

Homework | module 2 > week 7 > day 17

Topics covered: JOINS

Standard Exercise:

For this homework session, you will work in Google BigQuery with the data set called `thelook_ecommerce` in the `bigquery-public-data` project.

1. To get acquainted with the structure of the 7 tables contained in the data set, start by designing its Entity Relationship (ER) Diagram. I already created a skeleton with the 7 tables and their variables which you can find at [this Lucidchart link](#). Follow the link, go to File > Duplicate to make a copy of the diagram which you can now edit. Your task is to:
 - a. Add the PK / FK keys in the left column of each table (like in the events table which I have pre-filled)
 - b. Add the relations between each table using lines to connect the PK <-> FK variables
 - c. Pro level: you may have noticed that the lines connecting each table have different endings, learn more about those symbols at [this page](#).
2. Use a Left Join to connect the orders table to the users table and return all variables from both tables.

SELECT

*,

FROM `bigquery-public-data.thelook_ecommerce.orders` as orders

LEFT JOIN bigquery-public-data.thelook_ecommerce.users as users

on orders.user_id = users.users.id

3. Return a table containing all the customers' *full names* in the first column and the count of how many orders they made in their history in the second column. Sort them by the second column in descending order. What is the name of the customer that made the highest number of orders?

```
SELECT users.id, concat(users.first_name, " ", users.last_name) as full_name,
count(orders.order_id) as cnt_orders
FROM `bigquery-public-data.thelook_ecommerce.orders` as orders
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` as users
on orders.user_id = users.users.id
group by users.id, full_name
order by cnt_orders desc;
```

4. Now modify the query to show only the customers that made more than ten orders. How many are they (you can count the number of rows)?

```
SELECT id, concat(users.first_name, " ", users.last_name) as full_name,
count(orders.order_id) as cnt_orders
FROM `bigquery-public-data.thelook_ecommerce.orders` as orders
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` as users
on orders.user_id = users.id
group by id, full_name
having count(orders.order_id) > 3
order by cnt_orders desc;
```

5. [Not a real question] By the way, what's going on with the Smith family from the last query? 🤔

6. Your boss wants to know if there is a relationship between the status of an order and the age bracket of a user. Write a query that returns a pivot table showing the order status by row and four age brackets as separate columns like in the screenshot below:

Row	status	under20	age20_39	age40_59	age60_79	over80
1	Complete	2229	5387	5274	2747	0
2	Returned	269	637	638	297	0
3	Cancelled	113	315	297	170	0

```

select orders.status,
sum(case when users.age < 20 then 1 else 0 end) as `under20`,
sum(case when users.age >= 20 and users.age < 40 then 1 else 0 end) as `age20_39`,
sum(case when users.age >= 40 and users.age < 60 then 1 else 0 end) as `age40_59`,
sum(case when users.age >= 60 and users.age < 80 then 1 else 0 end) as `age60_79`,
sum(case when users.age >= 80 then 1 else 0 end) as `over80`
from `bigquery-public-data.thelook_ecommerce.orders` orders
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` users
on orders.user_id = users.id
group by orders.status ;

```

Advanced Exercise (optional):

Using the same data set, answer the following questions:

1. Use a Left Join to connect the order-item table and the products table;
what are the top 3 products in terms of average margin?

```

SELECT name,
avg(products.retail_price) - avg(products.cost) as margin_avg
FROM `bigquery-public-data.thelook_ecommerce.order_items` items
LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` products
on items.product_id = products.id
group by name
order by margin_avg desc;

```

2. Using one query, join together the users table to the order-item and
products tables; show all the variables from the three tables.

```

SELECT items.*, products.*, users.* --* <-- ( mi prendi 3 colonne di 3 tabelle differenti)
FROM `bigquery-public-data.thelook_ecommerce.order_items` as items
LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` as products
on items.product_id = products.id
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` as users
on items.user_id = users.id;

```

3. Produce a list of all the customers and the average margin per order they generated, sorted in descending order. Who generated the highest margin per order?

```
SELECT concat(users.first_name, " ", users.last_name) as full_name,  
       (SUM(products.retail_price) - SUM(products.cost))/count(distinct order_id) as  
margin_per_order  
FROM `bigquery-public-data.thelook_ecommerce.order_items` items  
LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` products  
on items.product_id = products.id  
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` users  
on items.user_id = users.id  
group by full_name  
order by margin_per_order desc;
```

4. From the query above, produce the same list but with customers that made at least five orders. Who generated the highest margin per order?

```
SELECT concat(users.first_name, " ", users.last_name) as full_name,  
       (SUM(products.retail_price) - SUM(products.cost))/count(distinct order_id) as  
margin_per_order  
FROM `bigquery-public-data.thelook_ecommerce.order_items` items  
LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` products  
on items.product_id = products.id  
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` users  
on items.user_id = users.id  
group by full_name  
having count(distinct order_id) > 5  
order by margin_per_order desc;
```

5. Write a query that shows the combination of all customers that live in the same city and have different genders.

```
SELECT a.city,  
       concat(a.first_name, " ", a.last_name) as user1,  
       concat(b.first_name, " ", b.last_name) as user2  
FROM `bigquery-public-data.thelook_ecommerce.users` a  
INNER JOIN `bigquery-public-data.thelook_ecommerce.users` b  
on a.id > b.id and a.city = b.city and a.gender <> b.gender  
where a.state = 'United States'  
order by a.city, user1, user2;
```

