Getting Familiar with SQL in Jupyter Notebooks-Copy1

November 4, 2025

1 Ungraded Lab : Getting Familiar with SQL in Jupyter Notebooks

1.1 Overview

In this hands-on lab, you'll learn how to write your first SQL queries using Jupyter Notebook. We'll work with real-world data to solve actual business problems.

1.2 Learning Objectives

By the end of this lab, you will be able to: - Connect to a SQL database in Jupyter Notebook
- Write and execute basic SELECT statements
- Explore real business data using SQL queries

1.3 Meet BookCycle

Before we dive in, let's understand our data context:

BookCycle is a growing used bookstore chain with three locations across Seattle. Founded by former librarian Sarah Chen, they specialize in making classic literature accessible through pre-owned books. Their business model involves buying books from community members and reselling them at 30-70% below retail prices.

Our dataset contains information about their book inventory and sales across their three locations (Downtown, University District, and Suburban).

1.4 Activities

Let's begin by seeing how SQL works in Jupyter Notebook.

1.4.1 Activity 1: Setting Up Your Environment

Step 1: Let's watch how to set up our environment:

[1]:
```
# Import the necessary libraries
import sqlite3
import pandas as pd

# Setting up the database. DO NOT edit the code given below
from db_setup import setup_database
setup_database()
```

```
# Connect to the database
conn = sqlite3.connect('bookcycle.db')
```

```
Database setup complete: Tables created and populated with data!
```

1.4.2 Activity 2 : Your First SQL Query

Step 1: Let's see how a basic query works. Here's an example to view the first 5 books in our inventory:

[2]:
```
# Example query showing how to select data
query = """SELECT * FROM books LIMIT 5;"""

# Execute the query and display results using pandas
results_df = pd.read_sql(query, conn)
display(results_df)
```

```
  book_id                          title            author           isbn  \
0   B1001              A Christmas Carol   Charles Dickens  9780141324524
1   B1002            A Farewell to Arms  Ernest Hemingway  9780684801469
2   B1003          A Tale of Two Cities   Charles Dickens  9780141439600
3   B1004  Adventures of Huckleberry Finn       Mark Twain  9780142437179
4   B1005                     Agnes Grey      Anne Bronte  9780140432107

             genre  condition  purchase_price  list_price date_acquired  \
0  Classic Fiction       Good             5.5        8.99    2023-01-15
1  Classic Fiction  Very Good             7.0       11.99    2023-01-15
2  Classic Fiction       Fair             4.5        7.99    2023-01-16
3  Classic Fiction       Good             6.0        9.99    2023-01-16
4  Classic Fiction       Fair             4.0        7.99    2023-01-17

  current_location  quantity
0         Suburban         2
1       University         3
2         Downtown         2
3       University         4
4         Suburban         1
```

Step 2: Now you try! Write a query to view ALL books in the inventory:

[3]:
```
# Write a query to see all books in invertory
query = """select * from books;"""

# Execute and display results
result = pd.read_sql(query, conn)
display(result)
```

```
  book_id                          title                author  \
```

```
0    B1001               A Christmas Carol         Charles Dickens
1    B1002             A Farewell to Arms        Ernest Hemingway
2    B1003          A Tale of Two Cities         Charles Dickens
3    B1004  Adventures of Huckleberry Finn              Mark Twain
4    B1005                      Agnes Grey             Anne Bronte
..     …                             …                     …
95   B1096         Thus Spoke Zarathustra     Friedrich Nietzsche
96   B1097                Treasure Island  Robert Louis Stevenson
97   B1098                        Ulysses             James Joyce
98   B1099                         Walden     Henry David Thoreau
99   B1100              Wuthering Heights            Emily Bronte

             isbn                genre  condition  purchase_price  list_price  \
0   9780141324524      Classic Fiction       Good             5.5        8.99
1   9780684801469      Classic Fiction  Very Good             7.0       11.99
2   9780141439600      Classic Fiction       Fair             4.5        7.99
3   9780142437179      Classic Fiction       Good             6.0        9.99
4   9780140432107      Classic Fiction       Fair             4.0        7.99
..            …                   …        …               …          …
95  9780140441185   Classic Philosophy   Like New             8.0       13.99
96  9780141321011      Classic Fiction       Fair             4.5        7.99
97  9780141182803      Classic Fiction  Very Good             8.0       13.99
98  9780140390445  Classic Non-Fiction       Good             6.5       10.99
99  9780141439556      Classic Fiction   Like New             7.5       12.99

   date_acquired current_location  quantity
0     2023-01-15         Suburban         2
1     2023-01-15       University         3
2     2023-01-16         Downtown         2
3     2023-01-16       University         4
4     2023-01-17         Suburban         1
..           …               …       …
95    2023-03-03       University         3
96    2023-03-04         Suburban         2
97    2023-03-04       University         3
98    2023-03-05       University         4
99    2023-03-05       University         4

[100 rows x 11 columns]
```

