

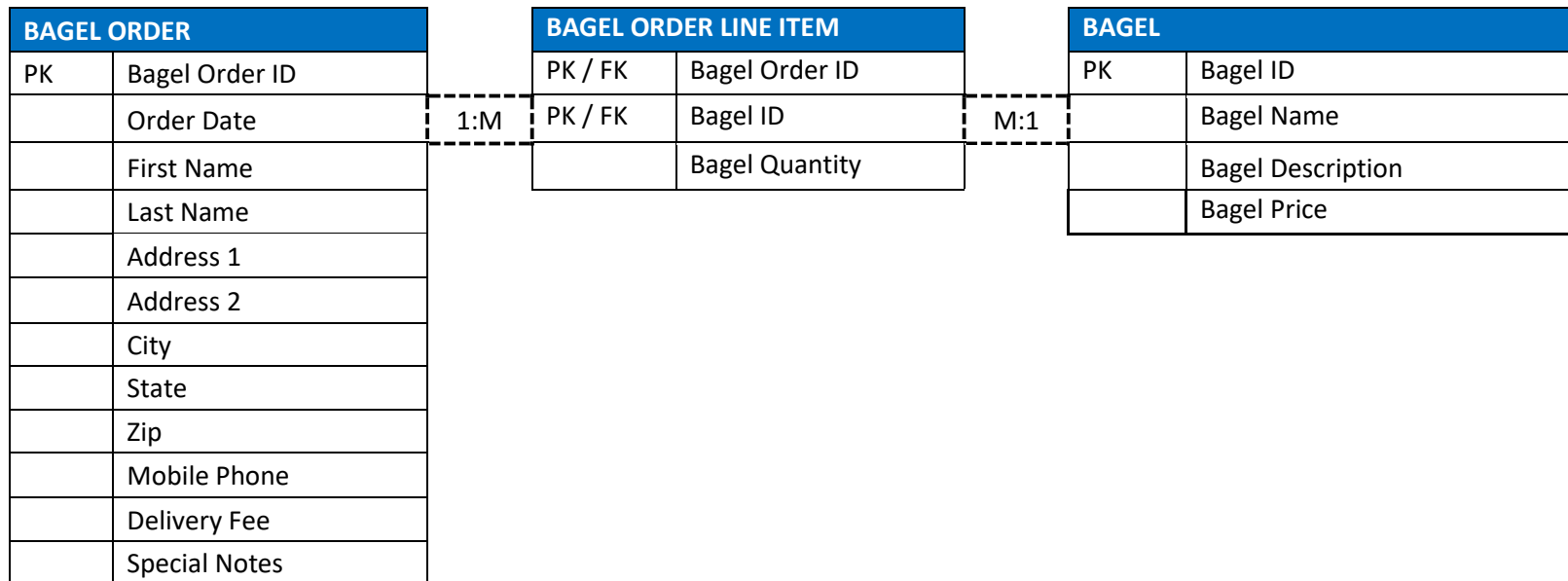
## Nora's Bagel Bin Database ER Diagram and Normalization

### 1<sup>st</sup> Normal Form

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

Using the Bagel Order Form Template, a 1<sup>st</sup> Normal form was created for the relation (table), grouping repeated attributes and maintaining single values in each cell.

