

```

create database if not exists sales_data_analysis;
use sales_data_analysis;
show tables;
select * from sales_data;

```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons like 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. Below the toolbar is a result grid showing three rows of data from the 'sales\_data' table. The columns are ORDERNUMBER, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER, SALES, Day, OrderDate, STATUS, and Q. The data includes rows for order numbers 10107, 10121, and 10134. Below the result grid is an 'Output' section containing an 'Action Output' table. This table lists three actions: 'use sales\_data\_analysis', 'show tables', and 'select \* from sales\_data'. Each action has a timestamp, message, and duration/fetch time.

#	Time	Action	Message	Duration / Fetch
1	15:54:15	use sales_data_analysis	0 row(s) affected	0.000 sec
2	15:54:54	show tables	1 row(s) returned	0.000 sec / 0.000 sec
3	15:54:56	select * from sales_data	2823 row(s) returned	0.000 sec / 0.016 sec

### -- 1.Retrieve the total number of orders

```
select count(*)
```

```
from sales_data;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons like 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. Below the toolbar is a result grid showing one row with the value '2823'. Below the result grid is an 'Output' section containing an 'Action Output' table. This table lists two actions: 'select count(\*) from sales\_data' and 'select count(\*) TotalNOofOrders from sales\_data'. Both actions have a timestamp, message, and duration/fetch time.

#	Time	Action	Message	Duration / Fetch
4	15:59:30	select count(*) from sales_data	1 row(s) returned	0.016 sec / 0.000 sec
5	16:02:03	select count(*) TotalNOofOrders from sales_data	1 row(s) returned	0.000 sec / 0.000 sec

*Inference: There were 2823 orders in total.*

### 2.Calculate the total sales amount.

```
select round(sum(sales),2) Total_Sales
```

```
from sales_data;
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons like 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. Below the toolbar is a result grid showing one row with the value '10032628.85'. Below the result grid is an 'Output' section containing an 'Action Output' table. This table lists two actions: 'select count(\*) TotalNOofOrders from sales\_data' and 'select round(sum(sales),2) Total\_Sales from sales\_data'. Both actions have a timestamp, message, and duration/fetch time.

#	Time	Action	Message	Duration / Fetch
5	16:02:03	select count(*) TotalNOofOrders from sales_data	1 row(s) returned	0.000 sec / 0.000 sec
6	16:05:09	select round(sum(sales),2) Total_Sales from sales_data	1 row(s) returned	0.000 sec / 0.000 sec

*Inference: The Total Sales that has happened between 2003 to 2005 is \$100032628.85.*

### 3.Find the average quantity ordered per order.

```
select OrderNumber,round(avg(quantityordered),2) `Average No. of Order`  
from sales_data  
group by ordernumber  
order by ordernumber;
```

The screenshot shows a database query results window. At the top, there is a table titled "Result Grid" with one row labeled "Result 6". The table has three columns: "OrderNumber", "Average No. of Order", and "Message". The data rows are:

OrderNumber	Average No. of Order	Message
10100	37.75	
10101	35.50	
10102	40.00	
10103	33.81	
...	34.00	

Below the table, there is an "Output" section with an "Action Output" table. This table lists two actions:

#	Time	Action	Message	Duration / Fetch
6	16:05:09	select round(sum(sales),2) Total_Sales from sales_data	1 row(s) returned	0.000 sec / 0.000 sec
7	16:13:59	select OrderNumber,round(avg(quantityordered),2) `Av...	307 row(s) returned	0.016 sec / 0.000 sec

### 4.List the orders with a status of 'Cancelled'.

```
select *  
from sales_data  
where status like 'cancelled'  
order by ordernumber;
```

The screenshot shows a database query results window. At the top, there is a table titled "Result Grid" with one row labeled "sales\_data 7". The table has nine columns: "ORDERNRNUMBER", "QUANTITYORDERED", "PRICEEACH", "ORDERLINENUMBER", "SALES", "Day", "OrderDate", "STATUS", and "Message". The data rows are:

ORDERNRNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Message
10167	44	100	9	5924.16	23	2003-10-23 00:00:00	Cancelled	307 row(s) returned
10167	20	79.66	11	1593.2	23	2003-10-23 00:00:00	Cancelled	
10167	46	70.11	10	3225.06	23	2003-10-23 00:00:00	Cancelled	
10167	21	69.88	2	1467.48	23	2003-10-23 00:00:00	Cancelled	

Below the table, there is an "Output" section with an "Action Output" table. This table lists two actions:

#	Time	Action	Message	Duration / Fetch
7	16:13:59	select OrderNumber,round(avg(quantityordered),2) `Av...	307 row(s) returned	0.016 sec / 0.000 sec
8	16:15:20	select * from sales_data where status like 'cancelled' ...	60 row(s) returned	0.000 sec / 0.000 sec

-- or

```
select *  
from sales_data  
where status = 'cancelled'  
order by ordernumber;
```

The screenshot shows a database query results window. At the top, there is a table titled "Result Grid" with one row labeled "sales\_data 8". The table has nine columns: "ORDERNRNUMBER", "QUANTITYORDERED", "PRICEEACH", "ORDERLINENUMBER", "SALES", "Day", "OrderDate", "STATUS", and "Message". The data rows are:

ORDERNRNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Message
10167	44	100	9	5924.16	23	2003-10-23 00:00:00	Cancelled	60 row(s) returned
10167	20	79.66	11	1593.2	23	2003-10-23 00:00:00	Cancelled	
10167	46	70.11	10	3225.06	23	2003-10-23 00:00:00	Cancelled	
10167	21	69.88	2	1467.48	23	2003-10-23 00:00:00	Cancelled	

Below the table, there is an "Output" section with an "Action Output" table. This table lists two actions:

#	Time	Action	Message	Duration / Fetch
8	16:15:20	select * from sales_data where status like 'cancelled' ...	60 row(s) returned	0.000 sec / 0.000 sec
9	16:16:31	select * from sales_data where status = 'cancelled' ord...	60 row(s) returned	0.000 sec / 0.000 sec

### **5.Identify the month with the highest total sales.**

```
select month_id Month, round(sum(sales)) TotalSales  
from sales_data  
group by month_id  
order by totalsales desc limit 1;  
  
-- or  
  
select month_id, sum(sales) ,  
rank() over( order by sum(sales) desc ) as sales_order  
from sales_data  
group by month_id;
```

Month	TotalSales
11	2118886

Result 9 × Read On

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	16:16:31	select * from sales_data where status = 'cancelled' ord...	60 row(s) returned	0.000 sec / 0.000 sec
10	16:20:12	select month_id Month, round(sum(sales)) TotalSales f...	1 row(s) returned	0.000 sec / 0.000 sec

**Inference: The month that has the highest amount sales is November and has made sales worth \$2118886**

### **6. Find the top 5 customers with the highest total sales.**

```
select CustomerName,round(sum(sales),2) TotalSales  
from sales_data  
group by customername  
order by totalsales desc limit 5;
```

CustomerName	TotalSales
Euro Shopping Channel	912294.11
Mini Gifts Distributors Ltd.	654858.06
Australian Collectors, Co.	200995.41
Muscle Machine Inc	197736.94
La Rochelle Gifts	180124.9

Result 10 × Read On

Output

Action Output

#	Time	Action	Message	Duration / Fetch
10	16:20:12	select month_id Month, round(sum(sales)) TotalSales f...	1 row(s) returned	0.000 sec / 0.000 sec
11	16:22:43	select CustomerName,round(sum(sales),2) TotalSales f...	5 row(s) returned	0.000 sec / 0.000 sec

### **7. Calculate the average price for each product line.**

```
select ProductLine,round(avg(sales),2) AvgSales  
from sales_data  
group by productline  
order by productline;
```

ProductLine	AvgSales
Classic Cars	4053.38
Motorcycles	3523.83
Planes	3186.29
Ships	3053.15
Trains	2938.23

Result 11 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
11	16:22:43	select CustomerName,round(sum(sales),2) TotalSales f...	5 row(s) returned	0.000 sec / 0.000 sec
12	16:37:05	select ProductLine,round(avg(sales),2) AvgSales from ...	7 row(s) returned	0.016 sec / 0.000 sec

### 8. List the orders made in the year 2003.

```
select *
from sales_data
where year_id = 2003;
```

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10107	30	95.7	2	2871	24	2003-02-24 00:00:00	Shipped	1
10121	34	81.35	5	2765.9	7	2003-05-07 00:00:00	Shipped	2
10134	41	94.74	2	3884.34	1	2003-07-01 00:00:00	Shipped	3
10145	45	83.26	6	3746.7	25	2003-08-25 00:00:00	Shipped	3

sales\_data 12 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
12	16:37:05	select ProductLine,round(avg(sales),2) AvgSales from ...	7 row(s) returned	0.016 sec / 0.000 sec
13	16:38:58	select * from sales_data where year_id = 2003	1000 row(s) returned	0.000 sec / 0.016 sec

### 9. Find the total sales amount for each product line.

```
select ProductLine,round(sum(sales),2) TotalSales
from sales_data
group by productline
order by productline;
```

ProductLine	TotalSales
Classic Cars	3919615.66
Motorcycles	1166388.34
Planes	975003.57
Ships	714437.13
Trains	226243.47

Result 13 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
13	16:38:58	select * from sales_data where year_id = 2003	1000 row(s) returned	0.000 sec / 0.016 sec
14	16:40:09	select ProductLine,round(sum(sales),2) TotalSales fro...	7 row(s) returned	0.015 sec / 0.000 sec

### 10. Calculate the total sales for each country.

```
select Country,round(sum(sales),2) TotalSales
from sales_data
group by country
order by country;
```

	Country	TotalSales
▶	Australia	630623.1
	Austria	202062.53
	Belgium	108412.62
	Canada	224078.56
	Denmark	245637.15

Result 14 ×

Read On

Output

Action Output

#	Time	Action	Message	Duration / Fetch
14	16:40:09	select ProductLine,round(sum(sales),2) TotalSales fro...	7 row(s) returned	0.015 sec / 0.000 sec
15	16:41:12	select Country,round(sum(sales),2) TotalSales from sal...	19 row(s) returned	0.000 sec / 0.000 sec

**11. Identify the product line with the highest total sales.**

```
select ProductLine, max(sales) MaxSales
```

```
from sales_data
```

```
group by productline
```

```
order by max_sales desc limit 1;
```

```
-- or
```

```
select ProductLine, Sales
```

```
from sales_data
```

```
where sales = (select max(sales) from sales_data);
```

	ProductLine	MaxSales
▶	Vintage Cars	14082.8

Result 16 ×

Read On

Output

Action Output

#	Time	Action	Message	Duration / Fetch
17	16:41:54	select ProductLine, max(sales) MaxSales from sales_d...	Error Code: 1054. Unknown column 'max_sales' in 'ord...	0.000 sec
18	16:42:02	select ProductLine, max(sales) MaxSales from sales_d...	1 row(s) returned	0.000 sec / 0.000 sec

**12. List the orders with a quantity ordered greater than 50.**

```
select *
```

```
from sales_data
```

```
where quantityordered >50;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS
▶	10417	66	100	2	7516.08	13	2005-05-13 00:00:00	Disputed
	10403	66	100	9	11886.6	8	2005-04-08 00:00:00	Shipped
	10417	56	100	4	9218.16	13	2005-05-13 00:00:00	Disputed
	10400	64	100	9	9661.44	1	2005-04-01 00:00:00	Shipped

sales\_data 17 ×

Read On

Output

Action Output

#	Time	Action	Message	Duration / Fetch
18	16:42:02	select ProductLine, max(sales) MaxSales from sales_d...	1 row(s) returned	0.000 sec / 0.000 sec
19	16:42:48	select * from sales_data where quantityordered >50	54 row(s) returned	0.000 sec / 0.000 sec

### 13. Find the average price each year.

