# Nora's Bagel Bin Database Blueprints *(continued)*

**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 |

One bagel order can have many Bagel line items (And many bagel order line items can go on the same order)

A bagel order line item can only have one bagel, but one bagel can be associated with many bagel order line items.

The bagel table should include all information about the bagel (that is related to the bagel ID primary Key).

The bagel order line item should contain all information about the order (That relates to the Bagel Order ID).

The Bagel Quantity belongs in the bagel order line item as that is not dependent on either the Bagel ID or the Bagel order ID.





**Third Normal Form (3NF)**

| Bagel Order | |
|---|---|
| PK | Bagel Order ID |
| FK | Customer ID |
| | Delivery Fee |
| | Order Date |
| | 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

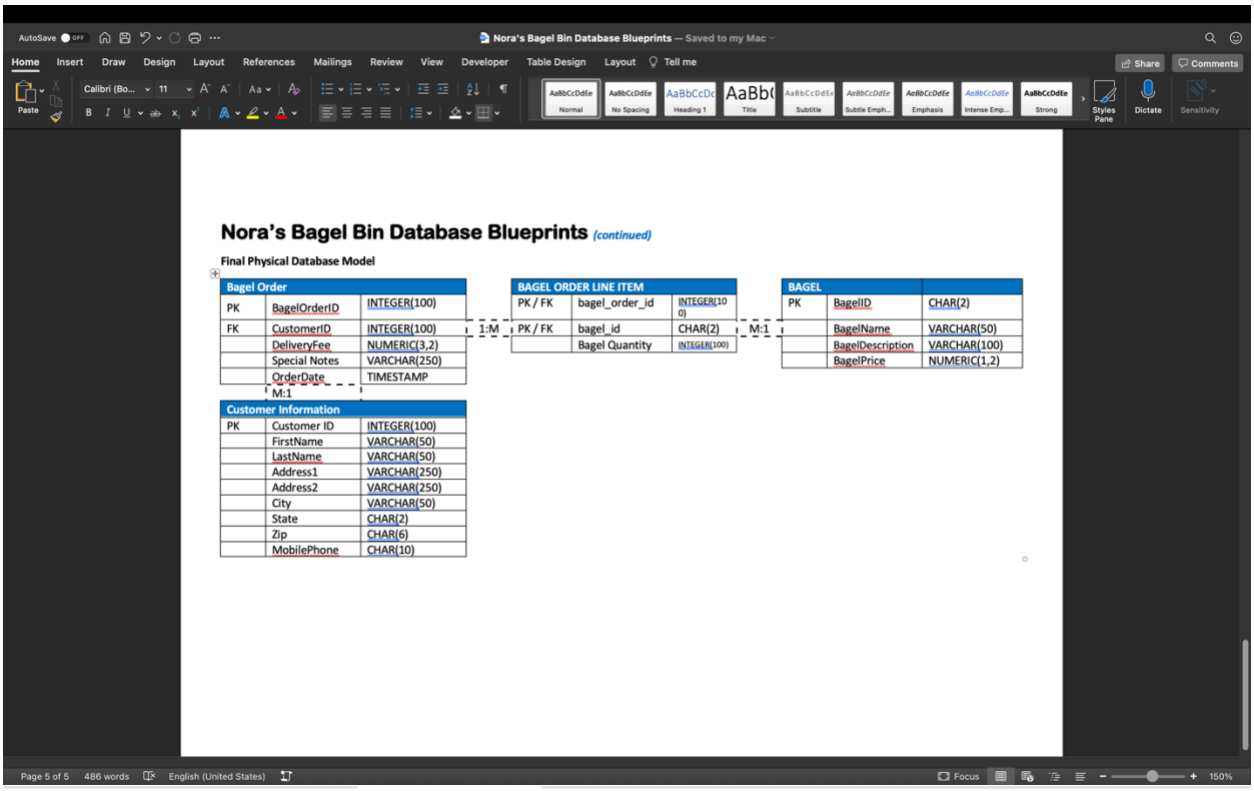
| Customer Information | |
|---|---|
| PK | Customer ID |
| | First Name |
| | Last Name |
| | Address 1 |
| | Address 2 |
| | City |
| | State |
| | Zip |
| | Mobile Phone |

Building on the 2NF table, for the 3NF table we need to move all of the customer information to it's own table. This is because the customer information does not rely on the Bagel Order ID. We then need to create a Primary Key for the Customers table and assign it as a Foreign Key in the Bagel Order Table.

One customer can be associated with many. Bagel orders, however a bagel order can only have one customer.

One bagel order can have many Bagel line items (And many bagel order line items can go on the same order)

A bagel order line item can only have one bagel, but one bagel can be associated with many bagel order line items.



