JENSON SQL Project

Presented by : Ajay Malik

Problems Uncovered in this project.

01 Find the total number of products sold by each store along with the store name.

02 Calculate the cumulative sum of quantities sold for each product over time.

03 Find the product with the highest total sales (quantity * price) for each category.

04 Find the customer who spent the most money on orders.

05 Find the highest-priced product for each category name

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

07 Find the names of staff members who have not made any sales.

08 Find the top 3 most sold products in terms of quantity.

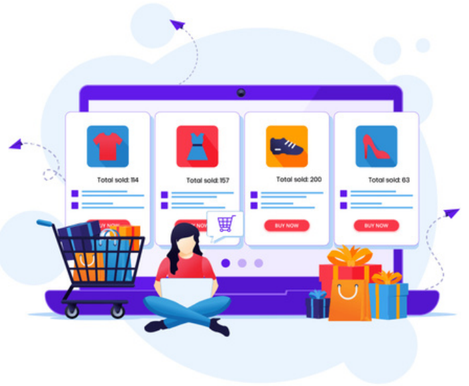
09 Find the median value of the price list.

10 List all products that have never been ordered.(use Exists)

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

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

Find the total number of products sold by each store along with the store name.

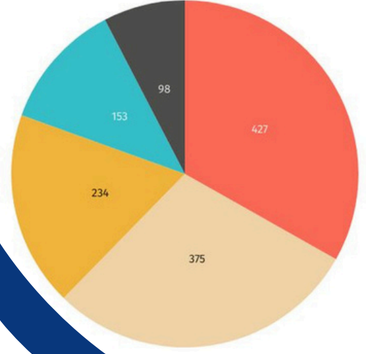



```sql
SELECT
    stores.store_name,
    SUM(order_items.quantity) AS total_quantity
FROM
    orders
        JOIN
    order_items ON order_items.order_id = orders.order_id
        JOIN
    stores ON stores.store_id = orders.store_id
GROUP BY stores.store_name;
```

Calculate the cumulative sum of quantities sold for each product over time.

```sql
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) as quantity
from orders join order_items
on orders.order_id = order_items.order_id
group by order_items.product_id,orders.order_date) a;
```

Calculate the cumulative sum of quantities sold for each product over time.




```sql
with a as
(select categories.category_id,categories.category_name,
products.product_id, products.product_name,
sum(order_items.quantity * (order_items.list_price - order_items.discount)) as 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 products.product_id,products.product_name,
categories.category_id,categories.category_name)

select * from
(select *,rank() over(partition by category_id order by sales desc) as rnk from a) as b
where rnk = 1;
```

Find the customer who spent the most money on orders.

```sql
with a as
(select customers.customer_id,
concat(customers.first_name," ", customers.last_name) as full_name,
sum(order_items.quantity * (order_items.list_price - order_items.discount)) as 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))

select * from
(select *,rank() over(order by sales desc) as rnk
from a) as b
where rnk = 1;
```

Find the highest-priced product for each category name.

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

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

```
SELECT
    orders.store_id,
    orders.customer_id,
    CONCAT(customers.first_name,
            ' ',
            customers.last_name) AS full_name,
    COUNT(orders.order_id)
FROM
    orders
        JOIN
    customers ON orders.customer_id = customers.customer_id
GROUP BY orders.store_id , orders.customer_id;
```

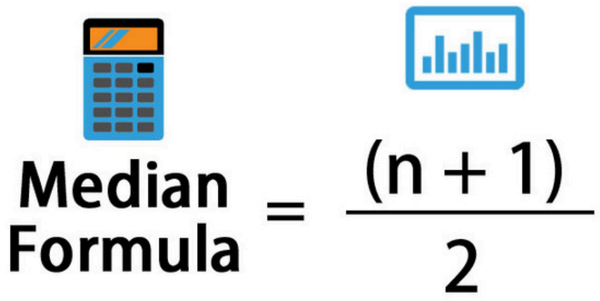
Find the names of staff members who have not made any sales.

```sql
SELECT
    staffs.staff_id,
    CONCAT(staffs.first_name, ' ', staffs.last_name) AS full_name
FROM
    staffs
WHERE
    NOT EXISTS( SELECT
            staff_id
        FROM
            orders
        WHERE
            orders.staff_id = staffs.staff_id);
```

Find the top 3 most sold products in terms of quantity.

```sql
select product_name from
(select products.product_id, products.product_name,
sum(order_items.quantity) as quantity,
rank() over(order by sum(order_items.quantity) desc) as rnk
from products join order_items
on products.product_id = order_items.product_id
group by products.product_id, products.product_name) as a
where rnk <= 3;
```