```
select year_id Year, round(avg(sales),2) AvgSales  
from sales_data  
group by year_id;
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays three rows of data: Year 2003 with AvgSales 3516.98, Year 2004 with AvgSales 3512.39, and Year 2005 with AvgSales 3747.88. The action history shows two actions: a select statement for quantityordered > 50 returning 54 rows, and the current query returning 3 rows.

Year	AvgSales
2003	3516.98
2004	3512.39
2005	3747.88

Result 18 x Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
19	16:42:48	select * from sales_data where quantityordered >50	54 row(s) returned	0.000 sec / 0.000 sec
20	16:43:26	select year_id Year, round(avg(sales),2) AvgSales fro...	3 row(s) returned	0.016 sec / 0.000 sec

### 14. Calculate the total sales for each territory.

```
select territory Territory , round(sum(sales),2) SalesByTerritory  
from sales_data  
group by territory  
order by territory asc;
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays four rows of data: APAC with SalesByTerritory 746121.83, EMEA with 4979272.41, Japan with 455173.22, and NA with 3852061.39. The action history shows two actions: a select statement for quantityordered > 50 returning 54 rows, and the current query returning 4 rows.

Territory	SalesByTerritory
APAC	746121.83
EMEA	4979272.41
Japan	455173.22
NA	3852061.39

Result 19 x Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
20	16:43:26	select year_id Year, round(avg(sales),2) AvgSales fro...	3 row(s) returned	0.016 sec / 0.000 sec
21	16:44:00	select territory Territory , round(sum(sales),2) SalesByT...	4 row(s) returned	0.015 sec / 0.000 sec

### 15. Identify the ordernumber with the min sales

```
select OrderNumber,Sales MinSales  
from sales_data  
where sales = (select round(min(sales),2) from sales_data);
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays one row of data: OrderNumber 10425 with MinSales 482.13. The action history shows two actions: a select statement for quantityordered > 50 returning 54 rows, and the current query returning 1 row.

OrderNumber	MinSales
10425	482.13

sales\_data 20 x Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
21	16:44:00	select territory Territory , round(sum(sales),2) SalesByT...	4 row(s) returned	0.015 sec / 0.000 sec
22	16:44:41	select OrderNumber,Sales MinSales from sales_data ...	1 row(s) returned	0.000 sec / 0.000 sec

### 16. List the orders made by customers in 'NYC'.

```
select *  
from sales_data  
where city = 'nyc';
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays four rows of data from the sales\_data table where city is 'nyc'. The action history shows two queries: one to select min(sales) and another to select all rows where city is 'nyc'.

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS
▶	10107	30	95.7	2	2871	24	2003-02-24 00:00:00	Shipped
	10237	23	100	7	2333.12	5	2004-04-05 00:00:00	Shipped
	10329	42	100	1	4396.14	15	2004-11-15 00:00:00	Shipped
	10163	21	100	1	4860.24	20	2003-10-20 00:00:00	Shipped

sales\_data 21 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
22	16:44:41	select OrderNumber,Sales MinSales from sales_data ...	1 row(s) returned	0.000 sec / 0.000 sec
23	16:48:03	select * from sales_data where city = 'nyc'	152 row(s) returned	0.000 sec / 0.000 sec

### 17. Calculate the total sales for each quarter

```
select qtr_id Quater,round(sum(sales),2) Sales  
from sales_data  
group by qtr_id;
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays four rows of data showing total sales for each quarter. The action history shows two queries: one to select all rows where city is 'nyc' and another to calculate total sales by quarter.

	Quater	Sales
▶	1	2350817.73
	2	2048120.3
	3	1758910.81
	4	3874780.01

Result 22 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
23	16:48:03	select * from sales_data where city = 'nyc'	152 row(s) returned	0.000 sec / 0.000 sec
24	16:49:54	select qtr_id Quater,round(sum(sales),2) Sales from sa...	4 row(s) returned	0.000 sec / 0.000 sec

### 18. Find the total sales amount for each customer.

```
select CustomerName,round(sum(sales),2) Sales  
from sales_data  
group by customername  
order by sales desc;
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays five rows of data showing total sales for each customer. The action history shows two queries: one to calculate total sales by customer and another to select all rows.

	CustomerName	Sales
▶	Euro Shopping Channel	912294.11
	Mini Gifts Distributors Ltd.	654858.06
	Australian Collectors, Co.	200995.41
	Muscle Machine Inc	197736.94
	La Rochelle Gifts	180124.9

Result 23 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
24	16:49:54	select qtr_id Quater,round(sum(sales),2) Sales from sa...	4 row(s) returned	0.000 sec / 0.000 sec
25	16:50:27	select CustomerName,round(sum(sales),2) Sales from ...	92 row(s) returned	0.015 sec / 0.000 sec

#### 19. Identify the orders with a deal size of 'small' and status 'cancelled'

```
select *\n\nfrom sales_data\n\nwhere dealsize like 'small' and status like 'cancelled';
```

The screenshot shows a database query results window. At the top, there is a table titled "sales\_data 24" with columns: ORDERNUMBER, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER, SALES, Day, OrderDate, and STATUS. Below the table, the "Output" section displays the results of the query. The "Action Output" table shows two actions: a successful select statement for CustomerName and another for sales, both returning 92 rows, and a successful select statement for sales\_data where dealsize like 'small' and status like 'cancelled', returning 27 rows. The total duration for all actions is 0.015 sec / 0.000 sec.

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS
10248	20	100	3	2910.4	7	2004-05-07 00:00:00	Cancelled
10253	22	100	11	2402.84	1	2004-06-01 00:00:00	Cancelled
10253	25	90.17	14	2254.25	1	2004-06-01 00:00:00	Cancelled
10253	23	83.93	9	1930.39	1	2004-06-01 00:00:00	Cancelled

sales\_data 24 ×

Output

#	Time	Action	Message	Duration / Fetch
25	16:50:27	select CustomerName,round(sum(sales),2) Sales from ...	92 row(s) returned	0.015 sec / 0.000 sec
26	16:51:31	select * from sales_data where dealsize like 'small' an...	27 row(s) returned	0.000 sec / 0.000 sec

Read On

#### 20. List the orders with a status of 'On Hold'.

```
select *\n\nfrom sales_data\n\nwhere status like 'on hold';
```

The screenshot shows a database query results window. At the top, there is a table titled "sales\_data 26" with columns: ORDERNUMBER, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER, SALES, Day, OrderDate, and STATUS. Below the table, the "Output" section displays the results of the query. The "Action Output" table shows two actions: a successful select statement for year\_id, month\_id, Month, and round(sum(sales)), returning 12 rows, and a successful select statement for sales\_data where status like 'on hold', returning 44 rows. The total duration for all actions is 0.016 sec / 0.000 sec.

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS
10414	19	100	3	2764.88	6	2005-05-06 00:00:00	On Hold
10334	26	100	2	3188.12	19	2004-11-19 00:00:00	On Hold
10407	59	100	11	7048.14	22	2005-04-22 00:00:00	On Hold
10407	76	100	2	14082.8	22	2005-04-22 00:00:00	On Hold

sales\_data 26 ×

Output

#	Time	Action	Message	Duration / Fetch
27	16:52:09	select year_id Year,month_id Month,round(sum(sales),...	12 row(s) returned	0.016 sec / 0.000 sec
28	16:53:08	select * from sales_data where status like 'on hold'	44 row(s) returned	0.000 sec / 0.000 sec

Read On

#### 21. Calculate the total sales for each month in the year 2004.

```
select year_id Year,month_id Month,round(sum(sales),2) TotalSales\n\nfrom sales_data\n\nwhere year_id = 2004\n\ngroup by month_id\n\norder by month_id;
```

The screenshot shows a database query results window. At the top, there is a table titled "Result 25" with columns: Year, Month, and TotalSales. Below the table, the "Output" section displays the results of the query. The "Action Output" table shows two actions: a successful select statement for sales\_data where dealsize like 'small' and status like 'cancelled', returning 27 rows, and a successful select statement for year\_id, month\_id, Month, and round(sum(sales)), returning 12 rows. The total duration for all actions is 0.016 sec / 0.000 sec.

Year	Month	TotalSales
2004	1	316577.42
2004	2	311419.53
2004	3	205733.73
2004	4	206148.12
2004	5	273438.39

Result 25 ×

Output

#	Time	Action	Message	Duration / Fetch
26	16:51:31	select * from sales_data where dealsize like 'small' an...	27 row(s) returned	0.000 sec / 0.000 sec
27	16:52:09	select year_id Year,month_id Month,round(sum(sales),...	12 row(s) returned	0.016 sec / 0.000 sec

Read On

**22. Identify the orders with a price each higher than the MSRP.**

```
select *
```

```
from sales_data
```

```
where priceeach > msrp;
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

OrderNumber	QuantityOrdered	PriceEach	OrderLineNumber	Sales	Day	OrderDate	Status	Q
10107	30	95.7	2	2871	24	2003-02-24 00:00:00	Shipped	1
10159	49	100	14	5205.27	10	2003-10-10 00:00:00	Shipped	4
10168	36	96.66	1	3479.76	28	2003-10-28 00:00:00	Shipped	4
10188	48	100	1	5512.32	18	2003-11-18 00:00:00	Shipped	4

