

JENSON

USA



BY: CHINMAY SINGOLE

INTRODUCTION

JENSON USA, ESTABLISHED IN 1994 BY MIKE CACHAT, IS A PROMINENT RETAILER SPECIALIZING IN BICYCLES, COMPONENTS, APPAREL, AND ACCESSORIES. WITH BOTH AN ONLINE PRESENCE AND PHYSICAL RETAIL LOCATIONS IN CORONA AND RIVERSIDE, CALIFORNIA, THE COMPANY OFFERS A DIVERSE SELECTION OF OVER 30,000 PRODUCTS CATERING TO MOUNTAIN BIKING, ROAD CYCLING, AND ELECTRIC BIKES. IN DECEMBER 2024, JENSON USA INTRODUCED AN INNOVATIVE BICYCLE SHIPPING SERVICE, JENSON BIKE SHIPPING, IN COLLABORATION WITH UPS. THIS SERVICE OFFERS CUSTOMERS DISCOUNTED SHIPPING LABELS, WITH SAVINGS OF UP TO 50%, ENHANCING THE AFFORDABILITY AND CONVENIENCE OF SHIPPING BIKES AND RELATED GEAR.

BY: CHINMAY SINGOLE

WRITE QUERIES FOR THE FOLLOWING QUESTIONS:

1. Find the total number of products sold by each store along with the store name.
2. Calculate the cumulative sum of quantities sold for each product over time.
3. Find the product with the highest total sales (quantity * price) for each category.
4. Find the customer who spent the most money on orders.
5. Find the highest-priced product for each category name.
6. Find the total number of orders placed by each customer per store.
7. Find the names of staff members who have not made any sales.
8. Find the top 3 most sold products in terms of quantity.
9. 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).

BY: CHINMAY SINGOLE

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

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

→ It will return total number of products sold by each store_name

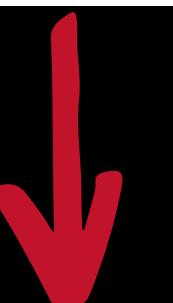


store_name	total_products
Santa Cruz Bikes	1516
Baldwin Bikes	4779
Rowlett Bikes	783

BY: CHINMAY SINGOLE

#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) Cumulative_sum  
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
```



It wil return Cumulative_sum →
of Product quantities with
respect to Time

product_id	order_date	quantity	Cumulative_sum
2	2016-01-03	2	2
2	2016-01-14	2	4
2	2016-01-18	1	5
2	2016-02-05	1	6
2	2016-02-09	1	7
2	2016-02-26	1	8
2	2016-02-28	2	10
2	2016-03-08	1	11
2	2016-03-14	2	13
2	2016-03-20	4	17
2	2016-03-21	1	18
2	2016-03-28	1	19
2	2016-04-08	1	20
2	2016-04-13	2	22
2	2016-04-15	2	24
2	2016-04-27	1	25
2	2016-04-30	2	27
2	2016-05-07	2	29
2	2016-05-12	2	31
2	2016-05-13	2	33
2	2016-05-17	3	36
2	2016-05-25	2	38
2	2016-06-03	2	40
2	2016-06-06	2	42
2	2016-06-08	1	43
2	2016-06-09	2	45

BY: CHINMAY SINGOLE

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

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



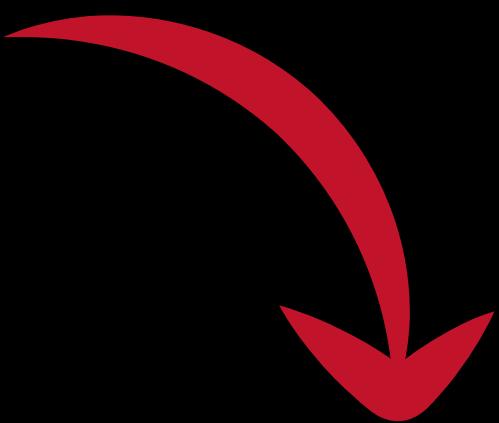
It will return the highest Total Sales for each Category

	category_id	category_name	product_id	product_name	Sales	rnk
▶	1	Children Bicycles	23	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	4619278.00	1
	2	Comfort Bicycles	26	Electra Townie Original 7D EQ - 2016	8039320.00	1
	3	Cruisers Bicycles	16	Electra Townie Original 7D EQ - 2016	9359304.00	1
	4	Cyclocross Bicycles	11	Surly Straggler 650b - 2016	25382383.00	1
	5	Electric Bikes	9	Trek Conduit+ - 2016	43499347.00	1
	6	Mountain Bikes	7	Trek Slash 8 275 - 2016	61599226.00	1
	7	Road Bikes	56	Trek Domane SLR 6 Disc - 2017	23649774.00	1

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

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

It will return the customer_id and there name who have spend most money on orders



	customer_id	full_name	sales	rnk
▶	10	Pamelia Newman	3780140.00	1

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

