Ruth Sablich
WGU

C170 VHT2 — VHT2 Task 1: Normalization and Database Design

Part A

A. 1. *Nora's Bagel Bin Database Blueprints*

Second Normal Form (2NF)

BAGEL ORDER	
PK	Bagel Order ID
	Order Date
	First Name
	Last Name
	Address 1
	Address 2
	City
	State
	Zip Code
	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

A. 1. c.

I assigned the attributes to the 2NF table according to their relationships with the composite keys in BAGEL ORDER LINE ITEM. If they depend on only one part of the primary key, they are assigned to their appropriate new table, named correspondingly with BAGEL ORDER or BAGEL. If they depend on both, I assigned them (Bagel Quantity) to the one "original" table, BAGEL ORDER LINE ITEM table which contains foreign keys of both new tables' primary keys (Bagel Order ID and Bagel ID). For cardinality, a single BAGEL ORDER can contain many BAGEL ORDER LINE ITEMs, as in one order must have at least one, but can have many, line item(s). Furthermore, at least one BAGEL ORDER LINE ITEM can have many BAGELs, as in there can be many line items, each allowing only one bagel per item.

A. 2.

Third Normal Form (3NF)

BAGEL ORDER	
PK	Bagel Order ID
FK	Customer ID
	Special Notes
	Delivery Fee
	Order Date

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

M:1

CUSTOMER	
PK	Customer ID
	First Name
	Last Name
	Address 1
	Address 2
	City
	State
	Zip Code
	Mobile Phone

A. 2. e.

I assigned the attributes to the 3NF table by further eliminating redundancy within the tables and creating a fourth table, CUSTOMER, that holds all previously redundant information regarding the customer (first and last names, addresses, city, state, zip, and mobile phone). A new foreign key in BAGEL ORDER table named Customer ID became the new primary key in the CUSTOMER table as a result of this, as well as the fact that a bagel order must have a relationship with the customer ordering. All other attributes remained the same. The cardinalities also remained the same, except for establishing a cardinality between BAGEL ORDER table and CUSTOMER table. This new cardinality is M:1 because a single customer must order at least one, but overall can have, many orders.

A. 3.

Final Physical Database Model

BAGEL ORDER		
PK	bagel_order_id	INTEGER
FK	customer_id	INTEGER
	special_notes	VARCHAR(40)
	delivery_fee	INTEGER
	order_date	TIMESTAMP

1:M

M:1

CUSTOMER		
PK	customer_id	INTEGER
	first_name	VARCHAR(40)
	last_name	VARCHAR(40)
	address_1	VARCHAR(30)
	address_2	VARCHAR(30)
	city	VARCHAR(30)
	state	CHAR(2)
	zip_code	INTEGER
	mobile_phone	VARCHAR(20)

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

M:1

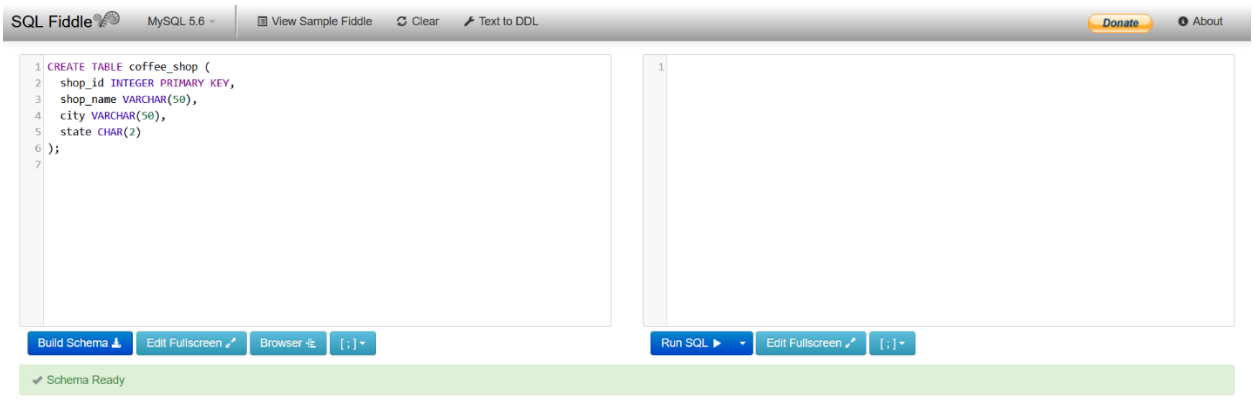
BAGEL		
PK	bagel_id	CHAR(2)
	bagel_name	VARCHAR(30)
	bagel_description	VARCHAR(30)
	bagel_price	NUMERIC(3,2)

