Benjamin Prendergast, Student ID: 00[redacted]
2023/1/11
VHT2

Part A: Nora's Bagel Bin Database Blueprints

A1a/A1b.

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

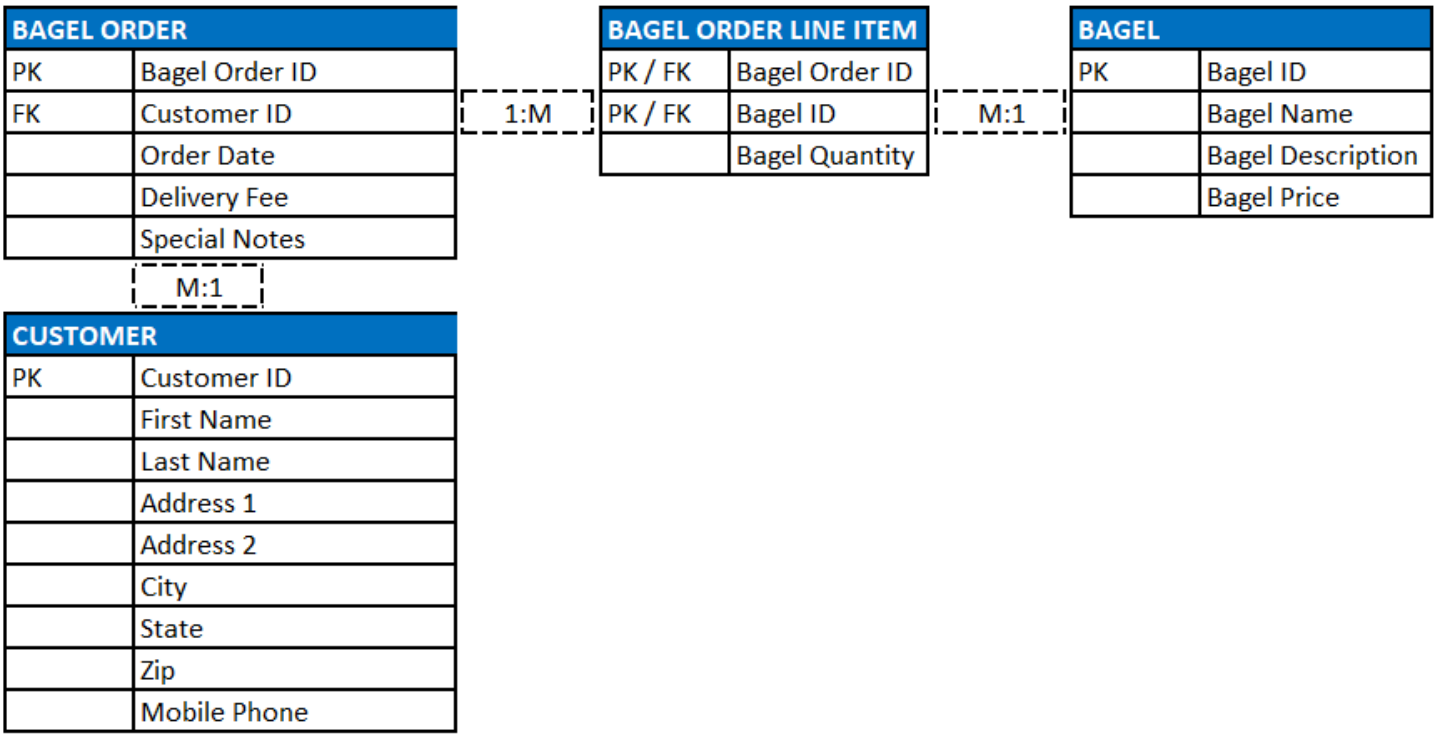
A1c.

The definition of second normal form says that all requirements of first normal form are met, and all non-key attributes must be functionally dependent on the entire primary key. The only attribute in the 1NF relation that depends on the entire primary key (*Bagel Order ID, Bagel ID*) is *Bagel Quantity*. Attributes *Bagel Name*, *Bagel Description*, and *Bagel Price* depend only on *Bagel ID*, but not *Bagel Order ID*, and can be moved to a separate relation. *Order Date, Delivery Fee, Special Notes*, and fields pertaining to customer information depend only on *Bagel Order ID*, but not *Bagel ID*, and can be moved to a separate relation. By removing the attributes not dependent on the entire primary key and placing them into separate relations in which they are dependent on the entire primary key, second normal form compliance is achieved.

Relationship cardinality can be determined by considering the actual ordering process. A Bagel Order can include many Bagel Order Line Items, while a Bagel Order Line Item belongs to one particular Bagel Order (1:M). A Bagel Order Line Item refers to one type of bagel, while a Bagel can belong to many Bagel Order Line Items(M:1).

A2a/A2b/A2c/A2d.

Third Normal Form (3NF)



A2e.

The definition of third normal form says that all requirements of second normal form are met, and all non-key attributes depend only on the primary key. Attributes pertaining to customer information do depend on *Bagel Order ID*. However, those attributes also depend on the identity of the customer, or a *Customer ID* attribute. By removing the attributes that depend on *Customer ID* from the *Bagel Order* relation and placing them into a separate *Customer* relation, third normal form compliance is achieved.