```
WITH MaxPricePerCategory AS (
  SELECT category_id,
         MAX(list_price) AS max_price
    FROM products
   GROUP BY category_id
)
SELECT c. category_name, p. product_name, p.list_price
  FROM MaxPricePerCategory mp
 JOIN products p ON mp. category_id = p. category_id AND mp.max_price = p.list_price
 JOIN
categories c ON p. category_id = c. category_id;
```

It will return
product_names which
have the highest price in
each category

category_name	product_name	list_price
Children Bicycles	Electra Cruiser 1 (24-Inch) - 2016	26999.00
Children Bicycles	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	26999.00
Children Bicycles	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	29999.00
Children Bicycles	Trek Boy's Kickster - 2015/2017	14999.00
Children Bicycles	Sun Bicycles Lil Kitt'n - 2017	10999.00
Children Bicycles	Haro Downtown 16 - 2017	32999.00
Children Bicycles	Trek Girl's Kickster - 2017	14999.00
Children Bicycles	Trek Precaliber 12 Boys - 2017	18999.00
Children Bicycles	Trek Precaliber 12 Girls - 2017	18999.00
Children Bicycles	Trek Precaliber 16 Boys - 2017	20999.00
Children Bicycles	Trek Precaliber 16 Girls - 2017	20999.00
Children Bicycles	Trek Precaliber 24 (21-Speed) - Girls - 2017	34999.00
Children Bicycles	Haro Shredder 20 - 2017	20999.00
Children Bicycles	Haro Shredder 20 Girls - 2017	20999.00
Children Bicycles	Haro Shredder Pro 20 - 2017	24999.00
Children Bicycles	Electra Girl's Hawaii 1 16" - 2017	29999.00
Children Bicycles	Electra Moto 3i (20-inch) - Boy's - 2017	34999.00
Children Bicycles	Electra Savannah 3i (20-inch) - Girl's - 2017	34999.00
Children Bicycles	Electra Sugar Skulls 1 (20-inch) - Girl's - 2017	29999.00
Children Bicycles	Electra Townie 7D (20-inch) - Boys' - 2017	33999.00
Children Bicycles	Trek MT 201 - 2018	24999.00
Children Bicycles	Strider Classic 12 Balance Bike - 2018	8999.00
Children Bicycles	Strider Sport 16 - 2018	24999.00
Children Bicycles	Strider Strider 20 Sport - 2018	28999.00
Children Bicycles	Trek Superfly 20 - 2018	39999.00
Children Bicycles	Trek Precaliber 12 Girl's - 2018	19999.00
Children Bicycles	Trek Kickster - 2018	15999.00

BY: CHINMAY SINGOLE

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

```
select store_id , customer_id, COUNT(*) total_no_orders  
from orders  
group by store_id, customer_id  
order by store_id, customer_id;
```



It will return Customer Name who has spent the most money on orders



store_id	customer_id	total_no_orders
1	2	3
1	3	3
1	5	3
1	24	3
1	30	3
1	31	3
1	32	3
1	33	3
1	40	3
1	46	3
1	47	3
1	53	3
1	60	2
1	67	2
1	69	2
1	72	2
1	78	2
1	81	2
1	82	2
1	87	2
1	89	2
1	91	2
1	97	2
1	104	2
1	109	2
1	110	2
1	120	2

BY: CHINMAY SINGOLE

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

