# 4. Python SQLite Game Shop Tutorial – Select

## Contents

Time required: 60 minutes

- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

## SQL Online Tutorial

- https://www.w3schools.com/sql/sql_select.asp

## SQLite with Python Tutorials

- SQLite Databases with Python - Full Course – FreeCodeCamp.org
- https://www.sqlitetutorial.net

## Explanation: Selecting and Fetching Data

Selecting and then fetching the data from records is simple as inserting them. The execute method uses the SQL command of getting all the data from the table using the following syntax.

```
SELECT * from table_name
```

All the table data can be fetched in an object in the form of a list of tuples. Each tuple is a record.

```
rows = cursor.execute("SELECT * FROM fish").fetchall()
print(rows)
# This shows the raw data structure returned: a list of tuples
[
   (1, 'Sammy', 'shark', 1),
   (2, 'Susie', 'cuttlefish', 7),
   (3, 'Nemo', 'clownfish', 6),
   (4, 'Louis', 'lungfish', 5)
]
# To print one record at a time
# Iterate through each tuple in the list with a loop
for record in rows:
    print(record)
# for record in rows, printing each record/tuple
(1, 'Sammy', 'shark', 1)
(2, 'Susie', 'cuttlefish', 7)
(3, 'Nemo', 'clownfish', 6)
(4, 'Louis', 'lungfish', 5)
```

1. The **cursor.execute()** function runs a SELECT statement to retrieve all values for the id, name, species, and tank_number columns in the fish table.
2. **fetchall()** retrieves all the results of the SELECT statement.
3. When we **print(rows)** we see a list of tuples.
4. Each tuple has four entries; one entry for each column we selected from the fish table.
5. The tuples contain the data we inserted earlier: one tuple for Sammy the shark, and one tuple each for the other fish.

If we wanted to retrieve rows in the fish table that match a specific set of criteria, we can use a **WHERE** clause:

```
target_fish_name = "Jamie"
rows = cursor.execute("""SELECT name,
    species,
    tank_number FROM fish WHERE name = ?
    """,
    (target_fish_name,),
).fetchall()
print(rows)
```

As with the previous example, **cursor.execute(<SQL statement>).fetchall()** is used to fetch all the results of a **SELECT** statement. The **WHERE** clause in the **SELECT** statement filters for rows where the value of name is target_fish_name.

Notice that we use **?** to substitute our **target_fish_name** variable into the **SELECT** statement. This uses **target_fish_name** as an **argument** in the SQL statement. We expected to only match one row, and indeed we only see the row for Jamie the cuttlefish returned.

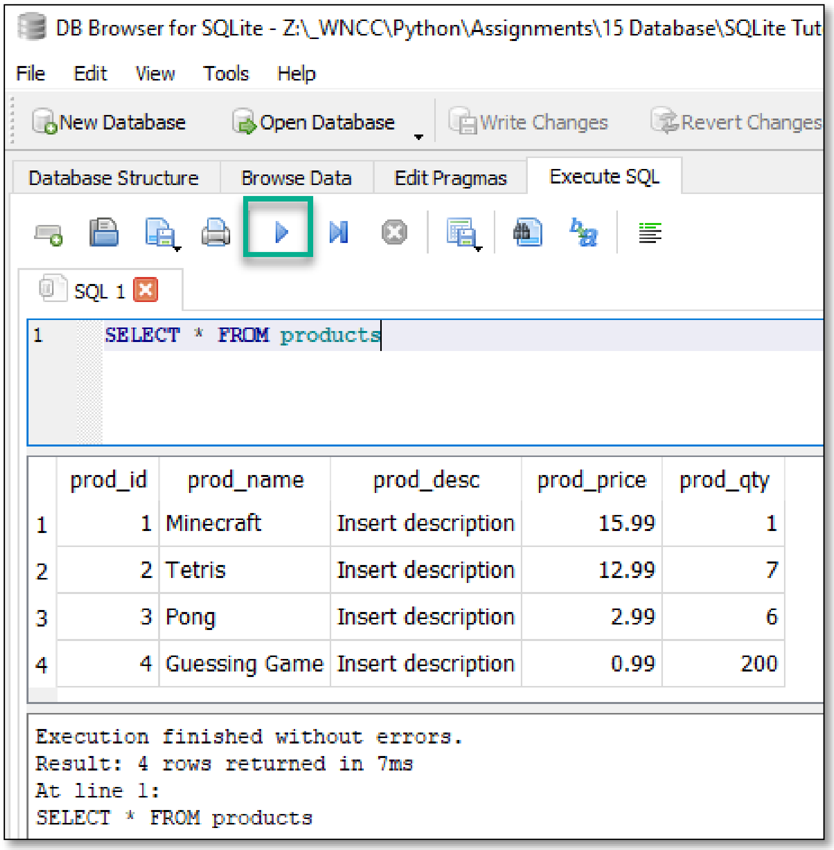
**Warning:** Never use Python string operations to dynamically create a SQL statement string. Using Python string operations to assemble a SQL statement string leaves you vulnerable to SQL injection attacks. SQL injection attacks can be used to steal, alter, or otherwise modify data stored in your database.