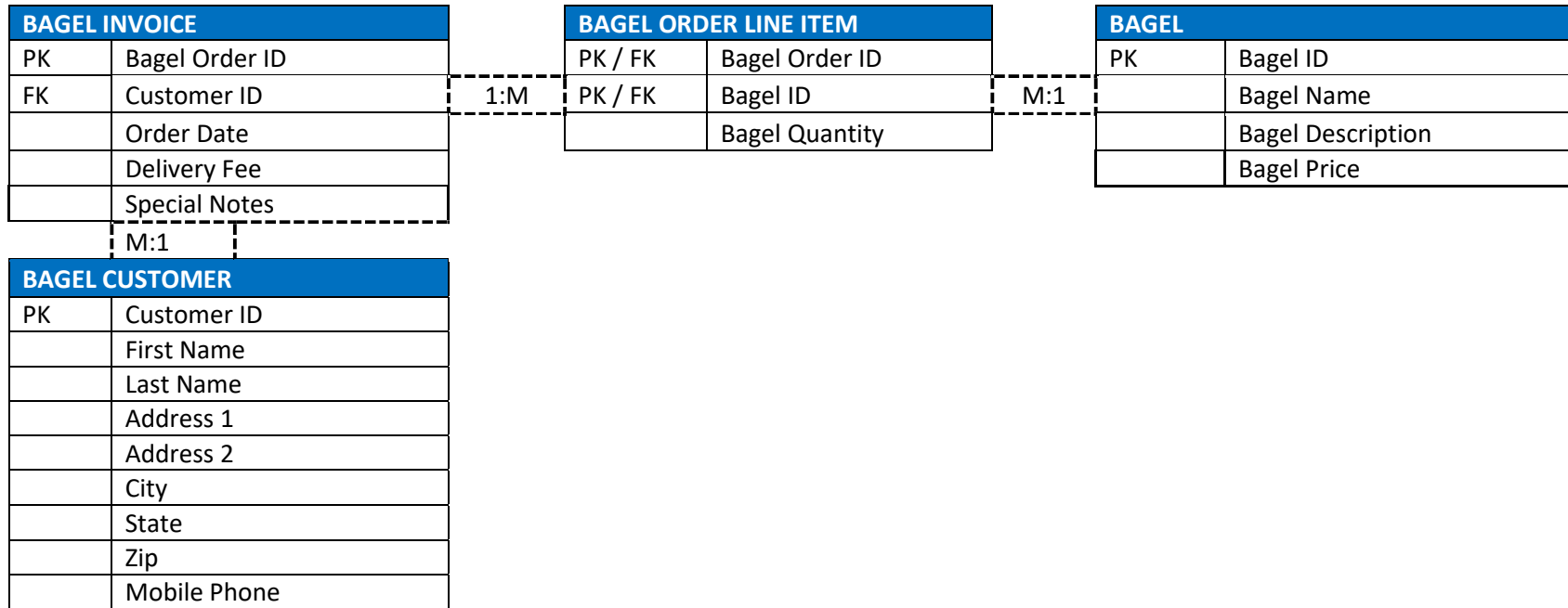
## 2<sup>nd</sup> Normal Form



After 1NF was completed, 2NF was achieved by making sure all non-key attributes (columns) were functionally dependent on the *entire* primary key (PK). Two new relations were created (Bagel Order Line Item and Bagel) with their respective non-key attributes to conform to 2NF standards. The Bagel relation contains all attributes that strictly describe bagels and not the order ID, so were separated from the Bagel Order relation. Bagel Quantity is the only attribute that is dependent on the entire composite primary key, so were separated in the Bagel Order Line Item relation. The rest of the attributes are dependent on the Bagel Order ID only, so were left in the original Bagel Order relation.

Cardinality between the Bagel Order entity and Bagel Order Line Item entity can be explained as one order can have many order line items, while each order line item instance can be on only one order (1 to Many). An Order Line Item can only have 1 type of bagel, while each bagel type can appear on multiple line items (Many to One).