## Nora's Bagel Bin Database Blueprints *(continued)*

**Final Physical Database Model**

| Bagel Order | | |
|---|---|---|
| PK | BagelOrderID | INTEGER(100) |
| FK | CustomerID | INTEGER(100) |
| | DeliveryFee | NUMERIC(3,2) |
| | Special Notes | VARCHAR(250) |
| | OrderDate | TIMESTAMP |

1:M

| BAGEL ORDER LINE ITEM | | |
|---|---|---|
| PK / FK | bagel_order_id | INTEGER(100) |
| PK / FK | bagel_id | CHAR(2) |
| | Bagel Quantity | INTEGER(100) |

M:1

| BAGEL | | |
|---|---|---|
| PK | BagelID | CHAR(2) |
| | BagelName | VARCHAR(50) |
| | BagelDescription | VARCHAR(100) |
| | BagelPrice | NUMERIC(1,2) |

M:1

| Customer Information | | |
|---|---|---|
| PK | Customer ID | INTEGER(100) |
| | FirstName | VARCHAR(50) |
| | LastName | VARCHAR(50) |
| | Address1 | VARCHAR(250) |
| | Address2 | VARCHAR(250) |
| | City | VARCHAR(50) |
| | State | CHAR(2) |
| | Zip | CHAR(6) |
| | MobilePhone | CHAR(10) |

db-fiddle.com

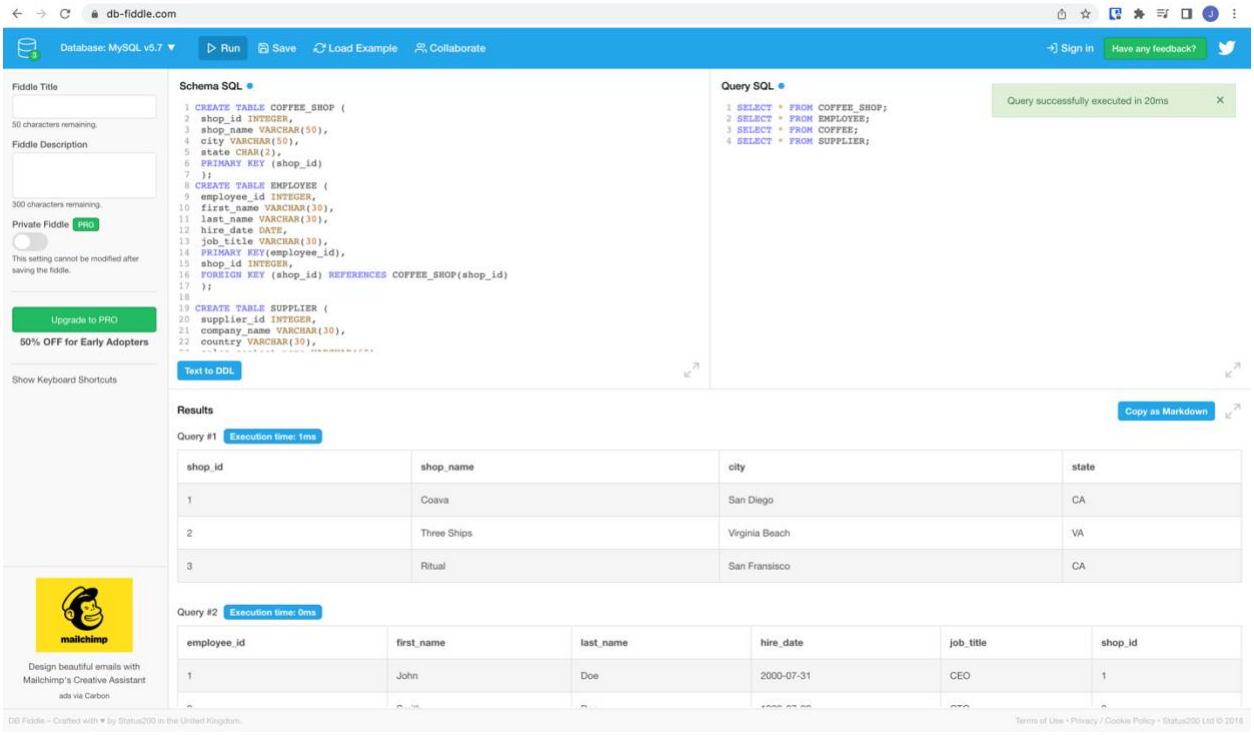
Database: MySQL v5.7 ▼ | Run | Save | Load Example | Collaborate | Sign in | Have any feedback?

