A poster for a video game

Description automatically generated with low confidence

**---A. Pizza Metrics**

**---How many pizzas were ordered?**

SELECT

COUNT(PIZZA\_ID) AS TOTAL\_PIZZAS\_ORDERED

FROM pizza\_runner.CUSTOMER\_ORDERS;

A picture containing text, font, line, screenshot

Description automatically generated

**2) How many unique customer orders were made?**

SELECT

COUNT(DISTINCT ORDER\_ID) AS TOTAL\_ORDERS

FROM pizza\_runner.CUSTOMER\_ORDERS;

A picture containing text, screenshot, font

Description automatically generated

**3) How many successful orders were delivered by each runner?**

SELECT

RUNNER\_ID,

COUNT(ORDER\_ID) AS SUCCESSFUL\_ORDERS

FROM pizza\_runner.RUNNER\_ORDERS

WHERE PICKUP\_TIME != 'null'

GROUP BY RUNNER\_ID

ORDER BY RUNNER\_ID;

A picture containing text, screenshot, line, font

Description automatically generated

**4) How many of each type of pizza was delivered?**

SELECT

CO.PIZZA\_ID,

PN.PIZZA\_NAME,

COUNT(CO.PIZZA\_ID) AS NO\_OF\_PIZZAS

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

USING(ORDER\_ID)

INNER JOIN pizza\_runner.PIZZA\_NAMES PN

ON CO.PIZZA\_ID = PN.PIZZA\_ID

WHERE PICKUP\_TIME != 'null'

GROUP BY CO.PIZZA\_ID, PN.PIZZA\_NAME

ORDER BY CO.PIZZA\_ID

A picture containing text, line, font, screenshot

Description automatically generated

**5) How many Vegetarian and Meatlovers were ordered by each customer?**

SELECT

CO.CUSTOMER\_ID,

COALESCE(SUM(CASE WHEN CO.PIZZA\_ID = 1 THEN 1 END),0) AS Meat\_Lovers,

COALESCE(SUM(CASE WHEN CO.PIZZA\_ID = 2 THEN 1 END),0) AS Vegetarian

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

ON CO.ORDER\_ID = RO.ORDER\_ID

INNER JOIN pizza\_runner.PIZZA\_NAMES PN

ON CO.PIZZA\_ID = PN.PIZZA\_ID

GROUP BY CO.CUSTOMER\_ID

ORDER BY CO.CUSTOMER\_ID;

A screenshot of a computer

Description automatically generated with low confidence

**6) What was the maximum number of pizzas delivered in a single order?**

SELECT

CO.ORDER\_ID,

COUNT(CO.PIZZA\_ID) AS NO\_OF\_PIZZAS

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

USING(ORDER\_ID)

WHERE PICKUP\_TIME != 'null'

GROUP BY CO.ORDER\_ID

ORDER BY NO\_OF\_PIZZAS DESC

LIMIT 1;

A picture containing text, screenshot, line, font

Description automatically generated

**7) For each customer, how many delivered pizzas had at least 1 change and how many ---had no changes?**

WITH EXTRAS AS(

SELECT

CO.CUSTOMER\_ID,

CASE WHEN (CO.EXCLUSIONS !='null' AND CO.EXCLUSIONS != '') OR (CO.EXTRAS !='null' AND CO.EXTRAS != '') THEN 'Y' else 'N' END AS PIZZAS\_count

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

USING(ORDER\_ID)

WHERE RO.PICKUP\_TIME != 'null'

)

SELECT

CUSTOMER\_ID,

COUNT(CASE WHEN PIZZAS\_COUNT = 'Y' THEN 1 END) AS PIZZAS\_WITH\_CHANGE,

COUNT(CASE WHEN PIZZAS\_COUNT = 'N' THEN 1 END) AS PIZZAS\_WITHOUT\_CHANGE

FROM EXTRAS

GROUP BY CUSTOMER\_ID

ORDER BY CUSTOMER\_ID;

A screenshot of a computer

Description automatically generated with medium confidence

**8) How many pizzas were delivered that had both exclusions and extras?**

SELECT

CO.CUSTOMER\_ID,

COUNT(CO.CUSTOMER\_ID) AS PIZZAS\_WITH\_EXCLUSIONS\_AND\_EXTRAS

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

USING(ORDER\_ID)

