Date and time continued

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

```
create table farmers_market.customer_purchases_date as
(
SELECT
     c.market_date,
     m.market_start_time,
     m.market_end_time,
     c.transaction_time,
     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
farmers_market.customer_purchases c
LEFT JOIN
farmers_market.market_date_info m
ON
c.market_date = m.market_date
)
```

Create customer_purchase_date table with purchases and market_date_info tables(My SQL)

```
create table farmers_market.customer_purchases_date as
(
SELECT
        c.market_date,
        m.market_start_time,
        m.market_end_time,
        c.transaction_time,
        STR_TO_DATE(CONCAT(c.market_date, " ", m.market_start_time), '%Y-%m-%d %h:%i %p') AS market_start_datetime,
        STR_TO_DATE(CONCAT(c.market_date, " ", m.market_end_time),'%Y-%m-%d %h:%i %p') AS market_end_datetime,
        STR_TO_DATE(CONCAT(c.market_date, " ", c.transaction_time),'%Y-%m-%d %H:%i:%s') AS market_date_transaction_time,
        c.product_id,
        c.vendor_id,
        c.customer_id,
        c.quantity,
        c.cost_to_customer_per_qty
FROM
                farmers_market.customer_purchases c
LEFT JOIN
                farmers_market.market_date_info m
ON
                c.market_date = m.market_date
)
```

Extract various date, time components (BigQuery)

```
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 farmers_market.datetime_demo
```

Extract various date, time components (MySQL)

```
SELECT market_start_datetime,
        EXTRACT(YEAR FROM market_start_datetime) AS date_year,
        EXTRACT(DAY FROM market_start_datetime) AS start_day,
        EXTRACT(MONTH FROM market_start_datetime) AS month_of_year,
        EXTRACT(HOUR FROM market_start_datetime) AS hour_of_day,
        EXTRACT(MINUTE FROM market_start_datetime) AS minute_of_time,
        weekday(market_start_datetime) as week_day_no,
        dayname(market_start_datetime) as day_name,
        monthname(market_start_datetime) as month_name
FROM farmers_market.datetime_demo;
```

Question: Suppose you wish to know from which year to which year data do we have in our database?

BigQuery

```
select
min(extract(year FROM market_start_datetime)) AS start_YEAR,
max(EXTRACT(YEAR FROM market_start_datetime)) AS end_YEAR
from `farmers_market.datetime_demo`
```

My SQL

```
select
min(extract(year FROM market_start_datetime)) AS start_YEAR,
max(EXTRACT(YEAR FROM market_start_datetime)) AS end_YEAR
from farmers_market.datetime_demo
```

Question: What if you only want to see the hour at which the market started and ended on each date?

Big Query

```
SELECT
market_start_datetime,
market_end_datetime,
EXTRACT(hour FROM market_start_datetime) AS start_hr,
EXTRACT(hour FROM market_end_datetime) AS end_hr
FROM
`farmers_market.datetime_demo`
```

MySQL

```
SELECT
market_start_datetime,
market_end_datetime,
EXTRACT(hour FROM market_start_datetime) AS start_hr,
EXTRACT(hour FROM market_end_datetime) AS end_hr
FROM
farmers_market.datetime_demo
```

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_datetime,
   market_date_transaction_time,
   date_ADD(market_start_datetime, INTERVAL 30 minute) AS first_30_mins
   FROM
   `farmers_market.customer_purchases_date`
) x
where market_date_transaction_time between x.market_start_datetime and
x.first_30_mins
```

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? (MySQL)

```
select
Count(*) from
(
        SELECT
        market_start_datetime,
        market_date_transaction_time,
        date_ADD(market_start_datetime, INTERVAL 30 minute) AS first_30_mins
        FROM
        farmers_market.customer_purchases_date
) x
where market_date_transaction_time between x.market_start_datetime and x.first_30_mins
```

Find the number of days between the first and last market dates.(Big Query)

```
SELECT
x.first_market,
x.last_market,
DATE_DIFF(x.last_market, x.first_market, day) days_first_to_last
FROM (
    SELECT
    min(market_start_datetime) first_market,
    max(market_start_datetime) last_market
    FROM farmers_market.datetime_demo
)x
```

<u>What if we want to find the difference in hour and minute</u>

```
SELECT
    TIMESTAMP_DIFF(max(market_end_datetime), min(market_start_datetime) ,HOUR)
      AS market_duration_hours,
    TIMESTAMP_DIFF(max(market_end_datetime), min(market_start_datetime), MINUTE)
      AS market_duration_mins
FROM farmers_market.datetime_demo
```

Note: We can also use the Date_diff to find hours too

Find the number of days between the first and last market dates.(MySQL)

