SQL PROJECT

ROYAL ENFIELD SALES SQL ANALYSIS

Presented by : Shaik Baji

Github_Account

PROJECT GOAL

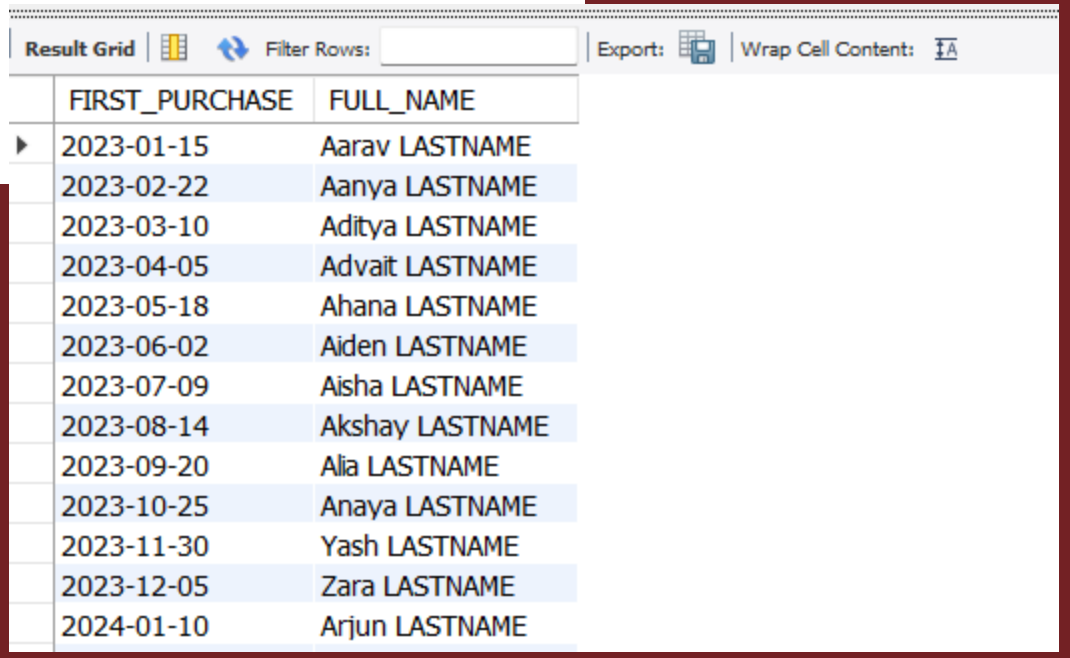
- The main objective of this project is to derive valuable insights from the dataset, gain a comprehension of customer behavior, examine trends in bike sales and pricing, monitor service history, and evaluate the performance of dealers. This data holds significance for decision-making, refining marketing strategies, and enhancing overall business operations pertaining to the sale and servicing of bikes.

TABLES USED

- bikes table
- sales table
- dealers table
- customer table
- feedback table
- Service records table

(1)FIND THE DATE OF THE FIRST PURCHASE FOR EACH CUSTOMER.

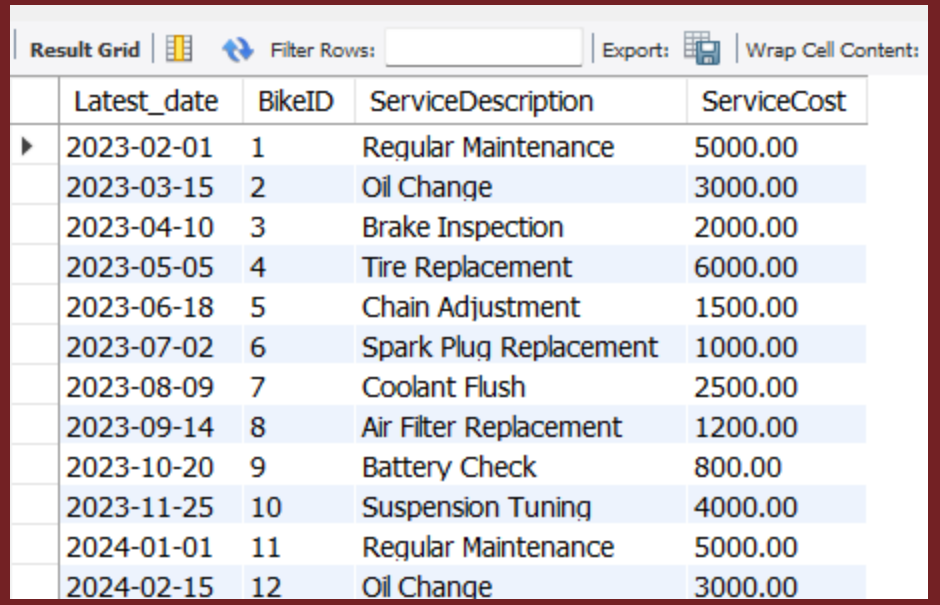
```
SELECT MIN(SALEDATE) AS FIRST_PURCHASE,
CONCAT(FIRSTNAME," ","LASTNAME") AS FULL_NAME
FROM CUSTOMERS A INNER JOIN SALES B
ON A.CUSTOMERID=B.CUSTOMERID
GROUP BY FULL_NAME
ORDER BY FIRST_PURCHASE;
```



