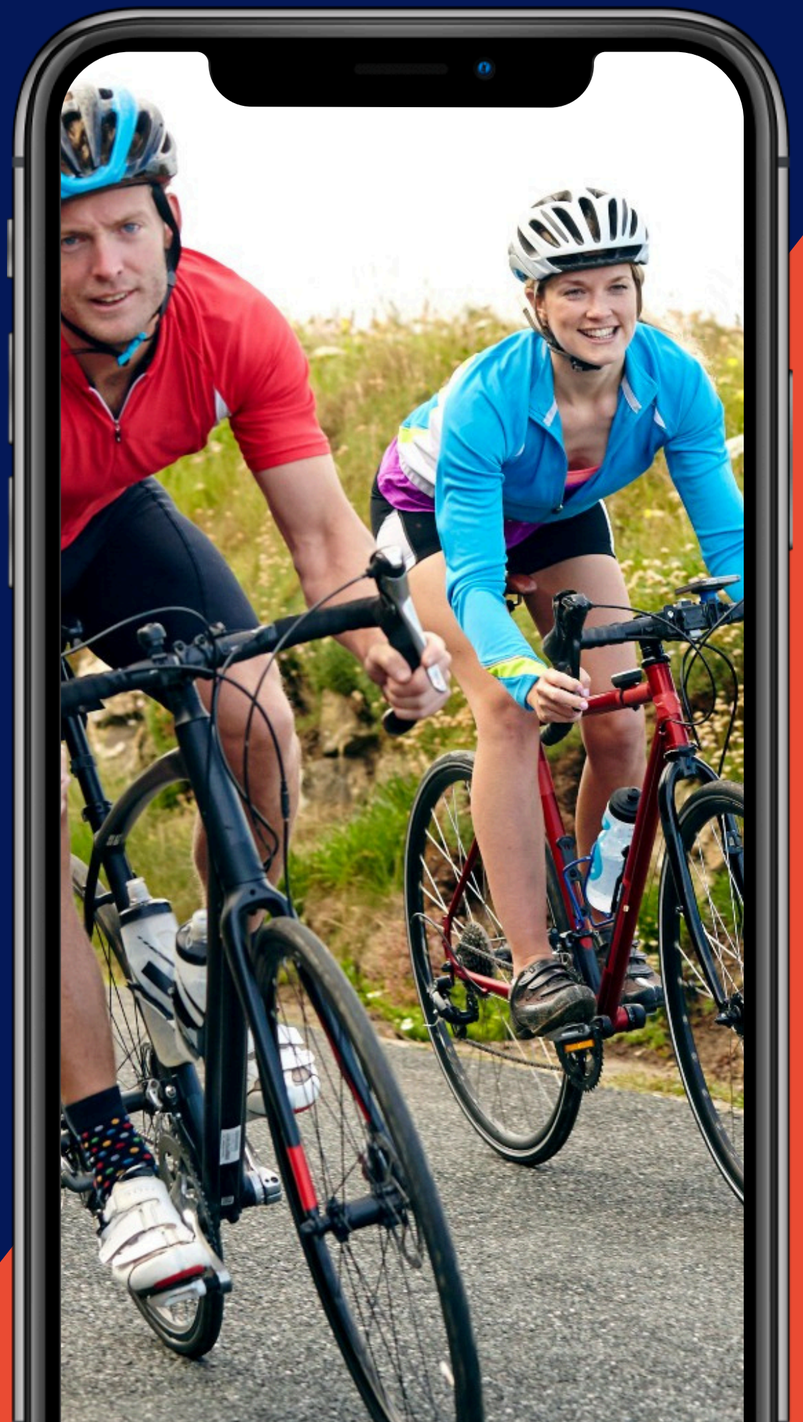
Jenson USA Sales Analysis Project

Submitted By: Paridhi Bhardwaj

WsCube Tech

DATASET OVERVIEW

The BikeStores dataset is a comprehensive sample database designed to represent the operations of a bicycle retail business.
It provides a realistic scenario for data analysis, encompassing various aspects of sales, inventory, and customer management.

