2. Python SQLite Game Shop POS

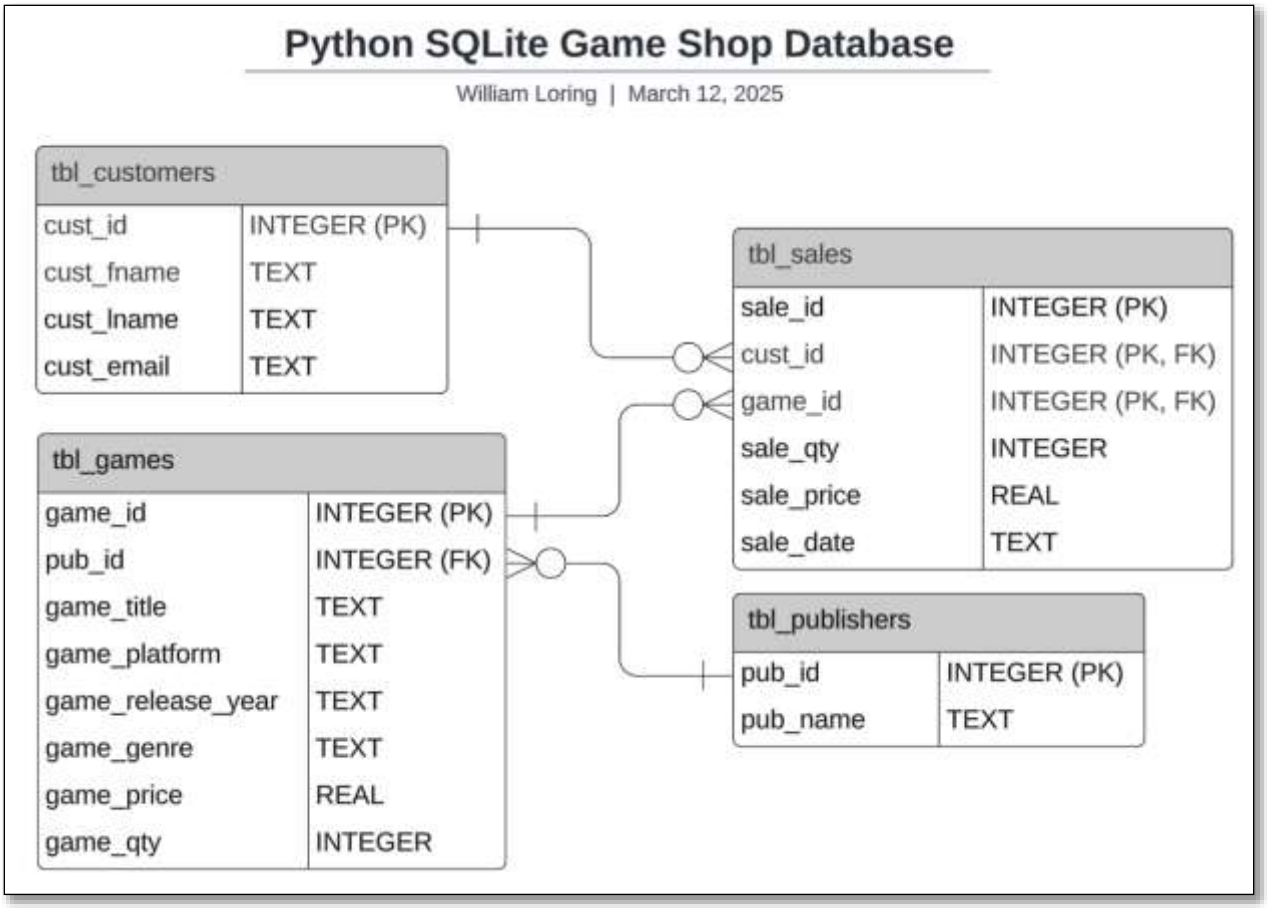
Contents

Time required: 90 minutes

A humorous and educational video on Python SQLite: SQLite in Python

- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.
- Remember to modify to create your own database

We redesigned our database in Part 1. Let's take a look at how to convert the ERD to a table.



Customers Table

- A customer can purchase many games.
- A game can be purchased by many customers.

