Robert Gal

PART A: Nora's bagel bin database blueprint

First normal form, 1NF:

BAGEL ORDER	
PK	Bagel_Order_ID
PK	Bagel_ID
	Order_Date
	First_Name
	Last_Name
	Address 1
	Address 2
	City
	State
	Zip
	Mobile_Phone
	Delivery_Fee
	Bagel_Name
	Bagel_Description
	Bagel_Price
	Bagel_Quantity
	Special_Notes

This is the original table in first normal form, 1NF.

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

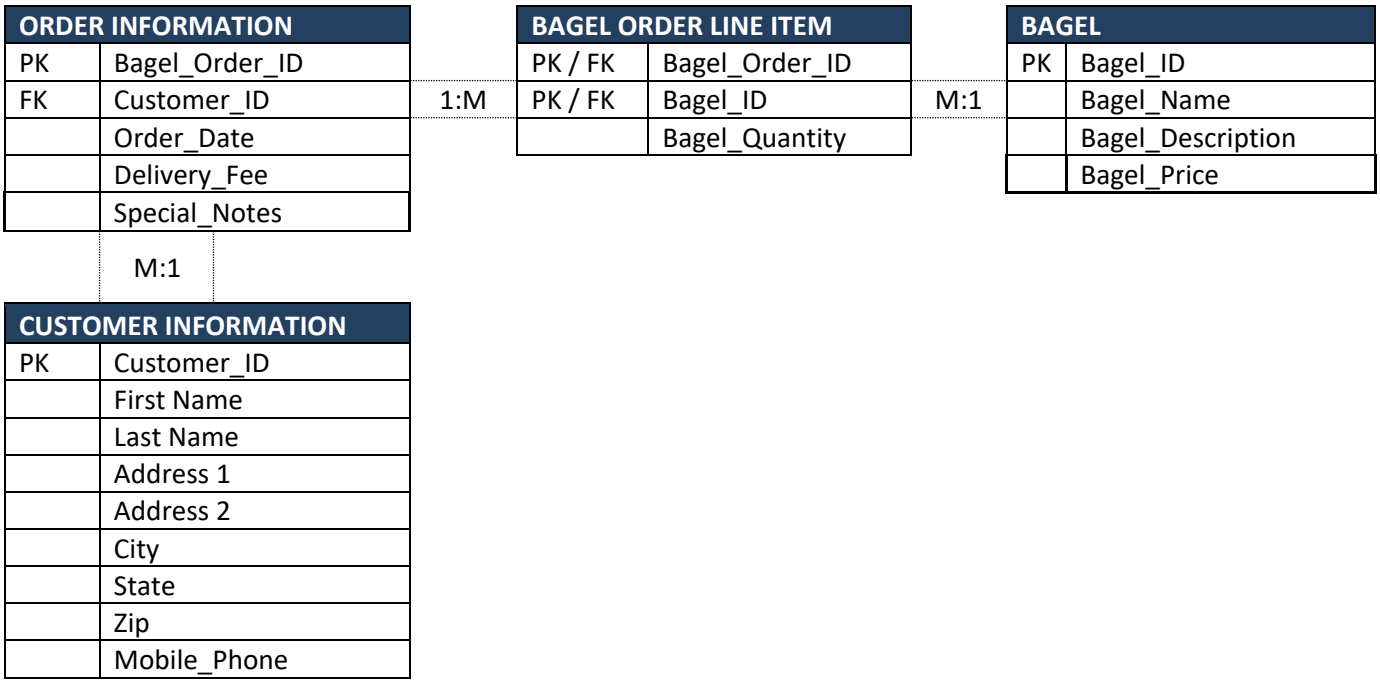
M:1

BAGEL	
PK	Bagel_ID
	Bagel_name
	Bagel_description
	Bagel_price

This is the second normal form, 2NF. We have to split the original 1NF to qualify for the 2NF by creating two new tables “Bagel Order Line Item” and “Bagel”. This will remove partial dependencies as all attributes must to depend on their primary keys.
“Bagel Order” connects to “Bagel Order Line Item” in one to many relationship, one order can have many items but items only can have one order. “Bagel Order Line Item” connects to the “Bagel” table with a many to one relationship. Order items can have only one bagel, and one bagel can have many order items.

“Bagel order” table has a primary key “Bagel_Order_ID” and the other necessary attributes included in the table. “Bagel Order Line Item” table has 3 columns with 2 primary keys “Bagel_Order_ID” and “Bagel_ID”. They are foreign keys (composite keys) as well pointing to the other two tables. We have “Quantity” attribute which is dependent on the primary keys. The “Bagel” table has a primary key(Bagel_id) which is the foreign key in the “Bagel Order Line Item” table. All other attributes are dependent on the bagel_id primary key.

