PART A: Nora's Bagel Bin Database Blueprints

Western Governors University – C170

Coots, Anthony

Nora’s Bagel Bin Database Blueprints

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

Nora’s Bagel Bin Database Blueprints

(2NF) Second Normal Form

BAGEL ORDER	
PK	Bagel Order ID
	First Name
	Last Name
	Address 1
	Address 2
	City
	State
	Zip
	Mobile Phone
	Delivery Fee
	Bagel Quantity
	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

(2NF) Second Normal Form

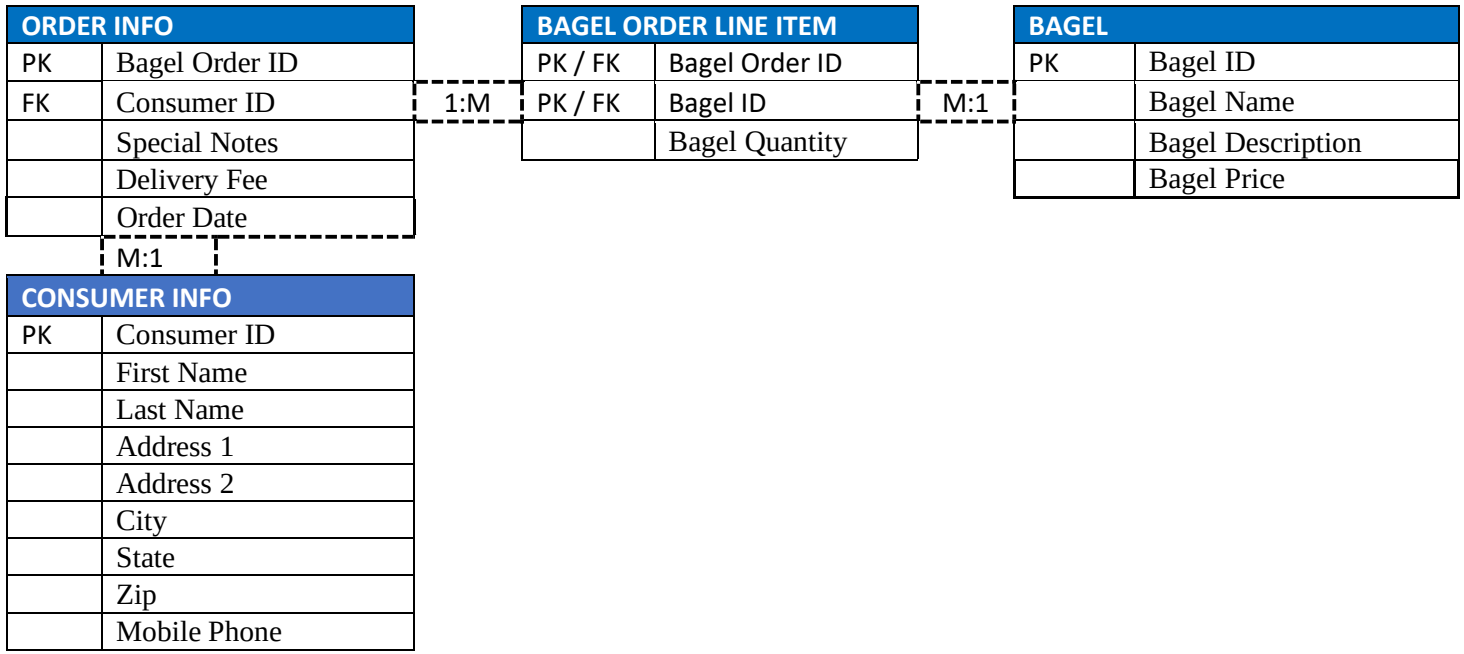
1b. For each one bagel order, assigned an ID for the one order, there may or may not be many bagels assigned with unique bagel IDs. Hence, the one-to-many (1:M) relationship.

For the potential many bagels, given each a unique bagel ID, there should only be one bagel for each many bagel IDs. Hence, the many-to-one (M:1) relationship.

1c. Reading from left most table to right most table; The 'Bagel Order' table is assigned information related to the consumer along with the assigned bagel order ID. Each bagel order will generate its own new bagel order ID with input consumer information. As the one order ID is made, the order may consist of multiple (more than one) different bagels. Therefore, each bagel within the bagel order shall be separated to associated order ID, the unique ID of the bagel itself, and the quantity of such bagel within the order. Following the unique bagel ID is the information for each bagel, that being its matching bagel ID, name, description and price.

Nora’s Bagel Bin Database Blueprints

(3NF) Third Normal Form



(3NF) Third Normal Form

2d. Describing the relationship from top to bottom; For the potential many orders a customer may have, there should only exist one record for each consumer ID, regardless of the number of order IDs that consumer ID may appear in. Hence the many-to-one (M:1) relationship.