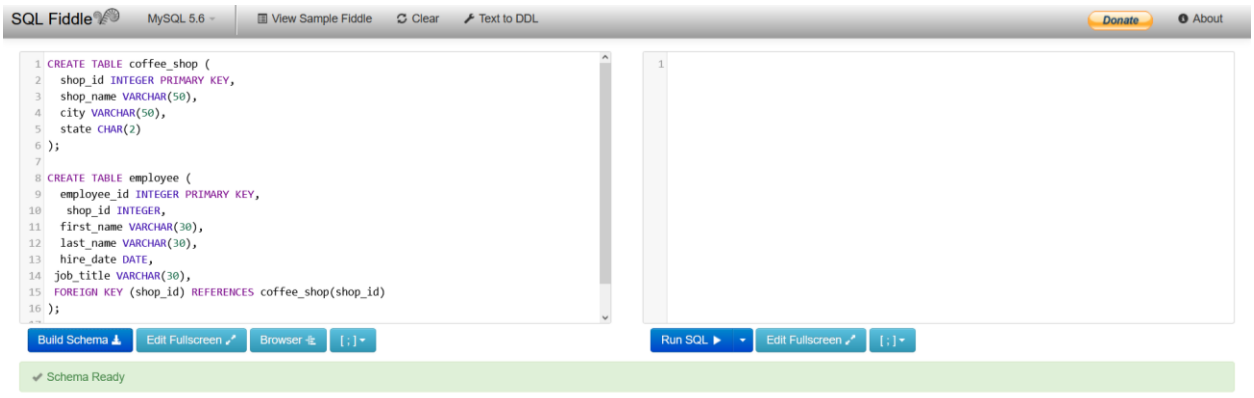
PART B

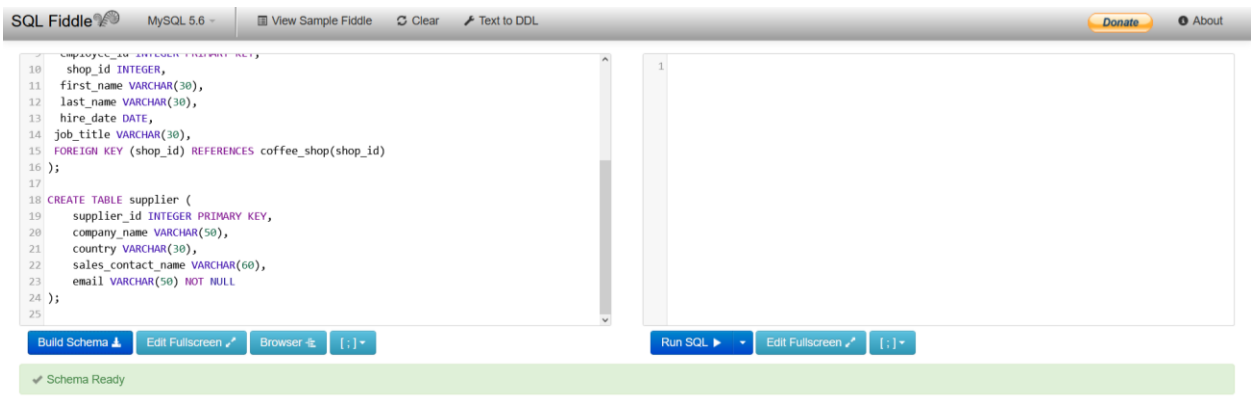
B. 1. a.