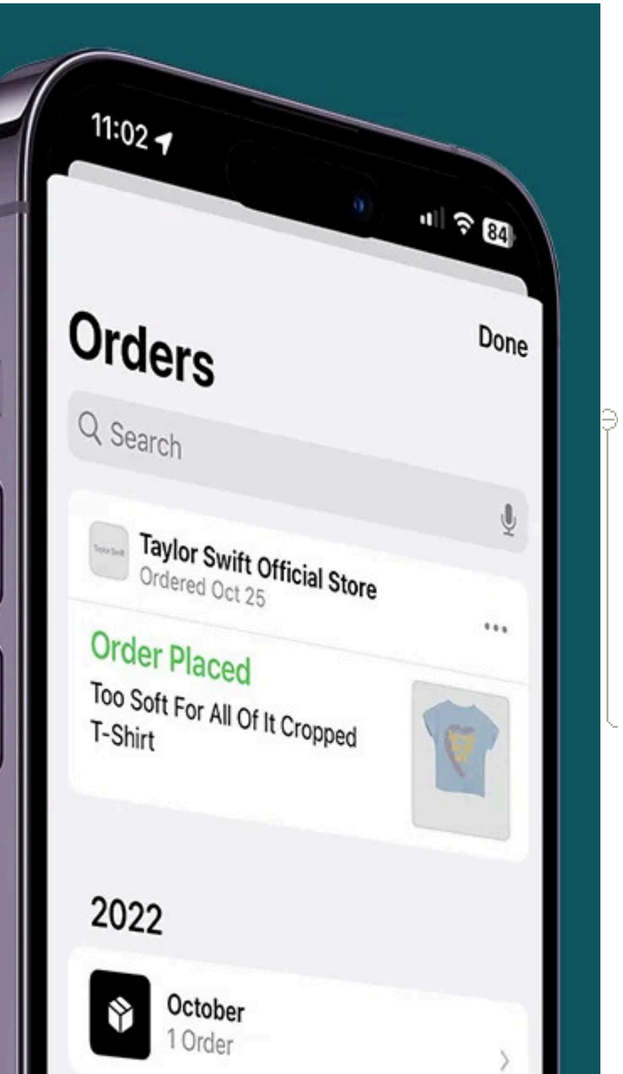
Find the median value of the price list.




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

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

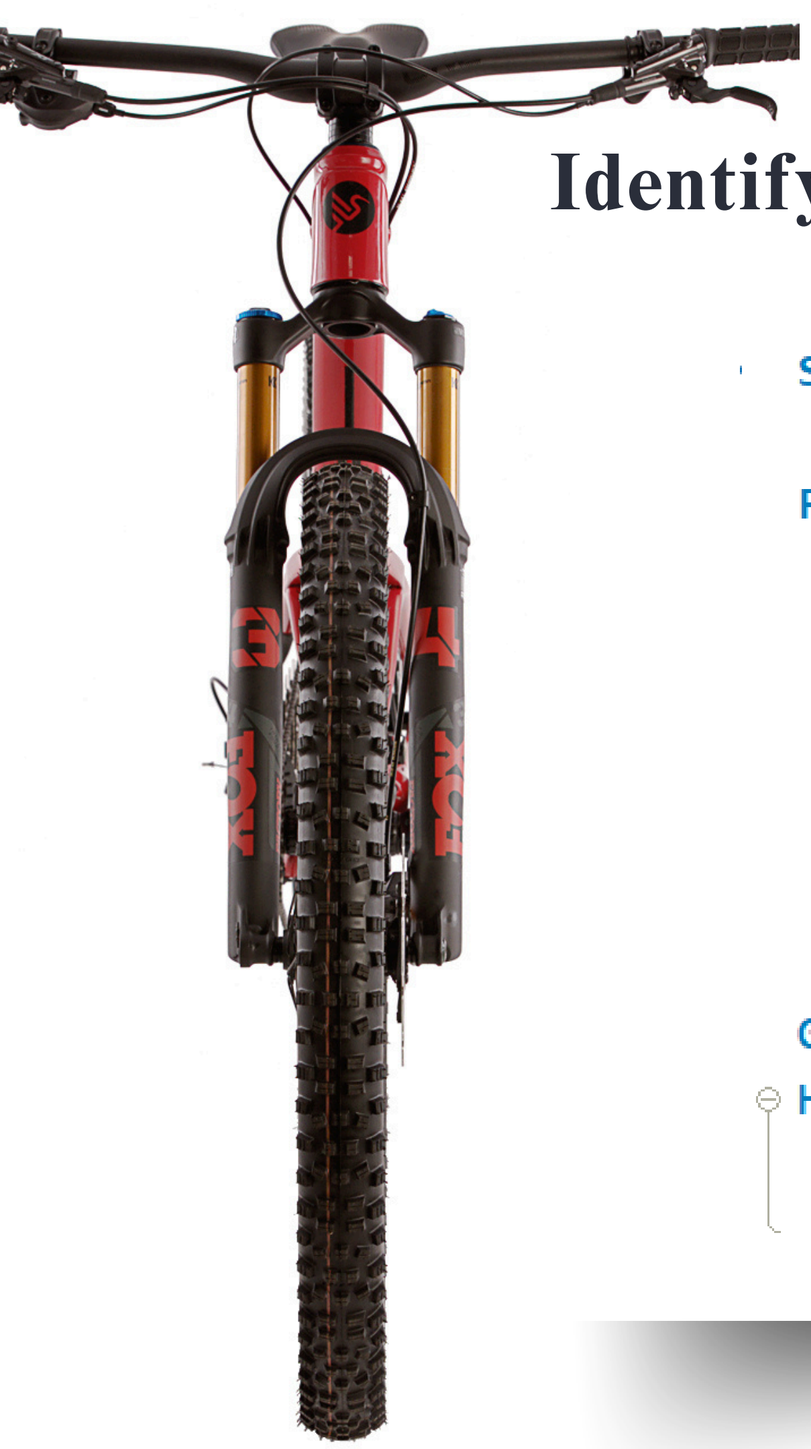
List all products that have never been ordered. (use Exists)




```sql
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);
```

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

```
SELECT
    staffs.staff_id,
    COALESCE(SUM(order_items.quantity * (order_items.list_price - order_items.discount)),
            0) AS 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
HAVING SUM(order_items.quantity * (order_items.list_price - order_items.discount)) > (SELECT
        AVG(sales)
    FROM
        (SELECT
            staffs.staff_id,
                COALESCE(SUM(order_items.quantity * (order_items.list_price - order_items.discount)), 0) AS 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);
```



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 AS 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!

Contact

Mail
ajaymalik021@gmail.com

LinkedIn
Ajay Malik