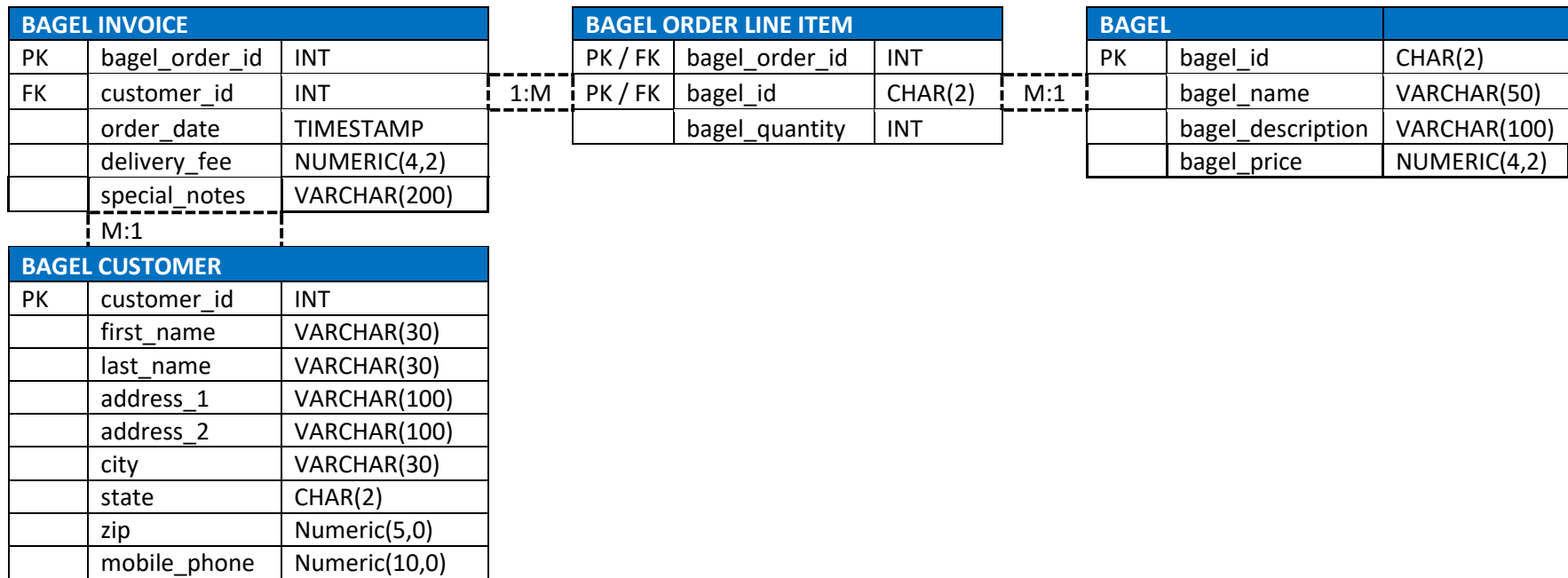
### 3<sup>rd</sup> Normal Form



Once 2NF is done, 3NF was performed to eliminate transitive dependencies (non-key attributes with dependencies on other non-key attributes). A new attribute (Customer ID) was created as a foreign key in the renamed Bagel Order (Bagel Invoice) and used as a PK in Bagel Customer to tie all customer attributes to the new PK.

Cardinality remained the same for the old relations, but for the new relations a bagel invoice can only have one customer, while a customer can have many invoices (many to one).

## Final Physical Database Model



All attributes were renamed to conform to database naming conventions (no spaces) and data types were described for each attribute. Some string (CHAR) and number (Numeric) data types were defined for attributes that are known to have a fixed length.

# Jaunty Coffee Co. Database Creation in MySQL

The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The 'SCHEMAS' panel on the left shows the 'jaunty\_coffee\_co' database selected. The main editor window, titled 'Query1', contains the following SQL script:

```
1 CREATE TABLE IF NOT EXISTS COFFEE_SHOP (  
2     shop_id INT,  
3     shop_name VARCHAR(50),  
4     city VARCHAR(50),  
5     state CHAR(2),  
6     PRIMARY KEY (shop_id)  
7 );  
8  
9 CREATE TABLE IF NOT EXISTS EMPLOYEE (  
10     employee_id INT,  
11     first_name VARCHAR(30),  
12     last_name VARCHAR(30),  
13     hire_date DATE,  
14     job_title VARCHAR(30),  
15     shop_id INT,  
16     PRIMARY KEY (employee_id),  
17     FOREIGN KEY (shop_id)  
18     REFERENCES COFFEE_SHOP (shop_id)  
19 );  
20  
21 CREATE TABLE IF NOT EXISTS SUPPLIER (  
22     supplier_id INT,  
23     company_name VARCHAR(50),  
24     country VARCHAR(30),  
25     sales_contact_name VARCHAR(60),  
26     email VARCHAR(50) NOT NULL,  
27     PRIMARY KEY (supplier_id)  
28 );  
29  
30 CREATE TABLE IF NOT EXISTS COFFEE (  
31     coffee_id INT,
```