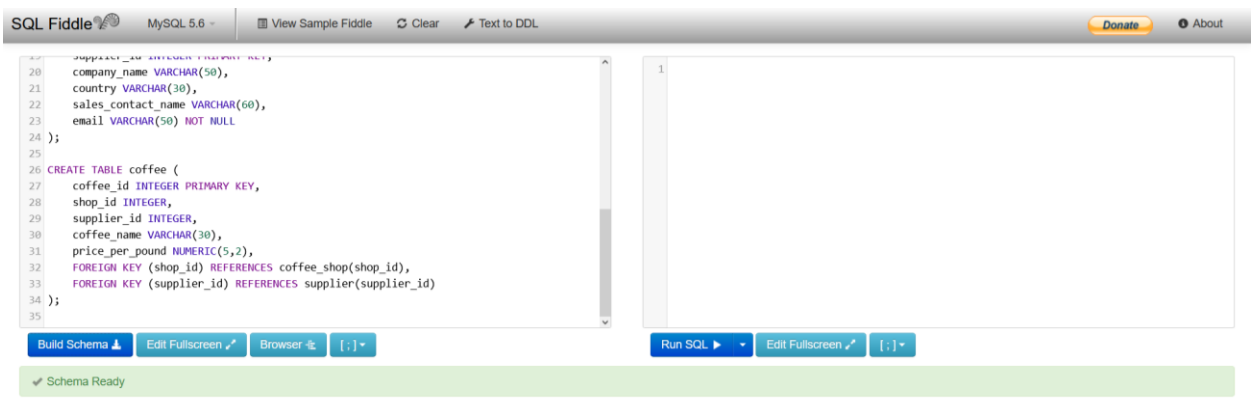
```
CREATE TABLE coffee_shop (
        shop_id INTEGER PRIMARY KEY,
        shop_name VARCHAR(50),
        city VARCHAR(50),
        state CHAR(2)
);

CREATE TABLE employee (
        employee_id INTEGER PRIMARY KEY,
        shop_id INTEGER,
        first_name VARCHAR(30),
        last_name VARCHAR(30),
        hire_date DATE,
        job_title VARCHAR(30),
        FOREIGN KEY (shop_id) REFERENCES coffee_shop(shop_id)
);

CREATE TABLE supplier (
        supplier_id INTEGER PRIMARY KEY,
        company_name VARCHAR(50),
        country VARCHAR(30),
        sales_contact_name VARCHAR(60),
        email VARCHAR(50) NOT NULL
);

CREATE TABLE coffee (
        coffee_id INTEGER PRIMARY KEY,
        shop_id INTEGER,
        supplier_id INTEGER,
        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)
);
```

B. 1. b.


SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
CREATE TABLE coffee_shop (
shop_id INTEGER PRIMARY KEY,
shop_name VARCHAR(50),
city VARCHAR(50),
state CHAR(2)
);
Build Schema
Edit Fullscreen
Browser
[;]
Run SQL
Edit Fullscreen
[;]
Schema Ready


SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
CREATE TABLE coffee_shop (
shop_id INTEGER PRIMARY KEY,
shop_name VARCHAR(50),
city VARCHAR(50),
state CHAR(2)
);
CREATE TABLE employee (
employee_id INTEGER PRIMARY KEY,
shop_id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
hire_date DATE,
job_title VARCHAR(30),
FOREIGN KEY (shop_id) REFERENCES coffee_shop(shop_id)
);
Build Schema
Edit Fullscreen
Browser
[;]
Run SQL
Edit Fullscreen
[;]
Schema Ready

