

MySQL queries on Sales Data

Key Metadata and Structure

- **Rows:** 2,823
- **Columns:** 25
- **Column Names and Descriptions (based on available documentation; data types inferred as typical for such datasets—e.g., integers for IDs and quantities, floats for prices, strings for names and dates):**
 - **ORDERNUMBER:** Integer - Unique order identifier.
 - **QUANTITYORDERED:** Integer - Number of items ordered.
 - **PRICEEACH:** Float - Price per unit.
 - **ORDERLINENUMBER:** Integer - Line item number within the order.
 - **SALES:** Float - Total sales amount (**QUANTITYORDERED * PRICEEACH**).
 - **ORDERDATE:** String (date format like MM/DD/YYYY) - Date of the order.
 - **STATUS:** String - Order status (e.g., 'Shipped', 'Cancelled').
 - **QTR_ID:** Integer - Quarter of the year (1-4).
 - **MONTH_ID:** Integer - Month of the year (1-12).
 - **YEAR_ID:** Integer - Year of the order (e.g., 2003, 2004, 2005).
 - **PRODUCTLINE:** String - Category of product (e.g., 'Classic Cars', 'Trucks and Buses').
 - **MSRP:** Integer - Manufacturer's suggested retail price.
 - **PRODUCTCODE:** String - Unique product code.
 - **CUSTOMERNAME:** String - Name of the customer.
 - **PHONE:** String - Customer phone number.
 - **ADDRESSLINE1:** String - Primary address line.
 - **ADDRESSLINE2:** String - Secondary address line (often null).
 - **CITY:** String - Customer city.
 - **STATE:** String - Customer state (for applicable countries).
 - **POSTALCODE:** String - Postal code.
 - **COUNTRY:** String - Customer country.
 - **TERRITORY:** String - Sales territory (e.g., 'EMEA', 'APAC').
 - **CONTACTLASTNAME:** String - Last name of contact person.
 - **CONTACTFIRSTNAME:** String - First name of contact person.
 - **DEALSIZE:** String - Size of the deal (e.g., 'Small', 'Medium', 'Large').

- 1. What is the total sales amount by year?

```
select YEAR_ID, sum(SALES) AS TOTAL_SALES FROM sales_data GROUP BY YEAR_ID ORDER BY YEAR_ID;
```

Result Grid	
YEAR_ID	TOTAL_SALES
2003	3516979.540000001
2004	4724162.599999997
2005	1791486.71

- 2. Which product line has the highest average sales per order?

```
SELECT PRODUCTLINE, avg(SALES) AVG_SALES FROM sales_data GROUP BY PRODUCTLINE ORDER BY AVG_SALES DESC;
```

Result Grid	
PRODUCTLINE	AVG_SALES
Classic Cars	4053.3771044467394
Trucks and Buses	3746.8100996677726
Motorcycles	3523.831842900303
Planes	3186.2861764705885
Vintage Cars	3135.3391103789113
Ships	3053.150128205128
Trains	2938.2268831168826

- 3. Find the top 5 countries by total quantity ordered.

```
SELECT COUNTRY, SUM(QUANTITYORDERED) AS total_quantity_ordered FROM sales_data GROUP BY COUNTRY LIMIT 5;
```

Result Grid	
COUNTRY	total_quantity_ordered
USA	35659
France	11090
Norway	2842
Australia	6246
Finland	3192

- 4. What is the total sales for orders in 'Shipped' status during 2004?

```
SELECT STATUS, SUM(SALES) TOTAL_SALES, YEAR_ID FROM sales_data WHERE STATUS = 'Shipped' AND YEAR_ID = '2004';
```

Result Grid		
STATUS	TOTAL_SALES	YEAR_ID
Shipped	4528047.219999993	2004

- 5. List customers with more than 10 orders, sorted by total sales descending.

```
SELECT CUSTOMERNAME, COUNT(ORDERLINENUMBER) AS order_count, SUM(SALES) AS total_sales FROM sales_data GROUP BY CUSTOMERNAME HAVING order_count > 10 ORDER BY total_sales DESC;
```

Result Grid		
CUSTOMERNAME	order_count	total_sales
Euro Shopping Channel	259	91294.1100000002
Mini Gifts Distributors Ltd.	180	654858.06
Australian Collectors, Co.	55	200995.4099999997
Muscle Machine Inc	48	197736.9399999997
La Rochelle Gifts	53	180124.9
Dragon Souveniers, Ltd.	43	172989.6800000008
Land of Toys Inc.	49	164069.4400000003

