Jenson USA – Data Insights via SQL - Milestone-2

by Shubham Verma

Project Objective

- To derive actionable business insights using SQL from the Jenson USA dataset.
- Utilize advanced SQL techniques including:
    - Joins & Subqueries
    - Aggregations & Window Functions
    - CTEs & EXISTS filtering

Dataset:

https://drive.google.com/drive/folders/1feFkClnYME7Be3kjmz-TD2PV1uVkXNAN

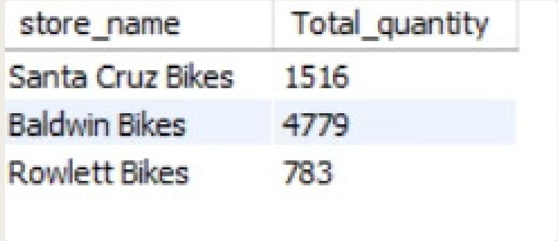
Analysis Questions

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).

Analysis Questions

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

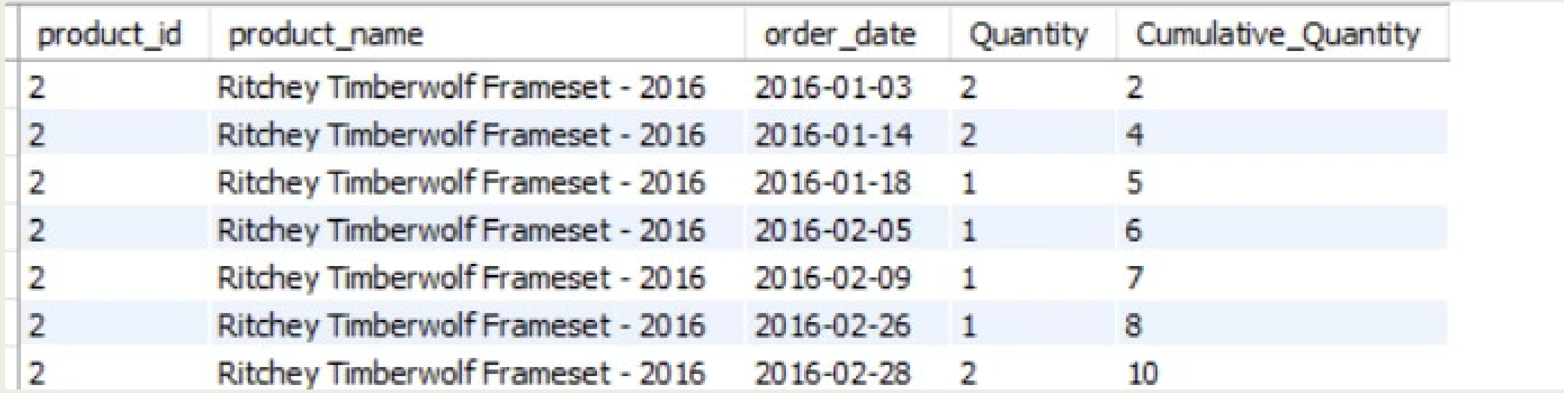
```
select stores.store_name,

sum(order_items.quantity)  as Total_quantity

from stores join orders  using (Store_id)

join order_items using (order_id)

group by stores.store_name  ;
```



