Database Report

XIONG Zhipeng Sid: 44231536

June 19, 2024

Question 1

For each of the above query descriptions Q1, Q2, Q3, generate SQL queries by using ChatGPT. Show the generated queries.

- Q1: Find mid of menuitems such that the number of guests who ordered the menuitem only once is maximum. Report such mid and the maximum number of guests.

```
WITH GuestOrderCount AS (
    SELECT o.mid, o.gid, COUNT(*) AS order_count
    FROM Order o
    GROUP BY o.mid, o.gid
),
MenuItemGuestCount AS (
    SELECT mid, COUNT(gid) AS guest_count
    FROM GuestOrderCount
    WHERE order_count = 1
    GROUP BY mid
)
SELECT mid, guest_count
FROM MenuItemGuestCount
ORDER BY guest_count DESC
LIMIT 1;
```

- Q2: Find guests who always order an item that is most expensive within the table.

```
-- Step 1: Identify the most expensive menu item
WITH MostExpensiveItem AS (
    SELECT mid, price
    FROM MenuItem
    WHERE price = (SELECT MAX(price) FROM MenuItem)
),
-- Step 2: Find guests who have only ordered the most expensive item
GuestOrders AS (
    SELECT gid
    FROM Order
    WHERE mid IN (SELECT mid FROM MostExpensiveItem)
    GROUP BY gid
    HAVING COUNT(DISTINCT mid) = (SELECT COUNT(*) FROM MostExpensiveItem)
),

-- Step 3: Ensure these guests have not ordered any other items
GuestValid AS (
    SELECT go.gid
    FROM GuestOrders go
    LEFT JOIN Order o ON go.gid = o.gid
    LEFT JOIN MostExpensiveItem mei ON o.mid = mei.mid
    GROUP BY go.gid
    HAVING COUNT(DISTINCT o.mid) = COUNT(DISTINCT mei.mid)
)

-- Step 4: Select the guests who meet the criteria
SELECT g.gid, g.tableid
FROM Guest g
JOIN GuestValid gv ON g.gid = gv.gid;
```

- Q3: Find the golden guest pair such that their total amount in two visits is maximum than any other guest pairs. Guest pairs are any two guests coming on the same day and occupy one table by the two.

```
WITH GuestVisitAmount AS (
    SELECT
        g1.gid AS gid1,
        g2.gid AS gid2,
        g1.tableid,
        o1.date,
        SUM(o1.quantity * m1.price) AS amount1,
        SUM(o2.quantity * m2.price) AS amount2,
        SUM(o1.quantity * m1.price) + SUM(o2.quantity * m2.price)
            AS total_amount
    FROM
        Guest g1
        JOIN Guest g2 ON g1.tableid = g2.tableid AND g1.gid < g2.gid
        JOIN Order o1 ON g1.gid = o1.gid
        JOIN Order o2 ON g2.gid = o2.gid AND o1.date = o2.date
        JOIN MenuItem m1 ON o1.mid = m1.mid
        JOIN MenuItem m2 ON o2.mid = m2.mid
    GROUP BY
        g1.gid, g2.gid, g1.tableid, o1.date
),

MaxTotalAmount AS (
    SELECT
        gid1, gid2, total_amount
    FROM
        GuestVisitAmount
    ORDER BY
        total_amount DESC
    LIMIT 1
)

SELECT
    mta.gid1, mta.gid2, mta.total_amount
FROM
    MaxTotalAmount mta;
```

Question 2

For the queries generated from Q1, Q2, Q3.

(2-1)

Argue whether each generated query is correctly expressing its query description. Construct a table instance for explaining your answer. If the generated query is wrong, point out its errors, and show a correct SQL query that returns the expected answer for the query description.

- For Q1, the generated query is correct.

  GuestOrderCount table instance: Calculate the number of times each guest has ordered each menu item. It groups the orders by 'mid' and 'gid' and counts the number of orders for each combination.

  MenuItemGuestCount table instance: Filter the results to include only those guests who ordered a menu item exactly once.