Hint: Remove the LIMIT clause from the example above

1.4.3 Activity 3: Selecting Specific Columns

Step 1: Let's see how to select specific columns. Here's an example finding just the titles:

[4]:
```
# Example Query to Select only book titles
query = """
```

```
SELECT title
FROM books;
"""

df = pd.read_sql_query(query, conn)
display(df)
```

```
                           title
0                A Christmas Carol
1               A Farewell to Arms
2             A Tale of Two Cities
3   Adventures of Huckleberry Finn
4                       Agnes Grey
..                             …
95          Thus Spoke Zarathustra
96                 Treasure Island
97                         Ulysses
98                          Walden
99                Wuthering Heights

[100 rows x 1 columns]
```

Step 2: Now, help BookCycle's staff by showing title, author, and price for inventory checking:

[7]:
```
# Write a query to Select and Display title, author, and purchase_price for␣
↪inventory checking:
query = """
select title, author, purchase_price
from books;
"""

# Execute and display results
result = pd.read_sql(query, conn)
display(result)
```

```
                           title                  author  purchase_price
0                A Christmas Carol         Charles Dickens             5.5
1               A Farewell to Arms        Ernest Hemingway             7.0
2             A Tale of Two Cities         Charles Dickens             4.5
3   Adventures of Huckleberry Finn              Mark Twain             6.0
4                       Agnes Grey             Anne Bronte             4.0
..                             …                       …               …
95          Thus Spoke Zarathustra     Friedrich Nietzsche             8.0
96                 Treasure Island  Robert Louis Stevenson             4.5
97                         Ulysses             James Joyce             8.0
98                          Walden     Henry David Thoreau             6.5
99                Wuthering Heights            Emily Bronte             7.5
```

[100 rows x 3 columns]

1.4.4 Activity 4 : Filtering Results

Step 1: Here's how we can filter results with a WHERE clause:

[8]:
```
# Example Query to find books priced above $11
query = """
SELECT title, list_price
FROM books
WHERE list_price > 11;
"""
df = pd.read_sql_query(query, conn)
display(df)
```

	title	list_price
0	A Farewell to Arms	11.99
1	Anna Karenina	11.99
2	Anne of Green Gables	12.99
3	Brave New World	12.99
4	Crime and Punishment	13.99
5	Don Quixote	11.99
6	Fahrenheit 451	12.99
7	Frankenstein	11.99
8	Iliad	13.99
9	Invisible Man	11.99
10	Les Misérables	11.99
11	Little Women	12.99
12	Lord of the Flies	11.99
13	Moby Dick	13.99
14	Native Son	11.99
15	Nineteen Eighty-Four	12.99
16	Of Mice and Men	11.99
17	One Hundred Years of Solitude	12.99
18	Persuasion	12.99
19	Pride and Prejudice	11.99
20	Silas Marner	11.99
21	Tale of Two Cities	11.99
22	The Aeneid	11.99
23	The Bell Jar	12.99
24	The Brothers Karamazov	12.99
25	The Catcher in the Rye	13.99
26	The Count of Monte Cristo	11.99
27	The Divine Comedy	12.99
28	The Grapes of Wrath	11.99
29	The Great Gatsby	12.99
30	The Hound of the Baskervilles	11.99
31	The Little Prince	12.99

