Zachary Durso

C170 – Data Management Applications

4-22-2022

VHT2

A.1

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

A.1.c – 2NF Explanation

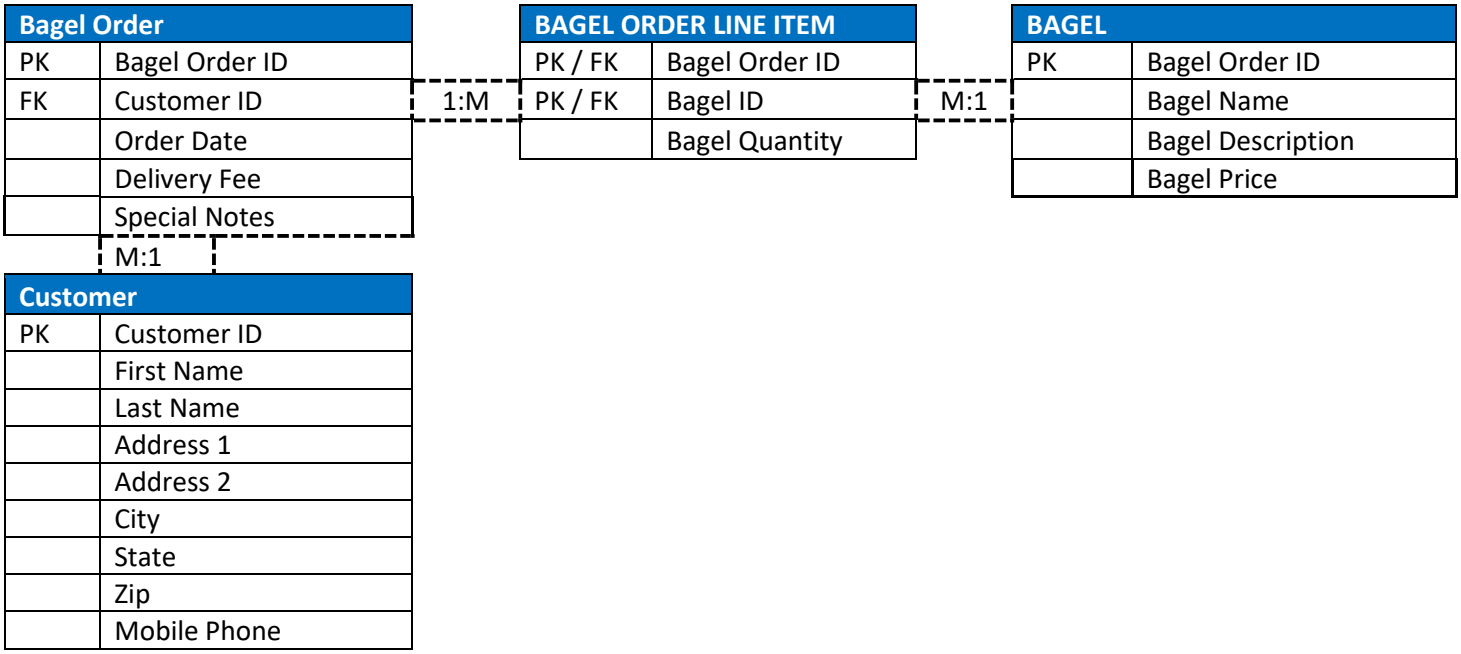
I placed Bagel Name, Description, and Price into the Bagel table because they're all unique and dependent on the individual bagel, described by the Bagel ID, which is the primary key for the table as it's simple, stable, and relatively meaningless if it's not corresponding to these other data points.

The Bagel Order Line Item table has a M:1 relationship with the Bagel table because while there may be many line items, each bagel order line item can only correspond to one particular bagel. This table includes two keys, and one other data item that doesn't really fit anywhere else, Bagel Quantity. This is the best place for Bagel Quantity because it is unique, not necessarily to the bagel, but to the Bagel Order Line Item which denotes, the intended bagel.

In the Bagel Order table, we have everything else, the information is all specific to the order and this is the only logical place to put the remaining data from the order form. This table has a 1:M relationship with Bagel Order Line Item, this is because one order may have many line items.

A.2

Third Normal Form (3NF)



A.2.e – 3NF Explanation

Here we have split the Bagel Order table into two parts, one still named Bagel Order, and one named Customer. This makes sense because usually, you'll have an account with your personal information already saved to a profile. This way when you place an order, you don't need to fill out all your personal details each time. A new key needed to be created though, to link the customer table with the order table. This new key is named Customer ID. This is the best option because it's relatively useless otherwise and should never change.

