The Look Ecommerce

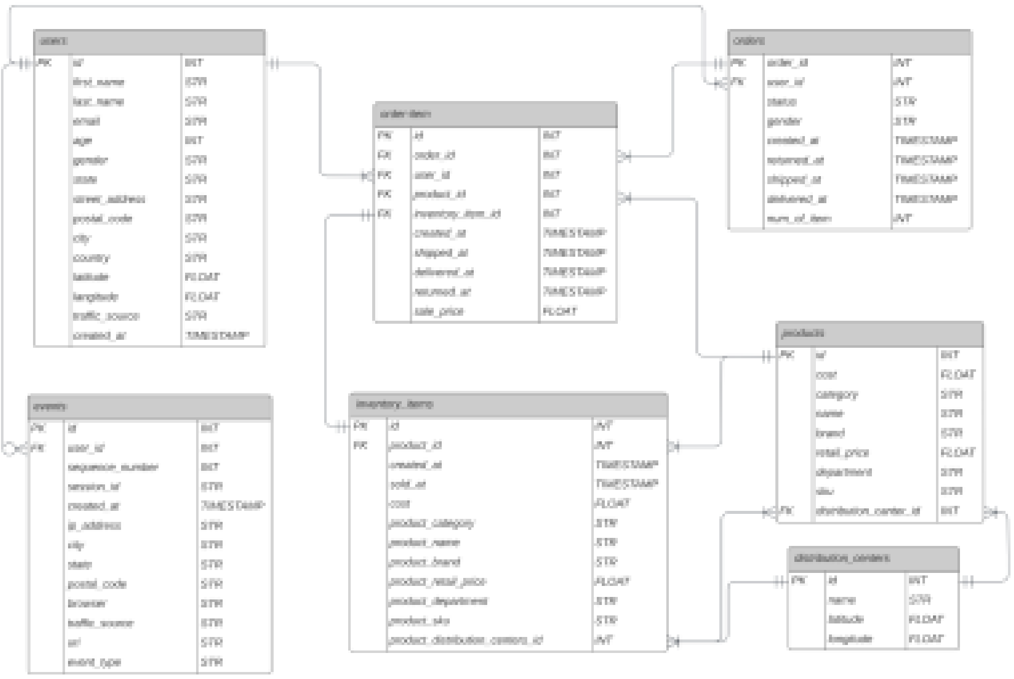
You will work in Google BigQuery with the data set called thelook_ecommerce.

To get acquainted with the structure of the 7 tables contained in the data set, start by designing its Entity Relationship (ER) Diagram.

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



2. Use a Left Join to connect the orders table to the users table and return all variables from both tables.

```
SELECT a.*, b.*
FROM `bigquery-public-data.thelook_ecommerce.orders` a
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` b
 on a.user_id = b.id
```

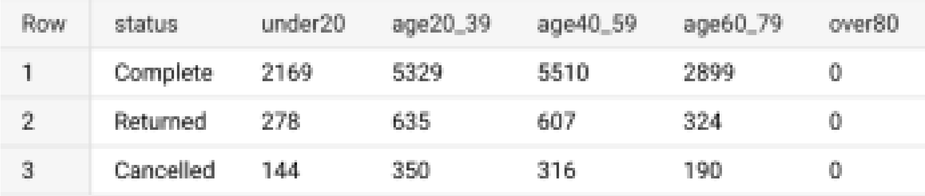
3. Return a table containing all the customers' *full names* in the first column and the count of how many orders they made in their history in the second column. Sort them by the second column in descending order. What is the name of the customer that made the highest number of orders?

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

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

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

5. 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	2169	5329	5510	2899	0
2	Returned	278	635	607	324	0
3	Cancelled	144	350	316	190	0

```
select a.status,
 sum(case when b.age < 20 then 1 else 0 end) as `under20`,
 sum(case when b.age >= 20 and b.age < 40 then 1 else 0 end) as
 `age20_39`, sum(case when b.age >= 40 and b.age < 60 then 1 else 0 end)
 as `age40_59`, sum(case when b.age >= 60 and b.age < 70 then 1 else 0
 end) as `age60_79`, sum(case when b.age >= 80 then 1 else 0 end) as
 `over80`
from `bigquery-public-data.thelook_ecommerce.orders` a
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` b
 on a.user_id = b.id
group by a.status
```

Advanced Exercise

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(b.retail_price) - avg(b.cost) as margin_avg
FROM `bigquery-public-data.thelook_ecommerce.order_items` a
LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` b
on a.product_id = b.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 a.*, b.*, c.*
FROM `bigquery-public-data.thelook_ecommerce.order_items` a
LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` b
on a.product_id = b.id
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` c
on a.user_id = c.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(c.first_name, " ", c.last_name) as full_name,
 (SUM(b.retail_price) - SUM(b.cost))/count(distinct order_id) as
margin_per_order FROM
`bigquery-public-data.thelook_ecommerce.order_items` a
LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` b
on a.product_id = b.id
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` c
on a.user_id = c.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(c.first_name, " ", c.last_name) as full_name,
 (SUM(b.retail_price) - SUM(b.cost))/count(distinct order_id) as
margin_per_order FROM
`bigquery-public-data.thelook_ecommerce.order_items` a
LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` b
on a.product_id = b.id
LEFT JOIN `bigquery-public-data.thelook_ecommerce.users` c
on a.user_id = c.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
```