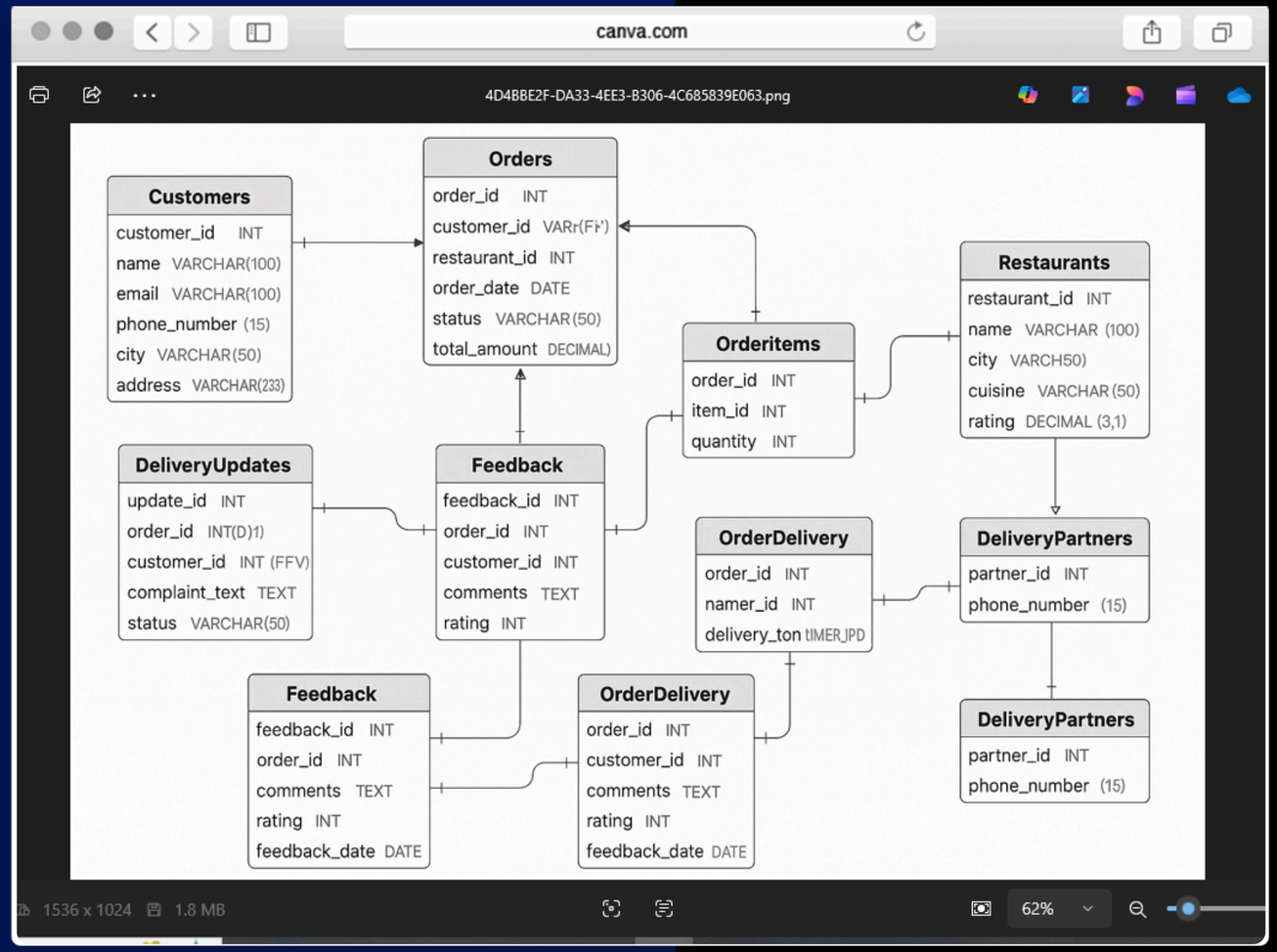


MAIN TABLES & THEIR PURPOSE

- stores: Contains information about each physical store location..
- categories: Classifies products into different types of bicycles.
- brands: Lists the different bicycle brands available.
- products: Detailed information about each bicycle product.
- customers: Records information about the customers who place orders.
- staffs: Stores details about the employees, including their roles and the store they work at.
- orders: Contains high-level information for each sales order.
- order items: Details the specific products included in each order.
- stocks: Tracks the current inventory level of each product at each store.

