

PART A

Nora's Bagel Bin Database Blueprints

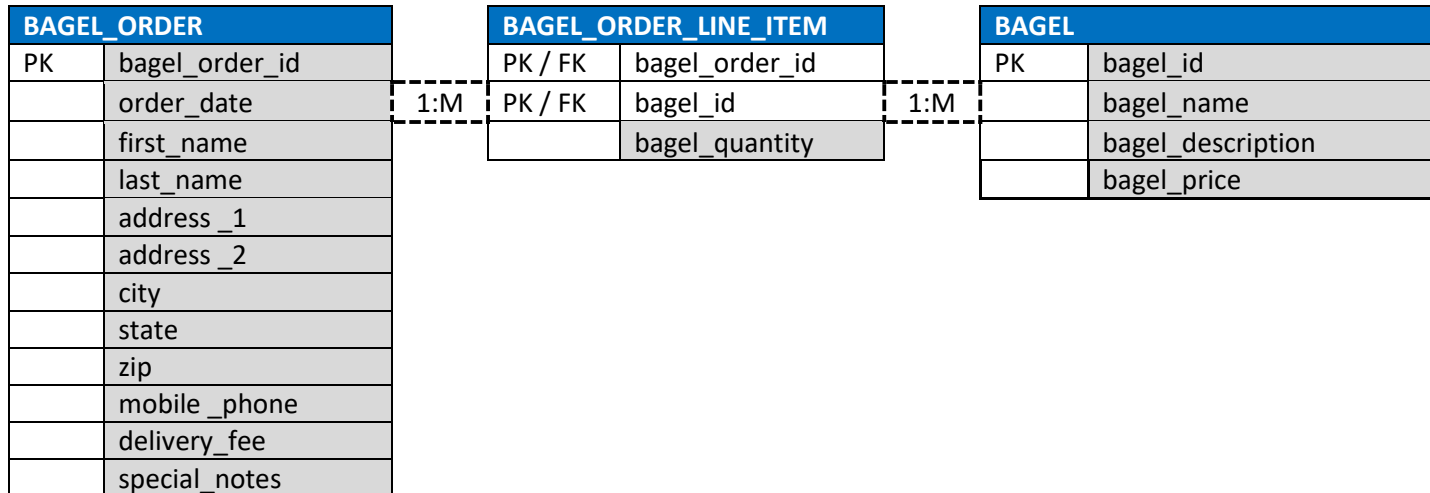
First Normal Form (1NF)

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

Having been provided the 1NF of Nora's bagel ordering process there exists a composite key indicating multiple entities in the table. Because of this the data is functionally dependent on multiple entities. It is necessary to perform normalization of the table above to eliminate data redundancy and improve functionality.

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

Second Normal Form (2NF)