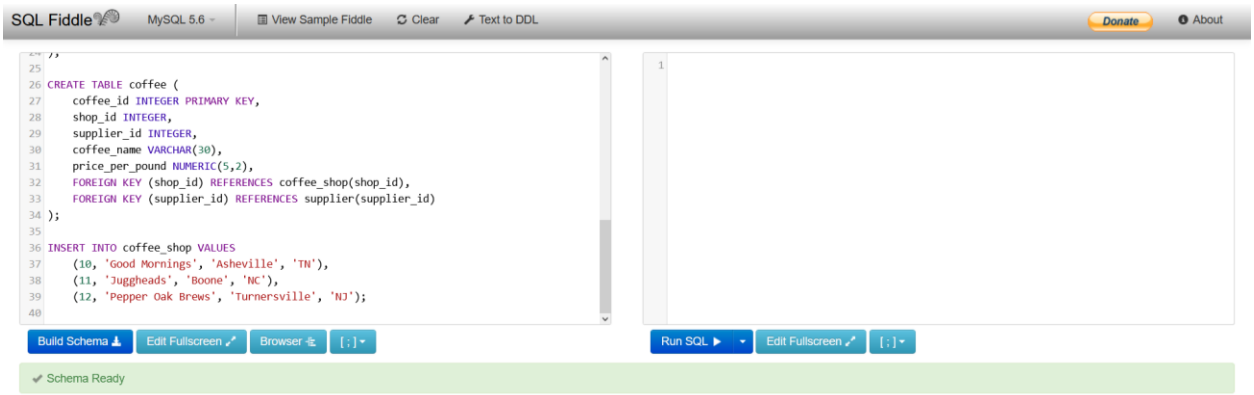

SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
shop_id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
hire_date DATE,
job_title VARCHAR(30),
FOREIGN KEY (shop_id) REFERENCES coffee_shop(shop_id)
);
CREATE TABLE supplier (
supplier_id INTEGER PRIMARY KEY,
company_name VARCHAR(50),
country VARCHAR(30),
sales_contact_name VARCHAR(60),
email VARCHAR(50) NOT NULL
);
Build Schema
Edit Fullscreen
Browser
[;]
Run SQL
Edit Fullscreen
[;]
Schema Ready

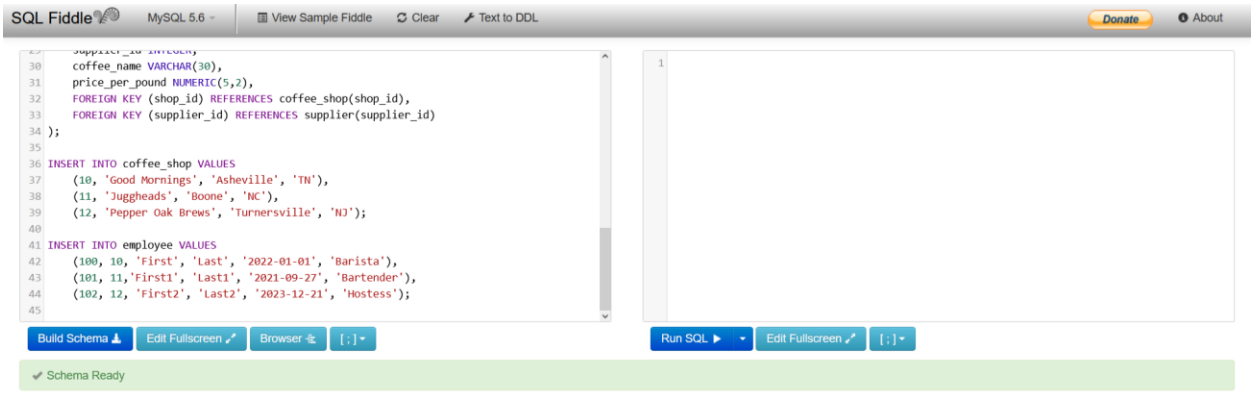

SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
company_name VARCHAR(50),
country VARCHAR(30),
sales_contact_name VARCHAR(60),
email VARCHAR(50) NOT NULL
);
CREATE TABLE coffee (
coffee_id INTEGER PRIMARY KEY,
shop_id INTEGER,
supplier_id INTEGER,
coffee_name VARCHAR(30),
price_per_pound NUMERIC(5,2),
FOREIGN KEY (shop_id) REFERENCES coffee_shop(shop_id),
FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)
);
Build Schema
Edit Fullscreen
Browser
[;]
Run SQL
Edit Fullscreen
[;]
Schema Ready