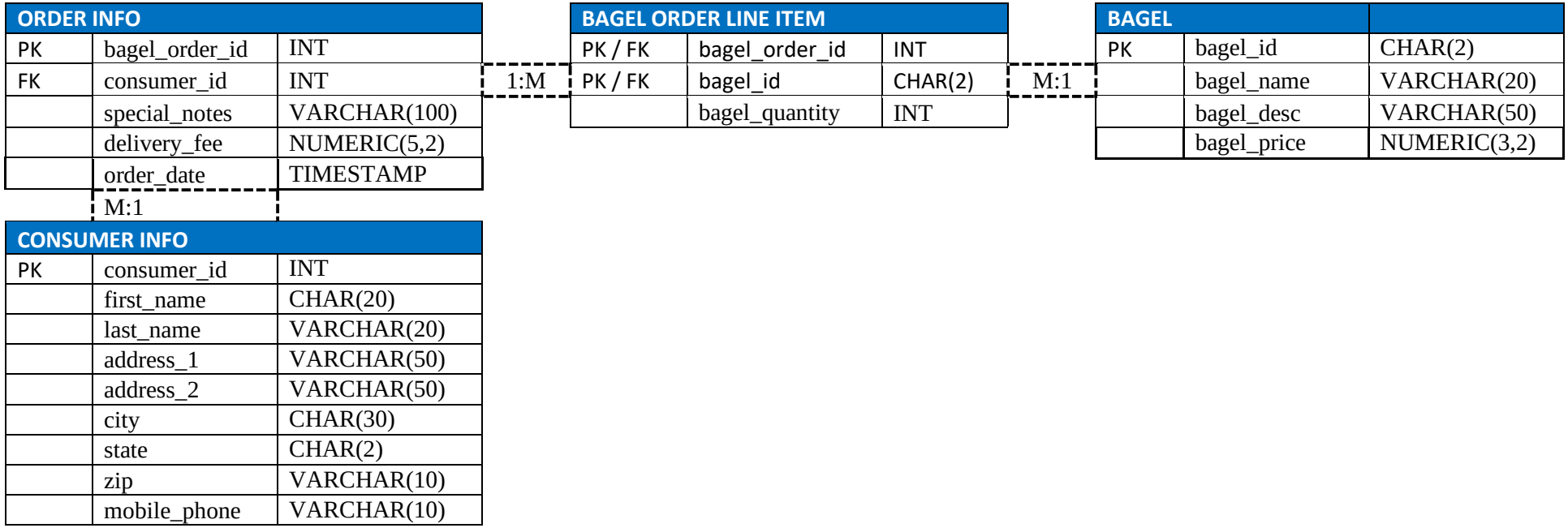
Now following the left most table to right most table; each order will contain a one unique bagel order ID, the bagel order ID may or may not return many bagel IDs as each order can consist of one or more (many) bagels. Hence the one-to-many (1:M) relationship.

For the potential many bagels, given each a unique bagel ID, there should only be one bagel for each many bagel IDs. Hence, the many-to-one (M:1) relationship. Note: There is no change from the 2NF and 3NF here.

2e. The changes from 2NF to 3NF only consist of one notable adjustment. Referencing (**1c.)**, what was previously 'Bagel Order Info' table now splits to parent table 'Order Info' to child table 'Consumer Info'. Along with what already exists among section **1c's** information, now, the consumer info deviates into its own table consisting of first name, last name, addresses one and two, city, state, zip, and mobile phone now represented by its own 'Consumer ID' primary key, which is referenced by the foreign key in the now 'Order Info' parent table.

Nora’s Bagel Bin Database Blueprints

Final Physical Database Model



PART B: Jaunty Coffee Co. ERD

Western Governors University – C170

Coots, Anthony

B1.

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

CREATE TABLE EMPLOYEE(
  employee_id INT,
  first_name VARCHAR(30),
  last_name VARCHAR(30),
  hire_date DATE,
  job_title VARCHAR(30),
  shop_id INT,
  PRIMARY KEY (employee_id),
  FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id)
);

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

```
CREATE TABLE COFFEE( -- created by acoots5@wgu.edu
  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 EMPLOYEE(shop_id),
  FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
);
```

B2.

```
INSERT INTO COFFEE_SHOP VALUES
(100, 'Barstucks', 'Los Angeles', 'CA'),
(101, 'Smooth Brew', 'New York City', 'NY'),
(102, 'Top Pot', 'Arlington', 'TX');

INSERT INTO EMPLOYEE VALUES
(10201, 'Michael', 'Jordan', '1984-10-26', 'Manager', 102),
(10101, 'Mikal', 'Bridges', '2018-10-17', 'Shift Lead', 101),
(10001, 'LeBron', 'James', '2003-10-29', 'Owner', 100);

INSERT INTO SUPPLIER VALUES
(1, 'Big Face Brand', 'United States of America', 'Jimmy Butler', 'jimmygbuckets@bigfacecomp.net'),
(2, 'Beans n\' Stuff', 'United States of America', 'Adam Silver', 'Adam_Silver@real_domain.net'),
(3, 'Java James Naismith', 'Canada', 'Dr. James Naismith', 'PeachBasketAthletics@JJCo.com');

INSERT INTO COFFEE VALUES
(1, 100, 3, 'G.O.A.T Fuel', 6.23),
(2, 102, 2, 'Six A.M. Energy', 4.50),
(3, 101, 1, 'DOUBLESTAR', 5.22);
```

B3.

```
CREATE VIEW full_name AS
SELECT employee_id,
    CONCAT(first_name, ' ', last_name) AS employee_full_name,
    first_name,
    last_name,
    hire_date,
    job_title,
    shop_id
FROM EMPLOYEE;
```

B4.

```
CREATE INDEX coffee_name_index
ON COFFEE(coffee_name);
```

B5.