FIRST_PURCHASE	FULL_NAME
2023-01-15	Aarav LASTNAME
2023-02-22	Aanya LASTNAME
2023-03-10	Aditya LASTNAME
2023-04-05	Advait LASTNAME
2023-05-18	Ahana LASTNAME
2023-06-02	Aiden LASTNAME
2023-07-09	Aisha LASTNAME
2023-08-14	Akshay LASTNAME
2023-09-20	Alia LASTNAME
2023-10-25	Anaya LASTNAME
2023-11-30	Yash LASTNAME
2023-12-05	Zara LASTNAME
2024-01-10	Arjun LASTNAME

(2)RETRIEVE THE LATEST SERVICE FOR EACH BIKE.

```
SELECT MAX(ServiceDate) AS Latest_date,
BikeID,ServiceDescription,ServiceCost
FROM servicerecords
GROUP BY BikeID,ServiceDescription,ServiceCost
ORDER BY YEAR(Latest_date),MONTH(Latest_date),DATE(Latest_date) DESC;
```



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

Latest_date	BikeID	ServiceDescription	ServiceCost
2023-02-01	1	Regular Maintenance	5000.00
2023-03-15	2	Oil Change	3000.00
2023-04-10	3	Brake Inspection	2000.00
2023-05-05	4	Tire Replacement	6000.00
2023-06-18	5	Chain Adjustment	1500.00
2023-07-02	6	Spark Plug Replacement	1000.00
2023-08-09	7	Coolant Flush	2500.00
2023-09-14	8	Air Filter Replacement	1200.00
2023-10-20	9	Battery Check	800.00
2023-11-25	10	Suspension Tuning	4000.00
2024-01-01	11	Regular Maintenance	5000.00
2024-02-15	12	Oil Change	3000.00

(3)FIND THE PRICE DIFFERENCE BETWEEN THE CURRENT BIKE AND THE NEXT BIKE IN THE SAME YEAR.

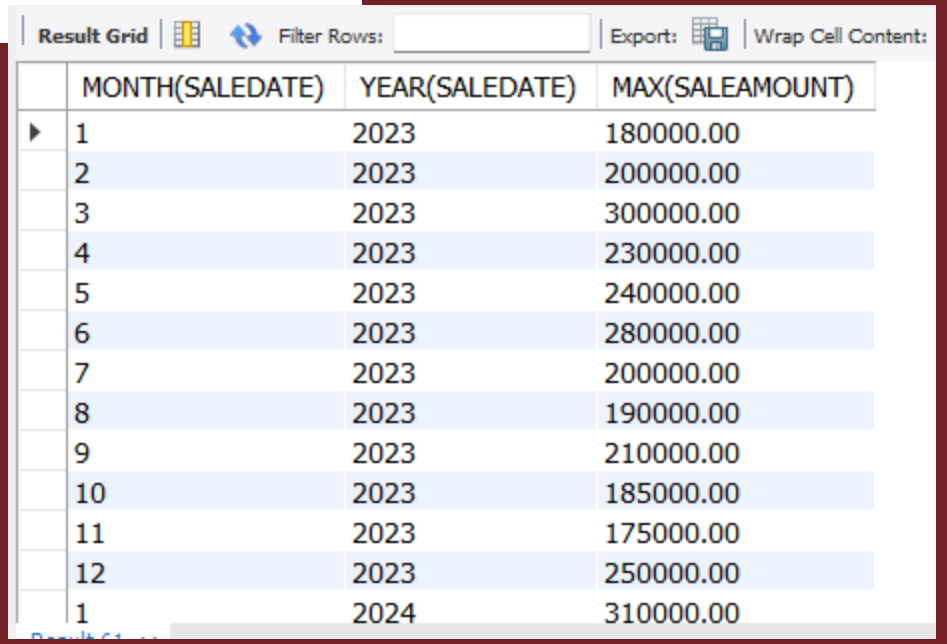
```
SELECT YEAR,MODEL,PRICE,
LEAD(PRICE) OVER(PARTITION BY YEAR ORDER BY PRICE) - PRICE AS PRICE_DIFF
FROM BIKES;
```

