sql_joins

November 23, 2020

Using SQL JOIN to Query Multiple Tables

1 1. `JOIN` Allows us to Expand our Query's Result Set

- If you'll recall, we used `leftjoin()` from *dplyr* in R
  - This function was inspired by SQL
  - Frequently hear the term "SQL-style join"

1.1 1.1 Some Terminology

- Joins are performed by using a given column that appears in both tables being joined
- Frequently, the column being used to join on is "primary key" or a "foreign key"
  - A primary key is the column that acts as the unique identifier for a given record (i.e., row)
  - A foreign key is a column in a table that is associated with a primary key in another table
- Caveat:
  - Both primary keys and foreign keys can be a combination of columns (ignore this for now)

1.2 1.2 Example `JOIN`

- Recall our `amzn.db` database has several tables
  - `orders`
  - `products`
  - `customer`
- Suppose we want to know what each customer paid for their order
- This will invole a query across `orders` and `products`, since the price information is in the `products` table

```
In [2]: %load data/amzn.db
```

1.2.1 1.2.1 Quick Refresher on SQL SELECT

```
In [3]: SELECT
            customer,       -- these are the columns (i.e., "fields") we want
            date,
            product_id
        FROM
            orders          -- this is the table we pull from
        ;
```

```
Out[3]: +----------+------------+------------+
        | customer | date       | product_id |
        +----------+------------+------------+
        | lee      | 2018-12-23 | 44         |
        +----------+------------+------------+
        | smith    | 2020-03-12 | 33         |
        +----------+------------+------------+
        | jones    | 2019-05-01 | 212        |
        +----------+------------+------------+
        | yang     | 2020-09-12 | 12         |
        +----------+------------+------------+
        | guerra   | 2020-08-03 | 12         |
        +----------+------------+------------+
        | diaz     | 2020-11-28 | 123        |
        +----------+------------+------------+
        | riley    | 2019-05-18 | 232        |
        +----------+------------+------------+
        | chan     | 2018-10-03 | 28         |
        +----------+------------+------------+
```

1.2.2 1.2.2 Recall the SELECT * Idiom

- This will show you all rows and columns for a given table(s)
- Not usually advisable

```
In [6]: SELECT
            *
        FROM
            products
        ;
```

```
Out[6]: +------------+-----------+--------+-------------+
        | product_id | product   | price  | supplier_id |
        +------------+-----------+--------+-------------+
        | 44         | rake      | 19.99  | 65656       |
        +------------+-----------+--------+-------------+
        | 33         | shoe horn | 5.99   | 3434        |
        +------------+-----------+--------+-------------+
        | 22         | potato    | 0.99   | 65656       |
        +------------+-----------+--------+-------------+
        | 12         | bike      | 123.5  | 232         |
        +------------+-----------+--------+-------------+
        | 123        | table     | 78.55  | 54545       |
        +------------+-----------+--------+-------------+
        | 232        | cup       | 2.5    | 4333        |
        +------------+-----------+--------+-------------+
        | 28         | ball      | 5.5    | 2323        |
        +------------+-----------+--------+-------------+
        | 22         | pencil    | 2.99   | 3232        |
        +------------+-----------+--------+-------------+
        | 11         | teapot    | 12.49  | 6565        |
        +------------+-----------+--------+-------------+
        | 13         | fork      | 1.99   | 86787       |
        +------------+-----------+--------+-------------+
        | 14         | shoelace  | 0.5    | 8787        |
        +------------+-----------+--------+-------------+
        | 555        | hammer    | 17.49  | 7878        |
        +------------+-----------+--------+-------------+
        | 66         | door      | 159.99 | 9889        |
        +------------+-----------+--------+-------------+
```

1.3 1.3 Motivating Example

- Suppose we want to obtian the total cost of every order
- Note that our `orders` table does not contain price information
- The `orders` table does have a `product_id` field
  - the `product_id` field can be used to link (i.e., "join") to the `product` table

In [7]:
```
SELECT
    customer,
    quantity,
    product,
    price
FROM
    orders
    JOIN products ON orders.product_id = products.product_id
;
```

Out[7]:
```
+----------+----------+-----------+-------+
| customer | quantity | product   | price |
+----------+----------+-----------+-------+
| lee      | 1        | rake      | 19.99 |
+----------+----------+-----------+-------+
| smith    | 1        | shoe horn | 5.99  |
+----------+----------+-----------+-------+
| yang     | 1        | bike      | 123.5 |
+----------+----------+-----------+-------+
| guerra   | 2        | bike      | 123.5 |
+----------+----------+-----------+-------+
| diaz     | 2        | table     | 78.55 |
+----------+----------+-----------+-------+
| riley    | 4        | cup       | 2.5   |
+----------+----------+-----------+-------+
| chan     | 1        | ball      | 5.5   |
+----------+----------+-----------+-------+
```

1.3.1 1.3.1 Using Table Aliases

- It is more idiomatic to alias our table names to something short (e.g., `ord` for `orders`)
- We can then use that in the `SELECT` section of our query

