3, 'Jittery Delivery', 'Java', 'Jack', 'javajack@jitdev.com');

INSERT INTO COFFEE (coffee_id, shop_id, supplier_id, coffee_name, price_per_pound)
VALUES (1, 1, 1, 'Single Roast Blonde', 9.99),
        (2, 2, 2, 'Twice Roasted Medium', 10.10),
    (3, 3, 3, 'Tri-State Dark', 11.12);

INSERT INTO EMPLOYEE (employee_id, first_name, Last_name, hire_date, job_title, shop_id)
VALUES (1, 'Javantae', 'Jackson', '20211112', 'Jaunty CEO', 1),
        (2, 'Brissa', 'Bolsenaro', '20220506', 'Boardmember', 2),
        (3, 'Carly', 'Cantor', '20220605', 'Cashier', 3);

/* VIEW CREATION - PART B SECTION 3 */
```

```
CREATE VIEW Employee_View AS
SELECT 'employee_id',
        Concat(EMPLOYEE.first_name,' ',EMPLOYEE.last_name)
    employee_full_name,
    hire_date,
    job_title,
    shop_id
    FROM EMPLOYEE;

/* INDEX CREATION - PART B SECTION 4 */

CREATE INDEX coffee_name ON COFFEE (coffee_name);
```