SQL-datetime & CTE
-BigQuery

Extract various date, time components (BigQuery)

```sql
select market_start_datetime,
   extract(date from market_start_datetime) as date,
   extract(time from market_start_datetime) as time,
   extract(year from market_start_datetime) as year_no,
   extract(quarter from market_start_datetime) as q_no,
   extract(month from market_start_datetime) as month_no,
   extract(day from market_start_datetime) as day_no,
   extract(week from market_start_datetime) as week_no,
   extract(DAYOFWEEK from market_start_datetime) as week_day,
   extract(hour from market_start_datetime) as hr,
   extract(minute from market_start_datetime) as minu,
   extract(second from market_start_datetime) as second,
   format_datetime("%B", market_start_datetime) as month_name,
   format_datetime("%A", market_start_datetime) as day_name
from farmer_market.datetime_demo
```

Creating a new date_time_demo table (Big query)

```
CREATE TABLE farmer_market.datetime_demo AS
(
SELECT
    market_date,
    market_start_time,
    market_end_time
    ,parse_datetime('%Y-%m-%d %I:%M %P’,
                    CONCAT(market_date, ' ', market_start_time))
    AS market_start_datetime
    , parse_datetime('%Y-%m-%d %I:%M %P’,
                    CONCAT(market_date, ' ', market_end_time))
    AS market_end_datetime
FROM farmer_market.market_date_info
)
```

Create customer_purchase_date table with purchases and market_date_info tables(big query)

```
create table farmer_market.customer_purchases_date as
(
    SELECT
    c.market_date,
    time(PARSE_DATETIME('%I:%M %P', m.market_start_time)) AS market_start_time,
    time(PARSE_DATETIME('%I:%M %P', m.market_end_time)) AS market_end_time,
    time(c.transaction_time) AS transaction_time, -- it was already in the time format, so no need to parse_datetime

    PARSE_DATETIME('%Y-%m-%d %I:%M %P', CONCAT(c.market_date, " ", m.market_start_time )) AS market_start_datetime,
    PARSE_DATETIME('%Y-%m-%d %I:%M %P', CONCAT(c.market_date, " ", m.market_end_time )) AS market_end_datetime,
    PARSE_DATETIME('%Y-%m-%d %H:%M:%S', CONCAT(c.market_date, " ", c.transaction_time )) AS market_date_transaction_time,
    c.product_id,
    c.vendor_id,
    c.customer_id,
    c.quantity,
    c.cost_to_customer_per_qty
    FROM
        farmer_market.customer_purchases c
    LEFT JOIN
        farmer_market.market_date_info m
    ON
        c.market_date = m.market_date
)
```

Question: Let’s say you want to calculate how many sales occurred within the first 30 minutes after the farmer’s market opened, how would you dynamically determine what cutoff time to use? (Big Query)

```
select
Count(*) from
(
    SELECT
        market_start_time,
        transaction_time,
        TIME_ADD(market_start_time, INTERVAL 30 minute) AS first_30_mins
    FROM
    `farmer_market.customer_purchases_date`
) x
where transaction_time between x.market_start_time and
x.first_30_mins
```

Question: Let’s say we wanted to get a profile of each farmer’s market customer’s habits over time.
* Customer’s first purchase
* Customer’s last purchase
* Total days they made a purchase
* How long they are a customer.
* Days since their last purchase (bigQuery)

```
SELECT
    customer_id
    , min(market_date) as first_purchase
    , max(market_date) as last_purchase
    , count(distinct market_date) as total_days_purchased
    , date_diff(max(market_date), min(market_date), week) as customer_lifetime
    , date_diff(current_date(), max(market_date), day) as days_since_last_purchase

    FROM
farmer_market.customer_purchases
GROUP BY
customer_id
```

Question: Write a query that gives us the days between each visit a customer makes.(big Query)

```
SELECT
    customer_id
    , market_date
    , LAG(market_date) OVER (partition BY customer_id ORDER BY market_date) AS previous_purchase
    ,date_diff(market_date, LAG(market_date) OVER (partition BY customer_id ORDER BY
    market_date), day) as days_since_prev_purchase
FROM (
    SELECT DISTINCT customer_id, market_date
    FROM
        farmer_market.customer_purchases
    ORDER BY
        customer_id, market_date
) x
```

Question: Assume today’s date is May 31, 2019, and the marketing director of the farmer’s market wants to give infrequent customers(made less than 2 day of purchases) in the past 30 days an incentive to return to the market in June**(Big Query)**

```
select customer_id,
count(market_date) as market_visits
from
(
    SELECT
    DISTINCT customer_id, market_date
    FROM
    farmer_market.customer_purchases
    where date_diff("2019-05-31", market_date, day) between 0 and 31
) x
group by customer_id
having market_visits <2
```

CTE - Find the customers first name and lastname who made an overall purchase less than 200$ **(Big Query)**

```
with customer_sales as
      (
           select customer_id,
           sum(cost_to_customer_per_qty*quantity) overall_sales
           from `farmer_market.customer_purchases`
           group by customer_id
      )

      , customer_names as (
           select cs.customer_id,
           c.customer_first_name,
           c.customer_last_name,
           cs.overall_sales
           from customer_sales cs left join farmer_market.customer c
           on cs.customer_id=c.customer_id
      )

      , filtered_customer as
      (
           select * from customer_names where overall_sales<200
      )
select * from filtered_customer
```