SQL Challenge

Questions

You are presented with sample rows of the following 3 tables:

Table1: purchases

visitor	transaction
Mary	120
Mary	98
aaron	70
henry	153
Amy	10

...

Table2: demos

name	age
Mary	26
Mary	26
alice	29
alice	32
henry	NULL

...

Table3: fraud

visitor
bill
Ella

...

There are some data quality issues, so you'll have to be careful when handling data. The following apply to all of the questions that follow.

1. First, there are some duplicates in the "demos" table, so if someone has multiple age entries, use their max age.
2. There are also people with no listed age (NULL), or who do not appear in the "demos" table, which you will need to keep in mind.
3. Finally, some visitors committed fraud (those listed in the "fraud" table), in which case we don't want to use their data in either of the following questions.

Note that the visitor column is referred to as name in the "demos" table.

You may use a SQL standard of your choice in your solutions. If unsure about any specifics in the data or the questions, make assumptions as needed and note them in your solutions.

Questions:

1. In order to understand the demographic of visitors, output a distribution of age by % of total visitors (i.e. for each age, what % of the total population is that age?). Do not include NULLs/missing ages in your calculation and output.

Also, since many ages might appear infrequently in the data, only return those ages that appear at least 5% in the population.

1. Create a table that shows the mean of transactions per visitor in the "purchases" table, but only for those visitors who begin with the letter A (in uppercase or lowercase), for example aaron and Amy (assuming neither of them committed fraud).

Also, add an indicator (0/1) column that indicates whether the visitor had at least one single transaction of over 100 (For example, Mary would be assigned a 1 for this column, as she has at least one transaction over 100.)

Return this table sorted by the mean of transactions in descending order, showing only the top 10 results.

Answers

NOTE

Assumptions for both challenges:

I) There are no significant outliers. If so, we could first plot a graphic overview and then go into more deeper detail finding outliers using methods as Z-score calculation for each data point.

II) We are 100% sure that there are no different data types per field/column. (This has a lot to do with the data source).

III) Empty rows, if exist, have been deleted.

Pandas is very useful to perform these kind of analysis.

1st Challenge

Assumptions for this challenge:

A) All names will be converted to UPPER case for this analysis. Taking this into consideration, if there are two or more rows with the same 'name' and 'age' data, only one will be kept.

B) One decimal needed for the age percentage distribution calculation.

Query:

In [ ]:
```
WITH
        fraud_names AS (
                SELECT DISTINCT
                        UPPER(visitor) AS fraud_nameUp
                FROM
                        fraud
        ),

        demos_cleansed AS (
                SELECT
                        *
                FROM
                        (
                        SELECT DISTINCT -- see 'Assumptions' (discarding duplicates)
                                UPPER(name) AS nameUp,
                                MAX(age) AS max_age  -- quality issue #1
                        FROM
                                demos
                        GROUP BY
                                nameUp
                        ) AS demos_preCleansed
                WHERE
                        max_age IS NOT NULL  -- quality issue #2 & challenge question
                        AND
                        nameUp NOT IN (SELECT fraud_nameUp FROM fraud_names) -- quality issue #3
        ),

        percentage_distributions AS (
                SELECT
                        max_age,
                        ROUND(
                                        (COUNT(max_age)
                                                * 100
                                        / (SELECT COUNT(*) AS total_visitors
                                           FROM demos_cleansed)),
                                1)
                                AS "percentage distribution" -- see 'Assumptions' [round()] & challenge question
                FROM
                        demos_cleansed
                GROUP BY
                        max_age
        )

SELECT
        max_age AS "visitor age",
        "percentage distribution"
FROM
        percentage_distributions
WHERE
        "percentage distribution" >= 5.0; -- see challenge question
```

2st Challenge

Assumptions for this challenge:

A) All 'names' will be converted to UPPER case.

Taking this into consideration, if there are two or more rows with the same 'name' and 'transaction' data, only one will be kept.

Query:

In [ ]:
```
WITH
        fraud_names AS (
                SELECT DISTINCT
                        UPPER(visitor) AS fraud_name
                FROM
                        fraud
        ),

        purchases_cleansed AS (
                SELECT *
                FROM
                        (
                        SELECT DISTINCT -- see 'Assumptions' (discarding duplicates)
                                UPPER(visitor) AS visitor,
                                transaction
                        FROM
                                purchases) AS purchasesUpper
                WHERE
                        visitor LIKE ('A%') -- see challenge question (visitors who begin with the letter A)
                        AND
                        visitor NOT IN (SELECT fraud_name FROM fraud_names) -- quality issue #3
        ),

        transaction_indicator AS (
                SELECT
                        visitor,
                        MAX("pre transaction indicator") AS "transaction indicator" -- see challenge question
                FROM
                        (
                        SELECT
                                visitor,
                                CASE
                                        WHEN transaction >= 100
                                                THEN 1
                                        ELSE 0
                                END "pre transaction indicator"
                        FROM
                                purchases_cleansed
                        ) AS pre_transaction_indicators
                GROUP BY
                        visitor
        ),

        avg_transaction AS (
                SELECT
                        visitor,
                        ROUND(AVG(transaction)) AS "mean of transactions"  -- see challenge question
                FROM
                        purchases_cleansed
                GROUP BY
                        visitor
        )

SELECT * INTO table_report FROM -- see challenge question (create a table)
        (
        SELECT
                a.visitor,
                a."mean of transactions",
                i."transaction indicator"
        FROM
                avg_transaction AS a
        INNER JOIN
                transaction_indicator AS i
        ON
                a.visitor = i.visitor
        ORDER BY
                "mean of transactions" DESC -- see challenge question
        LIMIT 10 -- see challenge question
        ) AS t;
```

This exercise asks to create a table from the output. If we do so, the data would become a dataset, no more related to the source of origin (database) that would change over time.

So, as a practical way to end this task, we could send the output as a CSV file (table_report) to whoever is interested in this info.

Finally, this query can be configured to run periodically, which would be useful to see the evolution of the information over time.

Anyway, if we imperiously need a table in our database we can use the option in the above code

In [ ]:
```
INSERT INTO final_report
        SELECT [...] FROM [...];
```

or

In [ ]:
```
CREATE TABLE final_report AS
        SELECT [...] FROM [...];
```