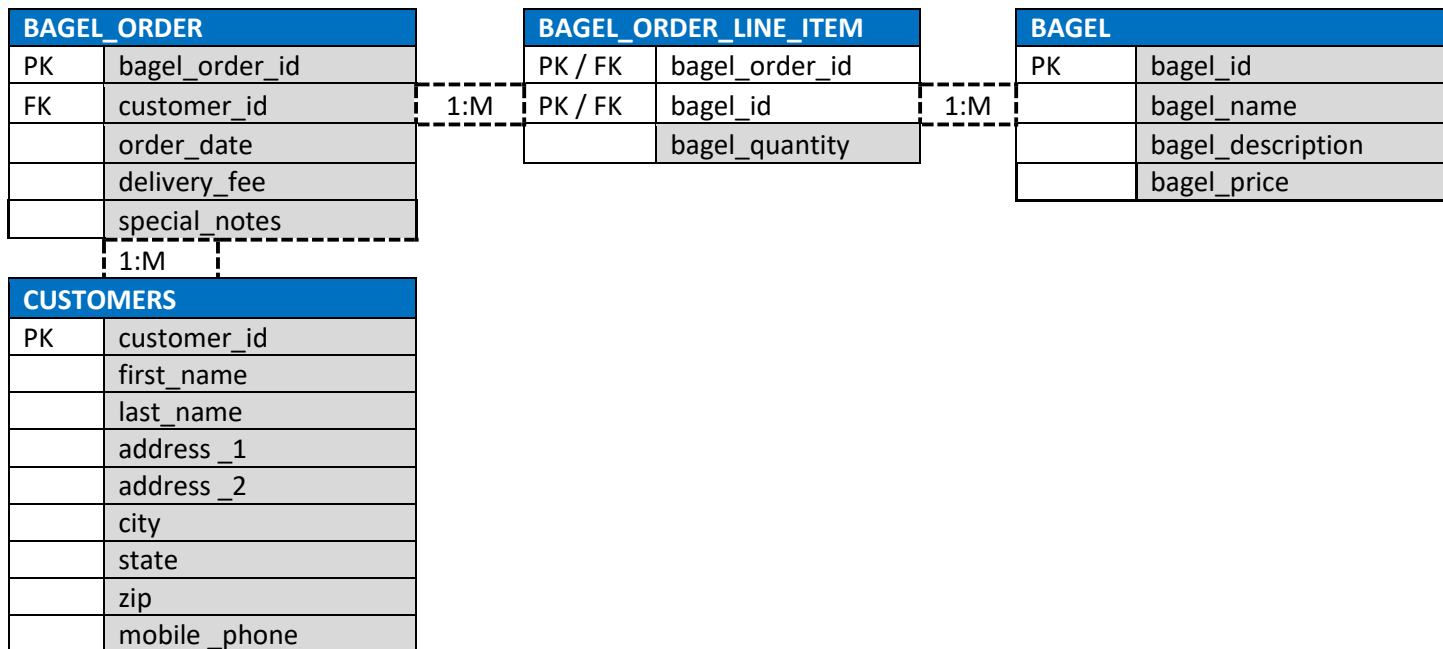
Achieving second normal form the initial entity has been separated into three distinct entities BAGEL_ORDER, BAGEL_ORDER_LINE_ITEM, and BAGEL. The BAGEL_ORDER entity containing the bagel_order_id primary key contains information that will be most likely be unique to each order submitted of the customer's information with bagel_order_id, order_date, etc. The BAGEL entity containing the bagel_id primary key stores information relating to the products offered by Nora. As the information of the products offered will not change or have very little chance to change from order to order it was best to separate it into its own. The BAGEL_ORDER_LINE_ITEM table acts as a bridge between the BAGEL_ORDER and BAGEL entities being the associate table that links the inherent many-to-many relationship between the two. The bagel_quantity field exists within the associate table so that the quantity is dependent on each new order, while also maintaining the association to the BAGEL entity it is related to.

The relationships between each entity were determined by:

- Each BAGEL_ORDER will have many BAGEL_ORDER_LINE_ITEMS, but each is associated with one specific order relying on the bagel_order_id establishing the 1:M relationship between BAGEL_ORDER and BAGEL_ORDER_LINE_ITEM.
- Similarly, though each BAGEL_ORDER_LINE_ITEM may be associated with many different bagel products, the bagel_id and bagel_quantity ensure that each instance of a line item will be independent of the bagels products available.

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

Third Normal Form (3NF)