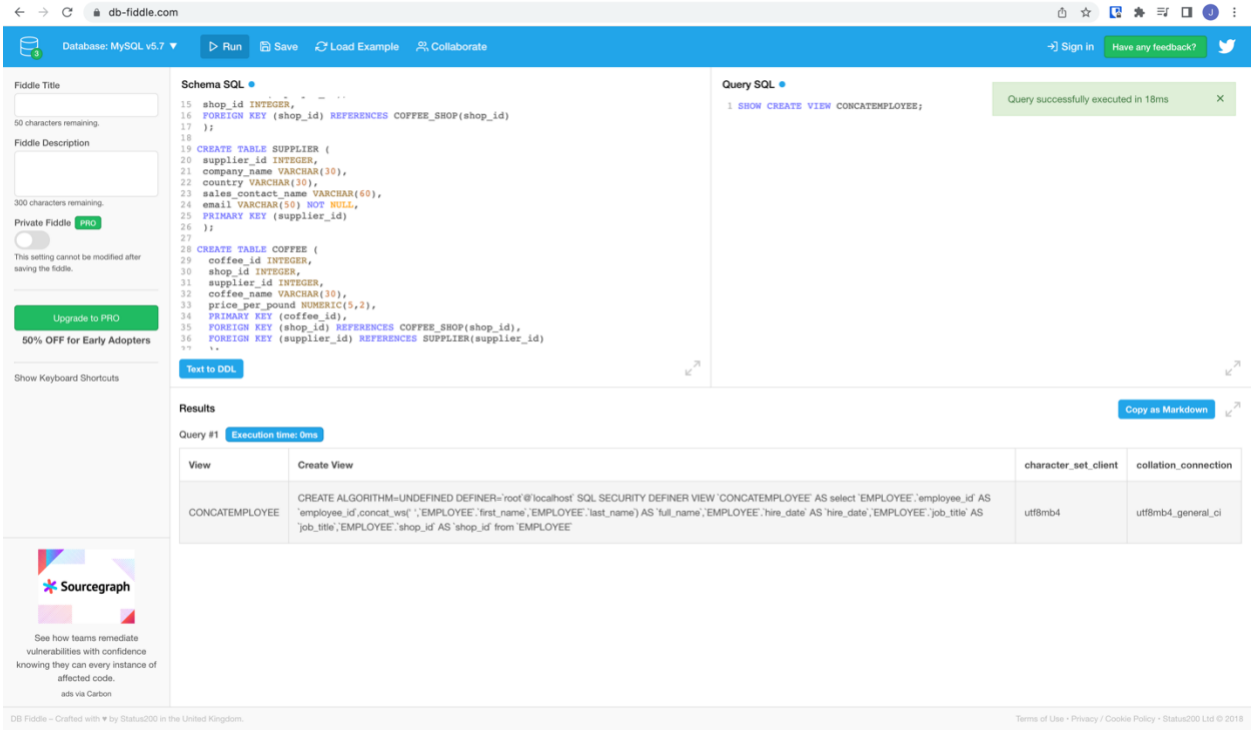
Schema SQL

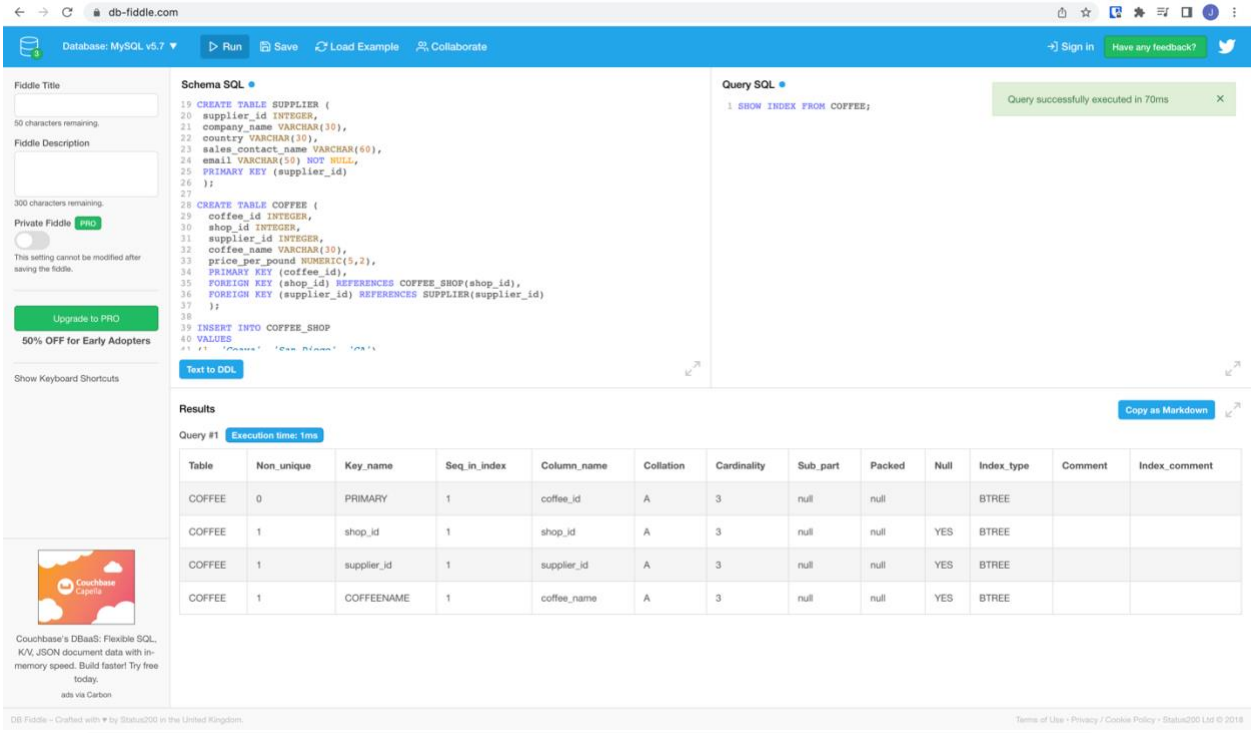
Query successfully executed in 19ms

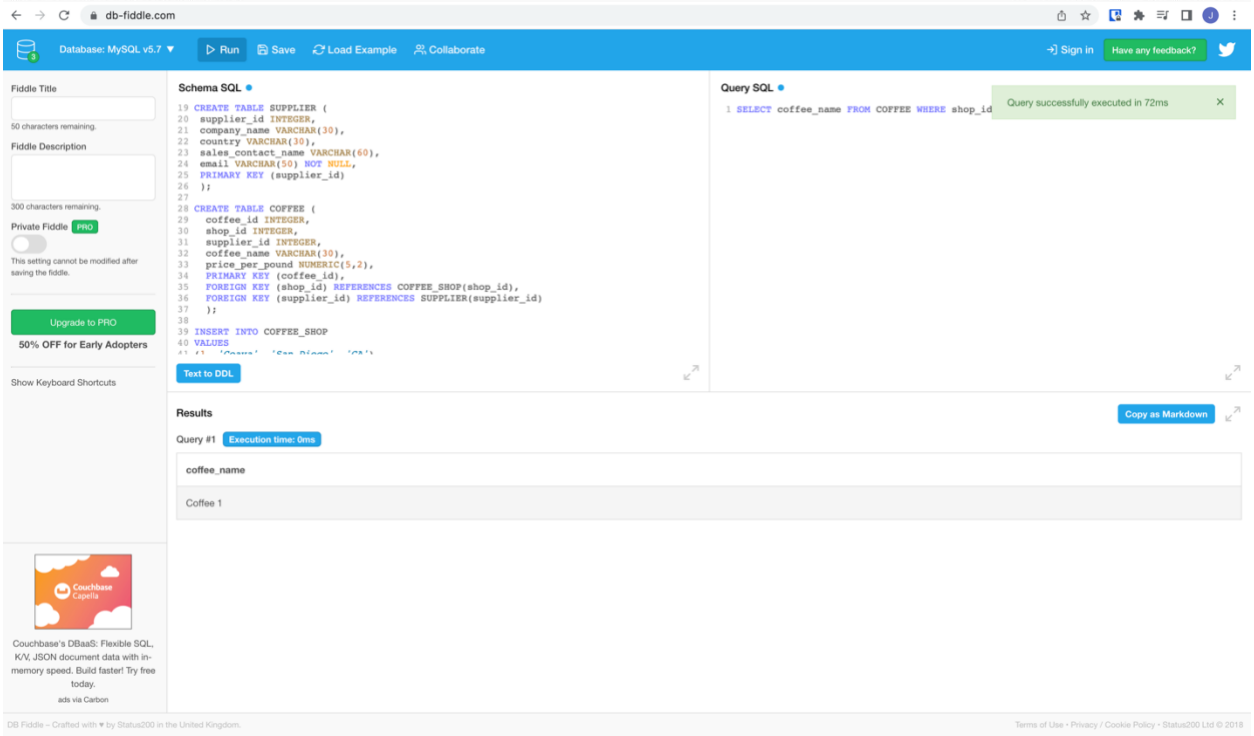
```sql
CREATE TABLE COFFEE_SHOP (
 shop_id INTEGER,
 shop_name VARCHAR(50),
 city VARCHAR(50),
 state CHAR(2),
 PRIMARY KEY (shop_id)
 );
CREATE TABLE EMPLOYEE (
 employee_id INTEGER,
 first_name VARCHAR(30),
 last_name VARCHAR(30),
 hire_date DATE,
 job_title VARCHAR(30),
 PRIMARY KEY(employee_id),
 shop_id INTEGER,
 FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id)
 );

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

CREATE TABLE COFFEE (
  coffee_id INTEGER,
  shop_id INTEGER,
  supplier_id INTEGER,
  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)
  );
```