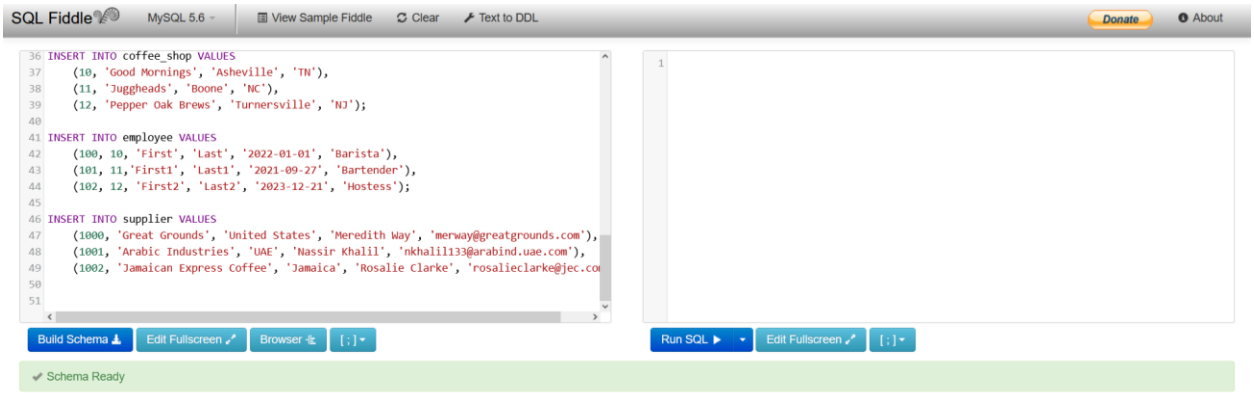
B. 2. a.

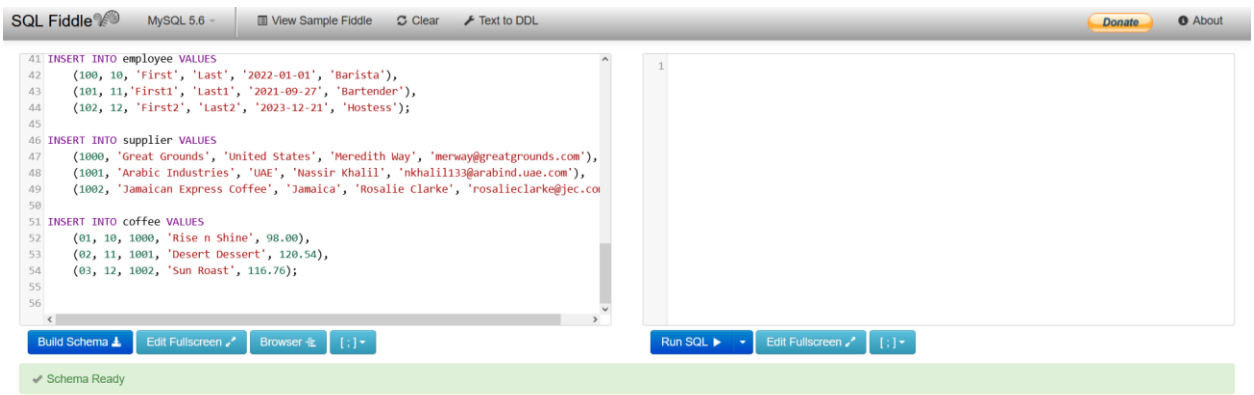
```
INSERT INTO coffee_shop VALUES
        (10, 'Good Mornings', 'Asheville', 'TN'),
        (11, 'Juggheads', 'Boone', 'NC'),
        (12, 'Pepper Oak Brews', 'Turnersville', 'NJ');

INSERT INTO employee VALUES
        (100, 10, 'First', 'Last', '2022-01-01', 'Barista'),
        (101, 11,'First1', 'Last1', '2021-09-27', 'Bartender'),
        (102, 12, 'First2', 'Last2', '2023-12-21', 'Hostess');

INSERT INTO supplier VALUES
        (1000, 'Great Grounds', 'United States', 'Meredith Way', 'merway@greatgrounds.com'),
        (1001, 'Arabic Industries', 'UAE', 'Nassir Khalil', 'nkhalil133@arabind.uae.com'),
        (1002, 'Jamaican Express Coffee', 'Jamaica', 'Rosalie Clarke', 'rosalieclarke@jec.com');

INSERT INTO coffee VALUES
        (01, 50, 1000, 'Rise n Shine', 98.00),
        (02, 51, 1001, 'Desert Dessert', 120.54),
        (03, 52, 1002, 'Sun Roast', 116.76);
```

