```sql
-- 1. How many pizza were ordered?
select count(pizza_id) AS total_number_of_pizza_ordered from customer_orders

-- 2. How many unique customer_orders were made?
select count(distinct order_id) AS number_of_distinct_customer_orders from customer_orders

-- 3. How many successful orders were delivered by each runner?
select ro.runner_id, count(ro.order_id) AS delivered_orders from runner_orders AS ro
where ro.pickup_time != 'NULL'
group by ro.runner_id
```

97 %

Results | Messages

	runner_id	delivered_orders
1	1	4
2	2	3
3	3	1

```sql
-- 4. How many of each type of pizza was delivered?
SELECT
  pn.pizza_name,
  COUNT(co.order_id) as total_number_of_delivered_orders
FROM
  customer_orders as co
  INNER JOIN pizza_names as pn on co.pizza_id = pn.pizza_id
  INNER JOIN runner_orders as ro on ro.order_id = co.order_id
WHERE
ro.pickup_time != 'NULL'
GROUP BY
  pn.pizza_name;

-- 5. How many Vegetarian and MeatLovers were ordered by each customers?
select count(co.pizza_id) as total_number_of_pizzas_ordered, co.customer_id, pn.pizza_name from customer_orders as co
inner join pizza_names as pn on co.pizza_id = pn.pizza_id
group by co.customer_id, pn.pizza_name
```

97 %

```sql
-- 6. What was the maximum order of pizzas delivered in a single order?
SELECT
-- TOP 1
ro.order_id,
COUNT(co.pizza_id) as number_of_delivered_pizzas
FROM
customer_orders as co
INNER JOIN runner_orders as ro on ro.order_id = co.order_id
WHERE
ro.pickup_time != 'NULL'
GROUP BY
ro.order_id
ORDER BY
COUNT(co.order_id) DESC
```

97 %

Results | Messages

	order_id	number_of_delivered_pizzas
1	4	3
2	3	2
3	10	2
4	1	1
5	2	1
6	5	1
7	7	1
8	8	1

Query executed successfully.

```sql
-- 7. For each customer, how many delivered pizzas had at least 1 change and how many had no changes?
select co.customer_id,
COUNT(CASE WHEN (co.exclusions IS NOT NULL AND co.exclusions !='NULL' AND LEN(co.exclusions)> 0)
        AND (co.extras IS NOT NULL AND co.extras != 'null' AND LEN(co.extras)> 0) THEN co.pizza_id ELSE NULL END) AS number_of_delivered_pizzas_with_one_change,
COUNT(CASE WHEN (co.exclusions IS NULL AND co.exclusions = 'NULL' AND LEN(co.exclusions)> 0)
        AND (co.extras IS NULL AND co.extras = 'NULL' AND LEN(co.extras)> 0) THEN co.pizza_id ELSE NULL END) AS number_of_delivered_pizzas_with_no_chnage
from customer_orders as co
inner join runner_orders as ro on co.order_id = co.order_id
where ro.pickup_time != 'NULL'
group by co.customer_id
```

97 %

```sql
-- 8. How many pizzas were delivered that had both exclusions and extras
select COUNT(co.pizza_id) as number_of_pizzas_delivered from customer_orders as co
inner join runner_orders as ro
on co.order_id = ro.order_id
where ro.pickup_time != 'NULL'
AND
(co.exclusions IS NOT NULL and co.exclusions != 'NULL' and LEN(co.exclusions)>0)
AND
(co.extras IS NOT NULL and co.extras != 'NULL' and LEN(co.extras)>0)
```

97 %

Results | Messages

	number_of_pizzas_delivered
1	1

```sql
-- 9. What was the total volume of pizzas ordered for each hour of the day?
select COUNT(order_id) AS total_volume_of_orders,
DATEPART(hour, order_time) AS hour
from customer_orders
group by DATEPART(hour, order_time)
```

97 %

Results | Messages

	total_volume_of_orders	hour
1	1	11
2	3	13
3	3	18
4	1	19
5	3	21
6	3	23

Query executed successfully.

```sql
-- 10. What was the volume of orders for each day of the week?
select COUNT(order_id) AS total_volume_orders,
DATEPART(WEEKDAY, order_time) AS week_day,
DATENAME(DW, order_time) AS day_name_of_the_week
from customer_orders
group by DATEPART(WEEKDAY, order_time), DATENAME(DW, order_time)
```

97 %

Results | Messages

	total_volume_orders	week_day	day_name_of_the_week
1	1	6	Friday
2	5	7	Saturday
3	3	5	Thursday
4	5	4	Wednesday

```sql
-- 11. Select * from the runners table plus extract the name of the day and start of week from the date column
select *,
DATENAME(WEEKDAY, registration_date) AS day_name,
DATETRUNC(WEEK, registration_date) AS start_of_week
from runners
```

97 %

Results | Messages

	runner_id	registration_date	day_name	start_of_week
1	1	2021-01-01	Friday	2020-12-27
2	2	2021-01-03	Sunday	2021-01-03
3	3	2021-01-08	Friday	2021-01-03
4	4	2021-01-15	Friday	2021-01-10

```sql
-- 12. How many runners signed up for each start_of_week
select COUNT(runner_id) AS number_of_runners,
DATETRUNC(WEEK, registration_date) AS start_of_week
from runners
group by DATETRUNC(WEEK, registration_date)
```

97 %

Results | Messages

	number_of_runners	start_of_week
1	1	2020-12-27
2	2	2021-01-03
3	1	2021-01-10