

# JENSON



# USA

## SQL

## Analysis

Presented by :  
Gowri Abbigeri



# About Company

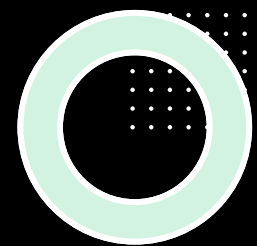
**Jenson USA** began in 1994 with a simple idea: to bring quality bikes and cycling gear to riders who share the joy of being on the trail, the road, or anywhere a bike can take them.

Over the years, the company has grown into a well-known cycling retailer with a wide online store and welcoming retail locations. Their team is made up of passionate riders who understand what cyclists need — because they ride too.

Whether someone is gearing up for their first ride or fine-tuning a race bike, Jenson USA delivers expert guidance, top-brand products, and a supportive community built on the love of riding.



# OUR OBJECTIVE



- **ANALYZE CUSTOMER BEHAVIOR**

- Understand purchasing patterns, customer preferences, and spending habits to identify key customer segments and improve personalization strategies.

- **EVALUATE STAFF PERFORMANCE**

- Measure sales contribution by individual staff members, identify top performers, and opportunities for training or performance improvement.

- **OPTIMIZE INVENTORY MANAGEMENT**

- Track product sales over time, identify top-selling and unsold items, and support data-driven inventory restocking and clearance decisions

- **ENHANCE STORE OPERATIONS**

- Compare performance across different store locations by analyzing order volume, product movement, and customer engagement to support operational efficiency.

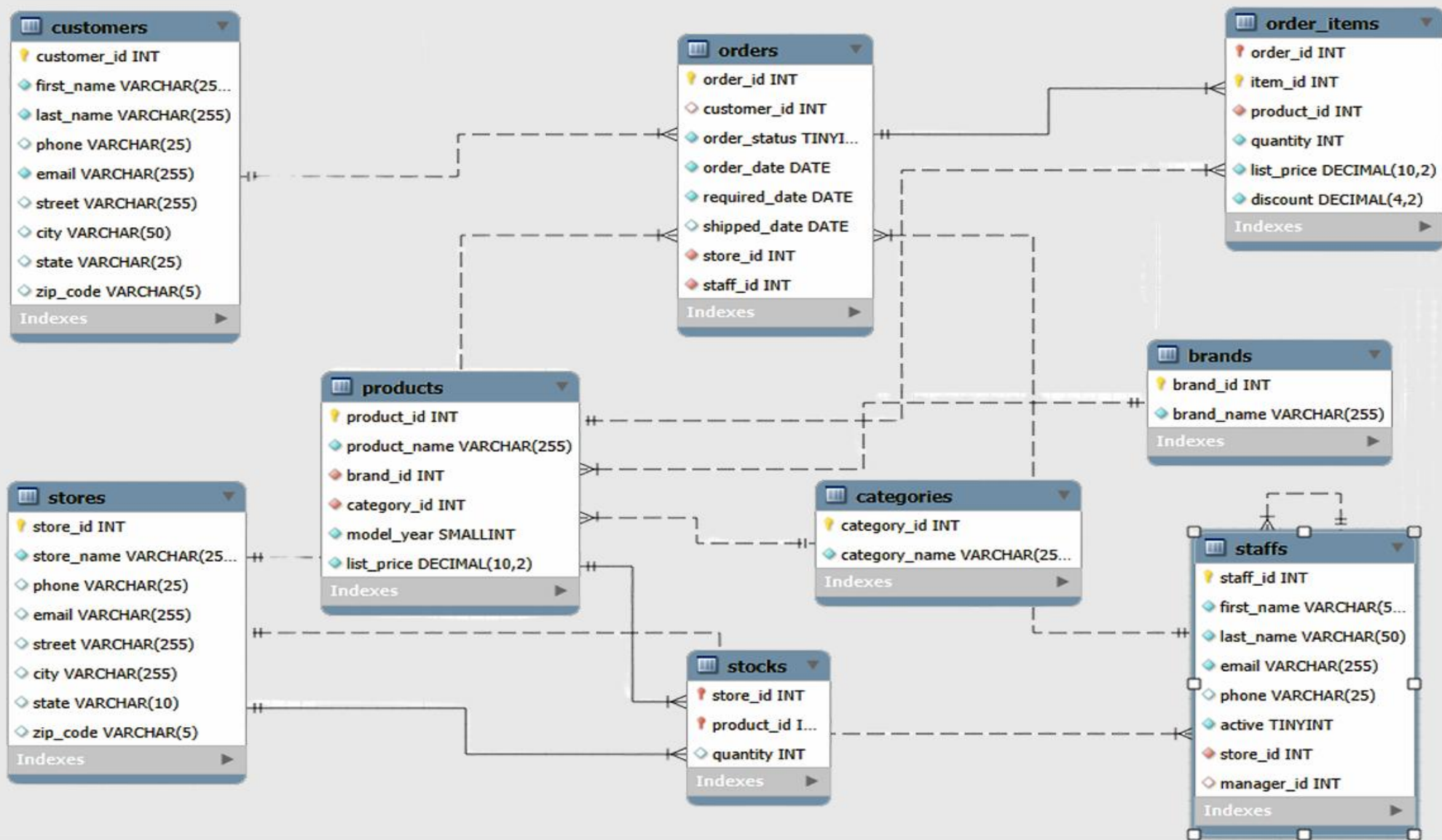
- **IDENTIFY HIGH-VALUE PRODUCTS AND CATEGORIES**

Determine products and categories contributing most to revenue to inform promotional strategies and inventory focus.