WHERE RO.PICKUP\_TIME != 'null' AND CO.EXCLUSIONS !='' and CO.EXCLUSIONS != 'null' and CO.EXTRAS !='' and CO.EXTRAS != 'null'

GROUP BY CO.CUSTOMER\_ID;

A picture containing text, font, line, screenshot

Description automatically generated

**9) What was the total volume of pizzas ordered for each hour of the day?**

SELECT

DATE\_PART('HOUR',ORDER\_TIME) AS HOUR\_OF\_THE\_DAY,

COUNT(ORDER\_ID)

FROM pizza\_runner.CUSTOMER\_ORDERS

GROUP BY DATE\_PART('HOUR',ORDER\_TIME)

ORDER BY HOUR\_OF\_THE\_DAY;

A picture containing text, screenshot, line, parallel

Description automatically generated

**10) What was the total volume of pizzas ordered for each hour of the day?**

SELECT

TO\_CHAR(ORDER\_TIME, 'DAY') AS DAY\_OF\_THE\_WEEK,

COUNT(ORDER\_ID)

FROM pizza\_runner.CUSTOMER\_ORDERS

GROUP BY TO\_CHAR(ORDER\_TIME, 'DAY')

ORDER BY DAY\_OF\_THE\_WEEK;

A picture containing text, screenshot, line, font

Description automatically generated

**B. Runner and Customer Experience**

**1) How many runners signed up for each 1 week period? (i.e. week starts 2021-01-01)**

SELECT

DATE\_PART(‘WEEK’,REGISTRATION\_DATE+3) AS WEEK,

COUNT(RUNNER\_ID)

FROM pizza\_runner.RUNNERS

GROUP BY DATE\_PART(‘WEEK’,REGISTRATION\_DATE+3)

ORDER BY WEEK;

A picture containing line, text, screenshot

Description automatically generated

**2) What was the average time in minutes it took for each runner to arrive at the Pizza Runner HQ to pickup the order?**

SELECT

RO.RUNNER\_ID,

ROUND(AVG(EXTRACT(MINUTE FROM

(CAST(RO.PICKUP\_TIME AS TIMESTAMP) - CAST(CO.ORDER\_TIME AS TIMESTAMP))

))::DECIMAL,0) AS AVERAGE\_TIME

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

ON CO.ORDER\_ID = RO.ORDER\_ID

WHERE RO.PICKUP\_TIME != 'null'

GROUP BY RO.RUNNER\_ID

ORDER BY RO.RUNNER\_ID;

A picture containing text, screenshot, line, font

Description automatically generated

**3) Is there any relationship between the number of pizzas and how long the order takes to prepare?**

SELECT

CO.ORDER\_ID,

COUNT(PIZZA\_ID),

EXTRACT(MINUTE FROM

(CAST(RO.PICKUP\_TIME AS TIMESTAMP) - CAST(CO.ORDER\_TIME AS TIMESTAMP))) AS TIME\_TAKEN\_IN\_MINUTES

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

ON CO.ORDER\_ID = RO.ORDER\_ID

WHERE RO.PICKUP\_TIME != 'null'

GROUP BY CO.ORDER\_TIME, RO.PICKUP\_TIME, CO.ORDER\_ID

ORDER BY ORDER\_ID;

A screenshot of a computer

Description automatically generated with low confidence

**4) What was the average distance travelled for each customer?**

SELECT

CO.CUSTOMER\_ID,

ROUND(AVG(REPLACE(DISTANCE,'km','')::DECIMAL),0) AS DISTANCE

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

ON CO.ORDER\_ID = RO.ORDER\_ID

WHERE RO.PICKUP\_TIME != 'null'

GROUP BY CO.CUSTOMER\_ID

ORDER BY CO.CUSTOMER\_ID;

A picture containing screenshot, text, line, font

Description automatically generated

**5) What was the difference between the longest and shortest delivery times for all orders?**

SELECT max(REPLACE(REPLACE(REPLACE(DURATION,'minutes',''),'mins',''),'minute','')::DECIMAL)- MIN(REPLACE(REPLACE(REPLACE(DURATION,'minutes',''),'mins',''),'minute','')::DECIMAL) AS DIFFERENCE

