Nora’s Bagel Bin

Part A

1. **a, b)** 2NF Table:

BAGEL ORDER			BAGEL ORDER LINE ITEM			Bagel	
PK	Bagel Order ID		PK/ FK	Bagel Order ID		PK	Bagel ID
	Order Date	1:M	PK/ FK	Bagel ID	M:1		Bagel Name
	First Name			Bagel Quantity			Bagel Description
	Last Name						Bagel Price
	Address 1						
	Address 2						
	City						
	State						
	Zip						
	Mobile Phone						
	Delivery Fee						
	Special Notes						

1c) Explanation:

<u>Table Attributes</u>

- **Bagel Order Table**
  - *Attributes:*
    - Customer & order properties
  - *Explanation*:
    - Represents customer contact details and order details
- **Bagel Table**
  - *Attributes:*
    - Bagel properties
  - *Explanation:*
    - Represents bagel details

- **Bagel Order Line Item Table**
  - *Attributes:*
    - Primary keys of Bagel-Order and Bagel tables
    - Quantity of each bagel item
  - *Explanation:*
    - Contains the quantity of bagels for each bagel in a Bagel-Order
    - Joins together the Bagel-Order and Bagel tables by using their primary keys as foreign keys & primary keys to ensure uniqueness

Table Cardinalities

Bagel-Orders and Bagels is **Many-to-Many**

- Bagel-Orders can contain many Bagels
- Bagels can be part of many Bagel-Orders

- Bagel-Orders and Bagel-Order Line Item is **One-to-Many**
  - Bagel-Orders can contain many Bagel-Order Line Items
  - Bagel-Order Line item" can only belong to one Bagel Order

- Bagels and "Bagel-Order Line Item" is **One-to-Many**
  - Bagel can be part of many Bagel-Order Line items
  - Bagel-Order Line items can only reference one Bagel

2. **a, b, c, d)** 3NF Table:

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

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

2e) Explanation:

- Table Attributes:
  - **Bagel Order**
    - *Attributes:*
      - Customer primary key & order details
    - *Explanation:*
      - Customer primary key is used as foreign key in order to associate the Bagel-Order item with the Customer who made the order
      - Order attributes describe the relevant order details of the Bagel-Order
  - **Customer**
    - *Attributes:*
      - Customer contact details
    - *Explanation:*
      - Customer table represents the contact details for customers who placed orders
- Table Cardinalities:
  - Bagel-Order and Customer is **Many-to-One**
    - Customer can make many bagel orders
    - Bagel-Order can only refer to one customer

3. Final Physical Database Model:

Final Physical Database Model

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

M:1

CUSTOMER		
PK	customer_id	INT
	first_name	VARCHAR(255)
	last_name	VARCHAR(255)
	address_1	VARCHAR(255)
	address_2	VARCHAR(255)
	city	VARCHAR(50)
	state	CHAR(2)
	zip	CHAR(5)
	mobile_phone	CHAR(11)

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	INT
	bagel_name	VARCHAR(255)
	bagel_description	VARCHAR(255)
	bagel_price	NUMERIC

Jaunty Coffee Co

Part B

1. Create Table Statements
    a. SQL Code:

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

CREATE TABLE supplier (
 supplier_id INT AUTO_INCREMENT PRIMARY KEY,
 company_name VARCHAR(50),
 country VARCHAR(30),
 sales_contact_name VARCHAR(60),
 email VARCHAR(50)
);

CREATE TABLE coffee (
 coffee_id INT AUTO_INCREMENT PRIMARY KEY,
 shop_id INT,
 supplier_id INT,
 coffee_name VARCHAR(30),
 price_per_pound DECIMAL(5,2),
 FOREIGN KEY (shop_id)
   REFERENCES coffee_shop (shop_id)
     ON DELETE CASCADE
     ON UPDATE CASCADE,
 FOREIGN KEY (supplier_id)
   REFERENCES supplier (supplier_id)
     ON DELETE CASCADE
     ON UPDATE CASCADE
);
```