```
select staffs.staff_id,  
concat(staffs.first_name, " ", staffs.last_name) full_name  
from staffs where not exists  
(select staff_id from orders  
where orders.staff_id = staffs.staff_id);
```

It will return staff_id and
Staff Name who have not
made any sales



	staff_id	full_name
▶	1	Fabiola Jackson
	4	Virgie Wiggins
	5	Jannette David
	10	Bernardine Houston

BY: CHINMAY SINGOLE

#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;
```

It will return Top
Three sold Product
Name based on
Quantity



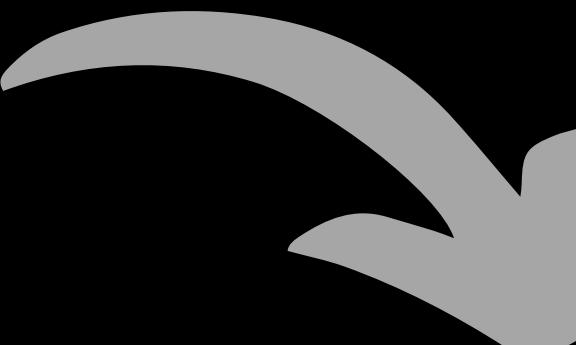
product_name
Surly Ice Cream Truck Frameset - 2016
Electra Cruiser 1 (24-Inch) - 2016
Electra Townie Original 7D EQ - 2016

BY: CHINMAY SINGOLE

#Find the median value of the price list.

```
WITH m AS ( SELECT list_price,  
ROW_NUMBER() OVER (ORDER BY list_price) As rn,  
COUNT(*) OVER () AS 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;
```

It will return middle
value of Price_list
also know as median



Result Grid	
	median
▶	59999.000000

BY: CHINMAY SINGOLE

#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);
```

It will return Product Name with Product_id thst have never been ordered

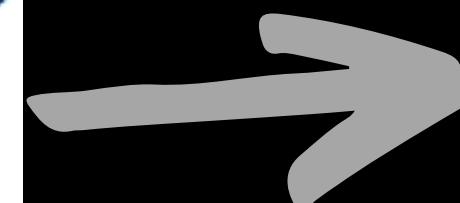
product_id	product_name
1	Trek 820 - 2016
121	Surly Krampus Frameset - 2018
125	Trek Kids' Dual Sport - 2018
154	Trek Domane SLR 6 Disc Women's - 2018
195	Electra Townie Go! 8i Ladies' - 2018
267	Trek Precaliber 12 Girl's - 2018
284	Electra Savannah 1 (20-inch) - Girl's - 2018
291	Electra Sweet Ride 1 (20-inch) - Girl's - 2018
316	Trek Checkpoint ALR 4 Women's - 2019
317	Trek Checkpoint ALR 5 - 2019
318	Trek Checkpoint ALR 5 Women's - 2019
319	Trek Checkpoint SL 5 Women's - 2019
320	Trek Checkpoint SL 6 - 2019
321	Trek Checkpoint ALR Frameset - 2019
HULL	HULL

BY: CHINMAY SINGOLE

#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) 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) 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);
```



