Individual Project 4
DS160-01
Introduction to Data Science
Spring 2022

Writing SQL Queries(50 points)

Goal: The goal of this project is to provide you practice for writing simple SQL queries.

Instructions: Create a new .sql file titled **IP4_XXX.sql**, where **XXX** are your initials; and rename this file to **IP4_XXX.docx**. Also create a GitHub repository titled **IP4_XXX** to which you can push both your .sql file and this document.

Attached to the assignment are two .sql files. HenryBooksCreate.sql which will create the database on you MySQL server and HenryBooksFill.sql which will fill it with data.

You will form the following queries for the Henry Books database and provide the statement and the output in this document. All of your SQL statements should also be in the .sql file. Use a comment (# mark) in your .sql file to number the statements according to this document. Each problem is marked with its point worth.

1. Retrieve the first name and last name of each author in the author relation. Order does not matter. **(1 points)**

   SQL Statement:

   ```
   SELECT firstname, lastname
   FROM author;
   ```

   Output:

2. Retrieve the title and book type in the book relation. Order does not matter. **(1 points)**

   SQL Statement:

   ```
   SELECT title, TYPE as Genre
   FROM book;
   ```

   Output:

3. Retrieve the publisherCode in the book relation. List each publisherCode only once in the result. Order does not matter. **(1 points)**

   SQL Statement:

   ```
   SELECT publisherCode
   FROM book;
   ```

   Output:

4. Retrieve the title and price of each book in the book relation. Further add a calculated column named 'discount' that shows the price the book with 25% discount. Order does not matter. Show the first five rows of the result (**LIMIT 5**). **(3 points)**

   SQL Statement:

   SELECT title, price, (price * 0.75) AS "25% off"
   FROM book
   LIMIT 5;

   Output:

5. Retrieve the title and price for any book whose price is higher than $20.00 in the book relation. Show the full result. **(3 points)**

   SQL Statement:

   SELECT title, price
   FROM book
   WHERE price>20;

   Output:

6. Retrieve the publisherName of all publishers that are in New York only in the publisher relation. Order does not matter. **(2 points)**

   SQL Statement:

   SELECT publishername, city
   FROM publisher
   WHERE city = "New York";

   Output:

7. Retrieve the publisherName of all publishers that are not in New York in the publisher relation. (use != for inequality). Order does not matter. Show the full result. **(3 points)**

   SQL Statement:

   SELECT publishername, city
   FROM publisher
   WHERE city != "New York";

   Output:

8. Retrieve the bookCode and onHand for each book for which a branch has between 2 and 4 copies in the inventory relation. **Use the BETWEEN keyword in this query.** Order does not matter. Show the full result. **(3 points)**

   SQL Statement:

   ```
   SELECT bookcode, onhand
   FROM inventory
   WHERE onhand BETWEEN 2 AND 4;
   ```

9. Retrieve a count of the number of books published by Penguin USA. Name the column 'Penguin Books'. Order does not matter. Show your full result. **(3 points)**

   SQL Statement:

   ```
   SELECT count(book.title) AS "Penguin Books"
   FROM book
   WHERE publishercode = "PE";
   ```

   Output:

10. Retrieve the number of books in the book relation whose prices is $20.00 or lower. Order does not matter. Show your full result. **(3 points)**

    SQL Statement:

    ```
    SELECT count(title)
    FROM book
    WHERE price <= 20;
    ```

    Output:

11. Retrieve all of the columns from the book and publisher relations in one result. Use aliases in your query and use the simple JOIN syntax (WHERE clause). Order does not matter. **(5 points)**

    SQL Statement:

    ```
    SELECT * #(* as in selecting all columns)
    FROM publisher, book
    JOIN publisher AS pub
    WHERE publisher.publisherCode = book.publisherCode;
    ```

    Output:

12. Rewrite the previous query using the ON keyword. **(5 points)**

    SQL Statement:

    ```
    SELECT * #(* as in selecting all columns)
    FROM publisher
    JOIN book ON
    ```

publisher.publisherCode = book.publisherCode;

Output:

13. Retrieve the title from the book relation and the city from the publisher relation using a JOIN query. Use aliases in your query. Order the result by title. **(5 points)**

    SQL Statement:

    ```
    SELECT title, city
    FROM book
    JOIN publisher ON
    book.publishercode = publisher.publishercode
    ORDER BY title;
    ```

    Output:

14. Retrieve the title from the book relation and branchNum and onHand from the inventory relation. Use aliases in your query. Order the result by title. **(5 points)**

    SQL Statement:

    ```
    SELECT title, onhand, branchnum
    FROM book, inventory
    ORDER BY title;
    ```

    Output:

15. Retrieve the title from the book relation and compute the number of copies of the title that all branches have on hand. Name this computed column 'Inventory' **Hint: You will need to join book and inventory and do an aggregate query.** Use aliases in your query. Order the result by the total number of copies of the book in descending order. Show the first two rows of your result. **(5 points)**

    SQL Statement:

    ```
    SELECT title, sum(onhand) as inventory
    FROM book
    JOIN inventory ON
    inventory.bookcode = book.bookcode
    GROUP BY title
    ORDER BY inventory DESC
    LIMIT 2;
    ```

    Output:

16. Retrieve the first name and last name from the author relation and the title from the book relation for all paperback books in the book relation. Order the result by the author last name and title. **(5 points)**

    SQL Statement:

    SELECT firstname, lastname, title
    FROM book, author
    WHERE paperback = "Y"
    ORDER BY lastname, title

    Output:

BONUS (5 points):

Retrieve the title from the book relation and the author lastName from the author relation. Order by author lastName. Use aliases in your query. **This will involve JOINING the book, author and wrote relations. (5 points)**

SQL Statement:

Output: