

We Keep You Pedaling

JENSON USA Performance Analysis and driven Insights using Advance SQL

Presented By:- SHARAD MITTAL



Blog

Products

About us

contact

Login

Project: Jenson USA

We Keep You Pedaling



Customer Behavior

Staff Performance

Inventory Management

Store Operations

Business Insights



Objective

To leverage SQL queries for generating actionable insights into customer behavior, staff performance, inventory management, and store operations, enabling data-driven decision-making that improves business efficiency, customer satisfaction, and overall profitability.

Use Advance SQL to analyze business data for better decisions.



Aim

- Customer Behavior → Understand buying patterns, spending habits, and customer engagement to enhance marketing and retention.
- Staff Performance → Measure sales contributions, identify high/low performers, and optimize workforce productivity.
- Inventory Management → Track product demand, stock movement, and pricing trends to reduce overstock/stockouts.
- Store Operations → Monitor store-wise sales, order volume, and revenue distribution for operational improvements.

Get detailed insights into customers, staff, inventory, and stores to maximize growth & efficiency





Tools Used

MySQL With MySQL Workbench **Jenson USA Dataset** (SQL Text File Format)







Total
Products
Sold by Store

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

```
SELECT

stores.store_name, SUM(order_items.quantity) products_sold

FROM

stores

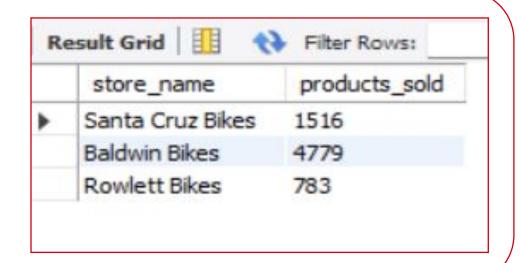
LEFT JOIN

orders USING (store_id)

LEFT JOIN

order_items USING (order_id)

GROUP BY stores.store_name;
```



Objective

To analyze **store-level sales performance** by identifying how many products each store sells. This helps compare stores, measure efficiency, and find top/underperforming stores.



Cumulative Product Sales Over Time

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) total_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(total_quantity)
over(partition by product_id order by order_date)
cum_quantity from a;
```

	product_id	product_name	order_date	total_quantity	cum_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

Objective

To track **sales trends of each product over time**, showing how demand builds up. This helps in identifying **best-selling products**, forecasting inventory needs, and understanding product life cycles



Top-Selling Product by Category (Revenue)

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) as sales
    from products join order_items using(product_id)
    join categories using(category_id) group by
    categories.category_id, categories.category_name,
    products.product_id, products.product_name)
    select * from
    (select *, dense_rank()
    over(partition by category_id order by sales desc)
    rnk from a) b
    where rnk = 1;
Result Grid ## Filter Row

category_id category

category_id category

2    Comfort

3    Crusers

4    Cyclogro

5    Electric B

6    Mountain

7    Road Bik

where rnk = 1;
```

	category_id	category_name	product_id	product_name	sales	rnk
	1	Children Bicydes	23	Electra Girl's Hawaii 1 (20-inch) - 2015/2016	4619846.00	1
	2	Comfort Bicycles	26	Electra Townie Original 7D EQ - 2016	8039866.00	1
	3	Cruisers Bicycles	16	Electra Townie Original 7D EQ - 2016	9359844.00	1
	4	Cyclocross Bicycles	11	Surly Straggler 650b - 2016	25382949.00	1
	5	Electric Bikes	9	Trek Conduit+ - 2016	43499855.00	1
	6	Mountain Bikes	7	Trek Slash 8 275 - 2016	61599846.00	1
	7	Road Bikes	56	Trek Domane SLR 6 Disc - 2017	23649957.00	1

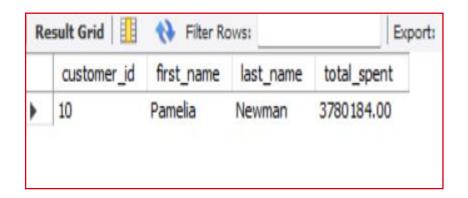
Objective

To identify the **top revenue-generating product in each category**, helping the business understand which items drive the most sales value. This supports **pricing strategies**, **category management**, **and marketing focus**.



