<none> 21280074_HuynhThiThuThoang.sql Thursday, May 2, 2024, 10:24 PM

```sql
--21280074 Huynh Thi Thu Thoang - CASE STUDY - DBMS

--1. Co bao nhieu pizza da duoc dat
SELECT count(pizza_id) AS count_pizza
FROM customer_orders co;
/*
count_pizza|
-----------+
         14|
 */

--2. Co bao nhieu don dat hang khac nhau da duoc dat
SELECT count(DISTINCT(order_id)) count_order
FROM customer_orders co;
/*
count_order|
-----------+
         10|
 */

--3. Voi moi runner, bao nhieu don hang da duoc giao thanh cong
SELECT
        runner_id,
        count(*)
FROM  runner_orders
WHERE cancellation != 'Restaurant Cancellation' AND cancellation != 'Customer
Cancellation' OR  cancellation IS NULL
GROUP BY runner_id
ORDER BY runner_id;
/*
runner_id|count|
---------+-----+
        1|    4|
        2|    3|
        3|    1|
 */

--4.Voi moi customer, bao nhieu loai 'Vegetarian' va 'Meatlovers' da duoc dat
SELECT DISTINCT(co.customer_id), pn.pizza_name, count(pn.pizza_name)
FROM customer_orders co
JOIN pizza_names pn ON co.pizza_id = pn.pizza_id
GROUP BY pn.pizza_name, co.customer_id
ORDER BY co.customer_id ;
/*
 customer_id|pizza_name|count|
-----------+----------+-----+
        101|Meatlovers|    2|
        101|Vegetarian|    1|
        102|Meatlovers|    2|
        102|Vegetarian|    1|
        103|Meatlovers|    3|
        103|Vegetarian|    1|
        104|Meatlovers|    3|
```

```
        105|Vegetarian|    1|
 */

--5. So luong pizza toi da duoc gia cua mot don hang la bao nhieu
SELECT order_id,count(order_id) AS count_pizza_order
FROM customer_orders co
GROUP BY order_id
ORDER BY count(order_id) DESC LIMIT 1;
/*
order_id|count_pizza_order|
--------+-----------------+
       4|                3|
 */

--6. Khoi luong don dat hang moi ngay trong tuan la bao nhieu
SELECT DISTINCT
    CASE
        WHEN CAST(order_time AS DATE) < '2020-01-06' then  1
        ELSE 2
    END
    AS week,
    DATE_TRUNC('week', order_time) AS start_of_week,
    CAST(order_time AS DATE) AS order_time,
    count(*) AS order_count
FROM customer_orders co
GROUP BY CAST(order_time AS DATE), order_time
ORDER BY order_time;
SELECT CURRENT_DATE + INTERVAL '1 day';
/*
week|start_of_week          |order_time|order_count|
----+-----------------------+----------+-----------+
   1|2019-12-30 00:00:00.000|2020-01-01|          1|
   1|2019-12-30 00:00:00.000|2020-01-02|          2|
   1|2019-12-30 00:00:00.000|2020-01-04|          3|
   2|2020-01-06 00:00:00.000|2020-01-08|          1|
   2|2020-01-06 00:00:00.000|2020-01-09|          1|
   2|2020-01-06 00:00:00.000|2020-01-10|          1|
   2|2020-01-06 00:00:00.000|2020-01-11|          2|
 */

--7. Co bao nhieu runners dang ky moi tuan? Tuan bat dau 2021-01-01
SELECT
    DATE_TRUNC('week', registration_date) AS start_of_week,
    COUNT(*) AS new_runners
FROM
    runners
GROUP BY
    start_of_week
ORDER BY
    start_of_week;
/*
start_of_week               |new_runners|
----------------------------+-----------+
```

```
2020-12-28 00:00:00.000 +0700|          2|
2021-01-04 00:00:00.000 +0700|          1|
2021-01-11 00:00:00.000 +0700|          1|
 */

--8. Thoi gian trung binh tinh bang phut de moi runner den tru so Pizza
Runner de nhan don hang la bao nhieu
SELECT
    ro.runner_id,
    AVG(EXTRACT(EPOCH FROM (ro.pickup_time::timestamp -
co.order_time::timestamp)) / 60) AS average_time_minutes
FROM
    runner_orders ro
JOIN
    customer_orders co ON ro.order_id = co.order_id
WHERE pickup_time != 'null'
GROUP BY
    ro.runner_id;
/*
runner_id|average_time_minutes|
---------+--------------------+
        3| 10.4666666666666667|
        2| 23.7200000000000000|
        1| 15.6777777777777778|
 */

--9. Voi moi customer, quang duong trung binh can phai di la bao nhieu
SELECT DISTINCT
    customer_id,
    avg(CAST(SUBSTRING(ro.distance FROM '([0-9]+\.?[0-9]*)') AS float))  AS
distance
FROM
    customer_orders co
JOIN
    runner_orders ro ON co.order_id = ro.order_id
WHERE ro.distance != 'null'
GROUP BY co.customer_id
