BIKE SERVICE CENTER DATA ANALYSYS

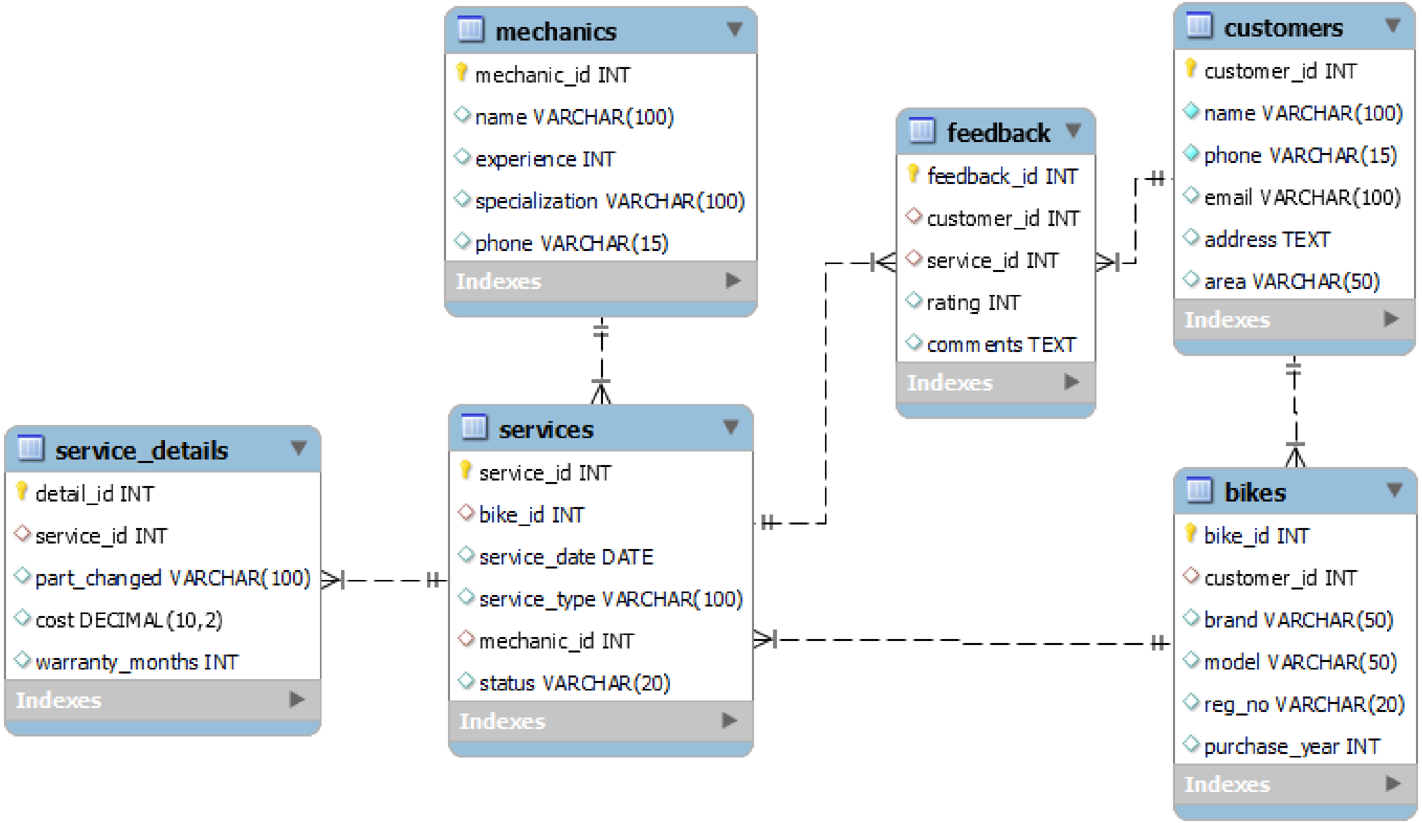
MOHAMMAD HASAN

Introduction

Objective:

- The objective of this project is to analyze the Bike Service Center's customer record management system to optimize overall service efficiency.
- This includes ensuring timely availability of spare parts, tracking revenue, and identifying performing employees.
- Based on customer feedback, we aim to monitor service quality and implement meaningful improvements.
- Overall, the goal is to make data-driven decisions that enhance customer satisfaction and business growth.

DTATBASE STRUCTURE



mechanics

mechanic_id INT
name VARCHAR(100)
experience INT
specialization VARCHAR(100)
phone VARCHAR(15)

Indexes

customers

customer_id INT
name VARCHAR(100)
phone VARCHAR(15)
email VARCHAR(100)
address TEXT
area VARCHAR(50)

Indexes

feedback

feedback_id INT
customer_id INT
service_id INT
rating INT
comments TEXT

Indexes

service_details

detail_id INT
service_id INT
part_changed VARCHAR(100)
cost DECIMAL(10,2)
warranty_months INT

Indexes

services

service_id INT
bike_id INT
service_date DATE
service_type VARCHAR(100)
mechanic_id INT
status VARCHAR(20)

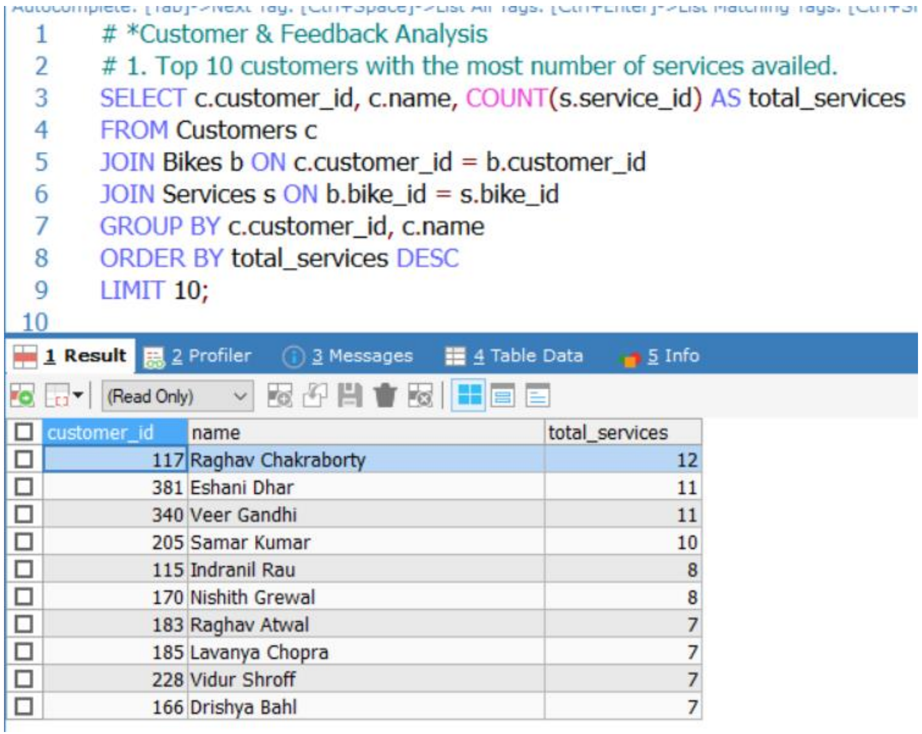
Indexes

bikes

bike_id INT
customer_id INT
brand VARCHAR(50)
model VARCHAR(50)
reg_no VARCHAR(20)
purchase_year INT

