

OBJECTIVE: Analyze monthly revenue and order volume

Tools: MySQL

1. Overview of the Sales Dataset

Query 1 SQL File 3* x

Limit to 1000 rows

```
1 SELECT * FROM sales;
2
3
```

Result Grid

InvoiceNo	StockCode	Products	Quantity	OrderDate	UnitPrice	CustomerID	Country	Discount	PaymentMethod	ShippingCost	Category	SalesChannel	Re
221958	SKU_1964	White Mug	38	01-01-2020	1.71	37039	Australia	0.47	Bank Transfer	10.79	Apparel	In-store	Not
771155	SKU_1241	White Mug	18	01-01-2020	41.25	19144	Spain	0.19	paypal	9.51	Electronics	Online	Not
231932	SKU_1501	Headphones	49	01-01-2020	29.11	50472	Germany	0.35	Bank Transfer	23.03	Electronics	Online	Reb
465838	SKU_1760	Desk Lamp	14	01-01-2020	76.68	96586	Netherlands	0.14	paypal	11.08	Accessories	Online	Not
359178	SKU_1386	USB Cable	-30	01-01-2020	-68.11	53887	United Kingdom	1.501433043	Bank Transfer	13.98	Electronics	In-store	Not
744167	SKU_1006	Office Chair	47	01-01-2020	70.16	46567	Sweden	0.48	Credit Card	12.92	Electronics	Online	Not
210268	SKU_1087	USB Cable	25	01-01-2020	85.74	75098	Belgium	0.15	Bank Transfer	6.48	Stationery	Online	Not
832180	SKU_1597	Notebook	8	01-01-2020	95.65	87950	Norway	0.04	Bank Transfer	12.56	Electronics	In-store	Not
154886	SKU_1907	Wireless Mouse	19	01-01-2020	98.19	39718	Belgium	0.05	paypal	7.71	Apparel	Online	Not
237337	SKU_1866	Headphones	40	01-01-2020	98.17	13030	Italy	0.16	Bank Transfer	9.83	Apparel	Online	Not
621430	SKU_1144	Notebook	49	01-01-2020	87.56	32466	United Kingdom	0.19	paypal	12	Stationery	In-store	Reb
187498	SKU_1450	Office Chair	41	01-01-2020	59.51	89794	Australia	0.39	Bank Transfer	15.58	Furniture	In-store	Not
999159	SKU_1330	Blue Pen	41	01-01-2020	25.59	97817	Australia	0.01	Credit Card	19.22	Electronics	In-store	Not
275203	SKU_1709	Headphones	3	01-01-2020	2.09	43017	Spain	0.47	Credit Card	14.85	Stationery	Online	Not

sales 2 x

Read Only

2. Total Revenue by Country

Database Server Tools Scripting Help

Query 1 SQL File 3* x

Limit to 1000 rows

```
1 SELECT * FROM sales;
2
3 -- Total Revenue by Country
4 SELECT Country, sum(UnitPrice*Quantity) AS TotalRevenue
5 FROM sales
6 GROUP BY Country
7 ORDER BY TotalRevenue DESC;
8
9
10
11
12
13
14
15
16
```

Result Grid

Country	TotalRevenue
Italy	210670.0300000001
France	205477.69
Belgium	202238.93000000005
Netherlands	201569.07999999993
Portugal	195372.50000000006
United Kingdom	189840.019999999987
Norway	187345.4400000001
Germany	176286.51999999981
United States	176005.93
Spain	168602.18999999994

Result 16 x

Read Only

3. Top 5 Best-Selling Products

Query 1 SQL File 3* x

Limit to 1000 rows

```
-- Top 5 Best Selling Products
SELECT Products,sum(Quantity) AS TotalUnitSold FROM sales
GROUP BY Products
ORDER BY TotalUnitSold DESC
LIMIT 5;
```

Result Grid

Products	TotalUnitSold
T-shirt	4340
Wall Clock	4176
Office Chair	4027
Wireless Mouse	3936
Backpack	3802

Result 17

4. Sales by Category and Month

Query 1 SQL File 3* x

Limit to 1000 rows

```
-- Sales by Category and Month
SELECT Category,MONTH(str_to_date(OrderDate,'%d-%m-%y')) AS SalesMonth,sum(UnitPrice*Quantity) AS MonthlyRevenue
FROM sales
GROUP BY Category,SalesMonth
ORDER BY Category,SalesMonth;
```

Result Grid

Category	SalesMonth	MonthlyRevenue
Accessories	1	176235.80999999997
Accessories	2	150278.35
Accessories	3	89404.81000000004
Apparel	1	177546.280000000017
Apparel	2	186510.299999999996
Apparel	3	95321.949999999998
Electronics	1	198047.08000000001
Electronics	2	175812.84
Electronics	3	89439.180000000001
Furniture	1	160712.990000000002

Result 18

5. Online VS In-Store Revenue Split

The screenshot shows a SQL IDE window titled "Query 1" with a tab "SQL File 3". The query editor contains the following SQL code:

```
-- Online vs In-store Revenue Split
SELECT SalesChannel, sum(UnitPrice*Quantity) AS TotalRevenue
FROM sales
GROUP BY SalesChannel;
```

The result grid below the query shows the following data:

SalesChannel	TotalRevenue
In-store	1100473.25
Online	1107787.6400000013

The interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Filter Rows" field. The result grid has buttons for "Export", "Wrap Cell Content", and "Read Only".