In [8]:
```
SELECT
    ord.customer,
    ord.quantity,
    ord.product_id,
    pro.product,
    pro.price
FROM
    orders          AS ord
    JOIN products AS pro ON ord.product_id = pro.product_id
;
```

Out[8]:
```
+----------+----------+------------+-----------+-------+
| customer | quantity | product_id | product   | price |
+----------+----------+------------+-----------+-------+
| lee      | 1        | 44         | rake      | 19.99 |
+----------+----------+------------+-----------+-------+
| smith    | 1        | 33         | shoe horn | 5.99  |
+----------+----------+------------+-----------+-------+
| yang     | 1        | 12         | bike      | 123.5 |
+----------+----------+------------+-----------+-------+
| guerra   | 2        | 12         | bike      | 123.5 |
+----------+----------+------------+-----------+-------+
| diaz     | 2        | 123        | table     | 78.55 |
+----------+----------+------------+-----------+-------+
| riley    | 4        | 232        | cup       | 2.5   |
+----------+----------+------------+-----------+-------+
| chan     | 1        | 28         | ball      | 5.5   |
+----------+----------+------------+-----------+-------+
```

1.3.2 1.3.2 Doing Math in `SELECT` Section

In [9]:
```
SELECT
    ord.customer,
    ord.quantity,
    ord.product_id,
    pro.product,
    pro.price,
    (pro.price * ord.quantity) AS total_cost
FROM
    orders        AS ord
    JOIN products AS pro ON ord.product_id = pro.product_id
;
```

Out[9]:
```
+----------+----------+------------+-----------+-------+------------+
| customer | quantity | product_id | product   | price | total_cost |
+----------+----------+------------+-----------+-------+------------+
| lee      | 1        | 44         | rake      | 19.99 | 19.99      |
+----------+----------+------------+-----------+-------+------------+
| smith    | 1        | 33         | shoe horn | 5.99  | 5.99       |
+----------+----------+------------+-----------+-------+------------+
| yang     | 1        | 12         | bike      | 123.5 | 123.5      |
+----------+----------+------------+-----------+-------+------------+
| guerra   | 2        | 12         | bike      | 123.5 | 247.0      |
+----------+----------+------------+-----------+-------+------------+
| diaz     | 2        | 123        | table     | 78.55 | 157.1      |
+----------+----------+------------+-----------+-------+------------+
| riley    | 4        | 232        | cup       | 2.5   | 10.0       |
+----------+----------+------------+-----------+-------+------------+
| chan     | 1        | 28         | ball      | 5.5   | 5.5        |
+----------+----------+------------+-----------+-------+------------+
```

1.3.3 1.3.3 Filtering Result using `WHERE`

- Suppose we only care about order with 2 or more items

In [10]:
```
SELECT
    ord.customer,
    ord.quantity,
    ord.product_id,
    pro.product,
    pro.price,
    (pro.price * ord.quantity) AS total_cost
FROM
    orders        AS ord
    JOIN products AS pro ON ord.product_id = pro.product_id
WHERE
    ord.quantity > 1
;
```

Out[10]:
```
+----------+----------+------------+---------+-------+------------+
| customer | quantity | product_id | product | price | total_cost |
+----------+----------+------------+---------+-------+------------+
| guerra   | 2        | 12         | bike    | 123.5 | 247.0      |
+----------+----------+------------+---------+-------+------------+
| diaz     | 2        | 123        | table   | 78.55 | 157.1      |
+----------+----------+------------+---------+-------+------------+
| riley    | 4        | 232        | cup     | 2.5   | 10.0       |
+----------+----------+------------+---------+-------+------------+
```

1.3.4 1.3.4 Filterin on Multiple Criteria

- Recall that we can use the `WHERE` clause to filter according to any number of criteria
- Suppose we want either orders with more than 2 items *or* those with a *particular* item

In [12]:
```
SELECT
    ord.customer,
    ord.quantity,
    ord.product_id,
    pro.product,
    pro.price,
    (pro.price * ord.quantity) AS total_cost
FROM
    orders        AS ord
    JOIN products AS pro ON ord.product_id = pro.product_id
WHERE
    ord.quantity > 1
    OR pro.product IN ('rake', 'bike')
;
```

Out[12]:
```
+----------+----------+------------+---------+-------+------------+
| customer | quantity | product_id | product | price | total_cost |
+----------+----------+------------+---------+-------+------------+
| lee      | 1        | 44         | rake    | 19.99 | 19.99      |
+----------+----------+------------+---------+-------+------------+
| yang     | 1        | 12         | bike    | 123.5 | 123.5      |
+----------+----------+------------+---------+-------+------------+
| guerra   | 2        | 12         | bike    | 123.5 | 247.0      |
+----------+----------+------------+---------+-------+------------+
| diaz     | 2        | 123        | table   | 78.55 | 157.1      |
+----------+----------+------------+---------+-------+------------+
| riley    | 4        | 232        | cup     | 2.5   | 10.0       |
+----------+----------+------------+---------+-------+------------+
```