One-to-Many and Many-to-Many Joins - Lab

Introduction

In this lab, you'll practice your knowledge of `one-to-many` and `many-to-many` relationships!

Objectives

You will be able to:

- Explain `one-to-many` and `many-to-many` joins as well as implications for the size of query results
- Query data using `one-to-many` and `many-to-many` joins

One-to-Many and Many-to-Many Joins

Connect to the Database

Include the relevant imports, then connect to the database located at `data.sqlite`.

```
# brian-answer
import pandas as pd
import sqlite3
conn = sqlite3.connect('data.sqlite')
```

Employees and Their Offices (a One-to-One Join)

Select all of the employees including their first name and last name along with the city and state of the office that they work out of (if they have one). Include all employees and order them by their first name, then their last name.

```
# brian-answer
q = """
    SELECT firstName, lastName, city, state FROM employees
    JOIN offices
    USING(officeCode)
    ORDER BY firstName, lastName;
"""
df = pd.read_sql(q, conn)
print('Total number of results:', len(df))
print('The top 5 results are:')
df.head()
```

```
Total number of results: 23
The top 5 results are:

  firstName lastName           city state
0      Andy   Fixter         Sydney
1   Anthony      Bow  San Francisco    CA
2     Barry    Jones         London
3     Diane   Murphy  San Francisco    CA
4  Foon Yue    Tseng            NYC    NY
```

Customers and Their Orders (a One-to-Many Join)

Select all of the customer contacts (first and last names) along with details for each of the customers' order numbers, order dates, and statuses.

```
q = 'select * from customers;'
pd.read_sql(q, conn).head(1)


   customerNumber       customerName contactLastName contactFirstName
\
0             103  Atelier graphique         Schmitt           Carine


        phone    addressLine1 addressLine2    city state postalCode
country  \
0  40.32.2555  54, rue Royale               Nantes            44000
France

  salesRepEmployeeNumber  creditLimit
0                   1370        21000

# brian-answer
q = """
    SELECT
        contactFirstName, contactLastName,
        orderNumber, orderDate, status
    FROM customers
    JOIN orders
    USING(customerNumber);
"""
df = pd.read_sql(q, conn)
print('Total number of results:', len(df))
print('The top 5 results are:')
df.head()

Total number of results: 326
The top 5 results are:
```

```
  contactFirstName contactLastName  orderNumber   orderDate   status
0          Carine          Schmitt        10123  2003-05-20  Shipped
1          Carine          Schmitt        10298  2004-09-27  Shipped
2          Carine          Schmitt        10345  2004-11-25  Shipped
3             Jean             King        10124  2003-05-21  Shipped
4             Jean             King        10278  2004-08-06  Shipped
```

Customers and Their Payments (Another One-to-Many Join)

Select all of the customer contacts (first and last names) along with details for each of the customers' payment amounts and date of payment. Sort these results in descending order by the payment amount.

```
# brian-answer
q = """
    SELECT
       contactFirstName, contactLastName,
       amount, paymentDate
    FROM customers
    JOIN payments AS p
    USING(customerNumber)
    ORDER BY
        p.amount DESC;
"""
df = pd.read_sql(q, conn)
print('Total number of results:', len(df))
print('The top 5 results are:')
df.head()
```

```
Total number of results: 273
The top 5 results are:
```

```
  contactFirstName contactLastName     amount paymentDate
0           Diego           Freyre  120166.58  2005-03-18
1           Diego           Freyre  116208.40  2004-12-31
2            Susan          Nelson  111654.40  2003-08-15
3             Eric       Natividad  105743.00  2003-12-26
4            Susan          Nelson  101244.59  2005-03-05
```

Orders, Order Details, and Product Details (a Many-to-Many Join)

Select all of the customer contacts (first and last names) along with the product names, quantities, and date ordered for each of the customers and each of their orders. Sort these in descending order by the order date.

Note: This will require joining 4 tables! This can be tricky! Give it a shot, and if you're still stuck, turn to the next section where you'll see how to write subqueries that can make complex queries such as this much simpler!

```
q = """
SELECT
    contactFirstName,
    contactLastName,
    productName,
    quantityOrdered,
    orderDate
FROM customers
JOIN orders
    USING(customerNumber)
JOIN orderdetails
    USING(orderNumber)
JOIN products
    USING (productCode)
ORDER BY orderDate DESC;
"""
pd.read_sql(q, conn).head()
```

```
  contactFirstName contactLastName                    productName  \
0          Janine          Labrune          1962 LanciaA Delta 16V
1          Janine          Labrune              1957 Chevy Pickup
2          Janine          Labrune  1998 Chrysler Plymouth Prowler
3          Janine          Labrune          1964 Mercedes Tour Bus
4          Janine          Labrune           1926 Ford Fire Engine

   quantityOrdered   orderDate
0               38  2005-05-31
1               33  2005-05-31
2               28  2005-05-31
3               38  2005-05-31
4               19  2005-05-31
```

```
q = 'SELECT * FROM orderdetails;'
pd.read_sql(q, conn).head(1)

# q = 'SELECT * FROM products;'
# pd.read_sql(q, conn).head(1)
```

```
   orderNumber productCode  quantityOrdered  priceEach
orderLineNumber
0        10100    S18_1749               30      136.0
3

# brian-added
q = """
SELECT
    c.contactFirstName, c.contactLastName,
```

```
    p.productName,
    od.quantityOrdered,
    o.orderDate
FROM customers AS c
JOIN ORDERS AS o
    USING(customerNumber)
JOIN orderdetails as od
    USING(orderNumber)
JOIN products as p
    USING(productCode)
ORDER BY
    orderDate DESC;
"""
df = pd.read_sql(q, conn)
print('Total number of results:', len(df))
print('The top 5 results are:')
df.head()
```

```
Total number of results: 2996
The top 5 results are:
```

	contactFirstName	contactLastName	productName	\
0	Janine	Labrune	1962 LanciaA Delta 16V	
1	Janine	Labrune	1957 Chevy Pickup	
2	Janine	Labrune	1998 Chrysler Plymouth Prowler	
3	Janine	Labrune	1964 Mercedes Tour Bus	
4	Janine	Labrune	1926 Ford Fire Engine	

	quantityOrdered	orderDate
0	38	2005-05-31
1	33	2005-05-31
2	28	2005-05-31
3	38	2005-05-31
4	19	2005-05-31

Finally, `close` the connection.

```
conn.close()
```

Summary

In this lab, you practiced your knowledge of `one-to-many` and `many-to-many` relationships!