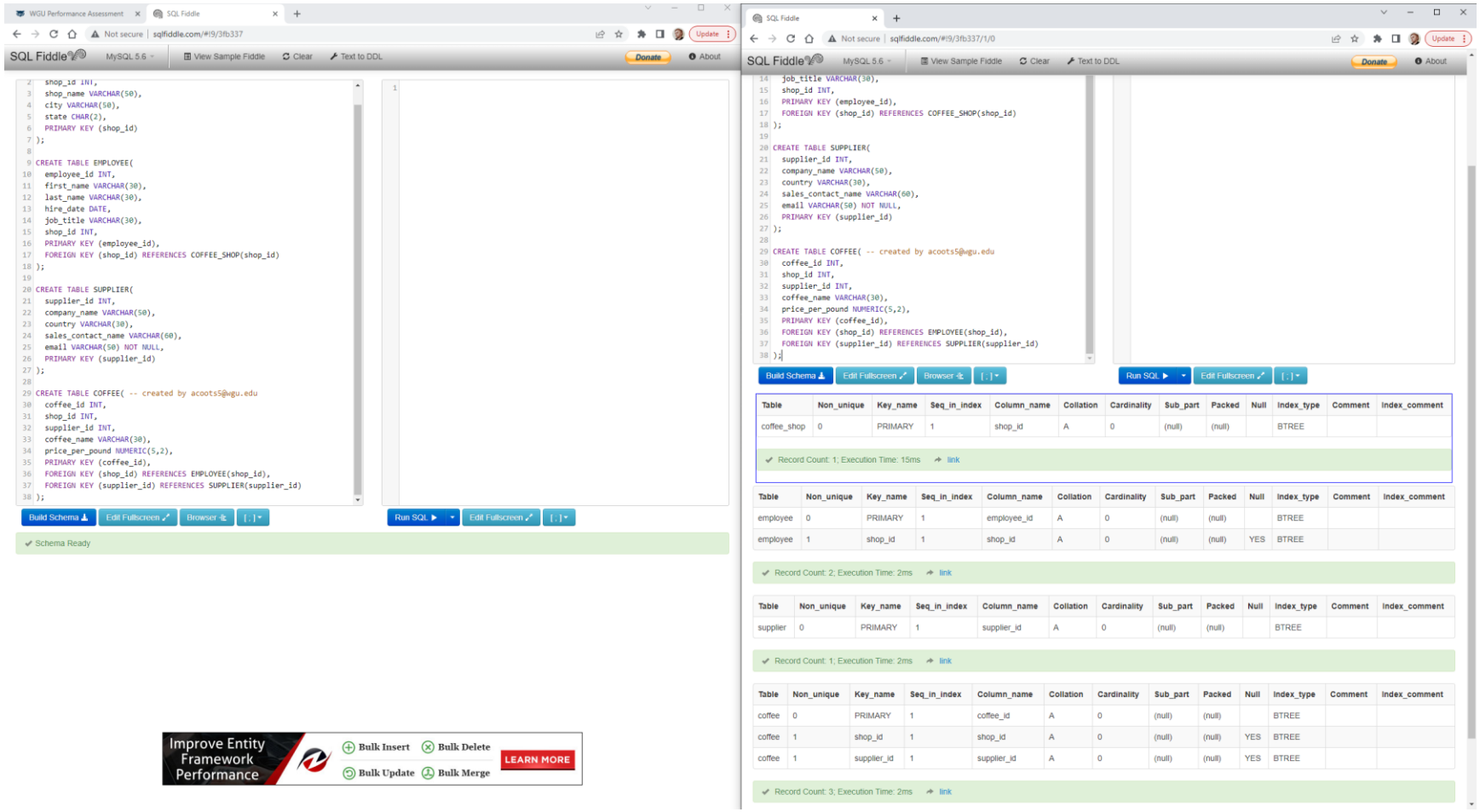
```
SELECT city
FROM COFFEE_SHOP
WHERE shop_id = 100;


SELECT email
FROM SUPPLIER
WHERE email LIKE '%.net';
```

