**Pizza Runner- SQL Challenge**

1. **What are the standard ingredients for each pizza?**

with ingredients as (

SELECT b.pizza\_id,cast(topping\_id as INT64) as topping\_id,a.pizza\_name

FROM `dannys-diner-394605.pizza.pizza\_recipes`b ,

unnest (split(toppings,',')) as topping\_id

inner join `dannys-diner-394605.pizza.pizza\_names` a

on a.pizza\_id=b.pizza\_id)

select c.topping\_name

from ingredients i

join `dannys-diner-394605.pizza.pizza\_toppings`c

on i.topping\_id =c.topping\_id

group by c.topping\_name

having count(distinct pizza\_id)=2

1. **What was the most commonly added extra?**

SELECT b.topping\_name, count(pizza\_id) as added\_extra

FROM `dannys-diner-394605.pizza.customer\_orders` a,

unnest (split(extras,',')) as value

join `dannys-diner-394605.pizza.pizza\_toppings` b

on cast(value as INT64)=b.topping\_id

WHERE value <> 'null'

AND SAFE\_CAST(value AS INT64) > 0

group by b.topping\_name

1. **What was the most common exclusion?**

SELECT b.topping\_name, count(pizza\_id) as exclusion

FROM `dannys-diner-394605.pizza.customer\_orders` a,

unnest (split(exclusions,','))

join `dannys-diner-394605.pizza.pizza\_toppings` b

on safe\_cast(exclusions as int64)=b.topping\_id

where exclusions<>'null'

and safe\_cast(exclusions as int64) is not null

group by b.topping\_name

1. **Generate an order item for each record in the customers\_orders table in the format of one of the following:**
   * **Meat Lovers**
   * **Meat Lovers - Exclude Beef**
   * **Meat Lovers - Extra Bacon**
   * **Meat Lovers - Exclude Cheese, Bacon - Extra Mushroom, Peppers**

with extras as (

SELECT a.order\_id, a.pizza\_id,a.extras,

string\_agg(distinct b.topping\_name, ",") as added\_extra

FROM `dannys-diner-394605.pizza.customer\_orders` a,

unnest (split(extras,',')) as value

join `dannys-diner-394605.pizza.pizza\_toppings` b

on cast(value as INT64)=b.topping\_id

WHERE value <> 'null'

AND SAFE\_CAST(value AS INT64) > 0

group by a.order\_id, a.pizza\_id, a.extras

)

, excluded as (

SELECT a.order\_id, a.pizza\_id,a.exclusions,

string\_agg(distinct b.topping\_name, ",") as excluded

FROM `dannys-diner-394605.pizza.customer\_orders` a,

unnest (split(exclusions,',')) as value

join `dannys-diner-394605.pizza.pizza\_toppings` b

on cast(value as INT64)=b.topping\_id

WHERE value <> 'null'

AND SAFE\_CAST(value AS INT64) > 0

group by a.order\_id, a.pizza\_id, a.exclusions

)

select a.order\_id,concat (case when p.pizza\_name='Meat Lovers' then 'Meat Lovers' else p.pizza\_name end,

coalesce('- Extra ' || added\_extra, ''),

coalesce('- Exclude ' || excluded, '') ) as order\_details

from `dannys-diner-394605.pizza.customer\_orders` a

left join extras as ext on ext.order\_id=a.order\_id and ext.pizza\_id=a.pizza\_id and ext.extras=a.extras

left join excluded as exc on exc.order\_id=a.order\_id and exc.pizza\_id=a.pizza\_id and exc.exclusions=a.exclusions

inner join `dannys-diner-394605.pizza.pizza\_names` p

on p.pizza\_id=a.pizza\_id

1. **Generate an alphabetically ordered comma separated ingredient list for each pizza order from the customer\_orders table and add a 2x in front of any relevant ingredients**
   * **For example: "Meat Lovers: 2xBacon, Beef, ... , Salami"**

with extras as (

SELECT a.order\_id, a.pizza\_id,a.extras,b.topping\_id,

b.topping\_name

FROM `dannys-diner-394605.pizza.customer\_orders` a,

unnest (split(extras,',')) as value

join `dannys-diner-394605.pizza.pizza\_toppings` b

on cast(value as INT64)=b.topping\_id

WHERE value <> 'null'

AND SAFE\_CAST(value AS INT64) > 0

)

