--VANQOUVER SQL ANALYSIS--

```sql
--1. Show the entire table
select* from orders;

--2 Show the product column
select "ProductID" from orders;

--3. What is the total quantity of products ordered?
select sum("Quantity") from orders;

--4. What is the average price of products in each category?
select "ProductCategory", round(avg("Price"),2)  from products
group by "ProductCategory";

--5. What product is the most ordered in each city?
select "PropertyID", "ProductID", sum("Quantity")
from orders
Group by "PropertyID", "ProductID"
Order by "PropertyID" ASC, sum ("Quantity") DESC;

--6. What is the total quantity for each product ordered in New York?--
--Here, you need to get the PropertyID of New York.--
--The difference here is the use of where statement which is the filter statement--
--where "PropertyID" = 1 = New York--

select "PropertyID", "ProductID",sum ("Quantity") from orders
where "PropertyID" = 1
group by "PropertyID","ProductID"
order by sum ("Quantity") desc;
--7. What are our top 5 products with a quantity of more than 100—
```

```
select "ProductID", sum("Quantity")
from orders
group by "ProductID"
having sum("Quantity") > 100
order by sum("Quantity") desc
limit 5;

--8. What transactions occurred between 2015-01-16 to 2015-01-20--
select *
from orders
where "OrderDate"
between '2015-01-16' and '2015-01-20';

--9. Identify the product with the highest average price--
--Answer if working with ProductID--

select "ProductID", round(avg("Price"),2)
from products
group by "ProductID"
order by round(avg("Price"),2)desc
limit 1;

--9. Identify the product with the highest average price--
--Answer if working with ProductName--
select "ProductName", round(avg("Price"),2)  from products
group by "ProductName"
order by round(avg("Price"),2)desc
limit 1;
--9. Identify the product with the highest average price--
--Answer if working with ProductCategory--
```

```
select "ProductCategory", round(avg("Price"),2)
from products
group by "ProductCategory"
order by round(avg("Price"),2)desc
limit 1;

--10. Find out the total revenue made from the sales of product in each state--
--You need the information about Price, Quantity and PropertyID ie. for each state name--
--ProductID is common to both the orders table and products table. This is inner join--
--("Quantity" * "Price") as "Revenue"--
--The state name is in propertyinfo table as PropertyState then we need to connect to--
--the propertyinfo table using inner join statement--

select "PropertyID", "Quantity", "Price", ("Quantity" *"Price") as "Revenue",
orders."ProductID", products."ProductID", "PropertyState","ProductName"
from orders
inner join products
on orders."ProductID" = products."ProductID"
inner join propertyinfo
on orders."PropertyID" = propertyinfo."PropID";

--10. Find out the total revenue made from the sales of product in each state--
select "PropertyID", "Quantity", "Price",("Quantity" * "Price") as "Revenue",
"PropertyState","ProductName" from orders
inner join products
on orders."ProductID" = products."ProductID"
inner join propertyinfo
on orders."PropertyID" = propertyinfo."PropID";
```

```sql
--11. In what days were orders made?--
--Use select distinct to remove duplicate from "OrderDate"--
--You can select to see other columns in addition to "OrderDate" if desired--
--You can use order by in DESC if desired---

select distinct "OrderDate"
from orders
order by "OrderDate" desc;

--12. Last year a raffle draw was held in a city with the letter “e”,--
--which cities with the letter “e” do we get orders?--

select "PropertyCity"
from propertyinfo
where "PropertyCity" like '%e%';

--13. Classify the Revenue ( < 100  – Small Income, 100 – 199 Mid Income, --
--200 – 299 High Income, 300 >Mighty Income--

select ("Price" * "Quantity") as "Revenue",
case
when ("Price" * "Quantity") <100 then 'Small Income'
when ("Price" * "Quantity") <200 then 'Mid Income'
when ("Price" * "Quantity") <300 then 'High Income'
else 'Mighty Income'
end as "Financial Statement"
from orders
inner join products
on orders."ProductID" = products."ProductID";
```