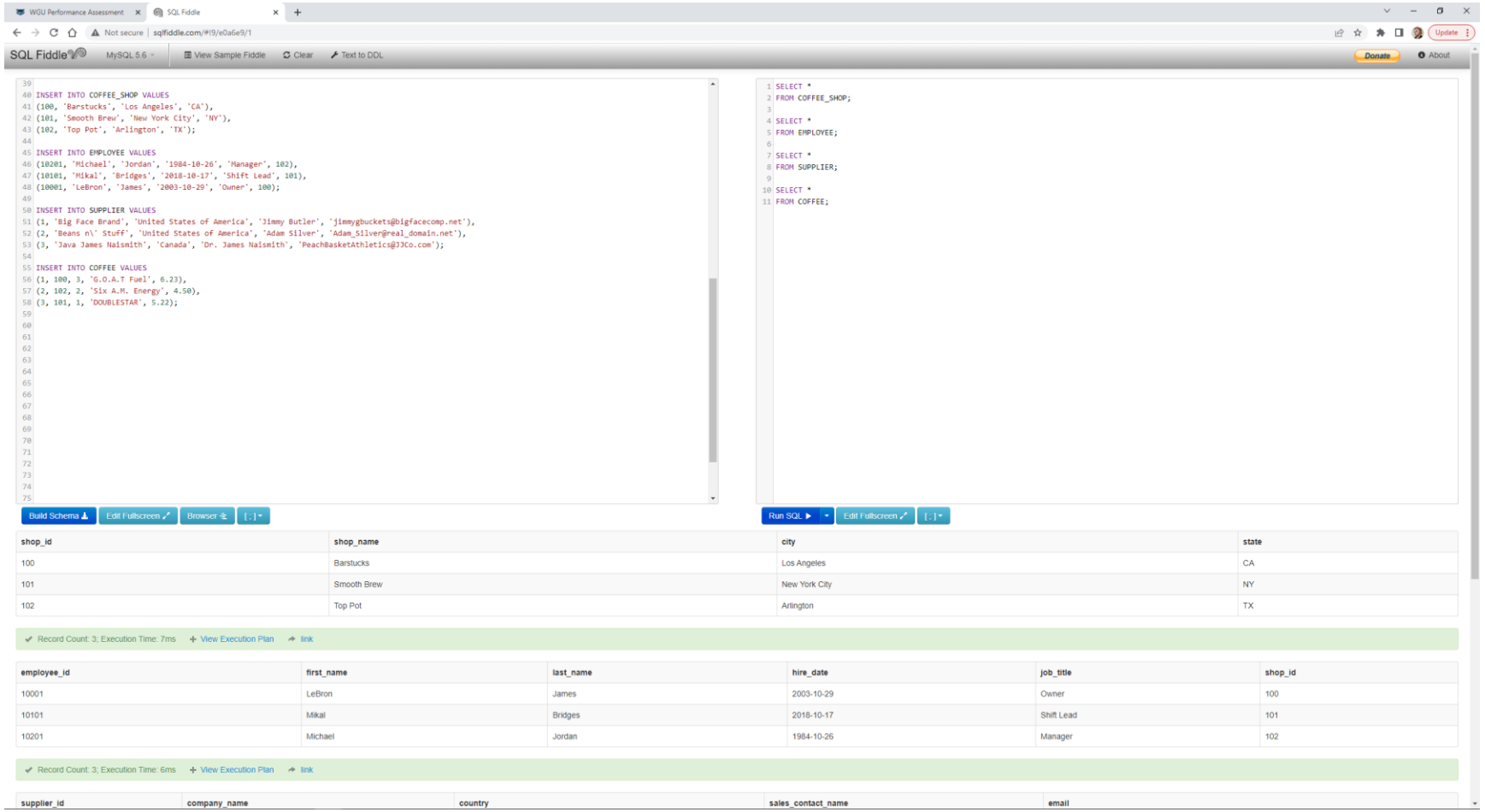
B6.

```
SELECT cs.shop_name
    ,cs.city
    ,cs.state
    ,emp.first_name
    ,emp.last_name
    ,emp.job_title
    ,cfe.coffee_name
FROM COFFEE_SHOP cs INNER JOIN EMPLOYEE emp
            ON cs.shop_id = emp.shop_id INNER JOIN COFFEE cfe
                        ON cs.shop_id = cfe.shop_id AND emp.employee_id = 10001;
```

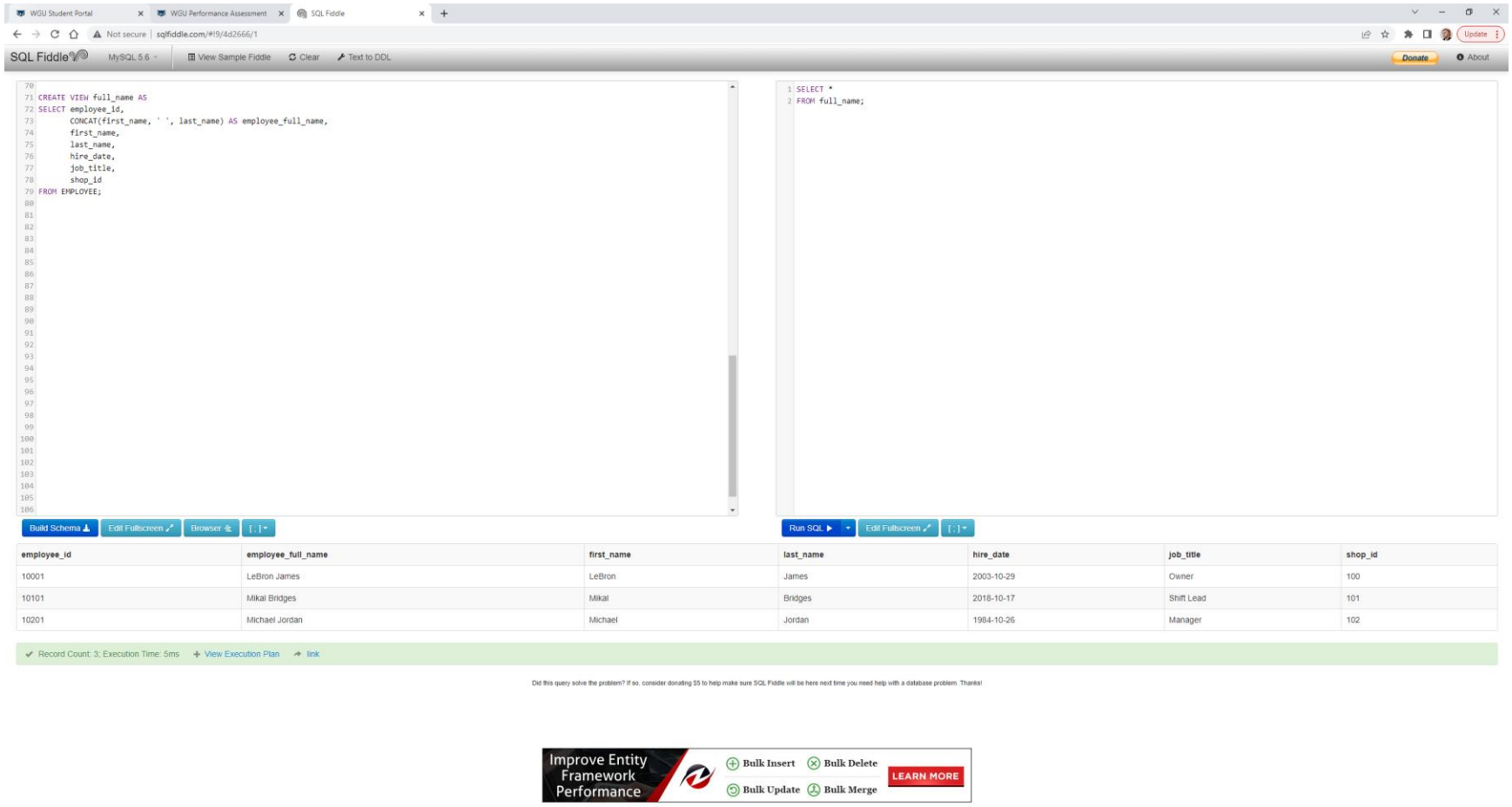
B1.


SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
Build Schema
Edit Fullscreen
Browser
Run SQL
Schema Ready
Improve Entity Framework Performance
Bulk Insert
Bulk Delete
Bulk Update
Bulk Merge
LEARN MORE

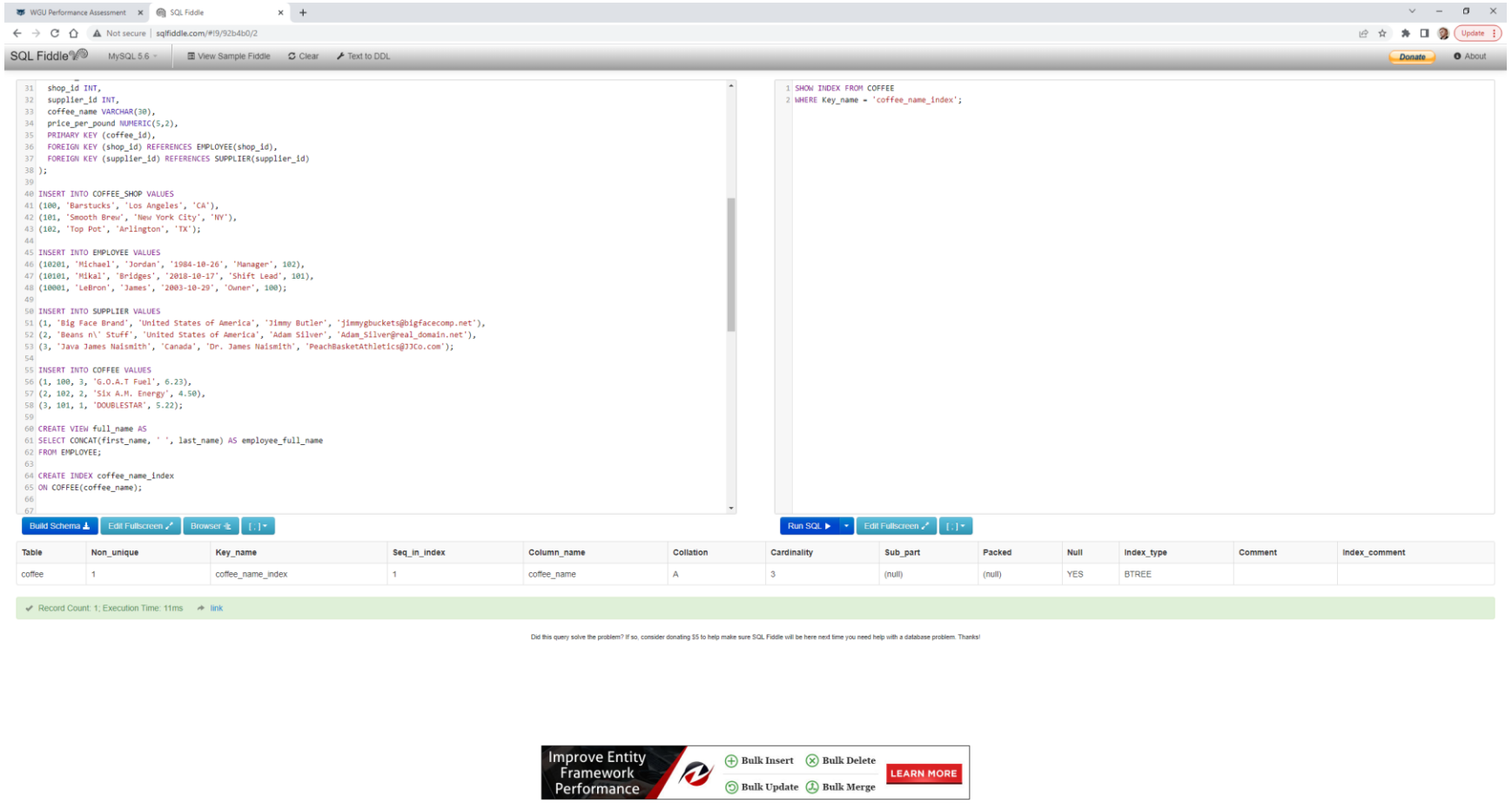
B2.