- **SUPPORT STRATEGIC DECISION MAKING**

- Provide actionable insights to management for improving sales, customer satisfaction, and overall business performance using reliable, data-driven analysis

# Relational Schema Diagram





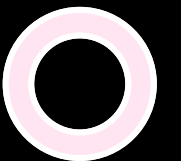
# AD-HOC Analysis

Sales Performance

Customer Insights

Product & Inventory

Staff & Store Efficiency



# Sales Performance

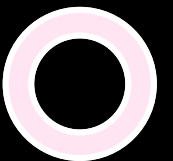
## Regions are most profitable per order

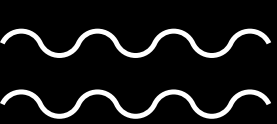
```
SELECT s.state,  
Round((ROUND(SUM(oi.quantity*(oi.list_price*(1 -  
oi.discount))),2) ) / count(distinct oi.order_id) ,2) as  
avg_revenue_per_order  
FROM jenkins.stores s  
join orders o  
on o.store_id=s.store_id  
join order_items oi  
on oi.order_id=o.order_id  
group by state  
order by avg_revenue_per_order desc
```

## Output

	state	avg_revenue_per_order
▶	TX	4985.87
	NY	4771.96
	CA	4614.43

Actionable Suggestions:  
Increase marketing efforts, open new stores, or expand distribution in top-performing states to maximize ROI.





# Sales Performance

## Brands drive our revenue

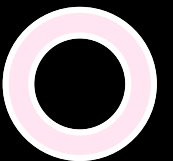
```
SELECT
b.brand_id,
b.brand_name,
Round(sum((oi.list_price*(1-oi.discount))*oi.quantity),0) as revenue
FROM
order_items oi
join products p
on oi.product_id=p.product_id
join
brands b on p.brand_id=b.brand_id
group by brand_name
order by revenue desc;
```

## Output

	brand_id	brand_name	revenue
►	9	Trek	4602754
	1	Electra	1205321
	8	Surly	949507
	7	Sun Bicycles	341995
	2	Haro	185385
	3	Heller	171459
	4	Pure Cycles	149476
	5	Ritchey	78899
	6	Strider	4320

## Actionable Suggestions:

Give top brands better shelf placement, run co-branded promotions, and negotiate strategic partnerships.



# Sales Performance

## Best sellers

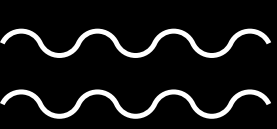
```
SELECT p.product_name, sum(oi.quantity) as
total_order,
Round(sum((oi.list_price*(1-
oi.discount))*oi.quantity),0) as revenue
FROM jenkins.order_items oi
join products p on p.product_id=oi.product_id
group by product_name
order by revenue desc;
```

## Output

	product_name	total_order	revenue
►	Trek Slash 8 275 - 2016	154	555559
	Trek Conduit+ - 2016	145	389249
	Trek Fuel EX 8 29 - 2016	143	368473
	Surly Straggler 650b - 2016	151	226766
	Trek Domane SLR 6 Disc - 2017	43	211585
	Surly Straggler - 2016	147	203508
	Trek Remedy 29 Carbon Frameset - 2016	125	203381
	Trek Powerfly 8 FS Plus - 2017	41	188250
	Trek Madone 92 - 2017	39	175900
	Trek Silque SLR 8 Women's - 2017	29	174525
	Trek Silque SLR 7 Women's - 2017	30	163080
	Trek Fuel EX 98 275 Plus - 2017	33	159053
	Electra Townie Original 7D EQ - 2016	290	155169
	Heller Shagamaw Frame - 2016	129	151161
	Trek Fuel EX 98 29 - 2017	33	147450
	Electra Townie Original 21D - 2016	289	143393
	Trek Boone 7 - 2017	42	127610
	Trek Domane SL 6 - 2017	37	113645
	Surly Wednesday Frameset - 2016	126	112289
	Trek Remedy 98 - 2017	22	105470
	Trek Domane S 6 - 2017	39	93528
	Trek Boone Race Shop Limited - 2017	29	91245
	Surly Karate Monkey 275+ Frameset - 2017	39	87475

## Actionable Suggestions:

Feature high-revenue items in premium displays; bundle high-quantity items with accessories to increase upsell potential.