The cardinality of the relationships between *Bagel Order* and *Customer* can be determined by considering the ordering process. A Bagel Order is placed by one individual Customer, while an individual Customer can place many Bagel Orders (M:1).

A3a/A3b.

Final Physical Database Model

BAGEL ORDER		
PK	bagel_order_id	INT
FK	customer_id	INT
	order_date	TIMESTAMP
	delivery_fee	NUMERIC(4,2)
	special_notes	VARCHAR(80)

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(30)
	bagel_description	VARCHAR(50)
	bagel_price	NUMERIC(3,2)

M:1

CUSTOMER		
PK	customer_id	INT
	first_name	VARCHAR(20)
	last_name	VARCHAR(20)
	address_1	VARCHAR(30)
	address_2	VARCHAR(30)
	city	VARCHAR(20)
	state	CHAR(2)
	zip	CHAR(5)
	mobile_phone	CHAR(10)

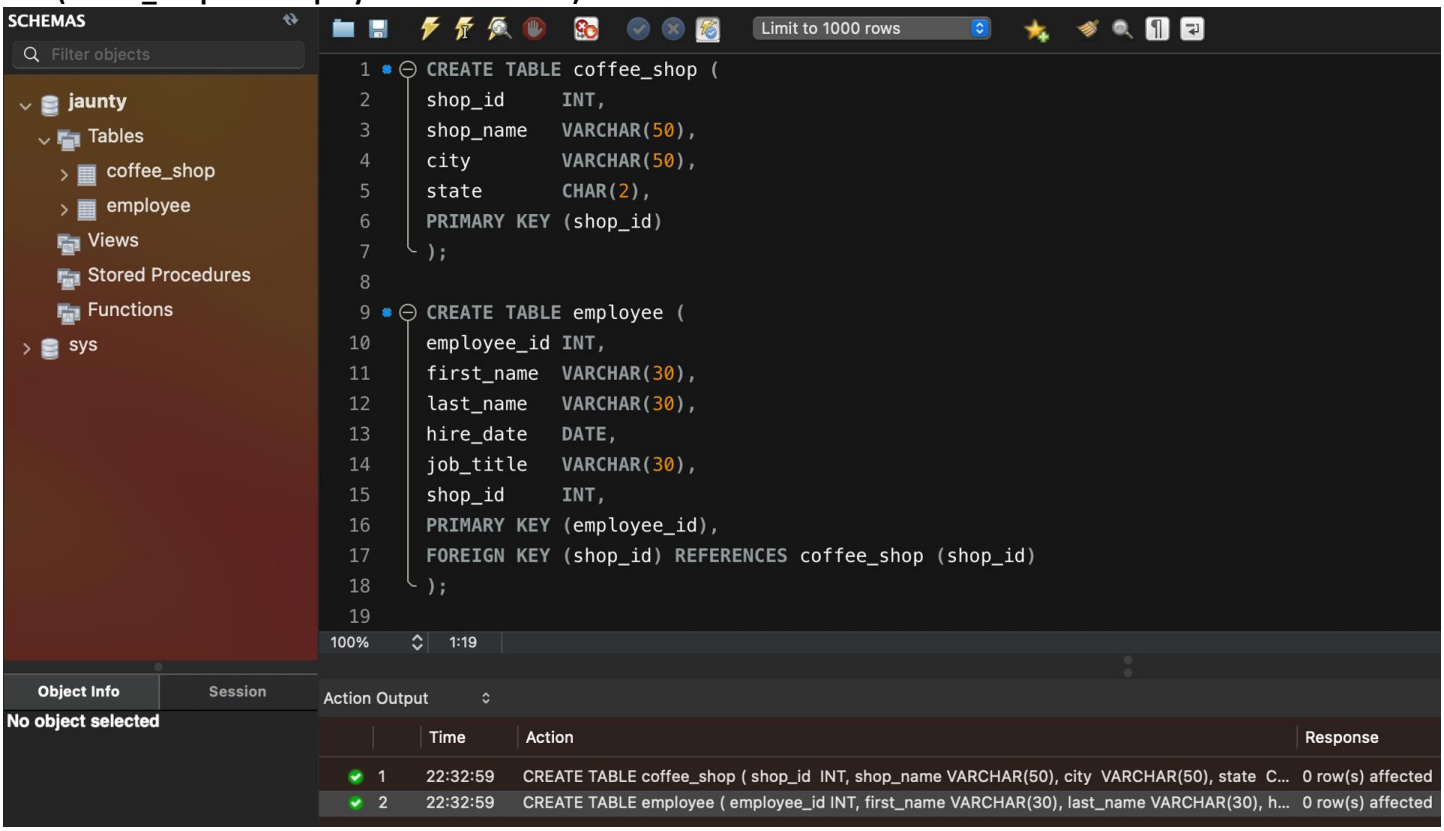
Part B: Jaunty Coffee Co. Database

B1a (coffee_shop and employee table creation).

```
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)
);
```

B1b (coffee_shop and employee table creation).

