README.md

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

Introduction

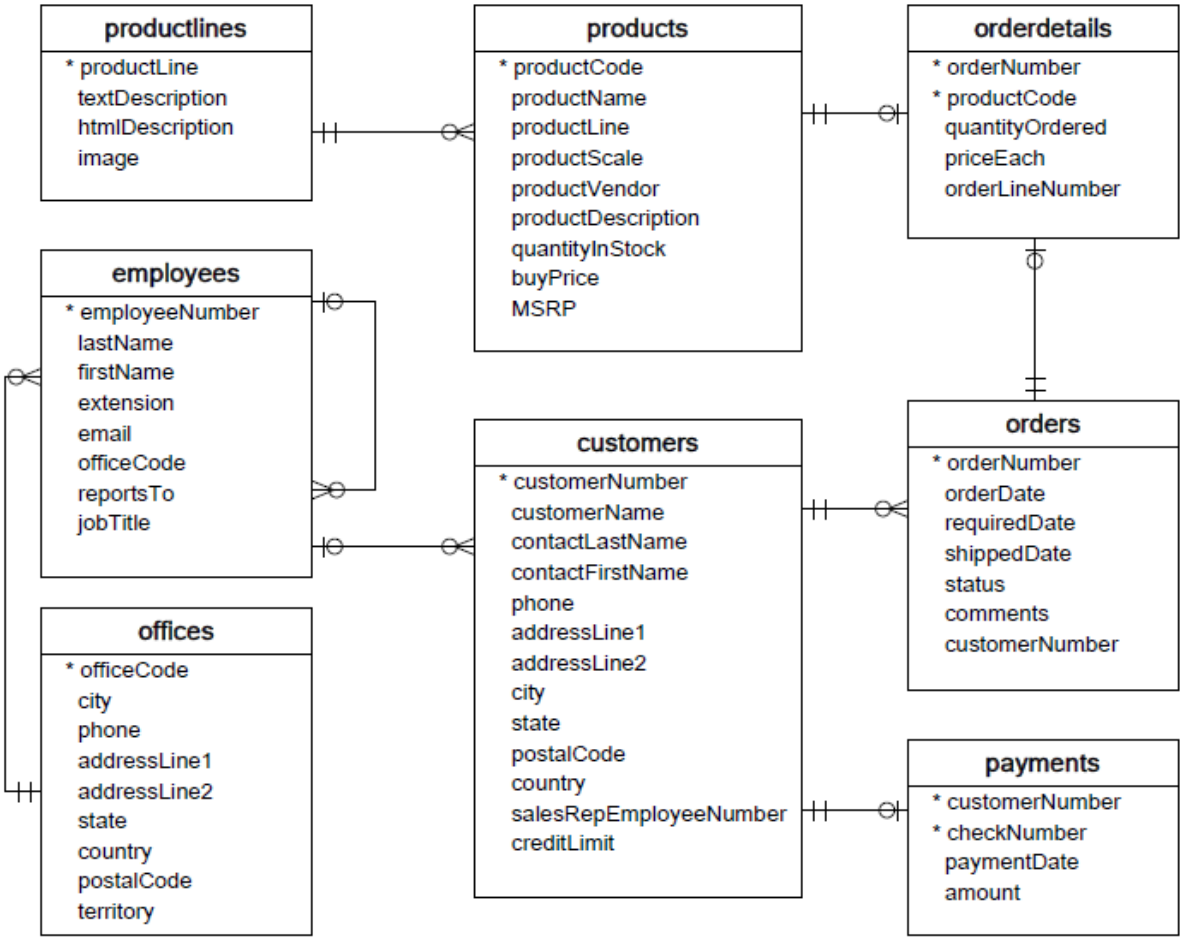
In this lab, you'll practice your knowledge of one-to-many and many-to-many relationships!

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`.

```
import sqlite3
import pandas as pd
```

```
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.

```
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))
df.head()
```

```
Total number of results: 23
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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
    contactFirstName,
    contactLastName,
```

```
    orderNumber,
    orderDate,
    status
FROM customers
JOIN orders
    USING(customerNumber)
;
"""
df = pd.read_sql(q, conn)
print('Total number of results:', len(df))
df.head()
```

```
Total number of results: 326
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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.

```
q = """
SELECT
    contactFirstName,
    contactLastName,
    amount,
    paymentDate
FROM customers
JOIN payments
    USING(customerNumber)
ORDER BY amount DESC
;
"""
df = pd.read_sql(q, conn)
print('Total number of results:', len(df))
df.head()
```

```
Total number of results: 273
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	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

	contactFirstName	contactLastName	amount	paymentDate
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
;"""
df = pd.read_sql(q, conn)
print('Total number of results:', len(df))
df.head()
```

```
Total number of results: 2996
```

<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }

```
.dataframe tbody tr th {
    vertical-align: top;
```

```
}

.dataframe thead th {
    text-align: right;
}
```

</style>

	contactFirstName	contactLastName	productName	quantityOrdered	
0	Janine	Labrune	1962 LanciaA Delta 16V	38	
1	Janine	Labrune	1957 Chevy Pickup	33	
2	Janine	Labrune	1998 Chrysler Plymouth Prowler	28	
3	Janine	Labrune	1964 Mercedes Tour Bus	38	
4	Janine	Labrune	1926 Ford Fire Engine	19	

Summary

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

Releases

No releases published

Packages

No packages published

Contributors 8

Languages

- **Jupyter Notebook** 100.0%