32	The Odyssey	12.99
33	The Republic	13.99
34	The Sound and the Fury	11.99
35	The Sun Also Rises	12.99
36	The Three Musketeers	11.99
37	Thus Spoke Zarathustra	13.99
38	Ulysses	13.99
39	Wuthering Heights	12.99

Step 2: Now, help BookCycle find affordable books under $10:

[9]:
```
# Write a query to find books under $10
query = """
select title, list_price
from books
where list_price < 10;
"""
df = pd.read_sql_query(query, conn)
display(df)
```

	title	list_price
0	A Christmas Carol	8.99
1	A Tale of Two Cities	7.99
2	Adventures of Huckleberry Finn	9.99
3	Agnes Grey	7.99
4	Anthem	8.99
5	Beowulf	9.99
6	Bleak House	8.99
7	Candide	8.99
8	Canterbury Tales	7.99
9	David Copperfield	9.99
10	Dracula	7.99
11	Great Expectations	9.99
12	Gulliver's Travels	8.99
13	Heart of Darkness	9.99
14	In Cold Blood	9.99
15	Julius Caesar	7.99
16	Macbeth	9.99
17	Madame Bovary	8.99
18	Medea	9.99
19	Middlemarch	9.99
20	Night	9.99
21	Notes from Underground	7.99
22	Oliver Twist	9.99
23	Othello	9.99
24	Paradise Lost	8.99
25	Robinson Crusoe	7.99
26	Sense and Sensibility	9.99

```
27                              Sons and Lovers        9.99
28                   Tess of the d'Urbervilles        8.99
29                        The Age of Innocence        9.99
30                        The Call of the Wild        9.99
31                        The Canterbury Tales        8.99
32             The Importance of Being Earnest        9.99
33                                  The Jungle        8.99
34                                  The Prince        8.99
35  The Strange Case of Dr. Jekyll and Mr. Hyde        9.99
36                            The Tale of Genji        9.99
37                               The Waste Land        9.99
38                      The Wind in the Willows        9.99
39                              Treasure Island        7.99
```

1.4.5 Close the Connection

It's good practice to close the database connection when you're done

[10]:
```
# Close the database connection
conn.close()
```

1.5 Success Checklist

After each query, check:

Does your output display correctly?

Do the results make sense for BookCycle's business?

Are the number of columns what you expected?

1.6 Common Issues & Solutions

- Problem: Database connection error "no such table: books"
  - Solution: Verify that 'bookcycle.db' is in the correct directory and table name is spelled correctly
- Problem: Column names returning "no such column" error
  - Solution: Check case sensitivity and exact spelling of column names (e.g., 'purchase_price' not 'price')
- Problem: Query returns empty results
  - Solution: Check WHERE clause conditions; ensure you're not filtering out all data (e.g., price $>$ 1000 might return nothing)
- Problem: Syntax errors in SQL queries
  - Solution: Verify all SQL statements end with semicolons and string values are properly quoted

1.7 Summary

In this lab, you've learned to interact with a real-world bookstore database using SQL in Jupyter Notebook. You've practiced writing queries to retrieve and filter data, essential skills for data

analysis and business intelligence. ### Key Points - SQL queries can be executed directly in Jupyter Notebook using pandas and sqlite3 - SELECT statements allow specific column selection for focused data retrieval - WHERE clauses enable data filtering based on business requirements - The combination of Python and SQL provides powerful data analysis capabilities

Remember: This is your space to experiment! If something breaks, you can always restart and try again.