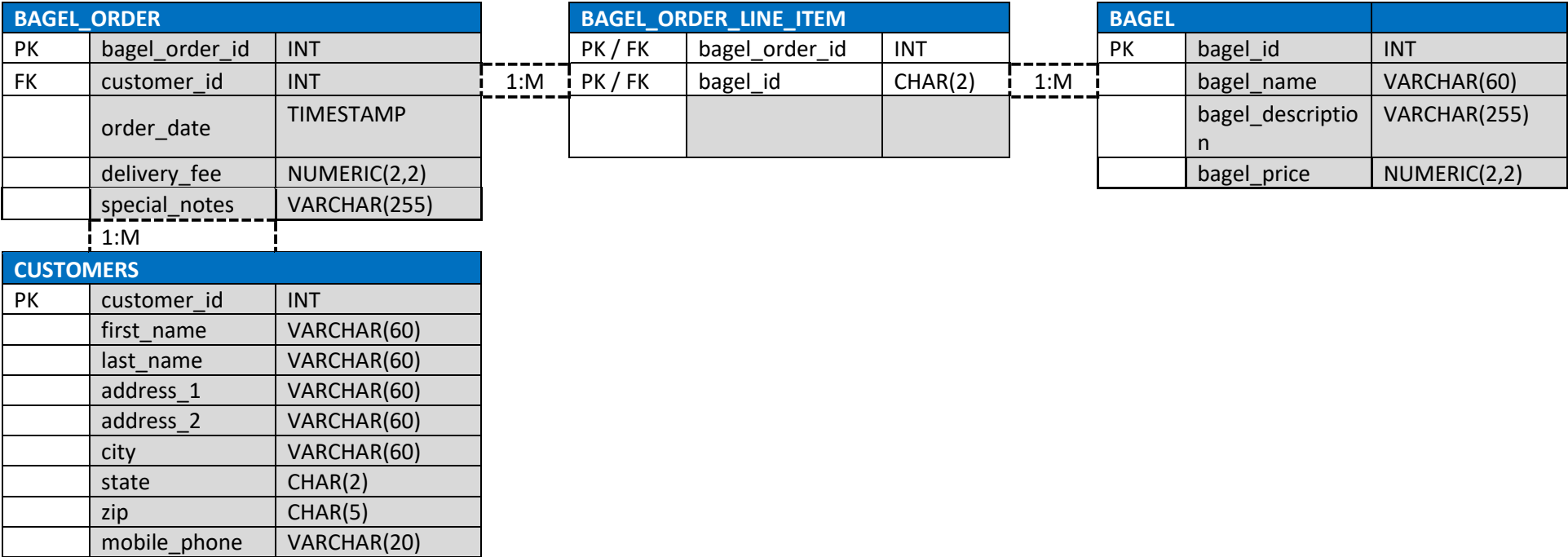


Achieving third normal form the entities in the second normal form still exist with the addition of the BAGEL_ORDER entity being further broken down to separate customer information into its own CUSTOMERS entity. One customer can place

many different orders, but each order is associated with only one customer. Creating the CUSTOMERS entity allows for it to be independent of every order, reducing data redundancy from duplicate customer information in BAGEL_ORDER for every new order of the same customer with the second normal form database previously.

Nora’s Bagel Bin Database Blueprints *(continued)*

Final Physical Database Model



The final database model reflects the changes made in the third normal form with the appropriate datatypes assigned.

PART B

Jaunty Coffee Co. ERD

1. Develop SQL code to create *each* table as specified in the attached “Jaunty Coffee Co. ERD”

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view of the database structure, including tables like 'coffee', 'coffee_shop', 'employee', and 'supplier'. The main editor window shows the following SQL code:

```
1 CREATE TABLE COFFEE_SHOP (  
2     shop_id INT NOT NULL AUTO_INCREMENT, -- PK  
3     shop_name VARCHAR(50) NOT NULL,  
4     city VARCHAR(50) NOT NULL,  
5     state CHAR(2) NOT NULL,  
6     PRIMARY KEY (shop_id)  
7 );  
8  
9 CREATE TABLE SUPPLIER (  
10     supplier_id INT NOT NULL, -- PK  
11     company_name VARCHAR(50) NOT NULL,  
12     country VARCHAR(30) NOT NULL,  
13     sales_contact_name VARCHAR(60),  
14     email VARCHAR(50) NOT NULL,  
15     PRIMARY KEY (supplier_id)  
16 );  
17  
18 CREATE TABLE EMPLOYEE (  
19     employee_id INT NOT NULL, -- PK  
20     first_name VARCHAR(30) NOT NULL,  
21     last_name VARCHAR(30) NOT NULL,  
22     hire_date DATE NOT NULL,  
23     job_title VARCHAR(30) NOT NULL,  
24     shop_id INT, -- FK  
25     KEY fk_employee_shop_id(shop_id),  
26     CONSTRAINT fk_employee_shop_id FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id),  
27     PRIMARY KEY (employee_id)  
28 );  
29  
30 CREATE TABLE COFFEE (  
31     coffee_id INT NOT NULL, -- PK  
32     shop_id INT, -- FK  
33     supplier_id INT, -- FK  
34     coffee_name VARCHAR(30) NOT NULL,  
35     price_per_pound NUMERIC(5,2) NOT NULL,  
36     KEY fk_coffee_shop_id(shop_id),  
37     KEY fk_supplier_id(supplier_id),  
38     CONSTRAINT fk_coffee_shop_id FOREIGN KEY (shop_id) REFERENCES COFFEE_SHOP(shop_id),  
39     CONSTRAINT fk_supplier_id FOREIGN KEY (supplier_id) REFERENCES SUPPLIER(supplier_id)  
40 );  
41
```

MySQL Workbench

Local instance MySQL80

FileEditViewQueryDatabaseServerToolsScriptingHelp

create-databasesinvoicescreate-jaunty-coffee-cocoffeecoffee_shopemployeesupplier

Limit to 1000 rows

1 •

SELECT * FROM `jaunty_coffee_co`.`coffee_shop`;

Result Grid

Filter Rows:

Edit:Export/Import:Wrap Cell Content:

	shop_id	shop_name	city	state
*	NULL	NULL	NULL	NULL

coffee_shop 1

ApplyRevert

Activate Windows

Go to Settings to activate Windows.

