Danny Ma SQL Challenge #2

Part A : Pizza Metrics

--1. How many pizzas were ordered ?

select count(order\_id) as pizza\_counts from customer\_orders;

--2. How many unique customer orders were made ?

select count(distinct(customer\_id)) as customer\_counts from customer\_orders;

A screenshot of a computer

Description automatically generated

--3. How many successful orders were delivered by each runner?

With creating a temporary table or any kind of updates

with cte1 as

(

select \*,

case

when cancellation="" then null

when cancellation="null" then null

else cancellation

end as updated\_cancellation

from runner\_orders

)

select runner\_id,count(order\_id)

from cte1

where updated\_cancellation is null

group by runner\_id

order by runner\_id;

A screenshot of a computer

Description automatically generated

Another way :

UPDATE runner\_orders

SET cancellation = NULL

WHERE cancellation = "" OR cancellation = "null";

Then

select runner\_id, count(order\_id) from runner\_orders

where cancellation is null

group by runner\_id

order by runner\_id;

--4

select pizza\_name,count(pizza\_name) from customer\_orders

inner join pizza\_names on customer\_orders.pizza\_id=pizza\_names.pizza\_id

inner join runner\_orders on customer\_orders.order\_id=runner\_orders.order\_id

where cancellation is null

group by pizza\_name;

A screenshot of a computer

Description automatically generated

--5

select customer\_id,pizza\_name,count(pizza\_name) from customer\_orders

inner join pizza\_names on customer\_orders.pizza\_id=pizza\_names.pizza\_id

inner join runner\_orders on customer\_orders.order\_id=runner\_orders.order\_id

group by customer\_id,pizza\_name

order by customer\_id;

A screenshot of a computer

Description automatically generated

--6

select runner\_orders.order\_id, count(runner\_orders.order\_id) as orders from customer\_orders

inner join pizza\_names on customer\_orders.pizza\_id=pizza\_names.pizza\_id

inner join runner\_orders on customer\_orders.order\_id=runner\_orders.order\_id

where cancellation is null

group by order\_id

order by orders desc

limit 1;

A screenshot of a computer

Description automatically generated

--7

Update query

UPDATE customer\_orders

SET

exclusions =

CASE

WHEN exclusions = "" THEN NULL

WHEN exclusions = "null" THEN NULL

ELSE exclusions

END,

extras =

CASE

WHEN extras = "" THEN NULL

WHEN extras = "null" THEN NULL

ELSE extras

END;

No change query

select customer\_id,count(customer\_orders.order\_id) as no\_change from customer\_orders

inner join runner\_orders on runner\_orders.order\_id=customer\_orders.order\_id

where cancellation is null and extras is null and exclusions is null

group by customer\_id

order by customer\_id;

A screenshot of a computer

Description automatically generated

Change query

select customer\_id,count(customer\_orders.order\_id) as no\_change from customer\_orders

inner join runner\_orders on runner\_orders.order\_id=customer\_orders.order\_id

where cancellation is null and (extras is not null or exclusions is not null )

group by customer\_id

order by customer\_id;

A screenshot of a computer

Description automatically generated

-8

select \* from customer\_orders

inner join runner\_orders on runner\_orders.order\_id=customer\_orders.order\_id

where cancellation is null & exclusions is not null & extras is not null;

A screenshot of a computer

Description automatically generated

9

select hour(customer\_orders.order\_time) AS 'Hour',

date(customer\_orders.order\_time) as 'date',

count(customer\_orders.order\_id) AS 'Number of pizzas ordered',

round(100\*count(customer\_orders.order\_id) /sum(count(customer\_orders.order\_id)) over(), 2) AS 'Volume of pizzas ordered'

from customer\_orders

inner join runner\_orders on runner\_orders.order\_id=customer\_orders.order\_id

where cancellation is null

group by 2;

A screenshot of a computer

Description automatically generated

10.

select dayname(order\_time) AS 'Day Of Week',hour(customer\_orders.order\_time) AS 'Hour',

date(customer\_orders.order\_time) as 'date',

count(customer\_orders.order\_id) AS 'Number of pizzas ordered',

round(100\*count(customer\_orders.order\_id) /sum(count(customer\_orders.order\_id)) over(), 2) AS 'Volume of pizzas ordered'

from customer\_orders

inner join runner\_orders on runner\_orders.order\_id=customer\_orders.order\_id

where cancellation is null

group by 1;

A screenshot of a computer

Description automatically generated