

KTM Sales Website

Introduction to KTM Sales Website:

Welcome to the KTM Sales Portal – your one-stop destination for exploring, purchasing, and managing KTM motorcycles. Designed with both customers and dealerships in mind, this platform offers a seamless experience for browsing KTM models, tracking sales, managing payments, and monitoring deliveries.

KTM is known worldwide for its **performance-driven motorcycles**, from stylish **naked street bikes** like the Duke series to **sporty RC models**, and robust **Adventure and Enduro machines**. Our sales website brings all of this together into a **digital showroom**, making it easier than ever for customers to connect with authorized dealers, compare models, and finalize their dream ride.

KTM, an Austrian motorcycle manufacturer, has become a global leader in the **premium sports bike segment**. With its aggressive styling, advanced engineering, and powerful performance, KTM has built a strong reputation among **motorcycle enthusiasts, racers, and adventure riders**. In countries like India, KTM bikes such as the **Duke, RC, and Adventure series** have witnessed exponential growth in demand due to their affordability, reliability, and superior performance.

Key Highlights:

- **Explore KTM Models:** View detailed specifications, features, and pricing of all KTM motorcycles.
- **Customer Dashboard:** Track your purchases, delivery status, and payment history in one place.
- **Dealership Management:** Sales shops and agents can log in to manage customer orders, payments, and deliveries.
- **Secure Payments:** Multiple options like UPI, Credit/Debit Cards, Net Banking, and Cash.
- **Real-Time Updates:** Stay informed about delivery status, payment confirmations, and offers.

Queries for the KTM Sales Database:

1. List all customers from Maharashtra-

Problem Statement: KTM wants to know how many customers are from *Maharashtra* to analyze sales in that state.

```
SELECT * FROM customers  
WHERE state = 'Maharashtra';
```

Output:-

customer_id	name	phone	email	city	state
1	Rajesh Kumar	9876543210	rajesh.kumar@email.com	Mumbai	Maharashtra
2	Neha Sharma	9123456780	neha.sharma@email.com	Pune	Maharashtra
12	Pooja Desai	9784512365	pooja.d@email.com	Nagpur	Maharashtra
28	Sunita Sharma	9976543211	sunita.s@email.com	Nagpur	Maharashtra
44	Sonal Deshmukh	9967891234	sonal.d@email.com	Pune	Maharashtra
NULL	NULL	NULL	NULL	NULL	NULL

2. Find total Sales per month -

Problem Statement: KTM wants to analyze *monthly sales trends* to understand peak and low seasons.

```
SELECT DATE_FORMAT(s.sale_date, '%Y-%m') AS month, SUM(s.total_amount) AS  
total_sales  
FROM sales s  
GROUP BY month  
ORDER BY month;
```

Output :-

month	total_sales
2023-01	1250000.00
2023-02	6580000.00
2023-03	9670000.00
2023-04	14370000.00
2023-05	2950000.00

3. List all models launched after 2020 -

Problem Statement: KTM wants to check the list of *new models released after 2020* to study their sales performance.

```
SELECT * FROM models  
WHERE launch_year > 2020;
```

Output:-

	model_id	model_name	engine_cc	category	launch_year	base_price
▶	11	KTM Adventure 890	890	Adventure	2021	1200000.00
	12	KTM RC 8C	889	SuperSports	2021	1380000.00
	14	KTM 890 Duke R	890	Naked	2021	950000.00
*	16	KTM 1290 Super Adventure R	1301	Adventure	2021	1800000.00
*	NUL	NUL	NUL	NUL	NUL	NUL

models 12 ×

4. Count the Number of sales per region -

Problem Statement: Management wants to see *which region has the highest sales* to focus marketing efforts.

```
SELECT region, COUNT(*) AS sales_count  
FROM sales  
GROUP BY region;
```

Output:-

	region	sales_count
▶	West	23
	North	11
	South	16

Result 10 ×

5. Find all sales of the KTM Duke 390 –

Problem Statement: Management needs to check how many *KTM Duke 390 bikes* were sold, along with customer and sales details.

```
SELECT s.sale_id, c.name AS customer_name, s.total_amount, s.sale_date  
FROM sales s  
JOIN models m ON s.model_id = m.model_id  
JOIN customers c ON s.customer_id = c.customer_id  
WHERE m.model_name = 'KTM Duke 390';
```

