Congratulations! You passed!

Grade received 100%

To pass 80% or higher

Go to next item

Reorder and Connect Tables

Total points 5

1. Let's suppose we want to write a query to answer both of these questions: **1 / 1 point**

- How many items have been purchased?
- How many items do we have?

Please choose the best set of columns for a final query that would answer these questions:

- (●) Item_count

  Items_ever_purchased_count

- ( ) Item_count

  User_count

- ( ) User_count

  Item_id

  View_count

- ( ) Category

  item_count

Correct

We don't need to include unnecessary information like the number of users, or the views to answer our question. If it turns out you actually want to ask different questions, decide that before you start coding.

2. Please select all tables that will be necessary answer both of these questions: **1 / 1 point**

- How many items have been purchased?
- How many items do we have?

- [x] Items

Correct
This table will help answer both questions as it counts items.

- [x] Orders

Correct
This table will help answer both questions as it counts items purchased.

- [ ] Users
- [ ] View_items
- [ ] Events

3. We've decided to only use the items and orders tables to answer the following questions: **1 / 1 point**

- How many items have been purchased?
- How many items do we have?

Can we compute the columns Items_count, items_ever_purchased_count without a subquery?

- ( ) No
- (•) Yes

Correct

4. We've decided to answer the following questions: **1 / 1 point**

- How many items have been purchased?
- How many items do we have?

Which of the following queries will answer both those questions without further computation?

◉ SELECT

COUNT(DISTINCT items.id)

AS items_count,

COUNT(DISTINCT orders.item_id)

AS items_ever_purchased_count

FROM

dsv1069.items

LEFT OUTER JOIN

dsv1069.orders

ON

items.id = orders.item

○ SELECT

```
 COUNT(items.id)
AS items_count,
 COUNT(orders.item_id)
AS items_ever_purchased_count
FROM
 dsv1069.items
JOIN
 dsv1069.orders
ON
 items.id = orders.item
GROUP BY
 items.id
```

○ SELECT

```
 COUNT(DISTINCT items.id)
AS items_count,
 COUNT(DISTINCT orders.item_id)
AS items_ever_purchased_count
FROM
 dsv1069.items
JOIN
 dsv1069.orders
ON
 items.id = orders.item
```

✓ **Correct**

This query will answer both questions without any further computation.

5. In the previous question we decided that the query below could answer the questions : **1 / 1 point**

- How many items have been purchased?
- How many items do we have?

```
--------------------------------------

SELECT

 COUNT(DISTINCT items.id)       AS items_count,

 COUNT(DISTINCT orders.item_id) AS items_ever_purchased_count

FROM

 dsv1069.items

LEFT OUTER JOIN

 dsv1069.orders

ON

 items.id = orders.item

--------------------------------------
```

Is this the only possible way to answer the question? Justify your answer.

No it is not the only way but it is the more simple.

Also a subquery could have been utilized