Mysql Based queries:

```
select * from
(
select
       employee_id,first_name,hire_date,job_id,salary,
       ntile(5) over(order by salary desc) as decile_demo
 from employees
 )t
 where t.decile_demo = 4;

 #
 select * from
(
select
       employee_id,first_name,hire_date,job_id,salary,
       dense_rank() over(order by salary desc) as decile_demo
 from employees
 )t
 where t.decile_demo = 4;

 ##show me the name of employee earning highest salary across table?
 select * from
 (
 select
       employee_id,first_name,hire_date,job_id,salary,
       dense_rank() over(order by salary desc) as decile_demo
 from employees
 ) t where t.decile_demo = 1; #karen

 select
       employee_id,first_name,hire_date,job_id,salary
 from employees
 order by salary desc
 limit 1 offset 3; #john

 #nth_value

  select
       employee_id,first_name,hire_date,job_id,salary,
       nth_value(salary,1) over(order by salary desc) as decile_demo,
       nth_value(first_name,1) over(order by salary desc) as decile_demo
 from employees;

 #first_value last_value
select
       employee_id,first_name,hire_date,job_id,salary,
```

```
       first_value(salary) over(order by salary desc) as decile_demo
from employees;

select
       employee_id,first_name,hire_date,job_id,salary,
       last_value(salary) over(order by salary desc
       RANGE BETWEEN unbounded preceding and unbounded following) as decile_demo
from employees;

select
       employee_id,first_name,hire_date,job_id,salary,
       last_value(salary) over(order by salary desc
       RANGE BETWEEN current row and unbounded following) as decile_demo
from employees;

##date time fns

SELECT market_start_datetime,
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(week from market_start_datetime) as week_no,
EXTRACT(day from market_start_datetime) as day_no,
EXTRACT(hour from market_start_datetime) as hr,
EXTRACT(minute from market_start_datetime) as minute,
EXTRACT(second from market_start_datetime) as second
FROM farmers_market.datetime_demo;

select
       market_start_datetime,
       year(market_start_datetime) as my_yr,
       month(market_start_datetime) as my_mnth,
       hour(market_start_datetime) as my_hour,
       date(market_start_datetime) as my_date,
       time(market_start_datetime) as my_time,
       dayname(market_start_datetime) as my_day_name,
       monthname(market_start_datetime) as my_month_name
       #month(market_start_datetime) as my_month_number
FROM farmers_market.datetime_demo;

select datediff(t.end_year,t.start_year)/365 as no_of_days from
(
select
  min(market_date) as start_year,
  max(market_date) as end_year
from farmers_market.datetime_demo
) t ;
```

```
#DCS
select
       employee_id,first_name,hire_date,job_id,salary,
       ntile(5) over(order by salary desc) as decile_demo
 from employees;

 select count(*) from employees where department_id = 50;

 select
       employee_id,first_name,hire_date,job_id,salary,department_id,
       ntile(5) over(partition by department_id  order by salary desc) as decile_demo
 from employees;


Big Query related Queries:

select * from `farmers_market.datetime_demo`;

#Question: Suppose you wish to know from which year to which year data do we have in
our database?
select
 min(extract(year from market_date)) as start_year,
 max(extract(year from market_date)) as end_year
from farmers_market.datetime_demo;

SELECT market_start_datetime,
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(week from market_start_datetime) as week_no,
EXTRACT(day from market_start_datetime) as day_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 minute,
EXTRACT(second from market_start_datetime) as second,
format_date("%A",market_start_datetime) as day_name,
format_date("%B",market_start_datetime) as month_name,
FROM `farmers_market.datetime_demo`;
```

```sql
#date_add and date_sub
select
market_start_datetime,
date_add(market_start_datetime, INTERVAL 30 Minute) as dr_strange,
date_sub(market_start_datetime, INTERVAL 30 Minute) as dr_strange_bck,
date_add(market_start_datetime, INTERVAL -30 Minute) as dr_strange_check
FROM `farmers_market.datetime_demo`;

#Question: Find the number of days between the first and last market dates.
select date_diff(t.end_year,t.start_year,YEAR) as no_of_days from
(
select
 min(market_date) as start_year,
 max(market_date) as end_year
from farmers_market.datetime_demo
) t ;

#Question: Let's say we wanted to get a profile of each farmer's market customer's
habits over time.
#First purchase date
#Last purchase date
#Count of distinct purchases

SELECT customer_id,
     MIN(market_date) AS first_purchase,
     MAX(market_date) AS last_purchase,
     COUNT(DISTINCT market_date) AS count_of_purchase_dates,
     #DATE_DIFF(MAX(market_date), MIN(market_date), day) AS
days_between_first_last_purchase
FROM farmers_market.customer_purchases
GROUP BY customer_id;


#Q1. If we wanted to determine for how long this person has been a customer of the
farmer's market?
SELECT customer_id,
```

```
     MIN(market_date) AS first_purchase,
     MAX(market_date) AS last_purchase,
     COUNT(DISTINCT market_date) AS count_of_purchase_dates,
     DATE_DIFF(MAX(market_date), MIN(market_date), day) AS
days_between_first_last_purchase
FROM farmers_market.customer_purchases
GROUP BY customer_id;

#Q2. If we wanted to also know how long it's been since the customer last made a
purchase?

SELECT customer_id,
     MIN(market_date) AS first_purchase,
     MAX(market_date) AS last_purchase,
     COUNT(DISTINCT market_date) AS count_of_purchase_dates,
     DATE_DIFF(MAX(market_date), MIN(market_date), day) AS
days_between_first_last_purchase,
     DATE_DIFF(CURRENT_DATE(), MAX(market_date), day) AS days_since_last_purchase
 FROM farmers_market.customer_purchases
 GROUP BY customer_id;

 #Question: Write a query that gives us the days between each purchase a customer
makes.

SELECT customer_id,
market_date,
       LAG(market_date,1) OVER (PARTITION BY customer_id ORDER BY market_date) AS
last_purchase
 FROM farmers_market.customer_purchases
 order by 1;

select * 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 count_bw_prchs
FROM farmers_market.customer_purchases
) t
where t.count_bw_prchs > 60
order by customer_id;
```