

Jenson USA

- Jenson USA is an e-commerce and brick-and-mortar cycling retailer.
- Founded in 1994, offering a wide selection of bicycle parts, accessories, clothing, and bikes.
- Their mission is to inspire people to ride, experience, and explore.

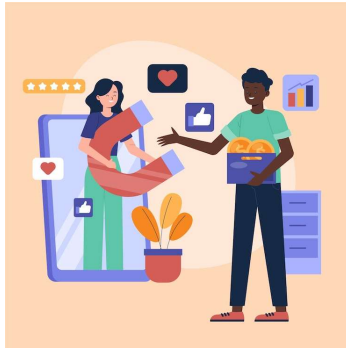




Key aspects of Jenson USA:

- **E-commerce and Retail:** They operate both online and through physical stores.
- **Product Range:** They offer a vast selection, including mountain and road bike parts, clothing, and accessories.
- **Employee Experience:** They emphasize a culture of welcome, open communication, and trust.
- **Community:** They actively support cycling communities and organizations
- **Location:** Their headquarters and one retail location are in Riverside, California, near Sycamore Canyon Park.
- **Shipping:** They offer bike and part shipping services.

Objective

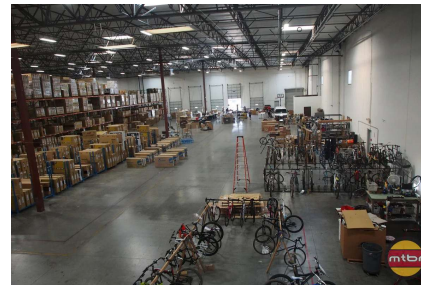


customer behavior

- ✓ customers interact with a brand, product, or service
- ✓ preferences, motivations, and decision-making processes

inventory management

- ✓ optimize inventory levels, improve
 - ✓ supply chain efficiency
- understand customer demand.





staff performance

- ✓ understand employee effectiveness
- ✓ identify areas for improvement

store operations

- ✓ sales,
- ✓ customer behavior,
- ✓ inventory management
- ✓ marketing effectiveness



Total Products Sold by Each Store



SQL Query

```
SELECT
    s.store_name, SUM(oi.quantity) AS total_products_sold
FROM
    order_items oi
    JOIN
    orders o ON oi.order_id = o.order_id
    JOIN
    stores s ON o.store_id = s.store_id
GROUP BY s.store_name;
```

Output

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

Cumulative Quantity Sold for Each Product Over Time



SQL Query

```
SELECT
    oi.product_id AS Product_ID,
    o.order_date AS Order_Date,
    SUM(oi.quantity) OVER (
        PARTITION BY oi.product_id
        ORDER BY o.order_date
    ) AS Cumulative_Quantity
FROM
    order_items oi
JOIN orders o ON oi.order_id = o.order_id;
```

Output

Product_ID	Order_Date	Cumulative_Quantity
2	2016-01-03	2
2	2016-01-14	4
2	2016-01-18	5
2	2016-02-05	6
2	2016-02-09	7
2	2016-02-26	8
2	2016-02-28	10
2	2016-02-28	10
2	2016-03-08	11

Highest Revenue Product in Each Category

SQL Query

```
WITH ranked_products AS (
    SELECT
        c.category_name,
        p.product_name,
        SUM(oi.quantity * oi.list_price) AS total_sales,
        RANK() OVER (
            PARTITION BY c.category_id
            ORDER BY SUM(oi.quantity * oi.list_price) DESC
        ) AS rnk
    FROM
        order_items oi
    JOIN products p ON oi.product_id = p.product_id
    JOIN categories c ON p.category_id = c.category_id
    GROUP BY
        c.category_id, c.category_name, p.product_name
)
SELECT
    category_name,
    product_name,
    total_sales
FROM
    ranked_products
WHERE
    rnk = 1;
```

Output

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



Top-Spending Customer (After Discounts)



SQL Query

```
SELECT
    CONCAT(c.first_name, ' ', c.last_name) AS Customer,
    SUM(oi.quantity * oi.list_price * (1 - oi.discount / 100)) AS Total_Spent
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
    Customer
ORDER BY
    Total_Spent DESC
LIMIT 1;
```

