Jenson USA offers a comprehensive selection of over 30,000 cycling-related products, including bikes, parts, accessories, and apparel. The company caters to a wide range of riders from casual commuters and road cyclists to competitive mountain bikers and BMX enthusiasts. With a strong e-commerce presence and retail stores in Riverside and Corona, Jenson USA combines the convenience of online shopping with the personali At its core, Jenson USA believes in the power of cycling to transform lives and communities. Its mission, “ Ride, Exper,”

Agenda

This project aims to deliver a comprehensive sales and performance analysis for Jenson USA by leveraging data from various sources like orders, products, customers, staff, and stores. Key objectives include evaluating store performance through total products sold, tracking cumulative product sales over time, and identifying top-selling and highest-priced products in each category. Customer insights will be gained by analyzing spending behavior, order frequency per store, and identifying those who have purchased from each product category. Staff performance will be reviewed by highlighting those with no sales and those exceeding average sales. Additional analysis includes finding the top 3 most sold products, calculating the median product price, and identifying products that have never been ordered. The results will guide strategic decisions in sales, marketing, staffing and inventory management.

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

Input Query

```
SELECT
    s1.store_name,
    SUM(order_items.quantity) AS product_sold
FROM
    stores AS s1
JOIN
    orders AS o1
ON s1.store_id = o1.store_id
JOIN
    order_items
ON o1.order_id = order_items.order_id
GROUP BY s1.store_name;
```

Output Query

store_name	product_sold
Santa Cruz Bikes	1516
Baldwin Bikes	4779
Rowlett Bikes	783

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

Input Query

```
with t1 as(SELECT p1.product_name,
o1.order_date,
sum(order_items.quantity) As total_quantity
FROM products As p1
JOIN order_items
ON p1.product_id = order_items.product_id
JOIN orders As o1
ON o1.order_id = order_items.order_id
GROUP BY p1.product_name,
o1.order_date)
SELECT *,
sum(total_quantity)
over(partition by product_name
order by order_date) As cummulative_quantity
FROM t1;
```

Output Query

product_name	order_date	total_quantity	cummulative_quantity
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-01	1	1
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-01-21	2	3
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	2018-04-30	2	5
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-01-29	2	2
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-02-28	1	3
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-03	1	4
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-03-09	2	6
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-06	1	7
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-15	2	9
Electra Amsterdam Fashion 7i Ladies' - 2017	2017-04-16	1	10

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

Input Query

```
with t1 as(SELECT c1.category_name,
 p1.product_name,
 sum( order_items.quantity* order_items.list_price) As total_sales
 FROM categories As c1
 JOIN products As p1
 ON c1.category_id = p1.category_id
 JOIN order_items
 ON p1.product_id = order_items.product_id
 GROUP BY
 c1.category_name,
 p1.product_name),
 t2 as( SELECT *,
 dense_rank()
 over(partition by category_name
 order by total_sales desc) As rnk FROM t1)
 SELECT * FROM t2
 WHERE rnk <=1;
```

Output Query

category_name	product_name	total_sales	rnk
Children Bicycles	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	4619846.00	1
Comfort Bicycles	Electra Townie Original 7D EQ - 2016	8039866.00	1
Cruisers Bicycles	Electra Townie Original 7D EQ - 2016	9359844.00	1
Cyclocross Bicycles	Surly Straggler 650b - 2016	25382949.00	1
Electric Bikes	Trek Conduit+ - 2016	43499855.00	1
Mountain Bikes	Trek Slash 8 275 - 2016	61599846.00	1
Road Bikes	Trek Domane SLR 6 Disc - 2017	23649957.00	1

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

Input Query

```
with t1 as(SELECT c1.customer_id,
sum(order_items.quantity*order_items.list_price -
(order_items.quantity*order_items.list_price)*((order_items.discount)/100)) as spent_amount
FROM customers As c1
JOIN orders As o1
ON c1.customer_id = o1.customer_id
JOIN order_items
ON order_items.order_id = o1.order_id
GROUP BY
c1.customer_id)
SELECT * FROM (SELECT *, dense_rank()
over(order by spent_amount desc) As rnk from t1) As t2
WHERE rnk <=1;
```

Output Query

customer_id	spent_amount	rnk
10	3729598.42000000	1

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

Input Query

```
SELECT * FROM
(SELECT categories.category_name,
products.product_name,
products.list_price,
max(products.list_price) over(partition by categories.category_name) As max_price
FROM categories JOIN products
ON categories.category_id = products.category_id) As t
WHERE list_price = max_price;
```

Output Query

category_name	product_name	list_price	max_price
Children Bicycles	Trek Superfly 24 - 2017/2018	48999.00	48999.00
Children Bicycles	Electra Straight 8 3i (20-inch) - Boy's - 2017	48999.00	48999.00
Children Bicycles	Electra Townie 3i EQ (20-inch) - Boys' - 2017	48999.00	48999.00
Comfort Bicycles	Electra Townie Go! 8i - 2017/2018	259999.00	259999.00
Cruisers Bicycles	Electra Townie Commute Go! - 2018	299999.00	299999.00
Cruisers Bicycles	Electra Townie Commute Go! Ladies' - 2018	299999.00	299999.00
Cyclocross Bicycles	Trek Boone 7 Disc - 2018	399999.00	399999.00
Electric Bikes	Trek Powerfly 8 FS Plus - 2017	499999.00	499999.00
Electric Bikes	Trek Super Commuter+ 8S - 2018	499999.00	499999.00
Electric Bikes	Trek Powerfly 7 FS - 2018	499999.00	499999.00
Mountain Bikes	Trek Fuel EX 98 275 Plus - 2017	529999.00	529999.00
Mountain Bikes	Trek Remedy 98 - 2017	529999.00	529999.00
Road Bikes	Trek Domane SLR 9 Disc - 2018	1199999.00	1199999.00