The 'Output' panel at the bottom shows the execution results of the script:

#	Time	Action	Message	Duration / Fetch
1	01:59:00	CREATE TABLE IF NOT EXISTS COFFEE_SHOP ( shop_id INT, shop_name VARCHAR(50), city VARCHAR(50), state CHAR(2), PRIMARY KEY (...)	0 row(s) affected	0.000 sec
2	01:59:00	CREATE TABLE IF NOT EXISTS EMPLOYEE ( employee_id INT, first_name VARCHAR(30), last_name VARCHAR(30), hire_date DATE, job_title V...	0 row(s) affected	0.031 sec
3	01:59:00	CREATE TABLE IF NOT EXISTS SUPPLIER ( supplier_id INT, company_name VARCHAR(50), country VARCHAR(30), sales_contact_name VARC...	0 row(s) affected	0.000 sec
4	01:59:00	CREATE TABLE IF NOT EXISTS COFFEE ( coffee_id INT, shop_id INT, supplier_id INT, coffee_name VARCHAR(30), price_per_pound NUMERIC...	0 row(s) affected	0.016 sec

All tables were created as specified in the Entity Relationship Diagram (ERD) and added to the jaunty\_coffee\_co database.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

jaunty\_coffee\_co

- Tables
- Views
- Stored Procedures
- Functions

sakila

sys

world

Administration Schemas

Information

Schema: jaunty\_coffee\_co

Query 1

```
1 • CREATE DATABASE JAUNTY_COFFEE_CO;
2 • CREATE TABLE IF NOT EXISTS COFFEE_SHOP (
9 • CREATE TABLE IF NOT EXISTS EMPLOYEE (
20 • CREATE TABLE IF NOT EXISTS SUPPLIER (
28 • CREATE TABLE IF NOT EXISTS COFFEE (
40 • INSERT INTO COFFEE_SHOP
41 VALUES
42 (1, 'Jaunt1', 'Dallas', 'TX'),
43 (2, 'Jaunt2', 'Houston', 'TX'),
44 (3, 'Jaunt3', 'Austin', 'TX');
45 • INSERT INTO EMPLOYEE
46 VALUES
47 (1, 'John', 'Smith', '2022-01-01', 'clerk', 1),
48 (2, 'Mary', 'Jane', '2022-02-01', 'barista', 2),
49 (3, 'Jacob', 'Johnston', '2022-03-01', 'manager', 3);
50 • INSERT INTO SUPPLIER
51 VALUES
52 (1, 'Sup1', 'USA', 'Venessa Cruz', 'vcruz1@gmail.com'),
53 (2, 'Sup2', 'Canada', 'Maurice Leal', 'mleal2@gmail.com'),
54 (3, 'Sup3', 'China', 'Mike Zhao', 'mzhao3@gmail.com');
55 • INSERT INTO COFFEE
56 VALUES
57 (1, 1, 1, 'cold', 111.11),
58 (2, 2, 2, 'warm', 222.22),
59 (3, 3, 3, 'hot', 333.33);
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 1	02:26:33	INSERT INTO COFFEE_SHOP VALUES (1, 'Jaunt1', 'Dallas', 'TX'), (2, 'Jaunt2', 'Houston', 'TX'), (3, 'Jaunt3', 'Austin', 'TX')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
✓ 2	02:26:33	INSERT INTO EMPLOYEE VALUES (1, 'John', 'Smith', '2022-01-01', 'clerk', 1), (2, 'Mary', 'Jane', '2022-02-01', 'barista', 2), (3, 'Jacob', 'Johnston', '2022-03-01', 'manager', 3);	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
✓ 3	02:26:33	INSERT INTO SUPPLIER VALUES (1, 'Sup1', 'USA', 'Venessa Cruz', 'vcruz1@gmail.com'), (2, 'Sup2', 'Canada', 'Maurice Leal', 'mleal2@gmail.com'), (3, 'Sup3', 'China', 'Mike Zhao', 'mzhao3@gmail.com');	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec
✓ 4	02:26:33	INSERT INTO COFFEE VALUES (1, 1, 1, 'cold', 111.11), (2, 2, 2, 'warm', 222.22), (3, 3, 3, 'hot', 333.33);	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec

Object Info Session

3 rows were added to each table with random values.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

jaunty\_coffee\_co

- Tables
- Views
- Stored Procedures
- Functions

sakila

sys

world

Query 1

Limit to 1000 rows