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

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

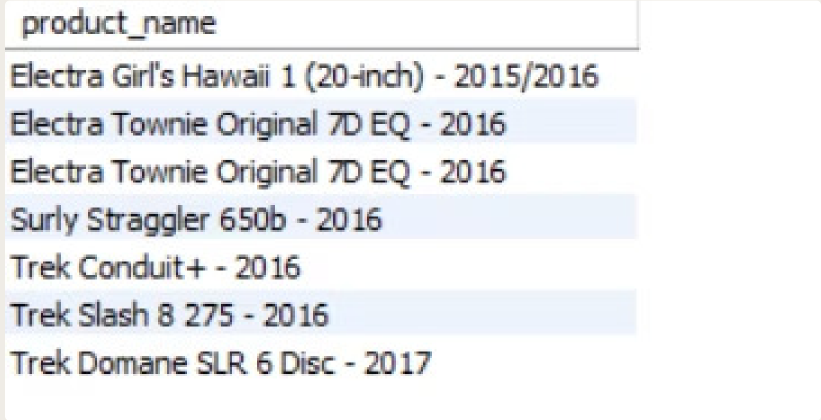
```
with a as

( Select products.product_id, products.product_name, orders.order_date,
sum(order_items.quantity) as Quantity

from products

join order_items using (product_id)

join orders using (order_id)

group by  products.product_id, products.product_name, orders.order_date )

Select *, sum(quantity) over (partition by product_id order by order_date) as  Cumulative_Quantity from a ;
```



product_id	product_name	order_date	Quantity	Cumulative_Quantity
2	Ritchey Timberwolf Frameset - 2016	2016-01-03	2	2
2	Ritchey Timberwolf Frameset - 2016	2016-01-14	2	4
2	Ritchey Timberwolf Frameset - 2016	2016-01-18	1	5
2	Ritchey Timberwolf Frameset - 2016	2016-02-05	1	6
2	Ritchey Timberwolf Frameset - 2016	2016-02-09	1	7
2	Ritchey Timberwolf Frameset - 2016	2016-02-26	1	8
2	Ritchey Timberwolf Frameset - 2016	2016-02-28	2	10

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

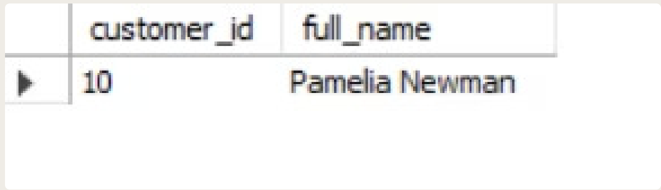
```
with a as
(
 Select products.product_id, products.product_name, products.category_id,
sum(order_items.quantity * order_items.List_price) as Sales
 from products join order_items using (product_id)
group by products.product_id, products.product_name, products.category_id
)
 Select product_name from
(
 Select*, rank() over (partition by category_id order by sales desc ) rnk from a
) b
where rnk = 1;
```



product_name
Electra Girl's Hawaii 1 (20-inch) - 2015/2016
Electra Townie Original 7D EQ - 2016
Electra Townie Original 7D EQ - 2016
Surly Straggler 650b - 2016
Trek Conduit+ - 2016
Trek Slash 8 275 - 2016
Trek Domane SLR 6 Disc - 2017

Find the customer who spent the most money on orders.