**Action Output:**

#	Time	Action	Message	Duration / Fetch
28	16:53:08	select * from sales_data where status like 'on hold'	44 row(s) returned	0.000 sec / 0.000 sec
29	16:55:27	select * from sales_data where priceeach > msrp	843 row(s) returned	0.016 sec / 0.000 sec

**23. List the orders with a sales amount greater than \$10000.**

```
select ordernumber OrderNumber,sales Sales
```

```
from sales_data
```

```
where sales >10000
```

```
order by sales desc;
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

OrderNumber	Sales
10407	14082.8
10322	12536.5
10424	12001
10412	11887.8
10403	11886.6

**Action Output:**

#	Time	Action	Message	Duration / Fetch
29	16:55:27	select * from sales_data where priceeach > msrp	843 row(s) returned	0.016 sec / 0.000 sec
30	16:55:50	select ordernumber OrderNumber,sales Sales from sal...	16 row(s) returned	0.016 sec / 0.000 sec

**24. Find the average quantity ordered per customer.**

```
select customername CustomerName, avg(quantityordered) AverageOrderQuantity
```

```
from sales_data
```

```
group by customername;
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

CustomerName	AverageOrderQuantity
Land of Toys Inc.	33.2857
Reims Collectables	34.9512
Lyon Souveniers	34.2000
Toys4GrownUps.com	35.3333
Corporate Gift Ideas Co.	35.2927

**Action Output:**

#	Time	Action	Message	Duration / Fetch
30	16:55:50	select ordernumber OrderNumber,sales Sales from sal...	16 row(s) returned	0.016 sec / 0.000 sec
31	16:56:35	select customername CustomerName, avg(quantityord...	92 row(s) returned	0.016 sec / 0.000 sec

## 25. Calculate the total sales for each state in USA

```
select state State, round(sum(sales),2) Sales  
from sales_data  
where country like 'usa'  
group by state  
order by state;
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

State	Sales
CA	1505542.23
CT	238661.28
MA	666443.72
NH	131685.3
NJ	83228.19

**Action Output:**

#	Time	Action	Message	Duration / Fetch
31	16:56:35	select customermame CustomerName, avg(quantityord...	92 row(s) returned	0.016 sec / 0.000 sec
32	16:57:19	select state State, round(sum(sales),2) Sales from sale...	8 row(s) returned	0.000 sec / 0.000 sec

## 26. Identify the product line with the lowest total sales.

```
select ordernumber OrderNumber,productline ProductLine,sales  
from sales_data  
where sales = (select min(sales) from sales_data);
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

OrderNumber	ProductLine	sales
10425	Trucks and Buses	482.13

**Action Output:**

#	Time	Action	Message	Duration / Fetch
32	16:57:19	select state State, round(sum(sales),2) Sales from sale...	8 row(s) returned	0.000 sec / 0.000 sec
33	16:58:21	select ordernumber OrderNumber,productline Product...	1 row(s) returned	0.000 sec / 0.000 sec

## 27. List the orders made by customers with the last name 'Young'.

```
select *  
from sales_data  
where contactlastname like 'young';
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

ORDERNRNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10145	45	83.26	6	3746.7	25	2003-08-25 00:00:00	Shipped	3
10322	40	100	1	6000.4	4	2004-11-04 00:00:00	Shipped	4
10145	37	100	9	5192.95	25	2003-08-25 00:00:00	Shipped	3
10145	33	100	8	5176.38	25	2003-08-25 00:00:00	Shipped	3

**Action Output:**

#	Time	Action	Message	Duration / Fetch
33	16:58:21	select ordernumber OrderNumber,productline Product...	1 row(s) returned	0.000 sec / 0.000 sec
34	16:59:27	select * from sales_data where contactlastname like 'y...'	115 row(s) returned	0.016 sec / 0.000 sec

## 28. Calculate the total sales for each city.

```
select country Country,city City,round(sum(sales),2) TotalSales  
from sales_data  
group by city,country  
order by country asc , totalsales desc;
```

Country	City	TotalSales
Australia	Melbourne	200995.41
Australia	North Sydney	153996.13
Australia	Chatswood	151570.98
Australia	Glen Waverly	64591.46
Australia	South Brisbane	59469.12

Result 33 × Read On

Output

Action Output
# Time Action Message Duration / Fetch
34 16:59:27 select * from sales_data where contactlastname like 'y...' 115 row(s) returned 0.016 sec / 0.000 sec
35 17:00:17 select country Country,city City,round(sum(sales),2) To... 73 row(s) returned 0.016 sec / 0.000 sec

## 29. Find the orders placed on weekends.

```
select *  
from sales_data  
where dayofweek(orderdate) in (1,7)  
order by ordedate;
```

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10143	49	100	15	5597.76	10	2003-08-10 00:00:00	Shipped	3
10143	32	100	7	5248	10	2003-08-10 00:00:00	Shipped	3
10143	46	74.84	13	3442.64	10	2003-08-10 00:00:00	Shipped	3
10143	34	100	12	3455.76	10	2003-08-10 00:00:00	Shipped	3

sales\_data 34 × Read On

Output

Action Output
# Time Action Message Duration / Fetch
36 17:00:46 select * from sales_data where dayofweek(orderdate)i... Error Code: 1054. Unknown column 'ordedate' in 'orde... 0.000 sec
37 17:00:53 select * from sales_data where dayofweek(orderdate)i... 244 row(s) returned 0.016 sec / 0.000 sec

## 30. Identify the product line with the highest average sales price.

```
select productline ProductLine,round(avg(sales),2) HighestAverageSales  
from sales_data  
group by productline  
order by avg(sales) desc  
limit 1;
```

ProductLine	HighestAverageSales
Classic Cars	4053.38

Result 35 × Read On

Output

Action Output
# Time Action Message Duration / Fetch
37 17:00:53 select * from sales_data where dayofweek(orderdate)i... 244 row(s) returned 0.016 sec / 0.000 sec
38 17:01:50 select productline ProductLine,round(avg(sales),2) Hig... 1 row(s) returned 0.000 sec / 0.000 sec

**31. List the orders with a quantity ordered less than 10.**

```
select *
```

```
from sales_data
```

```
where quantityordered <10;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_I
▶	10409	6	100	2	785.64	23	2005-04-23 00:00:00	Shipped	2
	10407	6	90.19	3	541.14	22	2005-04-22 00:00:00	On Hold	2

sales\_data 36 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
38	17:01:50	select productline ProductLine,round(avg(sales),2) Hig...	1 row(s) returned	0.000 sec / 0.000 sec
39	17:03:30	select * from sales_data where quantityordered <10	2 row(s) returned	0.000 sec / 0.000 sec

**32. Calculate the total sales for each postal code.**

```
select postalcode PostalCode, round(sum(sales),2) TotalSales
```

```
from sales_data
```

```
where postalcode <>"
```

```
group by postalcode
```

```
order by TotalSales desc;
```

	PostalCode	TotalSales
▶	28034	912294.11
	97562	756752.85
	10022	560787.77
	94217	331011.85
	50553	207874.86

Result 37 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
39	17:03:30	select * from sales_data where quantityordered <10	2 row(s) returned	0.000 sec / 0.000 sec
40	17:03:56	select postalcode PostalCode, round(sum(sales),2) Tot...	73 row(s) returned	0.000 sec / 0.000 sec

**- 33. Find the orders made by customers located in 'New York'.**

```
select *
```

```
from sales_data
```

```
where state like 'ny';
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_I
▶	10107	30	95.7	2	2871	24	2003-02-24 00:00:00	Shipped	1
	10237	23	100	7	2333.12	5	2004-04-05 00:00:00	Shipped	2
	10329	42	100	1	4396.14	15	2004-11-15 00:00:00	Shipped	4
	10163	21	100	1	4860.24	20	2003-10-20 00:00:00	Shipped	4

sales\_data 38 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
40	17:03:56	select postalcode PostalCode, round(sum(sales),2) Tot...	73 row(s) returned	0.000 sec / 0.000 sec
41	17:04:28	select * from sales_data where state like 'ny'	178 row(s) returned	0.000 sec / 0.000 sec

-- 34. Identify the orders with a status of 'In Process'

```
select *
```

```
from sales_data
```

```
where status like 'in process';
```

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10424	50	100	6	12001	31	2005-05-31 00:00:00	In Process	
10425	38	100	12	5894.94	31	2005-05-31 00:00:00	In Process	
10424	49	100	3	7969.36	31	2005-05-31 00:00:00	In Process	
10425	33	100	4	4692.6	31	2005-05-31 00:00:00	In Process	

sales\_data 39 × Read Only

Output:

Action Output		
#	Time	Action
41	17:04:28	select * from sales_data where state like 'hy'
42	17:05:02	select * from sales_data where status like 'in process'

Message Duration / Fetch  
178 row(s) returned 0.000 sec / 0.000 sec  
41 row(s) returned 0.016 sec / 0.000 sec

-- 35. List the orders with a deal size of 'Small'.

```
select *
```

```
from sales_data
```

```
where dealsize = 'small';
```

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10107	30	95.7	2	2871	24	2003-02-24 00:00:00	Shipped	1
10121	34	81.35	5	2765.9	7	2003-05-07 00:00:00	Shipped	2
10180	29	86.13	9	2497.77	11	2003-11-11 00:00:00	Shipped	4
10201	22	98.57	2	2168.54	1	2003-12-01 00:00:00	Shipped	4

sales\_data 40 × Read Only

Output:

Action Output		
#	Time	Action
42	17:05:02	select * from sales_data where status like 'in process'
43	17:05:28	select * from sales_data where dealsize = 'small'

Message Duration / Fetch  
41 row(s) returned 0.016 sec / 0.000 sec  
1282 row(s) returned 0.000 sec / 0.015 sec

-- 36. Calculate the total sales made on 3rd quarter and status like shipped

```
select *
```

```
from sales_data
```

```
where status like 'shipped' and qtr_id like 3;
```

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10134	41	94.74	2	3884.34	1	2003-07-01 00:00:00	Shipped	3
10145	45	83.26	6	3746.7	25	2003-08-25 00:00:00	Shipped	3
10275	45	92.83	1	4177.35	23	2004-07-23 00:00:00	Shipped	3
10285	36	100	6	4099.68	27	2004-08-27 00:00:00	Shipped	3

sales\_data 41 × Read Only

Output:

Action Output		
#	Time	Action
43	17:05:28	select * from sales_data where dealsize = 'small'
44	17:05:55	select * from sales_data where status like 'shipped' an...

Message Duration / Fetch  
1282 row(s) returned 0.000 sec / 0.015 sec  
503 row(s) returned 0.000 sec / 0.000 sec

-- 37. Identify the orders with a price each lower than the MSRP.