Top Spending Customer

Find the customer who spent the most money on orders



Objective

To identify the **highest-spending customer**, enabling the business to focus on **customer loyalty**, **personalized offers**, **and retention strategies** for top-value clients.



Highest-Price d Product by Category

Find the highest-priced product for each category name

```
categories.category_id, categories.category_name,
    max(products.list_price) as highest_price
    from products join categories using (category_id)
    group by
    categories.category_id, categories.category_name
    order by highest_price desc;
```

	category_id	category_name	highest_price	
•	7	Road Bikes	1199999.00	
	6	Mountain Bikes	529999.00	
	5	Electric Bikes	499999.00	
	4	Cydocross Bicydes	399999.00	
	3	Cruisers Bicycles	299999.00	
	2	Comfort Bicycles	259999.00	
	1	Children Bicydes	48999.00	

Objective

To determine the **premium product in each category**, helping in **pricing analysis**, **category positioning**, **and customer purchasing behavior studies**. This also assists in identifying luxury/high-margin items.



Customer Orders per Store

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

```
select customers.customer_id,customers.first_name,
customers.last_name, stores.store_id,
stores.store_name, count(orders.order_id) as total_orders
from orders join customers using(customer_id)
join stores using(store_id)
group by customers.customer_id,customers.first_name,
customers.last_name, stores.store_id,
stores.store_name
order by customers.customer_id, stores.store_id desc;
```

Re	esult Grid	♦ Filter Ro	ows:		Export: Wr	rap Cell Content:
	customer_id	first_name	last_name	store_id	store_name	total_orders
•	1	Debra	Burks	2	Baldwin Bikes	3
	2	Kasha	Todd	1	Santa Cruz Bikes	3
	3	Tameka	Fisher	1	Santa Cruz Bikes	3
	4	Daryl	Spence	2	Baldwin Bikes	3
	5	Charolette	Rice	1	Santa Cruz Bikes	3

Objective

To analyze **customer engagement across different stores** by tracking how many orders each customer places in each location. This helps in **understanding customer loyalty, store preference, and regional demand patterns**.



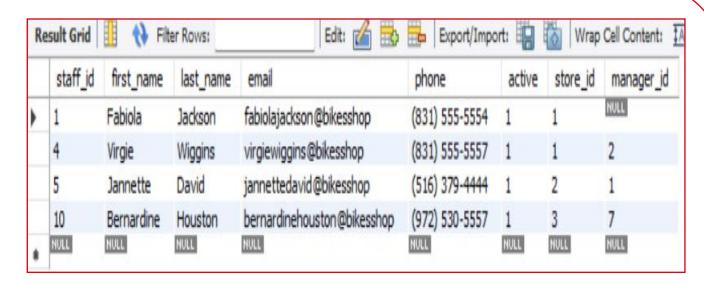
Staff Without Sales

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

```
FROM
staffs
WHERE
NOT EXISTS( SELECT

FROM
orders
WHERE

orders.staff_id = staffs.staff_id);
```



Objective

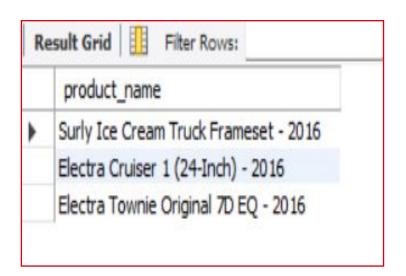
To identify **inactive or underperforming staff members** who haven't generated sales. This insight helps management with **performance evaluation**, **training needs**, **or workforce optimization**.



Top 3 Most Sold Products (by Quantity)

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

```
with a as (select products.product_id, products.product_name
sum(order_items.quantity) as total_quantity
from order_items join products
using(product_id) group by products.product_id,
products.product_name)
select product_name from (select *, rank()
over(order by total_quantity desc)
as rnk from a ) as b
where rnk <=3;</pre>
```



Objective

To determine the **highest-demand products** by sales volume. This helps in **inventory planning**, **promotional strategies**, **and identifying customer preferences**.



