Reporting with SQL Part 4 Notes

Team Treehouse

June 5, 2020

1 Overview

2 Retrieving Results in a Particular Order

- ORDER BY
  - Allows to retrieve items in a particular order
  - **Syntax:** SELECT * FROM *table name* ORDER BY *column name* [ASC—DESC];

3 Retrieving Results in a Particular Order

- **Syntax:** SELECT * FROM *table name* ORDER BY *column name* [ASC—DESC], *column 2 name* [ASC—DESC], ..., *column n name* [ASC—DESC];

Example:

```
1   SELECT * FROM books ORDER BY title ASC;
2
3
4   SELECT * FROM products WHERE name = "Sonic T-Shirt" ORDER BY
    stock_count DESC;
5
6
7   SELECT * FROM users ORDER BY signed_up_on DESC;
8
9
10  SELECT * FROM countries ORDER BY population DESC;
11
12
```

```
13  SELECT * FROM books ORDER BY    genre ASC,
14                                  title ASC;
15
16
17  SELECT * FROM books ORDER BY    genre ASC,
18                                  year_published DESC;
19
20
21  SELECT * FROM users ORDER BY    last_name ASC,
22                                  first_name ASC;
23
```

4 Exercise 1

- Solution included in *exercise_1.sql*

5 Retrieving Results in a Particular Order

- LIMIT
  - **Syntax (SQLite, PostgreSQL, MySQL):** SELECT *columns name* FROM *table name* ... LIMIT *# of rows*;
  - Must be placed at the end

6 Exercise 2

- Solution included in *exercise_2.sql*

7 Paging Thrugh Results

- OFFSET
  - **Syntax 1 (SQLite, PostgreSQL, MySQL):** SELECT ¡columns¿ FROM *table name* LIMIT *# of rows* OFFSET *skipped rows*;

  - **Syntax 2 (SQLite, PostgreSQL, MySQL):** SELECT ¡columns¿ FROM *table name* LIMIT *skipped rows*, *# of rows*;
  - Is based on number of rows, and NOT by pages (i.e. LIMIT 10 OFFSET 10 is on page 2)
  - Is useful when creating multi-page reports, blog archive, or listing search results

8 Exercise 3

- Solution included in *exercise_3.sql*

9 Practice Session

- OFFSET
  - **Syntax 1 (SQLite, PostgreSQL, MySQL):** SELECT ¡columns¿ FROM *table name* LIMIT *# of rows* OFFSET *skipped rows*;
  - **Syntax 2 (SQLite, PostgreSQL, MySQL):** SELECT ¡columns¿ FROM *table name* LIMIT *skipped rows*, *# of rows*;
  - Is based on number of rows, and NOT by pages (i.e. LIMIT 10 OFFSET 10 is on page 2)
  - Is useful when creating multi-page reports, blog archive, or listing search results

10 Quiz 1

1. Please fill in the correct answer in each blank provided below.

   I would type in ___ at the end of an‘ORDER BY‘ clause to order in ascending order.

   Answer: ASC

2. Can you guess what the following query would do?

```
1    SELECT * FROM passport_holders WHERE name = "Lauren" LIMIT 0, 50;
2
```

A. Return the first 50 results for the query SELECT FROM passport_holders WHERE name = "Lauren"

B. Return the last 50 results for the query SELECT FROM passport_holders WHERE name = "Lauren"

Answer: A

3. Please fill in the correct answer in each blank provided below.

   I would type in ___ at the end of an'ORDER BY' clause to order in ascending order.

   Answer: DESC

4. Back in the teachers notes I included a reference to how other databases LIMIT results. Which of the following is a valid way to order results in another relational database system?

   A. SELECT name FROM people TOP 50;

   B. SELECT TOP 50 name FROM people;

   Answer: B

5. When using the ORDER BY keywords, what is the default order that the results will appear in?

   For example: SELECT name, population FROM cities ORDER BY name;

   A. Descending order

   B. Ascending order

   C. The order it was entered in the database

   Answer: B