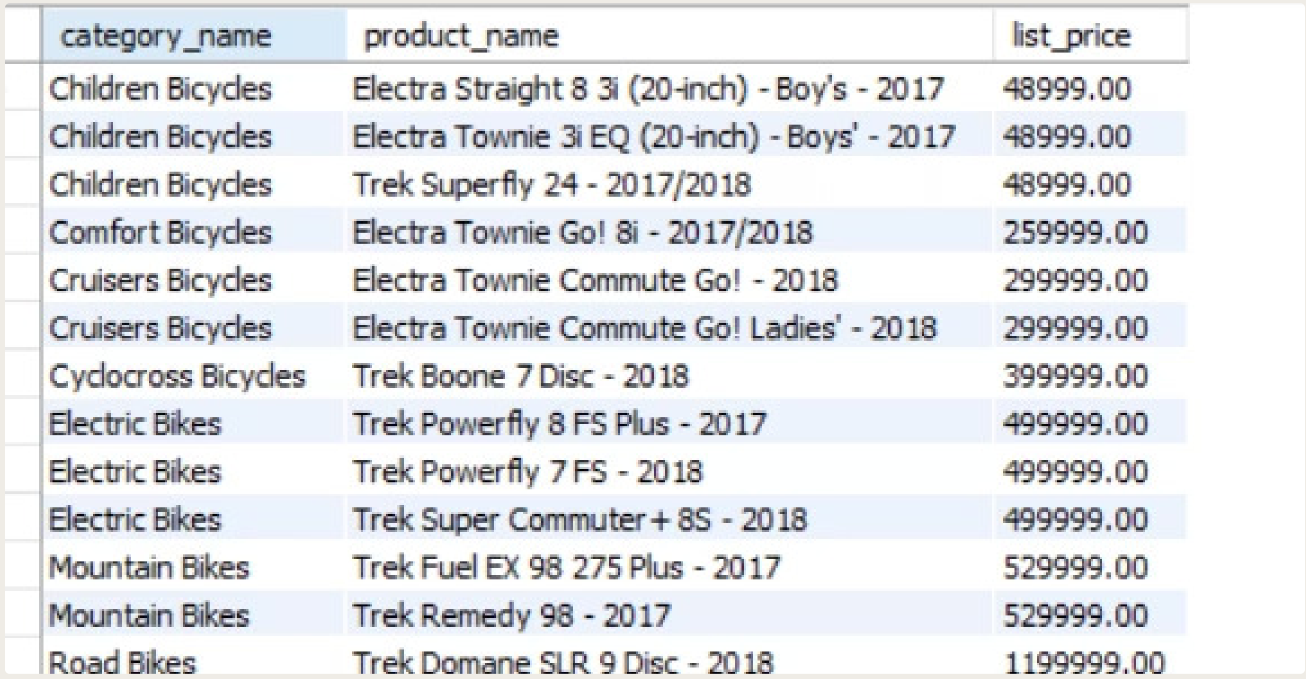
```
with a as

(

select customers.customer_id, concat(customers.first_name, " " ,customers.last_name) as full_name,

sum(order_items.quantity * order_items.list_price) as sales

from customers join orders using(customer_id) join order_items using (order_id)

group by customers.customer_id, full_name

)

select customer_id, full_name from a

where sales = (select  max(Sales) from a );
```



customer_id	full_name
10	Pamelia Newman

Find the highest-priced product for each category name.

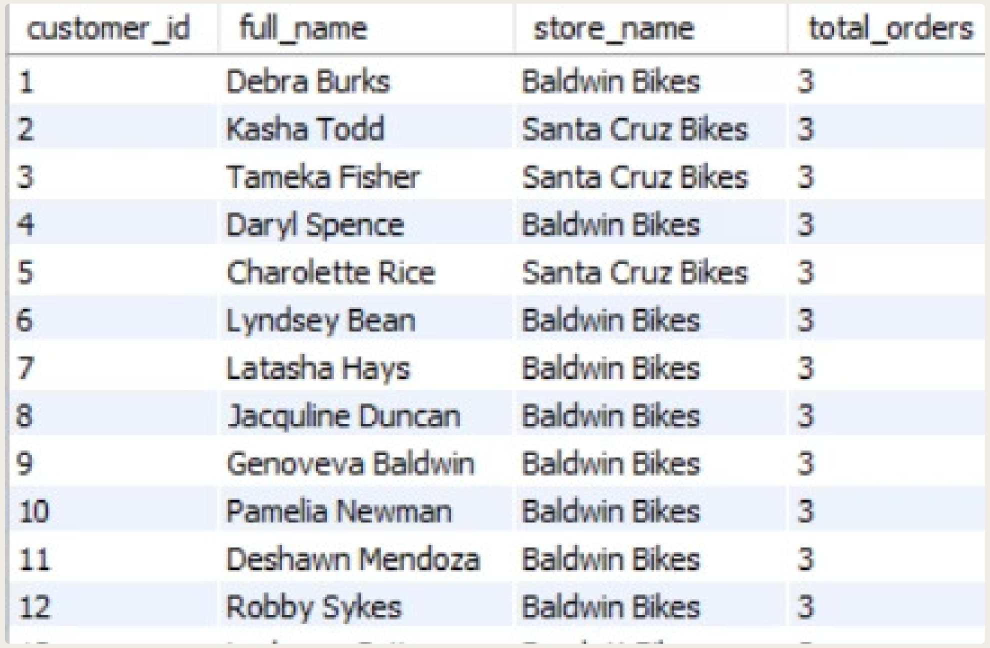
```
with a as

(

select categories.category_name, products.product_name, products.list_price,

rank() over (partition by products.category_id order by products.list_price desc)

as rnk

from products

join categories using(category_id)

)

select category_name, product_name, list_price

from a

where rnk = 1 ;
```