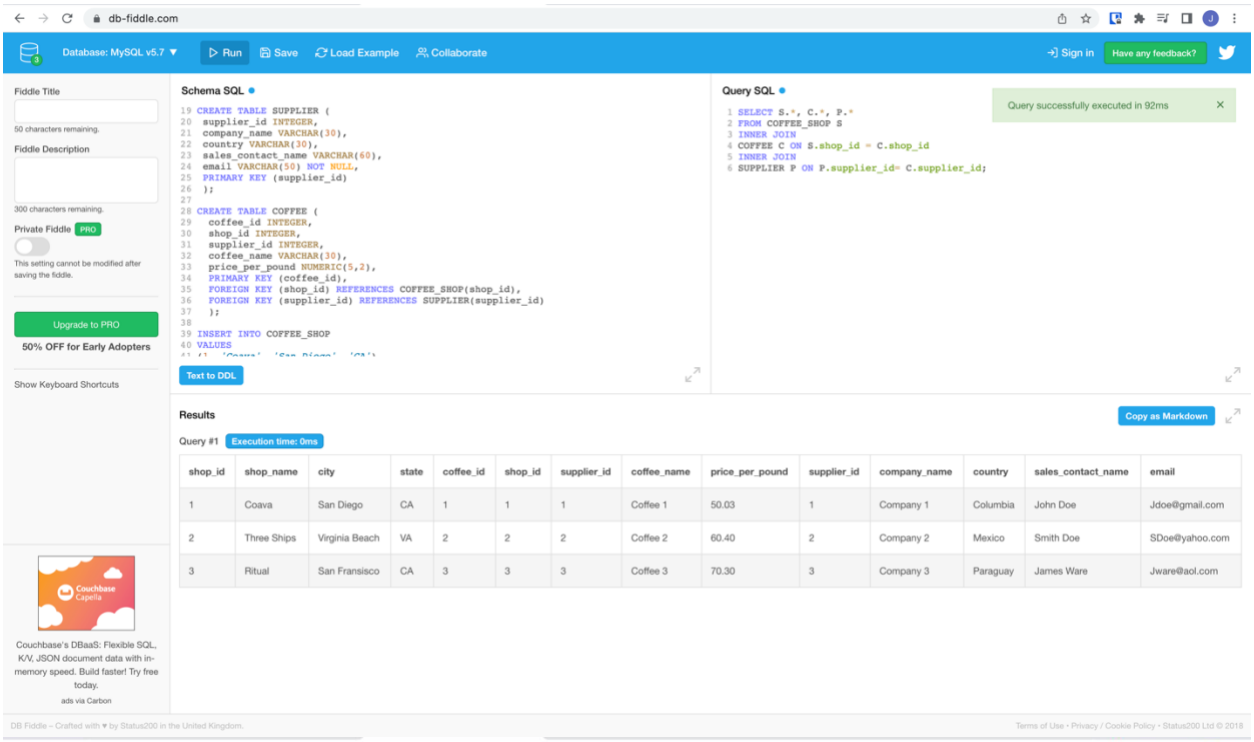
Text to DDL

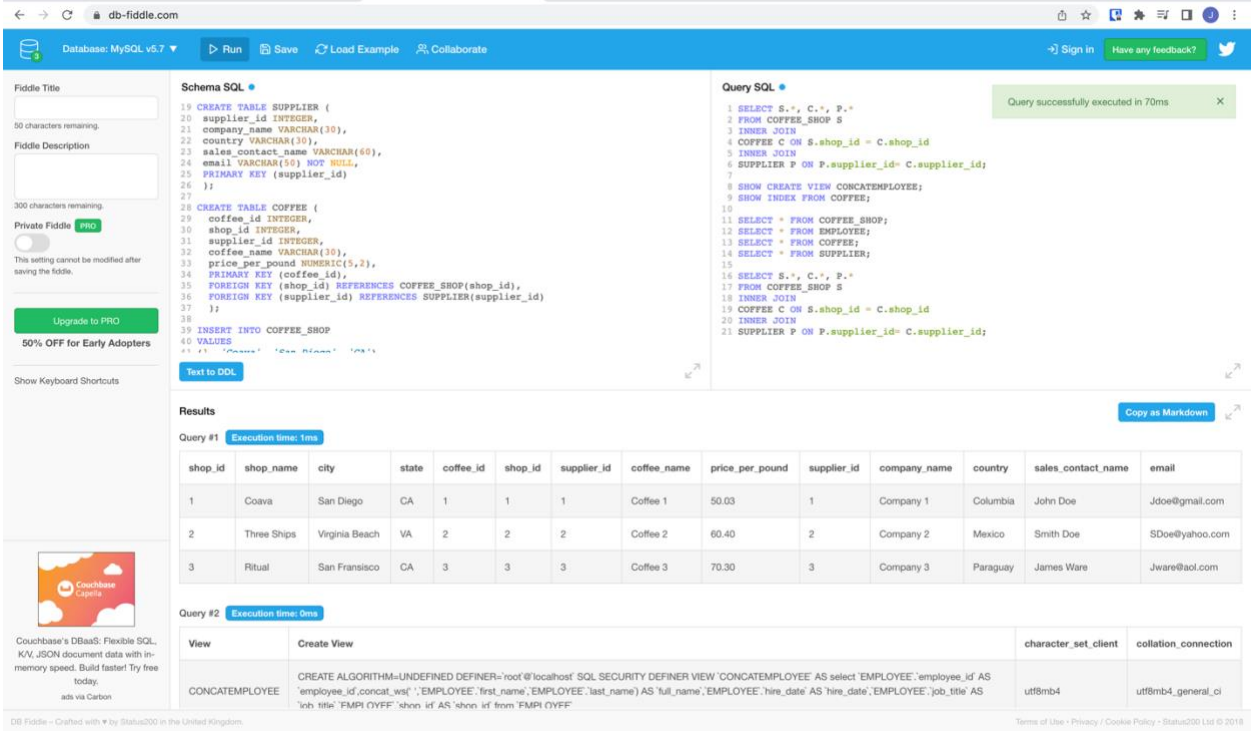

db-fiddle.com
Database: MySQL v5.7
Run
Save
Load Example
Collaborate
Sign in
Have any feedback?
Fiddle Title
50 characters remaining.
Fiddle Description
300 characters remaining.
Private Fiddle PRO
This setting cannot be modified after saving the fiddle.
Upgrade to PRO
50% OFF for Early Adopters
Show Keyboard Shortcuts
Schema SQL
CREATE TABLE COFFEE_SHOP (
shop_id INTEGER,
shop_name VARCHAR(50),
city VARCHAR(50),
state CHAR(2),
PRIMARY KEY (shop_id)
);
CREATE TABLE EMPLOYEE (
employee_id INTEGER,
first_name VARCHAR(30),
last_name VARCHAR(30),
hire_date DATE,
job_title VARCHAR(30),
PRIMARY KEY(employee_id),
shop_id INTEGER,
FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id)
);
CREATE TABLE SUPPLIER (
supplier_id INTEGER,
company_name VARCHAR(30),
country VARCHAR(30),
Text to DDL
Query SQL
SELECT * FROM COFFEE_SHOP;
SELECT * FROM EMPLOYEE;
SELECT * FROM COFFEE;
SELECT * FROM SUPPLIER;
Query successfully executed in 20ms
Results
Copy as Markdown
Query #1
Execution time: 1ms
shop_id | shop_name | city | state
1 | Coava | San Diego | CA
2 | Three Ships | Virginia Beach | VA
3 | Ritual | San Fransisco | CA
Query #2
Execution time: 0ms
employee_id | first_name | last_name | hire_date | job_title | shop_id
1 | John | Doe | 2000-07-31 | CEO | 1
mailchimp
Design beautiful emails with Mailchimp's Creative Assistant
ads via Carbon
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.
Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018


db-fiddle.com
Database: MySQL v5.7
Run
Save
Load Example
Collaborate
Sign in
Have any feedback?
Fiddle Title
50 characters remaining.
Fiddle Description
300 characters remaining.
Private Fiddle PRO
This setting cannot be modified after saving the fiddle.
Upgrade to PRO
50% OFF for Early Adopters
Show Keyboard Shortcuts
Schema SQL
shop_id INTEGER,
FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id)
);
CREATE TABLE SUPPLIER (
supplier_id INTEGER,
company_name VARCHAR(30),
country VARCHAR(30),
sales_contact_name VARCHAR(60),
email VARCHAR(50) NOT NULL,
PRIMARY KEY (supplier_id)
);
CREATE TABLE COFFEE (
coffee_id INTEGER,
shop_id INTEGER,
supplier_id INTEGER,
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)
Text to DDL
Query SQL
SHOW CREATE VIEW CONCATEMPLOYEE;
Query successfully executed in 18ms