B. 2. b.




SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
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)
);
INSERT INTO coffee_shop VALUES
(10, 'Good Mornings', 'Asheville', 'TN'),
(11, 'Juggheads', 'Boone', 'NC'),
(12, 'Pepper Oak Brews', 'Turnersville', 'NJ');
INSERT INTO employee VALUES
(100, 10, 'First', 'Last', '2022-01-01', 'Barista'),
(101, 11,'First1', 'Last1', '2021-09-27', 'Bartender'),
(102, 12, 'First2', 'Last2', '2023-12-21', 'Hostess');
Build Schema
Edit Fullscreen
Browser
[;]
Run SQL
Edit Fullscreen
[;]
Schema Ready

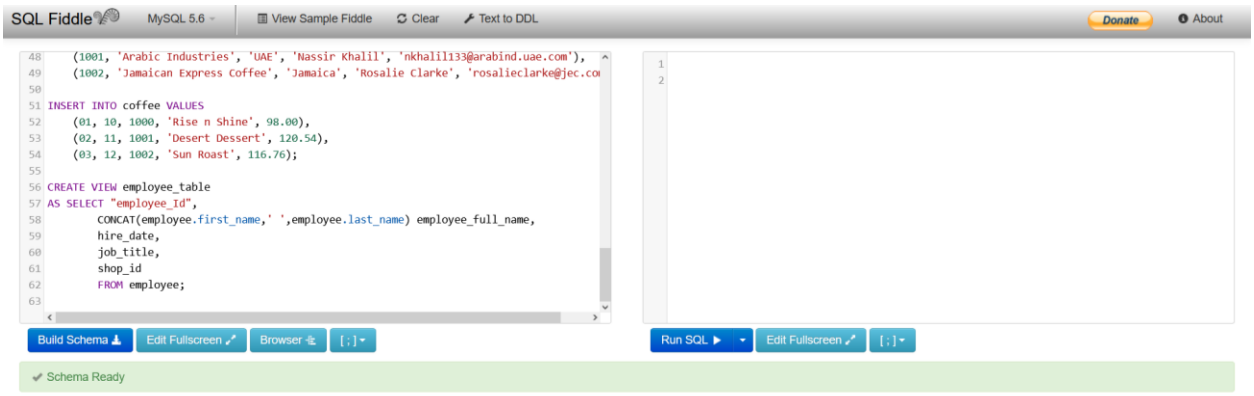

SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
INSERT INTO coffee_shop VALUES
(10, 'Good Mornings', 'Asheville', 'TN'),
(11, 'Juggheads', 'Boone', 'NC'),
(12, 'Pepper Oak Brews', 'Turnersville', 'NJ');
INSERT INTO employee VALUES
(100, 10, 'First', 'Last', '2022-01-01', 'Barista'),
(101, 11,'First1', 'Last1', '2021-09-27', 'Bartender'),
(102, 12, 'First2', 'Last2', '2023-12-21', 'Hostess');
INSERT INTO supplier VALUES
(1000, 'Great Grounds', 'United States', 'Meredith Way', 'merway@greatgrounds.com'),
(1001, 'Arabic Industries', 'UAE', 'Nassir Khalil', 'nkhalil133@arabind.uae.com'),
(1002, 'Jamaican Express Coffee', 'Jamaica', 'Rosalie Clarke', 'rosalieclarke@jec.co
Build Schema
Edit Fullscreen
Browser
[;]
Run SQL
Edit Fullscreen
[;]
Schema Ready


SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
INSERT INTO employee VALUES
(100, 10, 'First', 'Last', '2022-01-01', 'Barista'),
(101, 11,'First1', 'Last1', '2021-09-27', 'Bartender'),
(102, 12, 'First2', 'Last2', '2023-12-21', 'Hostess');
INSERT INTO supplier VALUES
(1000, 'Great Grounds', 'United States', 'Meredith Way', 'merway@greatgrounds.com'),
(1001, 'Arabic Industries', 'UAE', 'Nassir Khalil', 'nkhalil133@arabind.uae.com'),
(1002, 'Jamaican Express Coffee', 'Jamaica', 'Rosalie Clarke', 'rosalieclarke@jec.co
INSERT INTO coffee VALUES
(01, 10, 1000, 'Rise n Shine', 98.00),
(02, 11, 1001, 'Desert Dessert', 120.54),
(03, 12, 1002, 'Sun Roast', 116.76);
Build Schema
Edit Fullscreen
Browser
[;]
Run SQL
Edit Fullscreen
[;]
Schema Ready

B. 3. a.

```
CREATE VIEW employee_table
AS SELECT "employee_Id",
    CONCAT(employee.first_name,' ',employee.last_name) employee_full_name,
    hire_date,
    job_title,
    shop_id
    FROM employee;
```

B. 3. b.



B. 4. a.

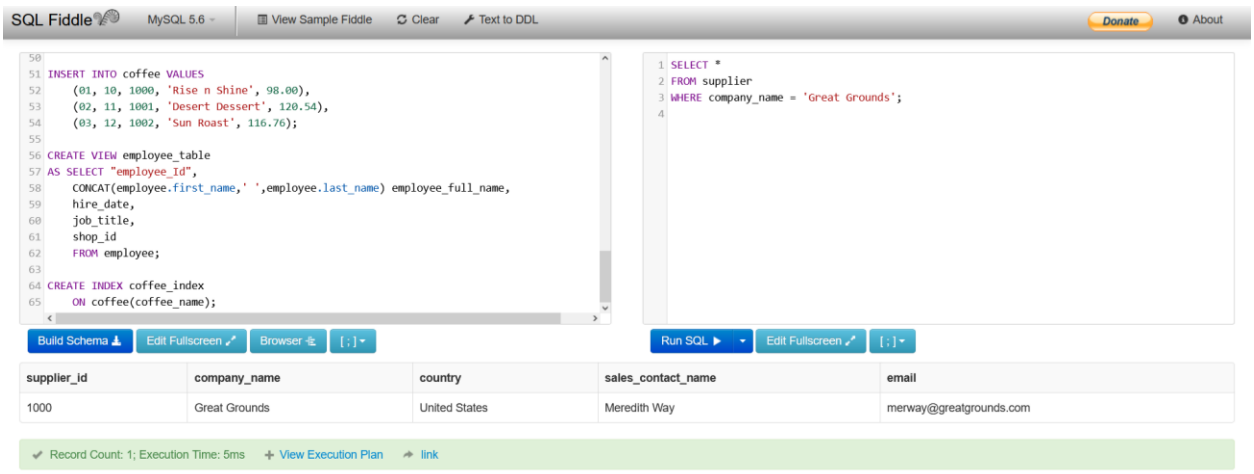
```
CREATE INDEX coffee_index
    ON coffee(coffee_name);
```