DATA MODELING

It clearly shows which tables need to be joined to answer specific business questions e.g., joining orders and customers to analyze customer purchasing behavior; joining products, categories, and brands to understand product characteristics.



OBJECTIVE

The primary objective of this data analysis project is to leverage the BikeStores database to derive meaningful and actionable insights that can support strategic business decisions. By analyzing the provided sales, product, customer, and inventory data. Ultimately, the goal is to provide data-driven recommendations that can lead to improved sales strategies, better inventory control, and enhanced customer satisfaction for BikeStores.

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

```sql
SELECT
    s.store_name, SUM(oi.quantity) AS total_quantity_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 store_name;
```

store_name	total_q
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 ,
 oi.product_id,
 o.order_date,
 oi.quantity,
sum(oi.quantity)  over(partition by p.product_name order by o.order_date) as cumulative_quantity
from products p join order_items oi on p.product_id=oi.product_id
 join orders o on oi.order_id = o.order_id;
```

- PARTITION BY p.product_id: Start a new running total for each product.
- ORDER BY o.order_date: Sort each product's sales by date before computing cumulative sum.

product_name	product_id	order_date	quantity	cumulative_quantity
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	257	2018-01-01	1	1
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	257	2018-01-21	2	3
Electra Amsterdam Fashion 3i Ladies' - 2017/2018	257	2018-04-30	2	5
Electra Amsterdam Fashion 7i Ladies' - 2017	81	2017-01-29	2	2
Electra Amsterdam Fashion 7i Ladies' - 2017	81	2017-02-28	1	3
Electra Amsterdam Fashion 7i Ladies' - 2017	81	2017-03-03	1	4
Electra Amsterdam Fashion 7i Ladies' - 2017	81	2017-03-09	2	6
Electra Amsterdam Fashion 7i Ladies' - 2017	81	2017-04-06	1	7
Electra Amsterdam Fashion 7i Ladies' - 2017	81	2017-04-15	2	9

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

```sql
with a as
 (SELECT
    c.category_name,
    p.product_name,
    SUM(oi.list_price * oi.quantity) sales
FROM
    categories c
        JOIN
    products p ON c.category_id = p.category_id
        JOIN
    order_items oi ON p.product_id = oi.product_id
GROUP BY c.category_name , p.product_name)


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

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

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

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

```
with a as (select c.category_name , p.product_name , p.list_price
 from products p join categories c
 on p.category_id = c.category_id)

select* from
 (select *, dense_rank()
 over(partition by category_name order by list_price desc) rnk from a)as b
 where rnk = 1;
```

category_name	product_name	list_price	rnk
Children Bicycles	Electra Straight 8 3i (20-inch) - Boy's - 2017	48999.00	1
Children Bicycles	Electra Townie 3i EQ (20-inch) - Boys' - 2017	48999.00	1
Children Bicycles	Trek Superfly 24 - 2017/2018	48999.00	1
Comfort Bicycles	Electra Townie Go! 8i - 2017/2018	259999.00	1
Cruisers Bicycles	Electra Townie Commute Go! - 2018	299999.00	1
Cruisers Bicycles	Electra Townie Commute Go! Ladies' - 2018	299999.00	1
Cyclocross Bicycles	Trek Boone 7 Disc - 2018	399999.00	1
Electric Bikes	Trek Powerfly 7 FS - 2018	499999.00	1
Electric Bikes	Trek Super Commuter+ 8S - 2018	499999.00	1
Electric Bikes	Trek Powerfly 8 FS Plus - 2017	499999.00	1
Mountain Bikes	Trek Fuel EX 98 275 Plus - 2017	529999.00	1
Mountain Bikes	Trek Remedy 98 - 2017	529999.00	1
Road Bikes	Trek Domane SLR 9 Disc - 2018	1199999.00	1

