Les04b-Subquery in FROM practice and notes

Here are more examples to read through and understand.

Problem 1

Show all product\_no that are less than 40400.

The simple way is

SELECT product\_id

FROM products

WHERE product\_id < 23

You get 22 rows

Using a subquery on a FROM statement. Sometimes called an (INLINE views)

SELECT product\_id

FROM (SELECT product\_id

FROM products

WHERE product\_id < 23)

Result is the same. Of course, there was no advantage to using a subquery other than to show it can also be done on the FROM statement.

PROBLEM 2. A bit more difficult. Show the product number and price as well as the count of how often it was ordered.

Product number and selling price (list\_price) come from the PRODUCTS table. But to count the number of times it was ordered comes from the ORDER\_ITEMS table. Since it is 2 different tables, a join is needed. For this we will use a subquery on the FROM

First, we need data like product\_id and count

SELECT product\_id, count(product\_id)

FROM order\_items

GROUP BY product\_id

The result is 256 rows the last row looks like 🡺 139 1

Secondly, we apply the above SELECT results as a table to join to. So, for example product\_id 60 in the above can join product\_id in PRODUCTS table.

So far, the SQL becomes

SELECT P.product\_id, list\_price, count\_alias.product\_count

FROM products P, (SELECT product\_id, count(product\_id) AS product\_count -- alias column name

FROM order\_items

GROUP BY product\_id) count\_alias - - added alias table name which is used in line 1

Aside: without the join it returns over 5000 rows which exceeds the limit supported by this script and version

Net page ….

Now do the WHERE clause to do the JOIN

SELECT P.product\_id, list\_price, count\_alias.product\_count

FROM products P, (SELECT product\_id, count(product\_id) AS product\_count

FROM order\_items

GROUP BY product\_id) count\_alias

**WHERE p.product\_id = count\_alias.product\_id;**

The result is 256 lines that look like these last 2 lines

111 827.37 1

139 65.92 1

Improving it one more step for management is to ORDER BY

SELECT P.product\_id, list\_price, count\_alias.product\_count

FROM products P, (SELECT product\_id, count(product\_id) AS product\_count

FROM order\_items

GROUP BY product\_id) count\_alias

WHERE p.product\_id = count\_alias.product\_id

**ORDER BY p.product\_id;**

Using a JOIN with the USING method

SELECT product\_id, list\_price, count\_alias.product\_count

FROM products P JOIN (SELECT product\_id, count(product\_id) AS product\_count

FROM order\_items

GROUP BY product\_id) count\_alias

USING (product\_id)

ORDER BY product\_id

As an aside if you modify the previous SQL rather than write new statement you may find lots of errors caused by aliases not needed and bracketing