```
CREATE TABLE employee (
 employee_id INT AUTO_INCREMENT PRIMARY KEY,
 first_name VARCHAR(30),
 last_name VARCHAR(30),
 hire_date DATE,
 job_title VARCHAR(30) DEFAULT 'Employee',
 shop_id INT,
 FOREIGN KEY (shop_id)
   REFERENCES coffee_shop (shop_id)
     ON DELETE CASCADE
     ON UPDATE CASCADE
);
```

b. Create Table Response:

```
MySQL CREATE TABLE coffee_shop (
   ->   shop_id INT AUTO_INCREMENT PRIMARY KEY,
   ->   shop_name VARCHAR(50),
   ->   city VARCHAR(50),
   ->   state CHAR(2)
   -> );
Query OK, 0 rows affected
Time: 0.099s
MySQL CREATE TABLE supplier (
   ->   supplier_id INT AUTO_INCREMENT PRIMARY KEY,
   ->   company_name VARCHAR(50),
   ->   country VARCHAR(30),
   ->   sales_contact_name VARCHAR(60),
   ->   email VARCHAR(50)
   -> );
Query OK, 0 rows affected
Time: 0.082s
MySQL CREATE TABLE coffee (
   ->   coffee_id INT AUTO_INCREMENT PRIMARY KEY,
   ->   shop_id INT,
   ->   supplier_id INT,
   ->   coffee_name VARCHAR(30),
   ->   price_per_pound DECIMAL(5,2),
   ->   FOREIGN KEY (shop_id)
   ->     REFERENCES coffee_shop (shop_id)
   ->       ON DELETE CASCADE
   ->       ON UPDATE CASCADE,
   ->   FOREIGN KEY (supplier_id)
   ->     REFERENCES supplier (supplier_id)
   ->       ON DELETE CASCADE
   ->       ON UPDATE CASCADE
   -> );
Query OK, 0 rows affected
Time: 0.142s
MySQL CREATE TABLE employee (
   ->   employee_id INT AUTO_INCREMENT PRIMARY KEY,
   ->   first_name VARCHAR(30),
   ->   last_name VARCHAR(30),
   ->   hire_date DATE,
   ->   job_title VARCHAR(30) DEFAULT 'Employee',
   ->   shop_id INT,
   ->   FOREIGN KEY (shop_id)
   ->     REFERENCES coffee_shop (shop_id)
   ->       ON DELETE CASCADE
   ->       ON UPDATE CASCADE
   -> );
Query OK, 0 rows affected
Time: 0.119s
```

2. Table Population

a. SQL Code:

```
INSERT INTO coffee_shop
 (shop_name, city, state)
VALUES
 ('Krusty Krabs', 'Bikini Bottom', 'HI'),
 ('Central Perk', 'Manhattan', 'NY'),
 ('The Three Broomsticks', 'Hogsmeade', 'WA');

INSERT INTO supplier
 (company_name, country, sales_contact_name, email)
VALUES
 ('Westrock Coffee Company', 'USA', 'Jason Munez', 'jasonm@yahoo.com'),
 ('King Coffee Supplier', 'Brazil', 'Bob Saget', 'bobs@yahoo.com'),
 ('Zap Coffee Inc', 'Colombia', 'Bill Burr', 'billb@yahoo.com');

INSERT INTO coffee
 (shop_id, supplier_id, coffee_name, price_per_pound)
VALUES
 (
   (
     SELECT shop_id
     FROM coffee_shop
     WHERE shop_name = 'Krusty Krabs'
   ),
   (
     SELECT supplier_id
     FROM supplier
     WHERE company_name = 'King Coffee Supplier'
   ),
   'Brazilian Santos',
   00044.45
 ),
 (
   (
     SELECT shop_idRR
     FROM coffee_shop
     WHERE shop_name = 'Central Perk'
   ),
```

```
  (
    SELECT supplier_id
    FROM supplier
    WHERE company_name = 'Zap Coffee Inc'
  ),
  'Colombian Supremo',
  00059.10
),
(
  (
    SELECT shop_id
    FROM coffee_shop
    WHERE shop_name = 'The Three Broomsticks'
  ),
  (
    SELECT supplier_id
    FROM supplier
    WHERE company_name = 'Westrock Coffee Company'
  ),
  'Organic Mexican',
  00038.70
);