  The final part of the query selects the 'mid' with the maximum count of guests who ordered the menu item exactly once.

- For Q2, the generated query is incorrect. The logic of the generated query is to find the guests who have only ordered the most expensive item. However, guests who always order an item that is most expensive within the table don't necessarily have to order the most expensive item only. The correct query should be:

```
WITH GuestMaxPrice AS (
    SELECT
        g.gid,
        g.tableid,
        o.date,
        MAX(m.price) AS max_price
    FROM
        Guest g
        JOIN Order1 o ON g.gid = o.gid
        JOIN MenuItem m ON o.mid = m.mid
    GROUP BY
        g.gid, g.tableid, o.date
),

TableMaxPrice AS (
    SELECT
        g.tableid,
        o.date,
        MAX(m.price) AS max_price
    FROM
        Guest g
        JOIN Order1 o ON g.gid = o.gid
        JOIN MenuItem m ON o.mid = m.mid
    GROUP BY
        g.tableid, o.date
),

ConsistentMaxPriceGuests AS (
    SELECT
        gmp.gid,
        gmp.tableid,
        gmp.date
    FROM
        GuestMaxPrice gmp
        JOIN TableMaxPrice tmp ON
         gmp.tableid = tmp.tableid AND gmp.date = tmp.date
    WHERE
        gmp.max_price = tmp.max_price
),

EligibleGuests AS (
    SELECT
        gid
    FROM
        ConsistentMaxPriceGuests
    GROUP BY
        gid
    HAVING
        COUNT(*) = (SELECT COUNT(*) FROM GuestMaxPrice
            WHERE gid = ConsistentMaxPriceGuests.gid)
)

SELECT
    g.gid,
    g.tableid
FROM
    Guest g
JOIN
    EligibleGuests eg ON g.gid = eg.gid;
```

- For Q3, the generated query is incorrect. The code only select the guest pair who have the maximum total amount in one visit.

```
WITH GuestVisitAmounts AS (
    SELECT
        g.gid,
        g.tableid,
        o.date,
        SUM(o.quantity * m.price) AS total_amount
    FROM
        Guest g
        JOIN Order1 o ON g.gid = o.gid
        JOIN MenuItem m ON o.mid = m.mid
    GROUP BY
        g.gid, g.tableid, o.date
),

GuestPairs AS (
    SELECT
        gva1.gid AS gid1,
        gva2.gid AS gid2,
        gva1.tableid,
        gva1.date,
        gva1.total_amount AS amount1,
        gva2.total_amount AS amount2,
        gva1.total_amount + gva2.total_amount AS combined_amount
    FROM
        GuestVisitAmounts gva1
        JOIN GuestVisitAmounts gva2
            ON gva1.tableid = gva2.tableid
            AND gva1.date = gva2.date
            AND gva1.gid < gva2.gid
),

TotalAmountsByPair AS (
    SELECT
        gid1,
        gid2,
        SUM(combined_amount) AS total_combined_amount
    FROM
        GuestPairs
    GROUP BY
        gid1, gid2
),

MaxTotalAmountPair AS (
    SELECT
        gid1,
        gid2,
        total_combined_amount
    FROM
        TotalAmountsByPair
    ORDER BY
        total_combined_amount DESC
    LIMIT 1
)

SELECT
    mtap.gid1,
    mtap.gid2,
    mtap.total_combined_amount
FROM
    MaxTotalAmountPair mtap;
```

(2-2)

Revise the natural language prompt so that the correct SQL query or its equivalent query, for the query description is generated.

- For Q2, the natural language prompt should be revised as follows:

  For the query ”Find guests who always order an item that is most expensive within the table.”, the targeted guest don’t have to order the most expensive menu item. It should be his/her order’s price is more expesive than other guest in the same table. And There isn't constraints on data, just table.

- For Q3, the natural language prompt should be revised as follows:

  For Query 3:"Find the golden guest pair such that their total amount in two visits is maximum than any other guest pairs. Guest pairs are any two guests coming on the same day and occupy one table by the two.", you need to find the guest pair that their total amount in two visits is maximum rather than comparing with one visit.