WGU Performance Assessment
SQL Fiddle
Not secure | sqlfiddle.com/#!9/e0a6e9/1
SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
Update
INSERT INTO COFFEE_SHOP VALUES
(100, 'Barstucks', 'Los Angeles', 'CA'),
(101, 'Smooth Brew', 'New York City', 'NY'),
(102, 'Top Pot', 'Arlington', 'TX');
INSERT INTO EMPLOYEE VALUES
(10201, 'Michael', 'Jordan', '1984-10-26', 'Manager', 102),
(10101, 'Mikal', 'Bridges', '2018-10-17', 'Shift Lead', 101),
(10001, 'LeBron', 'James', '2003-10-29', 'Owner', 100);
INSERT INTO SUPPLIER VALUES
(1, 'Big Face Brand', 'United States of America', 'Jimmy Butler', 'jimmygbuckets@bigfacecomp.net'),
(2, 'Beans n\' Stuff', 'United States of America', 'Adam Silver', 'Adam_Silver@real_domain.net'),
(3, 'Java James Naismith', 'Canada', 'Dr. James Naismith', 'PeachBasketAthletics@JJCo.com');
INSERT INTO COFFEE VALUES
(1, 100, 3, 'G.O.A.T Fuel', 6.23),
(2, 102, 2, 'Six A.M. Energy', 4.50),
(3, 101, 1, 'DOUBLESTAR', 5.22);
SELECT *
FROM COFFEE_SHOP;
SELECT *
FROM EMPLOYEE;
SELECT *
FROM SUPPLIER;
SELECT *
FROM COFFEE;
Build Schema
Edit Fullscreen
Browser
Run SQL
Edit Fullscreen
shop_id
shop_name
city
state
100
Barstucks
Los Angeles
CA
101
Smooth Brew
New York City
NY
102
Top Pot
Arlington
TX
Record Count: 3; Execution Time: 7ms
View Execution Plan
link
employee_id
first_name
last_name
hire_date
job_title
shop_id
10001
LeBron
James
2003-10-29
Owner
100
10101
Mikal
Bridges
2018-10-17
Shift Lead
101
10201
Michael
Jordan
1984-10-26
Manager
102
Record Count: 3; Execution Time: 6ms
View Execution Plan
link
supplier_id
company_name
country
sales_contact_name
email

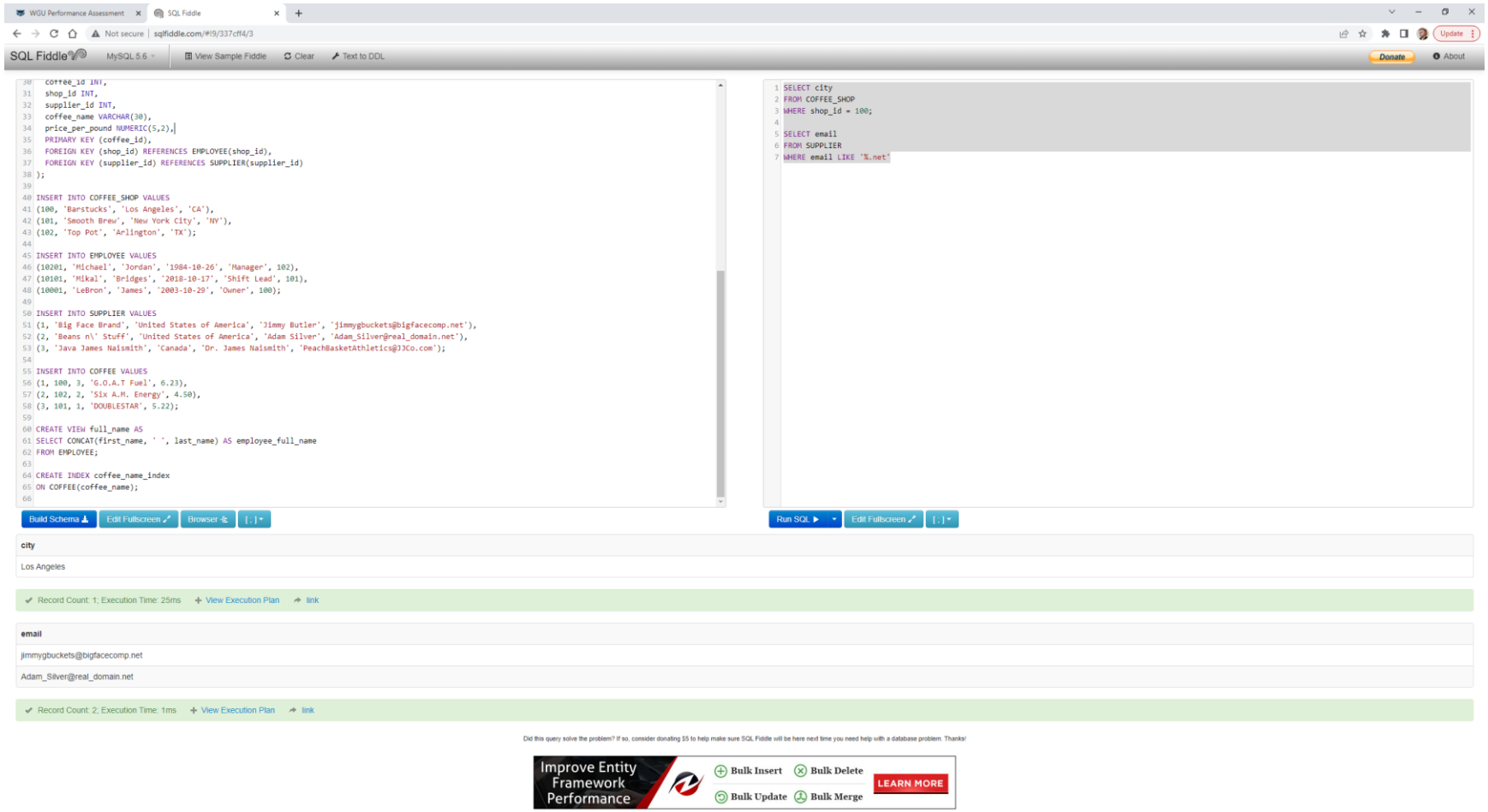
B3.