Always use the **?** placeholder in your SQL statements to dynamically substitute values from your Python program. Pass a tuple of values as the second argument to **cursor.execute()** to bind your values to the SQL statement.

## Tutorial 1: Select All Records with DB Browser

Let's make sure our records were inserted correctly.

1. Start DB Browser for SQLite. Open **game_stop_3.db**
2. Click the **Execute SQL** tab.
3. Type in the SQL code shown below. (Add the **prod_rank** field.)
4. Click the **Run** button.



```
SELECT * FROM products
```

| | prod_id | prod_name | prod_desc | prod_price | prod_qty |
|---|---|---|---|---|---|
| 1 | 1 | Minecraft | Insert description | 15.99 | 1 |
| 2 | 2 | Tetris | Insert description | 12.99 | 7 |
| 3 | 3 | Pong | Insert description | 2.99 | 6 |
| 4 | 4 | Guessing Game | Insert description | 0.99 | 200 |

```
Execution finished without errors.
Result: 4 rows returned in 7ms
At line 1:
SELECT * FROM products
```

```
SELECT * FROM products
```

**SELECT** selects records and fields from a database.

* selects all records and fields

**FROM** indicates the table the records are being selected from

**products** table we are selecting records from.

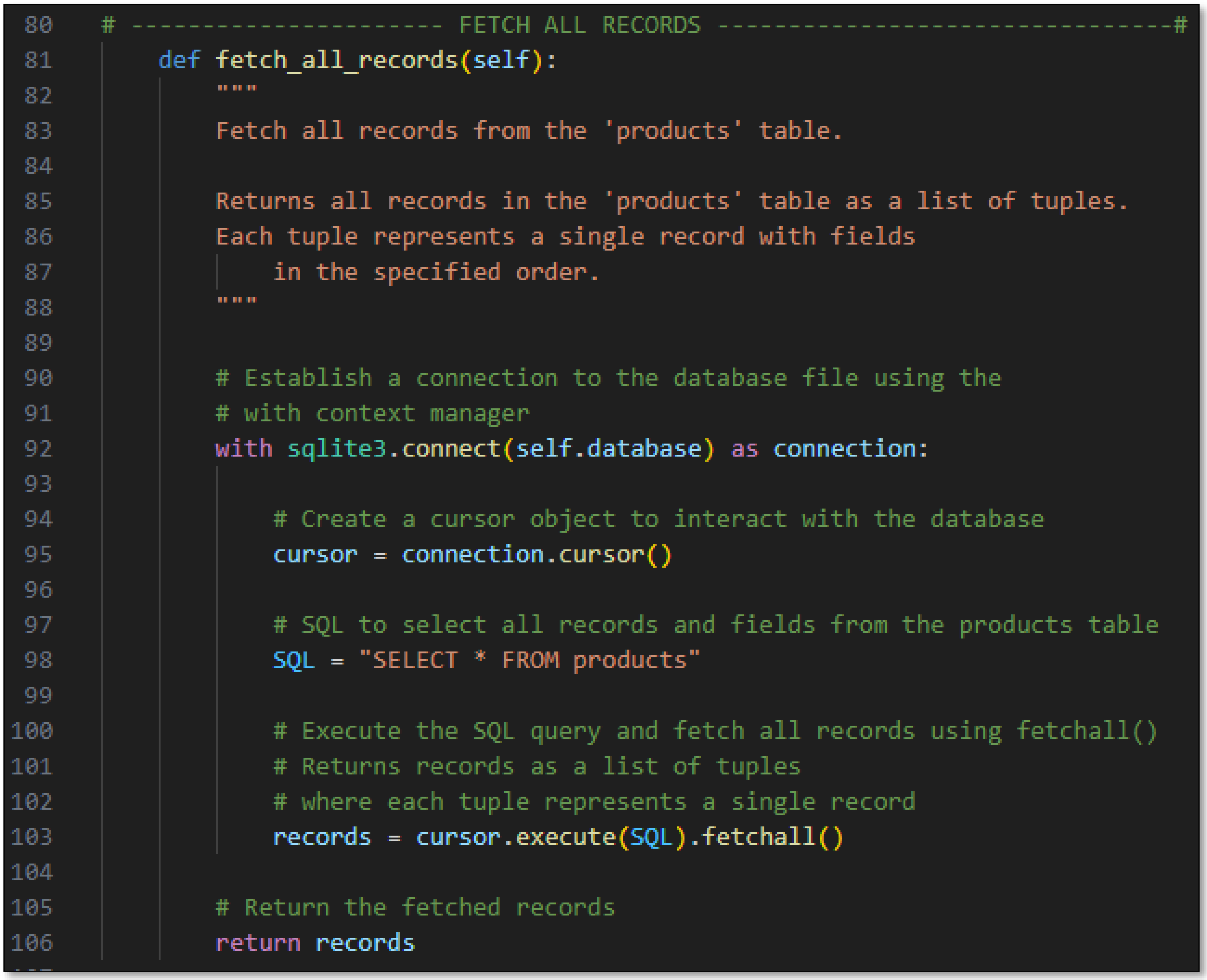
```
SELECT
    prod_id,
    prod_name,
    prod_desc,
    prod_price,
    prod_rank,
    prod_qty
    FROM products
    WHERE prod_name = ?
```

SELECT field names FROM products table where prod_name = ?

