

# JENSON USA

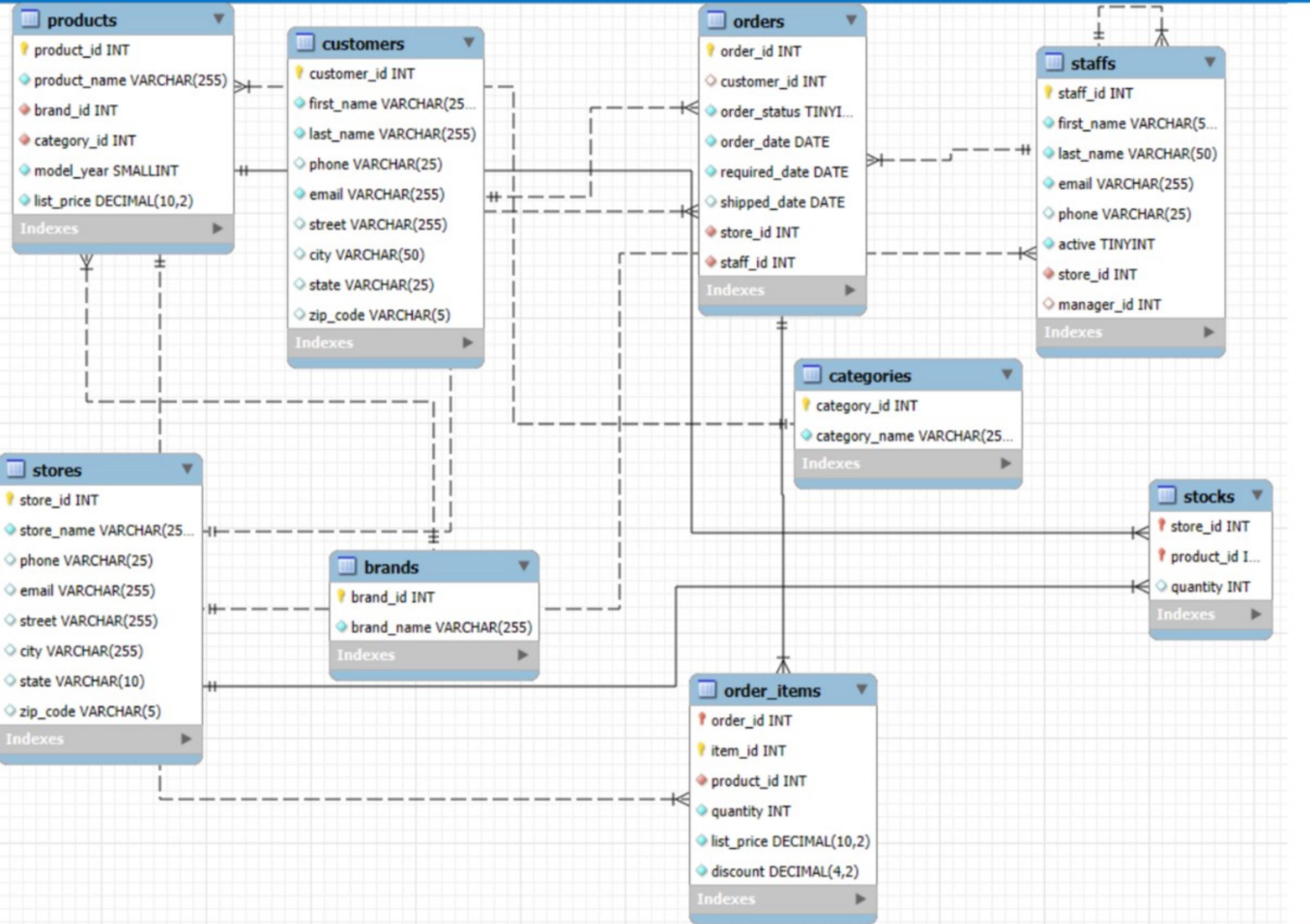
## SQL ANALYSIS



# ABOUT COMPANY

JENSON USA Jenson USA is a leading American bicycle retailer founded in 1994, driven by a passion for cycling and a commitment to serving riders of all levels. We offer a wide selection of bikes, parts, gear, and accessories from top brands, backed by expert support from a team of fellow enthusiasts. With both a strong online presence and physical locations, we deliver fast, reliable service and personalized guidance to help our customers ride with confidence. At Jenson USA, we're not just about selling bikes—we're about building a community that rides, explores, and grows together.