Result Grid | Filter Rows: | Export:

YEAR	MODEL	PRICE	PRICE_DIFF
2022	Classic 350	180000	10000
2022	Meteor 350	190000	10000
2022	Thunderbird 350	200000	10000
2022	Interceptor 350	210000	10000
2022	Bullet 500	220000	10000
2022	Classic 500	230000	10000
2022	Himalayan	240000	40000
2022	Continental GT 535	280000	20000
2022	Interceptor 650	300000	20000

(4)FIND THE MAXIMUM SALE AMOUNT FOR EACH MONTH.

```
SELECT MONTH(SALEDATE),YEAR(SALEDATE),MAX(SALEAMOUNT)
FROM SALES
GROUP BY MONTH(SALEDATE),YEAR(SALEDATE);
```



MONTH(SALEDATE)	YEAR(SALEDATE)	MAX(SALEAMOUNT)
1	2023	180000.00
2	2023	200000.00
3	2023	300000.00
4	2023	230000.00
5	2023	240000.00
6	2023	280000.00
7	2023	200000.00
8	2023	190000.00
9	2023	210000.00
10	2023	185000.00
11	2023	175000.00
12	2023	250000.00
1	2024	310000.00

(5)CONCATENATE THE FIRST AND LAST NAMES OF CUSTOMERS, AND DISPLAY THEM IN UPPER CASE.

```
SELECT FIRSTNAME,LASTNAME,
UPPER(CONCAT(FIRSTNAME," ",LASTNAME)) AS FULL_NAME
FROM CUSTOMERS;
```

Result Grid | Filter Rows: | Export:

FIRSTNAME	LASTNAME	FULL_NAME
Aarav	Sharma	AARAV SHARMA
Aanya	Patel	AANYA PATEL
Aditya	Verma	ADITYA VERMA
Advait	Singh	ADVAIT SINGH
Ahana	Kumar	AHANA KUMAR
Aiden	Gupta	AIDEN GUPTA
Aisha	Das	AISHA DAS
Akshay	Chatterjee	AKSHAY CHATTERJEE
Alia	Mukherjee	ALIA MUKHERJEE
Anaya	Joshi	ANAYA JOSHI
Yash	Sinha	YASH SINHA
Zara	Nair	ZARA NAIR
Arjun	Shah	ARJUN SHAH

(6)DETERMINE THE QUARTER IN WHICH EACH SALE OCCURRED.