Results
Copy as Markdown
Query #1
Execution time: 0ms
View | Create View | character_set_client | collation_connection
CONCATEMPLOYEE | CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `CONCATEMPLOYEE` AS select `EMPLOYEE`.`employee_id` AS `employee_id`,concat_ws(' ',`EMPLOYEE`.`first_name`,`EMPLOYEE`.`last_name`) AS `full_name`,`EMPLOYEE`.`hire_date` AS `hire_date`,`EMPLOYEE`.`job_title` AS `job_title`,`EMPLOYEE`.`shop_id` AS `shop_id` from `EMPLOYEE` | utf8mb4 | utf8mb4_general_ci
Sourcegraph
See how teams remediate vulnerabilities with confidence knowing they can every instance of affected code.
ads via Carbon
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.
Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018


db-fiddle.com
Database: MySQL v5.7
Run
Save
Load Example
Collaborate
Sign in
Have any feedback?
Fiddle Title
50 characters remaining.
Fiddle Description
300 characters remaining.
Private Fiddle
PRO
This setting cannot be modified after saving the fiddle.
Upgrade to PRO
50% OFF for Early Adopters
Show Keyboard Shortcuts
Schema SQL
19 CREATE TABLE SUPPLIER (
20 supplier_id INTEGER,
21 company_name VARCHAR(30),
22 country VARCHAR(30),
23 sales_contact_name VARCHAR(60),
24 email VARCHAR(50) NOT NULL,
25 PRIMARY KEY (supplier_id)
26 );
27
28 CREATE TABLE COFFEE (
29 coffee_id INTEGER,
30 shop_id INTEGER,
31 supplier_id INTEGER,
32 coffee_name VARCHAR(30),
33 price_per_pound NUMERIC(5,2),
34 PRIMARY KEY (coffee_id),
35 FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id),
36 FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
37 );
38
39 INSERT INTO COFFEE_SHOP
40 VALUES
Text to DDL
Query SQL
1 SHOW INDEX FROM COFFEE;
Query successfully executed in 70ms
Results
Copy as Markdown
Query #1
Execution time: 1ms
Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment
COFFEE | 0 | PRIMARY | 1 | coffee_id | A | 3 | null | null | | BTREE
COFFEE | 1 | shop_id | 1 | shop_id | A | 3 | null | null | YES | BTREE
COFFEE | 1 | supplier_id | 1 | supplier_id | A | 3 | null | null | YES | BTREE
COFFEE | 1 | COFFEENAME | 1 | coffee_name | A | 3 | null | null | YES | BTREE
Couchbase Capella
Couchbase's DBaaS: Flexible SQL, K/V, JSON document data with in-memory speed. Build faster! Try free today.
ads via Carbon
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.
Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018


