LARANA, INC.

PROJECT PRESENTATION

ABOUT JENSON USA

Jenson USA is a popular retailer specializing in bicycles, bike components, and cycling accessories. They are known for offering a wide range of products for cycling enthusiasts, including mountain bikes, road bikes, electric bikes, apparel, and gear. Jenson USA often provides expert advice, customer reviews, and competitive pricing, making them a trusted name in the cycling community.

PROBLEM STATEMENT

- Find the total number of products sold by each store along with the store name.
- Calculate the cumulative sum of quantities sold for each product over time.
- Find the product with the highest total sales (quantity * price) for each category.
- Find the customer who spent the most money on orders.
- Find the highest-priced product for each category name.
- Find the total number of orders placed by each customer per store.
- Find the names of staff members who have not made any sales.
- Find the top 3 most sold products in terms of quantity.
- Find the median value of the price list.
- List all products that have never been ordered.(use Exists)
- List the names of staff members who have made more sales than the average number of sales by all staff members.

SOLUTIONS

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

QUERY :

```
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.**

QUERY :

```
select product_id,order_date, quantity,sum(quantity)
 over (partition by product_id order by order_date) as cumulative
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;
```

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

QUERY :

```
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
    products
        JOIN
    order_items ON products.product_id = order_items.product_id
        JOIN
    categories ON products.category_id = categories.category_id
GROUP BY products.product_id , products.product_name , categories.category_id , categories.category_name)
```

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

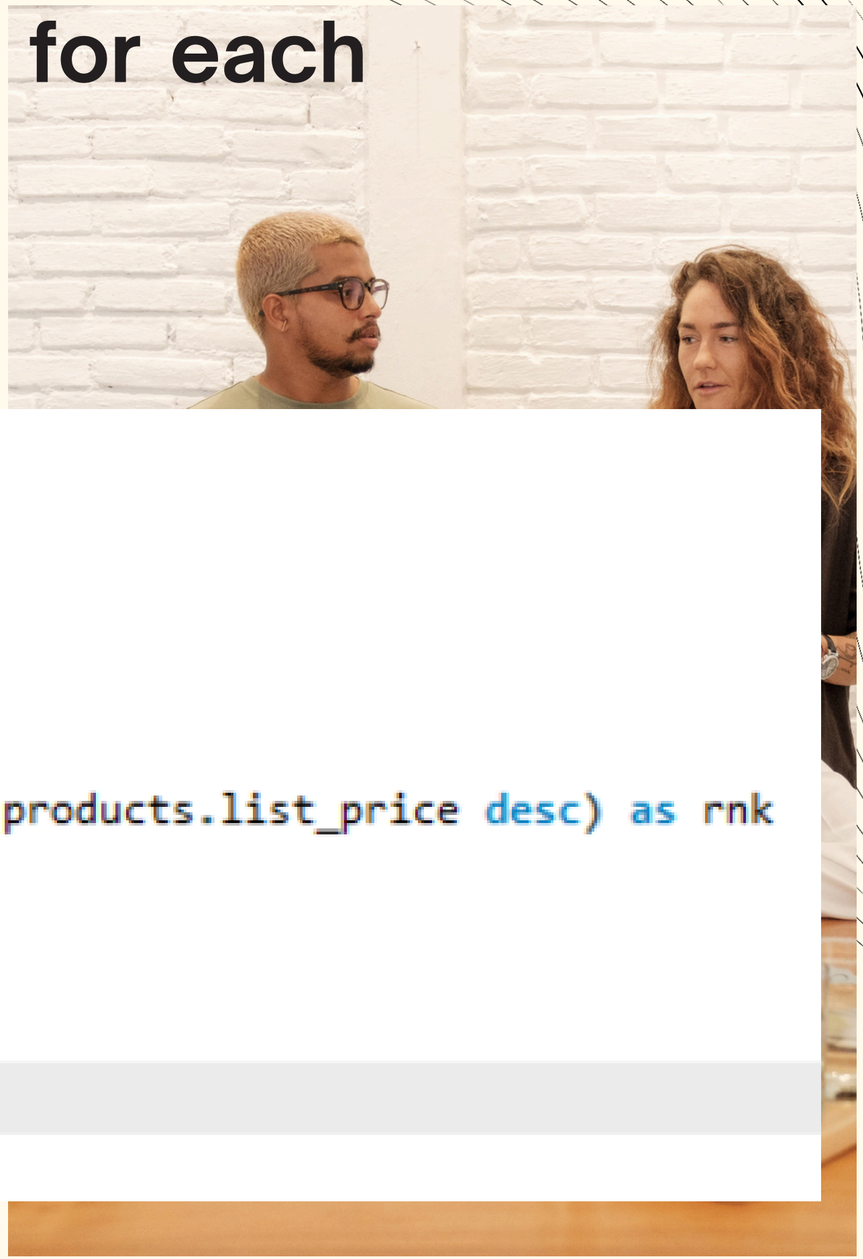
QUERY :

```
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)) 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,  full_name)

select *, rank() over(order by sales desc ) from a;
```

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

QUERY :



```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) as rnk
from products
join categories
on products.category_id=categories.category_id) a
where rnk=1;
```

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

QUERY :

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

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

QUERY :

```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.**

QUERY :

```sql
select product_name from
(SELECT
    products.product_id,
    products.product_name,
    SUM(order_items.quantity) AS total_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;
```

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

QUERY :

```
SELECT
    products.product_id, products.product_name
FROM
    products
WHERE
    NOT EXISTS( SELECT
            product_id
        FROM
            order_items
        WHERE
            products.product_id = order_items.product_id)
```

THANK YOU

LARANA, INC.