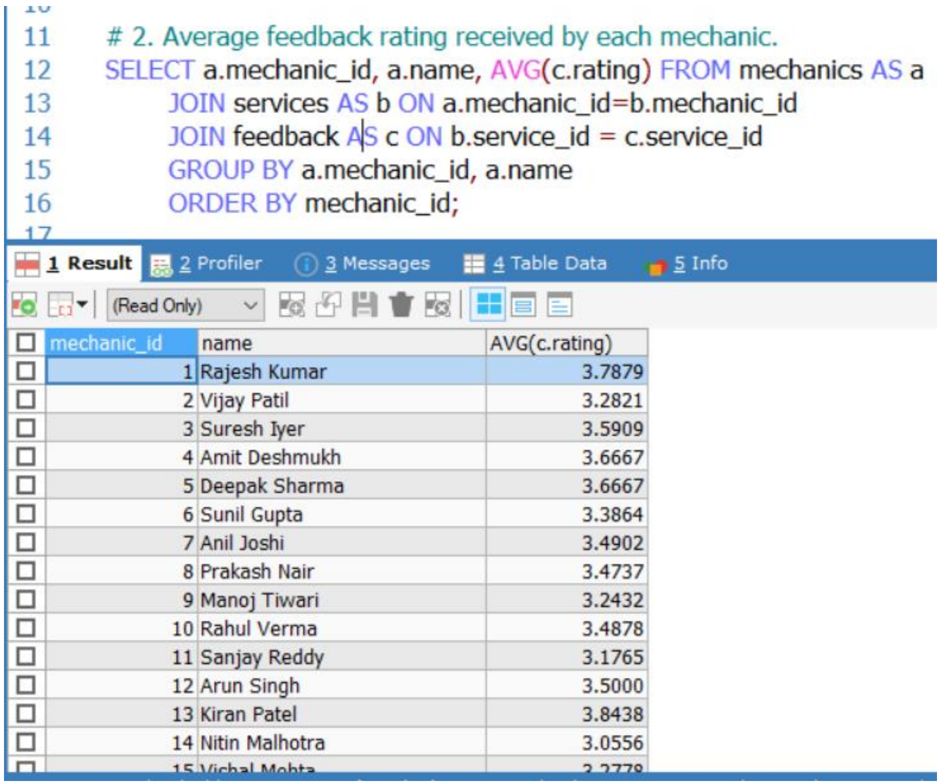
Indexes

Customer & Feedback Analysis


Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [Ctrl+Sh
1 Result
2 Profiler
3 Messages
4 Table Data
5 Info
(Read Only)

```sql
# *Customer & Feedback Analysis
# 1. Top 10 customers with the most number of services availed.
SELECT c.customer_id, c.name, COUNT(s.service_id) AS total_services
FROM Customers c
JOIN Bikes b ON c.customer_id = b.customer_id
JOIN Services s ON b.bike_id = s.bike_id
GROUP BY c.customer_id, c.name
ORDER BY total_services DESC
LIMIT 10;
```

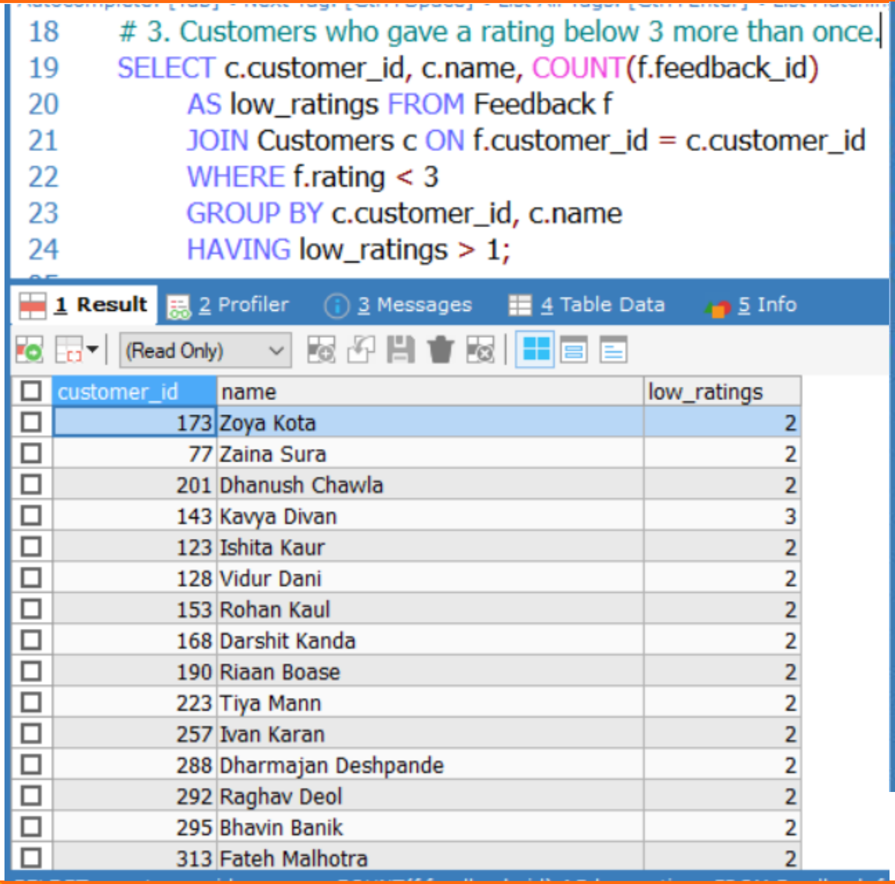
□	customer_id	name	total_services
□	117	Raghav Chakraborty	12
□	381	Eshani Dhar	11
□	340	Veer Gandhi	11
□	205	Samar Kumar	10
□	115	Indranil Rau	8
□	170	Nishith Grewal	8
□	183	Raghav Atwal	7
□	185	Lavanya Chopra	7
□	228	Vidur Shroff	7
□	166	Drishya Bahl	7



```sql
# 2. Average feedback rating received by each mechanic.
SELECT a.mechanic_id, a.name, AVG(c.rating) FROM mechanics AS a
    JOIN services AS b ON a.mechanic_id=b.mechanic_id
    JOIN feedback AS c ON b.service_id = c.service_id
    GROUP BY a.mechanic_id, a.name
    ORDER BY mechanic_id;
```

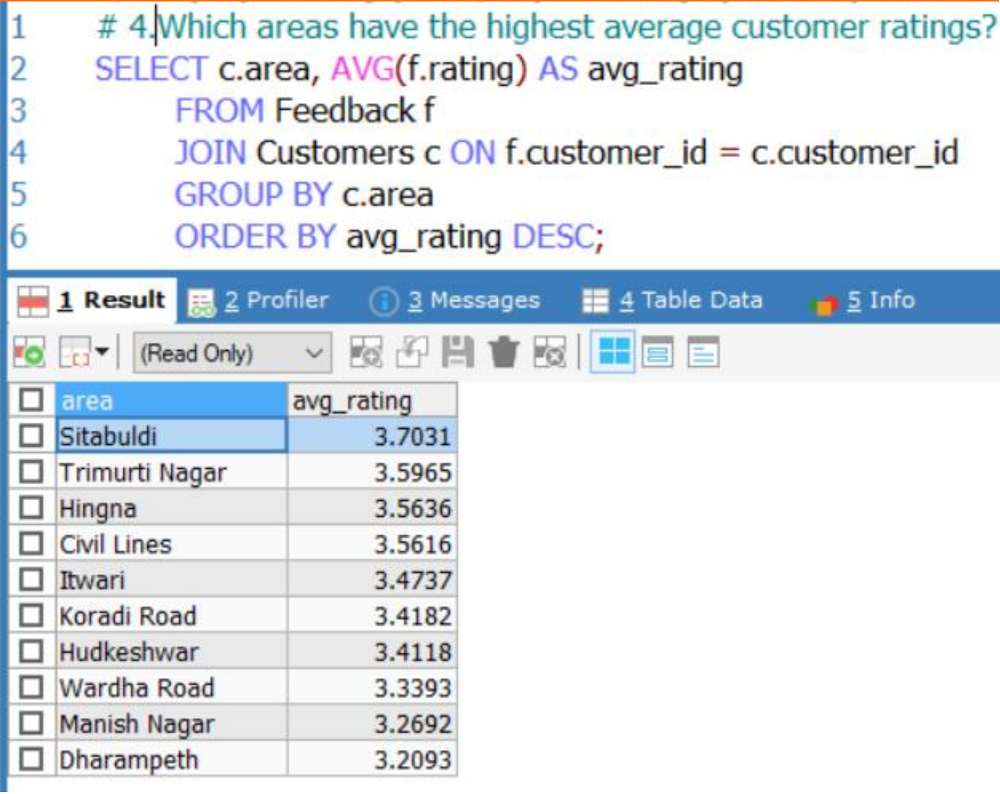
□	mechanic_id	name	AVG(c.rating)
□	1	Rajesh Kumar	3.7879
□	2	Vijay Patil	3.2821
□	3	Suresh Iyer	3.5909
□	4	Amit Deshmukh	3.6667
□	5	Deepak Sharma	3.6667
□	6	Sunil Gupta	3.3864
□	7	Anil Joshi	3.4902
□	8	Prakash Nair	3.4737
□	9	Manoj Tiwari	3.2432
□	10	Rahul Verma	3.4878
□	11	Sanjay Reddy	3.1765
□	12	Arun Singh	3.5000
□	13	Kiran Patel	3.8438
□	14	Nitin Malhotra	3.0556
□	15	Vishal Mehta	3.2778

