

SQL File 70* SQL File 71* SQL File 72* SQL File 73* SQL File 74* SQL File 75* SQL



```
4      AVG(tax) AS total_tax
5  FROM sales
6  GROUP BY customer_type
7  ORDER BY total_tax;
8  |
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	customer_type	total_tax
▶	Normal	15.149339
	Member	15.609840

```
3      city,  
4      ROUND(AVG(tax), 2) AS avg_tax  
5  FROM sales  
6  GROUP BY city  
7  ORDER BY avg_tax DESC;
```

Result Grid



Filter Rows:

Export:



W

city	avg_tax
Naypyitaw	16.05
Mandalay	15.23
Yangon	14.87

1 -- Which of the customer types brings the most revenue?
2 • SELECT
3 customer_type,
4 SUM(total) AS total_revenue
5 FROM sales
6 GROUP BY customer_type
7 ORDER BY total_revenue;

Result Grid |   Filter Rows: | Export:  | Wrap Cell Content: 

	customer_type	total_revenue
▶	Normal	158743.62
	Member	164223.81



```
2  -- Number of sales made in each time of the day per weekday
3  •  SELECT
4      time_of_day,
5      COUNT(*) AS total_sales
6  FROM sales
7  WHERE day_name = "Sunday"
8  GROUP BY time_of_day
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	time_of_day	total_sales
▶	Evening	58
	Afternoon	53
	Morning	22

QL File 67* SQL File 68* SQL File 69* SQL File 70* SQL File 71* SQL File 72* x



```
1  -- Which day of the week has the best average ratings per branch?
2  • SELECT
3      day_name,
4      COUNT(day_name) total_sales
5  FROM sales
6  WHERE branch = "r"
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	day_name	total_sales
▶	Saturday	54
	Tuesday	54
	Wednesday	50
	Thursday	48
	Sunday	46
	Friday	38
	Monday	38

QL File 66*

SQL File 67*

SQL File 68*

SQL File 69*

SQL File 70*

SQL File 71* x



Limit to 1000 rows

```
1  -- Which day fo the week has the best avg ratings?
2  • SELECT
3      day_name,
4      AVG(rating) AS avg_rating
5  FROM sales
6  GROUP BY day_name
7  ORDER BY avg_rating DESC;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	day_name	avg_rating
▶	Monday	7.15360
	Friday	7.07626
	Sunday	7.01128
	Tuesday	7.00316
	Saturday	6.90183

Result 1

QL File 65* SQL File 66* SQL File 67* SQL File 68* SQL File 69* SQL File 70* x



```
1  -- Which time of the day do customers give most ratings per branch?
2  • SELECT
3      time_of_day,
4      AVG(rating) AS avg_rating
5  FROM sales
6  WHERE branch = "A"
7  GROUP BY time_of_day
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	time_of_day	avg_rating
▶	Afternoon	7.18889
	Morning	7.00548
	Evening	6.89362

SQL File 64 SQL File 65 SQL File 66 SQL File 67 SQL File 68 SQL File 69



```
1  -- Which time of the day do customers give most ratings?
2  • SELECT
3      time_of_day,
4      AVG(rating) AS avg_rating
5  FROM sales
6  GROUP BY time_of_day
7  ORDER BY avg_rating DESC;
```



Result Grid



Filter Rows:

Export:



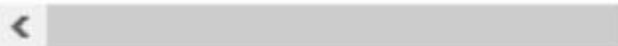
Wrap Cell Content:



	time_of_day	avg_rating
▶	Afternoon	7.03130
	Morning	6.96073
	Evening	6.92685



```
4      COUNT(*) as gender_cnt
5  FROM sales
6  WHERE branch = "C"
7  GROUP BY gender
8  ORDER BY gender_cnt DESC;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	gender	gender_cnt
▶	Female	178
	Male	150

QL File 62* SQL File 63* SQL File 64* SQL File 65* SQL File 66* SQL File 67*



```
1  -- What is the gender of most of the customers?
2  •  SELECT
3      gender,
4      COUNT(*) as gender_cnt
5  FROM sales
6  GROUP BY gender
7  ORDER BY gender_cnt DESC;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	gender	gender_cnt
▶	Female	501
	Male	499