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

ON CO.ORDER\_ID = RO.ORDER\_ID

WHERE RO.PICKUP\_TIME != 'null';

A picture containing text, screenshot, font

Description automatically generated

**6) What was the average speed for each runner for each delivery and do you notice any trend for these values?**

SELECT

RO.RUNNER\_ID,

CO.ORDER\_ID,

ROUND(AVG(REPLACE(DISTANCE,'km','')::DECIMAL/

REPLACE(REPLACE(REPLACE(DURATION,'minutes',''),'mins',''),'minute','')::DECIMAL)\*60,2)

FROM pizza\_runner.CUSTOMER\_ORDERS CO

INNER JOIN pizza\_runner.RUNNER\_ORDERS RO

ON CO.ORDER\_ID = RO.ORDER\_ID

WHERE RO.PICKUP\_TIME != 'null'

GROUP BY RO.RUNNER\_ID,CO.ORDER\_ID

ORDER BY RO.RUNNER\_ID,CO.ORDER\_ID;

A screenshot of a computer

Description automatically generated with low confidence

The highest speeds are of the order id : 8, 7 which shows that the speed is more when the delivery is after 10 pm.

**7) What is the successful delivery percentage for each runner?**

WITH ORDERS AS(

SELECT

RUNNER\_ID,

COUNT(CASE WHEN CANCELLATION LIKE '%Cancellation%' THEN 1 END) AS CANCEL\_PERCENT,

COUNT(ORDER\_ID) AS TOTAL\_PERCENT

FROM pizza\_runner.RUNNER\_ORDERS

GROUP BY RUNNER\_ID

)

SELECT

RUNNER\_ID, TOTAL\_PERCENT, CANCEL\_PERCENT,

ROUND(((TOTAL\_PERCENT -COALESCE(CANCEL\_PERCENT,0))::DECIMAL/TOTAL\_PERCENT)\*100,2) AS CANCELLATION\_PERCENTAGE

FROM ORDERS

GROUP BY RUNNER\_ID, TOTAL\_PERCENT, CANCEL\_PERCENT

ORDER BY RUNNER\_ID;

A screenshot of a computer

Description automatically generated with low confidence

**C. Ingredient Optimisation**

1. What are the standard ingredients for each pizza?

WITH TOPS AS(

SELECT

PIZZA\_ID,

CAST(UNNEST(STRING\_TO\_ARRAY(TOPPINGS,',')) AS INTEGER) AS TOPPING\_ID

FROM pizza\_runner.PIZZA\_RECIPES

)

SELECT

T.PIZZA\_ID,

ARRAY\_AGG(P.TOPPING\_NAME)

FROM pizza\_runner.PIZZA\_TOPPINGS P

INNER JOIN TOPS T

ON P.TOPPING\_ID = T.TOPPING\_ID

GROUP BY T.PIZZA\_ID

ORDER BY T.PIZZA\_ID;

A picture containing text, screenshot, font, line

Description automatically generated

1. What was the most commonly added extra?

WITH EXTRAS AS(

SELECT

UNNEST(STRING\_TO\_ARRAY(EXTRAS,','))::DECIMAL AS EXT

FROM pizza\_runner.CUSTOMER\_ORDERS

WHERE EXTRAS !='null' AND EXTRAS is not NULL AND EXTRAS != ''

ORDER BY EXT DESC

)

SELECT

TOPPING\_ID,

TOPPING\_NAME

FROM EXTRAS E

INNER JOIN pizza\_runner.PIZZA\_TOPPINGS T

ON E.EXT = T.TOPPING\_ID

A picture containing text, line, screenshot

Description automatically generated

1. What was the most common exclusion?

WITH EXCLUSIONS AS(

SELECT

UNNEST(STRING\_TO\_ARRAY(EXCLUSIONS,','))::DECIMAL AS EXT

FROM pizza\_runner.CUSTOMER\_ORDERS

WHERE EXCLUSIONS !='null' AND EXCLUSIONS is not NULL AND EXCLUSIONS != ''

ORDER BY EXT DESC

)