Customer & Feedback Analysis



```sql
# 3. Customers who gave a rating below 3 more than once.
SELECT c.customer_id, c.name, COUNT(f.feedback_id)
        AS low_ratings FROM Feedback f
        JOIN Customers c ON f.customer_id = c.customer_id
        WHERE f.rating < 3
        GROUP BY c.customer_id, c.name
        HAVING low_ratings > 1;
```

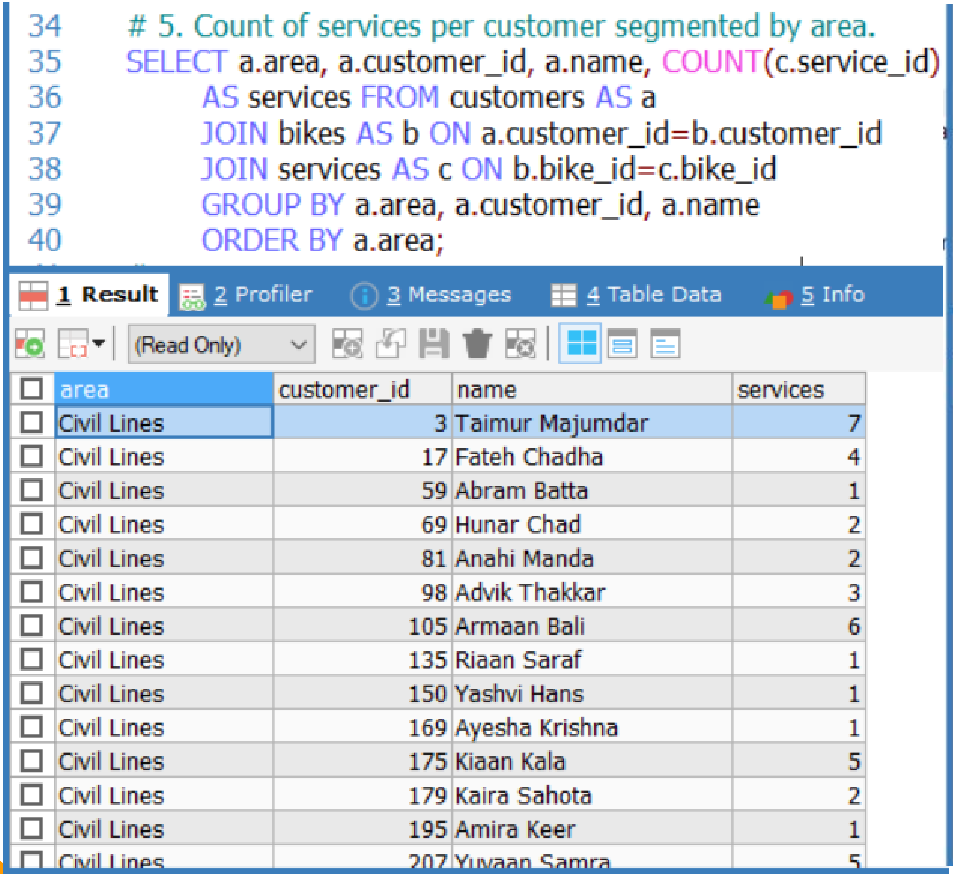
□	customer_id	name	low_ratings
□	173	Zoya Kota	2
□	77	Zaina Sura	2
□	201	Dhanush Chawla	2
□	143	Kavya Divan	3
□	123	Ishita Kaur	2
□	128	Vidur Dani	2
□	153	Rohan Kaul	2
□	168	Darshit Kanda	2
□	190	Riaan Boase	2
□	223	Tiya Mann	2
□	257	Ivan Karan	2
□	288	Dharmajan Deshpande	2
□	292	Raghav Deol	2
□	295	Bhavin Banik	2
□	313	Fateh Malhotra	2



```sql
# 4.Which areas have the highest average customer ratings?
SELECT c.area, AVG(f.rating) AS avg_rating
        FROM Feedback f
        JOIN Customers c ON f.customer_id = c.customer_id
        GROUP BY c.area
        ORDER BY avg_rating DESC;
```

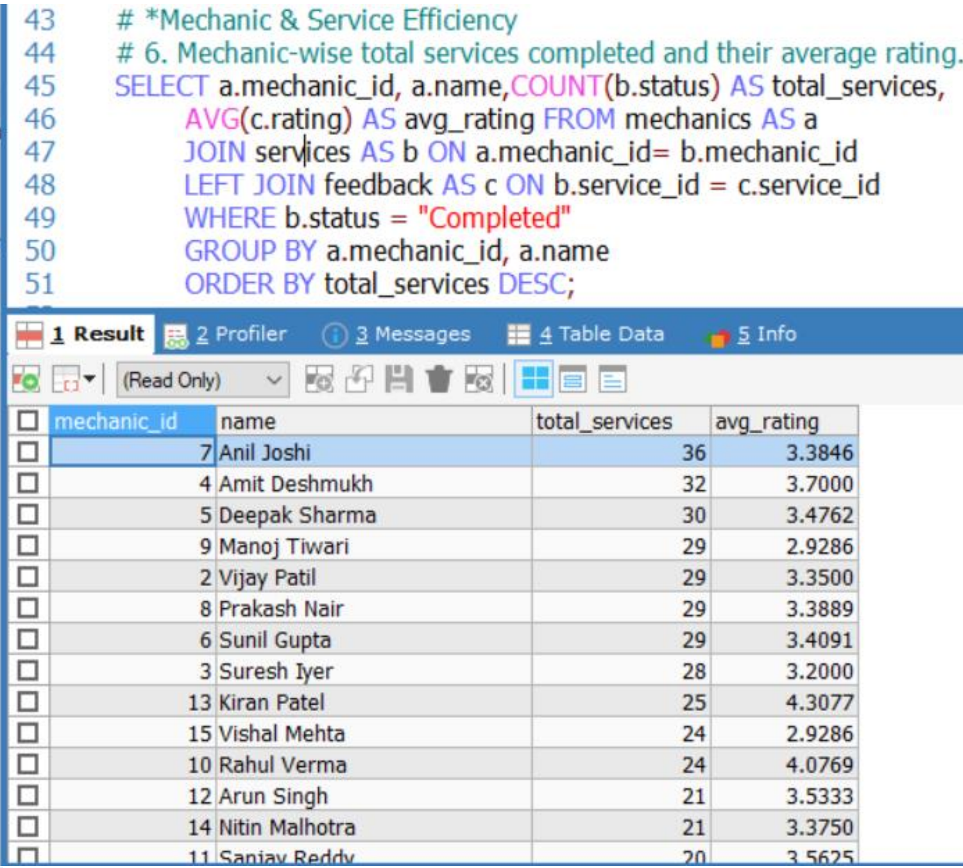
□	area	avg_rating
□	Sitabuldi	3.7031
□	Trimurti Nagar	3.5965
□	Hingna	3.5636
□	Civil Lines	3.5616
□	Itwari	3.4737
□	Koradi Road	3.4182
□	Hudkeshwar	3.4118
□	Wardha Road	3.3393
□	Manish Nagar	3.2692
□	Dharampeth	3.2093

Mechanic & Service Efficiency



```
# 5. Count of services per customer segmented by area.
SELECT a.area, a.customer_id, a.name, COUNT(c.service_id)
        AS services FROM customers AS a
        JOIN bikes AS b ON a.customer_id=b.customer_id
        JOIN services AS c ON b.bike_id=c.bike_id
        GROUP BY a.area, a.customer_id, a.name
        ORDER BY a.area;
```