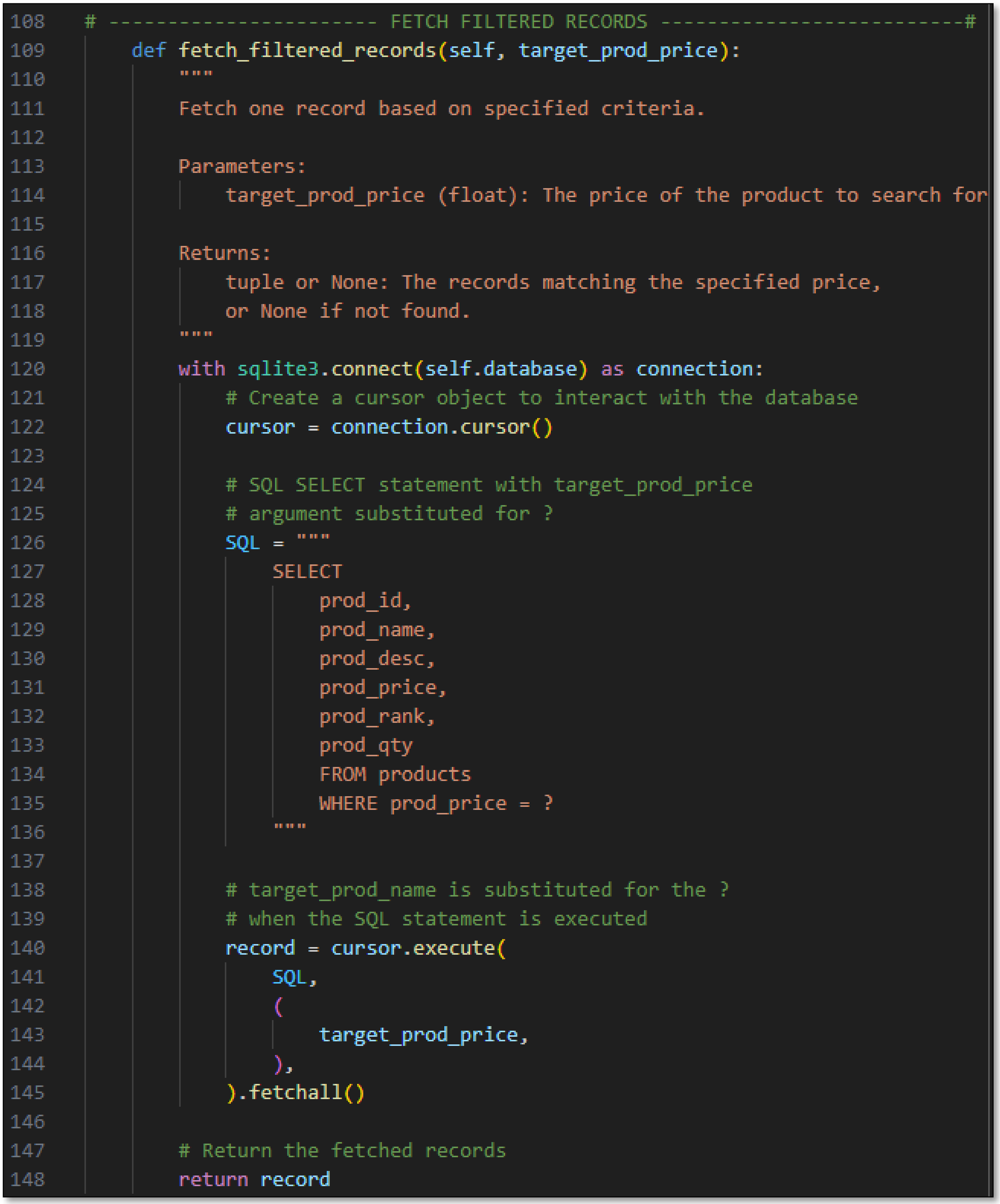
**?** is a placeholder for the field name added as a parameter to the query.