Median Product Price

Find the median value of the price list.

```
SELECT

AVG(list_price) AS median_price

FROM ( SELECT products.list_price,

ROW_NUMBER() OVER (ORDER BY products.list_price)

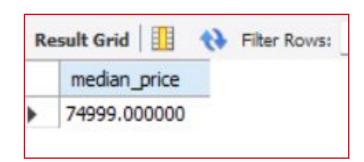
AS row_num, COUNT(*) OVER () AS total_rows

FROM Products ) products

WHERE row_num IN (

(total_rows + 1) / 2,

(total_rows + 2) / 2 );
```



Objective

To calculate the **middle price point** in the product list, which provides a better measure of **central pricing tendency** compared to the average, especially when there are extreme high or low prices. This helps in **pricing analysis and product positioning**.



Products Never Ordered

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
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
	321	Trek Checkpoint ALR Frameset - 2019

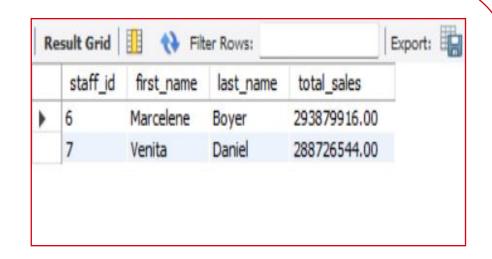
Objective

To identify **unsold or inactive products** that have never been included in any customer order. This helps in **inventory optimization**, **product discontinuation decisions**, **and marketing focus** for low-visibility products.



High-Performin g Staff (Above Average Sales) 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, staffs.first_name, staffs.last_name,
sum(order_items.quantity * (order_items.list_price -
order_items.discount)) as total_sales
from staffs Left join orders using(staff_id)
left join order_items using(order_id)
group by 1,2)
select * from a where total_sales>(select avg(total_sales) from a);
```



Objective

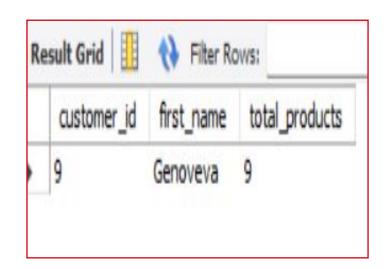
To identify **high-performing staff members** whose sales exceed the company-wide average. This helps in **recognizing top talent**, **planning incentives**, **and setting benchmarks for performance evaluation**.



Customers
Who Ordered
from All
Categories

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

```
select customers.customer_id, customers.first_name,
count(order_items.product_id) total_products
from customers join orders
using(customer_id) join order_items using(order_id)
join products using(product_id)
group by customers.customer_id,
customers.first_name
having count(distinct products.category_id) = (select count(*) from categories);
```



Objective

To find **highly engaged customers** who purchase across **all product categories**. This helps in identifying **loyal**, **diverse buyers** who contribute significantly to overall sales and can be targeted for **premium loyalty programs or cross-selling opportunities**.

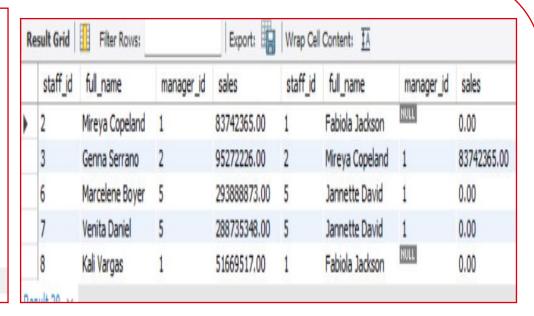


Staff with Higher Sales than Manager

find the staffs who have made sales more than their manager

```
with a as (select staffs.staff_id, concat(staffs.first_name," ",
staffs.last_name) as full_name, staffs.manager_id,
coalesce(sum(order_items.quantity * order_items.list_price),0) sales
from staffs left join orders using(staff_id)
left join order_items using(order_id)
group by staffs.staff_id, concat(staffs.first_name, " ",
staffs.last_name),staffs.manager_id)

select t1.*, t2.* from a as t1 join a as t2
where t1.manager_id = t2.staff_id and
t1.sales> t2.sales;
```



Objective

To identify **staff members outperforming their managers in sales**. This insight helps in **performance evaluation, leadership assessment, and recognizing staff with high potential for promotions or incentives**.



JENSON§

Thank You!

We Respect your valuable time with **JENSON USA**. If you have any questions, please reach us



