Les04a-Subquery on the HAVING

practice and explanation

Here is another subquery example and explanation.

USE PRODUCTS TABLE.

PROBLEM:

Find the category (use ID) in the product table that has an average selling price less than the category of product with the maximum average

This is going to need a subquery as you will need to know the maximum average list price first. Do the select to see ALL the averages first.

**SELECT category\_id, round((avg(list\_price)))**

**FROM products**

**GROUP BY category\_id**

The result is (rounded)

CATEGORY\_ID ROUND((AVG(LIST\_PRICE)))

----------- ------------------------

1 1387

2 1406

5 635

4 402

From here you can figure out what the answer will be, but supposing there were 2000 product lines. AND …

The user does not want to search through all the results manually.

Looking at that result the expected answer to the problem is to find all those below the maximum. The maximum average you can see is category 2, 1406

The next step is to look for the maximum using SQL.

**SELECT max( (avg(list\_price)))**

**FROM products**

**GROUP BY category\_id; gives 1406**

Now we need to pass the value 1406 back to the main query. Since we only want to show group items less than max. You will need a HAVING to limit what is shown.

Finish off the code without looking first.

**SELECT category\_id, round(avg(list\_price))**

**FROM products**

**GROUP BY category\_id**

**HAVING avg(list\_price) < (SELECT max((avg(list\_price)))**

**FROM products**

**GROUP BY category\_id);**

**CATEGORY\_ID ROUND(AVG(LIST\_PRICE))**

**----------- ----------------------**

**1 1387**

**5 635**

**4 402**