Problem # attempted:

3

Your answer:

This doesn't show any output because Species and Animal have no common primary keys as Animals has a multivariable primary key in it thus there is no matching.

Instructor's answer:

No common attribute within the name columns and no power to tell what column to join on due to the use of natural join.

What did you learn (or what were you doing wrong)? Please be articulate.

There was something common but they weren't any matching attributes. No multivariable key present and the natural join couldn't function properly due to the missing attributes.

Problem # attempted: 5

Your answer:

T1 is a join between Species and Animals on the Name field thus it gives us all columns for the three rows t that match on the name column. T2 adds in all of the fiends that match for table zoo in the combined tables in t1

Instructor's answer:

Has 8 columns and 3 rows after the first join. And then join with zoo's which was renamed t2.

What did you learn (or what were you doing wrong)? Please be articulate.

I learned that I needed to be more clear about using the alias name in my answer instead of the table name as the code had renamed the table zoo's to t2.

7

Your answer:

Select "Name", "Species"
From 'worldanimals"."Animals" where "YearBorn" = (Select "YearBorn" from
"worldanimals"."Animals"
Inner join (select "City"
From "worldanimals"."Zoos"
Where "Country" = 'USA') as cityCountry on "worldanimals"."Animals"."Zoo" = city
Where "YearBorn > '2005');

Instructor's answer:

Select name, species
From animals join zoos
On animals.zoo = zoos.city
Where yearBorn > 2005 and country = 'USA';

What did you learn (or what were you doing wrong)? Please be articulate.

I very much over complicated this. I also learned that I should remove inserts that were not originally in the data set when submitting my answers. I see now that even though I could get the correct output that I shouldn't need to use the subqueries like I did as joining the two tables is far easier and allows for my where filtering to be accomplished with less complication.

Problem # attempted:

10

Your answer:

Select "City", "worldanimals"."zoos".country", "Acres" from "worldanimals"."zoos" Inner join (Select "Country", Max("Acres") as macres from "worldanimals"."zoos" group by "Country") as maxacres on "worldanimals"."Zoos"."Acres" = maxacres.macres AND "worldanimals"."Zoos"."Country" = maxacores."Country" where "Acres" > 25;

Instructor's answer:

Select City, country, acres
From zoos
Where acres IN (Select Max(Acres) from zoos group by country having MAX(acres) > 25);

What did you learn (or what were you doing wrong)? Please be articulate.

Using a subquery instead of a join on this simplified the problem a lot. It meant that I didn't have to use aliases and specific joins to get the answers. Using group by also eliminates part of my join as well. Over thinking is my specialty. ☺

	11

Problem # attempted:

Your answer:

> Select "City"
> From 'worldanimals"."zoos"
> Except (select "worldanimals"."Zoos"."City" from "worldanimals"."Zoos" left join "worldanimals"."Animals" on "worldanimals"."Zoos"."City" = "worldanimals"."Animals"."Zoos" where "worldanimals"."Animals"."Species" = 'Giant Panda');

Instructor's answer:

> Select city From zoos
> Except (Select zoo
> From animals where Species = 'Giant Panda');

What did you learn (or what were you doing wrong)? Please be articulate.

> I didn't need the join at all, for some reason I figured that I needed to include that zoos table for san Diego in the code for the except. I did not. Learned that I over think and that I had the correct idea. Also learned that the select clause before and after the except are separate and function independently of each other.

13	

Problem # attempted:

Your answer:

> Select "worldanimals"."Animals"."Name"
> From "worldanimals"."Animals"
> Inner join (select "Name" from "worldanimals"."Species" where "IsEndangered" = 'true'
> Group by "Name") as animalSpecies on "worldanimals"."Animals"."Species" =
> animalSpecies."Name"
> Inner join (Select "City", "Country" from "worldanimals"."Zoos" where "Country" = 'USA'
> group by "City","Country") as animalCountry on "worldanimals"."Animals"."Zoo" =
> animalCountry."City";

Instructor's answer:

> Select animals.name
> From species join animals ON species.name = animal.species
> Join zoos ON zoos.city = animals.zoo
> Where country = 'USA' AND isEndangered = 'Y';

What did you learn (or what were you doing wrong)? Please be articulate.

After watching your video I realized that I was both attempting to filter the output while I was joining the tables. I should have focused on doing one at a time thus making my answer less complex. I also input the isEndanged field as a Boolean value thus put true instead of Y. I'm not sure if the Boolean value allows for both depending on which Boolean type you use for the database. Will need to research that further.