Name ______________________ (   )

Class ______________________ Date ______________________

Knowledge Management

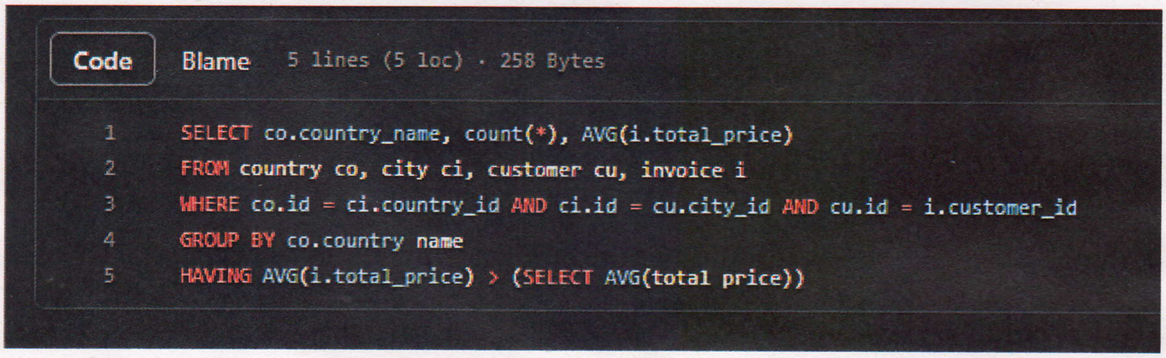
Opened: 26 July 2023

Reg Date: ______________ Reg ID: CSDB ______ Type: Implementation

Topic: Hacker Rank Skill Certification

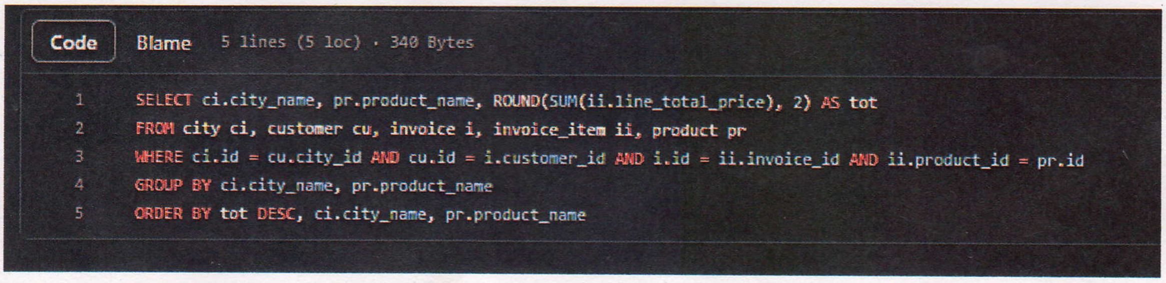
Question: SQL — Intermediate level

Discipline of learning: Database
- SQL language / SQL Command Line
- ROUND / GROUPBY / SUB-QUERIES



5 lines (5 loc) · 258 Bytes

```sql
SELECT co.country_name, count(*), AVG(i.total_price)
FROM country co, city ci, customer cu, invoice i
WHERE co.id = ci.country_id AND ci.id = cu.city_id AND cu.id = i.customer_id
GROUP BY co.country name
HAVING AVG(i.total_price) > (SELECT AVG(total price))
```



5 lines (5 loc) · 340 Bytes

```sql
SELECT ci.city_name, pr.product_name, ROUND(SUM(ii.line_total_price), 2) AS tot
FROM city ci, customer cu, invoice i, invoice_item ii, product pr
WHERE ci.id = cu.city_id AND cu.id = i.customer_id AND i.id = ii.invoice_id AND ii.product_id = pr.id
GROUP BY ci.city_name, pr.product_name
ORDER BY tot DESC, ci.city_name, pr.product_name
```

Hackrank SQL (Intermediate Level) Skill test Certificate

In this paper, the question is divided into 2 parts. Parts 1 and 2 are both applications of SQL queries involving output alias (AS) keyword, table aliasing, logical operation (AND, OR), formatting keyword (GROUP BY), sub-queries; and previously assessed keywords.

Part 1

The query prints out 3 columns of output data, COUNTRY NAME, qty of order, and the average number of invoice total price.

Firstly, selected 4 tables and named them with corresponding table alias;

Then, the WHERE clause started to match different criteria from columns between tables;

After, the GROUP BY function formats outputted entries into COUNTRY_NAME format.

Finally, because of the aggregate function COUNT(*) and AVG, you cannot use the WHERE statement, thus, the HAVING statement kicks in.

The final line needs to calculate the invoice total average price and compare it with the table overall average invoice total price, the former is being outputted.

Part 2

The query is to print out 3 columns of output data, CITY_NAME, PRODUCT_NAME, and alias call TOT which is an aggregate function output that SUMs the line total price then ROUNDED to 2nd decimal point behind.

First and second step is same with part 1 solution, which skip explaining.

Thirdly, GROUPING the selected entries BY CITY_NAME at first, then PRODUCT_NAME as secondary.

Finally, the output entries will pass an ORDER BY filter which orders by descending order of TOT values that, are then followed by 2 resting GROUPed columns.