Assignment - SQL

1. Find the errors in the below Query :

```
SELECT
vendor_id,
market_date,
product_id,
Original_price,
* RANK (OVER) (PARTITION  vendor_id ORDER BY original_price DESC)
AS price_rank
FROM farmers_market.vendor_inventory
```

* RANK() OVER (PARTITION By vendor_id ORDER BY original_price DESC)

2. Find the error in the below Query :

```
SELECT
vendor_id,
market_date
product_Id,
original_price,
i. DENSERANK() OVER (PARTITION BY product_id ORDER BY original_price
DESC) AS price_rank
ii. FROM vendor_inventory ii. FROM `farmers_market.vendor_inventory`
```

i. DENSE_RANK() OVER (PARTITION BY product_id ORDER BY original_price DESC) AS price_rank

3. Complete the below query to get the desired result (Total_amt paid by customer_id 4 & 5 per market_date)

```
SELECT
market_date,
customer_id,
sum (quantity * cost_to_customer_per_qty) AS total_amt_spent

FROM `farmers_market.customer_purchases`
WHERE customer_id = 4 or customer_id=5 ( or in(4,5) )
group by market_date,customer_id
ORDER BY market_date
```

4. Complete the below query to get the desired result (List down all the product details where product category contains “Fruits” in it.)

```
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 "%fruits%" )
```

5. Write a Query to Rank all the products bases on their Quantities sold per market date (Use all Rank(), Dense_Rank() & Row numbers() functions in your query)

```
select market_date,product_id,quantity,
row_number() over (partition by market_date order by quantity desc) as row_ranking,
rank() over (partition by market_date order by quantity desc) as ranking,
dense_rank() over (partition by market_date order by quantity desc) as dense_ranking
from `farmers_market.customer_purchases` order by market_date
```