Milestone 4

Quering the data

Task 1: How many stores does the business have and in which countires?

```
SELECT country_code, COUNT(country_code) as total_no_stores
FROM dim_store_details
GROUP BY country_code
ORDER BY total_no_stores desc;
```

Task 2: Which locations currently have the most stores?

```
SELECT locality, count(locality) as total_no_stores
FROM dim_store_details
GROUP BY locality
ORDER BY total_no_stores desc
limit 20;
```

Task 3: which months have produced the largest amount of sales?

```
SELECT
  EXTRACT(MONTH FROM order_date) AS month,
  SUM(sales_amount) AS total_sales
FROM
  order_table
GROUP BY
  month
ORDER BY
  total_sales DESC;
```

Task 4: How many sales are coming from online?

```
SELECT
        COUNT(orders_table.product_quantity) as total_sales,
        SUM(orders_table.product_quantity) as product_quantity_count,
        CASE
                WHEN dim_store_details.store_type = 'Web Portal' then 'Web'
                ELSE 'Offline'
        END AS location
FROM orders_table
        LEFT Join dim_store_details on orders_table.store_code = dim_store_details.store_code
GROUP BY location
ORDER BY product_quantity_count;
```

Task 5: What percentage of sales come through each type of store?

```
-SELECT
        dim_store_details.store_type as store_details,
        SUM(orders_table.product_quantity * dim_products.product_price) as number_of_sales,

        SUM(orders_table.product_quantity * dim_products.product_price)        /
        (SELECT SUM(orders_table.product_quantity * dim_products.product_price) FROM
orders_table
                LEFT JOIN dim_products on orders_table.product_code =
dim_products.product_code)*100 as total_percent

FROM orders_table
        LEFT JOIN dim_store_details on orders_table.store_code = dim_store_details.store_code
        LEFT JOIN dim_products on orders_table.product_code = dim_products.product_code
GROUP BY store_details
ORDER BY number_of_sales desc;
```

Task 6: Which month in each year produced the highest cost of sales?

```
WITH RankedSales AS (
   SELECT
      sales_amount,
      EXTRACT(YEAR FROM order_date) AS year,
      EXTRACT(MONTH FROM order_date) AS month,
      ROW_NUMBER() OVER (PARTITION BY EXTRACT(YEAR FROM order_date) ORDER BY
sales_amount DESC) AS rank
   FROM
      orders_table
)
SELECT
   sales_amount AS total_sales,
   year,
   month
FROM
   RankedSales
WHERE
   rank = 1;
```

Task 7: what is our staff headcount

```
SELECT SUM(staff_numbers) as total_staff_numbers, country_code
FROM dim_store_details
GROUP BY country_code
ORDER BY total_staff_numbers desc;
```

Task 8: Which German store type is selling the most?

```
SELECT
        COUNT(orders_table.user_uuid) as total_sales,
        dim_store_details.store_type,
        MAX(dim_store_details.country_code) as country_code
FROM orders_table
        LEFT JOIN dim_store_details on orders_table.store_code = dim_store_details.store_code
        LEFT JOIN dim_products on orders_table.product_code = dim_products.product_code
WHERE dim_store_details.country_code = 'DE'
GROUP BY dim_store_details.store_type;
```

Task 9:How quickly is the company making sales?

```
WITH time_table AS (
  SELECT
    EXTRACT(hour FROM CAST(timestamp AS TIME)) AS hour,
    EXTRACT(minute FROM CAST(timestamp AS TIME)) AS minutes,
    EXTRACT(second FROM CAST(timestamp AS TIME)) AS seconds,
    day AS day,
    month AS month,
    year AS year,
    date_uuid
  FROM dim_date_times
),
timestamp_table AS (
  SELECT
    MAKE_TIMESTAMP(
      CAST(time_table.year AS INT),
      CAST(time_table.month AS INT),
      CAST(time_table.day AS INT),
      CAST(time_table.hour AS INT),
      CAST(time_table.minutes AS INT),
      CAST(time_table.seconds AS FLOAT)
    ) AS order_timestamp,
        time_table.date_uuid AS date_uuid,
        time_table.year AS year
    FROM time_table
),
time_stamp_diffs AS (
    SELECT
        timestamp_table.year,
        LEAD(timestamp_table.order_timestamp) OVER (PARTITION BY timestamp_table.year ORDER BY timestamp_table.order_timestamp) - timestamp_table.order_timestamp AS time_diff
    FROM orders_table
    JOIN timestamp_table ON orders_table.date_uuid = timestamp_table.date_uuid
),
year_time_diffs AS (
    SELECT
        year,
        AVG(time_diff) AS average_time_diff
    FROM time_stamp_diffs
    GROUP BY year
    ORDER BY year
)

SELECT
    year,
    CONCAT(
        '"hours": ',
        EXTRACT(HOUR FROM average_time_diff),
        ', "minutes": ',
        EXTRACT(MINUTE FROM average_time_diff),
        ', "seconds": ',
        CAST(EXTRACT(SECOND FROM average_time_diff) AS INT),
        ', "milliseconds": ',
        CAST(EXTRACT(MILLISECOND FROM average_time_diff) AS INT)
    ) AS actual_time_taken
FROM year_time_diffs;
```