WGU Student Portal
WGU Performance Assessment
SQL Fiddle
Not secure | sqlfiddle.com/#!9/4d2666/1
SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
CREATE VIEW full_name AS
SELECT employee_id,
CONCAT(first_name, ' ', last_name) AS employee_full_name,
first_name,
last_name,
hire_date,
job_title,
shop_id
FROM EMPLOYEE;
SELECT *
FROM full_name;
Build Schema
Edit Fullscreen
Browser
Run SQL
Edit Fullscreen
employee_id | employee_full_name | first_name | last_name | hire_date | job_title | shop_id
10001 | LeBron James | LeBron | James | 2003-10-29 | Owner | 100
10101 | Mikal Bridges | Mikal | Bridges | 2018-10-17 | Shift Lead | 101
10201 | Michael Jordan | Michael | Jordan | 1984-10-26 | Manager | 102
Record Count: 3; Execution Time: 5ms
View Execution Plan
link
Did this query solve the problem? If so, consider donating $5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!
Improve Entity Framework Performance
Bulk Insert
Bulk Delete
Bulk Update
Bulk Merge
LEARN MORE

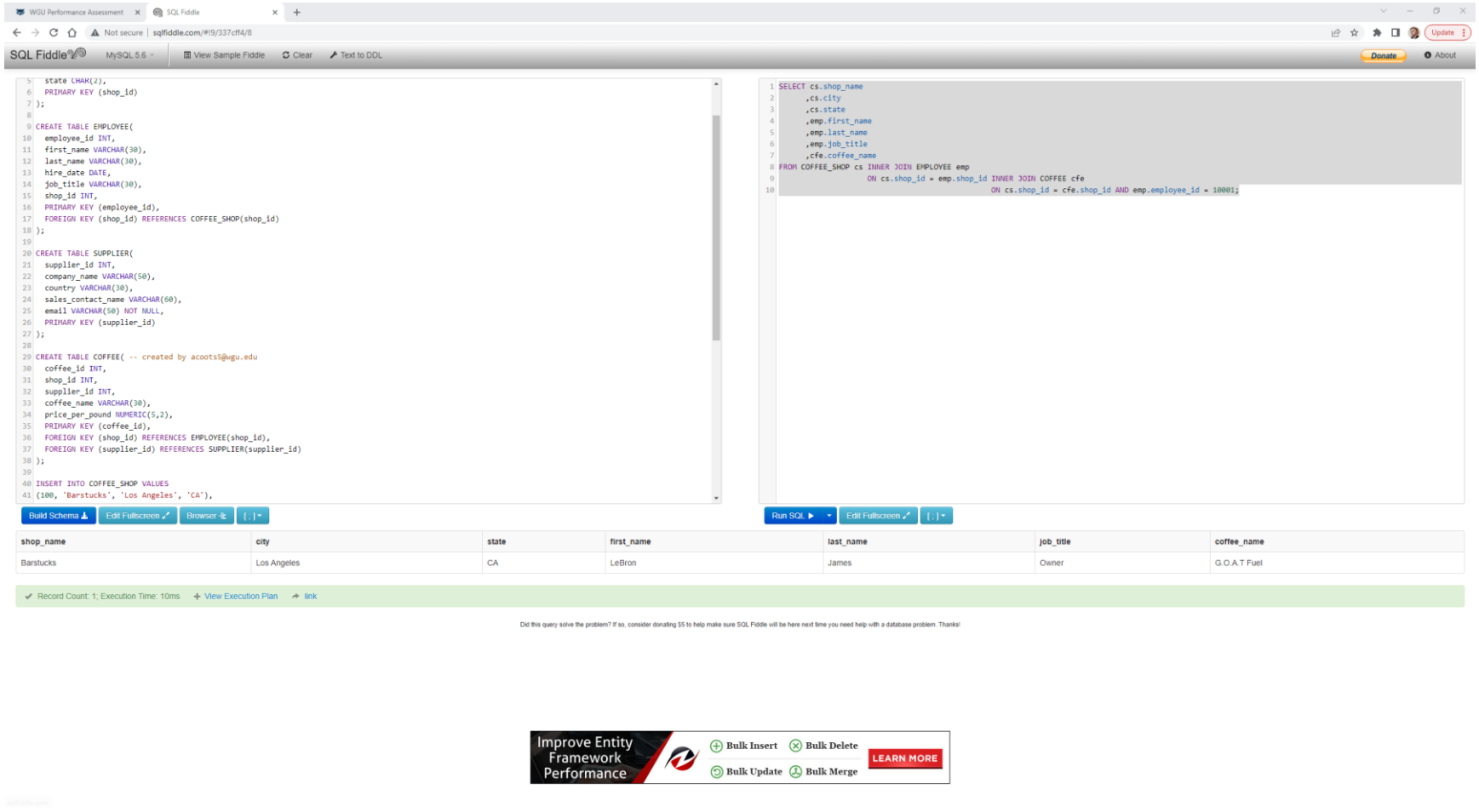
B4.