6. Top Customers by Spend

The screenshot shows a SQL IDE window titled "Query 1" with a tab "SQL File 3". The query editor contains the following SQL code:

```
-- Top Customers by Spend
SELECT CustomerID, sum(UnitPrice*Quantity) AS TotalSpend
FROM sales
GROUP BY CustomerID
ORDER BY TotalSpend DESC
LIMIT 10;
```

The result grid below the query shows the following data:

CustomerID	TotalSpend
71231	5403.110000000001
11372	4925
89157	4716.740000000001
79142	4666.63
10278	4653.47
39314	4548.179999999999
64254	4535.519999999999
86402	4521.34
33849	4519.759999999999
20077	4494.599999999999

The interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Filter Rows" field. The result grid has buttons for "Export", "Wrap Cell Content", "Fetch rows", and "Read Only".

7. Most Popular Payment Method

Query 1 SQL File 3* x

Limit to 1000 rows

```
52
53
54
55 -- Most Popular Payment Method
56 • SELECT PaymentMethod, count(*) AS UsageCount
57 FROM sales
58 GROUP BY PaymentMethod
59 ORDER BY UsageCount DESC;
60
61
62
63
64
65
66
--
```

Result Grid

PaymentMethod	UsageCount
Credit Card	636
Bank Transfer	604
paypal	581

Result 21 x Read Only

8. Shipment cost by Provider

Query 1 SQL File 3* x

Limit to 1000 rows

```
66
67
68 -- Shipment cost by Provider
69 • SELECT ShipmentProvider, AVG(ShippingCost) AS AvgShippingCost, SUM(ShippingCost) AS TotalShippingCost
70 FROM sales
71 GROUP BY ShipmentProvider
72 ORDER BY TotalShippingCost DESC;
73
74
75
76
77
78
79
80
--
```

Result Grid

ShipmentProvider	AvgShippingCost	TotalShippingCost
UPS	17.6931762295082	8634.270000000002
FedEx	17.237296137339047	8032.579999999996
DHL	16.890917030567675	7736.039999999995
Royal Mail	17.42562347188264	7127.08

Result 22 x Read Only

9.Monthly Orders Trend

Query 1 SQL File 3" x

Limit to 1000 rows

```
77
78
79
80
81 -- Monthly Orders Trend
82 • SELECT YEAR(str_to_date(OrderDate, '%d-%m-%y')) AS OrderYear,
83       Month(str_to_date(OrderDate, '%d-%m-%y')) AS OrderMonth,
84       count(distinct InvoiceNo) AS TotalOrders
85 FROM sales
86 GROUP BY OrderYear, OrderMonth;
87
88
89
90
91
--
```

Result Grid Filter Rows: Exports: Wrap Cell Content:

OrderYear	OrderMonth	TotalOrders
2020	1	744
2020	2	696
2020	3	381

Result 23 x Read Only

Form Editor Field Types

10.Category-wise Revenue Analysis

Query 1 SQL File 3" x

Limit to 1000 rows

```
92
93
94
95 -- Category-wise Revenue Analysis
96 • SELECT Category,
97       count(distinct InvoiceNo) AS TotalOrders,
98       sum(UnitPrice*Quantity) AS TotalRevenue
99 FROM sales
100 GROUP BY Category;
101
102
103
104
105
106
--
```

Result Grid Filter Rows: Exports: Wrap Cell Content:

Category	TotalOrders	TotalRevenue
Accessories	350	415918.969999999986
Apparel	375	459378.529999999985
Electronics	385	463299.099999999945
Furniture	337	391873.479999999999
Stationery	374	477790.810000000006

Result 24 x Read Only

Form Editor Field Types

Output

11.Monthly Revenue and Order Volume Report

Query 1 SQL File 3* x

Limit to 1000 rows

```
106
107
108 -- Monthly Revenue and Order Volume Report
109
110 • SELECT YEAR(str_to_date(OrderDate, '%d-%m-%y')) AS OrderYear,
111        MONTH(str_to_date(OrderDate, '%d-%m-%y')) AS OrderMonth,
112        sum(UnitPrice * Quantity) AS TotalRevenue,
113        count(distinct InvoiceNo) AS TotalOrders
114 FROM sales
115 GROUP BY
116        YEAR(str_to_date(OrderDate, '%d-%m-%y')),
117        MONTH(str_to_date(OrderDate, '%d-%m-%y'))
118 ORDER BY OrderYear, OrderMonth
119 LIMIT 12;
120
```

Result Grid Filter Rows: Export: Wrap Cell Content: [f1](#)

	OrderYear	OrderMonth	TotalRevenue	TotalOrders
▶	2020	1	887759.3800000005	744
	2020	2	835370.5700000006	696
	2020	3	485130.94	381

Result 25 x Read Only

Result Grid
Form Editor
Field Types