ORDER BY co.customer_id ;
/*
customer_id|distance          |
-----------+------------------+
        101|              20.0|
        102|16.733333333333334|
        103|23.399999999999995|
        104|              10.0|
        105|              25.0|
 */

--10. Su chenh lech giua thoi gian giao hang lau nhat va ngan nhat cho tat ca
cac don hang la bao nhieu
WITH cte AS (
    SELECT CAST(SUBSTRING(ro.duration FROM '([0-9]+\.?[0-9]*)') AS float) AS
duration
```

```
    FROM runner_orders ro
    WHERE duration != 'null')
SELECT
    max(duration),
    min(duration),
    max(duration) - min(duration) AS diff
FROM cte;
/*
max |min |diff|
----+----+----+
40.0|10.0|30.0|
 */

--11.Toc do trung binh cua moi runner trong moi lan giao hang la bao nhieu
WITH cte AS (
    SELECT
        runner_id,
        CAST(SUBSTRING(ro.duration FROM '([0-9]+\.?[0-9]*)') AS float) AS
duration,
        CAST(SUBSTRING(ro.distance FROM '([0-9]+\.?[0-9]*)') AS float)  AS
distance
    FROM runner_orders ro
    WHERE duration != 'null')
SELECT
    runner_id,
    avg(CAST(distance*60/duration AS float)) AS avg_speed_km_h
FROM cte
GROUP BY runner_id;
/*
runner_id|avg_speed_km_h   |
---------+-----------------+
        3|             40.0|
        2|             62.9|
        1|45.53611111111111|
 */

--12. Ty le phan tram gia hang thanh cong cua moi runner la bao nhieu
WITH successful_delivery AS (
    SELECT
        runner_id,
        COUNT(*) AS total,
        COUNT(*) FILTER (WHERE cancellation != 'Restaurant Cancellation' AND
cancellation != 'Customer Cancellation' OR  cancellation IS NULL) AS
successful
    FROM runner_orders
    GROUP BY  runner_id
)
SELECT
    runner_id,
    successful * 100.0 / total AS percent_successful
FROM successful_delivery;
/*
runner_id|percent_successful  |
```

<none> 21280074_HuynhThiThuThoang.sql Thursday, May 2, 2024, 10:24 PM

```
---------+--------------------+
        3| 50.0000000000000000|
        2| 75.0000000000000000|
        1|100.0000000000000000|
 */

--13. Cac thanh phan tieu chua cho moi pizza la gi
WITH cte AS (
    SELECT
        pn.pizza_name,
        UNNEST(string_to_array(pr.toppings, ', '))::integer AS topping
    FROM pizza_names pn
    JOIN pizza_recipes pr ON pn.pizza_id = pr.pizza_id
)
SELECT
    cte.pizza_name,
    string_agg( pt.topping_name,', ') AS toppings
FROM cte
JOIN pizza_toppings pt ON cte.topping = pt.topping_id
GROUP BY cte.pizza_name;
/*
pizza_name|
toppings                                                                |
----------
+-------------------------------------------------------------------------+
Meatlovers|Bacon, BBQ Sauce, Beef, Cheese, Chicken, Mushrooms, Pepperoni,
Salami|
Vegetarian|Cheese, Mushrooms, Onions, Peppers, Tomatoes, Tomato
Sauce           |
 */

--14. Topping nap thuong duoc them vao nhat
CREATE TEMP TABLE temp_extras AS
WITH cte as(
    SELECT
        UNNEST(string_to_array(extras, ', '))::integer AS most_extras
    FROM customer_orders co
    WHERE extras != 'null'
)
SELECT
    most_extras AS topping_id,
    COUNT(*) AS count_extras
FROM  cte
GROUP BY  most_extras
ORDER BY count_extras DESC;

SELECT *
FROM temp_extras
LIMIT 1;
/*
topping_id|count_extras|
----------+------------+
         1|           4|
```

```
 */

--15. Topping nap thuong bi loai bo nhat
CREATE TEMP TABLE temp_exclusion AS
WITH cte as(
    SELECT
        UNNEST(string_to_array(exclusions, ', '))::integer AS most_exclusions
    FROM customer_orders co
    WHERE exclusions != 'null'
)
SELECT
    most_exclusions AS topping_id,
    COUNT(*) AS count_exclusion
FROM  cte
GROUP BY  most_exclusions
ORDER BY count_exclusion DESC;

SELECT *
FROM temp_exclusion
LIMIT 1;
/*
topping_id|count_exclusion|
----------+---------------+
         4|              4|
 */

--16.Tong so luong cua tung thanh phan duoc su dung trong tat ca cac loai
pizza duoc giao la bao nhieu, sap xep theo so luong tu cao den thap
WITH cte as(
    SELECT
            pn.pizza_id ,