```
select *
```

```
from sales_data
```

```
where priceeach < msrp;
```

The screenshot shows a database interface with a results grid and an action history table.

**Result Grid:**

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10121	34	81.35	5	2765.9	7	2003-05-07 00:00:00	Shipped	2
10134	41	94.74	2	3884.34	1	2003-07-01 00:00:00	Shipped	3
10145	45	83.26	6	3746.7	25	2003-08-25 00:00:00	Shipped	3
10180	29	86.13	9	2497.77	11	2003-11-11 00:00:00	Shipped	4

**Action Output:**

#	Time	Action	Message	Duration / Fetch
44	17:05:55	select * from sales_data where status like 'shipped' an...	503 row(s) returned	0.000 sec / 0.000 sec
45	17:06:31	select * from sales_data where priceeach < msrp	1953 row(s) returned	0.000 sec / 0.015 sec

-- 38. List the orders made by customers located in 'London'.

```
select *
```

```
from sales_data
```

```
where city like 'london';
```

The screenshot shows a database interface with a results grid and an action history table.

**Result Grid:**

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10129	33	100	2	4398.24	12	2003-06-12 00:00:00	Shipped	2
10186	26	100	9	3854.24	14	2003-11-14 00:00:00	Shipped	4
10175	33	100	9	5362.83	6	2003-11-06 00:00:00	Shipped	4
10175	26	100	1	3543.28	6	2003-11-06 00:00:00	Shipped	4

**Action Output:**

#	Time	Action	Message	Duration / Fetch
45	17:06:31	select * from sales_data where priceeach < msrp	1953 row(s) returned	0.000 sec / 0.015 sec
46	17:06:56	select * from sales_data where city like 'london'	38 row(s) returned	0.016 sec / 0.000 sec

-- 39. Calculate the total sales for each day of the week.

```
select dayofweek(orderdate) DayNumberOfWeek,round(sum(sales),2) TotalSales
```

```
from sales_data
```

```
group by daynumberofweek
```

```
order by TotalSales desc;
```

The screenshot shows a database interface with a results grid and an action history table.

**Result Grid:**

DayNumberOfWeek	TotalSales
6	2140653.8
5	1947412.2
4	1947146.04
3	1900773.29
2	1254535.03

**Action Output:**

#	Time	Action	Message	Duration / Fetch
46	17:06:56	select * from sales_data where city like 'london'	38 row(s) returned	0.016 sec / 0.000 sec
47	17:07:20	select dayofweek(orderdate) DayNumberOfWeek,rou...	7 row(s) returned	0.000 sec / 0.000 sec

-- 40. Identify the orders with a status of 'Shipped'.

```
select *
```

```
from sales_data
```

```
where status like 'shipped';
```

sales\_data 45 x Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
47	17:07:20	select dayofweek(orderdate) DayNumberOfWeek,round...	7 row(s) returned	0.000 sec / 0.000 sec
48	17:07:52	select * from sales_data where status like 'shipped'	2617 row(s) returned	0.000 sec / 0.016 sec

-- 41. List the orders with a quantity ordered between 20 and 30.

```
select *
```

```
from sales_data
```

```
where quantityordered between 20 and 30;
```

sales\_data 46 x Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
48	17:07:52	select * from sales_data where status like 'shipped'	2617 row(s) returned	0.000 sec / 0.016 sec
49	17:08:15	select * from sales_data where quantityordered betwe...	993 row(s) returned	0.000 sec / 0.015 sec

-- 42. Calculate the total sales for each product code.

```
select productcode ProductCode, round(sum(sales),2) TotalSales
```

```
from sales_data
```

```
group by productcode;
```

Result 47 x Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
49	17:08:15	select * from sales_data where quantityordered betwe...	993 row(s) returned	0.000 sec / 0.015 sec
50	17:08:37	select productcode ProductCode, round(avg(priceeac...)	109 row(s) returned	0.016 sec / 0.000 sec

**-- 43. Find the average price for each product code.**

```
select productcode ProductCode, round(avg(priceeach),2) AvgPrice  
from sales_data  
group by productcode;
```

The screenshot shows a database query results window. At the top, there is a header row with columns 'ProductCode' and 'AvgPrice'. Below this, five rows of data are displayed, each representing a product code and its corresponding average price. The data is as follows:

ProductCode	AvgPrice
S10_1678	92.61
S10_1949	100
S10_2016	94.37
S10_4698	98.59
S10_4757	94.19

Below the table, a message bar says 'Result 48 x' and a 'Read Only' button is visible. Under the 'Output' section, there is an 'Action Output' table showing two log entries. The first entry is for a SELECT statement that returned 109 rows and took 0.016 seconds. The second entry is for the same SELECT statement again, also returning 109 rows and taking 0.016 seconds.

#	Time	Action	Message	Duration / Fetch
50	17:08:37	select productcode ProductCode, round(avg(priceeach),2) AvgPrice from sales_data group by productcode;	109 row(s) returned	0.016 sec / 0.000 sec
51	17:09:17	select productcode ProductCode, round(avg(priceeach),2) AvgPrice from sales_data group by productcode;	109 row(s) returned	0.016 sec / 0.000 sec

**-- 44. Identify the orders with a deal size of 'Medium'.**

```
select *  
from sales_data  
where dealsize = 'medium';
```

The screenshot shows a database query results window. At the top, there is a header row with columns 'ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER', 'SALES', 'Day', 'OrderDate', 'STATUS', and 'Q'. Below this, four rows of data are displayed, each representing an order with a deal size of 'Medium'. The data is as follows:

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10134	41	94.74	2	3884.34	1	2003-07-01 00:00:00	Shipped	3
10145	45	83.26	6	3746.7	25	2003-08-25 00:00:00	Shipped	3
10159	49	100	14	5205.27	10	2003-10-10 00:00:00	Shipped	4
10168	36	96.66	1	3479.76	28	2003-10-28 00:00:00	Shipped	4

Below the table, a message bar says 'sales\_data 49 x' and a 'Read Only' button is visible. Under the 'Output' section, there is an 'Action Output' table showing two log entries. The first entry is for a SELECT statement that returned 109 rows and took 0.016 seconds. The second entry is for the same SELECT statement again, returning 1384 rows and taking 0.000 seconds.

#	Time	Action	Message	Duration / Fetch
51	17:09:17	select productcode ProductCode, round(avg(priceeach),2) AvgPrice from sales_data group by productcode;	109 row(s) returned	0.016 sec / 0.000 sec
52	17:09:42	select * from sales_data where dealsize = 'medium'	1384 row(s) returned	0.000 sec / 0.000 sec

**-- 45. List the orders made by customers with the first name 'paul'.**

```
select *  
from sales_data  
where contactfirstname = 'paul';
```

The screenshot shows a database query results window. At the top, there is a header row with columns 'ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER', 'SALES', 'Day', 'OrderDate', 'STATUS', and 'Q'. Below this, four rows of data are displayed, each representing an order made by a customer with the first name 'paul'. The data is as follows:

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10121	34	81.35	5	2765.9	7	2003-05-07 00:00:00	Shipped	2
10359	48	54.68	6	2624.64	15	2004-12-15 00:00:00	Shipped	4
10359	42	100	8	4764.48	15	2004-12-15 00:00:00	Shipped	4
10121	50	100	4	8284	7	2003-05-07 00:00:00	Shipped	2

Below the table, a message bar says 'sales\_data 50 x' and a 'Read Only' button is visible. Under the 'Output' section, there is an 'Action Output' table showing two log entries. The first entry is for a SELECT statement that returned 1384 rows and took 0.000 seconds. The second entry is for the same SELECT statement again, returning 41 rows and taking 0.015 seconds.

#	Time	Action	Message	Duration / Fetch
52	17:09:42	select * from sales_data where dealsize = 'medium'	1384 row(s) returned	0.000 sec / 0.000 sec
53	17:10:04	select * from sales_data where contactfirstname = 'paul'	41 row(s) returned	0.015 sec / 0.000 sec

-- 46. Calculate the total sales for each month of the year.

```
select year_id Year,month_id Month,round(sum(sales),2) Total_Sales  
from sales_data  
group by year_id, month_id  
order by year_id, month_id asc;
```

The screenshot shows a database query results window. At the top, there is a table header with columns: Year, Month, and Total\_Sales. Below the header, there are five rows of data. The first row is highlighted in blue. The data is as follows:

Year	Month	Total_Sales
2003	1	129753.6
2003	2	140836.19
2003	3	174504.9
2003	4	201609.55
2003	5	192673.11

Below the table, there is a message bar with "Result 51" and a "Read On" button. Underneath the table, there is an "Output" section with a "Action Output" dropdown. A log table shows two entries:

#	Time	Action	Message	Duration / Fetch
53	17:10:04	select * from sales_data where contactfirstname = 'paul'	41 row(s) returned	0.015 sec / 0.000 sec
54	17:10:29	select year_id Year,month_id Month,round(sum(sales),...	29 row(s) returned	0.016 sec / 0.000 sec

-- 47. Identify the orders made by customers located in 'Paris'.

```
select ordernumber OrderNumber, customername CustomerNameAtParis  
from sales_data  
where city = 'paris';
```

The screenshot shows a database query results window. At the top, there is a table header with columns: OrderNumber and CustomerNameAtParis. Below the header, there are five rows of data. The first row is highlighted in blue. The data is as follows:

OrderNumber	CustomerNameAtParis
10134	Lyon Souveniers
10211	Auto Canal Petit
10134	Lyon Souveniers
10402	Auto Canal Petit
10134	Lyon Souveniers

Below the table, there is a message bar with "sales\_data 52" and a "Read On" button. Underneath the table, there is an "Output" section with a "Action Output" dropdown. A log table shows two entries:

#	Time	Action	Message	Duration / Fetch
54	17:10:29	select year_id Year,month_id Month,round(sum(sales),...	29 row(s) returned	0.016 sec / 0.000 sec
55	17:10:50	select ordernumber OrderNumber, customername Cust...	70 row(s) returned	0.000 sec / 0.000 sec

-- 48. List the orders with a quantity ordered equal to 1.