SELECT

TOPPING\_ID,

TOPPING\_NAME

FROM EXCLUSIONS E

INNER JOIN pizza\_runner.PIZZA\_TOPPINGS T

ON E.EXT = T.TOPPING\_ID

GROUP BY TOPPING\_ID,TOPPING\_NAME

ORDER BY COUNT(TOPPING\_ID) DESC LIMIT 1;

A picture containing text, line, font, screenshot

Description automatically generated

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 ORDERS AS(

SELECT

ROW\_NUMBER() OVER(ORDER BY ORDER\_ID) AS RN,

C.ORDER\_ID,

C.CUSTOMER\_ID,

C.PIZZA\_ID,

N.PIZZA\_NAME,

UNNEST(STRING\_TO\_ARRAY(COALESCE(NULLIF(REPLACE(EXCLUSIONS,'null','0'),''),'0'),','))::DECIMAL AS EXCLUDES,

UNNEST(STRING\_TO\_ARRAY(COALESCE(NULLIF(REPLACE(EXTRAS,'null','0'),''),'0'),','))::DECIMAL AS EXTRA

FROM pizza\_runner.CUSTOMER\_ORDERS C

INNER JOIN pizza\_runner.PIZZA\_NAMES N

ON C.PIZZA\_ID = N.PIZZA\_ID

),

ORDER\_WITH\_EXCLUSIONS AS(

SELECT

RN,

ORDER\_ID,

CUSTOMER\_ID,

PIZZA\_ID,

PIZZA\_NAME,

PT.TOPPING\_ID AS TOP\_EXCLUDES,

PT.TOPPING\_NAME AS TOP\_NAME\_EXCLUDES,

EXTRA

FROM ORDERS O

LEFT JOIN pizza\_runner.PIZZA\_TOPPINGS PT

ON O.EXCLUDES = PT.TOPPING\_ID

),

ALL\_ORDERS AS(

SELECT

RN,

ORDER\_ID,

CUSTOMER\_ID,

PIZZA\_ID,

PIZZA\_NAME,

COALESCE(OE.TOP\_EXCLUDES,'0') AS TOP\_EXCLUDES,

OE.TOP\_NAME\_EXCLUDES,

COALESCE(PT.TOPPING\_ID,'0') AS TOP\_EXTRAS,

PT.TOPPING\_NAME AS TOP\_NAME\_EXTRAS

FROM ORDER\_WITH\_EXCLUSIONS OE

LEFT JOIN pizza\_runner.PIZZA\_TOPPINGS PT

ON OE.EXTRA = PT.TOPPING\_ID

),

FINAL\_LIST AS(

SELECT

RN,

ORDER\_ID,

CUSTOMER\_ID,

PIZZA\_ID,

STRING\_AGG(TOP\_NAME\_EXCLUDES,', ') AS AGG\_EXCLUDES,

STRING\_AGG(TOP\_NAME\_EXTRAS,', ') AS AGG\_EXTRAS

FROM ALL\_ORDERS

GROUP BY RN, ORDER\_ID, CUSTOMER\_ID,PIZZA\_ID

ORDER BY RN, ORDER\_ID, CUSTOMER\_ID

)

SELECT

ORDER\_ID,

CUSTOMER\_ID,

CASE WHEN AGG\_EXCLUDES IS NULL AND AGG\_EXTRAS IS NULL THEN PIZZA\_NAME

WHEN AGG\_EXCLUDES IS NOT NULL AND AGG\_EXTRAS IS NULL THEN CONCAT(PIZZA\_NAME,' - Exclude ',AGG\_EXCLUDES)

WHEN AGG\_EXCLUDES IS NULL AND AGG\_EXTRAS IS NOT NULL THEN CONCAT(PIZZA\_NAME,' - Extra ',AGG\_EXTRAS)

ELSE

CONCAT(PIZZA\_NAME,' - Exclude ',AGG\_EXCLUDES,' - Extra ',AGG\_EXTRAS)

END AS PIZZA\_TYPE

FROM FINAL\_LIST F

INNER JOIN pizza\_runner.PIZZA\_NAMES N

ON F.PIZZA\_ID = N.PIZZA\_ID

ORDER BY ORDER\_ID;

A screenshot of a menu

Description automatically generated with medium confidence