B. 4. b.

B. 5. a.

```
SELECT *
    FROM supplier
    WHERE company_name = 'Great Grounds';
```

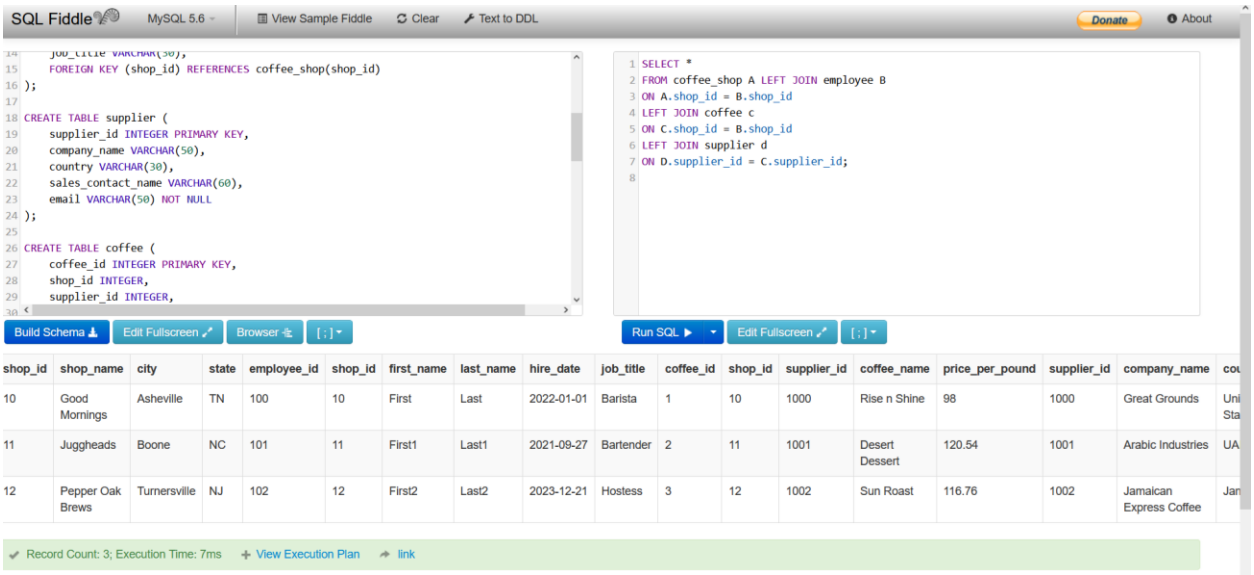
B. 5. b.



B. 6. a.

```
SELECT *
    FROM coffee_shop A LEFT JOIN employee B
    ON A.shop_id = B.shop_id
LEFT JOIN coffee c
    ON C.shop_id = B.shop_id
LEFT JOIN supplier d
    ON D.supplier_id = C.supplier_id;
```

B. 6. b.


SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
job_title VARCHAR(30),
FOREIGN KEY (shop_id) REFERENCES coffee_shop(shop_id)
);
CREATE TABLE supplier (
supplier_id INTEGER PRIMARY KEY,
company_name VARCHAR(50),
country VARCHAR(30),
sales_contact_name VARCHAR(60),
email VARCHAR(50) NOT NULL
);
CREATE TABLE coffee (
coffee_id INTEGER PRIMARY KEY,
shop_id INTEGER,
supplier_id INTEGER,
SELECT *
FROM coffee_shop A LEFT JOIN employee B
ON A.shop_id = B.shop_id
LEFT JOIN coffee c
ON C.shop_id = B.shop_id
LEFT JOIN supplier d
ON D.supplier_id = C.supplier_id;
Build Schema
Edit Fullscreen
Browser
[;]
Run SQL
Edit Fullscreen
[;]
shop_id shop_name city state employee_id shop_id first_name last_name hire_date job_title coffee_id shop_id supplier_id coffee_name price_per_pound supplier_id company_name
10 Good Mornings Asheville TN 100 10 First Last 2022-01-01 Barista 1 10 1000 Rise n Shine 98 1000 Great Grounds
11 Juggheads Boone NC 101 11 First1 Last1 2021-09-27 Bartender 2 11 1001 Desert Dessert 120.54 1001 Arabic Industries
12 Pepper Oak Brews Turnersville NJ 102 12 First2 Last2 2023-12-21 Hostess 3 12 1002 Sun Roast 116.76 1002 Jamaican Express Coffee
Record Count: 3; Execution Time: 7ms
View Execution Plan
link