            count(co.pizza_id) OVER (PARTITION BY co.pizza_id) AS
count_topping,
            UNNEST(string_to_array(pr.toppings, ', '))::integer AS topping
        FROM pizza_names pn
        JOIN pizza_recipes pr ON pn.pizza_id = pr.pizza_id
        INNER JOIN customer_orders co ON co.pizza_id = pr.pizza_id
)
SELECT DISTINCT topping,
                count(count_topping) +
                CASE
                    WHEN extras.count_extras IS NULL THEN 0
                    ELSE extras.count_extras
                END -
                CASE
                    WHEN exclution.count_exclusion IS NULL THEN 0
                    ELSE exclution.count_exclusion
                END
                AS total
FROM cte
LEFT JOIN temp_extras extras ON  cte.topping = extras.topping_id
LEFT JOIN temp_exclusion exclution ON cte.topping = exclution.topping_id
GROUP BY topping, extras.count_extras, exclution.count_exclusion
```

```
ORDER BY total DESC;
/*
topping|total|
-------+-----+
      1|   14|
      6|   13|
      4|   11|
      5|   11|
      3|   10|
      8|   10|
     10|   10|
      2|    9|
      7|    4|
      9|    4|
     11|    4|
     12|    4|
 */

--17.Neu 1 pizza MeatLovers co gia $12, Vegertarian co gia $10 thi tong so
tien Pizza Runner thu duoc la bao nhieu
WITH prices AS (
   SELECT
        co.pizza_id,
        CASE
            WHEN pn.pizza_name = 'Meatlovers' THEN 12
            WHEN pn.pizza_name = 'Vegetarian' THEN 10
            ELSE 0
        END AS pizza_price,
        COUNT(*) AS num_orders
   FROM customer_orders co
   JOIN pizza_names pn ON co.pizza_id = pn.pizza_id
   JOIN runner_orders ro ON ro.order_id = co.order_id
   WHERE ro.cancellation != 'Restaurant Cancellation' AND ro.cancellation !=
'Customer Cancellation' OR  ro.cancellation IS NULL
   GROUP BY co.pizza_id, pizza_price
),
order_prices AS (
   SELECT
        pizza_id,
        pizza_price * num_orders AS total_price
   FROM prices
),
total_revenue AS (
   SELECT
           SUM(total_price) AS total_revenue
   FROM order_prices
)
SELECT total_revenue
FROM total_revenue
GROUP BY total_revenue;
/*
total_revenue|
-------------+
```

```
          138|
 */

--18. Neu them $1 ch moi extras, thi tong so tien Runner thu duoc la bao nhieu
SELECT
   SUM(
       CASE
           WHEN p.pizza_name = 'Meatlovers' THEN 12 +
COALESCE(LENGTH(NULLIF(co.extras, '')), 0)
           WHEN p.pizza_name = 'Vegetarian' THEN 10 +
COALESCE(LENGTH(NULLIF(co.extras, '')), 0)
           ELSE 0
       END
   ) AS total_earnings
FROM customer_orders co
JOIN runner_orders ro ON ro.order_id = co.order_id
INNER JOIN pizza_names p ON co.pizza_id = p.pizza_id
WHERE ro.cancellation != 'Restaurant Cancellation' AND ro.cancellation !=
'Customer Cancellation' OR  ro.cancellation IS NULL;
/*
total_earnings|
--------------+
          152|
 */

--19.
WITH prices AS (
   SELECT
        co.pizza_id,
        CASE
            WHEN pn.pizza_name = 'Meatlovers' THEN 12
            WHEN pn.pizza_name = 'Vegetarian' THEN 10
            ELSE 0
        END AS pizza_price,
        COUNT(*) AS num_orders
   FROM customer_orders co
   JOIN pizza_names pn ON co.pizza_id = pn.pizza_id
   JOIN runner_orders ro ON ro.order_id = co.order_id
    WHERE ro.cancellation != 'Restaurant Cancellation' AND ro.cancellation !=
'Customer Cancellation' OR  ro.cancellation IS NULL
   GROUP BY co.pizza_id, pizza_price
),
order_prices AS (
   SELECT
        pizza_id,
        pizza_price * num_orders AS total_price
   FROM prices
),
total_revenue AS (
   SELECT
           SUM(total_price) AS total_revenue
   FROM order_prices
),
```

```sql
distance_info AS (
   SELECT order_id, runner_id,
    CASE
    WHEN distance LIKE '%km' THEN CAST(REPLACE(distance, 'km', '') AS NUMERIC)
    ELSE NULL
    END AS distance_numeric
   FROM runner_orders
)
SELECT total_revenue - (0.30 * SUM(d.distance_numeric)) AS net_revenue
FROM distance_info d
CROSS JOIN total_revenue
GROUP BY total_revenue;
/*
net_revenue|
-----------+
    104.460|
```