B5.


WGU Performance Assessment
SQL Fiddle
Not secure | sqlfiddle.com/#!9/337cff4/3
SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
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 EMPLOYEE(shop_id),
FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
);
INSERT INTO COFFEE_SHOP VALUES
(100, 'Barstucks', 'Los Angeles', 'CA'),
(101, 'Smooth Brew', 'New York City', 'NY'),
(102, 'Top Pot', 'Arlington', 'TX');
INSERT INTO EMPLOYEE VALUES
(10201, 'Michael', 'Jordan', '1984-10-26', 'Manager', 102),
(10101, 'Mikal', 'Bridges', '2018-10-17', 'Shift Lead', 101),
(10001, 'LeBron', 'James', '2003-10-29', 'Owner', 100);
INSERT INTO SUPPLIER VALUES
(1, 'Big Face Brand', 'United States of America', 'Jimmy Butler', 'jimmygbuckets@bigfacecomp.net'),
(2, 'Beans n\' Stuff', 'United States of America', 'Adam Silver', 'Adam_Silver@real_domain.net'),
(3, 'Java James Naismith', 'Canada', 'Dr. James Naismith', 'PeachBasketAthletics@JJCo.com');
INSERT INTO COFFEE VALUES
(1, 100, 3, 'G.O.A.T Fuel', 6.23),
(2, 102, 2, 'Six A.M. Energy', 4.50),
(3, 101, 1, 'DOUBLESTAR', 5.22);
CREATE VIEW full_name AS
SELECT CONCAT(first_name, ' ', last_name) AS employee_full_name
FROM EMPLOYEE;
CREATE INDEX coffee_name_index
ON COFFEE(coffee_name);
SELECT city
FROM COFFEE_SHOP
WHERE shop_id = 100;
SELECT email
FROM SUPPLIER
WHERE email LIKE '%.net'
Build Schema
Edit Fullscreen
Browser
Run SQL
Edit Fullscreen
city
Los Angeles
Record Count: 1, Execution Time: 25ms
View Execution Plan
link
email
jimmygbuckets@bigfacecomp.net
Adam_Silver@real_domain.net
Record Count: 2, Execution Time: 1ms
View Execution Plan
link
Did this query solve the problem? If so, consider donating $5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!
Improve Entity Framework Performance
Bulk Insert
Bulk Delete
Bulk Update
Bulk Merge
LEARN MORE

B6.


WGU Performance Assessment
SQL Fiddle
Not secure | sqlfiddle.com/#!9/337cff4/8
SQL Fiddle
MySQL 5.6
View Sample Fiddle
Clear
Text to DDL
Donate
About
state CHAR(2),
PRIMARY KEY (shop_id)
);
CREATE TABLE EMPLOYEE(
employee_id INT,
first_name VARCHAR(30),
last_name VARCHAR(30),
hire_date DATE,
job_title VARCHAR(30),
shop_id INT,
PRIMARY KEY (employee_id),
FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id)
);
CREATE TABLE SUPPLIER(
supplier_id INT,
company_name VARCHAR(50),
country VARCHAR(30),
sales_contact_name VARCHAR(60),
email VARCHAR(50) NOT NULL,
PRIMARY KEY (supplier_id)
);
CREATE TABLE COFFEE(-- created by acoots5@wgu.edu
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 EMPLOYEE(shop_id),
FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
);
INSERT INTO COFFEE_SHOP VALUES
(100, 'Barstucks', 'Los Angeles', 'CA'),
SELECT cs.shop_name
,cs.city
,cs.state
,emp.first_name
,emp.last_name
,emp.job_title
,cfe.coffee_name
FROM COFFEE_SHOP cs INNER JOIN EMPLOYEE emp
ON cs.shop_id = emp.shop_id INNER JOIN COFFEE cfe
ON cs.shop_id = cfe.shop_id AND emp.employee_id = 10001;
Build Schema
Edit Fullscreen
Browser
Run SQL
Edit Fullscreen
shop_name
city
state
first_name
last_name
job_title
coffee_name
Barstucks
Los Angeles
CA
LeBron
James
Owner
G.O.A.T Fuel
Record Count: 1; Execution Time: 10ms
View Execution Plan
link
Did this query solve the problem? If so, consider donating $5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!
Improve Entity Framework Performance
Bulk Insert
Bulk Delete
Bulk Update
Bulk Merge
LEARN MORE