Third normal form, 3NF:



This is the third normal form, 3NF. In order to qualify for 3NF we need to create a new table “customer information”. The customer information attributes did not relate to the Bagel_Order_ID, and every non key attributes need to be defined by the primary key. The first 3 tables cardinality did not change. The new “customer information” table has a many to one relationship. Orders only have one customer, while one customer can have many orders.

The new “customer information” table has a primary key(customer_id). We have to create a foreign key in the “order information” table in order to connect the two tables. All other attributes in the new table dependent on the primary key “customer_ID”.

Physical database:

ORDER INFORMATION		
PK	bagel_order_id	INT
FK	customer_id	INT
	order_date	TIMESTAMP
	delivery_fee	NUMERIC(2,2)
	special_notes	VARCHAR(90)

1:M

BAGEL ORDER LINE ITEM		
PK / FK	bagel_order_id	INT
PK / FK	bagel_id	INT
	bagel_quantity	INT

M:1

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

M:1

CUSTOMER INFORMATION		
PK	customer_id	INT
	first_name	VARCHAR(30)
	last_name	VARCHAR(30)
	address_1	VARCHAR(90)
	address_2	VARCHAR(90)
	City	VARCHAR(30)
	State	CHAR(2)
	Zip	CHAR(5)
	mobile_phone	CHAR(10)

This is the final database with all appropriate tables, columns and their cardinalities.

PART B: Jaunty Coffee Co.ERD

1.Develop SQL code to create each table as specified:

```
3  • CREATE TABLE employee(
4        employee_id INT PRIMARY KEY,
5        first_name VARCHAR(30),
6        last_name VARCHAR(30),
7        hire_date DATE,
8        job_title VARCHAR(30),
9        shop_id int
10   );
```

Output

Action Output

#	Time	Action	Message
1	23:06:18	CREATE TABLE employee(employee_id INT PRIMARY ...	0 row(s) affected

```
11
12 • CREATE TABLE coffee_shop(
13       shop_id INT PRIMARY KEY,
14       shop_name VARCHAR(50),
15       city VARCHAR(50),
16       state CHAR(2)
17   );
18
19 • ALTER TABLE employee
20   ADD FOREIGN KEY(shop_id) REFERENCES coffee_shop(shop_id) ON DELETE SET NULL;
```

Output

Action Output

#	Time	Action	Message
1	23:06:18	CREATE TABLE employee(employee_id INT PRIMARY ...	0 row(s) affected
2	23:08:09	CREATE TABLE coffee_shop(shop_id INT PRIMARY K...	0 row(s) affected
3	23:08:13	ALTER TABLE employee ADD FOREIGN KEY(shop_id) ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

```sql
CREATE TABLE coffee(
    coffee_id INT PRIMARY KEY,
    shop_id INT,
    supplier_id INT,
    coffee_name VARCHAR(30),
    price_per_pound NUMERIC(5, 2)
);
```

Output

Action Output

#	Time	Action	Message	Duration /
1	23:06:18	CREATE TABLE employee(employee_id INT PRIMARY ...	0 row(s) affected	0.016 sec
2	23:08:09	CREATE TABLE coffee_shop(shop_id INT PRIMARY K...	0 row(s) affected	0.015 sec
3	23:08:13	ALTER TABLE employee ADD FOREIGN KEY(shop_id) ...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
4	23:10:00	CREATE TABLE coffee(coffee_id INT PRIMARY KEY, ...	0 row(s) affected	0.016 sec

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

ALTER TABLE coffee
    ADD FOREIGN KEY(shop_id) REFERENCES coffee_shop(shop_id) ON DELETE SET NULL,
    ADD FOREIGN KEY(supplier_id) REFERENCES supplier(supplier_id) ON DELETE SET NULL;
```

Output

Action Output

#	Time	Action	Message
1	23:11:20	CREATE TABLE supplier(supplier_id INT PRIMARY KE...	0 row(s) affected
2	23:11:23	ALTER TABLE coffee ADD FOREIGN KEY(shop_id) RE...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

2. Develop SQL code to populate each table in the database design:

