Queries Used

```
-- Question: In the customer purchases, we have quantity and cost per qty separate,
-- calculate the total cost as quantity * cost_to_customer_per_qty
select
product_id, vendor_id, market_date, customer_id, quantity * cost_to_customer_per_qty as
total_cost
from farmers_market.customer_purchases limit 10;


-- Round off the total_cost to 2 decimal points
-- inbuilt function called as round -> 2 parameters [value, no of decimal points] -> round(column,
2) -> 2 decimal points
select
product_id, vendor_id, market_date, customer_id, round(quantity * cost_to_customer_per_qty,
2) as total_cost
from farmers_market.customer_purchases limit 10;


-- Get the floor value of total cost
-- inbuilt function called as floor to get the floor value
select
product_id, vendor_id, market_date, customer_id, quantity * cost_to_customer_per_qty as
total_cost , floor(quantity * cost_to_customer_per_qty) as f_total_cost
from farmers_market.customer_purchases limit 10;


-- Question: We want to merge each customer's name into a single column that contains the
first name, then a space, and then the last name. full_name
-- concat () -> it takes two or more strings and joins them together
select customer_id, concat(customer_first_name, ' ', customer_last_name) as full_name
from farmers_market.customer
limit 10;

-- Jane Connor. --> JANE Connor
-- solve for this kind of output
-- two inbuilt functions upper -> for converting all alphabets to upper case character
-- concat for joining two or more strings
select customer_id, concat(upper(customer_first_name), ' ', customer_last_name) as full_name
from `farmers_market.customer`
limit 10;


-- Jane Connor. --> jANE cONNOR
```

```
-- 3 inbuilt functions substr, concat, upper, lower
-- substr -> 2 or 3 inputs -> substr (column_name, start_pos, len)
-- The index starts from 1
select
  concat(
    concat(
      lower(substr(customer_first_name, 1, 1)),
      upper(substr(customer_first_name, 2))
    ),
    ' ',
    concat(
      lower(substr(customer_last_name, 1, 1)),
      upper(substr(customer_last_name, 2))
    )
  ) as full_name
from farmers_market.customer limit 10;


-- filtering using where clause
-- Question: Extract all the product names that are part of product category 1
select product_name from farmers_market.product where product_category_id = 1
order by product_name asc;

-- Question: Print a report of everything customer_id 4 has ever purchased at the farmer's
market,
-- sorted by market date, vendor ID, and product ID

select product_id, vendor_id, market_date, customer_id from
`farmers_market.customer_purchases`
where customer_id = 4
order by market_date asc, vendor_id asc, product_id asc;


-- Question: Get all the product info for products with id between 3 and 8 (not inclusive) and of
product with id 10.
-- what all ids I will select (4, 5, 6, 7, 10)
-- rest all I will not select
select * from farmers_market.product
where product_id = 10 or (product_id > 3 and product_id < 8)
order by product_id asc;

-- Question: Find the details of purchases made by customer 4 at vendor 7
-- we need to filter based on multiple columns
-- customer_id and vendor_id
```

```
select * from `farmers_market.customer_purchases`
where customer_id = 4 and vendor_id = 7 -- This is filtering based on multiple columns
order by product_id;


-- Filtering based on strings
-- Question: Find the customer detail with the first name of “Carlos” or the last name of “Diaz”
select * from `farmers_market.customer`
where lower(customer_first_name) = 'carlos' or lower(customer_last_name) = 'diaz';


-- Question: If you wanted to find out what booths vendor 3 was assigned to on or before (less
than or equal to) April 20, 2019
select * from `farmers_market.vendor_booth_assignments`
where vendor_id = 3 and market_date <= '2019-04-20';


-- Between Keyword (both the edges included)
-- Question: Find the booth assignments for vendor 7 for any market date that occurred
-- between April 3, 2019, and May 16, 2019, including either of the two dates.
select * from `farmers_market.vendor_booth_assignments`
where vendor_id = 7
and market_date between '2019-04-03' and '2019-05-16'; -- if I have a date column, I can use
between keyword for filtering records between 2 dates
-- both 2019-04-03 and 2019-05-16 are included in my filteration

-- IN Keyword
-- Question: Return a list of customers with selected last names - [Diaz, Edwards and Wilson].
-- if my last_name is diaz or last_name is edwards or last_name is wilson -> please return those
customer details
-- when you have multiple or clauses, you can use IN keyword and pass all values in form of list
select * from farmers_market.customer
where
lower(customer_last_name) = 'diaz' or
lower(customer_last_name) = 'edwards' or
lower(customer_last_name) = 'wilson';


select * from farmers_market.customer
where lower(customer_last_name) in ('diaz', 'edwards', 'wilson');


-- Like Clause
-- Question: You want to get data about a customer you knew as “Jerry,” but you aren’t
```

```
-- sure if he was listed in the database as “Jerry” or “Jeremy” or “Jeremiah"

select * from `farmers_market.customer` where
lower(customer_first_name) like 'jer%';

select * from `farmers_market.customer` where
lower(customer_first_name) like '_b%'; -- if the name contains b then given me the ans

-- % [0 or more character in like]
-- _ [exactly one character]

-- jer% -> my string starts with `jer` and can have 0 or more characters after it
-- _b% -> my second character should be `b` and there can 0 or more characters after it


-- IS Null
-- Question: Find all of the products from the product table without sizes.
select * from `farmers_market.product`
where product_size is null; -- `= null` is incorrect, you will always use `is null` instead
-- blank and null as a entity or value is different
-- null means emptiness, no datatype is associated with null
-- blank means empty string, is a string with ''

-- To also select black

select * from `farmers_market.product`
where product_size is null
or trim(product_size) = '';
-- trim is inbuilt function used for removing spaces in the beginning and in the end

-- Filtering using subqueries
-- Question: Analyze purchases made at the farmer’s market on days when it rained.
select * from `farmers_market.customer_purchases` where market_date in (
  select market_date from `farmers_market.market_date_info` where market_rain_flag = 1 --
days when it rained
);

-- why will be discussed in query optimization class

-- Question: List down all the Product details where product_category contains ‘Fresh’.

select * from `farmers_market.product` where product_category_id in (
  select product_category_id from `farmers_market.product_category`
  where lower(product_category_name) like '%fresh%'
```

);