# Relational Schema Diagram

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

```
SELECT  
    s.store_name,  
    (SELECT  
        SUM(ot.quantity)  
    FROM  
        orders o  
        JOIN  
        order_items ot ON o.order_id = ot.order_id  
    WHERE  
        o.store_id = s.store_id) AS total_quantity
```

FROM

stores s;

Output

	store_name	total_quantity
▶	Santa Cruz Bikes	1516
	Baldwin Bikes	4779
	Rowlett Bikes	783



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

```
select p.product_name , o.order_date, ot.quantity, sum(ot.quantity)
over(partition by p.product_name order by o.order_date) total_quantity
from products p join order_items ot
on ot.product_id = p.product_id join orders o on o.order_id =
ot.order_id
```



### Output

product_name	order_date	quantity	total_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



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

```
' with a as (select product_name , c.category_name, sum(ot.quantity*p.list_price) sales  
  from products p join categories c on p.category_id = c.category_id join order_items  
  ot on ot.product_id = p.product_id  
 group by product_name, c.category_id)  
 select * from  
(select *, dense_rank() over(partition by category_name order by sales desc) rnk from a ) b  
where rnk =1;
```



### Output

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



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

```
with a as (select c.customer_id,  
sum(ot.quantity * ot.list_price-(ot.list_price* ot.quantity)*((ot.discount)/100))  
amount from customers c join orders o on o.customer_id =  
c.customer_id join order_items ot on ot.order_id = o.order_id group by c.customer_id)  
select * from (select *, rank() over(order by amount desc) rnk from a) b where rnk =1
```



#### Output

customer_id	amount	rnk
10	3729598.42000000	1



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

```
select * from (select c.category_name ,p.product_name, p.list_price, max(p.list_price)
 over(partition by c.category_name) max_price from products p join
 categories c on c.category_id = p.category_id ) b
where list_price = max_price
```



### Output

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



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

```
select s.store_name, c.customer_id, count(o.order_id) from orders o join customers c  
on o.customer_id = c.customer_id  
join stores s on o.store_id = s.store_id group by s.store_name , c.customer_id
```



### Output

store_name	customer_id	count(o.order_id)
Santa Cruz Bikes	46	3
Santa Cruz Bikes	32	3
Santa Cruz Bikes	31	3
Santa Cruz Bikes	24	3
Santa Cruz Bikes	5	3
...		



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

```
SELECT st.staff_id, st.first_name, st.last_name  
FROM staffs st  
LEFT JOIN orders o ON o.staff_id = st.staff_id  
GROUP BY st.staff_id, st.first_name, st.last_name  
HAVING COUNT(o.order_id) = 0;
```



### Output

staff_id	first_name	last_name
1	Fabiola	Jackson
4	Virgie	Wiggins
5	Jannette	David
10	Bernardine	Houston



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

```
with a as (select p.product_name, p.product_id, sum(ot.quantity) total_quantity from products p join  
order_items ot on p.product_id = ot.product_id  
group by p.product_name, p.product_id)  
select * from (select *, row_number() over(order by total_quantity desc) rnk from a) b  
where rnk <=3;
```



### Output

product_name	product_id	total_quantity	rnk
Surly Ice Cream Truck Frameset - 2016	6	167	1
Electra Cruiser 1 (24-Inch) - 2016	13	157	2
Electra Townie Original 7D EQ - 2016	16	156	3



## 9.Find the median value of the price list.

```
with a as (select list_price, row_number() over(order by list_price) price ,  
count(*) over() n  from order_items)  
select *, case when n%2 =0 then(select avg(list_price) from a where price in ((n/2),(n/2)+1))  
else (select avg(list_price) from a where price = (n+1)/2)  
end as median  
from a ;
```



**Output**

list_price	price	n	median
8999.00	1	4722	59999.000000
8999.00	2	4722	59999.000000
8999.00	3	4722	59999.000000
8999.00	4	4722	59999.000000
8999.00	5	4722	59999.000000



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

```
SELECT p.product_id, p.product_name  
FROM products p  
WHERE NOT EXISTS (  
    SELECT 1  
    FROM order_items ot  
    WHERE ot.product_id = p.product_id
```