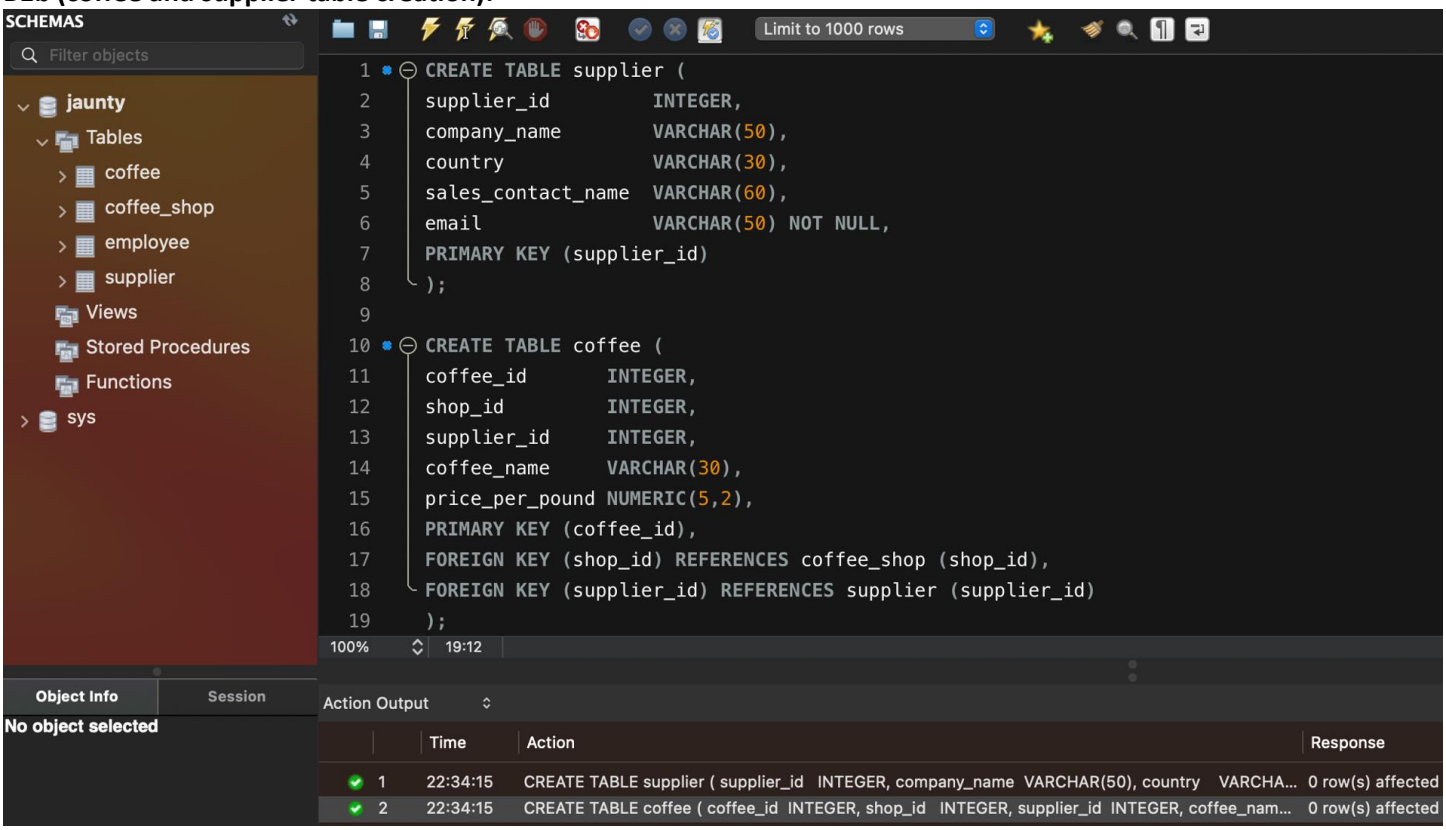


B1a (coffee and supplier table creation).

```
CREATE TABLE supplier (
supplier_id         INTEGER,
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       INTEGER,
shop_id         INTEGER,
supplier_id     INTEGER,
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)
);
```

