

# 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 Through 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*