```
57 (1, 1, 1, 'cold', 111.11),
58 (2, 2, 2, 'warm', 222.22),
59 (3, 3, 3, 'hot', 333.33);
60 CREATE VIEW EMPLOYEE_FULL_NAME AS
61 SELECT *, CONCAT (first_name, ' ', last_name) employee_full_name
62 FROM EMPLOYEE;
63 SELECT * FROM EMPLOYEE_FULL_NAME;
```

Result Grid

	employee_id	first_name	last_name	hire_date	job_title	shop_id	employee_full_name
1	John	Smith	2022-01-01	clerk	1	John Smith	
2	Mary	Jane	2022-02-01	barista	2	Mary Jane	
3	Jacob	Johnston	2022-03-01	manager	3	Jacob Johnston	

Administration Schemas

Information

Schema: jaunty\_coffee\_co

EMPLOYEE\_FULL\_NAME 5 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	03:22:16	CREATE VIEW EMPLOYEE_FULL_NAME AS SELECT *, CONCAT (first_name, ' ', last_name) employee_full_name FROM EMPLOYEE	0 row(s) affected	0.000 sec
2	03:22:40	SELECT * FROM EMPLOYEE_FULL_NAME LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

First and Last name were concatenated with a space into a new column, then added to a view of the EMPLOYEE table.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

jaunty\_coffee\_co

Tables

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

Schema: jaunty\_coffee\_co

Query 1

```
1 • CREATE DATABASE JAUNTY_COFFEE_CO;
2 • CREATE TABLE IF NOT EXISTS COFFEE_SHOP (
9 • CREATE TABLE IF NOT EXISTS EMPLOYEE (
20 • CREATE TABLE IF NOT EXISTS SUPPLIER (
28 • CREATE TABLE IF NOT EXISTS COFFEE (
40 • INSERT INTO COFFEE_SHOP
41 VALUES
42 (1, 'Jaunt1', 'Dallas', 'TX'),
43 (2, 'Jaunt2', 'Houston', 'TX'),
44 (3, 'Jaunt3', 'Austin', 'TX');
45 • INSERT INTO EMPLOYEE
46 VALUES
47 (1, 'John', 'Smith', '2022-01-01', 'clerk', 1),
48 (2, 'Mary', 'Jane', '2022-02-01', 'barista', 2),
49 (3, 'Jacob', 'Johnston', '2022-03-01', 'manager', 3);
50 • INSERT INTO SUPPLIER
51 VALUES
52 (1, 'Sup1', 'USA', 'Venessa Cruz', 'vcruz1@gmail.com'),
53 (2, 'Sup2', 'Canada', 'Maurice Leal', 'mleal2@gmail.com'),
54 (3, 'Sup3', 'China', 'Mike Zhao', 'mzhao3@gmail.com');
55 • INSERT INTO COFFEE
56 VALUES
57 (1, 1, 1, 'cold', 111.11),
58 (2, 2, 2, 'warm', 222.22),
59 (3, 3, 3, 'hot', 333.33);
60 • CREATE VIEW EMPLOYEE_FULL_NAME AS
61 SELECT *, CONCAT (first_name, ' ', last_name) employee_full_name
62 FROM EMPLOYEE;
63 • SELECT * FROM EMPLOYEE_FULL_NAME;
64 • CREATE INDEX coffee_index ON COFFEE(coffee_name);
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	03:28:48	CREATE INDEX coffee_index ON COFFEE(coffee_name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.032 sec

Object Info Session

An index named coffee\_index was created on the column coffee\_name in table COFFEE.