```
1  -- Which customer type buys the most?
2  •  SELECT
3      customer_type,
4      COUNT(*)
5  FROM sales
6  GROUP BY customer_type;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	customer_type	COUNT(*)
▶	Member	501
	Normal	499



```
1  -- What is the most common customer type?
2  •  SELECT
3      customer_type,
4      count(*) as count
5  FROM sales
6  GROUP BY customer_type
7  ORDER BY count DESC;
```



	customer_type	count
▶	Member	501
	Normal	499

```
1  -- How many unique payment methods does the data have?
2  • SELECT
3      DISTINCT payment
4  FROM sales;
5  |
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	payment
▶	Ewallet
	Cash
	Credit card

```
2
3      -- How many unique customer types does the data have?
4 •    SELECT
5          DISTINCT customer_type
6      FROM sales;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	customer_type
▶	Member
	Normal

sales 1 ×

1 -- What is the average rating of each product line
2 • SELECT
3 ROUND(AVG(rating), 2) as avg_rating,
4 product_line
5 FROM sales
6 GROUP BY product_line
7 ORDER BY avg_rating DESC;

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	avg_rating	product_line
▶	7.11	Food and beverages
	7.03	Fashion accessories
	7.00	Health and beauty
	6.92	Electronic accessories
	6.92	Sports and travel



```

2 • SELECT
3     gender,
4     product_line,
5     COUNT(gender) AS total_cnt
6 FROM sales
7 GROUP BY gender, product_line
8 ORDER BY total_cnt DESC;
    
```



Result Grid



Filter Rows:

Export:



Wrap Cell Cont

	gender	product_line	total_cnt
▶	Female	Fashion accessories	96
	Female	Food and beverages	90
	Male	Health and beauty	88
	Female	Sports and travel	88
	Male	Electronic accessories	86


```
3      branch,  
4      SUM(quantity) AS qty  
5  FROM sales  
6  GROUP BY branch  
7  HAVING SUM(quantity) > (SELECT AVG(quantity) FROM sales);
```

< Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	branch	qty
▶	A	1859
	C	1831
	B	1820



Limit to 1000 rows

```
1 • SELECT
2     product_line,
3     CASE
4         WHEN AVG(quantity) > 6 THEN "Good"
5         ELSE "Bad"
6     END AS remark
7 FROM sales
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content

	product_line	remark
▶	Health and beauty	Bad
	Electronic accessories	Bad
	Home and lifestyle	Bad
	Sports and travel	Bad
	Food and beverages	Bad
	Fashion accessories	Bad

Result 1

3

4 • SELECT

5 AVG(quantity) AS avg_qty

6 FROM sales;

7

<

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	avg_qty
▶	5.5100



```
2 • SELECT
3     branch,
4     city,
5     SUM(total) AS total_revenue
6 FROM sales
7 GROUP BY city, branch
8 ORDER BY total_revenue;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	branch	city	total_revenue
▶	B	Mandalay	106198.00
	A	Yangon	106200.57
	C	Naypyitaw	110568.86



2 • SELECT

```
4 SUM(total) as total_revenue
```

```
6 GROUP BY product_line
```

```
7 ORDER BY total_revenue DESC;
```



--	--

 \overline{IA}

	product_line	total_revenue
	Food and beverages	56144.96
▶	Sports and travel	55123.00
	Electronic accessories	54337.64
	Fashion accessories	54306.03
	Home and lifestyle	53861.96

QL File 49*

SQL File 50*

SQL File 51*

SQL File 52*

SQL File 53*



Limit to 1000 rows



```
1  -- What month had the largest COGS?
2  •  SELECT
3      month_name AS month,
4      SUM(cogs) AS cogs
5  FROM sales
6  GROUP BY month_name
7  ORDER BY cogs;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:

	month	cogs
▶	February	92589.88
	March	104243.34
	January	110754.16



```
1  -- What is the total revenue by month
2  •  SELECT
3      month_name AS month,
4      SUM(total) AS total_revenue
5  FROM sales
6  GROUP BY month_name
7  ORDER BY total_revenue;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Co

	month	total_revenue
▶	February	97219.58
	March	109455.74
	January	116292.11



```
1  -- What is the most selling product line
2  •  SELECT
3      SUM(quantity) as qty,
4      product_line
5  FROM sales
6  GROUP BY product_line
7  ORDER BY qty DESC;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



qty

product_line

▶ 971 Electronic accessories

952 Food and beverages

920 Sports and travel

911 Home and lifestyle

902 Fashion accessories

854 Health and beauty

Result 1 x