# Sales Performance

Month which had the **highest revenue** overall

```
SELECT o.order_id,  
date_format(o.order_date, '%Y-%m') AS month,  
Round(sum((oi.list_price*(1-oi.discount))*oi.quantity),0) as  
revenue  
FROM jenkins.orders o  
join order_items oi on oi.order_id=o.order_id  
group by month  
order by revenue desc  
limit 1;
```

## Output

	order_id	month	revenue
▶	1479	2018-04	817922

Actionable Suggestions:

Replicate successful strategies from that month (discounts, campaigns, product launches).



# Customer Insights

**Average order value** is higher than the **overall store average order value**.

```
WITH order_totals AS (  
    SELECT o.order_id,  
           SUM(oi.quantity * oi.list_price) AS order_value  
    FROM orders o  
    JOIN order_items oi ON o.order_id = oi.order_id  
    GROUP BY o.order_id),  
customer_avg AS (  
    SELECT  
        o.customer_id,  
        AVG(ot.order_value) AS avg_order_value  
    FROM order_totals ot  
    JOIN orders o ON ot.order_id = o.order_id  
    GROUP BY o.customer_id),  
store_avg AS (  
    SELECT AVG(order_value) AS store_avg_value  
    FROM order_totals)
```

```
SELECT  
    c.customer_id,  
    CONCAT(c.first_name, ' ', c.last_name) AS  
customer_name,  
    ca.avg_order_value,s.store_avg_value  
FROM customer_avg ca  
JOIN customers c ON ca.customer_id =  
c.customer_id  
JOIN store_avg s ON 1=1  
join orders o on o.customer_id=c.customer_id  
WHERE ca.avg_order_value >  
s.store_avg_value  
ORDER BY ca.avg_order_value DESC;
```

## Output

	customer_id	customer_name	avg_order_value
	122	Shena Carter	27618.95
	1224	Abram Copeland	26913.92
	1023	Adena Blake	22999.94
	1214	Brigid Sharp	22299.95
	227	Danielle Bond	22000.920000000002
	238	Cindi Larson	21816.93
	425	Augustina Joyner	21789.92
	464	Bess McBride	21097.949999999997
	1037	Ashanti Hammond	20997.94
	1005	Ruthanne Franco	20679.95
	1100	Penny Acevedo	20399.92
	339	Tamela Harrell	20199.949999999997
	525	Andreas Mayer	19699.92
	289	Charmain Webster	19499.96
	1007	Jimmy Russell	19419.94
	1421	Edris Barrett	19399.929999999997
	182	Jenniffer Bullock	19361.929999999997
	108	Shae Hickman	19263.949999999997
	607	Paul Lester	19081.91

Actionable Suggestions:  
Create a VIP / premium loyalty program with early access,  
exclusive bundles, or personalized offers.

# Customer Insights

Most valuable product category (highest revenue contributor) for each customer.

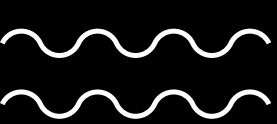
```
WITH category_revenue AS (  
  SELECT  
    c.customer_id,  
    ca.category_name,  
    SUM(oi.quantity * oi.list_price) AS revenue  
  FROM customers c  
  JOIN orders o ON c.customer_id = o.customer_id  
  JOIN order_items oi ON o.order_id = oi.order_id  
  JOIN products p ON p.product_id = oi.product_id  
  join categories ca on p.category_id=ca.category_id  
  GROUP BY c.customer_id, ca.category_name  
)  
,  
ranked AS (  
  SELECT *,  
    RANK() OVER (PARTITION BY customer_id ORDER  
  BY revenue DESC) AS rnk  
  FROM category_revenue  
)  
SELECT  
  customer_id,  
  category_name,  
  revenue AS max_revenue  
FROM ranked  
WHERE rnk = 1;
```

## Output

	customer_id	category_name	max_revenue
▶	1	Road Bikes	14699.97
	2	Road Bikes	13599.97
	3	Road Bikes	16999.97
	4	Mountain Bikes	8939.96
	5	Electric Bikes	9999.97
	6	Mountain Bikes	16819.949999999997
	7	Cydocross Bicycles	5999.98
	8	Comfort Bicycles	2333.96
	9	Road Bikes	12999.98
	10	Road Bikes	23999.98
	11	Road Bikes	2599.99
	12	Electric Bikes	16999.96
	13	Mountain Bikes	9089.949999999999
	14	Electric Bikes	8999.97
	15	Road Bikes	6399.98
	16	Mountain Bikes	18998.96
	17	Cruisers Bicycles	3757.96
	18	Road Bikes	6399.98
	19	Road Bikes	13599.97
	20	Cydocross Bicycles	6459.98
	21	Road Bikes	7699.969999999999
	22	Mountain Bikes	10599.98

Actionable Suggestions:

Run personalized category-based recommendations and cross-sell related products.