Desktop

12:52 PM3/25/2022

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co coffee coffee_shop employee supplier

Limit to 1000 rows

```
1 • SELECT * FROM `jaunty_coffee_co`.`supplier`;
```

Result Grid

supplier_id	company_name	country	sales_contact_name	email
NULL	NULL	NULL	NULL	NULL

supplier 1

Apply Revert

Activate Windows
Go to Settings to activate Windows.

Desktop 12:52 PM 3/25/2022

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co coffee coffee_shop employee supplier

Limit to 1000 rows

```
1 • SELECT * FROM `jaunty_coffee_co`.`employee`;
```

Result Grid

	employee_id	first_name	last_name	hire_date	job_title	shop_id
*	NULL	NULL	NULL	NULL	NULL	NULL

Activate Windows
Go to Settings to activate Windows.

employee 1 x Apply Revert

Desktop 12:52 PM 3/25/2022

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co **coffee** coffee_shop employee supplier

Limit to 1000 rows

```
1 • SELECT * FROM `jaunty_coffee_co`.`coffee`;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

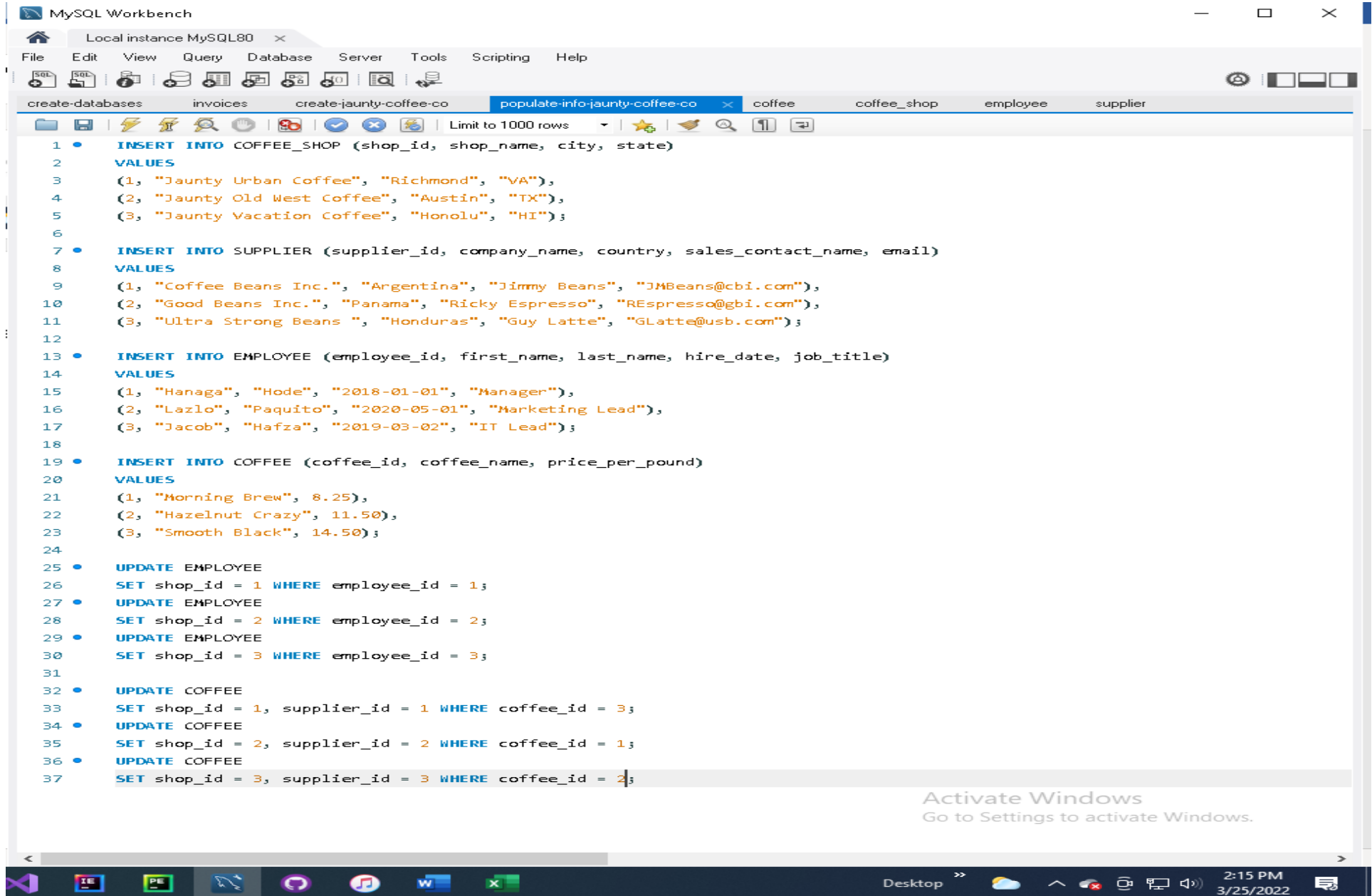
coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
-----------	---------	-------------	-------------	-----------------

Activate Windows
Go to Settings to activate Windows.

coffee 1 x Read Only

Desktop 12:52 PM 3/25/2022

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



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co populate-info-jaunty-coffee-co coffee coffee_shop employee supplier

Limit to 1000 rows