```
1  -- How many unique product lines does the data have
2  • SELECT
3      DISTINCT product_line
4  FROM sales;
```

< Result Grid |   Filter Rows: | Export:  | Wrap Cell Co

	product_line
▶	Health and beauty
	Electronic accessories
	Home and lifestyle
	Sports and travel
	Food and beverages
	Fashion accessories

1 -- In which city is each branch?
2 • SELECT
3 DISTINCT city,
4 branch
5 FROM sales;
6

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	city	branch
▶	Yangon	A
	Naypyitaw	C
	Mandalay	B

```
1  -- How many unique cities does the data have?
2  • SELECT
3      DISTINCT city
4  FROM sales;
5
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:

	city
	Yangon
	Naypyitaw
▶	Mandalay



```
1      -- Add month_name column
2  •    SELECT
3          date,
4          MONTHNAME(date)
5  FROM sales;
```



Result Grid



Filter Rows:

Export:



Wrap

	date	MONTHNAME(date)
▶	2019-01-05	January
	2019-03-08	March
	2019-03-03	March
	2019-01-27	January
	2019-02-08	February
	2019-03-25	March
	2019-02-25	February



```
UPDATE sales
```

```
SET day_name = DAYNAME(date);
```




```

1      -- Add day_name column
2  •    SELECT
3          date,
4          DAYNAME(date)
5  FROM sales;
    
```



Result Grid Filter Rows: Export: Wrap Cell Content






	date	DAYNAME(date)
▶	2019-01-05	Saturday
	2019-03-08	Friday
	2019-03-03	Sunday
	2019-01-27	Sunday
	2019-02-08	Friday
	2019-03-25	Monday
	2019-02-25	Monday
	2019-02-24	Sunday



1 • UPDATE sales
2 SET time_of_day = (
3 CASE
4 WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"
5 WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"
6 ELSE "Evening"
7 END
8);



```
1 -- Add the time_of_day column
2 SELECT
3     time,
4     (CASE
5         WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"
6         WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"
7         ELSE "Evening"
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 	Fetch rows: 
	time	time_of_day			
▶	13:08:00	Afternoon			
	10:29:00	Morning			
	13:23:00	Afternoon			
	20:33:00	Evening			
	10:37:00	Morning			

Result 1 x

sales_data

sales

sales

sales

sales

sales

sales

SQL File 33



```
1  -- Create table
2  • CREATE TABLE IF NOT EXISTS sales(
3      invoice_id VARCHAR(30) NOT NULL PRIMARY KEY,
4      branch VARCHAR(5) NOT NULL,
5      city VARCHAR(30) NOT NULL,
6      customer_type VARCHAR(30) NOT NULL,
7      gender VARCHAR(30) NOT NULL,
8      product_line VARCHAR(100) NOT NULL,
9      unit_price DECIMAL(10,2) NOT NULL,
10     quantity INT NOT NULL,
11     tax_pct FLOAT(6,4) NOT NULL,
12     total DECIMAL(12, 4) NOT NULL,
13     date DATETIME NOT NULL,
14     time TIME NOT NULL,
15     payment VARCHAR(15) NOT NULL,
16     cogs DECIMAL(10,2) NOT NULL,
17     gross_margin_pct FLOAT(11,9),
18     gross_income DECIMAL(12, 4),
19     rating FLOAT(2, 1)
20 );
```



Output

Sales_data Sales Sales Sales Sales Sales Sales									
Limit to 1000 rows									
1 • SELECT * FROM walmart_sales.sales;									
Result Grid									
Filter Rows: Export: Wrap Cell Content: Fetch rows:									
	invoice_id	branch	city	customer_type	gender	product_line	unit_price	quantity	t
▶	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	2
	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3
	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	1
	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	2
	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	3
	699-14-3026	C	Naypyitaw	Normal	Male	Electronic accessories	85.39	7	2
	355-53-5943	A	Yangon	Member	Female	Electronic accessories	68.84	6	2
	315-22-5665	C	Naypyitaw	Normal	Female	Home and lifestyle	73.56	10	3
	665-32-9167	A	Yangon	Member	Female	Health and beauty	36.26	2	3
	692-92-5582	B	Mandalay	Member	Female	Food and beverages	54.84	3	8
	351-62-0822	B	Mandalay	Member	Female	Fashion accessories	14.48	4	2
	529-56-3974	B	Mandalay	Member	Male	Electronic accessories	25.51	4	5
	365-64-0515	A	Yangon	Normal	Female	Electronic accessories	46.95	5	1
	353-56-3600	A	Yangon	Normal	Male	Food and beverages	42.40	10	2