Output:-

Result Grid				
	sale_id	customer_name	total_amount	sale_date
▶	1	Rajesh Kumar	280000.00	2023-01-15
	21	Anjali Jain	280000.00	2023-03-15
	41	Megha Kapoor	280000.00	2023-04-27

Result 5 X Read Only

6. Show total sales per model –

Problem Statement: KTM wants to identify the *best-selling bike models* and compare their sales performance.

```
SELECT m.model_name, SUM(s.total_amount) AS total_sales  
FROM sales s  
JOIN models m ON s.model_id = m.model_id  
GROUP BY m.model_name  
ORDER BY total_sales DESC;
```

Output:-

model_name	total_sales
KTM 450 Rally Replica	5000000.00
KTM 1290 Super Adventure R	3600000.00
KTM Adventure 790	3450000.00
KTM 1290 Super Duke R	3200000.00
KTM RC 8C	2760000.00
KTM Adventure 890	2400000.00
KTM 890 Duke R	1900000.00
KTM SX-F 450	1800000.00
KTM 790 Duke	1760000.00
KTM EXC 300 TPI	1440000.00
KTM Duke 125	960000.00
KTM Adventure 390	930000.00
KTM RC 390	900000.00
KTM Duke 390	840000.00
KTM Adventure 250	780000.00
KTM Freeride E-XC	700000.00
KTM Duke 250	690000.00
KTM RC 200	670000.00

7. List all pending deliveries -

Problem Statement: The delivery team needs to track all *orders that are still pending* for completion.

```
SELECT d.delivery_id, s.sale_id, c.name AS customer_name, d.delivery_status  
FROM deliveries d  
JOIN sales s ON d.sale_id = s.sale_id  
JOIN customers c ON s.customer_id = c.customer_id  
WHERE d.delivery_status = 'Pending';
```

Output:-

delivery_id	sale_id	customer_name	delivery_status
3	3	Amit Verma	Pending
7	7	Ravi Reddy	Pending
10	10	Sneha Kapoor	Pending
13	13	Kabir Khan	Pending
16	16	Farhan Sheikh	Pending
19	19	Lalit Yadav	Pending
22	22	Karan Thakur	Pending
25	25	Geeta Pillai	Pending
28	28	Sunita Sharma	Pending
31	31	Manoj Rao	Pending
34	34	Maya Nair	Pending
37	37	Harsh Vardhan	Pending
40	40	Naveen Reddy	Pending
43	43	Chirag Modi	Pending
46	46	Komal Yadav	Pending
49	49	Gaurav Mishra	Pending

Result 7 x

8. Show all unpaid payments –

Problem Statement: The finance department wants to know which *payments are still unpaid* to follow up with customers.

```
SELECT p.payment_id, c.name AS customer_name, p.amount, p.payment_status,  
p.transaction_status  
FROM payments p  
JOIN sales s ON p.sale_id = s.sale_id  
JOIN customers c ON s.customer_id = c.customer_id  
WHERE p.payment_status = 'Unpaid';
```

Output:-

Result Grid					
	payment_id	customer_name	amount	payment_status	transaction_status
▶	3	Amit Verma	310000.00	Unpaid	Pending
	7	Ravi Reddy	230000.00	Unpaid	Pending
	10	Sneha Kapoor	210000.00	Unpaid	Pending
	14	Kavita Joshi	950000.00	Unpaid	Pending
	17	Meera Menon	2500000.00	Unpaid	Pending
	22	Karan Thakur	190000.00	Unpaid	Pending
	25	Geeta Pillai	170000.00	Unpaid	Pending
	29	Vivek Chawla	1150000.00	Unpaid	Pending
	32	Ritu Bansal	1200000.00	Unpaid	Pending
	36	Tanvi Arora	1800000.00	Unpaid	Pending
	40	Naveen Reddy	720000.00	Unpaid	Pending
	43	Chirag Modi	310000.00	Unpaid	Pending
	46	Komal Yadav	320000.00	Unpaid	Pending
	49	Gaurav Mishra	1150000.00	Unpaid	Pending