db-fiddle.com
Database: MySQL v5.7
Run
Save
Load Example
Collaborate
Sign in
Have any feedback?
Fiddle Title
50 characters remaining.
Fiddle Description
300 characters remaining.
Private Fiddle
PRO
This setting cannot be modified after saving the fiddle.
Upgrade to PRO
50% OFF for Early Adopters
Show Keyboard Shortcuts
Schema SQL
19 CREATE TABLE SUPPLIER (
20 supplier_id INTEGER,
21 company_name VARCHAR(30),
22 country VARCHAR(30),
23 sales_contact_name VARCHAR(60),
24 email VARCHAR(50) NOT NULL,
25 PRIMARY KEY (supplier_id)
26 );
27
28 CREATE TABLE COFFEE (
29 coffee_id INTEGER,
30 shop_id INTEGER,
31 supplier_id INTEGER,
32 coffee_name VARCHAR(30),
33 price_per_pound NUMERIC(5,2),
34 PRIMARY KEY (coffee_id),
35 FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id),
36 FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
37 );
38
39 INSERT INTO COFFEE_SHOP
40 VALUES
Text to DDL
Query SQL
1 SELECT coffee_name FROM COFFEE WHERE shop_id
Query successfully executed in 72ms
Results
Copy as Markdown
Query #1
Execution time: 0ms
coffee_name
Coffee 1
Couchbase Capella
Couchbase's DBaaS: Flexible SQL, K/V, JSON document data with in-memory speed. Build faster! Try free today.
ads via Carbon
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.
Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018


db-fiddle.com
Database: MySQL v5.7
Run
Save
Load Example
Collaborate
Sign in
Have any feedback?
Fiddle Title
50 characters remaining.
Fiddle Description
300 characters remaining.
Private Fiddle
PRO
This setting cannot be modified after saving the fiddle.
Upgrade to PRO
50% OFF for Early Adopters
Show Keyboard Shortcuts
Schema SQL
19 CREATE TABLE SUPPLIER (
20 supplier_id INTEGER,
21 company_name VARCHAR(30),
22 country VARCHAR(30),
23 sales_contact_name VARCHAR(60),
24 email VARCHAR(50) NOT NULL,
25 PRIMARY KEY (supplier_id)
26 );
27
28 CREATE TABLE COFFEE (
29 coffee_id INTEGER,
30 shop_id INTEGER,
31 supplier_id INTEGER,
32 coffee_name VARCHAR(30),
33 price_per_pound NUMERIC(5,2),
34 PRIMARY KEY (coffee_id),
35 FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id),
36 FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
37 );
38
39 INSERT INTO COFFEE_SHOP
40 VALUES
Text to DDL
Query SQL
1 SELECT S.*, C.*, P.*
2 FROM COFFEE_SHOP S
3 INNER JOIN
4 COFFEE C ON S.shop_id = C.shop_id
5 INNER JOIN
6 SUPPLIER P ON P.supplier_id= C.supplier_id;
Query successfully executed in 92ms
Results
Copy as Markdown
Query #1
Execution time: 0ms
shop_id | shop_name | city | state | coffee_id | shop_id | supplier_id | coffee_name | price_per_pound | supplier_id | company_name | country | sales_contact_name | email
1 | Coava | San Diego | CA | 1 | 1 | 1 | Coffee 1 | 50.03 | 1 | Company 1 | Columbia | John Doe | Jdoe@gmail.com
2 | Three Ships | Virginia Beach | VA | 2 | 2 | 2 | Coffee 2 | 60.40 | 2 | Company 2 | Mexico | Smith Doe | SDoe@yahoo.com
3 | Ritual | San Fransisco | CA | 3 | 3 | 3 | Coffee 3 | 70.30 | 3 | Company 3 | Paraguay | James Ware | Jware@aol.com
Couchbase Capella
Couchbase's DBaaS: Flexible SQL, K/V, JSON document data with in-memory speed. Build faster! Try free today.
ads via Carbon
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.
Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018