INSERT INTO employee
 (first_name, last_name, hire_date, job_title, shop_id)
VALUES
 (
  'Spongebob', 'Squarepants', '21/12/01', 'Coffee Barista',
  (
    SELECT shop_id
    FROM coffee_shop
    WHERE shop_name = 'Krusty Krabs'
  )
 ),
 (
  'Rachel', 'Green', '07/05/22', 'Manager',
  (
    SELECT shop_id
    FROM coffee_shop
    WHERE shop_name = 'Central Perk'
  )
 ),
```

```
(
  'Ronald', 'Weasely', '11/03/14', 'Coffee Barista',
  (
    SELECT shop_id
    FROM coffee_shop
    WHERE shop_name = 'The Three Broomsticks'
  )
);
```

b. INSERT statement response:

```
MySQL INSERT INTO coffee_shop
   ->   (shop_name, city, state)
   -> VALUES
   ->   ('Krusty Krabs', 'Bikini Bottom', 'HI'),
   ->   ('Central Perk', 'Manhattan', 'NY'),
   ->   ('The Three Broomsticks', 'Hogsmeade', 'WA');
Query OK, 3 rows affected
Time: 0.107s
MySQL INSERT INTO supplier
   ->   (company_name, country, sales_contact_name, email)
   -> VALUES
   ->   ('Westrock Coffee Company', 'USA', 'Jason Munez', 'jasonm@yahoo.com'),
   ->   ('King Coffee Supplier', 'Brazil', 'Bob Saget', 'bobs@yahoo.com'),
   ->   ('Zap Coffee Inc', 'Colombia', 'Bill Burr', 'billb@yahoo.com');
Query OK, 3 rows affected
Time: 0.018s
MySQL INSERT INTO coffee
   ->   (shop_id, supplier_id, coffee_name, price_per_pound)
   -> VALUES
   ->   (
   ->     (
   ->       SELECT shop_id
   ->       FROM coffee_shop
   ->       WHERE shop_name = 'Krusty Krabs'
   ->     ),
   ->     (
   ->       SELECT supplier_id
   ->       FROM supplier
   ->       WHERE company_name = 'King Coffee Supplier'
   ->     ),
   ->     'Brazilian Santos',
   ->     00044.45
   ->   ),
   ->   (
   ->     (
   ->       SELECT shop_id
   ->       FROM coffee_shop
   ->       WHERE shop_name = 'Central Perk'
   ->     ),
   ->     (
   ->       SELECT supplier_id
   ->       FROM supplier
   ->       WHERE company_name = 'Zap Coffee Inc'
   ->     ),
   ->     'Colombian Supremo',
   ->     00059.10
   ->   ),
   ->   (
   ->     (
   ->       SELECT shop_id
   ->       FROM coffee_shop
   ->       WHERE shop_name = 'The Three Broomsticks'
   ->     ),
   ->     (
   ->       SELECT supplier_id
   ->       FROM supplier
   ->       WHERE company_name = 'Westrock Coffee Company'
   ->     ),
   ->     'Organic Mexican',
   ->     00038.70
   ->   );
Query OK, 3 rows affected
Time: 0.036s
MySQL INSERT INTO employee
   ->   (first_name, last_name, hire_date, job_title, shop_id)
   -> VALUES
   ->   (
   ->     'Spongebob', 'Squarepants', '21/12/01', 'Coffee Barista',
   ->     (
   ->       SELECT shop_id
   ->       FROM coffee_shop
   ->       WHERE shop_name = 'Krusty Krabs'
   ->     )
   ->   ),
   ->   (
   ->     'Rachel', 'Green', '07/05/22', 'Manager',
   ->     (
   ->       SELECT shop_id
   ->       FROM coffee_shop
   ->       WHERE shop_name = 'Central Perk'
   ->     )
   ->   ),
   ->   (
   ->     'Ronald', 'Weasely', '11/03/14', 'Coffee Barista',
   ->     (
   ->       SELECT shop_id
   ->       FROM coffee_shop
   ->       WHERE shop_name = 'The Three Broomsticks'
   ->     )
   ->   );
Query OK, 3 rows affected
Time: 0.025s
```

3. Create Views

a. SQL Code:

```
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 e;
```

b. View statement response:

```
mysql> 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 e;
Query OK, 0 rows affected (0.03 sec)

mysql> SELECT * FROM employee_view;
+-------------+----------------------+------------+---------------+---------+
| employee_id | employee_full_name   | hire_date  | job_title     | shop_id |
+-------------+----------------------+------------+---------------+---------+
|           1 | Spongebob Squarepants | 2021-12-01 | Coffee Barista |       1 |
|           2 | Rachel Green         | 2007-05-22 | Manager       |       2 |
|           3 | Ronald Weasely       | 2011-03-14 | Coffee Barista |       3 |
+-------------+----------------------+------------+---------------+---------+
```

4. Create Index

a. SQL Code:

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

b. Create index response:

```
MySQL CREATE INDEX coffee_name
   -> ON coffee(coffee_name);
Query OK, 0 rows affected
Time: 0.160s
MySQL SHOW INDEXES
   -> FROM coffee;
+--------+------------+-------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+
| Table  | Non_unique | Key_name    | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type |
+--------+------------+-------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+
| coffee | 0          | PRIMARY     | 1            | coffee_id   | A         | 2           | <null>   | <null> |      | BTREE      |
| coffee | 1          | shop_id     | 1            | shop_id     | A         | 2           | <null>   | <null> | YES  | BTREE      |
| coffee | 1          | supplier_id | 1            | supplier_id | A         | 2           | <null>   | <null> | YES  | BTREE      |
| coffee | 1          | coffee_name | 1            | coffee_name | A         | 3           | <null>   | <null> | YES  | BTREE      |
+--------+------------+-------------+--------------+-------------+-----------+-------------+----------+--------+------+------------+

4 rows in set
Time: 0.068s
```

5. Create SFW (SELECT-FROM-WHERE) Query

a. SQL Code:

```
SELECT *
FROM employee
WHERE hire_date > '10-01-01';
```

b. Select query response:

```
MySQL SELECT *
   -> FROM employee
   -> WHERE hire_date > '10-01-01';
+-------------+------------+-------------+------------+----------------+---------+
| employee_id | first_name | last_name   | hire_date  | job_title      | shop_id |
+-------------+------------+-------------+------------+----------------+---------+
| 1           | Spongebob  | Squarepants | 2021-12-01 | Coffee Barista | 1       |
| 3           | Ronald     | Weasely     | 2011-03-14 | Coffee Barista | 3       |
+-------------+------------+-------------+------------+----------------+---------+
2 rows in set
Time: 0.035s
```

6. Create JOIN query
   a. SQL code:

```
SELECT *
FROM coffee c
 INNER JOIN supplier s
   ON c.supplier_id = s.supplier_id
 INNER JOIN coffee_shop cs
   ON c.shop_id = cs.shop_id;
```

   b. Join query response:

```
MySQL SELECT *
   -> FROM coffee c
   ->   INNER JOIN supplier s
   ->     ON c.supplier_id = s.supplier_id
   ->   INNER JOIN coffee_shop cs
   ->     ON c.shop_id = cs.shop_id;
+-----------+---------+-------------+----------------+----------------+-------------+----------------------+---------+--------------------+-----------------+---------+-----------------------+-------------+-------+
| coffee_id | shop_id | supplier_id | coffee_name    | price_per_pound | supplier_id | company_name         | country | sales_contact_name | email           | shop_id | shop_name             | city        | state |
+-----------+---------+-------------+----------------+----------------+-------------+----------------------+---------+--------------------+-----------------+---------+-----------------------+-------------+-------+
| 3         | 3       | 1           | Organic Mexican | 38.70          | 1           | Westrock Coffee Company | USA     | Jason Munez        | jasonm@yahoo.com | 3       | The Three Broomsticks | Hogsmeade   | WA    |
| 1         | 1       | 2           | Brazilian Santos | 44.45          | 2           | King Coffee Supplier | Brazil  | Bob Saget          | bobs@yahoo.com  | 1       | Krusty Krabs          | Bikini Bottom | HI    |
| 2         | 2       | 3           | Colombian Supremo | 59.10          | 3           | Zap Coffee Inc       | Colombia | Bill Burr          | billb@yahoo.com | 2       | Central Perk          | Manhattan   | NY    |
+-----------+---------+-------------+----------------+----------------+-------------+----------------------+---------+--------------------+-----------------+---------+-----------------------+-------------+-------+

3 rows in set
Time: 0.010s
```