JENSON USA

BIKE SHIPPING

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

- Join orders with customers and stores
- Group by customer_id and store_id
- Count the number of orders for each pair
- Optionally, display customer and store names

```sql
select c.customer_id, concat(c.first_name ,' ' ,c.last_name)customer_name,
       s.store_name , count(o.order_id) as num_of_orders
from customers c join orders o
     on c.customer_id = o.customer_id
join stores s
        on o.store_id = s.store_id
group by  c.customer_id,s.store_name,customer_name;
```

customer_id	customer_name	store_name	num_
26	Theo Reese	Baldwin Bikes	2
27	Santos Valencia	Baldwin Bikes	2
29	Syreeta Hendricks	Baldwin Bikes	2
30	Jamaal Albert	Santa Cruz Bikes	3
31	Williemae Holloway	Santa Cruz Bikes	3
32	Araceli Golden	Santa Cruz Bikes	3
33	Deloris Burke	Santa Cruz Bikes	3
36	Bernita Mcdaniel	Baldwin Bikes	2
38	Zelma Browning	Baldwin Bikes	2
40	Ronna Butler	Santa Cruz Bikes	3
45	Bennett Armstrong	Baldwin Bikes	2
46	Monika Berg	Santa Cruz Bikes	3
47	Bridgette Guerra	Santa Cruz Bikes	3
49	Caroll Hays	Baldwin Bikes	2

Result 29

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

```sql
with a as (select c.customer_id, concat(c.first_name ,' ' ,c.last_name)customer_name,
 sum(oi.quantity * oi.list_price * (1 - oi.discount / 100))amount_spend
 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, customer_name)

SELECT *
FROM (
  SELECT *, RANK() OVER (ORDER BY amount_spend DESC) AS rnk
  FROM a
) AS b
WHERE rnk = 1;
```

Here, (quantity *list price* (1-discount/100) is taken as amount spend

customer_id	customer_name	amount_spend	rnk
10	Pamelia Newman	3729598.42000000	1

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

```
select s.staff_id, concat(s.first_name," ", s.last_name) staff_name
from staffs s
where not exists
(select 1 from orders o
 where s.staff_id = o.staff_id);
```

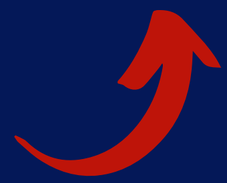
Here, the ‘subquery’ returns the output of those staff whose id match in the orders table i.e they made orders, but “Not Exists” only filter the staff for whom the subquery returns no results.

Result Grid | Filter Rows

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

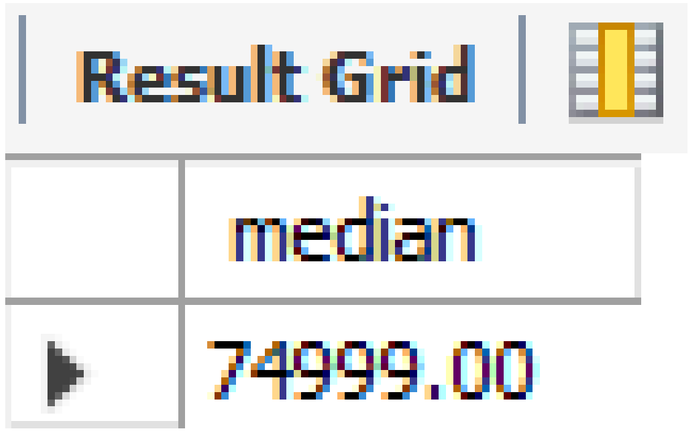
8.Find the top 3 most sold products in terms of quantity.