Bagel Order is now a concise table with only information relevant to the bagel order within it. It also includes, as a foreign key, the Customer ID key. This way, we can link the two tables together. The Bagel Order table has a many-to-one relationship with the Customer table because many orders may be placed by one customer. The bagel Order table also has a 1:M relationship with the Bagel Order Line Item table because one order may contain many line items.

The Bagel Order Line Item table has a M:1 relationship with the Bagel table because while there may be many line items, each bagel order line item can only correspond to one particular bagel. This table includes two keys, and one other data item that doesn't really fit anywhere else, Bagel Quantity. This is the best place for Bagel Quantity because it is unique, not necessarily to the bagel, but to the Bagel Order Line Item which denotes, the intended bagel. The data within each cell of the other two tables, Bagel Order Line Item and Bagel, are all exactly the same as they were in the 2NF diagram. These two tables remain unchanged.

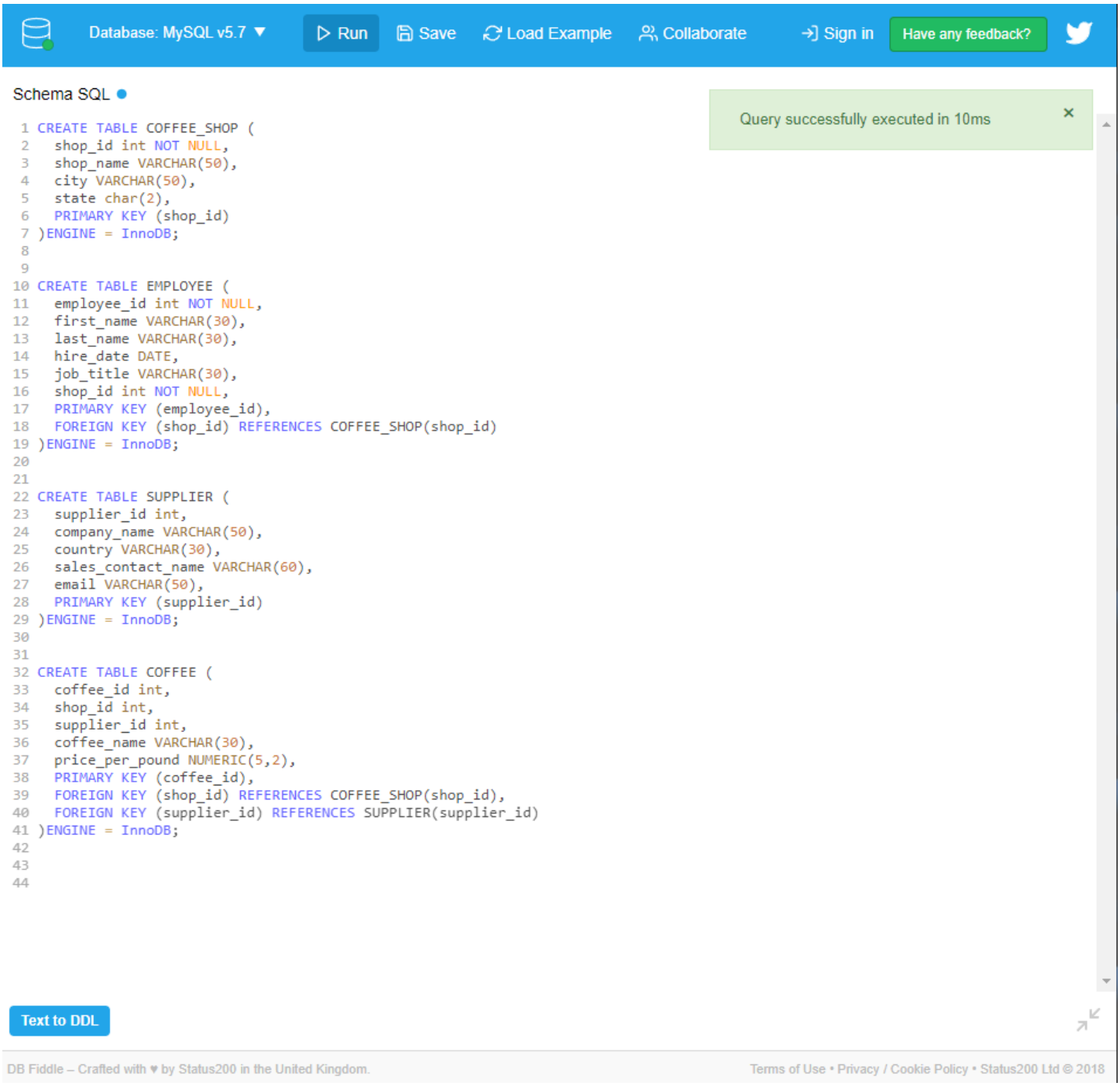
A.3

Final Physical Database Model

Bagel Order				BAGEL ORDER LINE ITEM				BAGEL		
PK	bagel_order_id	INT		PK / FK	bagel_order_id	INT		PK	bagel_id	CHAR(2)
FK	customer_id	INT	1:M	PK / FK	bagel_id	CHAR(2)	M:1		bagel_name	VARCHAR(45)
	order_date	TIMESTAMP			quantity	INT			bagel_description	VARCHAR(80)
	delivery_fee	NUMERIC(5,2)							bagel_price	NUMERIC(5,2)
	special_notes	VARCHAR(2500)								
	M:1									

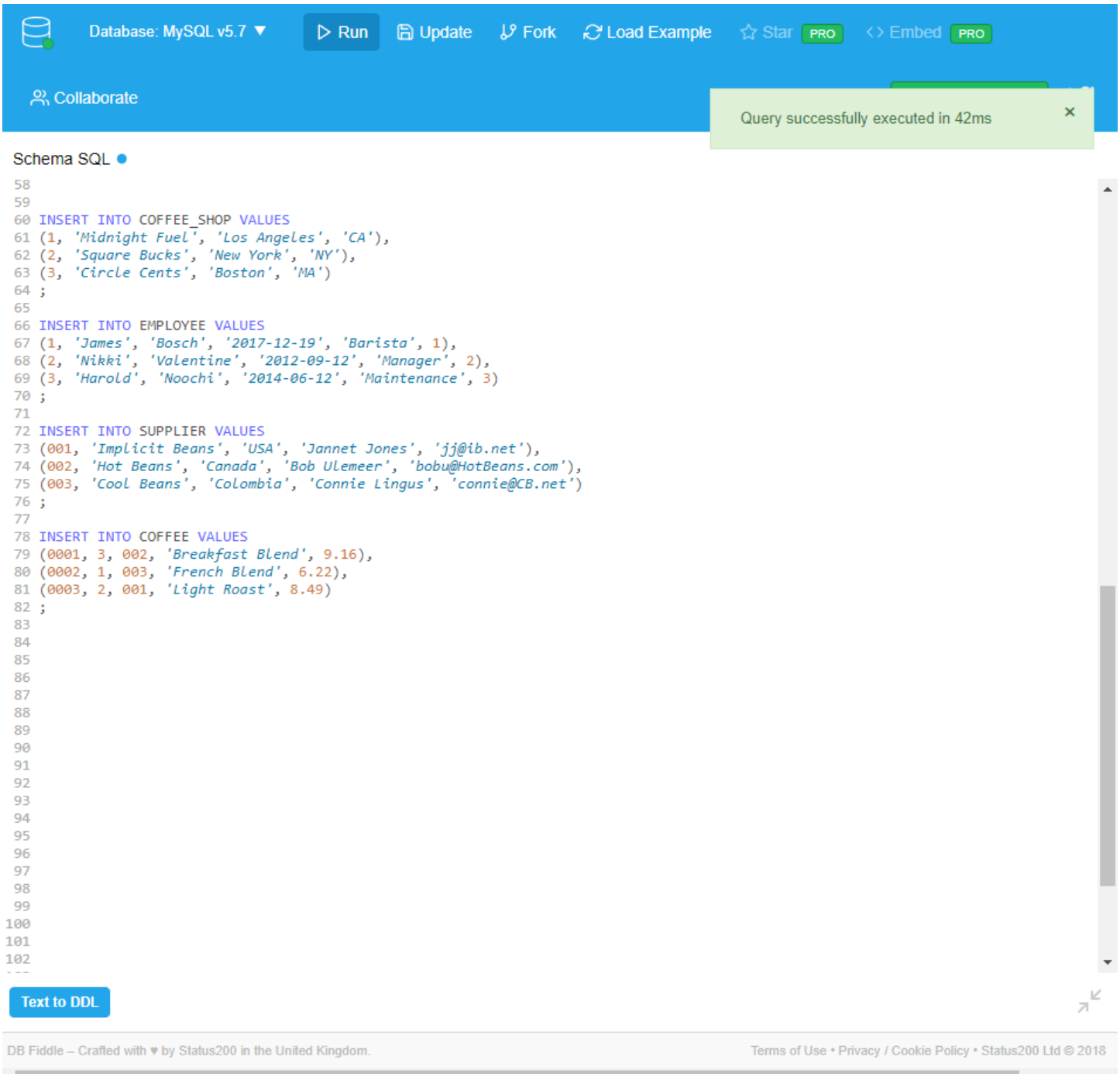
Customer		
PK	customer_id	INT
	first_name	VARCHAR(20)
	last_name	VARCHAR(20)
	address_1	VARCHAR(65)
	address_2	VARCHAR(65)
	city	VARCHAR(20)
	state	VARCHAR(20)
	zip	NUMERIC(5,0)
	mobile_phone	NUMERIC(10,0)