```
select *  
from sales_data  
where quantityordered = 1;
```

The screenshot shows a database query results window. At the top, there is a table header with columns: ORDERNUMBER, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER, SALES, Day, OrderDate, STATUS, QTR\_ID, and MONTH. Below the header, there are no visible rows of data. The message bar at the bottom has "sales\_data 53" and a "Read On" button. Underneath the header, there is an "Output" section with a "Action Output" dropdown. A log table shows two entries:

#	Time	Action	Message	Duration / Fetch
55	17:10:50	select ordernumber OrderNumber, customername Cust...	70 row(s) returned	0.000 sec / 0.000 sec
56	17:11:12	select * from sales_data where quantityordered = 1	0 row(s) returned	0.000 sec / 0.000 sec

-- 49. Calculate the total sales for each product line in the year 2004.

```
select productline ProductLine,round(sum(sales),2) TotalSalesIN2004
```

```
from sales_data
```

```
where year_id = 2004
```

```
group by productline;
```

The screenshot shows a database query results window. At the top, there is a table titled "Result Grid" with columns "ProductLine" and "TotalSalesIN2004". The data is as follows:

ProductLine	TotalSalesIN2004
Motorcycles	560545.23
Classic Cars	1762257.09
Trucks and Buses	529302.89
Vintage Cars	911423.77
Planes	502671.8

Below the table, there is a "Result 54" section and a "Read On" button. Under "Output", there is an "Action Output" table with two rows:

#	Time	Action	Message	Duration / Fetch
56	17:11:12	select * from sales_data where quantityordered = 1	0 row(s) returned	0.000 sec / 0.000 sec
57	17:11:37	select productline ProductLine,round(sum(sales),2) Tot...	7 row(s) returned	0.000 sec / 0.000 sec

-- 50. Identify the orders with a status of 'Resolved'.

```
select *
```

```
from sales_data
```

```
where status = 'resolved';
```

The screenshot shows a database query results window. At the top, there is a table titled "Result Grid" with columns corresponding to the sales data table. The data is as follows:

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10164	21	100	2	3536.82	21	2003-10-21 00:00:00	Resolved	4
10164	49	100	6	6563.06	21	2003-10-21 00:00:00	Resolved	4
10164	36	99.17	8	3570.12	21	2003-10-21 00:00:00	Resolved	4
10367	49	56.3	1	2758.7	12	2005-01-12 00:00:00	Resolved	1

Below the table, there is a "sales\_data 55" section and a "Read On" button. Under "Output", there is an "Action Output" table with two rows:

#	Time	Action	Message	Duration / Fetch
57	17:11:37	select productline ProductLine,round(sum(sales),2) Tot...	7 row(s) returned	0.000 sec / 0.000 sec
58	17:12:02	select * from sales_data where status = 'resolved'	47 row(s) returned	0.016 sec / 0.000 sec

-- 51. List the orders with a deal size of 'Small' and a sales amount less than \$5000.

```
select *
```

```
from sales_data
```

```
where dealsize like 'small' and sales < 5000;
```

The screenshot shows a database query results window. At the top, there is a table titled "Result Grid" with columns corresponding to the sales data table. The data is as follows:

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10107	30	95.7	2	2871	24	2003-02-24 00:00:00	Shipped	1
10121	34	81.35	5	2765.9	7	2003-05-07 00:00:00	Shipped	2
10180	29	86.13	9	2497.77	11	2003-11-11 00:00:00	Shipped	4
10201	22	98.57	2	2168.54	1	2003-12-01 00:00:00	Shipped	4

Below the table, there is a "sales\_data 56" section and a "Read On" button. Under "Output", there is an "Action Output" table with two rows:

#	Time	Action	Message	Duration / Fetch
58	17:12:02	select * from sales_data where status = 'resolved'	47 row(s) returned	0.016 sec / 0.000 sec
59	17:12:24	select * from sales_data where dealsize like 'small' and...	1282 row(s) returned	0.000 sec / 0.031 sec

-- 52. Calculate the total sales for each day of the month.

```
select day(orderdate) OrderDate,round(sum(sales),2) TotalSales  
from sales_data  
group by orderdate  
order by totalsales desc;
```

The screenshot shows a database query results window. The results grid displays two columns: OrderDate and TotalSales. The data is as follows:

OrderDate	TotalSales
24	137644.72
14	131236
6	114456.85
12	111156.73
2	109432.27

Below the results grid is an output log titled "Action Output" showing two log entries:

#	Time	Action	Message	Duration / Fetch
59	17:12:24	select * from sales_data where dealsize like 'small' and...	1282 row(s) returned	0.000 sec / 0.031 sec
60	17:12:44	select day(orderdate) OrderDate,round(sum(sales),2) T...	252 row(s) returned	0.000 sec / 0.000 sec

-- 53. Identify the orders with a price each equal to the MSRP.

```
select ordernumber OrderNumber, priceeach PriceEach,msrp MSRP  
from sales_data  
where priceeach = msrp;
```

The screenshot shows a database query results window. The results grid displays three columns: OrderNumber, PriceEach, and MSRP. All rows show values of 100 for both PriceEach and MSRP. The data is as follows:

OrderNumber	PriceEach	MSRP
10105	100	100
10128	100	100
10165	100	100
10177	100	100
10196	100	100

Below the results grid is an output log titled "Action Output" showing two log entries:

#	Time	Action	Message	Duration / Fetch
60	17:12:44	select day(orderdate) OrderDate,round(sum(sales),2) T...	252 row(s) returned	0.000 sec / 0.000 sec
61	17:13:06	select ordernumber OrderNumber, priceeach PriceEac...	27 row(s) returned	0.000 sec / 0.000 sec

-- 54. List the orders made by customers located in 'Sydney'.

```
select *  
from sales_data  
where city like 'sydney';
```

The screenshot shows a database query results window. The results grid is empty, indicating 0 row(s) returned. Below the results grid is an output log titled "Action Output" showing two log entries:

#	Time	Action	Message	Duration / Fetch
61	17:13:06	select ordernumber OrderNumber, priceeach PriceEac...	27 row(s) returned	0.000 sec / 0.000 sec
62	17:13:29	select * from sales_data where city like 'sydney'	0 row(s) returned	0.016 sec / 0.000 sec

-- 55. Calculate the total sales for each contact's first name.

```
select contactfirstname FirstName, round(sum(sales),2) TotalSales  
from sales_data  
group by contactfirstname  
order by contactfirstname;
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays a list of contacts with their first names and total sales. The action history shows two recent queries: one for sales data where city is like 'sydney' and another for the current query.

FirstName	TotalSales
Adrian	151570.98
Akiko	120562.74
Allen	81577.98
Ann	88804.5
Anna	153996.13

Result 60 × Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
62	17:13:29	select * from sales_data where city like 'sydney'	0 row(s) returned	0.016 sec / 0.000 sec
63	17:13:48	select contactfirstname FirstName, round(sum(sales),2)...	72 row(s) returned	0.000 sec / 0.000 sec

-- 56. Identify the orders with a status of 'Disputed'.

```
select *  
from sales_data  
where status like 'disputed';
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays four order entries with the status 'Disputed'. The action history shows two recent queries: one for sales data where status is like 'disputed' and another for the current query.

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	C
10417	66	100	2	7516.08	13	2005-05-13 00:00:00	Disputed	2
10417	45	100	5	5887.35	13	2005-05-13 00:00:00	Disputed	2
10417	56	100	4	9218.16	13	2005-05-13 00:00:00	Disputed	2
10417	21	100	1	3447.78	13	2005-05-13 00:00:00	Disputed	2

sales\_data 61 × Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
63	17:13:48	select contactfirstname FirstName, round(sum(sales),2)...	72 row(s) returned	0.000 sec / 0.000 sec
64	17:14:08	select * from sales_data where status like 'disputed'	14 row(s) returned	0.015 sec / 0.000 sec

-- 57. List the orders with a quantity ordered between 40 and 50.

```
select *  
from sales_data  
where quantityordered between 40 and 50;
```

The screenshot shows a database interface with a results grid and an action history. The results grid displays four order entries with a quantity ordered between 40 and 50. The action history shows two recent queries: one for sales data where status is like 'disputed' and another for the current query.

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10134	41	94.74	2	3884.34	1	2003-07-01 00:00:00	Shipped	3
10145	45	83.26	6	3746.7	25	2003-08-25 00:00:00	Shipped	3
10159	49	100	14	5205.27	10	2003-10-10 00:00:00	Shipped	4
10188	48	100	1	5512.32	18	2003-11-18 00:00:00	Shipped	4

sales\_data 62 × Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
64	17:14:08	select * from sales_data where status like 'disputed'	14 row(s) returned	0.015 sec / 0.000 sec
65	17:14:35	select * from sales_data where quantityordered betwe...	943 row(s) returned	0.000 sec / 0.015 sec

-- 58. Calculate the total sales for each year and month in each territory.

```
select year_id Year,month_id Month,territory Territory,round(sum(sales),2) SalesInTerritory  
from sales_data  
group by year_id,month_id,territory  
order by year_id;
```

	Year	Month	Territory	SalesInTerritory
▶	2003	1	EMEA	110756.3
	2003	1	NA	18997.3
	2003	2	EMEA	115052.43
	2003	2	NA	25783.76
	2003	3	EMEA	60766.92

Result 63 ×

Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
65	17:14:35	select * from sales_data where quantityordered betwe...	943 row(s) returned	0.000 sec / 0.015 sec
66	17:15:10	select year_id Year,month_id Month,territory Territory,r...	87 row(s) returned	0.015 sec / 0.000 sec

-- 59. Identify the orders with a price each higher than \$99.

```
select *  
from sales_data  
where priceeach > 99;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10159	49	100	14	5205.27	10	2003-10-10 00:00:00	Shipped	4
	10188	48	100	1	5512.32	18	2003-11-18 00:00:00	Shipped	4
	10211	41	100	14	4708.44	15	2004-01-15 00:00:00	Shipped	1
	10223	37	100	1	3965.66	20	2004-02-20 00:00:00	Shipped	1

sales\_data 64 ×

Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
66	17:15:10	select year_id Year,month_id Month,territory Territory,r...	87 row(s) returned	0.015 sec / 0.000 sec
67	17:15:33	select * from sales_data where priceeach > 99	1327 row(s) returned	0.000 sec / 0.016 sec

-- 60. List the orders made by customers located in 'Tokyo'.

```
select *  
from sales_data  
where state like 'tokyo';
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10258	32	100	6	7680.64	15	2004-06-15 00:00:00	Shipped	2
	10339	40	68.92	4	2756.8	23	2004-11-23 00:00:00	Shipped	4
	10339	39	76.67	3	2990.13	23	2004-11-23 00:00:00	Shipped	4
	10258	41	100	3	6668.24	15	2004-06-15 00:00:00	Shipped	2

sales\_data 65 ×

Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
67	17:15:33	select * from sales_data where priceeach > 99	1327 row(s) returned	0.000 sec / 0.016 sec
68	17:16:06	select * from sales_data where state like 'tokyo'	32 row(s) returned	0.000 sec / 0.000 sec

-- 61. Calculate the total sales for each product line in the year 2004.