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



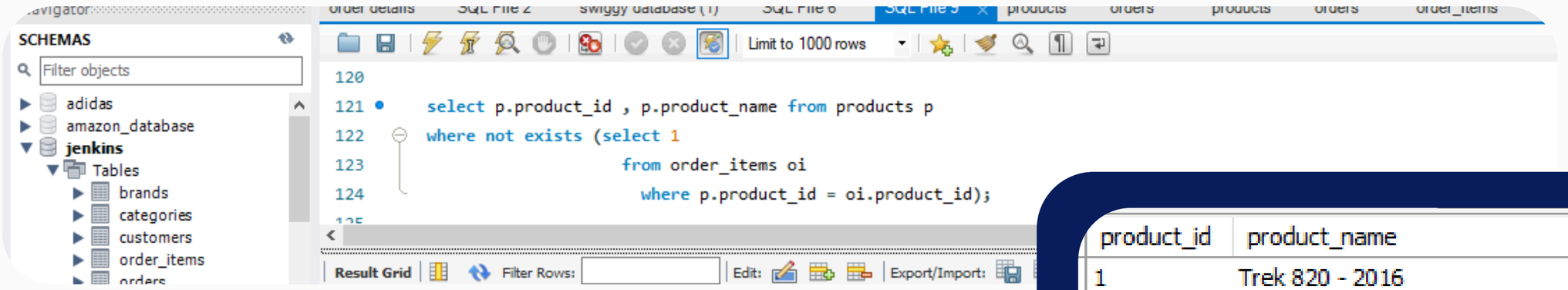
```
with a as (select p.product_id , p.product_name,
sum(oi.quantity) total_quantity
from products p
join order_items oi
on p.product_id = oi.product_id
group by  p.product_id , p.product_name)

select * from
(select * ,dense_rank() over (order by total_quantity desc)rnk from a) as b
where rnk <= 3;
```

9. Find the median value of the price list.




```sql
WITH a AS (
SELECT
  list_price,
  ROW_NUMBER() OVER (ORDER BY list_price) AS rownumber,
  COUNT(*) OVER () AS n
FROM products
)
SELECT
  CASE
    WHEN n % 2 = 0 THEN (
      SELECT AVG(list_price)
      FROM a
      WHERE rownumber IN (n / 2, n / 2 + 1)
    )
    ELSE (
      SELECT list_price
      FROM a
      WHERE rownumber = (n + 1) / 2
    )
  END AS median
FROM a
LIMIT 1;
```

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



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
	Trek Checkpoint ALR 5 Women's - 2019
	Trek Checkpoint SL 5 Women's - 2019
3	Trek Checkpoint SL 6 - 2019
3	Trek Checkpoint ALR Frameset - 2019
	NULL

“To boost sales of unsold products, highlight them with better images and descriptions, offer limited-time discounts or bundle deals, and promote them in high-visibility areas online and in-store. Gather feedback to improve appeal and address buyer concerns.”

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

```sql
with a as (select s.staff_id, concat(s.first_name,' ',s.last_name) staff_name,
coalesce(sum(oi.quantity* oi.list_price),0) total_sales
from staffs s left join orders o on s.staff_id = o.staff_id
left join order_items oi on o.order_id = oi.order_id
group by s.staff_id,staff_name)

select * from a
     where total_sales > (select avg(total_sales) from a);
```

staff_id	staff_name	total_sales
3	Genna Serrano	95272226.00
6	Marcelene Boyer	293888873.00
7	Venita Daniel	288735348.00

‘Genna Serrano’ should be appriciated for its outstanding job !

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

```sql
with a as (select o.customer_id  , count(distinct p.category_id) as category_count
from orders o join order_items oi
on o.order_id = oi.order_id
join products p on oi.product_id = p.product_id
group by customer_id )

select c.customer_id,c.first_name from customers c join a
on c.customer_id= a.customer_id
where a.category_count = (select count(*)from categories);
```

	customer_id	first_name
	9	Genoveva

The subquery Counts how many distinct product categories each customer has ordered from

“count(*) from ctegories” counts the total number of categories in category table.

JensonUSA

THANKYOU!

WsCube Tech