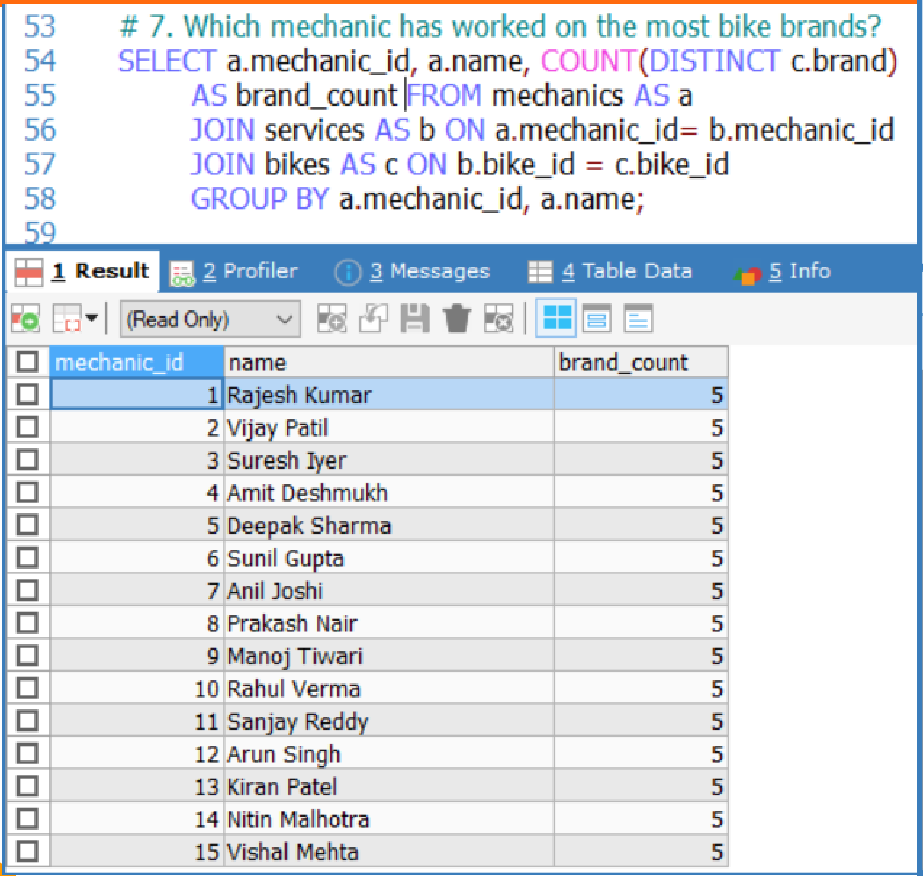
area	customer_id	name	services
Civil Lines	3	Taimur Majumdar	7
Civil Lines	17	Fateh Chadha	4
Civil Lines	59	Abram Batta	1
Civil Lines	69	Hunar Chad	2
Civil Lines	81	Anahi Manda	2
Civil Lines	98	Advik Thakkar	3
Civil Lines	105	Armaan Bali	6
Civil Lines	135	Riaan Saraf	1
Civil Lines	150	Yashvi Hans	1
Civil Lines	169	Ayesha Krishna	1
Civil Lines	175	Kiaan Kala	5
Civil Lines	179	Kaira Sahota	2
Civil Lines	195	Amira Keer	1
Civil Lines	207	Yuvaan Samra	5



```
# *Mechanic & Service Efficiency
# 6. Mechanic-wise total services completed and their average rating.
SELECT a.mechanic_id, a.name,COUNT(b.status) AS total_services,
    AVG(c.rating) AS avg_rating FROM mechanics AS a
    JOIN services AS b ON a.mechanic_id= b.mechanic_id
    LEFT JOIN feedback AS c ON b.service_id = c.service_id
    WHERE b.status = "Completed"
    GROUP BY a.mechanic_id, a.name
    ORDER BY total_services DESC;
```

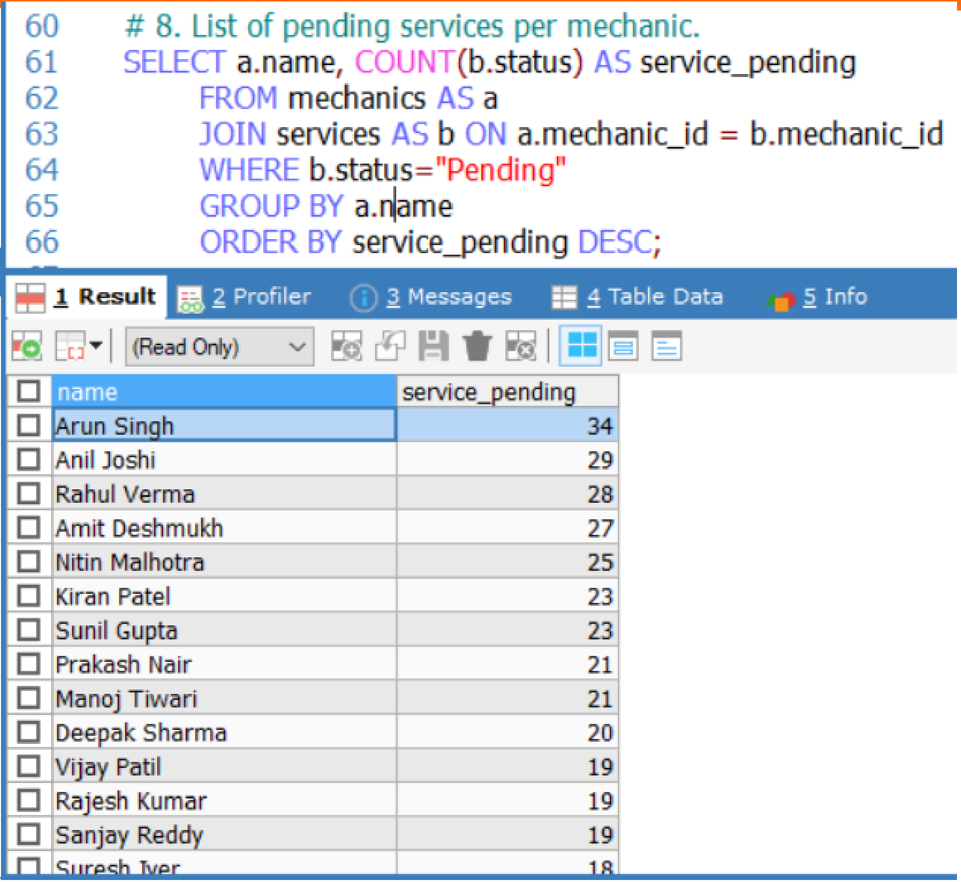
mechanic_id	name	total_services	avg_rating
7	Anil Joshi	36	3.3846
4	Amit Deshmukh	32	3.7000
5	Deepak Sharma	30	3.4762
9	Manoj Tiwari	29	2.9286
2	Vijay Patil	29	3.3500
8	Prakash Nair	29	3.3889
6	Sunil Gupta	29	3.4091
3	Suresh Iyer	28	3.2000
13	Kiran Patel	25	4.3077
15	Vishal Mehta	24	2.9286
10	Rahul Verma	24	4.0769
12	Arun Singh	21	3.5333
14	Nitin Malhotra	21	3.3750
11	Sanjay Reddy	20	3.5625

Mechanic & Service Efficiency



```
# 7. Which mechanic has worked on the most bike brands?
SELECT a.mechanic_id, a.name, COUNT(DISTINCT c.brand)
        AS brand_count FROM mechanics AS a
        JOIN services AS b ON a.mechanic_id= b.mechanic_id
        JOIN bikes AS c ON b.bike_id = c.bike_id
        GROUP BY a.mechanic_id, a.name;
```

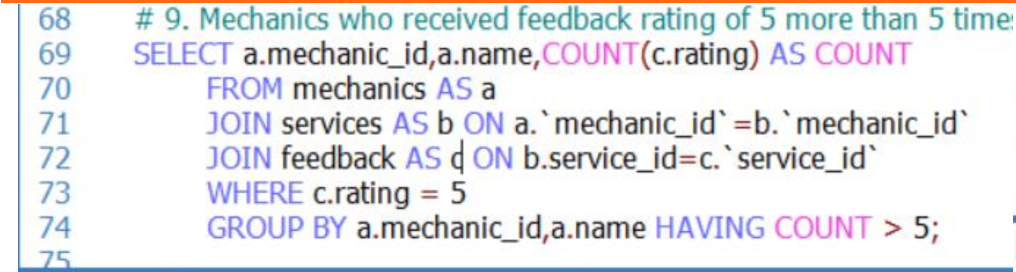
mechanic_id	name	brand_count
1	Rajesh Kumar	5
2	Vijay Patil	5
3	Suresh Iyer	5
4	Amit Deshmukh	5
5	Deepak Sharma	5
6	Sunil Gupta	5
7	Anil Joshi	5
8	Prakash Nair	5
9	Manoj Tiwari	5
10	Rahul Verma	5
11	Sanjay Reddy	5
12	Arun Singh	5
13	Kiran Patel	5
14	Nitin Malhotra	5
15	Vishal Mehta	5