# Customer Insights

Customers Who Bought From ALL Categories

```
WITH cust_cat AS (  
    SELECT DISTINCT  
        c.customer_id,  
        CONCAT(c.first_name, '_', c.last_name) AS  
customer_name,  
        p.category_id  
    FROM customers c  
    JOIN orders o ON c.customer_id = o.customer_id  
    JOIN order_items oi ON o.order_id = oi.order_id  
    JOIN products p ON oi.product_id = p.product_id  
)  
category_count AS (  
    SELECT  
        customer_id,  
        customer_name,  
        COUNT(DISTINCT category_id) AS category_count  
    FROM cust_cat  
    GROUP BY customer_id, customer_name  
)  
total_cat AS (  
    SELECT COUNT(*) AS total_categories FROM categories  
)
```

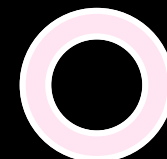
```
SELECT  
    cc.customer_id,  
    cc.customer_name,  
    cc.category_count  
FROM category_count cc  
CROSS JOIN total_cat tc  
WHERE cc.category_count =  
tc.total_categories;
```

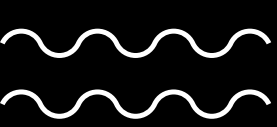
## Output

	customer_id	customer_name	category_count
▶	9	Genoveva_Baldwin	7

## Actionable Suggestions:

Treat them as brand ambassadors with referral bonuses, exclusive offers, and product trials.





# Customer Insights

## Customer concentration based on states

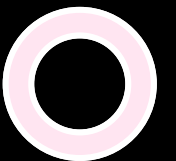
```
SELECT  
state,  
count(customer_id) as no_of_people  
FROM  
customers  
group by state  
order by no_of_people desc;
```

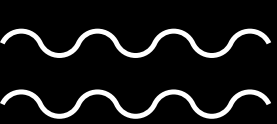
## Output

	state	no_of_people
▶	NY	1019
	CA	284
	TX	142

### Actionable Suggestions:

Prioritize store openings, localized ads, and regional campaigns in top 3–5 states.





# Customer Insights

Customers who have spent more than \$30000 in total.

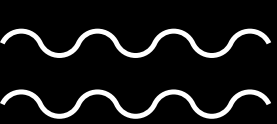
## Output

```
SELECT c.customer_id,concat(c.first_name,' ',last_name) as
customer_name,
Round(sum((oi.list_price*oi.quantity*(1-oi.discount))),2) as revenue
FROM orders o
Join customers c
on o.customer_id=c.customer_id
Join order_items oi
on oi.order_id=o.order_id
group by c.customer_id
having revenue>30000
order by revenue desc;
```

	customer_id	customer_name	revenue
▶	94	Sharyn Hopkins	34807.94
	10	Pamelia Newman	33634.26
	75	Abby Gamble	32803.01
	6	Lyndsey Bean	32675.07
	16	Emmitt Sanchez	31925.89
	73	Melanie Hayes	31913.69

Actionable Suggestions:  
Offer premium tiers, priority delivery, and dedicated support to retain them.





# Customer Insights

Customers who are frequent vs infrequent, each group contributes to revenue and their average order and spend behaviour

```
WITH customer_freq AS (  
  SELECT  
    c.customer_id,  
    COUNT(o.order_id) AS total_orders,  
    SUM(oi.list_price * oi.quantity * (1 - oi.discount)) AS  
total_spent,  
    CASE  
      WHEN COUNT(o.order_id) < 3  
        OR SUM(oi.list_price * oi.quantity * (1 -  
oi.discount)) < 4999  
        THEN 'infrequent buyers'  
      ELSE 'frequent buyers'  
    END AS cust_cat  
  FROM customers c  
  JOIN orders o  
    ON c.customer_id = o.customer_id  
  JOIN order_items oi  
    ON o.order_id = oi.order_id  
  GROUP BY c.customer_id  
)  
FROM customer_freq  
GROUP BY cust_cat  
ORDER BY total_revenue DESC;
```

```
SELECT  
  cust_cat,  
  COUNT(customer_id) AS  
total_customers,  
  ROUND(AVG(total_orders), 2) AS  
avg_orders_per_customer,  
  ROUND(AVG(total_spent), 2) AS  
avg_spent_per_customer,  
  ROUND(SUM(total_spent), 2) AS  
total_revenue
```

## Output

	cust_cat	total_customers	avg_orders_per_customer	avg_spent_per_customer	total_revenue
►	frequent buyers	503	4.76	10331.36	5196676.34
	infrequent buyers	942	2.47	2645.9	2492440.22

Actionable Suggestions:

Frequent: give loyalty rewards. Infrequent: run reactivation campaigns and limited-time offers.