Output

product_id	product_name
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

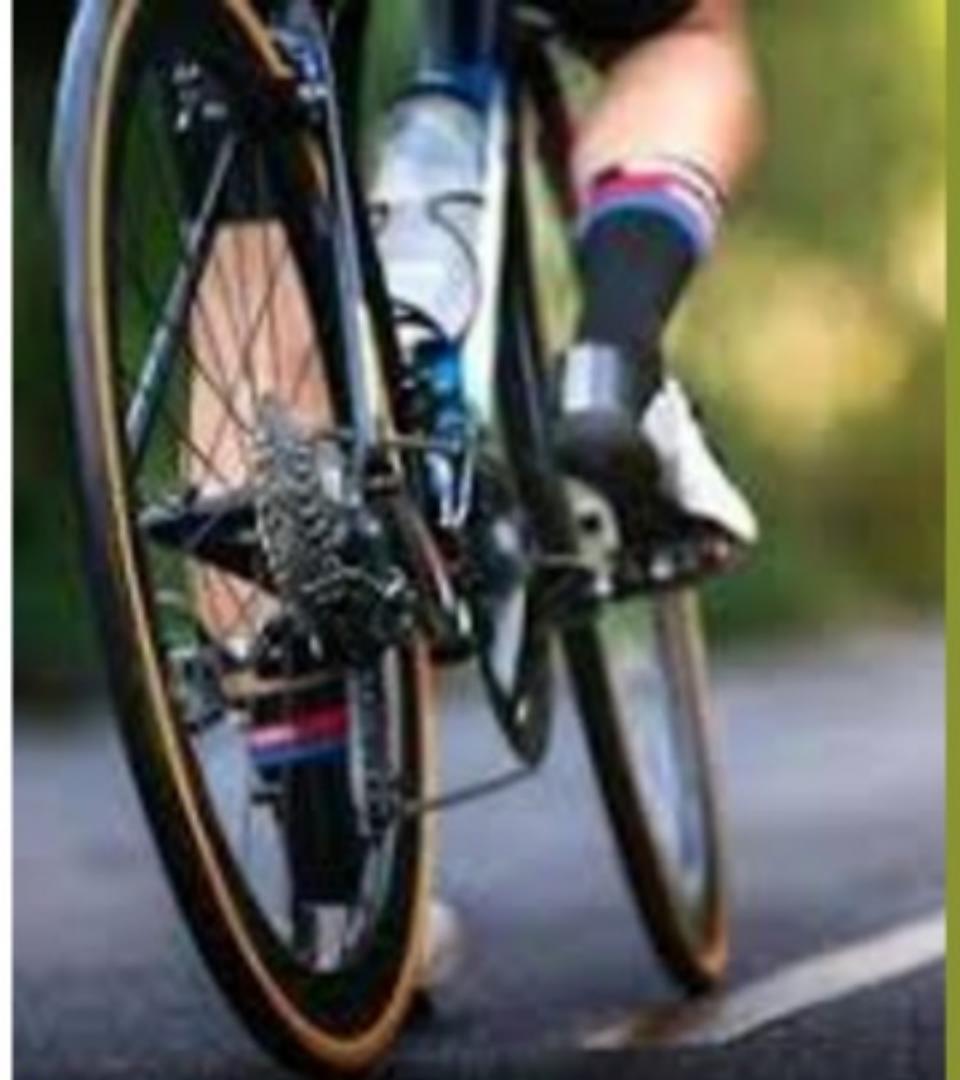


11. List the names of staff members who have made more sales than the average number of sales by all staff members.

```
with a as ( select staffs.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 a where total_sales > (select avg(total_sales) from a );
```

Output

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).

```
SELECT c.customer_id, c.first_name, c.last_name
FROM customers c JOIN orders o ON o.customer_id = c.customer_id
JOIN order_items ot ON ot.order_id = o.order_id JOIN products p ON p.product_id = ot.product_id
GROUP BY c.customer_id, c.first_name, c.last_name
HAVING COUNT(DISTINCT category_id) =
    (SELECT COUNT(*) FROM categories
);
```



Output

customer_id	first_name	last_name
9	Genoveva	Baldwin

# Thank you

Presented by :- Vikram Jakhar