## Tutorial 2: Select and Display Records in Python

1. Open **db_controller.py**
2. Insert the following methods into your program.



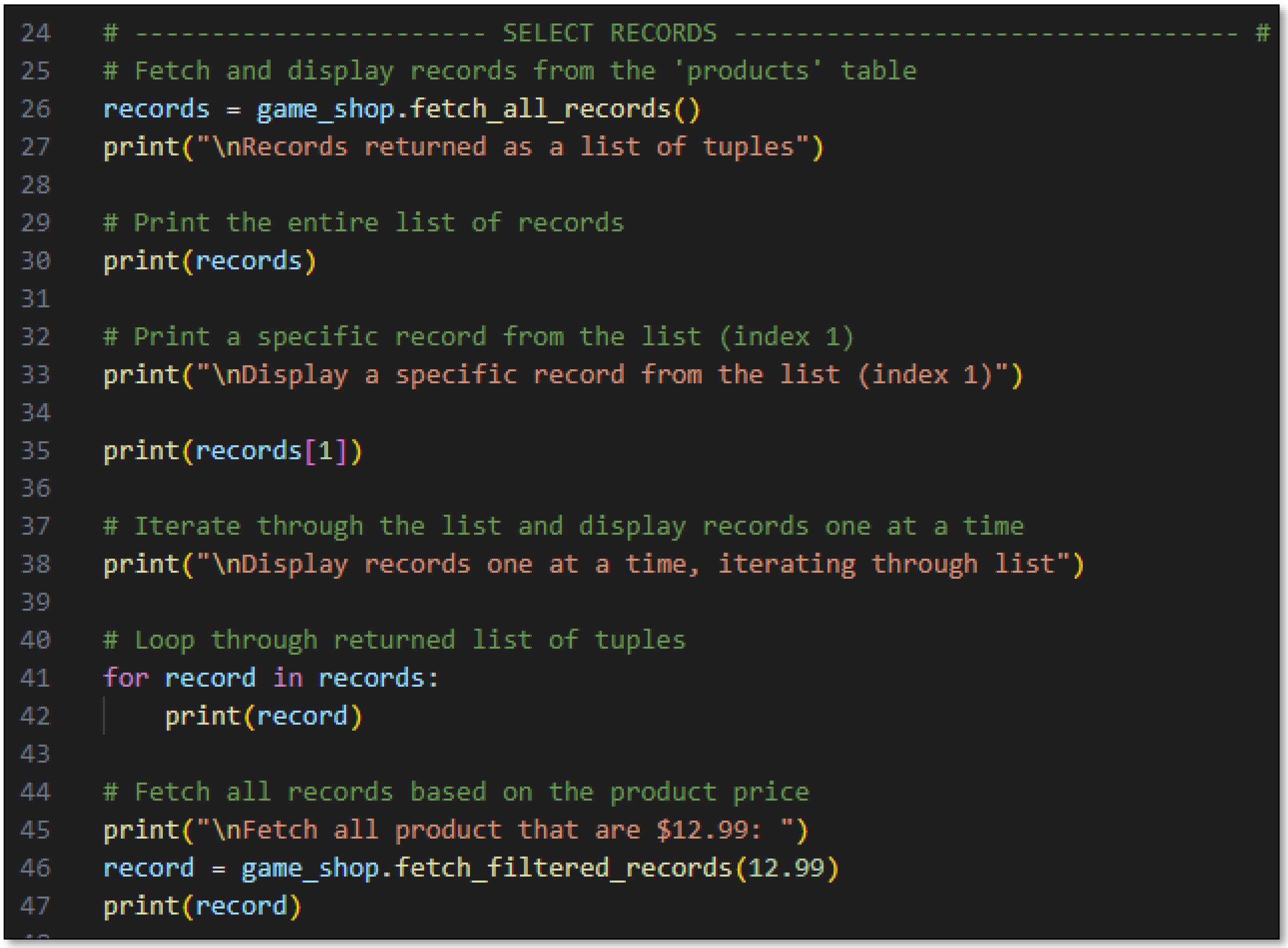
```
# --------------------- FETCH ALL RECORDS ------------------------------#
    def fetch_all_records(self):
        """
        Fetch all records from the 'products' table.

        Returns all records in the 'products' table as a list of tuples.
        Each tuple represents a single record with fields
            in the specified order.
        """

        # Establish a connection to the database file using the
        # with context manager
        with sqlite3.connect(self.database) as connection:

            # Create a cursor object to interact with the database
            cursor = connection.cursor()

            # SQL to select all records and fields from the products table
            SQL = "SELECT * FROM products"

            # Execute the SQL query and fetch all records using fetchall()
            # Returns records as a list of tuples
            # where each tuple represents a single record
            records = cursor.execute(SQL).fetchall()

        # Return the fetched records
        return records
```



```
# ----------------------- FETCH FILTERED RECORDS --------------------------#
    def fetch_filtered_records(self, target_prod_price):
        """
        Fetch one record based on specified criteria.

        Parameters:
            target_prod_price (float): The price of the product to search for

        Returns:
            tuple or None: The records matching the specified price,
            or None if not found.
        """
        with sqlite3.connect(self.database) as connection:
            # Create a cursor object to interact with the database
            cursor = connection.cursor()

            # SQL SELECT statement with target_prod_price
            # argument substituted for ?
            SQL = """
                SELECT
                    prod_id,
                    prod_name,
                    prod_desc,
                    prod_price,
                    prod_rank,
                    prod_qty
                    FROM products
                    WHERE prod_price = ?
                """

            # target_prod_name is substituted for the ?
            # when the SQL statement is executed
            record = cursor.execute(
                SQL,
                (
                    target_prod_price,
                ),
            ).fetchall()

        # Return the fetched records
        return record
```