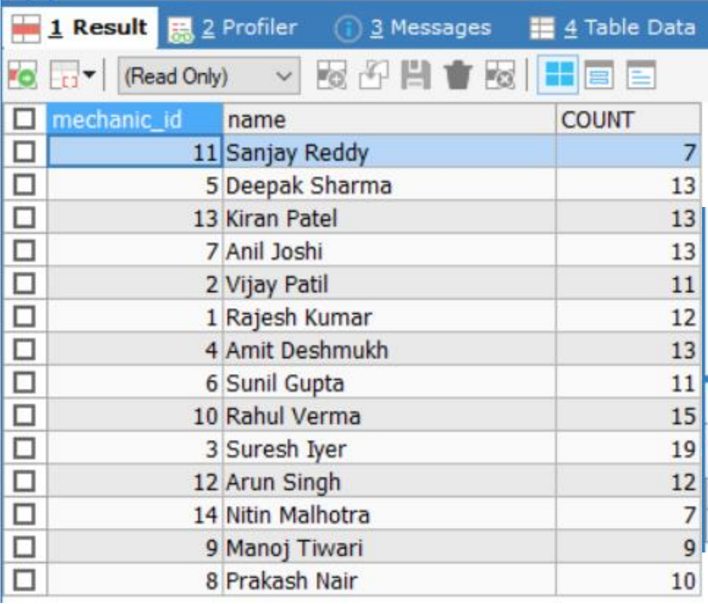
```
# 8. List of pending services per mechanic.
SELECT a.name, COUNT(b.status) AS service_pending
        FROM mechanics AS a
        JOIN services AS b ON a.mechanic_id = b.mechanic_id
        WHERE b.status="Pending"
        GROUP BY a.name
        ORDER BY service_pending DESC;
```

name	service_pending
Arun Singh	34
Anil Joshi	29
Rahul Verma	28
Amit Deshmukh	27
Nitin Malhotra	25
Kiran Patel	23
Sunil Gupta	23
Prakash Nair	21
Manoj Tiwari	21
Deepak Sharma	20
Vijay Patil	19
Rajesh Kumar	19
Sanjay Reddy	19
Suresh Iyer	18

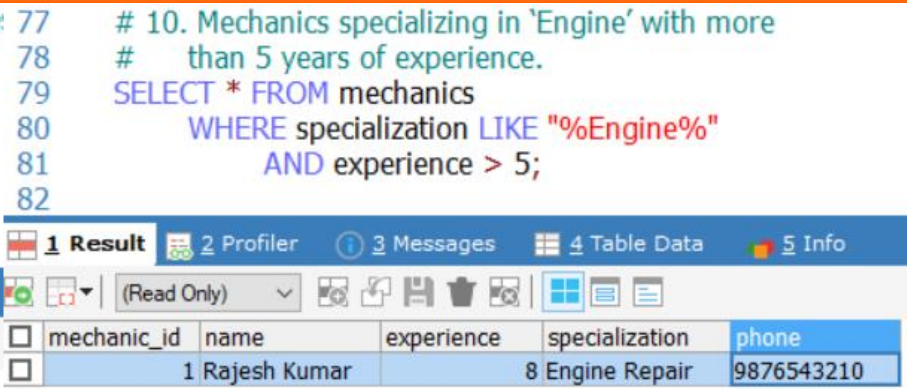
Mechanic & Service Efficiency



```sql
# 9. Mechanics who received feedback rating of 5 more than 5 time
SELECT a.mechanic_id,a.name,COUNT(c.rating) AS COUNT
        FROM mechanics AS a
        JOIN services AS b ON a.`mechanic_id`=b.`mechanic_id`
        JOIN feedback AS c ON b.service_id=c.`service_id`
        WHERE c.rating = 5
        GROUP BY a.mechanic_id,a.name HAVING COUNT > 5;
```

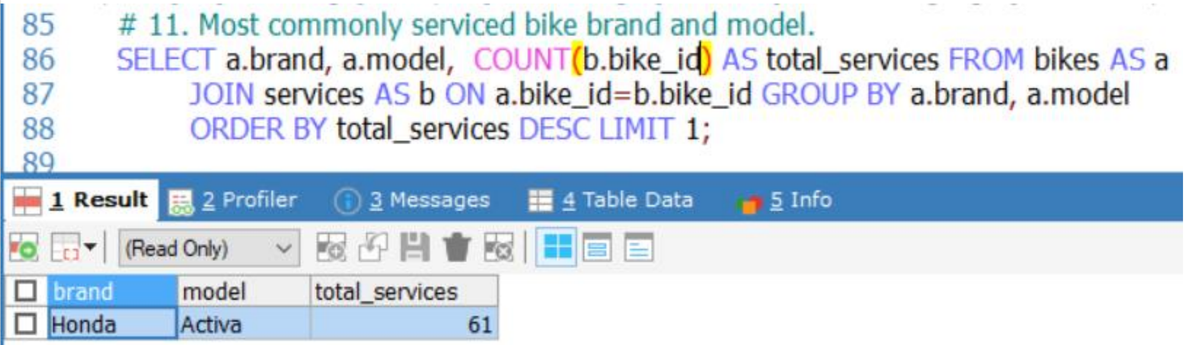


mechanic_id	name	COUNT
11	Sanjay Reddy	7
5	Deepak Sharma	13
13	Kiran Patel	13
7	Anil Joshi	13
2	Vijay Patil	11
1	Rajesh Kumar	12
4	Amit Deshmukh	13
6	Sunil Gupta	11
10	Rahul Verma	15
3	Suresh Iyer	19
12	Arun Singh	12
14	Nitin Malhotra	7
9	Manoj Tiwari	9
8	Prakash Nair	10



```sql
# 10. Mechanics specializing in 'Engine' with more
#      than 5 years of experience.
SELECT * FROM mechanics
        WHERE specialization LIKE "%Engine%"
                AND experience > 5;
```

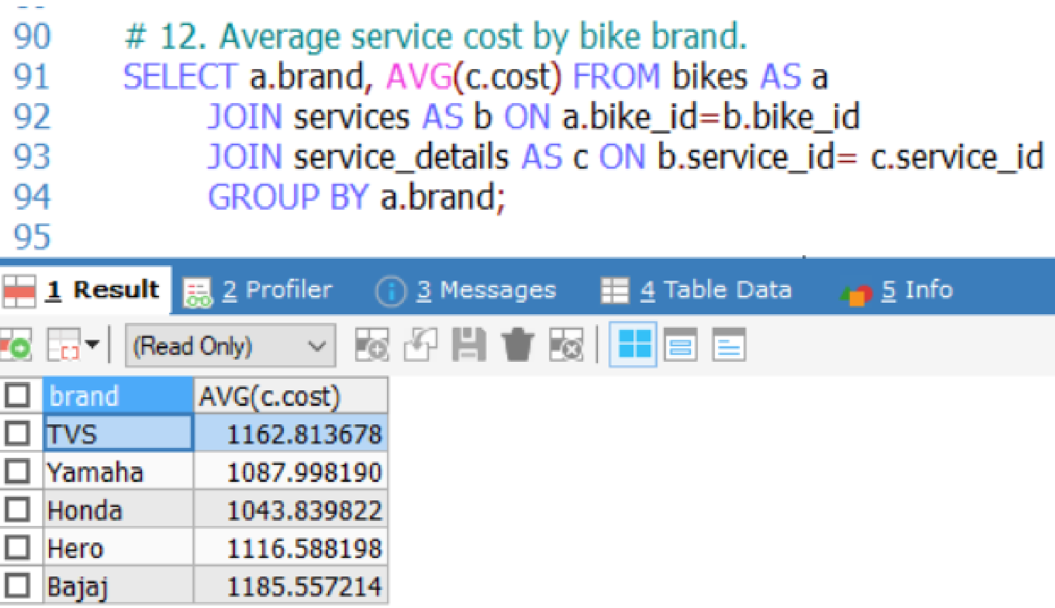
mechanic_id	name	experience	specialization	phone
1	Rajesh Kumar	8	Engine Repair	9876543210



```sql
# 11. Most commonly serviced bike brand and model.
SELECT a.brand, a.model,  COUNT(b.bike_id) AS total_services FROM bikes AS a
        JOIN services AS b ON a.bike_id=b.bike_id GROUP BY a.brand, a.model
        ORDER BY total_services DESC LIMIT 1;
```