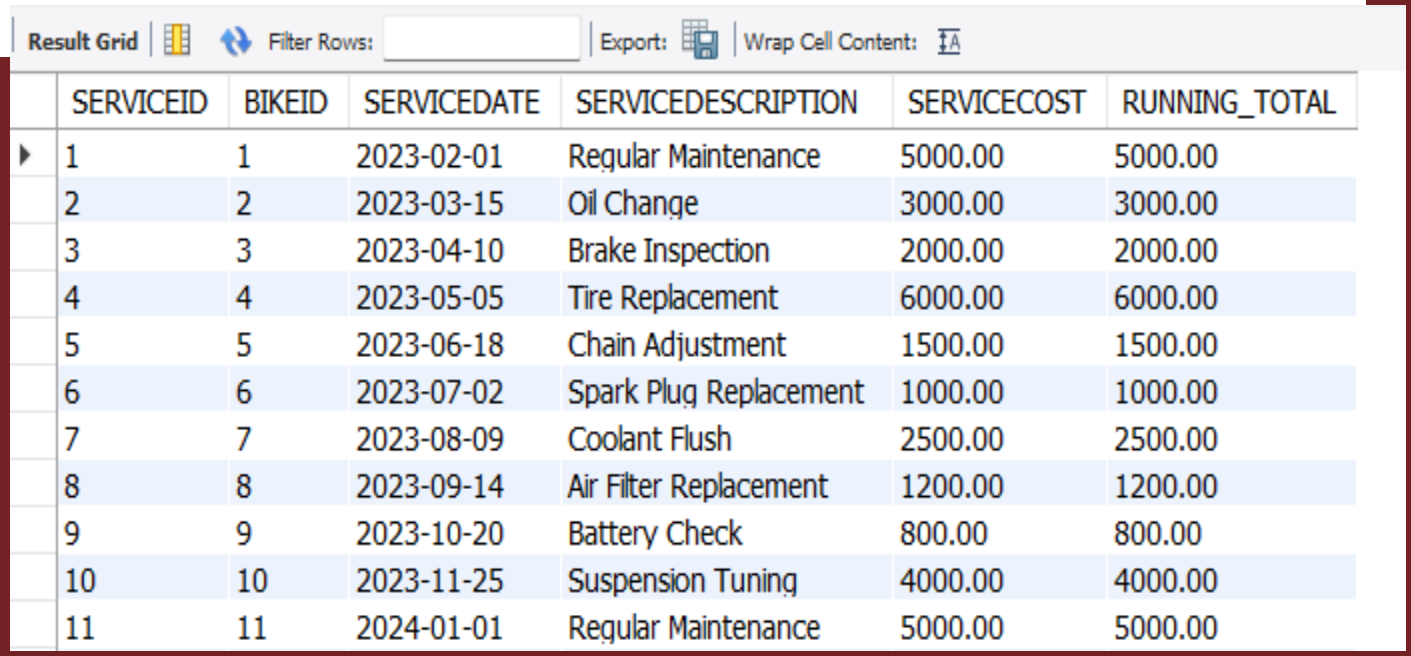
```
SELECT SALEID,QUARTER(SALEDATE) AS QUARTER,SALEDATE
FROM SALES
GROUP BY SALEID
ORDER BY SALEDATE;
```

Result Grid | Filter Rows:

SALEID	QUARTER	SALEDATE
1	1	2023-01-15
2	1	2023-02-22
3	1	2023-03-10
4	2	2023-04-05
5	2	2023-05-18
6	2	2023-06-02
7	3	2023-07-09
8	3	2023-08-14
9	3	2023-09-20
10	4	2023-10-25
11	4	2023-11-30
12	4	2023-12-05

(7)CALCULATE THE RUNNING TOTAL OF SERVICE COSTS FOR EACH BIKE.

```
SELECT SERVICEID,BIKEID,SERVICEDATE,SERVICEDESCRIPTION,SERVICECOST,
SUM(SERVICECOST) OVER(PARTITION BY BIKEID ORDER BY SERVICECOST) AS RUNNING_TOTAL
FROM SERVICERECORDS;
```



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

SERVICEID	BIKEID	SERVICEDATE	SERVICEDESCRIPTION	SERVICECOST	RUNNING_TOTAL
1	1	2023-02-01	Regular Maintenance	5000.00	5000.00
2	2	2023-03-15	Oil Change	3000.00	3000.00
3	3	2023-04-10	Brake Inspection	2000.00	2000.00
4	4	2023-05-05	Tire Replacement	6000.00	6000.00
5	5	2023-06-18	Chain Adjustment	1500.00	1500.00
6	6	2023-07-02	Spark Plug Replacement	1000.00	1000.00
7	7	2023-08-09	Coolant Flush	2500.00	2500.00
8	8	2023-09-14	Air Filter Replacement	1200.00	1200.00
9	9	2023-10-20	Battery Check	800.00	800.00
10	10	2023-11-25	Suspension Tuning	4000.00	4000.00
11	11	2024-01-01	Regular Maintenance	5000.00	5000.00

(8)FIND THE TOP DEALERS BASED ON THE TOTAL SALES AMOUNT ACROSS THE BIKES.

```sql
-- METHOD 1
SELECT B.BIKEID,A.DEALERNAME,SUM(B.SALEAMOUNT) AS TOTAL_SALES FROM DEALERS A
INNER JOIN SALES B
ON A.DEALERID=B.DEALERID
GROUP BY B.BIKEID,A.DEALERNAME;
-- METHOD 2
SELECT B.BIKEID,A.DEALERNAME,SUM(SALEAMOUNT) OVER(PARTITION BY A.DEALERID) AS TOTAL_SALES
FROM DEALERS A
INNER JOIN SALES B
ON A.DEALERID=B.DEALERID;
```

