SQL PROJECT

ROYAL ENFIELD
SALES SQL
ANALYSIS

Presented by: Picnic Rautaray
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.

TABLE

- **bikes table**
- **sales table**
- **dealers table**
- **customer table**
- **feedback table**
- **servicerecords table**

1. Find the date of the first purchase for each customer.

```sql
select min(saledate) as first_purchase,concat(firstname,lastname) as fullname
from sales as s
join customers as c
on s.customerid = c.customerid
group by fullname
order by first_purchase;
```

Data Output	Messages	Notifications

	first_purchase date 🔒	fullname text 🔒
1	2023-01-15	AaravSharma
2	2023-02-22	AanyaPatel
3	2023-03-10	AdityaVerma
4	2023-04-05	AdvaitSingh
5	2023-05-18	AhanaKumar
6	2023-06-02	AidenGupta
7	2023-07-09	AishaDas

2. Write a SQL query to find the top 3 most expensive bikes (based on price) for each year and model.

```sql
with price_rnk_cte as
(
select model,price, year,
RANK() OVER (PARTITION BY year,model order by price desc) as price_rnk
from bikes
group by year,model,price
order by price_rnk asc
)

select model,max(price) as max_price, year,price_rnk from price_rnk_cte
where price_rnk<=3
group by year,model,price,price_rnk
order by price_rnk asc,max_price desc;
```

	model character varying (50)	max_price integer	year integer	price_rnk bigint
1	Interceptor 650	345000	2023	1
2	Continental GT 650	340000	2023	1
3	Continental GT 650	320000	2022	1
4	Interceptor 650	300000	2022	1
5	Himalayan	290000	2023	1
6	Continental GT 535	280000	2022	1
7	Classic 500	255000	2023	1
8	Himalayan	240000	2022	1
9	Bullet 500	235000	2023	1

3. Retrieve the Latest Service for Each Bike.

```sql
select bikeid,max(servicedate) as latest_service,servicedescription,servicecost
from servicerecords
group by bikeid,servicedescription,servicecost
order by latest_service asc;
```

	bikeid integer	latest_service date	servicedescription character varying (255)	servicecost numeric (10,2)
1	1	2023-02-01	Regular Maintenance	5000.00
2	2	2023-03-15	Oil Change	3000.00
3	3	2023-04-10	Brake Inspection	2000.00
4	4	2023-05-05	Tire Replacement	6000.00
5	5	2023-06-18	Chain Adjustment	1500.00
6	6	2023-07-02	Spark Plug Replacement	1000.00
7	7	2023-08-09	Coolant Flush	2500.00
8	8	2023-09-14	Air Filter Replacement	1200.00
9	9	2023-10-20	Battery Check	800.00
10	10	2023-11-25	Suspension Tuning	4000.00
11	11	2024-01-01	Regular Maintenance	5000.00
12	12	2024-02-15	Oil Change	3000.00

4. Find the Price difference between the current bike and the next bike in the same year.

```sql
select bikeid,model,year,price,
LEAD(price) over(partition by year order by price) - price as price_diff
from bikes;
```

bikeid [PK] integer	model character varying (50)	year integer	price integer	price_diff integer
1	Classic 350	2022	180000	10000
8	Meteor 350	2022	190000	10000
3	Thunderbird 350	2022	200000	10000
9	Interceptor 350	2022	210000	10000
2	Bullet 500	2022	220000	10000
7	Classic 500	2022	230000	10000
4	Himalayan	2022	240000	40000
10	Continental GT 535	2022	280000	20000
5	Interceptor 650	2022	300000	20000
6	Continental GT 650	2022	320000	[null]
12	Bullet 350	2023	175000	5000
18	Bullet 350	2023	180000	5000

5. Find the Maximum sale amount for each month.

```sql
select extract(month from saledate) as month,
extract(year from saledate) as year,
max(saleamount) as maxsale_amt
from sales
group by month,year
order by year;
```

	month numeric	year numeric	maxsale_amt numeric
1	1	2023	180000.00
2	2	2023	200000.00
3	3	2023	300000.00
4	4	2023	230000.00
5	5	2023	240000.00
6	6	2023	280000.00
7	7	2023	200000.00
8	8	2023	190000.00
9	9	2023	210000.00
10	10	2023	185000.00
11	11	2023	175000.00
12	12	2023	250000.00