B.1 – Create Tables



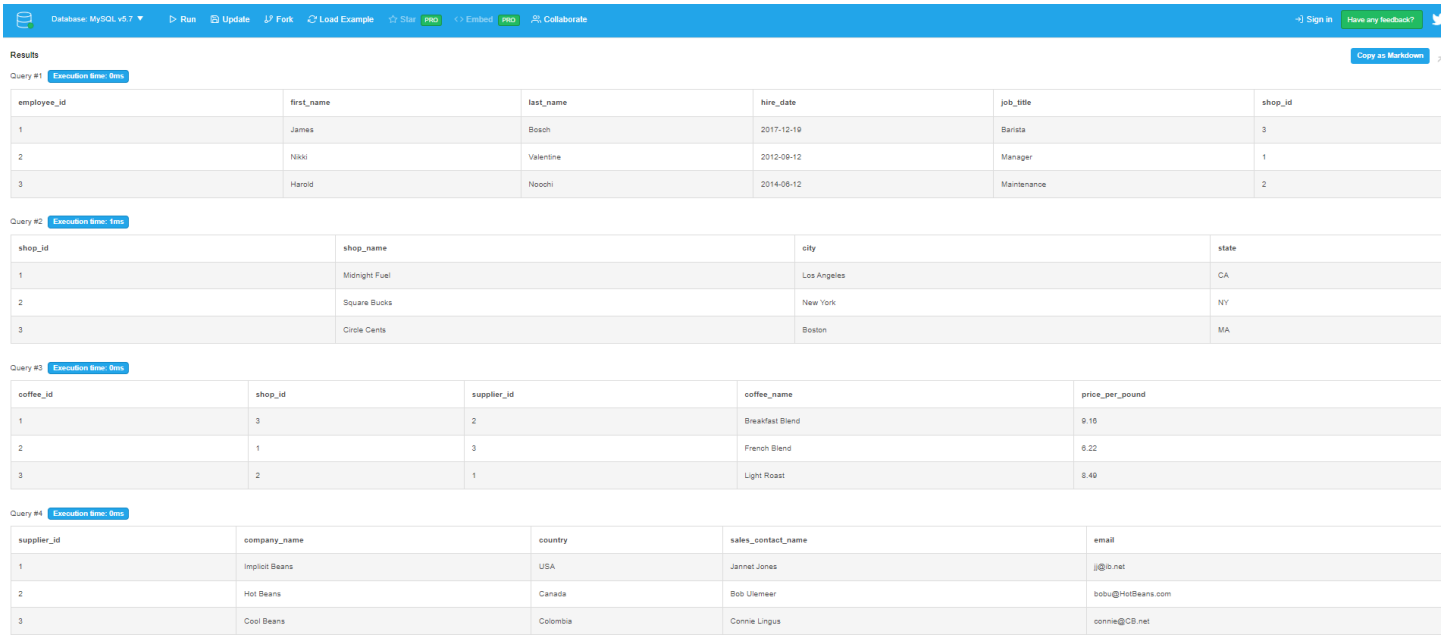

```
CREATE TABLE COFFEE_SHOP (
  shop_id int NOT NULL,
  shop_name VARCHAR(50),
  city VARCHAR(50),
  state char(2),
  PRIMARY KEY (shop_id)
)ENGINE = InnoDB;


CREATE TABLE EMPLOYEE (
  employee_id int NOT NULL,
  first_name VARCHAR(30),
  last_name VARCHAR(30),
  hire_date DATE,
  job_title VARCHAR(30),
  shop_id int NOT NULL,
  PRIMARY KEY (employee_id),
  FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id)
)ENGINE = InnoDB;


CREATE TABLE SUPPLIER (
  supplier_id int,
  company_name VARCHAR(50),
  country VARCHAR(30),
  sales_contact_name VARCHAR(60),
  email VARCHAR(50),
  PRIMARY KEY (supplier_id)
)ENGINE = InnoDB;


CREATE TABLE COFFEE (
  coffee_id int,
  shop_id int,
  supplier_id int,
  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)
)ENGINE = InnoDB;
```