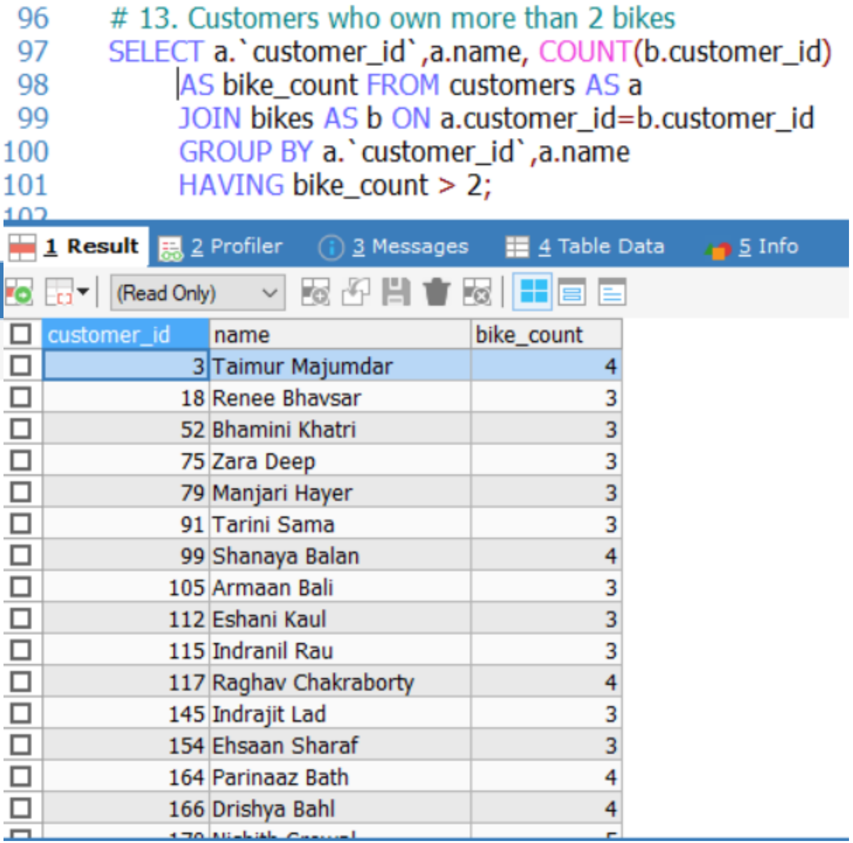
brand	model	total_services
Honda	Activa	61

Bike & Brand Analysis



```
# 12. Average service cost by bike brand.
SELECT a.brand, AVG(c.cost) FROM bikes AS a
        JOIN services AS b ON a.bike_id=b.bike_id
        JOIN service_details AS c ON b.service_id= c.service_id
        GROUP BY a.brand;
```

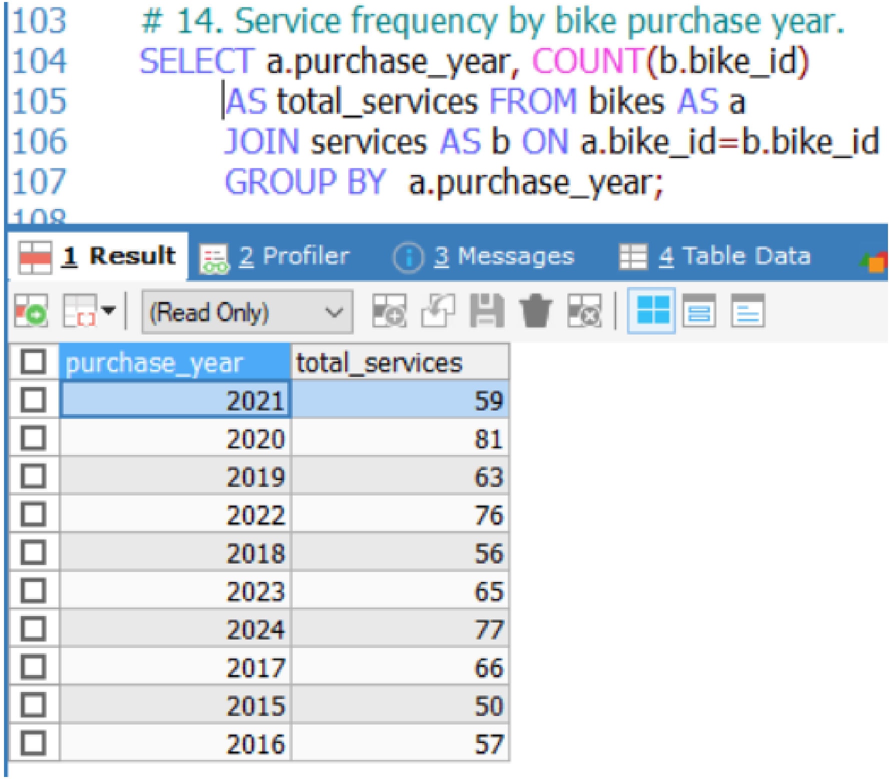
brand	AVG(c.cost)
TVS	1162.813678
Yamaha	1087.998190
Honda	1043.839822
Hero	1116.588198
Bajaj	1185.557214



```
# 13. Customers who own more than 2 bikes
SELECT a.`customer_id`,a.name, COUNT(b.customer_id)
    AS bike_count FROM customers AS a
    JOIN bikes AS b ON a.customer_id=b.customer_id
    GROUP BY a.`customer_id`,a.name
    HAVING bike_count > 2;
```

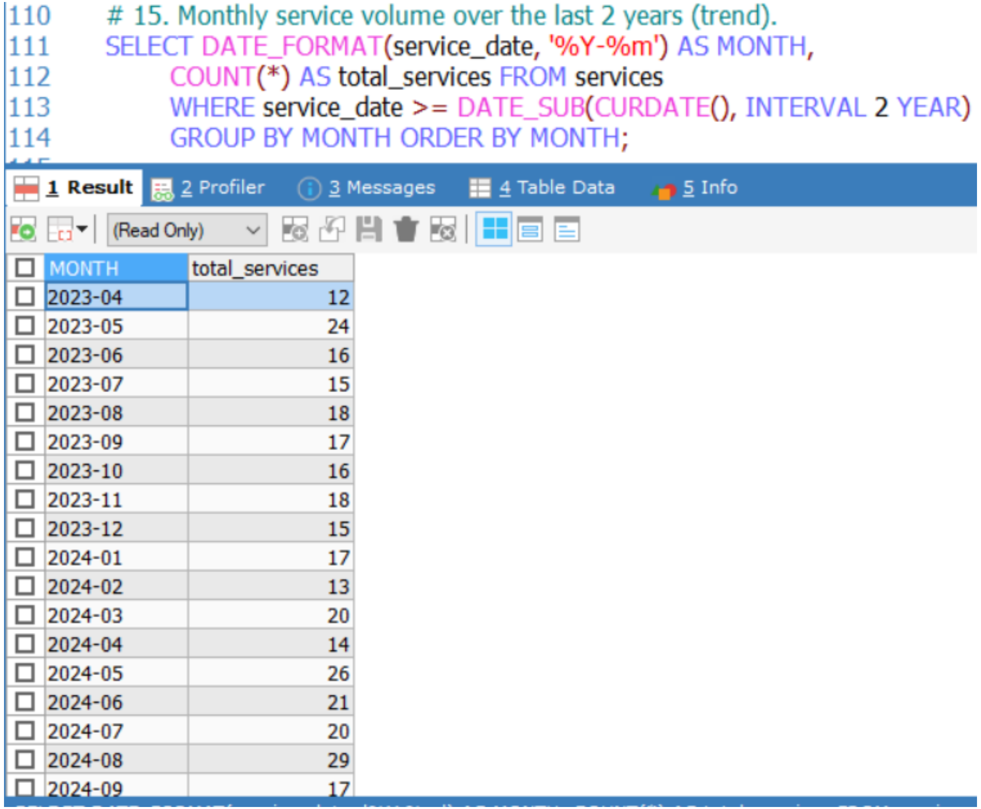
customer_id	name	bike_count
3	Taimur Majumdar	4
18	Renee Bhavsar	3
52	Bhamini Khatri	3
75	Zara Deep	3
79	Manjari Hayer	3
91	Tarini Sama	3
99	Shanaya Balan	4
105	Armaan Bali	3
112	Eshani Kaul	3
115	Indranil Rau	3
117	Raghav Chakraborty	4
145	Indrajit Lad	3
154	Ehsaan Sharaf	3
164	Parinaaz Bath	4
166	Drishya Bahl	4
170	Nishith Grewal	5