Result 8 ×

9. Find the top 5 customers by total purchase –

Problem Statement: KTM wants to identify its *top 5 loyal/high-spending customers* for rewards or offers.

```
SELECT c.name, SUM(s.total_amount) AS total_purchase  
FROM sales s  
JOIN customers c ON s.customer_id = c.customer_id  
GROUP BY c.name  
ORDER BY total_purchase DESC  
LIMIT 5;
```

Output:-

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:	Print
	name	total_purchase				
▶	Harsh Vardhan	2500000.00				
	Meera Menon	2500000.00				
	Farhan Sheikh	1800000.00				
	Tanvi Arora	1800000.00				
	Arjun Menon	1600000.00				

10. Show Total Payment received per payment method -

Problem Statement: Finance wants to analyze *which payment methods (Cash, UPI, Card, etc.) are most used by customers*.

```
SELECT payment_method, SUM(amount) AS total_received  
FROM payments  
WHERE payment_status = 'Paid'  
GROUP BY payment_method;
```

Output:-

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Print
	payment_method	total_received			
▶	Credit Card	5420000.00			
	UPI	7300000.00			
	Debit Card	7000000.00			
	Cash	520000.00			
	Net Banking	3370000.00			

11. Find Sales with quantity greater than 1 –

Problem Statement: KTM wants to track *bulk purchases* where customers bought more than one bike in a single order.

```

SELECT s.sale_id, c.name AS customer_name, m.model_name, s.quantity,
s.total_amount
FROM sales s
JOIN customers c ON s.customer_id = c.customer_id
JOIN models m ON s.model_id = m.model_id
WHERE s.quantity > 1;

```

Output:-

		Result Grid	Filter Rows:	Export:	Wrap Cell Content:
	sale_id	customer_name	model_name	quantity	total_amount
▶	6	Anita Mehta	KTM Duke 125	2	320000.00
	26	Rakesh Tiwari	KTM Duke 125	2	320000.00
	46	Komal Yadav	KTM Duke 125	2	320000.00

Result 14 ×

12. Show Deliveries along with payment status –

Problem Statement: Management needs a combined report of *delivery progress and payment status* for each customer.

```

SELECT d.delivery_id, c.name AS customer_name, m.model_name, d.delivery_status,
p.payment_status
FROM deliveries d
JOIN sales s ON d.sale_id = s.sale_id
JOIN customers c ON s.customer_id = c.customer_id
JOIN models m ON s.model_id = m.model_id
JOIN payments p ON s.sale_id = p.sale_id;

```

Output:-

		Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	delivery_id	customer_name	model_name	delivery_status	payment_status	
▶	6	Anita Mehta	KTM Duke 125	Delivered	Paid	
	26	Rakesh Tiwari	KTM Duke 125	Delivered	Paid	
	46	Komal Yadav	KTM Duke 125	Pending	Unpaid	
	2	Neha Sharma	KTM Duke 200	Delivered	Paid	
	22	Karan Thakur	KTM Duke 200	Pending	Unpaid	
	42	Yashwant Singh	KTM Duke 200	Delivered	Paid	
	7	Ravi Reddy	KTM Duke 250	Pending	Unpaid	
	27	Alok Rathi	KTM Duke 250	Delivered	Paid	
	47	Sameer Khan	KTM Duke 250	Delivered	Paid	
	1	Rajesh Kumar	KTM Duke 390	Delivered	Paid	

Result 17 ×

13. List all customers with pending payments and deliveries –

Problem Statement: Management wants to identify customers who *haven't paid and also haven't received delivery*, so they can resolve issues quickly.

```
SELECT DISTINCT c.name, p.payment_status, d.delivery_status
FROM customers c
JOIN sales s ON c.customer_id = s.customer_id
JOIN payments p ON s.sale_id = p.sale_id
JOIN deliveries d ON s.sale_id = d.sale_id
WHERE p.payment_status = 'Unpaid' AND d.delivery_status = 'Pending';
```

Output:-

	name	payment_status	delivery_status
▶	Amit Verma	Unpaid	Pending
	Ravi Reddy	Unpaid	Pending
	Sneha Kapoor	Unpaid	Pending
	Karan Thakur	Unpaid	Pending
	Geeta Pillai	Unpaid	Pending
	Naveen Reddy	Unpaid	Pending
	Chirag Modi	Unpaid	Pending
	Komal Yadav	Unpaid	Pending
	Gaurav Mishra	Unpaid	Pending