db-fiddle.com
Database: MySQL v5.7
Run
Save
Load Example
Collaborate
Sign in
Have any feedback?
Fiddle Title
50 characters remaining.
Fiddle Description
300 characters remaining.
Private Fiddle
PRO
This setting cannot be modified after saving the fiddle.
Upgrade to PRO
50% OFF for Early Adopters
Show Keyboard Shortcuts
Schema SQL
19 CREATE TABLE SUPPLIER (
20 supplier_id INTEGER,
21 company_name VARCHAR(30),
22 country VARCHAR(30),
23 sales_contact_name VARCHAR(60),
24 email VARCHAR(50) NOT NULL,
25 PRIMARY KEY (supplier_id)
26 );
27
28 CREATE TABLE COFFEE (
29 coffee_id INTEGER,
30 shop_id INTEGER,
31 supplier_id INTEGER,
32 coffee_name VARCHAR(30),
33 price_per_pound NUMERIC(5,2),
34 PRIMARY KEY (coffee_id),
35 FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id),
36 FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)
37 );
38
39 INSERT INTO COFFEE_SHOP
40 VALUES
Text to DDL
Query SQL
1 SELECT S.*, C.*, P.*
2 FROM COFFEE_SHOP S
3 INNER JOIN
4 COFFEE C ON S.shop_id = C.shop_id
5 INNER JOIN
6 SUPPLIER P ON P.supplier_id= C.supplier_id;
7
8 SHOW CREATE VIEW CONCATEMPLOYEE;
9 SHOW INDEX FROM COFFEE;
10
11 SELECT * FROM COFFEE_SHOP;
12 SELECT * FROM EMPLOYEE;
13 SELECT * FROM COFFEE;
14 SELECT * FROM SUPPLIER;
15
16 SELECT S.*, C.*, P.*
17 FROM COFFEE_SHOP S
18 INNER JOIN
19 COFFEE C ON S.shop_id = C.shop_id
20 INNER JOIN
21 SUPPLIER P ON P.supplier_id= C.supplier_id;
Query successfully executed in 70ms
Results
Copy as Markdown
Query #1
Execution time: 1ms
shop_id | shop_name | city | state | coffee_id | shop_id | supplier_id | coffee_name | price_per_pound | supplier_id | company_name | country | sales_contact_name | email
1 | Coava | San Diego | CA | 1 | 1 | 1 | Coffee 1 | 50.03 | 1 | Company 1 | Columbia | John Doe | Jdoe@gmail.com
2 | Three Ships | Virginia Beach | VA | 2 | 2 | 2 | Coffee 2 | 60.40 | 2 | Company 2 | Mexico | Smith Doe | SDoe@yahoo.com
3 | Ritual | San Fransisco | CA | 3 | 3 | 3 | Coffee 3 | 70.30 | 3 | Company 3 | Paraguay | James Ware | Jware@aol.com
Query #2
Execution time: 0ms
View | Create View | character_set_client | collation_connection
CONCATEMPLOYEE | CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `CONCATEMPLOYEE` AS select `EMPLOYEE`.`employee_id` AS `employee_id`,concat_ws(' ',`EMPLOYEE`.`first_name`,`EMPLOYEE`.`last_name`) AS `full_name`,`EMPLOYEE`.`hire_date` AS `hire_date`,`EMPLOYEE`.`job_title` AS `job_title` FMPLOYEE`.`shop_id` AS `shop_id` from `EMPLOYEE` | utf8mb4 | utf8mb4_general_ci
Couchbase Capella
Couchbase's DBaaS: Flexible SQL, K/V, JSON document data with in-memory speed. Build faster! Try free today.
ads via Carbon
DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom.
Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

# SQL Code:

```
CREATE TABLE COFFEE_SHOP (
 shop_id INTEGER,
 shop_name VARCHAR(50),
 city VARCHAR(50),
 state CHAR(2),
 PRIMARY KEY (shop_id)
 );
CREATE TABLE EMPLOYEE (
 employee_id INTEGER,
 first_name VARCHAR(30),
 last_name VARCHAR(30),
 hire_date DATE,
 job_title VARCHAR(30),
 PRIMARY KEY(employee_id),
 shop_id INTEGER,
 FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id)
 );

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

CREATE TABLE COFFEE (
  coffee_id INTEGER,
  shop_id INTEGER,
  supplier_id INTEGER,
  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)
  );

INSERT INTO COFFEE_SHOP
VALUES
(1, 'Coava', 'San Diego', 'CA'),
(2, 'Three Ships', 'Virginia Beach', 'VA'),
```

```
(3, 'Ritual', 'San Fransisco', 'CA');

INSERT INTO EMPLOYEE
VALUES
(1, 'John', 'Doe', '2000-07-31', 'CEO', 1),
(2, 'Smith', 'Doe', '1998-07-22', 'CTO', 2),
(3, 'James', 'Ware', '2000-06-05', 'CFO', 3);

INSERT INTO SUPPLIER
VALUES
(1, 'Company 1', 'Columbia', 'John Doe', 'Jdoe@gmail.com'),
(2, 'Company 2', 'Mexico', 'Smith Doe', 'SDoe@yahoo.com'),
(3, 'Company 3', 'Paraguay', 'James Ware', 'Jware@aol.com');

INSERT INTO COFFEE
VALUES
(1, 1, 1, 'Coffee 1', 50.03),
(2, 2, 2, 'Coffee 2', 60.40),
(3, 3, 3, 'Coffee 3', 70.30);

CREATE VIEW CONCATEMPLOYEE AS
SELECT employee_id, CONCAT_WS(' ', first_name, last_name) AS full_name, hire_date,
job_title, shop_id
FROM EMPLOYEE;

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

# SQL Queries:

```
SELECT S.*, C.*, P.*
FROM COFFEE_SHOP S
INNER JOIN
COFFEE C ON S.shop_id = C.shop_id
INNER JOIN
SUPPLIER P ON P.supplier_id= C.supplier_id;

SHOW CREATE VIEW CONCATEMPLOYEE;
SHOW INDEX FROM COFFEE;

SELECT * FROM COFFEE_SHOP;
SELECT * FROM EMPLOYEE;
SELECT * FROM COFFEE;
SELECT * FROM SUPPLIER;
```

```
SELECT S.*, C.*, P.*
FROM COFFEE_SHOP S
INNER JOIN
COFFEE C ON S.shop_id = C.shop_id
INNER JOIN
SUPPLIER P ON P.supplier_id= C.supplier_id;
```