B.2.a – SQL Code

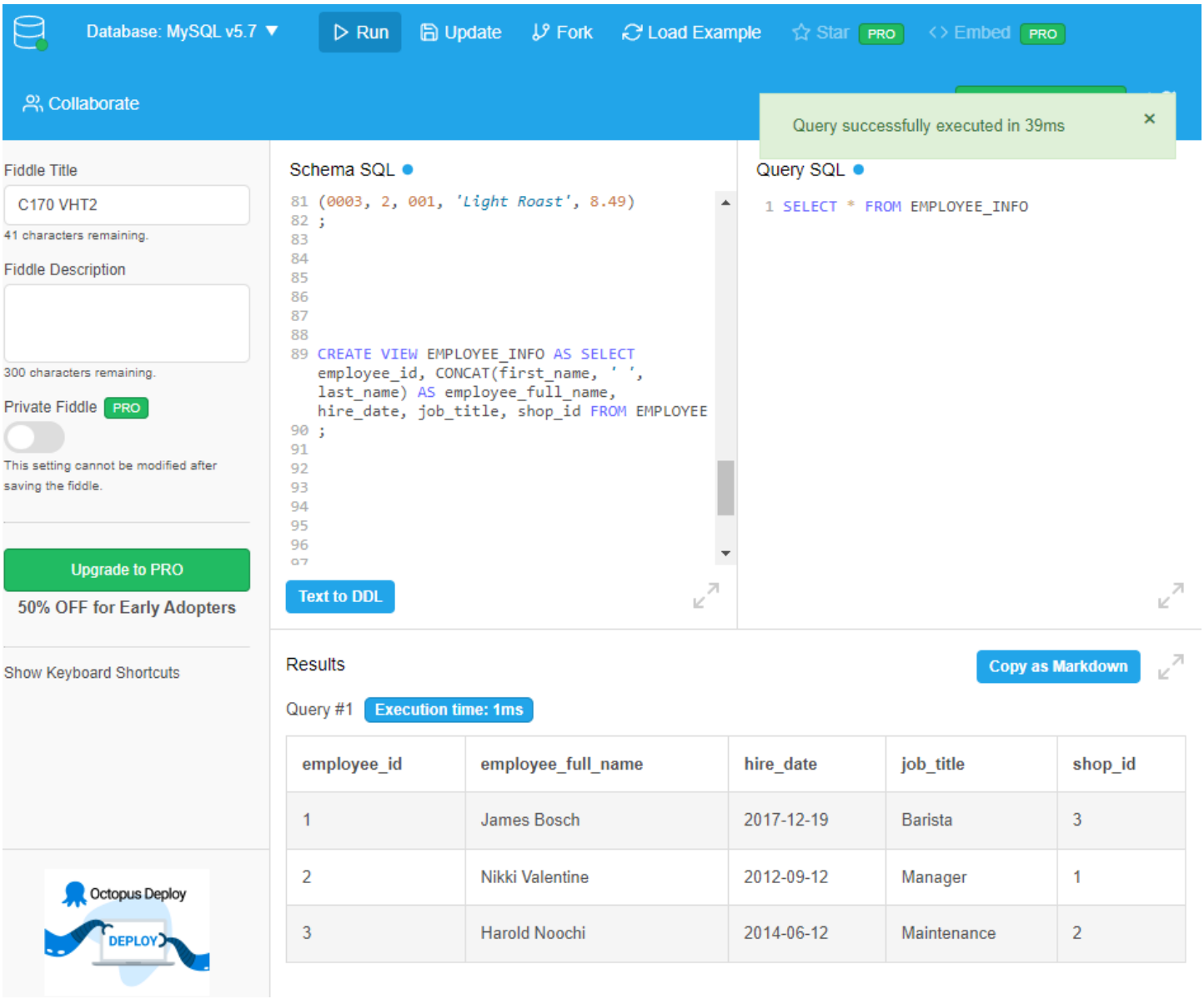



```
INSERT INTO COFFEE_SHOP VALUES
(1, 'Midnight Fuel', 'Los Angeles', 'CA'),
(2, 'Square Bucks', 'New York', 'NY'),
(3, 'Circle Cents', 'Boston', 'MA')
;

INSERT INTO EMPLOYEE VALUES
(1, 'James', 'Bosch', '2017-12-19', 'Barista', 1),
(2, 'Nikki', 'Valentine', '2012-09-12', 'Manager', 2),
(3, 'Harold', 'Noochi', '2014-06-12', 'Maintenance', 3)
;

INSERT INTO SUPPLIER VALUES
(001, 'Implicit Beans', 'USA', 'Jannet Jones', 'jj@ib.net'),
(002, 'Hot Beans', 'Canada', 'Bob Ulemeer', 'bobu@HotBeans.com'),
(003, 'Cool Beans', 'Colombia', 'Connie Lingus', 'connie@CB.net')
;

INSERT INTO COFFEE VALUES
(0001, 3, 002, 'Breakfast Blend', 9.16),
(0002, 1, 003, 'French Blend', 6.22),
(0003, 2, 001, 'Light Roast', 8.49)
;
```

