



# Using Advanced SQL

Presented by:Sangeeta Rajput



# About Jenson Usa

We've all experienced the beauty of the trail or the freedom of the open road. For many, cycling presents a world of possibilities and opportunities at every turn. At Jenson USA, our passion for cycling sprouted back in 1994 and has since taken root and grown into a community of people devoted to living life to its fullest. We're building a culture of people who respect each other, set the pace, and lead through service. We strive daily to better serve our customers, our fellow employees, and our community. Over the years we've found that happy people are hardworking people, and that the time we spend on two wheels together builds the greatest profit of all.



# What We Uncover

01

Customer  
behaviour

02

Staff  
Performance

03

Inventory  
Management

04

Store  
Operation

# Customer behavior





Search America's Bike Shop



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

```
SELECT
    customers.customer_id
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
        JOIN
    order_items ON order_items.order_id = orders.order_id
        JOIN
    products ON order_items.product_id = products.product_id
GROUP BY customers.customer_id
HAVING COUNT(DISTINCT products.category_id) = (SELECT
    COUNT(category_id)
FROM
    categories)
```



Search America's Bike Shop



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

```
with a as (select customers.customer_id,
concat(customers.first_name, " ",customers.last_name) full_name,
sum(order_items.quantity*(order_items.list_price - order_items.discount))sales
from customers
join orders
on customers.customer_id = orders.customer_id
join order_items
on orders.order_id = order_items.order_id
group by customers.customer_id, concat(customers.first_name, " ",customers.last_name) )

select * from
(select *, rank() over(order by sales desc) rnk
from a)b
where rnk = 1;
```

# Staff Performance





Search America's Bike Shop



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

```
select * from (select products.product_name, order_items.product_id,
sum(order_items.quantity),rank() over(order by sum(order_items.quantity) desc)ranks
from order_items join products
on order_items.product_id = products.product_id
group by products.product_name,order_items.product_id) ranked_products
where ranks <= 3;
```



Search America's Bike Shop



# Find the median value of the price list.

```
with price_data as(select list_price, row_number() over(order by list_price) as rn,
count(*) over() as total_count
from products)
select
case
when total_count % 2 = 0 then(select avg(list_price) from price_data
where rn in(total_count / 2,(total_count / 2) + 1))
else (select list_price from price_data
where rn = (total_count + 1) / 2)
end as median_price from price_data limit 1;
```



Search America's Bike Shop



# 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);
```



Search America's Bike Shop



# 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 - order_items.discount)) sales
  FROM
    order_items
      JOIN
    products ON products.product_id = order_items.product_id
      JOIN
    categories ON categories.category_id = products.category_id
  GROUP BY categories.category_id , categories.category_name , products.product_id , products.product_name)
select * from
(select *, rank() over(partition by category_id order by sales desc) as rnk
from a) b
where rnk=1;
```



Search America's Bike Shop



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

```
select * from (select categories.category_id, categories.category_name,
products.product_name, products.list_price, rank()
over(partition by categories.category_id order by products.list_price desc) rnk
from categories join products
on categories.category_id = products.category_id) a
where rnk = 1;
```



Search America's Bike Shop



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

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

# Inventory Management





Search America's Bike Shop



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

```
SELECT
    staffs.staff_id,
    CONCAT(staffs.first_name, ' ', staffs.last_name) fullname,
    orders.order_id
FROM
    staffs
        LEFT JOIN
    orders ON staffs.staff_id = orders.staff_id
WHERE
    orders.order_id IS NULL;
```



Search America's Bike Shop



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

```
with staff_sales as(select staffs.staff_id,
concat(staffs.first_name, " ", staffs.last_name) as staff_name,
sum(order_items.quantity) as total_sales from staffs
join orders on staffs.staff_id = orders.staff_id
join order_items on orders.order_id = order_items.order_id
group by staffs.staff_id, staff_name),
average_sales as(select avg(total_sales) as avg_sales from staff_sales)
select staff_name, total_sales
from staff_sales, average_sales
where staff_sales.total_sales > average_sales.avg_sales;
```

# Store Operation





Search America's Bike Shop



# 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_products_sold
```

**FROM**

```
order_items
```

**JOIN**

```
orders ON order_items.order_id = orders.order_id
```

**JOIN**

```
stores ON orders.store_id = stores.store_id
```

**GROUP BY** stores.store\_name;



Search America's Bike Shop

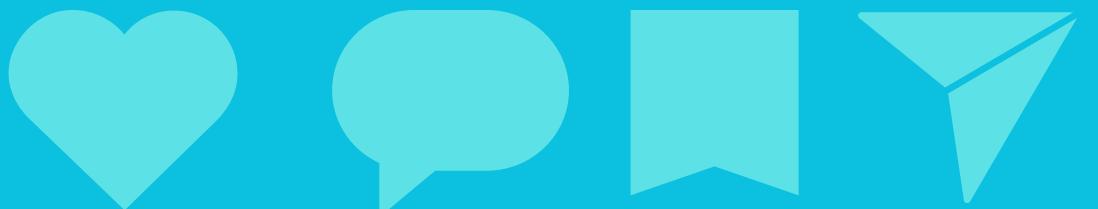


# Find the total number of orders placed by each customer per store.

```
SELECT
    store_id, customer_id, COUNT(order_id) AS total_orders
FROM
    orders
GROUP BY store_id , customer_id
ORDER BY store_id , customer_id;
```

# **THANK YOU**

**Like,comment,share,and follow**



Sangeeta Rajput