6. Concatenate the first and last names of customers, and display them in upper case.

```sql
select upper(concat(firstname,' ',lastname))as fullname
from customers;
```

	fullname text
1	AARAV SHARMA
2	AANYA PATEL
3	ADITYA VERMA
4	ADVAIT SINGH
5	AHANA KUMAR
6	AIDEN GUPTA
7	AISHA DAS
8	AKSHAY CHATTERJEE
9	ALIA MUKHERJEE
10	ANAYA JOSHI
11	YASH SINHA
12	ZARA NAIR

7. Determine the quarter in which each sale occurred.

```sql
select saleid,extract(quarter from saledate) as quarter, saledate
from sales
group by saleid
order by saledate;
```

	saleid [PK] integer	quarter numeric	saledate date
1	1	1	2023-01-15
2	2	1	2023-02-22
3	3	1	2023-03-10
4	4	2	2023-04-05
5	5	2	2023-05-18
6	6	2	2023-06-02
7	7	3	2023-07-09
8	8	3	2023-08-14
9	9	3	2023-09-20
10	10	4	2023-10-25

8. Calculate the running total of service costs for each bike.

```sql
select bikeid,servicedate,servicecost,
sum(servicecost) over (partition by bikeid order by servicedate) as runningtotal_cost
from servicerecords
```

	bikeid\ninteger	servicedate\ndate	servicecost\nnumeric (10,2)	runningtotal_cost\nnumeric
1	1	2023-02-01	5000.00	5000.00
2	2	2023-03-15	3000.00	3000.00
3	3	2023-04-10	2000.00	2000.00
4	4	2023-05-05	6000.00	6000.00
5	5	2023-06-18	1500.00	1500.00
6	6	2023-07-02	1000.00	1000.00
7	7	2023-08-09	2500.00	2500.00
8	8	2023-09-14	1200.00	1200.00
9	9	2023-10-20	800.00	800.00
10	10	2023-11-25	4000.00	4000.00

9. Find the top dealers based on the total sales amount across the bikes.

```sql
select s.bikeid,d.dealername,
sum(s.saleamount) over (partition by d.dealerid) as totalsale_amt
from dealers as d
join sales as s
on d.dealerid = s.dealerid
```

	bikeid integer 🔒	dealername character varying (255) 🔒	totalsale_amt numeric 🔒
1	1	Royal Motors	180000.00
2	3	Classic Bikes	200000.00
3	5	Thunder Motors	300000.00
4	7	Himalayan Cycles	230000.00
5	9	Bullet Riders	240000.00
6	11	Interceptor Bikes	280000.00
7	13	Classic Motors	200000.00
8	15	Vintage Bikes	190000.00
9	17	Royal Wheels	210000.00
10	19	Enfield Paradise	185000.00

10 . Find the count of bikes sold each year and categorize them into three groups: 'Low',' Medium', and 'High' based on their prices.

```sql
SELECT
    EXTRACT(year FROM s.saledate) AS year,
    COUNT(b.bikeid) AS total_bike_sold,
    SUM(CASE WHEN b.price < 200000 THEN 1 ELSE 0 END) AS "Low",
    SUM(CASE WHEN b.price >= 200000 AND b.price <= 300000 THEN 1 ELSE 0 END) AS "Medium",
    SUM(CASE WHEN b.price > 300000 THEN 1 ELSE 0 END) AS "High"
FROM
    sales AS s
JOIN
    bikes AS b ON s.bikeid = b.bikeid
GROUP BY EXTRACT(year FROM s.saledate)
order by year;
```

	year numeric	total_bike_sold bigint	Low bigint	Medium bigint	High bigint
1	2023	12	4	7	1
2	2024	12	0	9	3
3	2025	12	4	6	2
4	2026	12	3	7	2
5	2027	12	3	8	1
6	2028	12	1	8	3
7	2029	12	4	6	2
8	2030	12	2	9	1