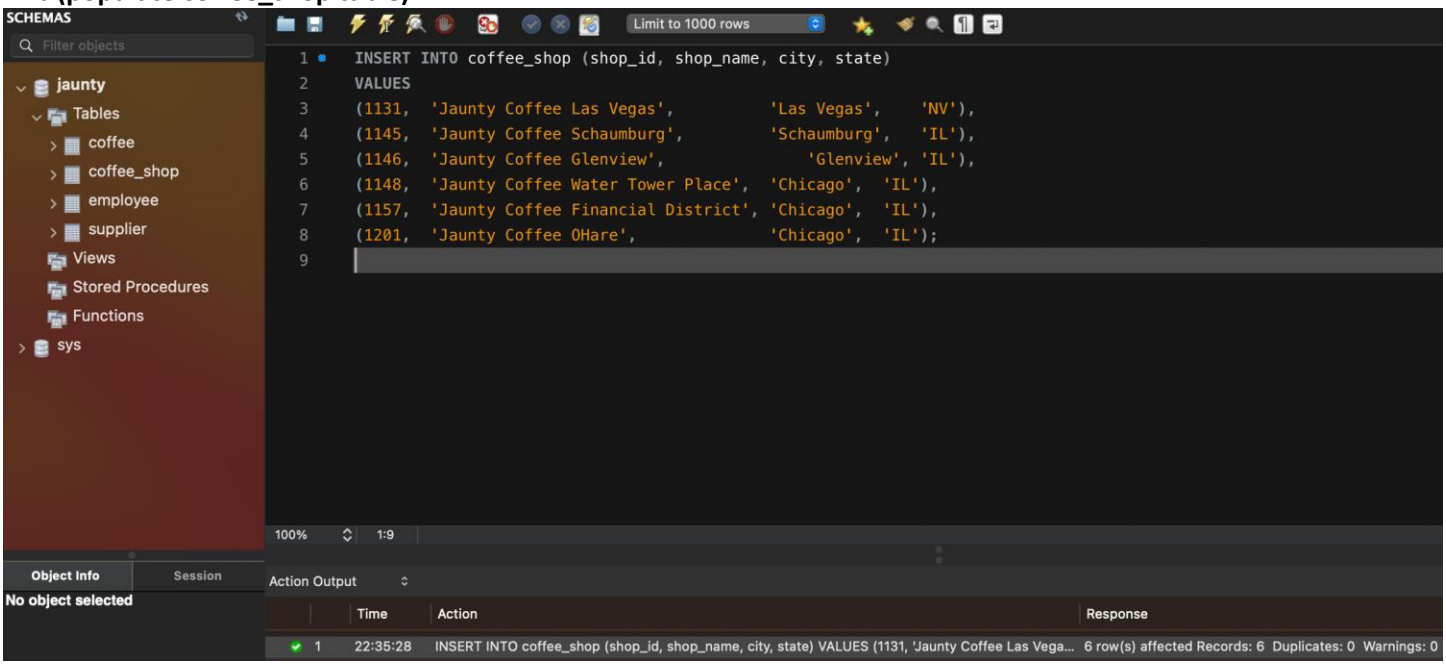
B1b (coffee and supplier table creation).



B2a (populate coffee_shop table).

```
INSERT INTO coffee_shop (shop_id, shop_name, city, state)
VALUES
(1131,  'Jaunty Coffee Las Vegas',          'Las Vegas',    'NV'),
(1145,  'Jaunty Coffee Schaumburg',         'Schaumburg',   'IL'),
(1146,  'Jaunty Coffee Glenview',              'Glenview', 'IL'),
(1148,  'Jaunty Coffee Water Tower Place',  'Chicago',  'IL'),
(1157,  'Jaunty Coffee Financial District', 'Chicago',  'IL'),
(1201,  'Jaunty Coffee OHare',              'Chicago',  'IL');
```

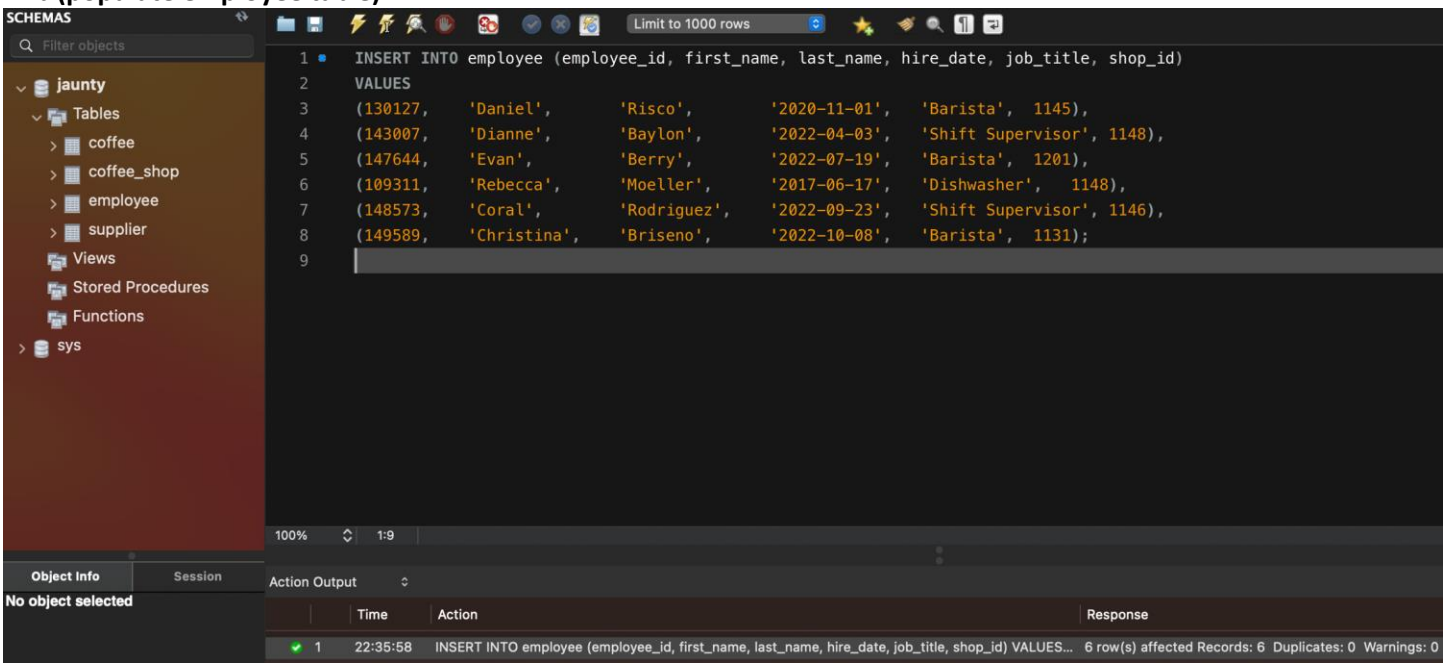
B2b (populate coffee_shop table).