# Customer Insights

Purchasing behaviour of customers from different regions

```
SELECT COUNT(distinct c.customer_id) AS  
total_customers, c.city,  
c.state, p.product_name, st.store_name  
FROM customers c  
JOIN orders o  
ON c.customer_id = o.customer_id  
JOIN stores st  
ON o.store_id = st.store_id  
JOIN order_items oi  
ON o.order_id = oi.order_id  
JOIN products p  
ON oi.product_id = p.product_id  
GROUP BY c.city  
ORDER BY total_customers desc;
```

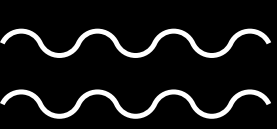
## Output

	total_customers	city	state	product_name	store_name
▶	20	Mount Vernon	NY	Electra Girl's Hawaii 1 16" - 2017	Baldwin Bikes
	17	Ballston Spa	NY	Trek Precaliber 24 (21-Speed) - Girls - 2017	Baldwin Bikes
	17	Scarsdale	NY	Surly Steamroller - 2017	Baldwin Bikes
	14	Canandaigua	NY	Surly Straggler - 2016	Baldwin Bikes
	13	Floral Park	NY	Electra Townie Original 7D EQ - 2016	Baldwin Bikes
	13	Longview	TX	Electra Townie Original 7D EQ - 2016	Rowlett Bikes
	13	Ossining	NY	Trek CrossRip+ - 2018	Baldwin Bikes
	12	Astoria	NY	Trek Powerfly 7 FS - 2018	Baldwin Bikes
	12	Canyon Country	CA	Electra Townie Original 21D - 2016	Santa Cruz Bikes
	12	Howard Beach	NY	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	Baldwin Bikes
	12	Merrick	NY	Trek Fuel EX 8 29 - 2016	Baldwin Bikes
	12	Richmond Hill	NY	Sun Bicycles Atlas X-Type - 2017	Baldwin Bikes
	12	Ronkonkoma	NY	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	Baldwin Bikes
	12	San Angelo	TX	Sun Bicycles Streamway 3 - 2017	Rowlett Bikes
	12	Smithtown	NY	Electra Moto 3i (20-inch) - Boy's - 2017	Baldwin Bikes

Actionable Suggestions:

Launch region-specific marketing and optimize inventory for top-selling items in each region





# Product & Inventory

Top 5 products with the highest revenue.

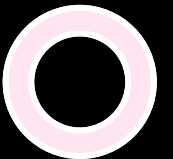
```
SELECT jp.product_name,  
Round(sum((ot.list_price*ot.quantity*(1-ot.discount))),2) as  
revenue  
FROM jenkins.products jp  
Join order_items ot  
on jp.product_id=ot.product_id  
group by jp.product_id  
order by revenue desc  
limit 5;
```

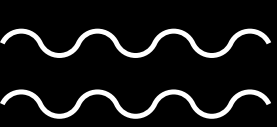
## Output

	product_name	revenue
▶	Trek Slash 8 275 - 2016	555558.61
	Trek Conduit+ - 2016	389248.7
	Trek Fuel EX 8 29 - 2016	368472.73
	Surly Straggler 650b - 2016	226765.55
	Trek Domane SLR 6 Disc - 2017	211584.62

Actionable Suggestions:

Highlight these products on store displays, online banners,  
and promote them through targeted digital campaigns.





# Product & Inventory

Second highest revenue product.

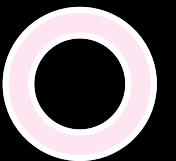
```
select product_name, revenue from (  
  
SELECT p.product_id,product_name,  
Round(sum((oi.list_price*oi.quantity*(1-oi.discount))),2) as revenue ,  
rank() over( order by Round(sum((oi.list_price*oi.quantity*(1-oi.discount))),2)  
desc ) as rak  
  
FROM jenkins.products p  
join order_items oi on p.product_id=oi.product_id  
group by product_name) as t  
  
where rak=2;
```

## Output

	product_name	revenue
►	Trek Conduit+ - 2016	389248.7

## Actionable Suggestions:

Use dynamic ranking in dashboards to monitor performance and adjust pricing or discounts for top-ranked items.



# Product & Inventory

Top 3 best-selling products in terms of quantity.

```
SELECT p.product_id,  
       product_name,  
       sum(oi.quantity) as total_order  
FROM jenkins.products p  
join order_items oi  
on p.product_id=oi.product_id  
group by p.product_name  
order by total_order desc  
limit 3;
```

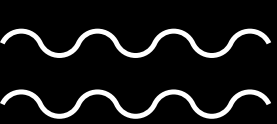
## Output

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

Actionable Suggestions:

Bundle high-quantity items with accessories to increase average order value (AOV).





