


☐

I'm not robot


reCAPTCHA

Continue

Khan academy sql challenge customer's orders

This module contained many exercises relating to mergers. Link tables together with primary and foreign keys. It's very important and important, it needs to be learned. Joining tables using JOIN The following challenge required a basic JOIN of a person's table and hobbies to display people's names along with their hobby in one table. The secondary table (hobbies) must have had a foreign key, in this case person_id linked to the primary id table. Joining related tables with LEFT JOIN LEFT OUTER JOIN IS A VERY USEFUL type of merger. This merger allows me to include every row that refers to an attribute that I specify, whether their value is NULL or not. In the challenge shown below, you can see LEFT OUTER JOIN used to search for customer names, emails and total money spent on orders, even if they haven't made any purchases yet. Self Join This area has given me so many problems are the ensuing in the ensuing I have managed to figure out after a long time. Self joined is used when we want to reference an attribute in a table to another row within the same table. In the example below, I linked the Harry Potter films to their sequels inside the same table. THE LEFT OUTER JOIN is used first to include each film, regardless of whether the following are available or not. Secondly, the text id from the main table must be linked to an id from the aliases table (called x). And a pretzel. Combining several joins This challenge has been quite difficult to concede. She needed two contacts. The first reunion was simple enough, the names were on the list along with hobbies. The second joining, however, was very complicated, much like the next challenge above. I had to connect people with their friends using a table of friends and self join. You cannot perform this action at this time. You signed in with another tab or window. Reload to refresh the seed. You have unsumoreed in another tab or window. Reload to refresh the seed. We use optional third-party analytics cookies to understand how you use GitHub.com so that we can build better products. More. We use optional third-party analytics cookies to understand how you use GitHub.com so that we can build better products. You can always update your selection by clicking Cookie settings at the bottom of the page. For more information, see our privacy statement. To perform essential features of the website, we use essential cookies, e.g. More We always actively use analytical cookies to understand how you use our websites so that we can make them better, e.g. they are used to collect information about the pages you visit and how many clicks you need to perform the task. Read more Share intro in SQL: Querying and managing data is a challenge patch. To earn money, the user must complete all 13 software challenges in the Intro to SQL: Query and manage a data theme. This badge can be Here at Khan Academy. The Course Course is divided into 4 sections: SQL Basics The user learns the basics of creating tables and selecting data in different ways. Challenge: Database list of books Challenge: Box Office hits database Challenge: TODO list database stats More advanced SQL queries User learns how to perform more advanced SQL queries using AND/OR, IN, LIKE, HAVING, and more. Sql relational queries Learn how to store linked data in multiple tables and use mergers (internal mergers, internal mergers, and self-management). Change databases with SQL Learn how to update data, delete data, and change table schemas with SQL UPDATE, DELETE, ALTER, and DROP commands. Challenge: Dynamic Documents Challenge: Changes to Clothing Similar Badges Category: CS Challenge Badges Community is available under CC-BY-SA, unless otherwise stated. Guys, can someone help me with this? My code shows what is expected, but I do not allow me to go to STEP 3. SELECT customers.name, customers.email, orders.item, SUM(price) as orders FROM CUSTOMERS LEFT OUTER JOIN orders ON customers.id = orders.customer_id GROUP BY NAME ORDER BY PRICE DESC; Step 2 is a syntax error...I gave step 1, which works by the way. SELECT name, email, item, price FROM customersLEFT OUTER JOIN ordersON customers.id = orders.customer_id When I add in Step 2, there is a syntax error. SELECT customers.name, customers.email, SUM(price) as orders FROM customers LEFT OUTER JOIN orders ON customers.id = orders.customer_id GROUP BY name ORDER BY price DESC; When I set step 2 myself, it works fine. What am I supposed to do? If you see this message, it means that we are having trouble uploading external resources to our website. If you are behind a web filter, make sure *.kastatic.org and *.kasandbox.org unblocked. Unblocked.