



GEN BUS 760

Homework 2 — Fall 2021

Note: The Snowflake Trial provides you 400 credits in total. To keep track of the credits you have used so far, run the following commands in a new worksheet:

```
use role accountadmin;
use database snowflake;
use schema snowflake.account_usage;
select sum(credits_used) as total_credits_used from warehouse_metering_history;
```

For each of the questions below, write a single SQL query using the StackOverflow database from Lab 4 (set your context correctly). In your Canvas submission, copy and paste both your query and the result/output separated by blank lines for each of the questions. See [here](#) for the StackOverflow database schema.

1. [1 point] Did questions that were edited to be improved receive more answers on average? Write a single query to compute the average number of answers (`AnswerCount`) received by questions (make sure to filter on `PostTypeId`) on StackOverflow that were (i) edited at least once, and (ii) never edited.

Hint: Use the `NVL2` function

2. [2 points] Among all users whose first answer on StackOverflow was after January 1, 2009, which user ID had the most answers, and how many answers did they have?

Hint: Use the `HAVING` clause

3. [2 points] Which user ID had the largest number of accepted answers, excluding the user with `ID=0` (i.e. excluding the StackOverflow test user)?

Hint: Try a self-join on the `AcceptedAnswerId` column

4. [5 points] Among all users with at least 100 answers, which user ID had the largest percentage of their answers accepted (i.e. their accepted answer count / their total answer count $\times 100$)? Print their accepted answer count, their total answer count, and the percentage of their answers accepted.

Hint: Combine you answer to Q3 with a subquery to get the total answer count

5. [2 points] Which post has the most number of downvotes compared to upvotes (i.e. maximum value of downvotes - upvotes)?

*Hint: Use the **Votes** table, use the **SUM** function with a **CASE WHEN** block to count upvotes and downvotes separately, check the **VoteTypes** table for the **VoteTypeID**s of upvotes (**UpMod**) and downvotes (**DownMod**)*

6. [3 points] Some users answer and accept the answers to their own questions; these are called “self-acceptances”. Which user had the largest number of self-acceptances (excluding the user with ID=0)?

*Hint: Try a self-join on the **AcceptedAnswerId** column, filter on the user IDs of the questioner and the answerer being equal, then group by the questioner user ID and count*

7. [1 point] Print the number of questions and the number of answers in the database, formatted as follows:

```
Number of Questions    ????
Number of Answers     ????
```

*Hint: Try a **UNION***

8. [3 points] Report the popularity (in terms of number of posts) and the popularity rank of the following database management systems: MySQL (filter by tag name like <mysql>), Microsoft SQL Server (filter by tag name like <tsql>), SQLite (filter by tag name like <sqlite>), and PostgreSQL (filter by tag name like <postgresql>). Format your output as below:

TAGNAME	POPULARITY	TAGRANK
<mysql>	31,999	1
<tsql>	6,672	2
<sqlite>	3,766	3
<postgresql>	3,440	4

*Hint: Try a **UNION** inside a subquery to compute popularities, and use **RANK()** over those popularities in descending order*

9. [6 points] Report the number of posts created in each year and month (as month name), along with a total for each year and a grand total over all years. Sort by the year and month number in increasing order. Note: This is a challenging query to get perfectly right.

Hints: Use GROUP BY ROLLUP in a subquery to get the month totals, year totals, and grand totals; include max(date_part(month, CreationDate)) as PostMonthNumber as one of the columns of this subquery; order by PostYear and PostMonthNumber; use the NVL function to replace null values in the with "Total"

Format your result as follows:

POSTYEAR	POSTMONTHNAMEORTOTAL	NUMPOSTS
2,008	Jul	6
2,008	Aug	19,466
2,008	Sep	77,220
2,008	Oct	70,144
2,008	Nov	54,505
2,008	Dec	53,194
2,008	Total	274,535
2,009	Jan	70,137
2,009	Feb	76,442
2,009	Mar	83,444
2,009	Apr	85,710
2,009	May	101,247
2,009	Jun	110,104
2,009	Jul	122,195
2,009	Aug	118,798
2,009	Sep	117,215
2,009	Oct	128,567
2,009	Nov	138,928
2,009	Dec	136,187
2,009	Total	1,288,974
2,010	Jan	155,450
2,010	Feb	149,562
2,010	Mar	169,703
2,010	Apr	158,857
2,010	May	164,011
2,010	Jun	173,593
2,010	Jul	190,157
2,010	Aug	196,758
2,010	Sep	186,826
2,010	Oct	193,791
2,010	Nov	214,750
2,010	Dec	212,227
2,010	Total	2,165,685
	Total	3,729,194