# Product & Inventory

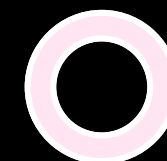
The products that have yet to be sold.

```
SELECT
p.product_name,
p.product_id
FROM products p
where not exists(SELECT * FROM order_items oi
where p.product_id=oi.product_id)
```

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

Actionable Suggestions:

Review product placement, revise pricing, promote through limited-time offers, or consider discontinuation.



# Product & Inventory

Total revenue generated from each product category

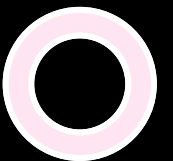
```
with cte1 as
(SELECT p.product_id,c.category_name as category_name,
Round(((oi.list_price*(1-oi.discount))*oi.quantity),2) as revenue
from order_items oi
join products p
on p.product_id =oi.product_id
join categories c
on c.category_id=p.category_id)
```

```
select category_name, round(sum(revenue),2) as total_revenue
FROM cte1
group by category_name
order by total_revenue desc
```

## Output

	category_name	total_revenue
▶	Mountain Bikes	2715077.72
	Road Bikes	1665097.93
	Cruisers Bicycles	995030.86
	Electric Bikes	916684.45
	Cyclocross Bicycles	711011.57
	Comfort Bicycles	394019.27
	Children Bicycles	292187.91

Actionable Suggestions:  
Increase inventory for top categories, prioritize supply chain flow, and plan category-specific campaigns.



# Product & Inventory

Products with low stock (quantity below a certain threshold). **Output**

```
SELECT
st.store_id,
s.store_name,
p.product_id,
p.product_name,
st.quantity
FROM stocks st
join stores s
on st.store_id=s.store_id
join products p
on p.product_id=st.product_id
where quantity < 1
```

	store_id	store_name	product_id	product_name	quantity
▶	1	Santa Cruz Bikes	6	Surly Ice Cream Truck Frameset - 2016	0
	1	Santa Cruz Bikes	8	Trek Remedy 29 Carbon Frameset - 2016	0
	1	Santa Cruz Bikes	32	Trek Farley Alloy Frameset - 2017	0
	1	Santa Cruz Bikes	42	Trek Fuel EX 5 275 Plus - 2017	0
	1	Santa Cruz Bikes	92	Haro Shredder 20 - 2017	0
	1	Santa Cruz Bikes	160	Trek Emonda SLR 6 - 2018	0
	1	Santa Cruz Bikes	163	Surly Pack Rat - 2018	0
	1	Santa Cruz Bikes	168	Surly Straggler - 2018	0
	1	Santa Cruz Bikes	246	Electra Townie Original 3i EQ Ladies' - 2018	0
	1	Santa Cruz Bikes	302	Electra Townie Original 1 - 2018	0
	2	Baldwin Bikes	22	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	0
	2	Baldwin Bikes	47	Trek Remedy 98 - 2017	0
	2	Baldwin Bikes	91	Trek Precaliber 24 (21-Speed) - Girls - 2017	0
	2	Baldwin Bikes	158	Trek CrossRip 1 - 2018	0
	2	Baldwin Bikes	175	Trek Domane SLR Frameset - 2018	0
	2	Baldwin Bikes	184	Trek Domane SL 6 Disc - 2018	0
	2	Baldwin Bikes	192	Electra Townie Go! 8i - 2017/2018	0
	2	Baldwin Bikes	198	Electra Townie Commute Go! - 2018	0
	2	Baldwin Bikes	251	Electra Townie Commute Go! - 2018	0
	2	Baldwin Bikes	299	Electra Townie Original 21D - 2018	0
	3	Rowlett Bikes	3	Surly Wednesday Frameset - 2016	0
	3	Rowlett Bikes	59	Trek Domane S 5 Disc - 2017	0
	3	Rowlett Bikes	220	Electra Cruiser 1 Ladies' - 2018	0
	3	Rowlett Bikes	259	Electra Amsterdam Royal 8i Ladies - 2018	0
	3	Rowlett Bikes	313	Electra Townie Original 1 Ladies' - 2018	0

Actionable Suggestions:  
Trigger automated stock alerts, reorder immediately, and  
optimize safety stock levels.

# Product & Inventory

The average time between order placement and shipment for each product.

with avg\_shipping as

```
(SELECT
```

```
p.product_id,p.product_name,o.order_date,o.shipped_d
```

```
ate,
```

```
datediff(o.shipped_date,o.order_date) as avg_days
```

```
from products p
```

```
join order_items oi
```

```
on oi.product_id=p.product_id
```

```
join orders o
```

```
on o.order_id=oi.order_id)
```

```
select * from avg_shipping
```

```
group by product_id
```

```
order by avg_days desc
```