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

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

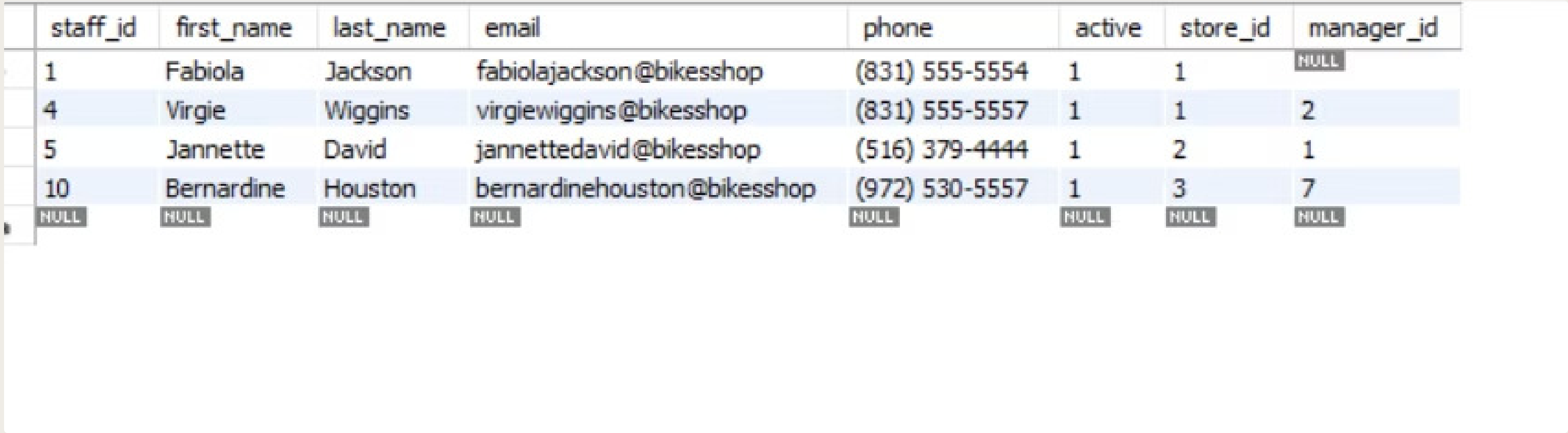
```
select customers.customer_id,

concat(customers.first_name, " " , customers.last_name) as full_name, stores.store_name,

count(orders.order_id) as total_orders

from customers

left join orders using(Customer_id)

left join stores using (store_id)

group by customers.customer_id, full_name, stores.store_name ;
```



customer_id	full_name	store_name	total_orders
1	Debra Burks	Baldwin Bikes	3
2	Kasha Todd	Santa Cruz Bikes	3
3	Tameka Fisher	Santa Cruz Bikes	3
4	Daryl Spence	Baldwin Bikes	3
5	Charolette Rice	Santa Cruz Bikes	3
6	Lyndsey Bean	Baldwin Bikes	3
7	Latasha Hays	Baldwin Bikes	3
8	Jacquline Duncan	Baldwin Bikes	3
9	Genoveva Baldwin	Baldwin Bikes	3
10	Pamelia Newman	Baldwin Bikes	3
11	Deshawn Mendoza	Baldwin Bikes	3
12	Robby Sykes	Baldwin Bikes	3