- 6. What is the average deal size (sales) by quarter across all years?

```
SELECT YEAR_ID, QTR_ID, AVG(SALES) AS avg_sales FROM sales_data GROUP BY YEAR_ID, QTR_ID ORDER BY YEAR_ID, QTR_ID;
```

Result Grid		
YEAR_ID	QTR_ID	avg_sales
2003	1	3423.8053076923093
2003	2	3471.39024691358
2003	3	3529.970326086956
2003	4	3549.6280343511426
2004	1	3578.2432618025773
2004	2	3436.1467713004467
2004	3	3477.731253918497
2004	4	3534.692842105262
2005	1	3549.6435761589414
2005	2	4088.036079545455

- 7. Find the month with the highest sales in 2005.

```
select MONTH_ID, sum(SALES) as Total_Sales from sales_data where YEAR_ID = "2005" group by MONTH_ID order by Total_Sales desc limit 1;
```

Result Grid		Filter Rows:
	MONTH_ID	Total_Sales
▶	5	457861.05999999965

- 8. Calculate the total sales and average price each for products where MSRP > 100.

```
select sum(SALES) as Total_Sales, avg(PRICEEACH) as avg_price, PRODUCTLINE from sales_data where MSRP>100 group by PRODUCTLINE order by Total_Sales desc;
```

Result Grid			Filter Rows:	Export:
	Total_Sales	avg_price	PRODUCTLINE	
▶	3132695.40999999	96.85467391304547	Classic Cars	
	865071.5200000004	95.22147058823529	Vintage Cars	
	832596.1600000001	96.72146596858639	Trucks and Buses	
	601709.1600000006	96.88937984496125	Motorcycles	
	318386.4500000007	94.43756756756756	Planes	
	114032.12999999999	96.26307692307691	Ships	

- 9. Which city in the USA had the most orders in large deals?

```
SELECT CITY, COUNT(*) AS order_count FROM sales_data WHERE COUNTRY = 'USA' AND DEALSIZE = 'Large' GROUP BY CITY ORDER BY order_count DESC LIMIT 1;
```

Result Grid		Filter Row
	CITY	order_count
▶	San Rafael	11

- 10. Get the sales growth between 2003 and 2004 by product line.

SELECT PRODUCTLINE,

SUM(CASE WHEN YEAR_ID = 2003 THEN SALES ELSE 0 END) AS sales_2003,

SUM(CASE WHEN YEAR_ID = 2004 THEN SALES ELSE 0 END) AS sales_2004,

(SUM(CASE WHEN YEAR_ID = 2004 THEN SALES ELSE 0 END) - SUM(CASE WHEN YEAR_ID = 2003 THEN SALES ELSE 0 END)) AS growth

FROM sales_data GROUP BY PRODUCTLINE ORDER BY growth DESC;

Result Grid						Filter Rows:	Export:	Web Cell Content:
	PRODUCTLINE	sales_2003	sales_2004	growth	YEAR_ID	SALES	PRICEEACH	DEALSIZE
▶	Antique Cars	175	175	0	2003	175	175	Large
	Automobiles	175	175	0	2003	175	175	Large
	Boats	175	175	0	2003	175	175	Large
	Bicycles	175	175	0	2003	175	175	Large
	Cameras	175	175	0	2003	175	175	Large
	Classical Cars	175	175	0	2003	175	175	Large
	Classic Cars	175	175	0	2003	175	175	Large
	Computer Components	175	175	0	2003	175	175	Large
	Electronics	175	175	0	2003	175	175	Large
	Furniture	175	175	0	2003	175	175	Large
	Home Decor	175	175	0	2003	175	175	Large
	Leather Goods	175	175	0	2003	175	175	Large
	Luggage	175	175	0	2003	175	175	Large
	Motorcycles	175	175	0	2003	175	175	Large
	Office Equipment	175	175	0	2003	175	175	Large
	Photography	175	175	0	2003	175	175	Large
	Power Tools	175	175	0	2003	175	175	Large
	Small Parts	175	175	0	2003	175	175	Large
	Software	175	175	0	2003	175	175	Large
	Tools	175	175	0	2003	175	175	Large
	Watches	175	175	0	2003	175	175	Large

❖ These questions focus on practical analysis scenarios for sales data, exercising MySQL features like GROUP BY, HAVING, aggregate functions (SUM, AVG, COUNT), conditional logic (CASE), and filtering. If you load the data into MySQL, ensure ORDERDATE is parsed correctly (e.g., using STR_TO_DATE(ORDERDATE, '%m/%d/%Y') during import for date-based queries).