The screenshot displays the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' panel with a search filter and a tree view of databases including 'jaunty\_coffee\_co', 'sakila', 'sys', and 'world'. The 'jaunty\_coffee\_co' database is selected, showing its tables, views, stored procedures, and functions. The main query editor shows a SQL script with line numbers 61 to 67. The script includes a SELECT statement with a CONCAT function, a FROM clause, a CREATE INDEX statement, and a WHERE clause filtering for 'John Smith'. The 'Result Grid' tab is active, showing a single row with the value 'John Smith' under the column 'employee\_full\_name'. The bottom status bar indicates the query was executed successfully, returning 1 row(s) in 0.000 seconds. A right-hand panel titled 'SQLAdditions' contains a message about automatic context help.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

jaunty\_coffee\_co

- Tables
- Views
- Stored Procedures
- Functions

sakila

sys

world

Query 1

```
61 SELECT *, CONCAT (first_name, ' ', last_name) employee_full_name
62 FROM EMPLOYEE;
63 SELECT * FROM EMPLOYEE_FULL_NAME;
64 CREATE INDEX coffee_index ON COFFEE(coffee_name);
65 SELECT employee_full_name
66 FROM EMPLOYEE_FULL_NAME
67 WHERE employee_full_name='John Smith';
```

Limit to 1000 rows

Result Grid

Filter Rows: Export: Wrap Cell Content: I

employee\_full\_name

John Smith

Administration Schemas

Information

Schema: jaunty\_coffee\_co

EMPLOYEE\_FULL\_NAME 7 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	03:34:54	SELECT employee_full_name FROM EMPLOYEE_FULL_NAME WHERE employee_full_name='John Smith' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

To demonstrate a SELECT FROM WHERE (SFW) query, an employee full name match with 'John Smith' was selected from the EMPLOYEE\_FULL\_NAME view.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

jaunty\_coffee\_co

- Tables
- Views
- Stored Procedures
- Functions
- sakila
- sys
- world

SQL\*

```
67 SELECT COFFEE.*, COFFEE_SHOP.*, SUPPLIER.*
68 FROM COFFEE
69 INNER JOIN COFFEE_SHOP
70 ON COFFEE.shop_id = COFFEE_SHOP.shop_id
71 INNER JOIN SUPPLIER
72 ON COFFEE.supplier_id = SUPPLIER.supplier_id
73 ORDER BY COFFEE.shop_id;
```

Result Grid

	coffee_id	shop_id	supplier_id	coffee_name	price_per_pound	shop_id	shop_name	city	state	supplier_id	company_name	country	sales_contact_name	email
1	1	1	1	cold	111.11	1	Jaunt1	Dallas	TX	1	Sup1	USA	Venessa Cruz	vcruz1@gmail.com
2	2	2	2	warm	222.22	2	Jaunt2	Houston	TX	2	Sup2	Canada	Maurice Leal	mleal2@gmail.com
3	3	3	3	hot	333.33	3	Jaunt3	Austin	TX	3	Sup3	China	Mike Zhao	mzhao3@gmail.com

Administration Schemas

Information

Schema: jaunty\_coffee\_co

Result 2 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	04:15:36	SELECT COFFEE.*, COFFEE_SHOP.*, SUPPLIER.* FROM COFFEE INNER JOIN COFFEE_SHOP ON COFFEE.shop_id = COFFEE_SHOP.shop_id IN...	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Query Completed

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Inner join was used on the 2 tables COFFEE, COFFEE\_SHOP, and SUPPLIER to display all attributes from all tables.