```
43 •  INSERT INTO coffee_shop
44    VALUES  (1, 'starbucks', 'Seattle', 'WA'),
45            (2, 'mycafe', 'Phoenix', 'AZ'),
46            (3, 'Peets', 'Los Angeles','CA');
47
48 •  INSERT INTO supplier
49    VALUES  (1, 'sip comp', 'USA', 'Bob','bob@freecafenow.com'),
50            (2, 'coco' , 'Japan' , 'Omohara','omohara@jonapot.com'),
51            (3, 'pepe CO', 'Austria' ,'Lokoto', 'lokoto@lokoto.com');
52
53 •  INSERT INTO employee
54    VALUES  (1, 'John', 'Conor', '1988-01-01', 'cleaner', 1),
55            (2, 'Sarah', 'Conor', '1988-01-02','owner', 2),
56            (3, 'Arnold', 'Machine', '1855-01-03', 'CEO', 3);
57
58 •  INSERT INTO coffee
59    VALUES  (1, 1, 1, 'light_roast' , 19.99),
60            (2, 2, 2, 'medium_roast', 19.89),
61            (3, 3, 3, 'Vanilla_roast', 21.99);
62
```

Output

Action Output

#	Time	Action
1	23:16:09	INSERT INTO coffee_shop VALUES(1, 'starbucks', 'Seattle', 'WA'), (2, 'mycafe', 'Phoenix', 'AZ'), (
2	23:16:13	INSERT INTO supplier VALUES(1, 'sip comp', 'USA', 'Bob','bob@freecafenow.com'), (2, 'coco' , 'Japar
3	23:16:16	INSERT INTO employee VALUES(1, 'John', 'Conor', '1988-01-01', 'cleaner', 1), (2, 'Sarah', 'Conor', '198
4	23:16:18	INSERT INTO coffee VALUES(1, 1, 1, 'light_roast' , 19.99), (2, 2, 2, 'medium_roast', 19.89), (3, 3,

3.Develop SQL code to create View:

```
64 •  CREATE VIEW my_employees AS
65    SELECT
66        employee_id,
67        CONCAT(first_name,' ', last_name) AS employee_full_name,
68        hire_date,
69        job_title,
70        shop_id
71    FROM employee;
72
73 •  SELECT * FROM my_employees;
74
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

employee_id	employee_full_name	hire_date	job_title	shop_id
1	John Conor	1988-01-01	cleaner	1
2	Sarah Conor	1988-01-02	owner	2
3	Arnold Machine	1855-01-03	CEO	3

_employees 1 ×

Output

Action Output

#	Time	Action
1	23:17:41	CREATE VIEW my_employees AS SELECT employee_id, CONCAT(first_name,' ', last
2	23:20:37	SELECT * FROM my_employees LIMIT 0, 1000

4.Develop SQL code to create index:

```
75
76 •  CREATE INDEX index_coffee_name
77        ON coffee(coffee_name);
78
```

Output

Action Output

#	Time	Action
1	23:22:50	CREATE INDEX index_coffee_name ON coffee(coffee_name)

5.Develop SQL code to create an SFW:

```
88 •  SELECT *
89    FROM employee
90    WHERE last_name = 'Conor';
91
```

Result Grid | Filter Rows: | Edit: | Export/Import:

employee_id	first_name	last_name	hire_date	job_title	shop_id
1	John	Conor	1988-01-01	cleaner	1
2	Sarah	Conor	1988-01-02	owner	2
NULL	NULL	NULL	NULL	NULL	NULL

Output

Action Output

#	Time	Action
1	23:22:50	CREATE INDEX index_coffee_name ON coffee(coffee_name)
2	23:23:59	SELECT * FROM employee WHERE last_name = 'Conor' LIMIT 0, 1000

6.Develop SQL code to create a query:

```
100 •  SELECT
101        CS.shop_id,
102        CS.shop_name,
103        C.coffee_id,
104        C.coffee_name,
105        S.supplier_id,
106        S.company_name
107    FROM coffee_shop CS
108    JOIN coffee C
109    ON CS.shop_id = C.shop_id
110    JOIN supplier S
111    ON C.supplier_id = S.supplier_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

shop_id	shop_name	coffee_id	coffee_name	supplier_id	company_name
1	starbucks	1	light_roast	1	sip comp
2	mycafe	2	medium_roast	2	coco
3	Peets	3	Vanilla_roast	3	pepe CO

Result 9

Output

Action Output

#	Time	Action	Message
1	23:40:02	SELECT CS.shop_id, CS.shop_name, C.coffee_id, C...	3 row(s) returned

Another SQL code with Join:

```
100 •  SELECT
101        shop_id,
102        shop_name,
103        coffee_id,
104        supplier_id
105    FROM coffee_shop
106    JOIN coffee
107    USING (shop_id)
108    JOIN supplier
109    USING (supplier_id);
110
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

shop_id	shop_name	coffee_id	supplier_id
1	starbucks	1	1
2	mycafe	2	2
3	Peets	3	3

Result 3

Output

Action Output

#	Time	Action
1	23:25:25	SELECT shop_id, shop_name, coffee_id, supplier_id FROM coffee_sh