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

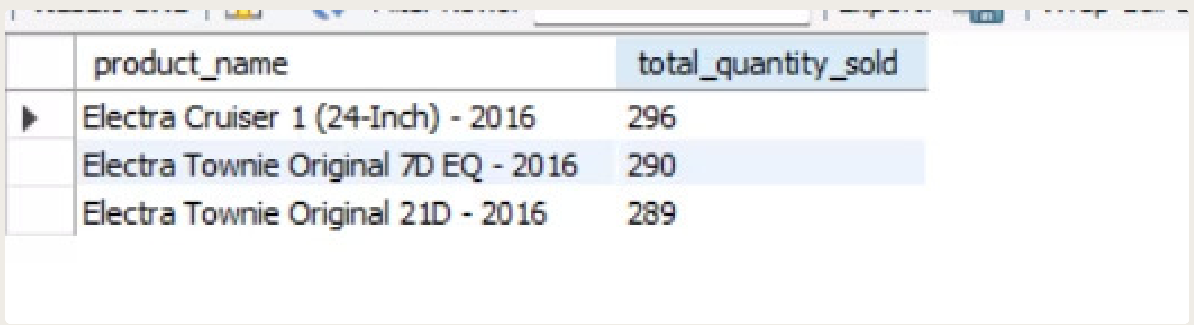
```
select * from staffs

where staff_id not in (select staff_id from orders)
```



staff_id	first_name	last_name	email	phone	active	store_id	manager_id
1	Fabiola	Jackson	fabiolajackson@bikesshop	(831) 555-5554	1	1	NULL
4	Virgie	Wiggins	virgiewiggins@bikesshop	(831) 555-5557	1	1	2
5	Jannette	David	jannettedavid@bikesshop	(516) 379-4444	1	2	1
10	Bernardine	Houston	bernardinehouston@bikesshop	(972) 530-5557	1	3	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

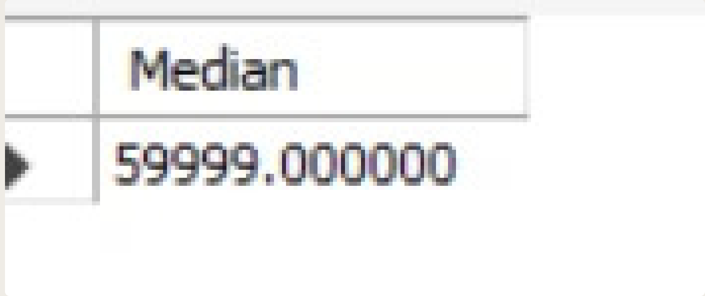
```
select product_name, sum(quantity) as total_quantity_sold

from order_items

join products using(product_id)

group by product_name

order by total_quantity_sold desc

limit 3 ;
```



product_name	total_quantity_sold
Electra Cruiser 1 (24-Inch) - 2016	296
Electra Townie Original 7D EQ - 2016	290
Electra Townie Original 21D - 2016	289

Find the median value of the price list.

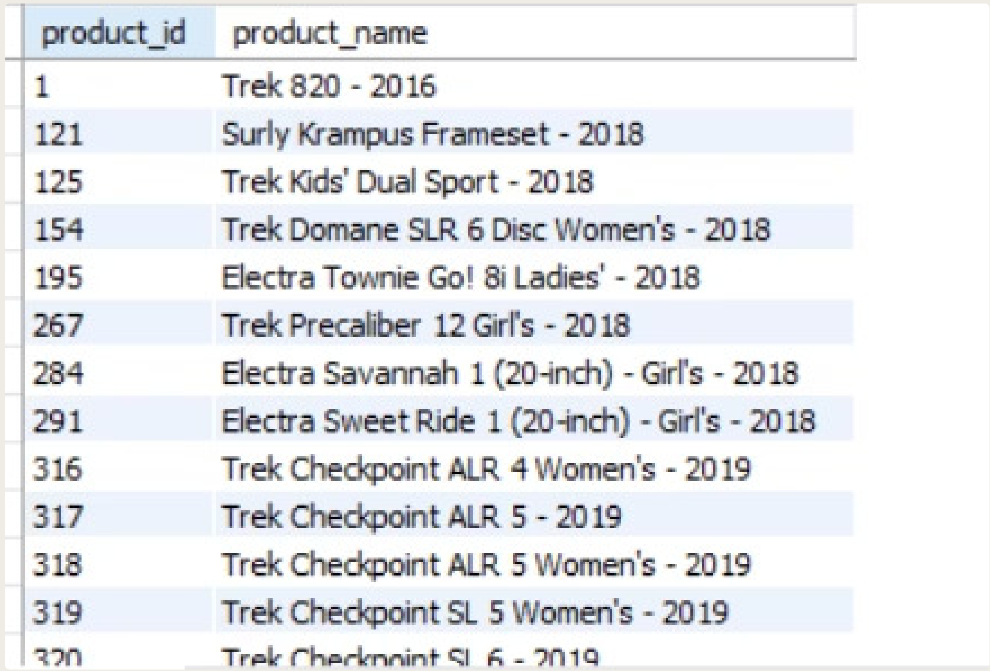
```
with a as (

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 avg(list_price) from a where pos in ((n/2), (n/2)+1)    )

else (    select list_price from a where pos = (n+1)/2   )

end as Median from a limit 1 ;
```



```
Median
59999.000000
```