Output

Customer	Total_Spent
Pamelia Newman	3729598.42000000

Highest-Priced Product per Category



SQL Query

```
SELECT
    c.category_name,
    p.product_name,
    p.list_price
FROM
    products p
JOIN categories c ON p.category_id = c.category_id
WHERE
    (p.category_id, p.list_price) IN (
        SELECT
            category_id, MAX(list_price)
        FROM
            products
        GROUP BY
            category_id
    );
```

Output

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
Cydocross 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

Count of Orders by Customer at Each Store



SQL Query

```
SELECT
    CONCAT(c.first_name, ' ', c.last_name) AS Customer,
    s.store_name AS Store,
    COUNT(*) AS Total_Orders
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
    Customer, Store;
```

Output

Customer	Store	Total_Orders
Johnathan Velazquez	Santa Cruz Bikes	1
Nova Hess	Santa Cruz Bikes	2
Neil McCall	Santa Cruz Bikes	2
Marvin Mullins	Santa Cruz Bikes	2
Maribel William	Santa Cruz Bikes	1
Lea Key	Santa Cruz Bikes	1
Sindy Anderson	Santa Cruz Bikes	1
Lanita Burton	Santa Cruz Bikes	1

Staff Who Have Never Made a Sale



SQL Query

```
SELECT
    CONCAT(s.first_name, ' ', s.last_name) AS Staff_Name
FROM
    staffs s
WHERE
    s.staff_id NOT IN (
        SELECT DISTINCT staff_id FROM orders
    );
```

Output

Staff_Name
Fabiola Jackson
Virgie Wiggins
Jannette David
Bernardine Houston

Top 3 Products by Quantity Sold



SQL Query

```
SELECT
    p.product_name AS Product,
    SUM(oi.quantity) AS Total_Quantity
FROM
    order_items oi
    JOIN products p ON oi.product_id = p.product_id
GROUP BY
    p.product_name
ORDER BY
    Total_Quantity DESC
LIMIT 3;
```

Output

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

Find Median List Price of All Products



SQL Query

```
SELECT
    AVG(list_price) AS Median_Price
FROM (
    SELECT
        list_price,
        ROW_NUMBER() OVER (ORDER BY list_price) AS row_num,
        COUNT(*) OVER () AS total_rows
    FROM
        products
) AS ordered
WHERE
    row_num IN (
        FLOOR((total_rows + 1) / 2),
        CEIL((total_rows + 1) / 2)
    );
```

Output

Median_Price
74999.000000

List Products Never Ordered Even Once



SQL Query

```
SELECT
    p.product_name AS Product
FROM
    products p
WHERE
    NOT EXISTS (
        SELECT 1
        FROM order_items oi
        WHERE oi.product_id = p.product_id
    );
```

Output

Product
Trek 820 - 2016
Surly Krampus Frameset - 2018
Trek Kids' Dual Sport - 2018
Trek Domane SLR 6 Disc Women's - 2018
Electra Townie Go! 8i Ladies' - 2018
Trek Precaliber 12 Girl's - 2018
Electra Savannah 1 (20-inch) - Girl's - 2018
Electra Sweet Ride 1 (20-inch) - Girl's - 2018
Trek Checkpoint ALR 4 Women's - 2019

Staff Who Made More Sales Than Average



SQL Query

```
SELECT
    CONCAT(s.first_name, ' ', s.last_name) AS Staff_Name,
    COUNT(o.order_id) AS Total_Sales
FROM
    staffs s
    JOIN orders o ON s.staff_id = o.staff_id
GROUP BY
    s.staff_id
HAVING
    COUNT(o.order_id) > (
        SELECT AVG(order_count)
        FROM (
            SELECT staff_id, COUNT(order_id) AS order_count
            FROM orders
            GROUP BY staff_id
        ) AS avg_sales
    );
```

Output

Staff_Name	Total_Sales
Marcelene Boyer	553
Venita Daniel	540

Customers Who Purchased from Every Product Category



SQL Query

```
SELECT
    CONCAT(c.first_name, ' ', c.last_name) AS Customer
FROM
    customers c
WHERE NOT EXISTS (
    SELECT *
    FROM categories cat
    WHERE NOT EXISTS (
        SELECT 1
        FROM orders o
        JOIN order_items oi ON o.order_id = oi.order_id
        JOIN products p ON oi.product_id = p.product_id
        WHERE o.customer_id = c.customer_id
        AND p.category_id = cat.category_id
    )
);
```

Output

Customer
Genoveva Baldwin



Thank You

• *By-Shivji*