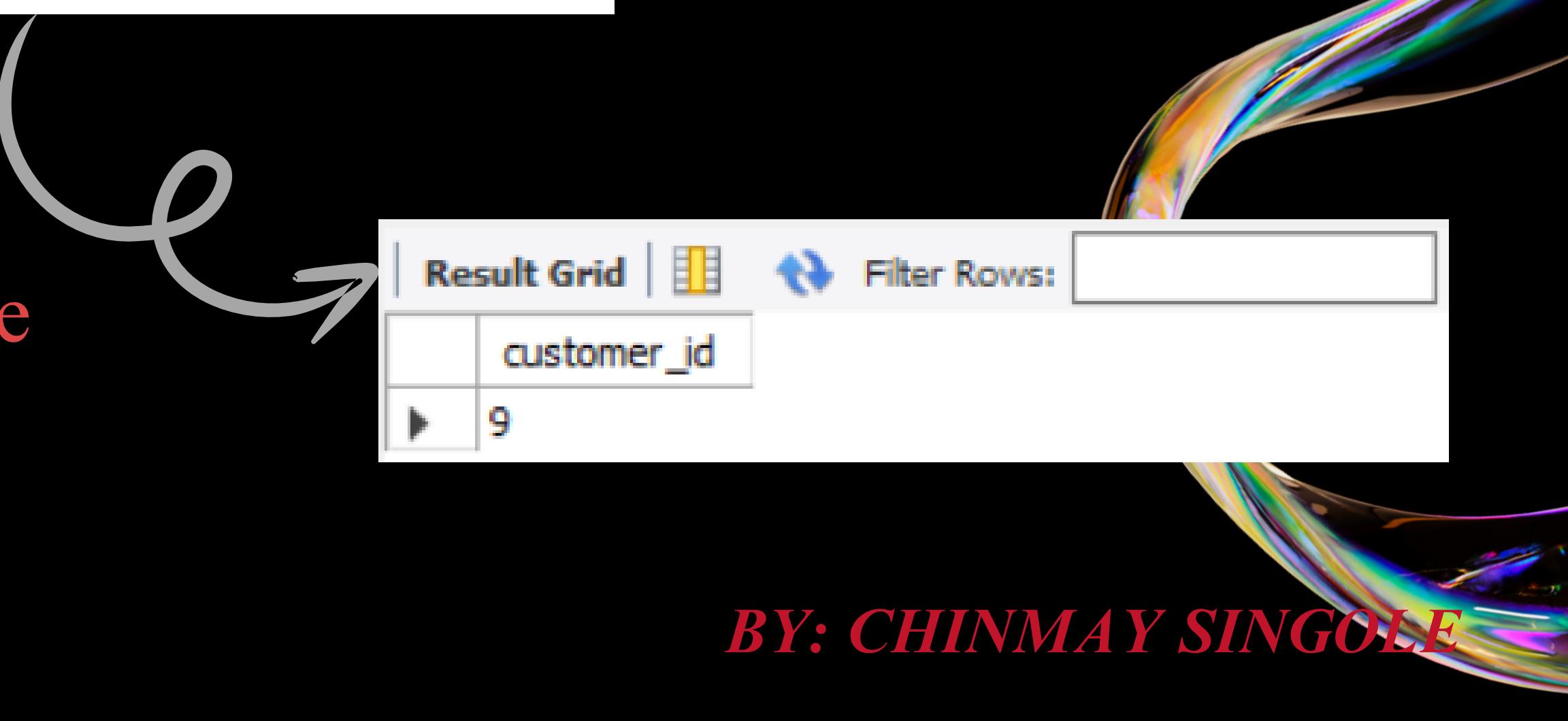
	staff_id	sales
▶	3	95269072.00
	6	293879916.00
	7	288726544.00

BY: CHINMAY SINGOLE

#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);  
  
select count (category_id) from categories
```

It will return
Customer_id who have
ordered all types of
product



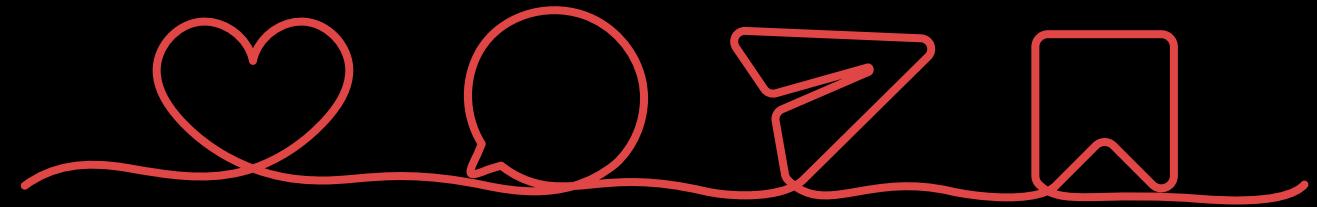
A screenshot of a database query results interface. At the top, there are buttons for "Result Grid" (with a grid icon), "Filter Rows:" (with a magnifying glass icon), and a search bar. Below this is a table with one column labeled "customer_id". There is one row in the table containing the value "9".

customer_id
9

BY: CHINMAY SINGOLE

THANK YOU...

Like, Comment, Share and Follow



Connect me on Linkedin



Chinmay Singole