Result Grid | Filter Rows: | Export:

BIKEID	DEALERNAME	TOTAL_SALES
1	Royal Motors	180000.00
3	Classic Bikes	200000.00
5	Thunder Motors	300000.00
7	Himalayan Cycles	230000.00
9	Bullet Riders	240000.00
11	Interceptor Bikes	280000.00
13	Classic Motors	200000.00
15	Vintage Bikes	190000.00
17	Royal Wheels	210000.00

(9)FIND THE COUNT OF BIKES SOLD EACH YEAR AND CATEGORIZE THEM INTO THREE GROUPS: 'LOW',' MEDIUM', AND 'HIGH' BASED ON THEIR PRICES.

```
SELECT YEAR(SALEDATE),COUNT(A.BIKEID) AS BIKE_COUNT,
SUM(CASE WHEN B.SALEAMOUNT<200000 THEN 1 ELSE 0 END) AS 'Low',
SUM(CASE WHEN B.SALEAMOUNT>200000 AND B.SALEAMOUNT<300000 THEN 1 ELSE 0 END) AS 'Medium',
SUM(CASE WHEN B.SALEAMOUNT>300000 THEN 1 ELSE 0 END) AS 'High'
FROM BIKES A INNER JOIN SALES B ON A.BIKEID=B.BIKEID
GROUP BY YEAR(SALEDATE);
```

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

YEAR(SALEDATE)	BIKE_COUNT	Low	Medium	High
2023	12	4	5	0
2024	12	2	5	4
2025	12	2	6	2
2026	12	2	6	2
2027	12	1	7	2
2028	12	2	6	2
2029	12	2	6	2
2030	12	2	6	2
2031	4	1	2	0

(10)FIND THE TOP 5 BIKE MODELS WITH THE HIGHEST COST.

```
SELECT MODEL,MAX(PRICE) AS HIGHEST_COST FROM BIKES
GROUP BY MODEL
ORDER BY HIGHEST_COST DESC
LIMIT 5;
```

Result Grid | Filter Rows:

MODEL	HIGHEST_COST
Interceptor 650	345000
Continental GT 650	340000
Himalayan	290000
Continental GT 535	280000
Classic 500	255000

(11)WRITE A QUERY TO COMPARE A BIKE MODEL PRICE IN YEARS 2022 AND 2023. RETRIEVE IN TWO DIFFERENT COLUMN FOR 2022 AND 2023.

```
SELECT MODEL,
MAX(CASE WHEN YEAR=2022 THEN PRICE END) AS BIKE_2022,
MAX(CASE WHEN YEAR=2023 THEN PRICE END) AS BIKE_2023
FROM BIKES
GROUP BY MODEL;
```

Result Grid | Filter Rows: | Export:

MODEL	BIKE_2022	BIKE_2023
Classic 350	180000	215000
Bullet 500	220000	235000
Thunderbird 350	200000	210000
Himalayan	240000	290000
Interceptor 650	300000	345000
Continental GT 650	320000	340000
Classic 500	230000	255000
Meteor 350	190000	220000
Interceptor 350	210000	220000
Continental GT 535	280000	NULL
Bullet 350	NULL	200000

(12)RETRIEVE THE COUNT OF HIGHLY SOLD BIKE MODEL IN BOTH THE YEAR WITH ITS SALEAMOUNT.

```
SELECT MODEL,COUNT(MODEL) AS BIKE_COUNT,SUM(PRICE) AS SALEAMOUNT FROM BIKES
GROUP BY MODEL
ORDER BY BIKE_COUNT DESC;
```

Result Grid | Filter Rows: | Export:

MODEL	BIKE_COUNT	SALEAMOUNT
Meteor 350	26	5550000
Classic 350	24	4918000
Himalayan	24	6450000
Classic 500	24	5867000
Interceptor 650	23	7615000
Continental GT 650	23	7420000
Thunderbird 350	22	4494000
Bullet 500	13	2935000
Bullet 350	13	2506000

(13)WRITE A QUERY TO RETRIEVE HOW MUCH BIKES ARE SOLD BY EACH DEALER IN YEAR 2023(DEALER NAME, BIKE SALES COUNT, TOTAL SALES AMOUNT)

```
SELECT A.DEALERNAME,COUNT(B.BIKEID) AS BIKE_COUNT,SUM(SALEAMOUNT) AS TOTAL_SALES
FROM DEALERS A INNER JOIN SALES B
ON B.DEALERID=A.DEALERID
WHERE YEAR(SALEDATE)=2023
GROUP BY A.DEALERNAME;
```