Bike & Brand Analysis



```sql
# 14. Service frequency by bike purchase year.
SELECT a.purchase_year, COUNT(b.bike_id)
        AS total_services FROM bikes AS a
        JOIN services AS b ON a.bike_id=b.bike_id
        GROUP BY  a.purchase_year;
```

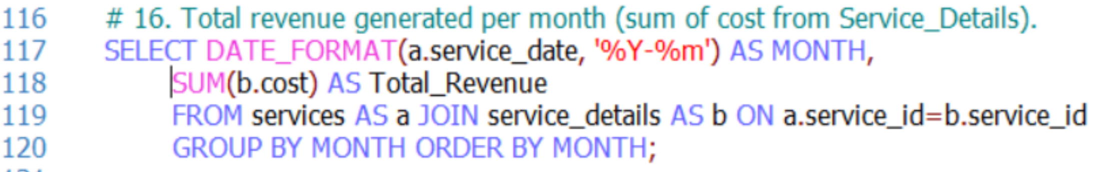
□	purchase_year	total_services
□	2021	59
□	2020	81
□	2019	63
□	2022	76
□	2018	56
□	2023	65
□	2024	77
□	2017	66
□	2015	50
□	2016	57



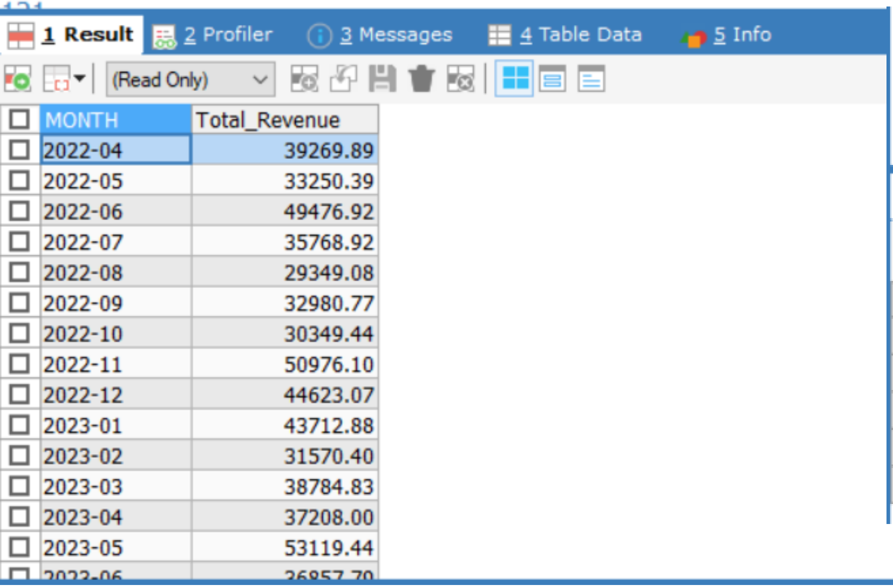
```sql
# 15. Monthly service volume over the last 2 years (trend).
SELECT DATE_FORMAT(service_date, '%Y-%m') AS MONTH,
       COUNT(*) AS total_services FROM services
       WHERE service_date >= DATE_SUB(CURDATE(), INTERVAL 2 YEAR)
       GROUP BY MONTH ORDER BY MONTH;
```

□	MONTH	total_services
□	2023-04	12
□	2023-05	24
□	2023-06	16
□	2023-07	15
□	2023-08	18
□	2023-09	17
□	2023-10	16
□	2023-11	18
□	2023-12	15
□	2024-01	17
□	2024-02	13
□	2024-03	20
□	2024-04	14
□	2024-05	26
□	2024-06	21
□	2024-07	20
□	2024-08	29
□	2024-09	17

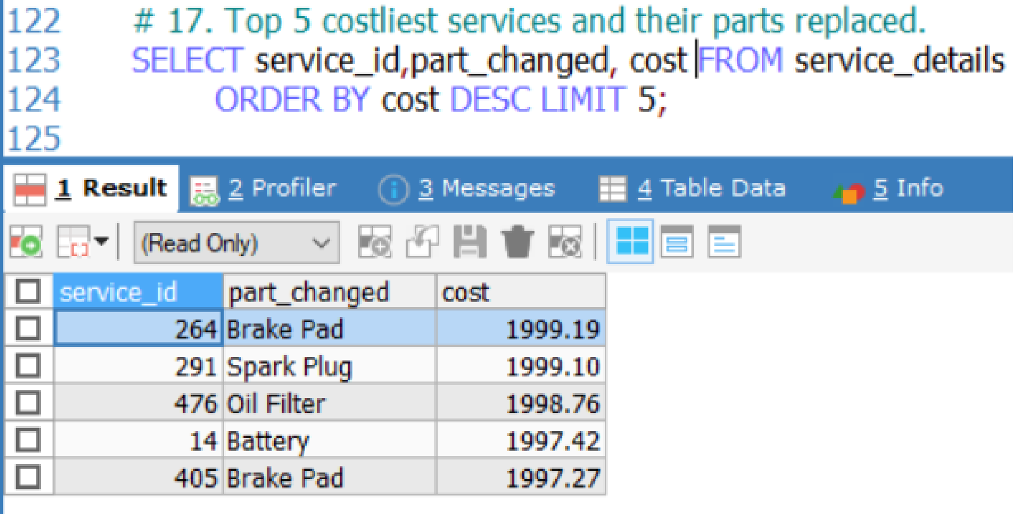
Service Volume & Costing



```
# 16. Total revenue generated per month (sum of cost from Service_Details).
SELECT DATE_FORMAT(a.service_date, '%Y-%m') AS MONTH,
    SUM(b.cost) AS Total_Revenue
    FROM services AS a JOIN service_details AS b ON a.service_id=b.service_id
    GROUP BY MONTH ORDER BY MONTH;
```



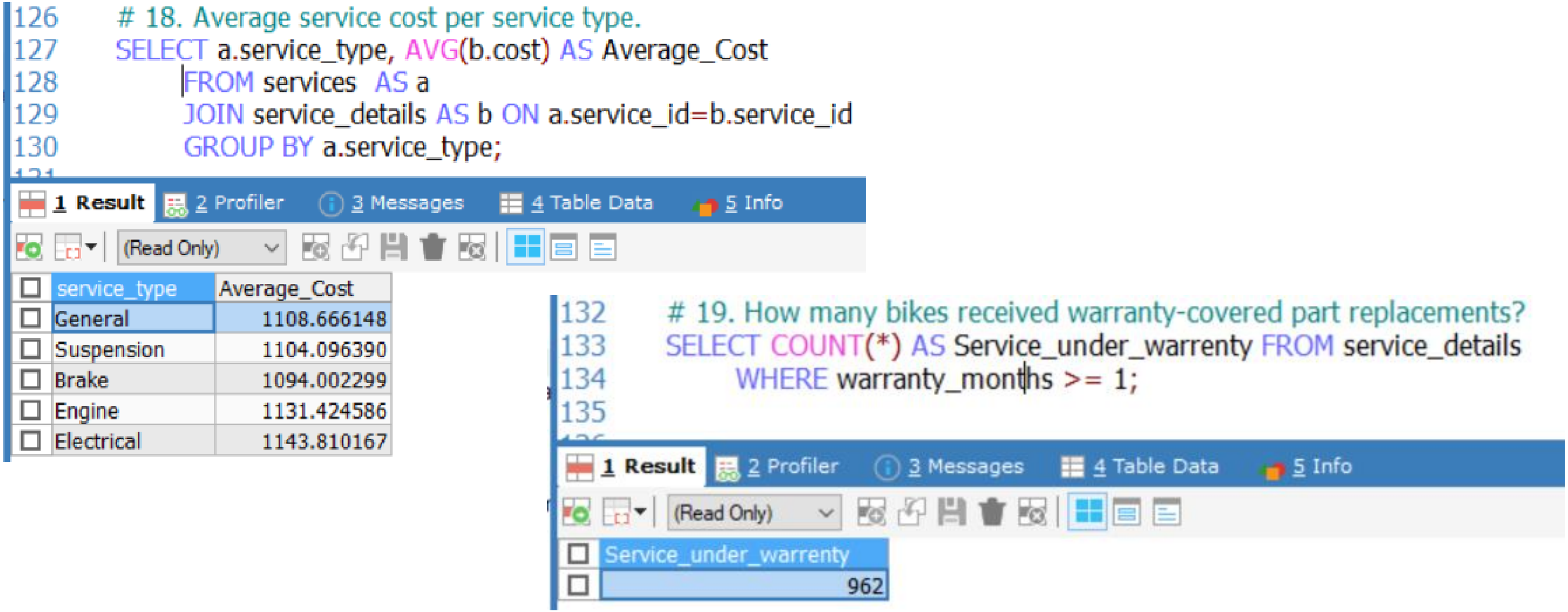
MONTH	Total_Revenue
2022-04	39269.89
2022-05	33250.39
2022-06	49476.92
2022-07	35768.92
2022-08	29349.08
2022-09	32980.77
2022-10	30349.44
2022-11	50976.10
2022-12	44623.07
2023-01	43712.88
2023-02	31570.40
2023-03	38784.83
2023-04	37208.00
2023-05	53119.44
2023-06	36857.70



```
# 17. Top 5 costliest services and their parts replaced.
SELECT service_id,part_changed, cost FROM service_details
    ORDER BY cost DESC LIMIT 5;
```