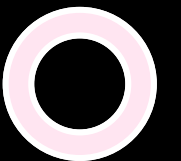
```
limit 10;
```

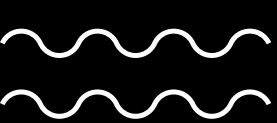
## Output

	product_id	product_name	order_date	shipped_date	avg_days
▶	6	Surly Ice Cream Truck Frameset - 2016	2016-01-09	2016-01-12	3
	97	Electra Savannah 3i (20-inch) - Girl's - 2017	2017-01-10	2017-01-13	3
	23	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	2016-01-09	2016-01-12	3
	21	Electra Cruiser 1 (24-Inch) - 2016	2016-01-12	2016-01-15	3
	13	Electra Cruiser 1 (24-Inch) - 2016	2016-01-08	2016-01-11	3
	68	Sun Bicycles Cruz 3 - 2017	2017-01-07	2017-01-10	3
	17	Pure Cycles Vine 8-Speed - 2016	2016-01-03	2016-01-06	3
	26	Electra Townie Original 7D EQ - 2016	2016-01-03	2016-01-06	3
	11	Surly Straggler 650b - 2016	2016-01-06	2016-01-09	3
	7	Trek Slash 8 275 - 2016	2016-01-05	2016-01-08	3

Actionable Suggestions:

Coordinate with suppliers or warehouses to reduce delays;  
consider priority shipping for slow-dispatch items.





# Product & Inventory

**Most popular product category** each month. (Use Window Function)

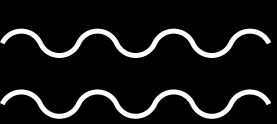
```
select month,category_name,total_quantity from (  
SELECT  
date_format(o.order_date, '%Y-%m') AS month,  
sum(oi.quantity) as total_quantity,  
c.category_name,  
rank() over(partition by date_format(o.order_date, '%Y-%m')  
order by sum(oi.quantity) desc)as rak  
FROM jenkins.orders o  
join order_items oi on oi.order_id=o.order_id  
join products p on p.product_id=oi.product_id  
join categories c on c.category_id=p.category_id  
group by month) as t  
where rak=1
```

Actionable Suggestions:

Use month-wise insights to forecast inventory, plan seasonal promotions, and align stock before demand peaks.

## Output

	month	category_name	total_quantity
▶	2016-01	Cruisers Bicycles	221
	2016-02	Cruisers Bicycles	223
	2016-03	Cruisers Bicycles	213
	2016-04	Mountain Bikes	176
	2016-05	Mountain Bikes	224
	2016-06	Mountain Bikes	199
	2016-07	Cruisers Bicycles	211
	2016-08	Children Bicycles	251
	2016-09	Cruisers Bicycles	281
	2016-10	Cruisers Bicycles	254
	2016-11	Mountain Bikes	181
	2016-12	Mountain Bikes	229
	2017-01	Children Bicycles	229
	2017-02	Cruisers Bicycles	263
	2017-03	Cruisers Bicycles	296
	2017-04	Comfort Bicycles	248
	2017-05	Children Bicycles	241
	2017-06	Children Bicycles	296
	2017-07	Road Bikes	249
	2017-08	Cruisers Bicycles	287
	2017-09	Cruisers Bicycles	237
	2017-10	Cruisers Bicycles	296
	2017-11	Children Bicycles	241



# Staff & Store Efficiency

Staff performance across each store based on orders

```
SELECT
  st.store_name,
  sf.first_name || ' ' || sf.last_name AS full_name,
  DENSE_RANK() OVER (
    PARTITION BY st.store_name
    ORDER BY COUNT(*) DESC
  ) AS rnk
FROM orders AS o
JOIN staffs AS sf ON sf.staff_id = o.staff_id
JOIN stores AS st ON st.store_id = o.store_id
GROUP BY st.store_name, sf.staff_id
ORDER BY st.store_name, rnk
```

## Output

	store_name	full_name	rnk
►	Baldwin Bikes	Marcelene Boyer	1
	Baldwin Bikes	Venita Daniel	2
	Rowlett Bikes	Kali Vargas	1
	Rowlett Bikes	Layla Terrell	2
	Santa Cruz Bikes	Genna Serrano	1
	Santa Cruz Bikes	Mireya Copeland	2

## Actionable Suggestions:

Recognize high performers, assign them high-impact sales tasks, and use them as mentors to train low-performing staff.



# Staff & Store Efficiency

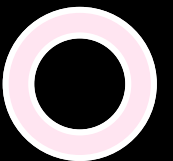
Staff Members Who Have Not Made Any Sales

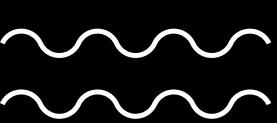
```
SELECT sf.staff_id,  
concat(sf.first_name, ' ', sf.last_name) AS full_name  
FROM jenkins.staffs sf  
left join orders o  
on o.staff_id=sf.staff_id  
where o.order_id is null;
```

## Output

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

Actionable Suggestions:  
Provide additional sales training, review role suitability, or  
pair them with strong performers for guided improvement.