```
select year_id Year,productline ProductLine,round(sum(sales),2) TotalSales  
from sales_data  
group by year_id,productline  
order by year_id;
```

	Year	ProductLine	TotalSales
▶	2003	Classic Cars	1484785.29
	2003	Motorcycles	370895.58
	2003	Planes	272257.6
	2003	Ships	244821.09
	2003	Trains	72802.29

Result 66 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
68	17:16:06	select * from sales_data where state like 'tokyo'	32 row(s) returned	0.000 sec / 0.000 sec
69	17:17:00	select year_id Year,productline ProductLine,round(su...	21 row(s) returned	0.015 sec / 0.000 sec

-- 62. Identify the orders with a status of 'On Hold'

-- and a sales amount greater than \$1000.

```
select *
```

```
from sales_data
```

```
where status like 'on hold' and sales >1000;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR
▶	10414	19	100	3	2764.88	6	2005-05-06 00:00:00	On Hold	2
	10334	26	100	2	3188.12	19	2004-11-19 00:00:00	On Hold	4
	10407	59	100	11	7048.14	22	2005-04-22 00:00:00	On Hold	2
	10407	76	100	2	14082.8	22	2005-04-22 00:00:00	On Hold	2

sales\_data 67 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
69	17:17:00	select year_id Year,productline ProductLine,round(su...	21 row(s) returned	0.015 sec / 0.000 sec
70	17:17:23	select * from sales_data where status like 'on hold' an...	43 row(s) returned	0.000 sec / 0.000 sec

-- 63. List the orders with a quantity ordered between 5 and 10.

```
select *
```

```
from sales_data
```

```
where quantityordered between 5 and 10;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR
▶	10419	10	100	11	1092.2	17	2005-05-17 00:00:00	Shipped	2
	10409	6	100	2	785.64	23	2005-04-23 00:00:00	Shipped	2
	10423	10	88.14	1	881.4	30	2005-05-30 00:00:00	In Process	2
	10407	6	90.19	3	541.14	22	2005-04-22 00:00:00	On Hold	2

sales\_data 68 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
70	17:17:23	select * from sales_data where status like 'on hold' an...	43 row(s) returned	0.000 sec / 0.000 sec
71	17:17:50	select * from sales_data where quantityordered betwe...	4 row(s) returned	0.000 sec / 0.000 sec

-- 64. Calculate the total sales for each product line in each territory.

```
select territory Territory,productline ProductLine,round(sum(sales),2) TotalSales  
from sales_data  
group by territory,productline;
```

The screenshot shows a database query results window. At the top right is a "Result Grid" button. Below it, a message says "Read Only". The main area displays a table with four columns: Territory, ProductLine, and TotalSales. The data is as follows:

Territory	ProductLine	TotalSales
NA	Motorcycles	524549.19
EMEA	Motorcycles	503096.7
APAC	Motorcycles	89968.76
EMEA	Classic Cars	2086994.66
NA	Classic Cars	1406261.44

Below the table, a "Result 69" button is shown. The "Output" section contains an "Action Output" table with two rows of logs:

#	Time	Action	Message	Duration / Fetch
71	17:17:50	select * from sales_data where quantityordered betwe...	4 row(s) returned	0.000 sec / 0.000 sec
72	17:18:21	select territory Territory,productline ProductLine,round(...	28 row(s) returned	0.016 sec / 0.000 sec

-- 65. Identify the orders with a price each less than \$50.

```
select ordernumber OrderNumber,priceeach PriceEach  
from sales_data  
where priceeach < 50;
```

The screenshot shows a database query results window. At the top right is a "Result Grid" button. Below it, a message says "Read Only". The main area displays a table with two columns: OrderNumber and PriceEach. The data is as follows:

OrderNumber	PriceEach
10375	34.91
10388	44.51
10337	48.05
10391	46.9
10371	35.71

Below the table, a "sales\_data 70" button is shown. The "Output" section contains an "Action Output" table with two rows of logs:

#	Time	Action	Message	Duration / Fetch
72	17:18:21	select territory Territory,productline ProductLine,round(...	28 row(s) returned	0.016 sec / 0.000 sec
73	17:18:52	select ordernumber OrderNumber,priceeach PriceEac...	252 row(s) returned	0.016 sec / 0.000 sec

-- 66. List the orders made by customers located in 'germany'.

```
select *  
from sales_data  
where country like 'germany';
```

The screenshot shows a database query results window. At the top right is a "Result Grid" button. Below it, a message says "Read Only". The main area displays a table with ten columns: ORDERNUMBER, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER, SALES, Day, OrderDate, STATUS, Q, and C. The data is as follows:

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q	C
10191	21	100	3	3840.9	20	2003-11-20 00:00:00	Shipped	4	
10300	33	100	5	5521.89	4	2003-10-04 00:00:00	Shipped	4	
10310	33	100	10	6934.62	16	2004-10-16 00:00:00	Shipped	4	
10230	43	100	1	7016.31	15	2004-03-15 00:00:00	Shipped	1	

Below the table, a "sales\_data 71" button is shown. The "Output" section contains an "Action Output" table with two rows of logs:

#	Time	Action	Message	Duration / Fetch
73	17:18:52	select ordernumber OrderNumber,priceeach PriceEac...	252 row(s) returned	0.016 sec / 0.000 sec
74	17:19:16	select * from sales_data where country like 'germany'	62 row(s) returned	0.000 sec / 0.000 sec

-- 67. Calculate the total sales for each quarter in each territory.

```
select territory Territory,qtr_id Quarter,round(sum(sales),2) Sales_in_Quarter  
from sales_data  
group by territory,qtr_id  
order by territory;
```

Territory	Quarter	Sales_in_Quarter
APAC	1	160599.61
APAC	2	166861.66
APAC	3	182764.11
APAC	4	235896.45
EMEA	1	1292025.33

Result 72 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
74	17:19:16	select * from sales_data where country like 'germany'	62 row(s) returned	0.000 sec / 0.000 sec
75	17:19:57	select Territory ,qtr_id Quarter,round(sum(sales),2) Sal...	16 row(s) returned	0.000 sec / 0.000 sec

-- 68. Identify the orders with a status of 'Cancelled'

-- and a sales amount greater than \$5000.

```
select *
```

```
from sales_data
```

```
where status like 'cancelled' and sales >5000;
```

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR
10167	44	100	9	5924.16	23	2003-10-23 00:00:00	Cancelled	4
10167	43	100	1	5763.72	23	2003-10-23 00:00:00	Cancelled	4
10262	49	100	9	6567.96	24	2004-06-24 00:00:00	Cancelled	2
10253	37	100	2	5177.04	1	2004-06-01 00:00:00	Cancelled	2

sales\_data 73 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
75	17:19:57	select Territory ,qtr_id Quarter,round(sum(sales),2) Sal...	16 row(s) returned	0.000 sec / 0.000 sec
76	17:20:20	select * from sales_data where status like 'cancelled' a...	8 row(s) returned	0.016 sec / 0.000 sec

-- 69. List the orders with a quantity ordered equal to the maximum quantity ordered.

```
select *
```

```
from sales_data
```

```
where quantityordered = (select max(quantityordered) from sales_data);
```

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR
10405	97	93.28	5	9048.16	14	2005-04-14 00:00:00	Shipped	2

sales\_data 74 × Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
76	17:20:20	select * from sales_data where status like 'cancelled' a...	8 row(s) returned	0.016 sec / 0.000 sec
77	17:20:46	select * from sales_data where quantityordered = (sele...	1 row(s) returned	0.016 sec / 0.000 sec

-- 70. Calculate the total sales for each product line in each country.

```
select country Country, productline ProductLine, round(sum(sales),2) TotalSales  
from sales_data  
group by country,productline  
order by country;
```

Country	ProductLine	TotalSales
Australia	Classic Cars	193085.54
Australia	Motorcycles	89968.76
Australia	Planes	74853.87
Australia	Ships	4159.76
Australia	Trains	1681.35

Result 75 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
77	17:20:46	select * from sales_data where quantityordered = (sele...	1 row(s) returned	0.016 sec / 0.000 sec
78	17:21:08	select country Country, productline ProductLine, round...	116 row(s) returned	0.016 sec / 0.000 sec

-- 71. Identify the orders with a price each higher than \$80 but less than 90.

```
select *  
from sales_data
```

where priceeach > 80 and priceeach <90;

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
10121	34	81.35	5	2765.9	7	2003-05-07 00:00:00	Shipped	2
10145	45	83.26	6	3746.7	25	2003-08-25 00:00:00	Shipped	3
10180	29	86.13	9	2497.77	11	2003-11-11 00:00:00	Shipped	4
10388	21	86.77	7	1822.17	3	2005-03-03 00:00:00	Shipped	1

sales\_data 76 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
78	17:21:08	select country Country, productline ProductLine, round...	116 row(s) returned	0.016 sec / 0.000 sec
79	17:21:32	select * from sales_data where priceeach > 80 and pri...	286 row(s) returned	0.000 sec / 0.015 sec

-- 72. List the orders made by customers located in 'Rome'.

```
select *  
from sales_data  
where city like 'rome';
```

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_ID	MONTH

sales\_data 77 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
79	17:21:32	select * from sales_data where priceeach > 80 and pri...	286 row(s) returned	0.000 sec / 0.015 sec
80	17:21:55	select * from sales_data where city like 'rome'	0 row(s) returned	0.000 sec / 0.000 sec

-- 73. Calculate the total sales for each year in each country.

```
select year_id Year, country Country, round(sum(sales), 2) TotalSales  
from sales_data  
group by year_id, country  
order by year_id;
```

	Year	Country	TotalSales
▶	2003	Australia	253134.45
	2003	Austria	82117.88
	2003	Belgium	3348.46
	2003	Canada	54609.5
	2003	Denmark	99192.72

Result 78 ×

Read On

Output :

Action Output

#	Time	Action	Message	Duration / Fetch
✓	80	17:21:55 select * from sales_data where city like 'rome'	0 row(s) returned	0.000 sec / 0.000 sec
✓	81	17:22:22 select year_id Year, country Country, round(sum(sales),...	49 row(s) returned	0.016 sec / 0.000 sec

-- 74. Identify the orders with a status of 'In Process' and a sales amount less than \$500.

