# **Bike Sales & Rider Analysis (2021-2022)**

This repository holds the assets for a data analysis project focused on a bike-sharing company's performance across 2021 and 2022. The project uses SQL for data preparation and Power BI for creating a dynamic, interactive dashboard to visualize key business insights.

## **📊 Power BI Dashboard**

The final output is a comprehensive dashboard that summarizes the key findings from the dataset. It provides a clear view of revenue, profitability, rider demographics, and usage patterns.

## **🚀 Key Insights from the Analysis**

* **Overall Financial Health**: The company is in a strong financial position, with **$4.05M in revenue** and **$2.79M in profit**, yielding an impressive **45.48% profit margin**.
* **Rider Demographics**: The business has a very strong loyal customer base, with **registered riders accounting for 81.8%** of the total.
* **Peak Usage Hours**: The most profitable times are during the **midday and early evening (10 AM - 7 PM)**, suggesting high usage during standard business hours and commute times.
* **Peak Usage Days**: **Wednesday and Friday** are the most profitable days, indicating strong midweek and end-of-week activity.
* **Seasonal Trends**: Revenue is highest in **Season 3 (Fall)**, showing a clear seasonal influence on rider activity.

## **🛠️ Tools and Technologies**

* **Data Transformation & Analysis**: **SQL** (using SQL Server)
* **Data Visualization & Dashboarding**: **Microsoft Power BI**

## **📂 Project Files**

* bike\_share\_yr\_0.csv: Raw bike sharing data for the year 2021.
* bike\_share\_yr\_1.csv: Raw bike sharing data for the year 2022.
* cost\_table.csv: Supplementary data containing cost and pricing information.
* SQLQuery3.sql: The SQL script used to clean, merge, and prepare the data for analysis.
* Dashboard ss.png: The screenshot of the final Power BI dashboard.

## **⚙️ Methodology**

The project was executed in two main stages:

### **1. Data Preparation (SQL)**

The raw data from the CSV files was first loaded into a SQL Server database. The SQLQuery3.sql script performs several key transformations:

1. **Union Data**: The datasets for 2021 (bike\_share\_yr\_0) and 2022 (bike\_share\_yr\_1) were combined into a single dataset using a Common Table Expression (CTE).
2. **Join Datasets**: The combined bike share data was joined with the cost\_table to bring pricing and cost information into the main dataset.
3. **Calculate Financial Metrics**: New columns for **Revenue** and **Profit** were calculated directly in the SQL query to simplify analysis in Power BI. The query calculates revenue (riders \* price) and profit ((riders \* price) - COGS).

-- A simplified version of the core logic in SQLQuery3.sql  
with cte as (  
 -- Combine both years of data  
 select \* from bike\_share\_yr\_0  
 union all  
 select \* from bike\_share\_yr\_1  
)  
  
select  
 dteday,  
 season,  
 hr,  
 weekday,  
 a.yr,  
 -- Calculate key financial metrics  
 riders\*price as Revenue,  
 (riders\*price) - COGS as Profit  
from cte a  
left join cost\_table b on a.yr = b.yr;

### **2. Visualization (Power BI)**

The cleaned and processed data from the SQL query was then connected to Power BI. The dashboard was built using several visualizations to highlight the key insights:

* **KPI Cards**: To display high-level metrics like Total Revenue, Profit, Profit Margin, and Total Riders.
* **Line and Bar Chart**: To show the trend of riders and revenue over the months.
* **Bar Chart**: To compare revenue generated across different seasons.
* **Donut Chart**: To visualize the breakdown of registered vs. casual riders.
* **Matrix**: To provide a detailed, granular view of revenue by hour.

## **(How to Reproduce)**

1. **Database Setup**: Import the three .csv files (bike\_share\_yr\_0.csv, bike\_share\_yr\_1.csv, cost\_table.csv) into a SQL database.
2. **Run SQL Script**: Execute the SQLQuery3.sql script to create the final, analysis-ready view or table.
3. **Connect Power BI**: Connect Power BI Desktop to your SQL database and import the data generated from the script.
4. **Build Dashboard**: Recreate the visuals from the dashboard screenshot using the imported data.