11 . Find the Top 5 Bike models with the highest cost.

```sql
select model, max(price) as highest_cost
from bikes
group by model
order by highest_cost
limit 5;
```

	model character varying (50) 🔒	highest_cost integer 🔒
1	Bullet 350	200000
2	Thunderbird 350	210000
3	Classic 350	215000
4	Meteor 350	220000
5	Interceptor 350	220000

12 . Write a query to compare a bike model price in years 2022 and 2023. retrieve in two different column for 2022 and 2023.

```sql
select model,
    max(case when year = 2022 then price end) as price_2022,
    max(case when year = 2023 then price end) as price_2023
from bikes
group by model;
```

	model character varying (50)	price_2022 integer	price_2023 integer
1	Interceptor 650	300000	345000
2	Himalayan	240000	290000
3	Thunderbird 350	200000	210000
4	Continental GT 535	280000	[null]
5	Meteor 350	190000	220000
6	Continental GT 650	320000	340000
7	Interceptor 350	210000	220000
8	Classic 350	180000	215000
9	Classic 500	230000	255000
10	Bullet 350	[null]	200000
11	Bullet 500	220000	235000

13 . Retrieve the count of highly sold bike model in both the year with its SaleAmount.

```
select count(*) as count,model,sum(price) as totalsale_amt from bikes
group by model
order by count desc;
```

	count bigint	model character varying (50)	totalsale_amt bigint
1	26	Meteor 350	5550000
2	24	Himalayan	6450000
3	24	Classic 500	5867000
4	24	Classic 350	4918000
5	23	Interceptor 650	7615000
6	23	Continental GT 650	7420000
7	22	Thunderbird 350	4494000
8	13	Bullet 350	2506000
9	13	Bullet 500	2935000
10	3	Interceptor 350	645000

14 . Write a query to retrieve how much bikes arw sold by each dealer in year 2023(dealer name, bike sales count, total sales amount)

```sql
SELECT
    d.dealername,
    COUNT(s.bikeid) AS bikesalecount,
    SUM(s.saleamount) AS totalsale_amt
FROM sales AS s
JOIN
bikes AS b ON s.bikeid = b.bikeid
JOIN
dealers AS d ON s.dealerid = d.dealerid
WHERE EXTRACT(year FROM s.saledate) = 2023
GROUP BY d.dealername;
```

	dealername character varying (255)	bikesalecount bigint	totalsale_amt numeric
1	Bullet Riders	1	240000.00
2	Classic Bikes	1	200000.00
3	Classic Motors	1	200000.00
4	Eagle Motors	1	175000.00
5	Enfield Paradise	1	185000.00
6	Golden Bikes	1	250000.00
7	Himalayan Cycles	1	230000.00
8	Interceptor Bikes	1	280000.00
9	Royal Motors	1	180000.00

15 . From the above Dealers table Retrieve the count of dealers in each location.

```
select count(*) as dealercount,location
from dealers
group by location;
```

	dealercount 🔒 bigint	location 🔒 character varying (255)
1	2	Jaipur
2	2	Chitradurga
3	1	Gwalior
4	1	Kochi
5	2	Davanagere
6	1	Navsari
7	1	Mathura
8	1	Karwar
9	1	Bhagalpur

16 . Retrieve the top 5 models from the bikes table and the max service cost of each bike with its description from servicerecord table.

```sql
select b.model,max(s.servicecost) as maxservicecost, s.servicedescription
from bikes as b
join servicerecords as s
on b.bikeid = s.bikeid
group by b.model,s.servicedescription
order by maxservicecost desc
limit 5;
```

	model character varying (50) 🔒	maxservicecost numeric 🔒	servicedescription character varying (255) 🔒
1	Meteor 350	6000.00	Tire Replacement
2	Bullet 350	6000.00	Tire Replacement
3	Himalayan	6000.00	Tire Replacement
4	Classic 350	6000.00	Tire Replacement
5	Classic 500	6000.00	Tire Replacement