# Staff & Store Efficiency

Staff with Sales Above Average

```
WITH cte1 AS (  
    SELECT  
        st.staff_id,  
        CONCAT(st.first_name, ' ', st.last_name) AS full_name,  
        ROUND(SUM((oi.list_price * (1 - oi.discount)) * oi.quantity),  
0) AS revenue  
    FROM jenkins.order_items oi  
    JOIN orders o ON o.order_id = oi.order_id  
    JOIN staffs st ON o.staff_id = st.staff_id  
    JOIN stores s ON o.store_id = s.store_id  
    GROUP BY st.staff_id, st.store_id  
)  
,  
avg_total AS (  
    SELECT AVG(revenue) AS avg_revenue FROM cte1  
)  
SELECT  
    c.staff_id,  
    c.full_name,  
    c.revenue,  
    ROUND(av.avg_revenue, 0) AS average_revenue  
FROM cte1 c  
CROSS JOIN avg_total av  
WHERE c.revenue > av.avg_revenue  
ORDER BY c.revenue DESC;
```

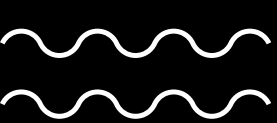
## Output

	staff_id	full_name	revenue	average_revenue
▶	6	Marcelene Boyer	2624121	1281520
	7	Venita Daniel	2591631	1281520

## Actionable Suggestions:

Reward high performers, and analyze their selling methods to replicate successful strategies across the team.





# Staff & Store Efficiency

Customers are new and how many are returning via a subquery

```
SELECT
  store_name,
  COUNT(CASE WHEN status = 'returning' THEN 1 END) AS
returning_customers,
  COUNT(CASE WHEN status = 'new' THEN 1 END) AS
new_customers,
  concat(ROUND(
    100.0 * (COUNT(CASE WHEN status = 'returning' THEN 1 END) )
    / COUNT(*),
    2
  ),'%') AS returning_percentage
FROM (
  SELECT
    cu.customer_id,
    s.store_name,
    CASE
      WHEN COUNT( o.order_id) > 1 THEN 'returning'
      ELSE 'new'
    END AS status
  FROM customers cu
  JOIN orders o ON o.customer_id = cu.customer_id
  JOIN stores s ON s.store_id = o.store_id
  GROUP BY
    cu.customer_id,store_name
) as customer_status_by_store
```

```
GROUP BY
  store_name
ORDER BY
  ROUND(
    100.0 * COUNT(CASE WHEN status = 'returning' THEN 1 END)
    / COUNT(*),
    2
  ) DESC
```

## Output

	store_name	returning_customers	new_customers	returning_percentage
▶	Santa Cruz Bikes	52	232	18.31%
	Rowlett Bikes	19	123	13.38%
	Baldwin Bikes	60	959	5.89%

## Actionable Suggestions:

Strengthen loyalty programs in low-retention stores; create personalized follow-ups and exclusive offers for repeat customers.



# Staff & Store Efficiency

Number of stocks based on category name and store name

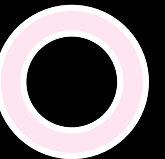
```
Select st.store_name, c.category_name,  
SUM(s.quantity) as sum_qty  
from categories c JOIN products p  
ON c.category_id = p.category_id  
JOIN stocks s  
ON p.product_id = s.product_id  
JOIN stores st ON  
st.store_id = s.store_id  
GROUP BY st.store_name, c.category_name  
ORDER BY st.store_name, c.category_name;
```

## Output

	store_name	category_name	sum_qty
►	Baldwin Bikes	Children Bicycles	769
	Baldwin Bikes	Comfort Bicycles	393
	Baldwin Bikes	Cruisers Bicycles	1137
	Baldwin Bikes	Cydocross Bicycles	97
	Baldwin Bikes	Electric Bikes	328
	Baldwin Bikes	Mountain Bikes	877
	Baldwin Bikes	Road Bikes	758
	Rowlett Bikes	Children Bicycles	887
	Rowlett Bikes	Comfort Bicycles	440
	Rowlett Bikes	Cruisers Bicycles	1148
	Rowlett Bikes	Cydocross Bicycles	159
	Rowlett Bikes	Electric Bikes	412
	Rowlett Bikes	Mountain Bikes	928
	Rowlett Bikes	Road Bikes	646
	Santa Cruz ...	Children Bicycles	952
	Santa Cruz ...	Comfort Bicycles	425
	Santa Cruz ...	Cruisers Bicycles	1093
	Santa Cruz ...	Cydocross Bicycles	158
	Santa Cruz ...	Electric Bikes	368
	Santa Cruz ...	Mountain Bikes	849
	Santa Cruz ...	Road Bikes	687

Actionable Suggestions:

Balance inventory based on store demand trends: reduce overstock, increase quantities for high-demand categories.



# THANK YOU

PRESENTED BY:  
Gowri Abbigeri

Thanks for taking the time to review this work. The findings support better decisions across customers, staff, inventory, and store operations.