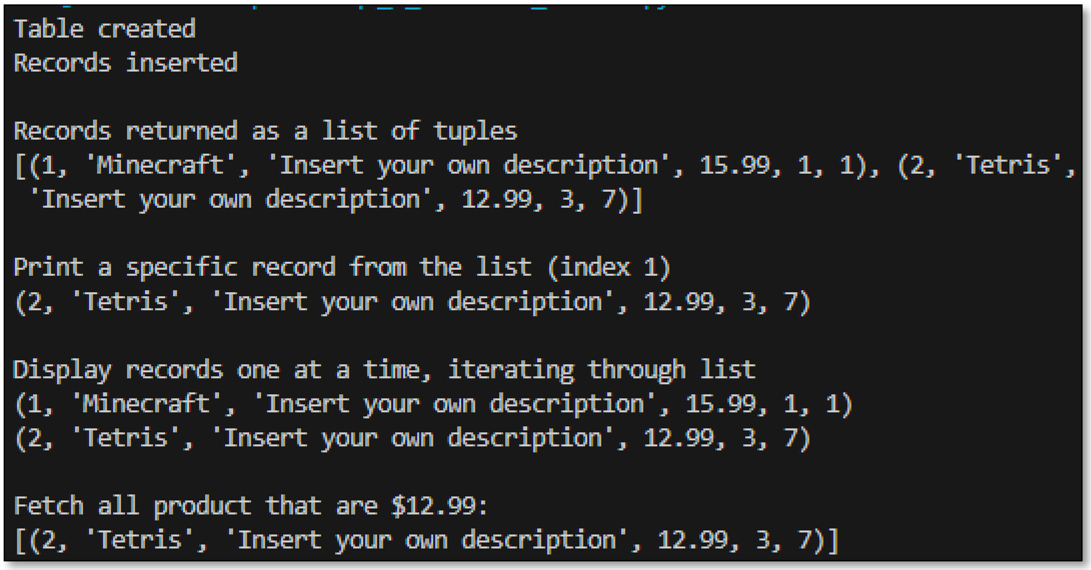
3. Copy **sql_3_tutorial_select.py** to **sql_4_tutorial_select.py**

4. Add the following.



```python
# ----------------------- SELECT RECORDS --------------------------------- #
# Fetch and display records from the 'products' table
records = game_shop.fetch_all_records()
print("\nRecords returned as a list of tuples")

# Print the entire list of records
print(records)

# Print a specific record from the list (index 1)
print("\nDisplay a specific record from the list (index 1)")

print(records[1])

# Iterate through the list and display records one at a time
print("\nDisplay records one at a time, iterating through list")

# Loop through returned list of tuples
for record in records:
    print(record)

# Fetch all records based on the product price
print("\nFetch all product that are $12.99: ")
record = game_shop.fetch_filtered_records(12.99)
print(record)
```

Example run:



```
Table created
Records inserted

Records returned as a list of tuples
[(1, 'Minecraft', 'Insert your own description', 15.99, 1, 1), (2, 'Tetris',
 'Insert your own description', 12.99, 3, 7)]

Print a specific record from the list (index 1)
(2, 'Tetris', 'Insert your own description', 12.99, 3, 7)

Display records one at a time, iterating through list
(1, 'Minecraft', 'Insert your own description', 15.99, 1, 1)
(2, 'Tetris', 'Insert your own description', 12.99, 3, 7)

Fetch all product that are $12.99:
[(2, 'Tetris', 'Insert your own description', 12.99, 3, 7)]
```

## Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.