```
SELECT
x.first_market,
x.last_market,
DATEDIFF(x.last_market, x.first_market) days_first_to_last
FROM (
        SELECT
        min(market_start_datetime) first_market,
        max(market_start_datetime) last_market
        FROM farmers_market.datetime_demo
)x
```

<u>What if we want to find the difference in hour and minute</u>

```
SELECT
hour(TIMEDIFF(max(market_end_datetime), min(market_start_datetime))) AS
market_duration_hours,
minute(TIMEDIFF(max(market_end_datetime), min(market_start_datetime))) AS
market_duration_mins
FROM farmers_market.datetime_demo
```

Question: Let’s say we wanted to get a profile of each farmer’s market customer’s habits over time.(BigQuery)

- **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**

```
SELECT customer_id,
   min(market_date) as first_transaction,
   max(market_date) as last_transaction,
   count(*) as total_transactions,
   count(distinct market_date) as total_market_visits,
   Date_diff(max(market_date), min(market_date), day) as customer_duration,
   Date_diff(current_date(), max(market_date), day) as days_since_lst_purchase
FROM farmers_market.customer_purchases_date
group by customer_id
```

Question: Let’s say we wanted to get a profile of each farmer’s market customer’s habits over time.(MY SQL)

- **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**

```
SELECT customer_id,
       min(market_date) as first_transaction,
       max(market_date) as last_transaction,
       count(*) as total_transactions,
       count(distinct market_date) as total_market_visits,
       datediff(max(market_date), min(market_date)) as customer_duration,
       datediff(curdate(), max(market_date)) as days_since_lst_purchase
FROM farmers_market.customer_purchases_date
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, 1) over (partition by customer_id order by market_date) as last_purchase,
    date_diff(market_date, lag(market_date, 1) over (partition by customer_id order by market_date), day) as days_from_prev_purchase
FROM (SELECT distinct
            customer_id,
            market_date FROM
      farmers_market.customer_purchases_date ) as X
```

Let’s Extend the query: Find the avg. days it take for the customer between 2 purchases or how long it takes on an avg for a customer to comeback to the market.

```
select customer_id, avg(days_from_prev_purchase) from
(
SELECT
     customer_id,
     market_date,
     lag(market_date, 1) over (partition by customer_id order by market_date) as last_purchase,
     date_diff(market_date, lag(market_date, 1) over (partition by customer_id order by market_date), day) as days_from_prev_purchase
      FROM (SELECT distinct
                 customer_id,
                 market_date FROM
               farmers_market.customer_purchases_date ) as X

)y
group by y.customer_id
```

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

```
SELECT
customer_id,
market_date,
lag(market_date, 1) over (partition by customer_id order by market_date) as last_purchase,
datediff(market_date, lag(market_date, 1) over (partition by customer_id order by market_date)) as days_from_prev_purchase
FROM (SELECT distinct
        customer_id,
        market_date FROM
    farmers_market.customer_purchases_date ) as X
```

Let’s Extend the query: Find the avg. days it take for the customer between 2 purchases or how long it takes on an avg for a customer to comeback to the market.

```
select customer_id, avg(days_from_prev_purchase) from
(
SELECT
   customer_id,
   market_date,
   lag(market_date, 1) over (partition by customer_id order by market_date) as last_purchase,
   datediff(market_date, lag(market_date, 1) over (partition by customer_id order by market_date)) as days_from_prev_purchase
FROM (SELECT distinct
                    customer_id,
                    market_date FROM
                    farmers_market.customer_purchases_date ) as X
)Y
group by Y.customer_id
```

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 (BigQuery)

```sql
SELECT x.customer_id,
COUNT(DISTINCT x.market_date) AS total_visits_in_30_days FROM
(
    SELECT
        DISTINCT customer_id,
        market_date,
        DATE_DIFF('2019-05-31', market_date, day) as days_before_curr_date
    FROM farmers_market.customer_purchases
    WHERE DATE_DIFF('2019-05-31', market_date, day) between 0 and 30
) x

GROUP BY x.customer_id
having total_visits_in_30_days <=2
```

Note: Why have we included days between 0 and 30 in the where clause?
The table has data even after May 31st so any days after May 31st will be -1, -2 and so on.
If the condition was only<30 then all those records after May 31st will also be included which is incorrect.

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(MySQL)

```
SELECT x.customer_id,
COUNT(DISTINCT x.market_date) AS total_visits_in_30_days FROM
(
    SELECT
    DISTINCT customer_id,
    market_date,
    DATEDIFF('2019-05-31', market_date) as days_before_curr_date
    FROM farmers_market.customer_purchases
    WHERE DATEDIFF('2019-05-31', market_date) between 0 and 30
    ) x

    GROUP BY x.customer_id
    having total_visits_in_30_days <=2
```

Note: Why have we included days between 0 and 30 in the where clause?
The table has data even after May 31st so any days after May 31st will be -1, -2 and so on.
If the condition was only<30 then all those records after May 31st will also be included which is incorrect.