B.2.b - Results

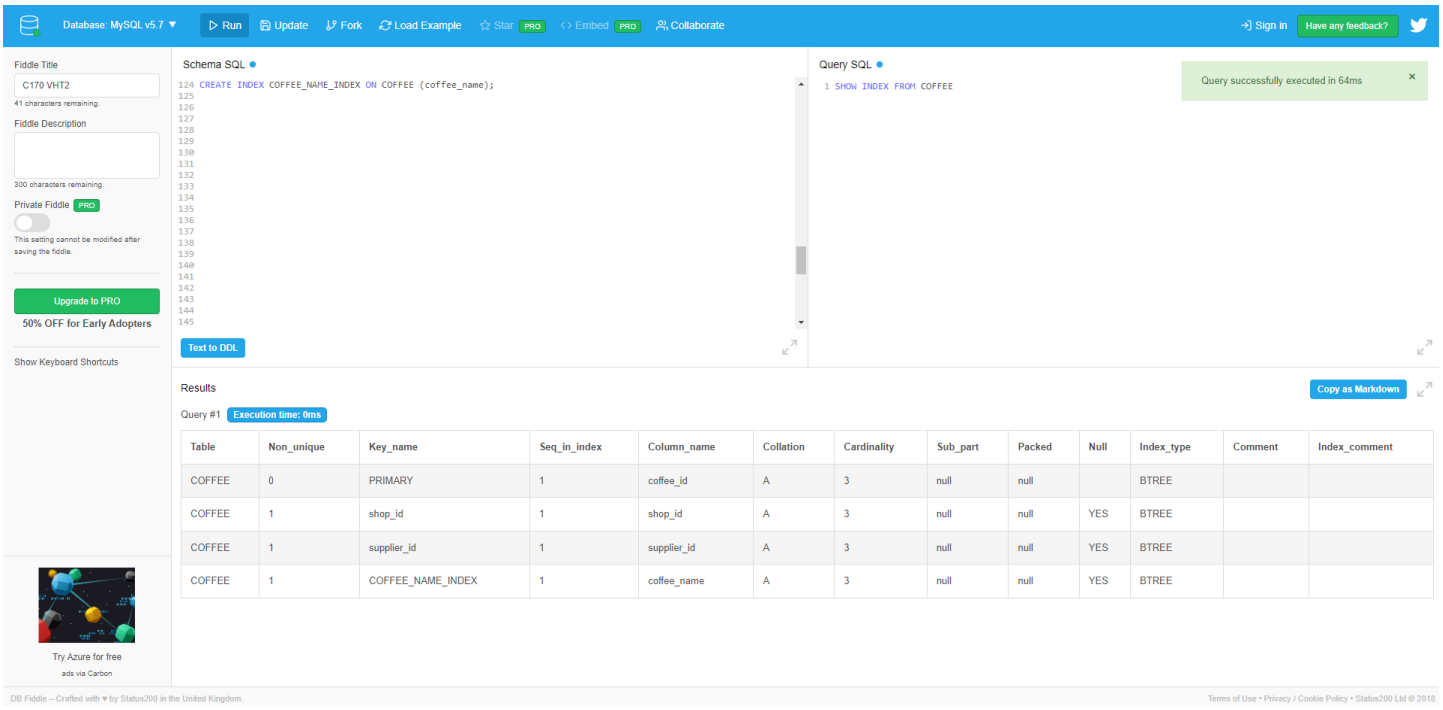

Database: MySQL v5.7
Run
Update
Fork
Load Example
Star PRO
Embed PRO
Collaborate
Sign in
Have any feedback?
Results
Copy as Markdown
Query #1
Execution time: 0ms
employee_id | first_name | last_name | hire_date | job_title | shop_id
1 | James | Bosch | 2017-12-19 | Barista | 3
2 | Nikki | Valentine | 2012-09-12 | Manager | 1
3 | Harold | Noochi | 2014-06-12 | Maintenance | 2
Query #2
Execution time: 1ms
shop_id | shop_name | city | state
1 | Midnight Fuel | Los Angeles | CA
2 | Square Bucks | New York | NY
3 | Circle Cents | Boston | MA
Query #3
Execution time: 0ms
coffee_id | shop_id | supplier_id | coffee_name | price_per_pound
1 | 3 | 2 | Breakfast Blend | 9.16
2 | 1 | 3 | French Blend | 6.22
3 | 2 | 1 | Light Roast | 8.49
Query #4
Execution time: 0ms
supplier_id | company_name | country | sales_contact_name | email
1 | Implicit Beans | USA | Jannet Jones | j@ib.net
2 | Hot Beans | Canada | Bob Ulemeer | bobu@HotBeans.com
3 | Cool Beans | Colombia | Connie Lingus | connie@CB.net