```
select *  
from sales_data  
where status like 'in process' and sales < 500;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_I
▶	10425	11	43.83	6	482.13	31	2005-05-31 00:00:00	In Process	2

sales\_data 81 ×

Read On

Output :

Action Output

#	Time	Action	Message	Duration / Fetch
✓	83	17:23:12 select * from sales_data where quantityordered = (sele...	2 row(s) returned	0.015 sec / 0.000 sec
✓	84	17:23:36 select * from sales_data where status like 'in process' ...	1 row(s) returned	0.000 sec / 0.000 sec

-- 75. List the orders with a quantity ordered equal to the minimum quantity ordered.

```
select *  
from sales_data  
where quantityordered = (select min(quantityordered) from sales_data);
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_I
▶	10409	6	100	2	785.64	23	2005-04-23 00:00:00	Shipped	2
	10407	6	90.19	3	541.14	22	2005-04-22 00:00:00	On Hold	2

sales\_data 79 ×

Read On

Output :

Action Output

#	Time	Action	Message	Duration / Fetch
✓	81	17:22:22 select year_id Year, country Country, round(sum(sales),...	49 row(s) returned	0.016 sec / 0.000 sec
✓	82	17:22:47 select * from sales_data where quantityordered = (sele...	2 row(s) returned	0.000 sec / 0.000 sec

-- 76. Calculate the total sales for each product line in each state.

```
select State,ProductLine,round(sum(sales),2) TotalSales  
from sales_data  
where state <> ''  
group by state, productline  
order by state;
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

State	ProductLine	TotalSales
BC	Classic Cars	30152.93
BC	Motorcycles	4177.49
BC	Planes	25510.07
BC	Ships	22919.06
BC	Trucks and Buses	32493.22

**Action Output:**

#	Time	Action	Message	Duration / Fetch
84	17:23:36	select * from sales_data where status like 'in process' ...	1 row(s) returned	0.000 sec / 0.000 sec
85	17:24:06	select State,ProductLine,round(sum(sales),2) TotalSal...	88 row(s) returned	0.015 sec / 0.000 sec

-- 77. Identify the orders with a price each less than \$20.

```
select *  
from sales_data  
where priceeach < 20;
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_ID	MONTH_ID
-------------	-----------------	-----------	-----------------	-------	-----	-----------	--------	--------	----------

**Action Output:**

#	Time	Action	Message	Duration / Fetch
85	17:24:06	select State,ProductLine,round(sum(sales),2) TotalSal...	88 row(s) returned	0.015 sec / 0.000 sec
86	17:24:33	select * from sales_data where priceeach < 20	0 row(s) returned	0.015 sec / 0.000 sec

-- 78. List the orders made by customers located in 'Madrid'.

```
select *  
from sales_data  
where city like 'madrid';
```

The screenshot shows a database interface with a results grid and an action output log.

**Result Grid:**

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS
10417	66	100	2	7516.08	13	2005-05-13 00:00:00	Disputed
10126	38	100	11	7329.06	28	2003-05-28 00:00:00	Shipped
10424	50	100	6	12001	31	2005-05-31 00:00:00	In Process
10417	45	100	5	5887.35	13	2005-05-13 00:00:00	Disputed

**Action Output:**

#	Time	Action	Message	Duration / Fetch
86	17:24:33	select * from sales_data where priceeach < 20	0 row(s) returned	0.015 sec / 0.000 sec
87	17:25:03	select * from sales_data where city like 'madrid'	304 row(s) returned	0.016 sec / 0.000 sec

-- 79. Calculate the total sales for each year in each state.

```
select year_id Year, State, round(sum(sales),2) TotalSales  
from sales_data  
where state <> ''  
group by year_id, state  
order by year_id;
```

	Year	State	TotalSales
▶	2003	BC	38662.21
	2003	CA	561196.24
	2003	CT	96565.63
	2003	MA	215792.05
	2003	NH	76114.7

Result 85 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
87	17:25:03	select * from sales_data where city like 'madrid'	304 row(s) returned	0.016 sec / 0.000 sec
88	17:25:46	select year_id Year, State,round(sum(sales),2) TotalSa...	37 row(s) returned	0.000 sec / 0.000 sec

-- 80. Identify the orders with a status of 'Shipped' and a sales amount greater than \$500.

```
select *  
from sales_data  
where status like 'shipped' and sales > 10000;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10150	45	100	8	10993.5	19	2003-09-19 00:00:00	Shipped	3
	10304	47	100	6	10172.7	11	2004-10-11 00:00:00	Shipped	4
	10312	48	100	3	11623.7	21	2004-10-21 00:00:00	Shipped	4
	10403	66	100	9	11886.6	8	2005-04-08 00:00:00	Shipped	2

sales\_data 86 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
88	17:25:46	select year_id Year, State,round(sum(sales),2) TotalSa...	37 row(s) returned	0.000 sec / 0.000 sec
89	17:26:05	select * from sales_data where status like 'shipped' an...	13 row(s) returned	0.000 sec / 0.000 sec

-- 81. List the orders with a quantity ordered greater to the average quantity ordered.

```
select *  
from sales_data  
where quantityordered > (select avg(quantityordered) from sales_data);
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10134	41	94.74	2	3884.34	1	2003-07-01 00:00:00	Shipped	3
	10145	45	83.26	6	3746.7	25	2003-08-25 00:00:00	Shipped	3
	10159	49	100	14	5205.27	10	2003-10-10 00:00:00	Shipped	4
	10168	36	96.66	1	3479.76	28	2003-10-28 00:00:00	Shipped	4

sales\_data 87 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
89	17:26:05	select * from sales_data where status like 'shipped' an...	13 row(s) returned	0.000 sec / 0.000 sec
90	17:26:26	select * from sales_data where quantityordered > (sele...	1348 row(s) returned	0.015 sec / 0.000 sec

-- 82. Calculate the total sales for each product line in each city.

```
select City,ProductLine,round(sum(sales),2) TotalSales  
from sales_data  
where city <> ''  
group by city, productline  
order by city;
```

Result 88 × Read Only

City	ProductLine	TotalSales
Aaarhus	Classic Cars	79375.64
Aaarhus	Ships	1753.06
Aaarhus	Trains	7146.23
Aaarhus	Trucks and Buses	9588.82
Aaarhus	Vintage Cars	2731.8

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
90	17:26:26	select * from sales_data where quantityordered > (sele...	1348 row(s) returned	0.015 sec / 0.000 sec
91	17:26:50	select City,ProductLine,round(sum(sales),2) TotalSales... 334 row(s) returned		0.000 sec / 0.000 sec

-- 83. Identify the orders with a price each equal to \$100.

```
select *  
from sales_data  
where priceeach = 100;
```

sales\_data 89 × Read Only

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_ID	MONTH
10159	49	100	14	5205.27	10	2003-10-10 00:00:00	Shipped	4	
10188	48	100	1	5512.32	18	2003-11-18 00:00:00	Shipped	4	
10211	41	100	14	4708.44	15	2004-01-15 00:00:00	Shipped	1	
10223	37	100	1	3965.66	20	2004-02-20 00:00:00	Shipped	1	

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
91	17:26:50	select City,ProductLine,round(sum(sales),2) TotalSales... 334 row(s) returned		0.000 sec / 0.000 sec
92	17:27:15	select * from sales_data where priceeach = 100	1304 row(s) returned	0.000 sec / 0.015 sec

-- 84. List the orders made by customers located in 'Moscow'.

```
select *  
from sales_data  
where city like 'moscow';
```

sales\_data 90 × Read Only

ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_ID	MONTH
-------------	-----------------	-----------	-----------------	-------	-----	-----------	--------	--------	-------

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
92	17:27:15	select * from sales_data where priceeach = 100	1304 row(s) returned	0.000 sec / 0.015 sec
93	17:27:33	select * from sales_data where city like 'moscow'	0 row(s) returned	0.015 sec / 0.000 sec

-- 85. Calculate the total sales for each year in each city.

```
select year_id Year, City, round(sum(sales),2) TotalSales  
from sales_data  
group by year_id,city  
order by year_id;
```

	Year	City	TotalSales
▶	2003	Aarhus	40321.61
	2003	Barcelona	48228.51
	2003	Bergamo	96259.03
	2003	Bergen	95277.18
	2003	Boras	48710.92

Result 91 × Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
93	17:27:33	select * from sales_data where city like 'moscow'	0 row(s) returned	0.015 sec / 0.000 sec
94	17:27:55	select year_id Year, City, round(sum(sales),2) TotalSal...	162 row(s) returned	0.016 sec / 0.000 sec

-- 86. Identify the orders with a status of 'Resolved' and a sales amount greater than \$5000 and less than \$8000.

```
select *  
from sales_data  
where status like 'resolved' and sales between 5000 and 8000;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10164	49	100	6	6563.06	21	2003-10-21 00:00:00	Resolved	4
	10164	45	100	3	5012.55	21	2003-10-21 00:00:00	Resolved	4
	10367	36	100	2	5018.4	12	2005-01-12 00:00:00	Resolved	1
	10386	37	100	10	5017.57	1	2005-03-01 00:00:00	Resolved	1

sales\_data 92 × Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
94	17:27:55	select year_id Year, City, round(sum(sales),2) TotalSal...	162 row(s) returned	0.016 sec / 0.000 sec
95	17:37:11	select * from sales_data where status like 'resolved' an...	5 row(s) returned	0.000 sec / 0.000 sec

-- 87. List the orders with a quantity ordered equal to the average quantity ordered.

```
select *  
from sales_data  
where quantityordered = (select round(avg(quantityordered),0) from sales_data);
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10215	35	100	3	6075.3	29	2004-01-29 00:00:00	Shipped	1
	10370	35	65.63	4	2297.05	20	2005-01-20 00:00:00	Shipped	1
	10272	35	100	2	5818.4	20	2004-07-20 00:00:00	Shipped	3
	10169	35	100	13	4639.25	4	2003-11-04 00:00:00	Shipped	4

sales\_data 93 × Read On

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
95	17:37:11	select * from sales_data where status like 'resolved' an...	5 row(s) returned	0.000 sec / 0.000 sec
96	17:44:15	select * from sales_data where quantityordered = (sele...	71 row(s) returned	0.000 sec / 0.000 sec

-- 88. Calculate the total sales for each product line in each postal code.