```
1 • INSERT INTO COFFEE_SHOP (shop_id, shop_name, city, state)
2 VALUES
3 (1, "Jaunty Urban Coffee", "Richmond", "VA"),
4 (2, "Jaunty Old West Coffee", "Austin", "TX"),
5 (3, "Jaunty Vacation Coffee", "Honolu", "HI");
6
7 • INSERT INTO SUPPLIER (supplier_id, company_name, country, sales_contact_name, email)
8 VALUES
9 (1, "Coffee Beans Inc.", "Argentina", "Jimmy Beans", "JMBeans@cbi.com"),
10 (2, "Good Beans Inc.", "Panama", "Ricky Espresso", "REspresso@gbi.com"),
11 (3, "Ultra Strong Beans ", "Honduras", "Guy Latte", "GLatte@usb.com");
12
13 • INSERT INTO EMPLOYEE (employee_id, first_name, last_name, hire_date, job_title)
14 VALUES
15 (1, "Hanaga", "Hode", "2018-01-01", "Manager"),
16 (2, "Lazlo", "Paquito", "2020-05-01", "Marketing Lead"),
17 (3, "Jacob", "Hafza", "2019-03-02", "IT Lead");
18
19 • INSERT INTO COFFEE (coffee_id, coffee_name, price_per_pound)
20 VALUES
21 (1, "Morning Brew", 8.25),
22 (2, "Hazelnut Crazy", 11.50),
23 (3, "Smooth Black", 14.50);
24
25 • UPDATE EMPLOYEE
26 SET shop_id = 1 WHERE employee_id = 1;
27 • UPDATE EMPLOYEE
28 SET shop_id = 2 WHERE employee_id = 2;
29 • UPDATE EMPLOYEE
30 SET shop_id = 3 WHERE employee_id = 3;
31
32 • UPDATE COFFEE
33 SET shop_id = 1, supplier_id = 1 WHERE coffee_id = 3;
34 • UPDATE COFFEE
35 SET shop_id = 2, supplier_id = 2 WHERE coffee_id = 1;
36 • UPDATE COFFEE
37 SET shop_id = 3, supplier_id = 3 WHERE coffee_id = 2;
```

Activate Windows
Go to Settings to activate Windows.

Desktop 2:15 PM 3/25/2022

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co populate-info-jaunty-coffee-co coffee coffee_shop employee supplier

Limit to 1000 rows

```
1 • SELECT * FROM `jaunty_coffee_co`.`coffee_shop`;
```

Result Grid

	shop_id	shop_name	city	state
▶	1	Jaunty Urban Coffee	Richmond	VA
	2	Jaunty Old West Coffee	Austin	TX
	3	Jaunty Vacation Coffee	Honolu	HI
*	NULL	NULL	NULL	NULL

Activate Windows
Go to Settings to activate Windows.

coffee_shop 1 x Apply Revert

Desktop 2:18 PM 3/25/2022

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co populate-info-jaunty-coffee-co coffee coffee_shop employee supplier

Limit to 1000 rows