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

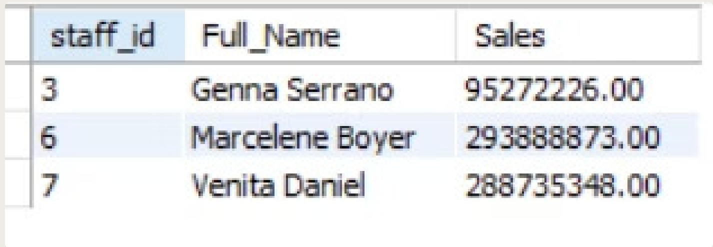
```
select product_id, product_name from products

where not exists

(

select product_id from order_items

where products.product_id = order_items.product_id

);
```



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

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

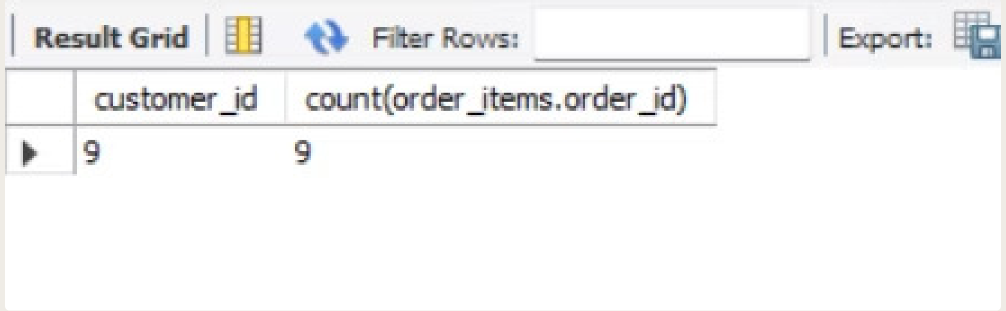
```
with a as

( select staffs.staff_id, concat(staffs.first_name , " " , staffs.last_name) as Full_Name,

coalesce(sum(order_items.list_price * order_items.quantity) ,0) Sales

from staffs left join orders using (staff_id)

left join order_items using (order_id)

group by staffs.staff_id, Full_Name

)

select * from a

where sales > (Select avg (sales) from a );
```



staff_id	Full_Name	Sales
3	Genna Serrano	95272226.00
6	Marcelene Boyer	293888873.00
7	Venita Daniel	288735348.00

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

```
select customers.customer_id, count(order_items.order_id)

from customers

join orders using (customer_id)

join order_items using (order_id)

join products using (product_id)

group by customers.customer_id

having count(distinct products.category_id) = (select count(category_id) from categories);
```



customer_id	count(order_items.order_id)
9	9

Conclusion

This project used SQL to explore Jenson USA's business data—highlighting best-selling products, top customers, and staff performance. The insights gained can help the company make smarter, data-driven decisions.

Thank you!