C. Ingredient Optimisation

1. What are the standard ingredients for each pizza?

WITH CTE AS (

SELECT pizza\_id,

topping\_name

FROM pizza\_recipes as p1

INNER JOIN pizza\_toppings as p2

ON p1.topping\_id = p2.topping\_id

)

SELECT pizza\_id, String\_agg(topping\_name,',') as Standard\_toppings

FROM CTE

GROUP BY pizza\_id;

2. What was the most commonly added extra?

WITH CTE AS (SELECT pizza\_id,

topping\_type,

topping

FROM (SELECT pizza\_id,

CAST(SUBSTRING(extras, 1,1) AS INT) AS topping\_1,

CAST(SUBSTRING(extras,3,3) AS INT) as topping\_2

FROM customer\_orders

WHERE extras is not null) p

UNPIVOT (topping for topping\_type in (topping\_1,topping\_2)) as unpvt)

SELECT Topping,

topping\_name,

COUNT(topping) AS Extra\_Topping\_Time

FROM CTE c

JOIN pizza\_toppings p ON c.topping = p.topping\_id

WHERE topping != 0

GROUP BY topping,topping\_name;

3. What was the most common exclusion?

WITH CTE AS (SELECT pizza\_id,

topping\_type,

topping

FROM (SELECT pizza\_id,

CAST(SUBSTRING(exclusions, 1,1) AS INT) AS exclusions\_1,

CAST(SUBSTRING(exclusions,3,3) AS INT) as exclusions\_2

FROM customer\_orders

WHERE exclusions is not null) p

UNPIVOT (topping for topping\_type in (exclusions\_1,exclusions\_2)) as unpvt)

SELECT topping,

topping\_name,

count(topping) AS exc\_Topping\_Time

FROM CTE c

JOIN pizza\_toppings p ON c.topping = p.topping\_id

WHERE topping != 0

GROUP BY 1,2

ORDER BY 3 DESC;

4. 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\_cte AS (

SELECT

record\_id,

'Extra ' + STRING\_AGG(t.topping\_name, ', ') as record\_options

FROM extras e,

pizza\_toppings t

WHERE e.topping\_id = t.topping\_id

GROUP BY record\_id

),

exclusions\_cte AS

(

SELECT

record\_id,

'Exclude ' + STRING\_AGG(t.topping\_name, ', ') as record\_options

FROM exclusions e,

pizza\_toppings t

WHERE e.topping\_id = t.topping\_id

GROUP BY record\_id

),

union\_cte AS

(

SELECT \* FROM extras\_cte

UNION

SELECT \* FROM exclusions\_cte

)

SELECT c.record\_id,

c.order\_id,

CONCAT\_WS(' - ', p.pizza\_name, STRING\_AGG(cte.record\_options, ' - ')) as pizza\_and\_topping

FROM customer\_orders c

JOIN pizza\_names p ON c.pizza\_id = p.pizza\_id

LEFT JOIN union\_cte cte ON c.record\_id = cte.record\_id

GROUP BY

c.record\_id,

p.pizza\_name,

c.order\_id

ORDER BY 1;

5. 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 INGREDIENT\_CTE AS (SELECT record\_id,

pizza\_name,

CASE WHEN p1.topping\_id in (

SELECT topping\_id

FROM extras e

WHERE C.record\_id = e.record\_id

)

THEN '2x' + p1.topping\_name

ELSE p1.topping\_name

END AS topping

FROM customer\_orders c

JOIN pizza\_names p2 ON c.pizza\_id = p2.pizza\_id

JOIN pizza\_recipes p1 ON c.pizza\_id = p1.pizza\_id

WHERE p1.topping\_id NOT IN (SELECT topping\_id

FROM exclusions e

WHERE e.record\_id = c.record\_id)

)

SELECT record\_id,

CONCAT(pizza\_name +':' ,STRING\_AGG(topping, ',' ) WITHIN GROUP (ORDER BY topping ASC)) AS ingredient\_list

FROM INGREDIENT\_CTE

GROUP BY record\_id,pizza\_name

ORDER BY 1;

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

WITH INGREDIENT\_CTE AS (SELECT record\_id,

pizza\_name,

topping\_name,

CASE WHEN p1.topping\_id in (

SELECT topping\_id

FROM extras e

WHERE C.record\_id = e.record\_id

) THEN 2

ELSE 1

END AS times\_used\_topping

FROM customer\_orders c

JOIN pizza\_names p2 ON c.pizza\_id = p2.pizza\_id

JOIN pizza\_recipes p1 ON c.pizza\_id = p1.pizza\_id

JOIN runner\_orders r ON c.order\_id = r.order\_id

WHERE p1.topping\_id NOT IN (SELECT topping\_id

FROM exclusions e

WHERE e.record\_id = c.record\_id)

and r.cancellation is NULL

)

SELECT topping\_name,

SUM(times\_used\_topping) AS times\_used\_topping

from INGREDIENT\_CTE

GROUP BY topping\_name

order by times\_used\_topping desc;

SELECT topping\_name,

SUM(times\_used\_topping) AS times\_used\_topping

from INGREDIENT\_CTE

GROUP BY topping\_name

order by times\_used\_topping desc;