**1.Goal: Here we use users table to pull a list of user email addresses. Edit the query to pull email addresses, but only for non-deleted users.**

SELECT id,email\_address

FROM dsv1069.users

where deleted\_at IS NULL

**2. Goal: Use the items table to count the number of items for sale in each category**

select category,count(distinct id)

from dsv1069.items

group by category

**3. --Goal: Select all of the columns from the result when you JOIN the users table to the orders table**

select \*

from dsv1069.users u

inner join dsv1069.orders o

on u.id=o.user\_id

**4. Goal: Check out the query below. This is not the right way to count the number of viewed\_item events. Determine what is wrong and correct the error. Starter Code:**

SELECT

COUNT(event\_id) AS events

FROM dsv1069.events

WHERE event\_name = ‘view\_item’

**Correct Query :**

SELECT

COUNT(event\_id) AS events

FROM dsv1069.events

where event\_name like '%view\_item%'

--group by event\_name

**5.Goal:Compute the number of items in the items table which have been ordered**

SELECT

COUNT(event\_id) AS events

FROM dsv1069.events

where event\_name like '%view\_item%'

**6. --Goal: For each user figure out IF a user has ordered something, and when their first purchase was.**

select u.id, min(o.paid\_at)

from dsv1069.users u

left join dsv1069.orders o

on o.user\_id=u.id

group by u.id

**7. Exercise 7: --Goal: Figure out what percent of users have ever viewed the user profile page, but this query isn’t right. Check to make sure the number of users adds up, and if not, fix the query.**

select (case when first\_view is NULL then false

else true end) as has\_viewed\_profile,

count(user\_id) as users

from

(select

u.id as User\_id,Min(e.event\_time) as first\_view

from dsv1069.users u

left outer join dsv1069.events e

on e.user\_id =u.id and e.event\_name='view\_user\_profile'

group by u.id) first\_profile\_views

GROUP by has\_viewed\_profile