B.3 – Create A View

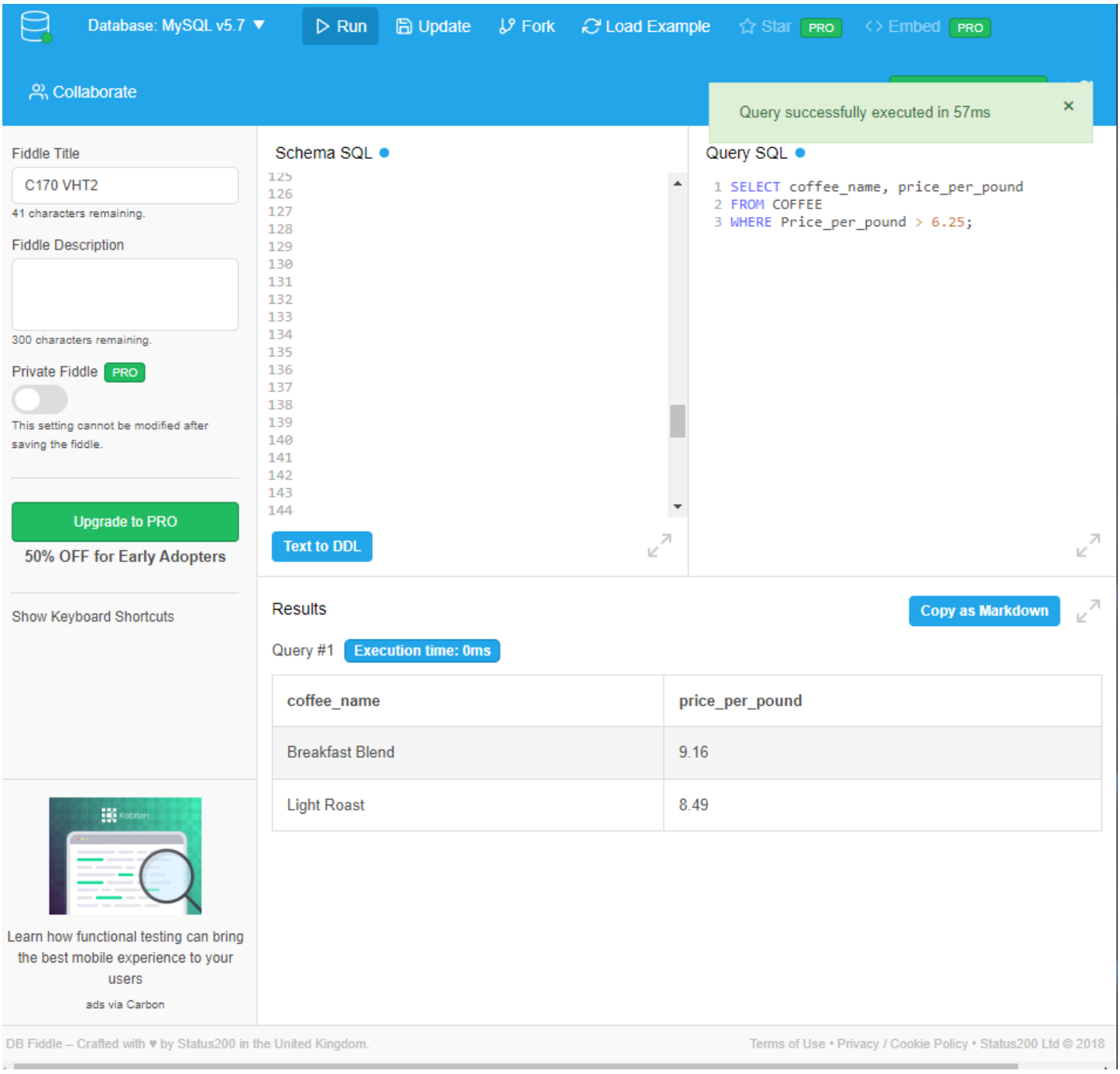


employee_id	employee_full_name	hire_date	job_title	shop_id
1	James Bosch	2017-12-19	Barista	3
2	Nikki Valentine	2012-09-12	Manager	1
3	Harold Noochi	2014-06-12	Maintenance	2

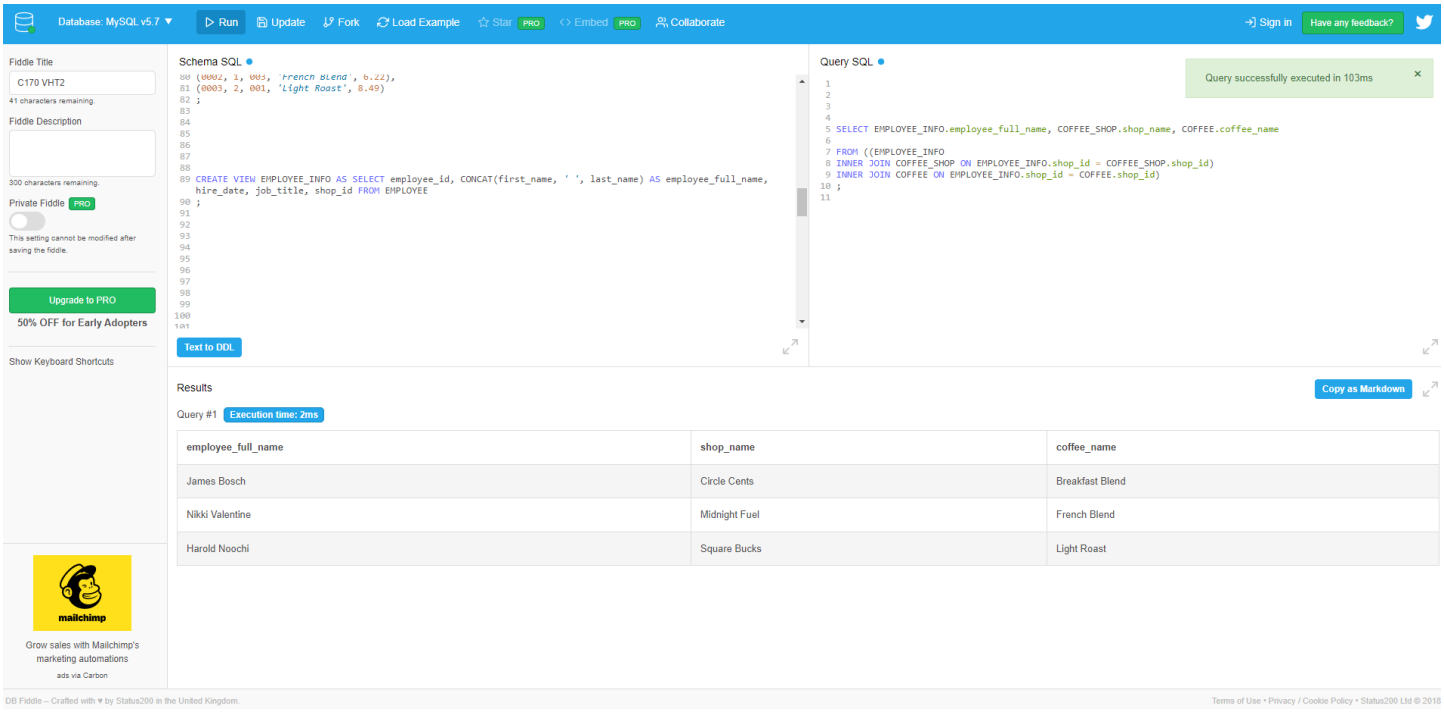
B.4 – Create An Index On Coffee Table



B.5 – Select From Where Query



B.6 – 3 Table Join



employee_full_name	shop_name	coffee_name
James Bosch	Circle Cents	Breakfast Blend
Nikki Valentine	Midnight Fuel	French Blend
Harold Noochi	Square Bucks	Light Roast