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

Input Query

```
SELECT DISTINCT customers.customer_id, customers.first_name,
stores.store_name,
count(orders.order_id) over(partition by customers.customer_id, stores.store_name)As total_order
FROM customers JOIN orders
ON customers.customer_id = orders.customer_id
JOIN stores
ON orders.store_id = stores.store_id;
```

Output Query

customer_id	first_name	store_name	total_order
1	Debra	Baldwin Bikes	3
2	Kasha	Santa Cruz Bikes	3
3	Tameka	Santa Cruz Bikes	3
4	Daryl	Baldwin Bikes	3
5	Charolette	Santa Cruz Bikes	3
6	Lyndsey	Baldwin Bikes	3
7	Latasha	Baldwin Bikes	3
8	Jacquline	Baldwin Bikes	3
9	Genoveva	Baldwin Bikes	3
10	Pamelia	Baldwin Bikes	3
11	Deshawn	Baldwin Bikes	3
12	Robby	Baldwin Bikes	3
13	Lashawn	Rowlett Bikes	3
14	Garry	Rowlett Bikes	3
15	Lippie	Baldwin Bikes	3

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

Input Query

```
SELECT staffs.first_name,
count(orders.order_id)As sales FROM staffs
LEFT JOIN orders
ON staffs.staff_id = orders.staff_id
GROUP BY staffs.first_name
HAVING count(order_id) = 0;
```

Output Query

first_name	sales
Fabiola	0
Virgie	0
Jannette	0
Bernardine	0

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

Input Query

```
with t1 as (SELECT products.product_name,
sum(order_items.quantity) As quantity_sold
FROM products JOIN order_items
ON products.product_id = order_items.product_id
GROUP BY products.product_name),
t2 as(SELECT *,
dense_rank() over(order by quantity_sold desc) As top3_products
FROM t1)
SELECT * FROM t2
WHERE top3_products <=3;
```

Output Query

product_name	quantity_sold	top3_products
Electra Cruiser 1 (24-Inch) - 2016	296	1
Electra Townie Original 7D EQ - 2016	290	2
Electra Townie Original 21D - 2016	289	3

9. Find the median value of the price list.

Input Query

```
 with t as(SELECT list_price,
row_number() over(order by list_price) As pos,
count(*) over() As n FROM order_items)
SELECT
CASE
when n % 2=0 then (SELECT round(avg(list_price),0) FROM t
WHERE pos in ((n/2), (n/2)+1))
else (SELECT round(list_price,0) FROM t WHERE pos = (n+1)/2)
end as median
FROM t
limit 1;
```

Output Query

	median
▶	59999

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

Input Query

```
SELECT products.product_name FROM products
where not exists (SELECT product_id FROM order_items
WHERE order_items.product_id = products.product_id);
```

Output Query

product_name
Trek 820 - 2016
Surly Krampus Frameset - 2018
Trek Kids' Dual Sport - 2018
Trek Domane SLR 6 Disc Women's - 2018
Electra Townie Go! 8i Ladies' - 2018
Trek Precaliber 12 Girl's - 2018
Electra Savannah 1 (20-inch) - Girl's - 2018
Electra Sweet Ride 1 (20-inch) - Girl's - 2018
Trek Checkpoint ALR 4 Women's - 2019
Trek Checkpoint ALR 5 - 2019
Trek Checkpoint ALR 5 Women's - 2019
Trek Checkpoint SL 5 Women's - 2019
Trek Checkpoint SL 6 - 2019
Trek Checkpoint ALR Frameset - 2019

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

Input Query

```
with t as(SELECT staffs.first_name,
coalesce(sum(order_items.list_price* order_items.quantity),0) total_sales
FROM staffs LEFT JOIN orders
ON orders.staff_id = staffs.staff_id
LEFT JOIN order_items
ON orders.order_id = order_items.order_id
GROUP BY staffs.first_name)
SELECT * FROM t
WHERE total_sales > (SELECT avg(total_sales) FROM t);
```

Output Query

first_name	total_sales
Genna	95272226.00
Marcelene	293888873.00
Venita	288735348.00

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

Input Query

```
SELECT customers.customer_id, customers.first_name,
count(order_items.product_id) As total_products
FROM customers JOIN orders
ON customers.customer_id = orders.customer_id
JOIN order_items
ON orders.order_id = order_items.order_id
JOIN products
ON order_items.product_id = products.product_id
GROUP BY customers.customer_id, customers.first_name
HAVING count(DISTINCT (category_id)) = (SELECT count(category_id)
 FROM categories);
```

Output Query

customer_id	first_name	total_products
9	Genoveva	9

THANK YOU

By:- Jaswinder Kaur