B2a (populate employee table).

```
INSERT INTO employee (employee_id, first_name, last_name, hire_date, job_title, shop_id)
VALUES
(130127,    'Daniel',       'Risco',        '2020-11-01',   'Barista',  1145),
(143007,    'Dianne',       'Baylon',       '2022-04-03',   'Shift Supervisor', 1148),
(147644,    'Evan',         'Berry',        '2022-07-19',   'Barista',  1201),
(109311,    'Rebecca',      'Moeller',      '2017-06-17',   'Dishwasher',   1148),
(148573,    'Coral',        'Rodriguez',    '2022-09-23',   'Shift Supervisor', 1146),
(149589,    'Christina',    'Briseno',      '2022-10-08',   'Barista',  1131);
```

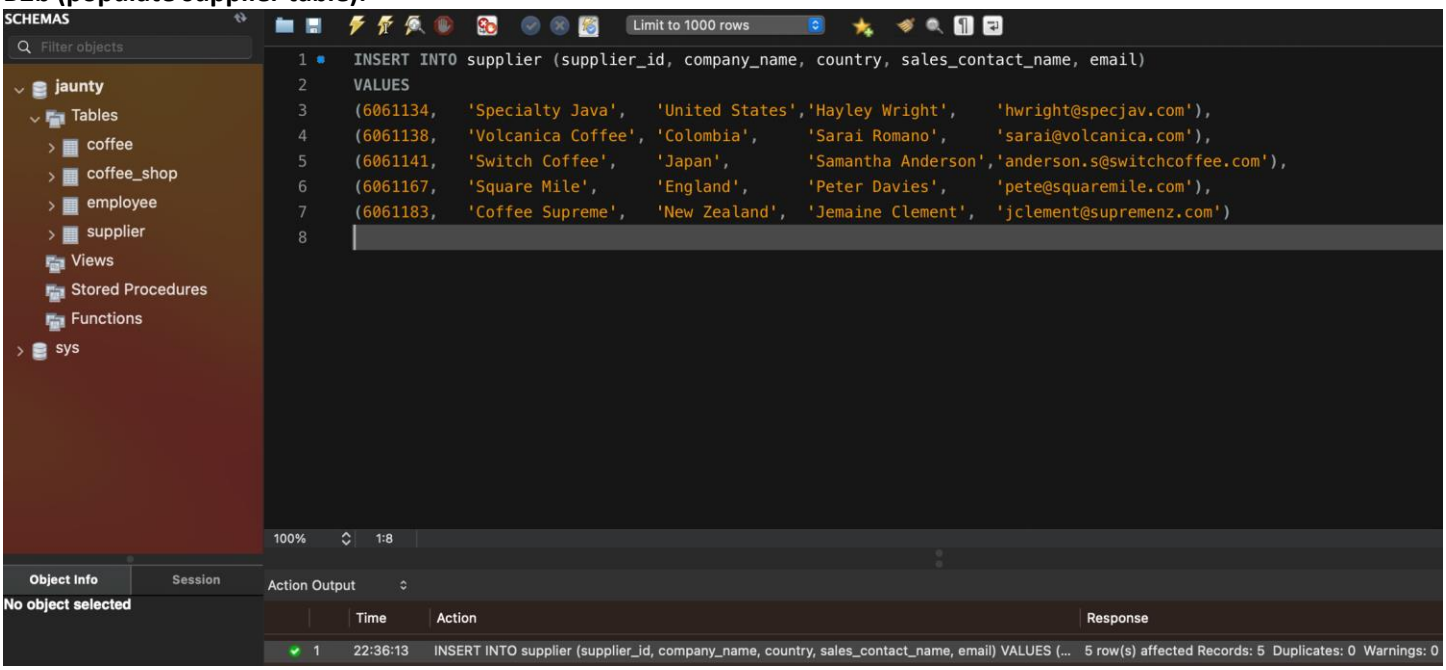
B2b (populate employee table).



