Part A

A1:

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

- Bagel Name, Bagel Description, and Bagel Price is only dependent on Bagel ID and not Bagel Order ID.
- Bagel Quantity was dependent both on Bagel ID and Bagel Order ID.
- All remaining attributes were dependent on Bagel Order ID and not Bagel ID.
- There can be multiple Bagel Order Line Items in a Bagel Order, but only one Bagel Order can have multiple Bagel Order Line Items.
- There is only one Bagel per Bagel Order Line Item, but multiple Bagel Order Line Items can have one Bagel.

A2

Third Normal Form (3NF)

BAGEL ORDER	
PK	Bagel Order ID
FK	Customer ID
	Order Date
	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

M:1

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

- A separate table was created called Customer with a Primary Key called Customer ID. All the information in the Customer table is now solely dependent on Customer ID and not on Bagel Order ID and can be referenced using Customer ID as a Foreign Key in the Bagel Order table.
- Multiple Bagel Orders can be created by one Customer but only one Customer can create multiple Bagel Orders.

A3

Final Physical Database Model

BAGEL ORDER		
PK	bagel_order_id	INT
FK	customer_id	INT
	order_date	TIMESTAMP
	delivery_fee	INT
	special_notes	VARCHAR(255)

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(255)
	bagel_descript	VARCHAR(255)
	bagel_price	NUMERIC(5,2)

M:1

CUSTOMER		
PK	customer_id	INT
	first_name	VARCHAR(32)
	last_name	VARCHAR(32)
	address_1	VARCHAR(255)
	address_2	VARCHAR(255)
	customer_city	VARCHAR(32)
	customer_state	CHAR(2)
	zip_code	VARCHAR(5)
	mobile_phone	VARCHAR(10)

Part B

B1

```
1  CREATE TABLE employee
2  (
3    employee_id INTEGER PRIMARY KEY,
4    first_name VARCHAR(30),
5    last_name VARCHAR(30),
6    hire_date DATE,
7    job_title VARCHAR(30),
8    shop_id INTEGER
9  );
10
11 CREATE TABLE coffee_shop
12 (
13   shop_id INTEGER PRIMARY KEY,
14   shop_name VARCHAR(50),
```

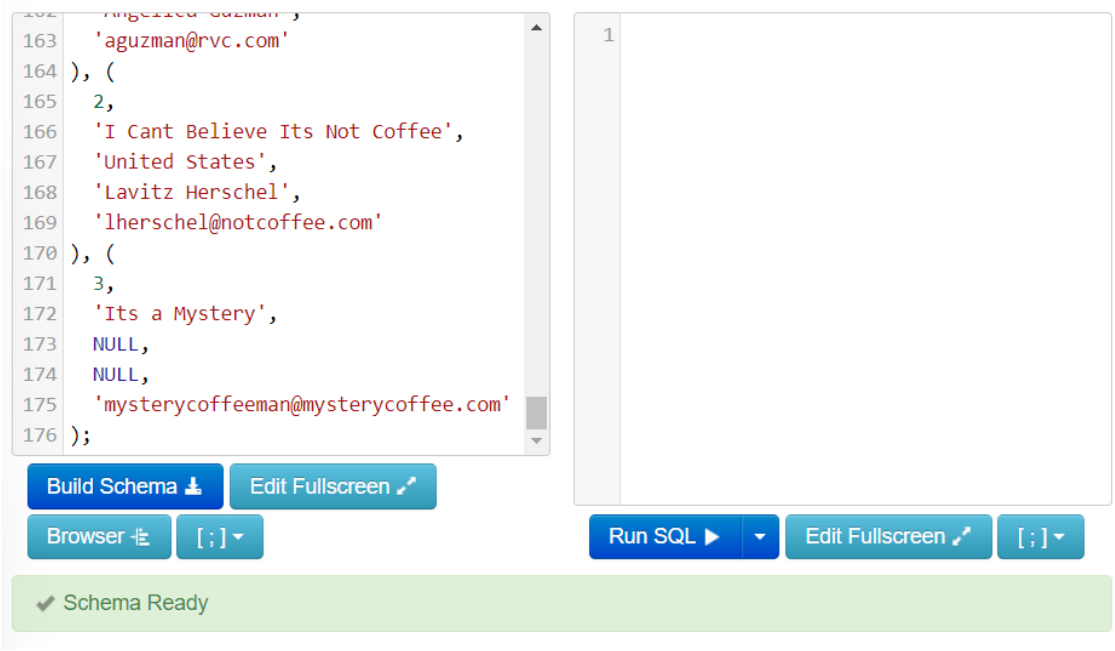
Build Schema | Edit Fullscreen | Browser | [;]

Run SQL | Edit Fullscreen | [;]

Schema Ready

(Full code at bottom of document)

B2



B3

B4

```
174    mysterycoffeeman@mysterycoffee.com
175 );
176
177 SELECT
178 CONCAT(first_name, ' ', last_name) AS e
179 FROM
180 employee;
181
182 CREATE VIEW EmployeeView AS
183 SELECT employee_id, employee_full_name,
184 FROM employee;
185
186 CREATE INDEX idx_coffee_name
187 ON coffee (coffee_name);
```

Build Schema | Edit Fullscreen | Browser | [;]

```
1
```

Run SQL | Edit Fullscreen | [;]

✔ Schema Ready

B5

```
124     hire_date,
125     job_title,
126     shop_id
127 FROM
128     employee;
129 CREATE INDEX idx_coffee_name ON
130     coffee(coffee_name);
131 SELECT
132     supplier_id
133 FROM
134     supplier
135 WHERE
136     sales_contact_name IS NOT NULL;
137
```