```
select PostalCode,ProductLine, round(sum(sales),2) TotalSales  
from sales_data  
where postalcode <> ""  
group by postalcode,productline  
order by postalcode;
```

Result 94 × Read On

	PostalCode	ProductLine	TotalSales
▶	10022	Classic Cars	260803.74
	10022	Motorcycles	98297.68
	10022	Planes	3789.72
	10022	Ships	39640.76
	10022	Trains	16640.15

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
✓	96 17:44:15	select * from sales_data where quantityordered = (sele...	71 row(s) returned	0.000 sec / 0.000 sec
✓	97 17:44:41	select PostalCode,ProductLine, round(sum(sales),2) T...	335 row(s) returned	0.016 sec / 0.000 sec

-- 89 Identify the orders with a price each is from 60 to 80.

```
select *  
from sales_data  
where priceeach between 60 and 80;
```

sales\_data 95 × Read On

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10361	20	72.55	13	1451	17	2004-12-17 00:00:00	Shipped	4
	10388	42	76.36	4	3207.12	3	2005-03-03 00:00:00	Shipped	1
	10339	40	68.92	4	2756.8	23	2004-11-23 00:00:00	Shipped	4
	10339	39	76.67	3	2990.13	23	2004-11-23 00:00:00	Shipped	4

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
✓	97 17:44:41	select PostalCode,ProductLine, round(sum(sales),2) T...	335 row(s) returned	0.016 sec / 0.000 sec
✓	98 17:45:02	select * from sales_data where priceeach between 60...	541 row(s) returned	0.000 sec / 0.016 sec

-- 90. List the orders made by customers located in 'Beijing'.

```
select *  
from sales_data  
where city like 'beijing';
```

sales\_data 96 × Read On

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	QTR_ID	MONTH_ID
--	-------------	-----------------	-----------	-----------------	-------	-----	-----------	--------	--------	----------

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
✓	98 17:45:02	select * from sales_data where priceeach between 60...	541 row(s) returned	0.000 sec / 0.016 sec
✓	99 17:45:28	select * from sales_data where city like 'beijing'	0 row(s) returned	0.000 sec / 0.000 sec

-- 91. Calculate the total sales for each year in each postal code.

```
select year_id Year,PostalCode, round(sum(sales),2) TotalSales  
from sales_data  
where postalcode <> ''  
group by year_id,postalcode  
order by year_id;
```

	Year	PostalCode	TotalSales
▶	2003	10022	215488.21
	2003	1227 MM	78086.98
	2003	13008	52481.84
	2003	1734	58871.11
	2003	2060	88983.71

Result 97 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
99	17:45:28	select * from sales_data where city like 'beijing'	0 row(s) returned	0.000 sec / 0.000 sec
100	17:45:53	select year_id Year,PostalCode, round(sum(sales),2) T...	160 row(s) returned	0.000 sec / 0.000 sec

-- 92. Identify the orders with a status of 'Disputed' and a sales amount greater than \$5000.

```
select *  
from sales_data  
where status like 'disputed' and sales > 5000;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q...
▶	10417	66	100	2	7516.08	13	2005-05-13 00:00:00	Disputed	2
	10417	45	100	5	5887.35	13	2005-05-13 00:00:00	Disputed	2
	10417	56	100	4	9218.16	13	2005-05-13 00:00:00	Disputed	2
	10406	61	100	3	8374.69	15	2005-04-15 00:00:00	Disputed	2

sales\_data 98 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
100	17:45:53	select year_id Year,PostalCode, round(sum(sales),2) T...	160 row(s) returned	0.000 sec / 0.000 sec
101	17:46:12	select * from sales_data where status like 'disputed' a...	7 row(s) returned	0.000 sec / 0.000 sec

-- 93. List the orders with a quantity ordered less than the average of the quantity ordered.

```
select *  
from sales_data  
where sales < (select avg(sales) from sales_data);
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q...
▶	10107	30	95.7	2	2871	24	2003-02-24 00:00:00	Shipped	1
	10121	34	81.35	5	2765.9	7	2003-05-07 00:00:00	Shipped	2
	10168	36	96.66	1	3479.76	28	2003-10-28 00:00:00	Shipped	4
	10180	29	86.13	9	2497.77	11	2003-11-11 00:00:00	Shipped	4

sales\_data 99 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
101	17:46:12	select * from sales_data where status like 'disputed' a...	7 row(s) returned	0.000 sec / 0.000 sec
102	17:46:36	select * from sales_data where sales < (select avg(sal...	1655 row(s) returned	0.000 sec / 0.016 sec

-- 94. Calculate the total sales for each product line in each country in the year 2005 and 2004.

```
select year_id Year,ProductLine,round(sum(sales),2) TotalSales  
from sales_data  
where year_id in (2004,2005)  
group by year_id,productline  
order by year_id;
```

	Year	ProductLine	TotalSales
▶	2004	Classic Cars	1762257.09
	2004	Motorcycles	560545.23
	2004	Planes	502671.8
	2004	Ships	341437.97
	2004	Trains	116523.85

Result 100 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
102	17:46:36	select * from sales_data where sales < (select avg(sal...	1655 row(s) returned	0.000 sec / 0.016 sec
103	17:47:02	select year_id Year,ProductLine,round(sum(sales),2) T...	14 row(s) returned	0.015 sec / 0.000 sec

-- 95. Identify the orders placed between the months april to august in the year 2004

```
select *  
from sales_data  
where (month(orderdate) between 4 and 8) and year_id like 2004  
order by orderdate;
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10235	24	76.03	3	1824.72	2	2004-04-02 00:00:00	Shipped	2
	10235	23	96.29	5	2214.67	2	2004-04-02 00:00:00	Shipped	2
	10235	33	60.05	12	1981.65	2	2004-04-02 00:00:00	Shipped	2
	10235	40	81.14	4	3245.6	2	2004-04-02 00:00:00	Shipped	2

sales\_data 101 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
103	17:47:02	select year_id Year,ProductLine,round(sum(sales),2) T...	14 row(s) returned	0.015 sec / 0.000 sec
104	17:47:23	select * from sales_data where (month(orderdate) betw...	447 row(s) returned	0.016 sec / 0.000 sec

-- 96. List the orders made by customers located in 'Japan'.

```
select *  
from sales_data  
where Country like 'japan';
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	Day	OrderDate	STATUS	Q
▶	10258	32	100	6	7680.64	15	2004-06-15 00:00:00	Shipped	2
	10210	23	100	2	3009.09	12	2004-01-12 00:00:00	Shipped	1
	10339	40	68.92	4	2756.8	23	2004-11-23 00:00:00	Shipped	4
	10210	34	100	1	6123.4	12	2004-01-12 00:00:00	Shipped	1

sales\_data 103 × Read Only

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
105	17:47:55	select * from sales_data where Country like 'japan'	52 row(s) returned	0.000 sec / 0.000 sec
106	17:48:07	select * from sales_data where Country like 'japan'	52 row(s) returned	0.016 sec / 0.000 sec

-- 97. Calculate the total sales for each product line in each state in the year 2004.

```
select ProductLine,round(sum(sales),2) Total_Sales  
from sales_data  
where year_id like 2004  
group by year_id, productline;
```

The screenshot shows a database query results window. At the top, there is a table titled "Result 104" with two columns: "ProductLine" and "Total\_Sales". The data is as follows:

ProductLine	Total_Sales
Motorcycles	560545.23
Classic Cars	1762257.09
Trucks and Buses	529302.89
Vintage Cars	911423.77
Planes	502671.8

Below the table, there is an "Output" section with an "Action Output" table. This table lists two actions with their details:

#	Time	Action	Message	Duration / Fetch
106	17:48:07	select * from sales_data where Country like 'japan'	52 row(s) returned	0.016 sec / 0.000 sec
107	17:48:36	select ProductLine,round(sum(sales),2) Total_Sales fro...	7 row(s) returned	0.000 sec / 0.000 sec

-- 98. Identify the orders with a status of 'Cancelled' and a sales amount greater than \$5000

```
select OrderNumber,Sales  
from sales_data  
where status like 'cancelled' and sales >5000;
```

The screenshot shows a database query results window. At the top, there is a table titled "sales\_data 105" with two columns: "OrderNumber" and "Sales". The data is as follows:

OrderNumber	Sales
10167	5924.16
10167	5763.72
10262	6567.96
10253	5177.04
10253	6773.6

Below the table, there is an "Output" section with an "Action Output" table. This table lists two actions with their details:

#	Time	Action	Message	Duration / Fetch
107	17:48:36	select ProductLine,round(sum(sales),2) Total_Sales fro...	7 row(s) returned	0.000 sec / 0.000 sec
108	17:48:56	select OrderNumber,Sales from sales_data where stat...	8 row(s) returned	0.000 sec / 0.000 sec

-- 99. Find the customer with the highest sales amount and total quantity ordered

```
select customername Name, sum(quantityordered) TotalQuantityOrdered,round(sum(sales),2) TotalSales  
from sales_data  
group by customername  
order by totalsales desc limit 1;
```

The screenshot shows a database query results window. At the top, there is a table titled "Result 106" with three columns: "Name", "TotalQuantityOrdered", and "TotalSales". The data is as follows:

Name	TotalQuantityOrdered	TotalSales
Euro Shopping Channel	9327	912294.11

Below the table, there is an "Output" section with an "Action Output" table. This table lists two actions with their details:

#	Time	Action	Message	Duration / Fetch
108	17:48:56	select OrderNumber,Sales from sales_data where stat...	8 row(s) returned	0.000 sec / 0.000 sec
109	17:49:18	select customername Name, sum(quantityordered) Tot...	1 row(s) returned	0.000 sec / 0.000 sec

- 100.Calculate the total sales for each product line in each city in the year 2004.

```
select ProductLine,City,round(sum(sales),2) TotalSales  
from sales_data  
where city <> '' and year_id like 2004  
group by productline, city  
order by productline;
```

The screenshot shows a database query results interface. At the top, there is a yellow header bar with the text "- 100.Calculate the total sales for each product line in each city in the year 2004.". Below this, the SQL query is displayed. To the right of the query, there is a "Result Grid" button and a "Read Only" status indicator. The main area shows a table with the following data:

	ProductLine	City	TotalSales
▶	Classic Cars	Aarhus	39054.03
	Classic Cars	Allentown	36144.2
	Classic Cars	Boras	14249.05
	Classic Cars	Boston	9751.49
	Classic Cars	Brickhaven	35409.95

Below the table, there is a "Result 107" message and a "Read Only" status. Under the "Output" tab, there is a "Action Output" section showing the following log entries:

#	Time	Action	Message	Duration / Fetch
109	17:49:18	select customername Name, sum(quantityordered) Tot...	1 row(s) returned	0.000 sec / 0.000 sec
110	17:49:44	select ProductLine,City,round(sum(sales),2) TotalSales...	230 row(s) returned	0.000 sec / 0.000 sec