, excluded as (

SELECT a.order\_id, a.pizza\_id,a.exclusions,b.topping\_id,

b.topping\_name as excluded

FROM `dannys-diner-394605.pizza.customer\_orders` a,

unnest (split(exclusions,',')) as value

join `dannys-diner-394605.pizza.pizza\_toppings` b

on cast(value as INT64)=b.topping\_id

WHERE value <> 'null'

AND SAFE\_CAST(value AS INT64) > 0

)

, orders as (

select a.order\_id,a.pizza\_id,cast(topping\_id as INT64) as topping\_id, topping\_name

from `dannys-diner-394605.pizza.customer\_orders` a

inner join `dannys-diner-394605.pizza.pizza\_recipes` b

on a.pizza\_id=b.pizza\_id,

unnest (split(toppings,',')) as toppping\_id

inner join `dannys-diner-394605.pizza.pizza\_toppings` c

on c.topping\_id= cast(topping\_id as INT64)

)

, orders\_as\_extras\_and\_exclusions as (

select o.order\_id,o.pizza\_id,o.topping\_id,topping\_name

from orders o

left join excluded as exc on exc.order\_id=o.order\_id and exc.pizza\_id=o.pizza\_id and exc.topping\_id=o.topping\_id

where exc.topping\_id is null

union all

select order\_id,pizza\_id,topping\_id,topping\_name

from extras

)

, ingredients\_total as (

select O.order\_id,pn.pizza\_name, O.topping\_name, count(topping\_id) as n

from orders\_as\_extras\_and\_exclusions O

inner join `dannys-diner-394605.pizza.pizza\_names` pn

on pn.pizza\_id=O.pizza\_id

group by order\_id,topping\_name,pn.pizza\_name

order by O.order\_id,pn.pizza\_name, O.topping\_name

)

, summary as (

select order\_id, pizza\_name,

string\_agg(distinct case when n>1 then n || 'x' || topping\_name

else topping\_name

end, ',') as ingred

from ingredients\_total

group by order\_id, pizza\_name

)

select order\_id, concat((case when pizza\_name= 'Meat Lovers' then 'Meat. Lovers' else pizza\_name end),': ',ingred) as ingredient\_list

from summary

1. **What is the total quantity of each ingredient used in all delivered pizzas sorted by most frequent first?**

with extras as (

SELECT a.order\_id, a.pizza\_id,a.extras,b.topping\_id,

b.topping\_name

FROM `dannys-diner-394605.pizza.customer\_orders` a,

unnest (split(extras,',')) as value

join `dannys-diner-394605.pizza.pizza\_toppings` b

on cast(value as INT64)=b.topping\_id

WHERE value <> 'null'

AND SAFE\_CAST(value AS INT64) > 0

)

, excluded as (

SELECT a.order\_id, a.pizza\_id,a.exclusions,b.topping\_id,

b.topping\_name as excluded

FROM `dannys-diner-394605.pizza.customer\_orders` a,

unnest (split(exclusions,',')) as value

join `dannys-diner-394605.pizza.pizza\_toppings` b

on cast(value as INT64)=b.topping\_id

WHERE value <> 'null'

AND SAFE\_CAST(value AS INT64) > 0

)

, orders as (

select a.order\_id,a.pizza\_id,cast(topping\_id as INT64) as topping\_id, topping\_name

from `dannys-diner-394605.pizza.customer\_orders` a

inner join `dannys-diner-394605.pizza.pizza\_recipes` b

on a.pizza\_id=b.pizza\_id,

unnest (split(toppings,',')) as toppping\_id

inner join `dannys-diner-394605.pizza.pizza\_toppings` c

on c.topping\_id= cast(topping\_id as INT64)

)

, orders\_as\_extras\_and\_exclusions as (

select o.order\_id,o.pizza\_id,o.topping\_id,topping\_name

from orders o

left join excluded as exc on exc.order\_id=o.order\_id and exc.pizza\_id=o.pizza\_id and exc.topping\_id=o.topping\_id

where exc.topping\_id is null

union all

select order\_id,pizza\_id,topping\_id,topping\_name

from extras

)

select topping\_name,count(topping\_id) as n

from orders\_as\_extras\_and\_exclusions o

inner join `dannys-diner-394605.pizza.runner\_orders`r

on o.order\_id=r.order\_id

where cancellation is null

or cancellation='null'

or cancellation='NaN'

or cancellation=''

group by topping\_name

order by count(topping\_id) desc