B2a (populate supplier table).

```
INSERT INTO supplier (supplier_id, company_name, country, sales_contact_name, email)
VALUES
(6061134,   'Specialty Java',   'United States','Hayley Wright',
'hwright@specjav.com'),
(6061138,   'Volcanica Coffee', 'Colombia',     'Sarai Romano',
'sarai@volcanica.com'),
(6061141,   'Switch Coffee',    'Japan',        'Samantha
Anderson','anderson.s@switchcoffee.com'),
(6061167,   'Square Mile',      'England',      'Peter Davies',
'pete@squaremile.com'),
(6061183,   'Coffee Supreme',   'New Zealand',  'Jemaine Clement',
'jclement@supremenz.com')
```

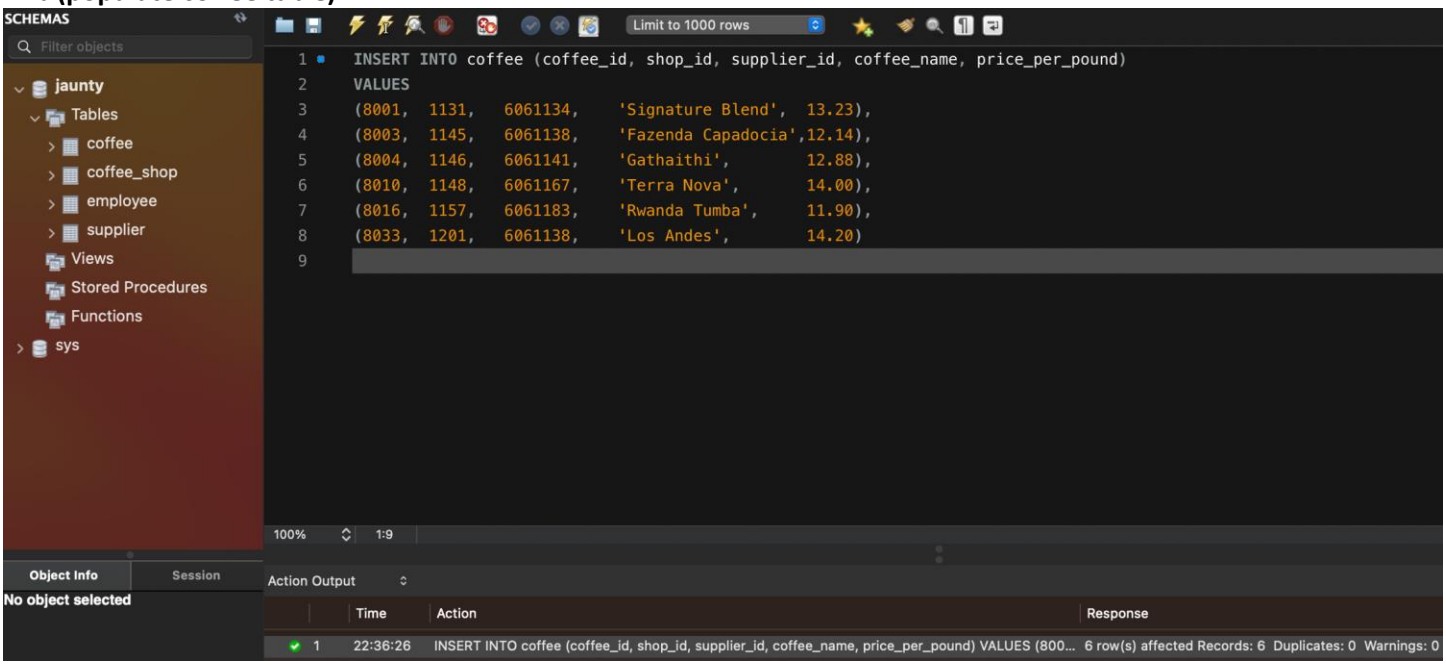
B2b (populate supplier table).



B2a (populate coffee table).

```
INSERT INTO coffee (coffee_id, shop_id, supplier_id, coffee_name, price_per_pound)
VALUES
(8001,  1131,   6061134,    'Signature Blend',  13.23),
(8003,  1145,   6061138,    'Fazenda Capadocia',12.14),
(8004,  1146,   6061141,    'Gathaithi',        12.88),
(8010,  1148,   6061167,    'Terra Nova',       14.00),
(8016,  1157,   6061183,    'Rwanda Tumba',     11.90),
(8033,  1201,   6061138,    'Los Andes',        14.20)
```