service_id	part_changed	cost
264	Brake Pad	1999.19
291	Spark Plug	1999.10
476	Oil Filter	1998.76
14	Battery	1997.42
405	Brake Pad	1997.27

Service Volume & Costing



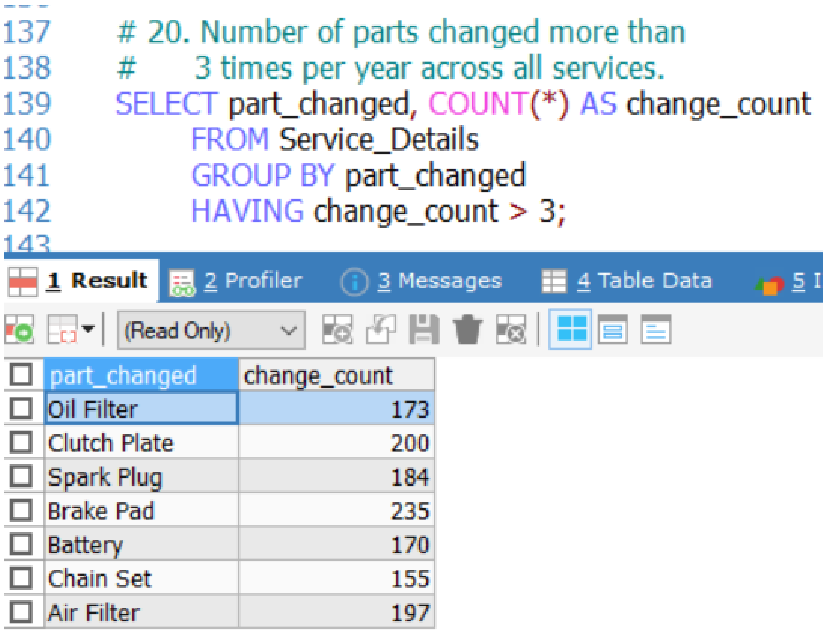
```
# 18. Average service cost per service type.
SELECT a.service_type, AVG(b.cost) AS Average_Cost
        FROM services  AS a
        JOIN service_details AS b ON a.service_id=b.service_id
        GROUP BY a.service_type;
```

□	service_type	Average_Cost
□	General	1108.666148
□	Suspension	1104.096390
□	Brake	1094.002299
□	Engine	1131.424586
□	Electrical	1143.810167

```
# 19. How many bikes received warranty-covered part replacements?
SELECT COUNT(*) AS Service_under_warrenty FROM service_details
        WHERE warranty_months >= 1;
```

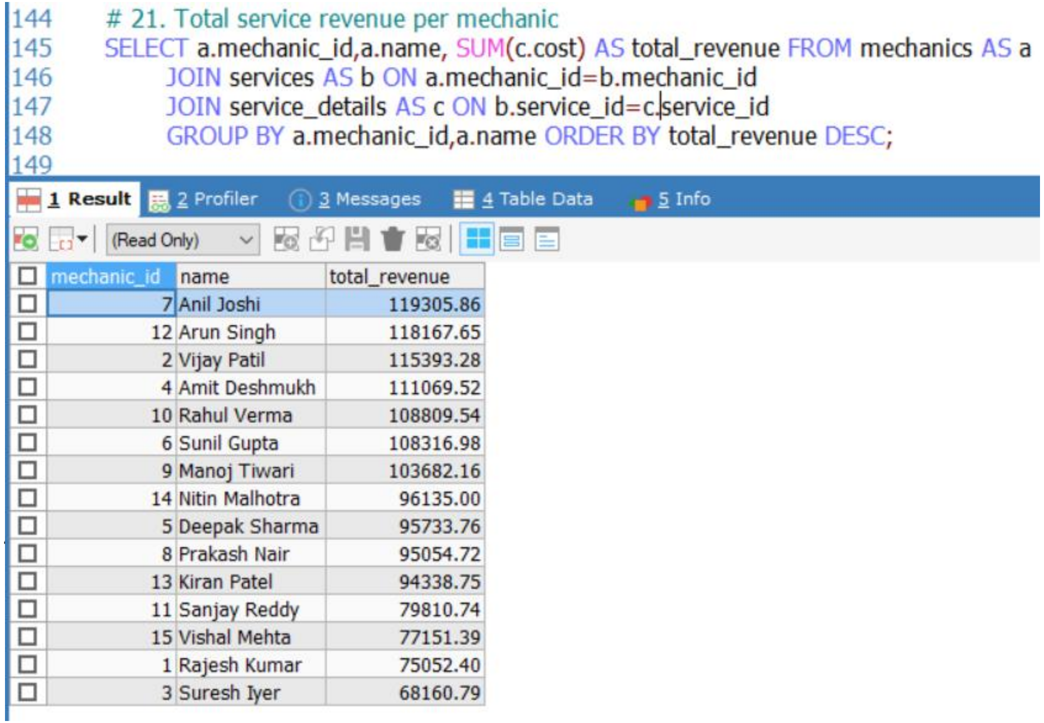
□	Service_under_warrenty
□	962

Service Volume & Costing



```
# 20. Number of parts changed more than
#     3 times per year across all services.
SELECT part_changed, COUNT(*) AS change_count
       FROM Service_Details
       GROUP BY part_changed
       HAVING change_count > 3;
```

□	part_changed	change_count
□	Oil Filter	173
□	Clutch Plate	200
□	Spark Plug	184
□	Brake Pad	235
□	Battery	170
□	Chain Set	155
□	Air Filter	197



```
# 21. Total service revenue per mechanic
SELECT a.mechanic_id,a.name, SUM(c.cost) AS total_revenue FROM mechanics AS a
      JOIN services AS b ON a.mechanic_id=b.mechanic_id
      JOIN service_details AS c ON b.service_id=c.service_id
      GROUP BY a.mechanic_id,a.name ORDER BY total_revenue DESC;
```

□	mechanic_id	name	total_revenue
□	7	Anil Joshi	119305.86
□	12	Arun Singh	118167.65
□	2	Vijay Patil	115393.28
□	4	Amit Deshmukh	111069.52
□	10	Rahul Verma	108809.54
□	6	Sunil Gupta	108316.98
□	9	Manoj Tiwari	103682.16
□	14	Nitin Malhotra	96135.00
□	5	Deepak Sharma	95733.76
□	8	Prakash Nair	95054.72
□	13	Kiran Patel	94338.75
□	11	Sanjay Reddy	79810.74
□	15	Vishal Mehta	77151.39
□	1	Rajesh Kumar	75052.40
□	3	Suresh Iyer	68160.79

Conclusion & Findings

Conclusion:

🔧 Service Efficiency Improvements

- Assign services based on mechanic specialization and availability.

📦 Parts Availability

- Track parts used frequently to restock on time.

💰 Revenue Tracking

- Focus more on high-revenue bike brands or models.

👷 Employee Performance

- Reward mechanics with high feedback ratings.
- Provide training to those with frequent pending services.

🌟 Quality Improvement (Customer Feedback)

- Introduce follow-up calls or messages after service.

🎯 Smart Offers & Discounts

- Offer loyalty discounts to regular customers.

Closure of Presentation

If you have any question ?

Thank You !