```
1 • SELECT * FROM `jaunty_coffee_co`.`supplier`;
```

Result Grid

	supplier_id	company_name	country	sales_contact_name	email
▶	1	Coffee Beans Inc.	Argentina	Jimmy Beans	JMBeans@cbi.com
	2	Good Beans Inc.	Panama	Ricky Espresso	REspresso@gbi.com
	3	Ultra Strong Beans	Honduras	Guy Latte	GLatte@usb.com
*	NULL	NULL	NULL	NULL	NULL

Activate Windows
Go to Settings to activate Windows.

supplier 1 x Apply Revert

Desktop 2:19 PM 3/25/2022

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co populate-info-jaunty-coffee-co coffee coffee_shop employee x supplier

Limit to 1000 rows

```
1 • SELECT * FROM `jaunty_coffee_co`.`employee`;
```

Result Grid

	employee_id	first_name	last_name	hire_date	job_title	shop_id
▶	1	Hanaga	Hode	2018-01...	Manager	1
	2	Lazlo	Paquito	2020-05...	Marketi...	2
	3	Jacob	Hafza	2019-03...	IT Lead	3
*	NULL	NULL	NULL	NULL	NULL	NULL

Activate Windows
Go to Settings to activate Windows.

employee 1 x Apply Revert

Desktop 2:20 PM 3/25/2022

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co populate-info-jaunty-coffee-co **coffee** coffee_shop employee supplier

Limit to 1000 rows

```
1 • SELECT * FROM `jaunty_coffee_co`.`coffee`;
```

Result Grid

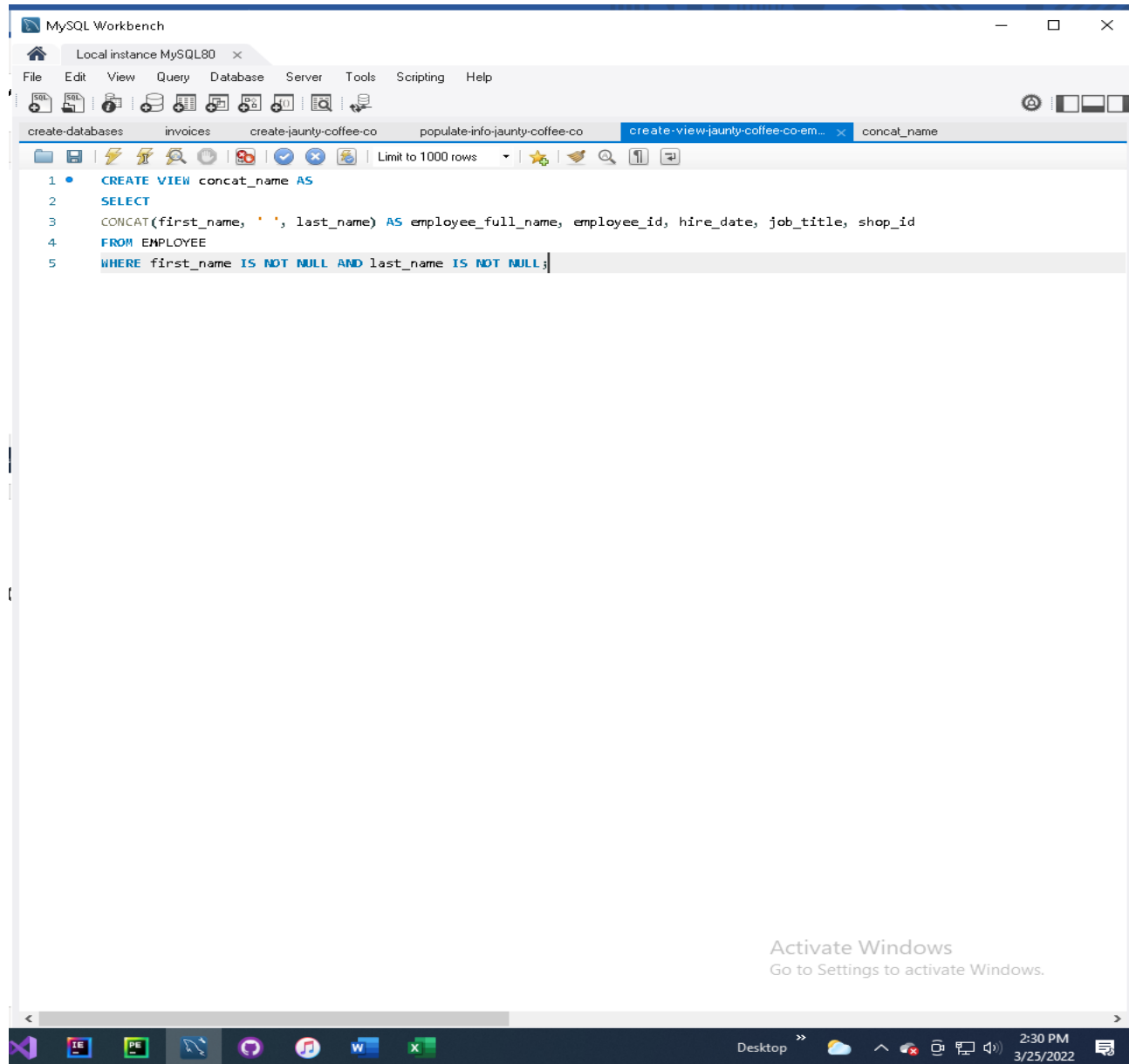
	coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
▶	1	2	2	Morning Brew	8.25
	2	3	3	Hazelnut Cr...	11.50
	3	1	1	Smooth Black	14.50

Activate Windows
Go to Settings to activate Windows.

coffee 1 x Read Only

Desktop 2:21 PM 3/25/2022

3. Develop SQL code to create a view from EMPLOYEE table, with new employee_full_name attribute



The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. Below the menu bar is a toolbar with various icons. The main window displays a SQL script in a text editor. The script is as follows:

```
1 • CREATE VIEW concat_name AS
2 SELECT
3 CONCAT(first_name, ' ', last_name) AS employee_full_name, employee_id, hire_date, job_title, shop_id
4 FROM EMPLOYEE
5 WHERE first_name IS NOT NULL AND last_name IS NOT NULL;
```

The script is numbered 1 through 5. The first line is a comment. The second line is the CREATE VIEW statement. The third line is the SELECT statement. The fourth line is the FROM clause. The fifth line is the WHERE clause. The script is executed in the context of a database named 'jaunty-coffee-co'.

At the bottom of the window, there is a taskbar with icons for various applications, including a web browser, a file explorer, and a music player. The system clock in the bottom right corner shows the time as 2:30 PM on 3/25/2022.

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-databases invoices create-jaunty-coffee-co populate-info-jaunty-coffee-co create-view-jaunty-coffee-co-em... concat_name

Limit to 1000 rows

```
1 • SELECT * FROM `jaunty_coffee_co`.`concat_name`;
```

Result Grid

	employee_full_name	employee_id	hire_date	job_title	shop_id
▶	Hanaga Hode	1	2018-01-01	Manager	1
	Lazlo Paquito	2	2020-05-01	Marketing Lead	2
	Jacob Hafza	3	2019-03-02	IT Lead	3

Activate Windows
Go to Settings to activate Windows.

concat_name 1 x Read Only

Desktop 2:32 PM 3/25/2022

4. Develop SQL code to create an index on the coffee_name field from the COFFEE table

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'jaunty_coffee_co.' database selected. The main editor window shows the following SQL code:

```
1 • CREATE INDEX coffee_index
2   ON COFFEE (coffee_name);
3
4 • SELECT coffee_name FROM COFFEE;
```

Below the code editor, the 'Result Grid' shows the output of the SELECT query:

coffee_name
Hazelnut Crazy
Morning Brew
Smooth Black

The bottom panel shows the 'Output' window with the 'Action Output' tab selected. It displays a log of database actions:

#	Time	Action	Message	Duration / Fetch
126	14:39:07	UPDATE EMPLOYEE SET shop_id = 3 WHERE ...	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.000 sec
127	14:39:07	UPDATE COFFEE SET shop_id = 1, supplier_id = ...	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.000 sec
128	14:39:07	UPDATE COFFEE SET shop_id = 2, supplier_id = ...	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.000 sec
129	14:39:07	UPDATE COFFEE SET shop_id = 3, supplier_id = ...	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.000 sec
130	14:39:10	CREATE INDEX coffee_index ON COFFEE (coffee_name);	0 row(s) affected Records: 0 Duplicates: 0 Warni...	0.031 sec
131	14:39:10	SELECT coffee_name FROM COFFEE LIMIT 0, 1...	3 row(s) returned	0.000 sec / 0.000 sec

5. Develop SQL code to create an SFW (SELECT-FROM-WHERE) query for any of your tables or views

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view of databases, with 'jaunty_coffee_co.' expanded to show tables like 'coffee', 'coffee_shop', 'employee', and 'supplier'. The main editor window contains the following SQL query:

```
1 SELECT *
2 FROM COFFEE
3 WHERE price_per_pound >= 10.00;
```

Below the query editor, the 'Result Grid' shows the results of the query:

	coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
2	3	3		Hazelnut Crazy	11.50
3	1	1		Smooth Black	14.50

At the bottom, the 'Output' pane shows the 'Action Output' log, which includes the following entries:

#	Time	Action	Message	Duration / Fetch
128	14:39:07	UPDATE COFFEE SET shop_id = 2, supplier_id = ...	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.000 sec
129	14:39:07	UPDATE COFFEE SET shop_id = 3, supplier_id = ...	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.000 sec
130	14:39:10	CREATE INDEX coffee_index ON COFFEE (coffee...	0 row(s) affected Records: 0 Duplicates: 0 Warni...	0.031 sec
131	14:39:10	SELECT coffee_name FROM COFFEE LIMIT 0, 1...	3 row(s) returned	0.000 sec / 0.000 sec
132	14:42:32	SELECT * FROM 'jaunty_coffee_co.'.coffee LIM...	3 row(s) returned	0.000 sec / 0.000 sec
133	14:42:52	SELECT * FROM COFFEE WHERE price_per_po...	2 row(s) returned	0.000 sec / 0.000 sec

6. Develop SQL code to create a query by joining three different tables, including attributes from all three tables

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

create-jaunty-coffee-co populate-info-jaunty-coffee-co create-index-coffee_name create-join-all-three-tables

Limit to 1000 rows

```
1 SELECT *
2 FROM COFFEE_SHOP
3 JOIN EMPLOYEE
4 ON EMPLOYEE.shop_id = COFFEE_SHOP.shop_id
5 JOIN COFFEE
6 ON COFFEE.shop_id = COFFEE_SHOP.shop_id
```

Result Grid

	shop_id	shop_name	city	state	employee_id	first_name	last_name	hire_date	job_title	shop_id	coffee_id	shop_id	supplier_id	coffee_name	price_per_pound
▶	2	Jaunty Old West Coffee	Austin	TX	2	Lazlo	Paquito	2020-05-01	Marketing Lead	2	1	2	2	Morning Brew	8.25
	3	Jaunty Vacation Coffee	Honolu	HI	3	Jacob	Hafza	2019-03-02	IT Lead	3	2	3	3	Hazelnut Crazy	11.50
	1	Jaunty Urban Coffee	Richmond	VA	1	Hanaga	Hode	2018-01-01	Manager	1	3	1	1	Smooth Black	14.50

Result 1

Output

Action Output

#	Time	Action	Message
✓ 132	14:42:32	SELECT * FROM `jaunty_coffee_co`.`coffee` LIMIT 0, 1000	3 row(s) returned
✓ 133	14:42:52	SELECT * FROM COFFEE WHERE price_per_pound >= 10.00 LIMIT 0, 1000	2 row(s) returned
✓ 134	14:47:34	SELECT * FROM COFFEE_SHOP JOIN EMPLOYEE ON EMPLOYEE.shop_id = COFFEE_SHOP.shop_id JOIN COFFEE ON COFFEE.shop_id = COFFEE_SHOP.shop_id LI...	3 row(s) returned
✓ 135	14:48:09	SELECT * FROM `jaunty_coffee_co`.`coffee_shop` LIMIT 0, 1000	3 row(s) returned
✓ 136	14:48:24	SELECT * FROM `jaunty_coffee_co`.`employee` LIMIT 0, 1000	3 row(s) returned
✓ 137	14:48:57	SELECT * FROM `jaunty_coffee_co`.`coffee` LIMIT 0, 1000	3 row(s) returned