Based on our new ERD, we need to add a customers table.

Create a new Python file named: **db_customers.py**

Use the same pattern that you used for your database that is the equivalent of the example database's games table.

Here is a link to last semester's assignments to help you out. GameShop SQLite Tutorial

You will have the following SQL scripts.

- CREATE_TABLE
- INSERT_RECORD

- FETCH_ALL_RECORDS
- FETCH_SPECIFIC_RECORDS
- FETCH_RECORD
- DELETE_RECORD
- UPDATE_RECORD

Sales Table

Based on our new ERD, we need to add a sales table as a bridge table between customers and games.

Create a new Python file named: **db_sales.py**

```
20  CREATE_TABLE = """
21      CREATE TABLE IF NOT EXISTS tbl_sales (
22      sale_id INTEGER PRIMARY KEY,  -- Unique identifier for each game
23      cust_id INTEGER,              -- Foreign key referencing customers table
24      game_id INTEGER,              -- Foreign key referencing games table
25      sale_qty INTEGER,             -- Number of copies sold
26      sale_price REAL,              -- Price of sale
27      sale_date TEXT,               -- Date of sale
28      FOREIGN KEY (cust_id) REFERENCES tbl_customers(cust_id),
29      FOREIGN KEY (game_id) REFERENCES tbl_games(game_id)
30      CONSTRAINT fk_customer
31          FOREIGN KEY (cust_id)
32          REFERENCES tbl_customers(cust_id)
33          ON DELETE CASCADE
34      CONSTRAINT fk_game
35          FOREIGN KEY (game_id)
36          REFERENCES tbl_games(game_id)
37          ON DELETE CASCADE
38  );
39  """
```

Here is a breakdown of the tbl_sales CREATE_TABLE SQL.

1. **Table Creation Statement**:

```
CREATE TABLE IF NOT EXISTS tbl_sales (
```

a. This line starts the creation of a new table named **tbl_sales** if it does not already exist.

2. **Column Definitions**:

   a. **sale_id INTEGER PRIMARY KEY**: Defines **sale_id** as an integer and the primary key, ensuring each sale has a unique identifier.

   b. **cust_id INTEGER**: Defines **cust_id** as an integer, which will act as a foreign key referencing the **tbl_customers** table.

   c. **game_id INTEGER**: Defines **game_id** as an integer, which will act as a foreign key referencing the **tbl_games** table.

   d. **sale_qty INTEGER**: Defines **sale_qty** as an integer, representing the number of copies sold.

   e. **sale_price REAL:** Defines sale_price as a float, representing the total sale.

   f. **sale_date TEXT**: Defines **sale_date** as a string, representing the date of the sale.

3. **Foreign Key Constraints**:

   a. **FOREIGN KEY (cust_id) REFERENCES tbl_customers(cust_id)**: Establishes a foreign key constraint on cust_id, linking it to the cust_id column in the tbl_customers table.

   b. **FOREIGN KEY (game_id) REFERENCES tbl_games(game_id)**: Establishes a foreign key constraint on game_id, linking it to the game_id column in the tbl_games table.

4. **Additional Constraints**:

```
CONSTRAINT fk_customer
    FOREIGN KEY (cust_id)
    REFERENCES tbl_customers(cust_id)
    ON DELETE CASCADE:
```

   a. Adds a named constraint **fk_customer** that enforces the foreign key relationship with **tbl_customers** and specifies that if a referenced customer is deleted, the corresponding sales records will also be deleted (ON DELETE CASCADE).

```
CONSTRAINT fk_game
    FOREIGN KEY (game_id)
    REFERENCES tbl_games(game_id)
    ON DELETE CASCADE:
```

b. Adds a named constraint **fk_game** that enforces the foreign key relationship with **tbl_games** and specifies that if a referenced game is deleted, the corresponding sales records will also be deleted (ON DELETE CASCADE).

This SQL code ensures that the tbl_sales bridge table is properly linked to the tbl_customers and tbl_games tables, maintaining referential integrity and cascading deletions when necessary.

Assignment Submission

- Attach the program files.
- Attach screenshots showing the successful operation of the program.
- Submit in Blackboard.