Result Grid | Filter Rows: | Export:

DEALERNAME	BIKE_COUNT	TOTAL_SALES
Royal Motors	1	180000.00
Classic Bikes	1	200000.00
Thunder Motors	1	300000.00
Himalayan Cycles	1	230000.00
Bullet Riders	1	240000.00
Interceptor Bikes	1	280000.00
Classic Motors	1	200000.00
Vintage Bikes	1	190000.00
Royal Wheels	1	210000.00
Enfield Paradise	1	185000.00
Eagle Motors	1	175000.00
Golden Bikes	1	250000.00

(14)FROM THE ABOVE DEALERS TABLE RETRIEVE THE COUNT OF DEALERS IN EACH LOCATION.

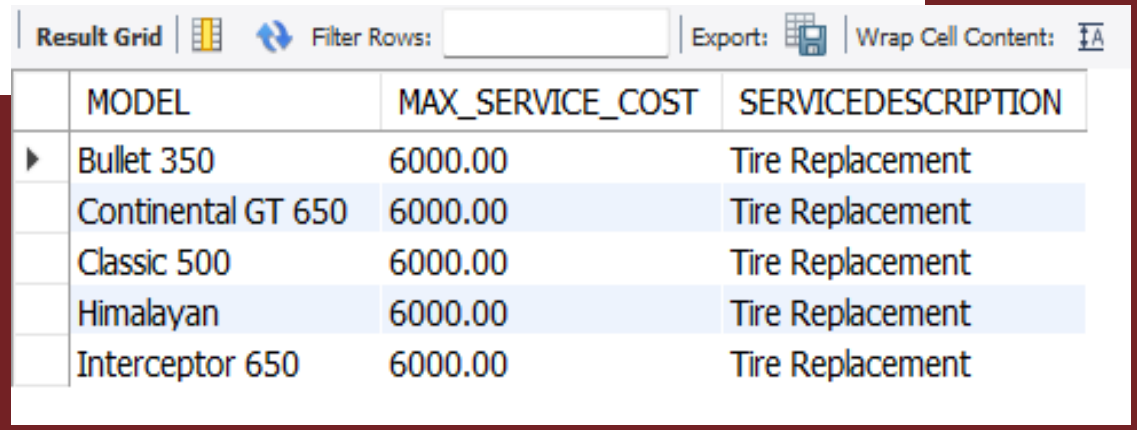
```
SELECT LOCATION,COUNT(*) AS COUNT_OF_DEALERS FROM DEALERS
GROUP BY LOCATION
ORDER BY COUNT_OF_DEALERS DESC;
```

Result Grid | Filter Rows:

LOCATION	COUNT_OF_DEALERS
Ahmedabad	3
Nagpur	3
Surat	3
Delhi	2
Mumbai	2
Bangalore	2
Chennai	2
Kolkata	2
Hyderabad	2
Pune	2
Jaipur	2
Lucknow	2
Chandigarh	2
Indore	2
Bhopal	2
Raipur	2
Vadodara	2
Panaji	2
Mangalore	2
Bidar	2
Raichur	2

(15)RETRIEVE THE TOP 5 MODELS FROM THE BIKES TABLE AND THE MAX SERVICE COST OF EACH BIKE WITH ITS DESCRIPTION FROM SERVICE RECORD TABLE.

```
SELECT A.MODEL,MAX(B.SERVICECOST) AS MAX_SERVICE_COST,B.SERVICEDESCRIPTION
FROM BIKES A INNER JOIN SERVICERECORDS B ON A.BIKEID=B.BIKEID
GROUP BY A.MODEL,B.SERVICEDESCRIPTION
ORDER BY MAX_SERVICE_COST DESC LIMIT 5;
```



MODEL	MAX_SERVICE_COST	SERVICEDESCRIPTION
Bullet 350	6000.00	Tire Replacement
Continental GT 650	6000.00	Tire Replacement
Classic 500	6000.00	Tire Replacement
Himalayan	6000.00	Tire Replacement
Interceptor 650	6000.00	Tire Replacement

THANK YOU

Shaik Baji

8688616393

bajibabblu3@gmail.com

Github_Account

LinkedIn_Account