Build Schema | Edit Fullscreen | Browser | [;]

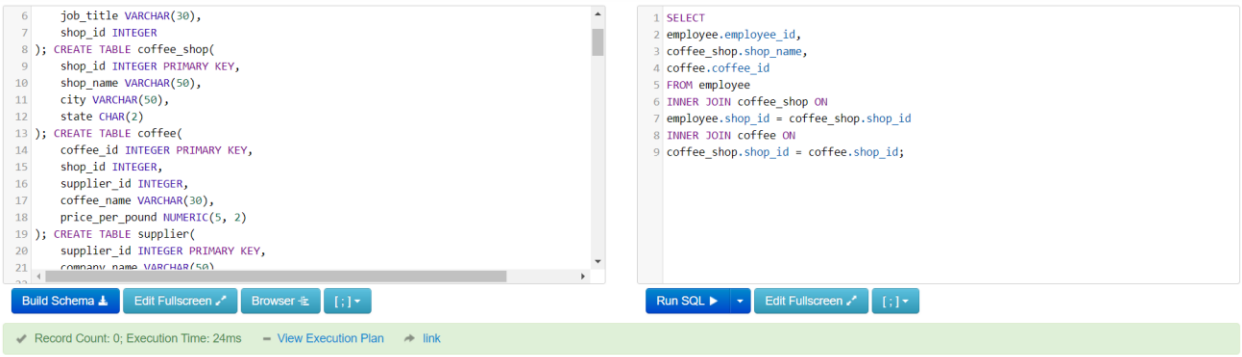
```
1 SELECT last_name
2 FROM employee
3 WHERE first_name = 'John';
```

Run SQL | Edit Fullscreen | [;]

✔ Record Count: 0; Execution Time: 31ms — View Execution Plan — link

id	select_type	table	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employee	ALL					1	100.00	Using where

B6



id	select_type	table	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employee	index	FK_shop_id1	FK_shop_id1	5		1	100.00	Using where; Using index
1	SIMPLE	coffee_shop	eq_ref	PRIMARY	PRIMARY	4	db_9_32d2a0.employee.shop_id	1	100.00	
1	SIMPLE	coffee	ref	FK_shop_id2	FK_shop_id2	5	db_9_32d2a0.employee.shop_id	1	100.00	Using index

Code:

```
CREATE TABLE employee(
    employee_id INTEGER PRIMARY KEY,
    first_name VARCHAR(30),
    last_name VARCHAR(30),
    hire_date DATE,
    job_title VARCHAR(30),
    shop_id INTEGER
); CREATE TABLE coffee_shop(
    shop_id INTEGER PRIMARY KEY,
    shop_name VARCHAR(50),
    city VARCHAR(50),
    state CHAR(2)
); CREATE TABLE coffee(
    coffee_id INTEGER PRIMARY KEY,
    shop_id INTEGER,
    supplier_id INTEGER,
    coffee_name VARCHAR(30),
```

```
    price_per_pound NUMERIC(5, 2)
); 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
); ALTER TABLE
    employee ADD CONSTRAINT FK_shop_id1 FOREIGN KEY(shop_id) REFERENCES coffee_shop(shop_id);
ALTER TABLE
    coffee ADD CONSTRAINT FK_shop_id2 FOREIGN KEY(shop_id) REFERENCES coffee_shop(shop_id);
ALTER TABLE
    coffee ADD CONSTRAINT FK_supplier_id FOREIGN KEY(supplier_id) REFERENCES supplier(supplier_id);
INSERT INTO employee(
    employee_id,
    first_name,
    last_name,
    hire_date,
    job_title,
    shop_id
)
VALUES(
    1,
    'John',
    'Doe',
    '2000-01-01',
    'Manager',
    1
),(
```

```
    2,
    'Jane',
    'Doe',
    '2007-07-07',
    'Barista',
    2
),(
    3,
    'Jimmy',
    'Joe',
    '2012-12-12',
    'Janitor',
    3
);
INSERT INTO coffee_shop(
    shop_id,
    shop_name,
    city,
    state
)
VALUES(
    1,
    'Casa Blanca',
    'New York City',
    'NY'
),(
    2,
    'Casa Amarilla',
    'Amarillo',
```

```
    'TX'
),(
    3,
    'Casa Anaranjada',
    'Los Angeles',
    'CA'
);
INSERT INTO coffee(
    coffee_id,
    shop_id,
    supplier_id,
    coffee_name,
    price_per_pound
)
VALUES(1, 1, 1, 'Vanilla Bean', 7.32),(2, 2, 2, 'Lima Bean', 2.89),(3, 3, 3, 'Mystery Bean', 5.45);
INSERT INTO supplier(
    supplier_id INTEGER,
    company_name VARCHAR(50),
    country VARCHAR(30),
    sales_contact_name VARCHAR(60),
    email VARCHAR(50) NOT NULL
)
VALUES(
    1,
    'Real Vanilla Coffee',
    'Mexico',
    'Angelica Guzman',
    'aguzman@rvc.com'
),(
```

```
    2,
    'I Cant Believe Its Not Coffee',
    'United States',
    'Lavitz Herschel',
    'lherschel@notcoffee.com'
),(
    3,
    'Its a Mystery',
    NULL,
    NULL,
    'mysterycoffeeman@mysterycoffee.com'
);
CREATE VIEW EmployeeView AS SELECT
    employee_id,
    CONCAT(first_name, ' ', last_name) AS employee_full_name,
    hire_date,
    job_title,
    shop_id
FROM
    employee;
CREATE INDEX idx_coffee_name ON
    coffee(coffee_name);
SELECT
    supplier_id
FROM
    supplier
WHERE
    sales_contact_name IS NOT NULL;
```