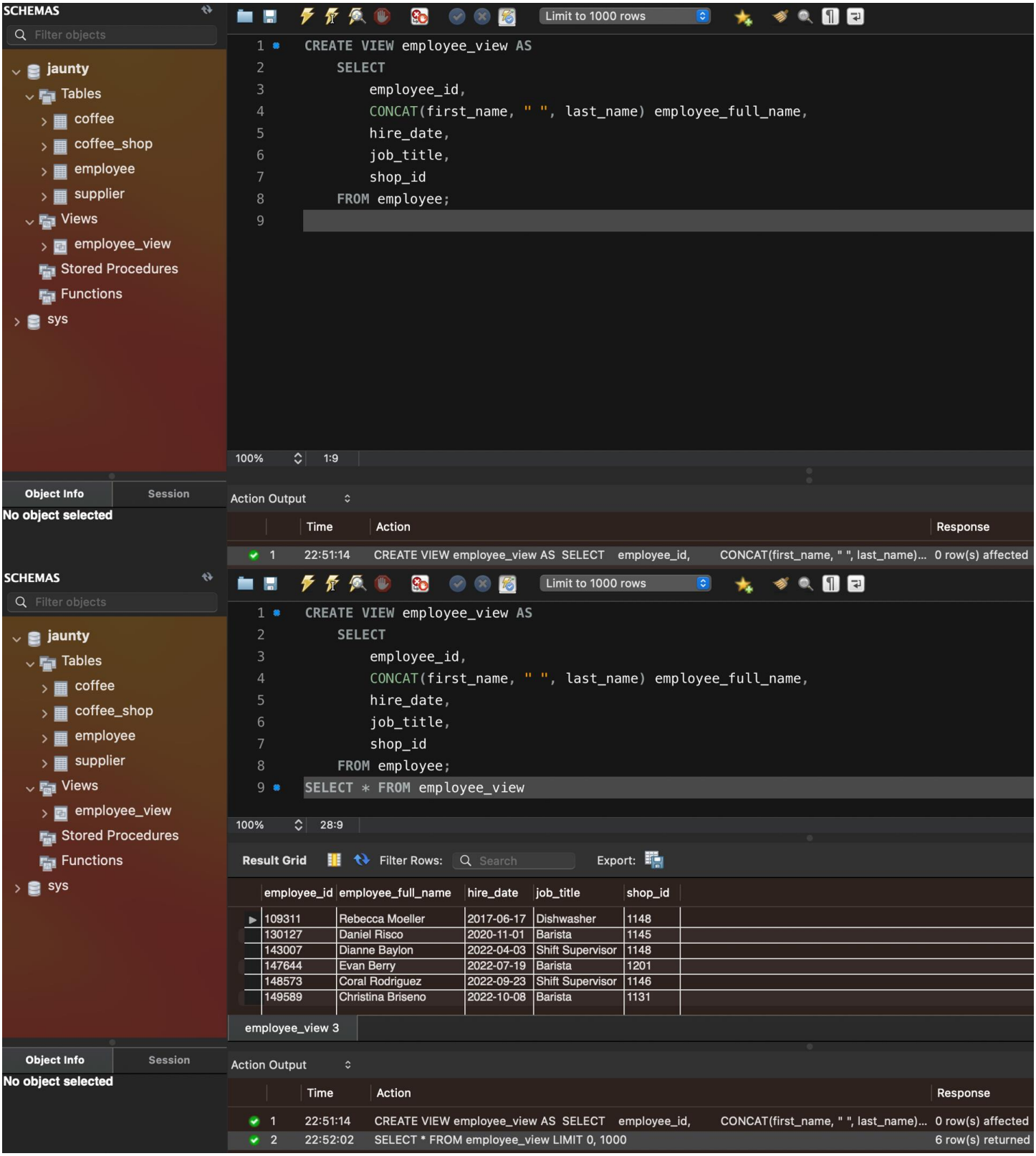
B2b (populate coffee table).



B3a.

```
CREATE VIEW employee_view AS
    SELECT
        employee_id,
        CONCAT(first_name, " ", last_name) employee_full_name,
        hire_date,
        job_title,
        shop_id
    FROM employee;
```

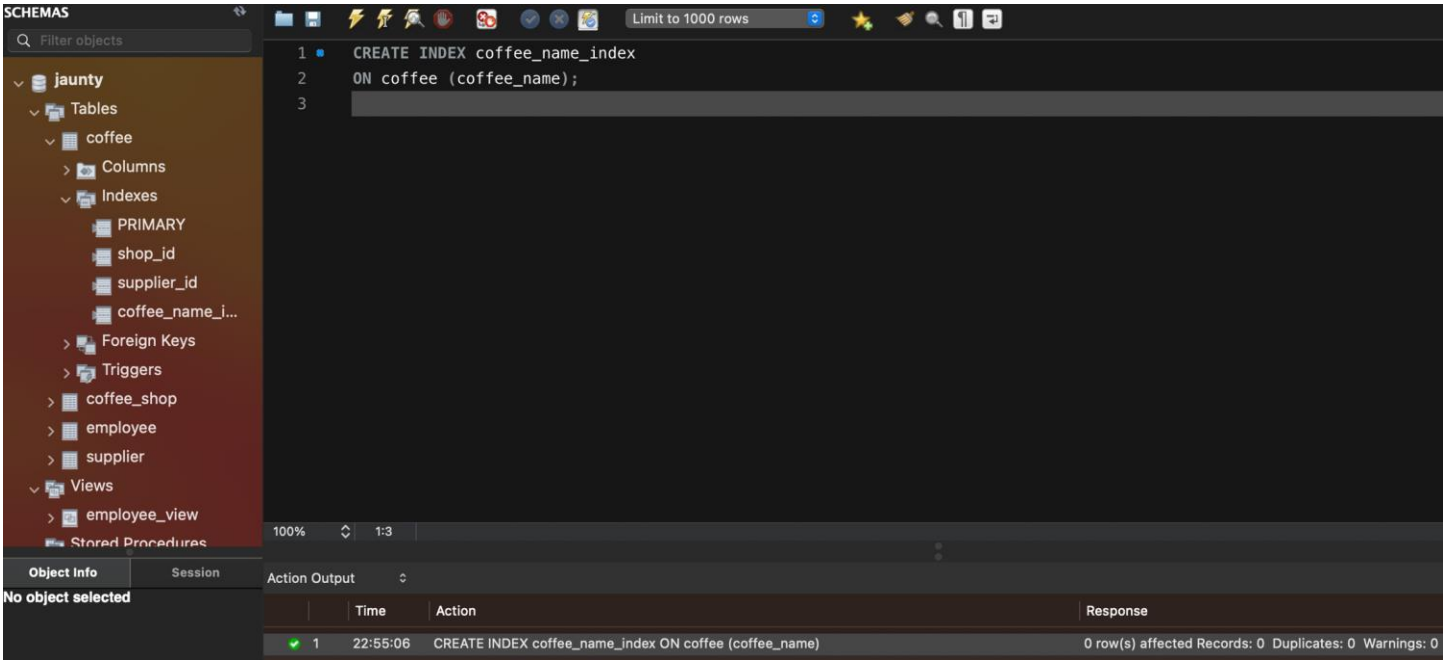
B3b.



