JENSON USA

SQL queries to derive various insights

Presented by : @elizabeth.pillai

WELCOME TO OUR COMMUNITY

Jenson USA is one of the original online bike shops and has been selling complete bikes, bicycle parts and accessories on the internet since 1996.

@elizabeth.pillai

_@elizabeth.pillai

```sql
#1# Find the total number of products sold by each store along with the store name.

SELECT
    stores.store_name,
    SUM(order_items.quantity) AS total_quantity
FROM
    orders
        JOIN
    order_items ON order_items.order_id = orders.order_id
        JOIN
    stores ON stores.store_id = orders.store_id
GROUP BY stores.store_name;
```

```sql
#2#Calculate the cumulative sum of quantities sold for each product over time.
select product_id,order_date, quantity, sum(quantity) over (partition by product_id order by order_date) from
(SELECT
    order_items.product_id,orders.order_date, SUM(order_items.quantity) quantity
FROM
    orders
        JOIN
    order_items ON orders.order_id = order_items.order_id
GROUP BY order_items.product_id, orders.order_date) a ;
```

_@elizabeth.pillai

```sql
#3#Find the product with the highest total sales (quantity * price) for each category.

SELECT
    categories.category_id,
    categories.category_name,
    products.product_id,
    products.product_name,
    SUM(order_items.quantity * (order_items.list_price - order_items.discount)) sales
FROM
    order_items
        JOIN
    products ON products.product_id = order_items.product_id
        JOIN
    categories ON categories.category_id = products.category_id
GROUP BY categories.category_id , categories.category_name , products.product_id , products.product_name;
```

_@elizabeth.pillai

```sql
#4#Find the customer who spent the most money on orders.
SELECT
    customers.customer_id,
    CONCAT(customers.first_name, " ",
            customers.last_name) full_name,
    SUM(order_items.quantity * (order_items.list_price - order_items.discount)) sales
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
        JOIN
    order_items ON order_items.order_id = orders.order_id
GROUP BY customers.customer_id , CONCAT(customers.first_name, " ",
        customers.last_name);
```

@elizabeth.pillai

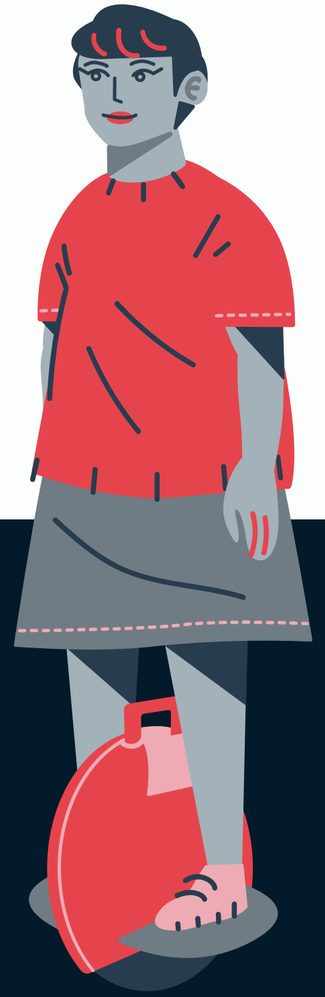
#5#Find the highest-priced product for each category name.

```sql
select * from
(select categories.category_id,categories.category_name, products.product_name, products.list_price,
 rank() over(partition by categories.category_id order by products.list_price desc) Rnk
 from products join categories
 on products.category_id= categories.category_id) a
 where Rnk = 1;
```

#6#Find the total number of orders placed by each customer per store.

```sql
SELECT
    store_id, customer_id, COUNT(order_id)
FROM
    orders
GROUP BY store_id , customer_id;
```

```sql
#7#Find the names of staff members who have not made any sales.
SELECT
    staffs.staff_id,
    CONCAT(staffs.first_name, ' ', last_name) full_name
FROM
    staffs
WHERE
    NOT EXISTS( SELECT
            staff_id
        FROM
            orders
        WHERE
            orders.staff_id = staffs.staff_id);
```



_@elizabeth.pillai

```sql
#8#Find the top 3 most sold products in terms of quantity.

select product_name from
    (SELECT
        products.product_id,
            products.product_name,
            SUM(order_items.quantity) quantity,
            rank() over(order by sum(order_items.quantity) desc) rnk
    FROM
        products
    JOIN order_items ON products.product_id = order_items.product_id
    GROUP BY products.product_id , products.product_name) a
    where rnk <= 3;
```

```sql
#9#Find the median value of the price list.

with m as (select list_price,
 row_number() over(order by list_price) rn,
 count(list_price) over() cn
from order_items)

select case
when cn % 2 =0 then (select avg (list_price) from m
where rn in (cn/2, (cn/2) + 1))
else (select list_price from m where rn = (cn+1)/2)
end as median from m limit 1;
```

_@elizabeth.pillai

```sql
#10#List all products that have never been ordered.(use Exists)
SELECT
    products.product_id, products.product_name
FROM
    products
WHERE
    NOT EXISTS( SELECT
            product_id
        FROM
            order_items
        WHERE
            order_items.product_id = products.product_id);
```

_@elizabeth.pillai

_@elizabeth.pillai

#11List the names of staff members who have made more sales than the average number of sales by all staff members.

```
select avg(sales) from
(SELECT
    staffs.staff_id,
    coalesce(sum(order_items.quantity * (order_items.list_price - order_items.discount)), 0) sales
FROM
    orders
        RIGHT JOIN
    staffs ON staffs.staff_id = orders.staff_id
        left JOIN
    order_items ON orders.order_id = order_items.order_id
GROUP BY staffs.staff_id) as a;
```

@elizabeth.pillai

```sql
#12# Identify the customers who have ordered all types of products (i.e. from every category)

SELECT
    customers.customer_id
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
        JOIN
    order_items ON order_items.order_id = orders.order_id
        JOIN
    products p ON p.product_id = order_items.product_id
    group by customers.customer_id
    having count(distinct p.category_id) = (select count(category_id) from categories);
```




THANK YOU

+0744-673-2035

elizabeth.pillai147@gmail.com

www.elizabeth.pillai

United Kingdom