

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

**SELECT**

**date\_part**('week',registration\_date) **AS** registration\_week,

**COUNT**(runner\_id) **AS** runner\_signup

**FROM** runners

**GROUP BY** registration\_week

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

**WITH** time\_table **AS** (**SELECT DISTINCT** runner\_id,

r.order\_id,

order\_time,

pickup\_time,

**EXTRACT**(minute **FROM** (pickup\_time - order\_time)) **as** time

**FROM** temp\_customer\_orders c

**INNER JOIN** temporary\_runners\_orders r

**ON** C.order\_id = R.order\_id

**WHERE** r.cancellation **IS NULL**

**GROUP BY** runner\_id,r.order\_id,order\_time, pickup\_time)

**SELECT** runner\_id, **round**(**AVG**(time)) **AS** average\_time

**FROM** time\_table

**GROUP BY** runner\_id;

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

```
with cte as (  
    select  
        c.order_id, count(c.pizza_id) as orders, EXTRACT(minute FROM (pickup_time -  
order_time)) as time_to_prepare  
    FROM temp_customer_orders c  
    INNER JOIN temporary_runners_orders r  
    ON C.order_id = R.order_id  
    WHERE r.cancellation IS NULL  
    group by c.order_id, pickup_time, order_time  
    order by orders desc  
)  
select order_id, orders, time_to_prepare  
from cte  
group by order_id, orders, time_to_prepare
```

--4. What was the average distance travelled for each customer?

```
select distinct customer_id, avg(distance)  
FROM temp_customer_orders c  
INNER JOIN temporary_runners_orders r  
using(order_id)  
where cancellation is null  
group by customer_id
```

--5. What was the difference between the longest and shortest delivery times for all orders?

```
select max(duration) as longest, min(duration) as shortest , max(duration)  
-min(duration) as difference
```

```
FROM temp_customer_orders c
```

```
INNER JOIN temporary_runners_orders r
```

```
using(order_id)
```

```
where cancellation is null
```

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

```
with cte as (
```

```
select distinct runner_id, order_id, round(distance/duration*60) as speed
```

```
from temporary_runners_orders
```

```
where cancellation is null
```

```
--group by runner_id,order_id
```

```
)
```

```
select runner_id,order_id, avg(speed)
```

```
from cte
```

```
group by runner_id,order_id
```

--7. What is the successful delivery percentage for each runner?

```
with cte as (
```

```
select distinct runner_id,order_id,
```

```

        case
            when cancellation is null then 1
        else 0 end as success
from temporary_runners_orders
)

select runner_id, 100*sum(success)/count(runner_id) as success_percentage
from cte
group by runner_id

```

### --C. Ingredient Optimisation

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

```

with cte1 As(
select pizza_id, UNNEST(STRING_TO_ARRAY(toppings, ',')::INT[]) AS toppings
from pizza_recipes pr
)

select c.pizza_id,pizza_name,toppings, topping_name
from pizza_toppings as p
join cte1 as c
on p.topping_id = c.toppings
join pizza_names as pn
on c.pizza_id = pn.pizza_id
order by pizza_id asc

```

## 2. What was the most commonly added extra?

```
with cte1 As(  
  
SELECT  
  
UNNEST(STRING_TO_ARRAY(extras, ',')::INT[]) AS extras  
  
FROM temp_customer_orders as c  
  
)  
  
cte2 as (  
  
select extras as extras_id, count(extras) as count_extras  
  
from cte1 as c  
  
group by extras  
  
order by count_extras desc  
  
)  
  
select extras_id, topping_name, count_extras  
  
from cte2  
  
join pizza_toppings  
  
on cte2.extras_id = pizza_toppings.topping_id  
  
order by count_extras desc  
  
limit 1;
```

## 3. What was the most common exclusion?

```

with cte1 As(
SELECT
unnest (string_to_array(exclusions,','))::INT[] as exclusions
FROM temp_customer_orders as c
),
cte2 as (
select exclusions as exclusions_id, count(exclusions) as count_exclusions
from cte1 as c
group by exclusions
order by count_exclusions desc
)
select exclusions_id, topping_name, count_exclusions
from cte2
join pizza_toppings
on cte2.exclusions_id = pizza_toppings.topping_id
order by count_exclusions desc
limit 1;

```

4.

```

select row_number () over() as row, order_id,
case
when pizza_id = 1 and exclusions = '4' and (extras
like '1, 5') then 'Meat Lovers - Cheese - Extra Bacon, Chicken'

```

```
when pizza_id = 1 and (exclusions like '2, 6') and  
(extras like '1, 4') then 'Meat Lovers - Exclude BBQ Sauce,Mushroom - Extra  
Bacon, Cheese'
```

```
when pizza_id = 1 and exclusions = '4' then  
'exclude cheese'
```

```
when pizza_id = 1 and extras = '1' then 'extra  
bacon'
```

```
else pizza_names.pizza_name  
end as pizza
```

```
from temp_customer_orders
```

```
join pizza_names
```

```
using(pizza_id)
```

```
--Join toppings_name
```

```
--using(pizza_id)
```

```
where pizza_id <> 2
```

----- 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"

--new exclusions table

**drop table if exists exclclusions\_table;**

**create table exclclusions As**

**SELECT row\_number () over() as row\_id,**

**unnest (string\_to\_array(exclusions,',')::INT[]) as exclusions**

**FROM temp\_customer\_orders as c**

-- new extras table

**drop table if exists extra\_table;**

**create table extras As**

**SELECT row\_number () over() as row\_id,**

**unnest (string\_to\_array(extras,',')::INT[]) as extras**

**FROM temp\_customer\_orders as c**

**with cte as (**

**select row\_number () over() as row\_id,\***

**from temp\_customer\_orders**

**),**

**cte2 as(**

**select row\_id,n.pizza\_name,topping\_name,exclusions,extras,CASE**

**WHEN r.toppings IN (**

**SELECT extras**



```

FROM extras AS x

where c.row_id = x.row_id

)

THEN '2x'

ELSE " END AS extra

from cte as c

join pizza_names as n on c.pizza_id = n.pizza_id

join toppings_name as r on c.pizza_id = r.pizza_id

WHERE r.toppings NOT IN (SELECT
exclusions

FROM exclclusions as e

where e.row_id = c.row_id

)

)

SELECT row_id,exclusions,extras,

CONCAT(pizza_name, ': ', STRING_AGG(CONCAT(extra,topping_name),', ')) AS
ingredient_list

FROM cte2

GROUP BY row_id,pizza_name,exclusions,extras

ORDER BY 1;

/*

SELECT n.pizza_name,

STRING_AGG(t.topping_name, ' ') AS standard_ingredients

FROM pizza_names AS n

```

```

JOIN toppings_name AS r
ON n.pizza_id = r.pizza_id
JOIN pizza_toppings AS t
ON r.toppings = t.topping_id
GROUP BY n.pizza_name
*/

```

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

```

with cte as (

select row_number () over() as row_id,*

from temp_customer_orders

),

cte2 as(

select order_id,topping_name,

case

    when toppings in (

        select extras

        from extras as e

        where e.row_id= c.row_id

    )then 2

    when toppings in (

        select exclusions

        from exclclusions as ex

        where ex.row_id= c.row_id

```

```
        )then 0
        else 1 end as topping_used
from toppings_name as t
join cte as c
using(pizza_id)
group by c.row_id,order_id,topping_name,toppings
)
select topping_name, sum(topping_used)
from cte2
group by topping_name
order by sum(topping_used) desc
```