**BIKESTORE SQL ANALYSIS**

This bikestore dataset contain 9 tables which are:

1. Production.brands
2. Production.categories
3. Production.products
4. Production.stocks
5. Sales.customers
6. Sales.order\_items
7. Sales.orders
8. Sales.stores
9. Sales.staffs

Therefore, here are 10 questions to explore in our Bikestores dataset, along with the queries and brief explanations:

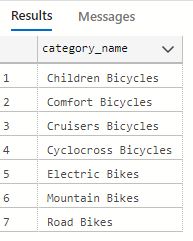
1. What are the available product categories?

SELECT

    category\_name

FROM

    production.categories



1. How many products does each brand have?

SELECT

    brand\_name,

    COUNT(\*) as product\_count

FROM

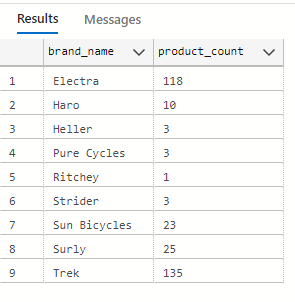
    production.brands pb

JOIN production.products pp

    ON pb.brand\_id = pp.brand\_id

GROUP BY

    brand\_name



1. Which store has the highest total sales?

SELECT

    top 1

    store\_name,

    SUM(order\_total) as total\_sales

FROM

    sales.stores ss

JOIN

    sales.orders so

    ON  ss.store\_id = so.store\_id

JOIN

    sales.order\_items si

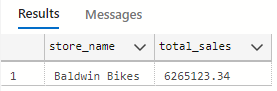
    ON si.order\_id = so.order\_id

GROUP BY

    store\_name

ORDER BY

    total\_sales DESC



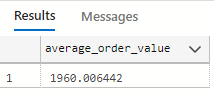
1. What is the average order value?

SELECT

    AVG(order\_total) as average\_order

FROM

    sales.order\_items



1. Who are the top 5 customers based on total spending?

SELECT

    TOP 5

    Full\_Name,

    SUM(order\_total) as total\_spending

FROM

    sales.customers sc

JOIN

    sales.orders so

    ON  sc.customer\_id = so.customer\_id

JOIN

    sales.order\_items si

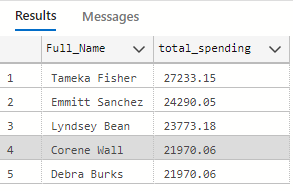
    ON  so.order\_id = si.order\_id

GROUP BY

    Full\_Name

ORDER BY

    total\_spending DESC



1. How many staff members are employed in each store?

SELECT

    store\_name,

    COUNT(\*) staff\_count

FROM

    sales.staffs ss

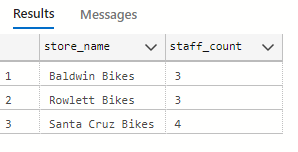
JOIN

    sales.stores st

    ON  ss.store\_id = st.store\_id

GROUP BY

    store\_name



1. Calculate the running total of sales for each store

WITH Running\_total AS

    (

SELECT

    store\_name

    ,order\_date

    ,SUM(order\_total) OVER (PARTITION BY store\_name ORDER BY order\_date) as total\_sales

FROM

    sales.stores ss

JOIN

    sales.orders so

    ON  ss.store\_id = so.store\_id

JOIN

    sales.order\_items si

    ON  so.order\_id = si.order\_id

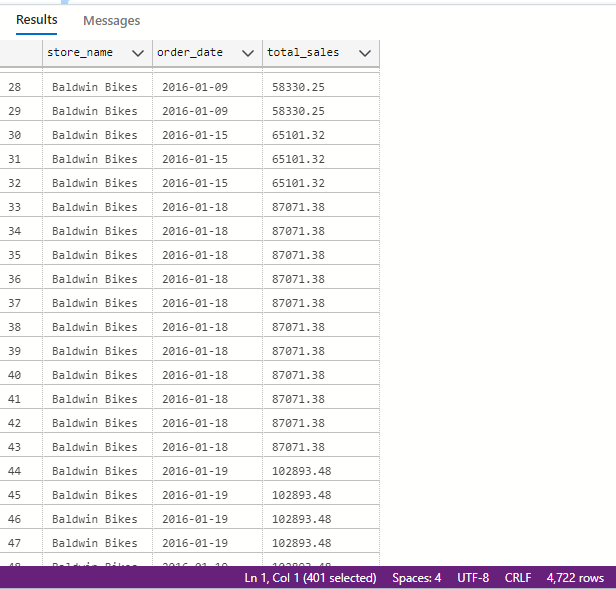
    )

SELECT

    store\_name,order\_date, total\_sales

FROM

    Running\_total



With over 4722 row records returned

1. Determine the top-selling brand in each category

WITH TopBrand AS (

SELECT

    category\_name,

    brand\_name,

    RANK() OVER (PARTITION BY category\_name ORDER BY total\_sales DESC) as brand\_rank

FROM (

        SELECT

            category\_name,

            brand\_name,

            SUM(order\_total) AS total\_sales

        FROM

            production.brands pb

        JOIN

            production.products pp

            ON

                pb.brand\_id = pp.brand\_id

        JOIN

            production.categories pc

            ON  pc.category\_id = pp.category\_id

        JOIN

            sales.order\_items so

            ON  so.product\_id = pp.product\_id

        GROUP BY

            category\_name, brand\_name

)categorybrandsales

    )

SELECT

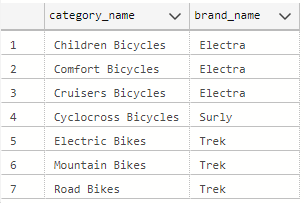
    category\_name, brand\_name

FROM

    TopBrand

WHERE

    brand\_rank = 1



1. Calculate the average sales per store for each category:

SELECT

    category\_name,

    AVG(total\_sales) AS Avg\_sales\_per\_store

FROM

    (

SELECT

    ss.store\_id

    ,category\_name

    ,SUM(order\_total) AS total\_sales

FROM

    sales.stores ss

JOIN

    sales.orders so

    ON  ss.store\_id = so.store\_id

JOIN

    sales.order\_items si

    ON  so.order\_id = si.order\_id

JOIN

    production.products pp

    ON  si.product\_id = pp.product\_id

JOIN

    production.categories pc

    ON  pp.category\_id = pc.category\_id

GROUP BY

    ss.store\_id,

    category\_name

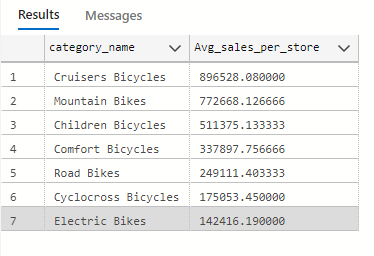
    ) StoreCategorySales

GROUP BY

    category\_name

ORDER BY

    Avg\_sales\_per\_store DESC



1. Identify the top 3 products with the highest sales in the last quarter:

SELECT

    TOP 3

    product\_name

    ,SUM(order\_total) as total\_sales

FROM

    sales.order\_items si

JOIN

    production.products pp

    ON  si.product\_id = pp.product\_id

JOIN

    sales.orders so

    ON  si.order\_id = so.order\_id

WHERE

    order\_date >= DATEADD(QUARTER, DATEDIFF(QUARTER, 0, 2018) - 1 , 0)

GROUP BY

    product\_name

ORDER BY

    total\_sales DESC