B4a.

```
CREATE INDEX coffee_name_index
ON coffee (coffee_name);
```

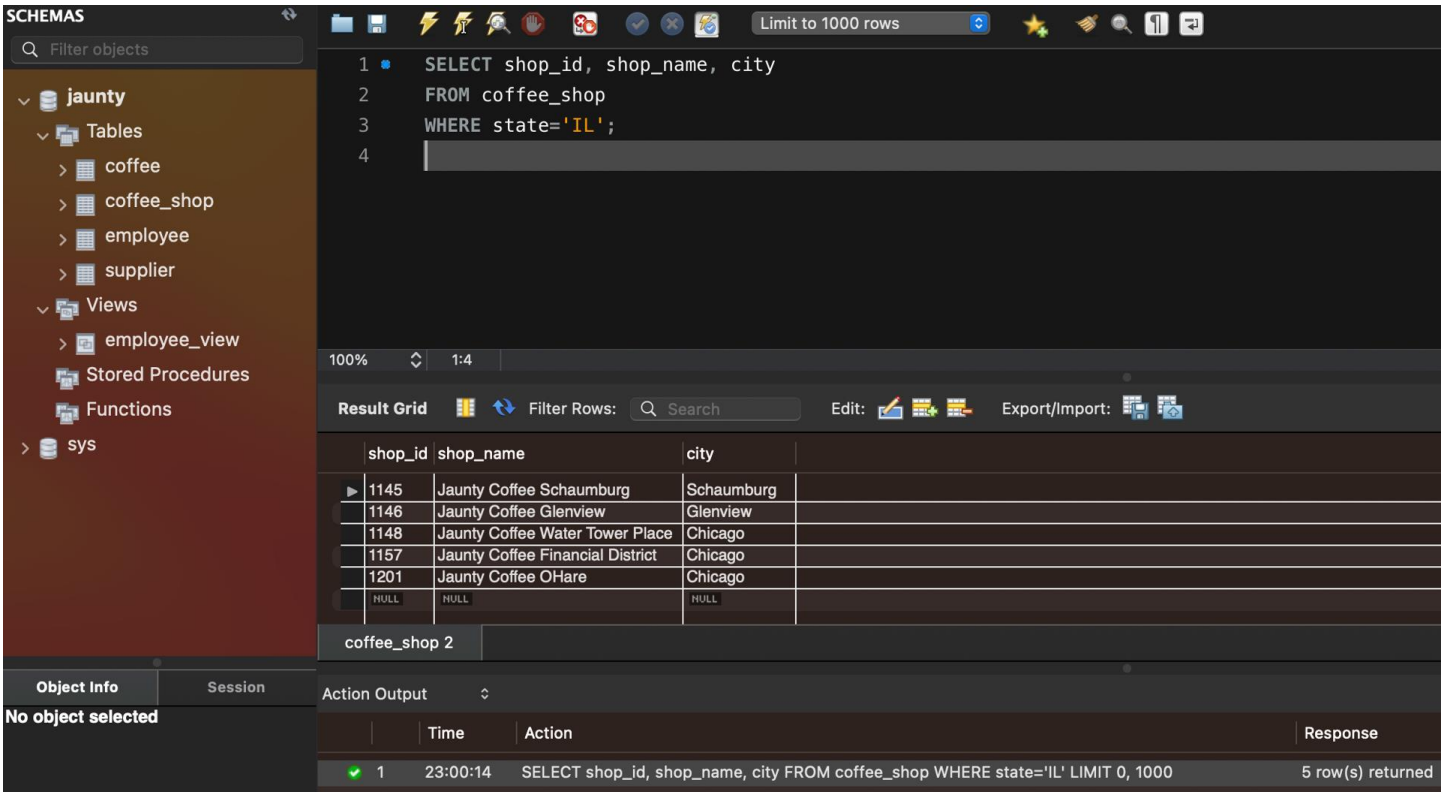
B4b.



B5a.

```
SELECT shop_id, shop_name, city
FROM coffee_shop
WHERE state='IL';
```

B5b.



B6a.

```
SELECT coffee_shop.shop_id, coffee.coffee_name, supplier.sales_contact_name,
supplier.email
FROM coffee
JOIN coffee_shop ON coffee.shop_id=coffee_shop.shop_id
JOIN supplier ON coffee.supplier_id=supplier.supplier_id
```

B6b.