

**-- Create database**

```
CREATE DATABASE IF NOT EXISTS walmartSales;
```

```
USE walmartSales;
```

**-- Create table**

```
CREATE TABLE IF NOT EXISTS sales(  
    invoice_id VARCHAR(30) NOT NULL PRIMARY KEY,  
    branch VARCHAR(5) NOT NULL,  
    city VARCHAR(30) NOT NULL,  
    customer_type VARCHAR(30) NOT NULL,  
    gender VARCHAR(30) NOT NULL,  
    product_line VARCHAR(100) NOT NULL,  
    unit_price DECIMAL(10,2) NOT NULL,  
    quantity INT NOT NULL,  
    tax_pct FLOAT(6,4) NOT NULL,  
    total DECIMAL(12, 4) NOT NULL,  
    date DATETIME NOT NULL,  
    time TIME NOT NULL,  
    payment VARCHAR(15) NOT NULL,  
    cogs DECIMAL(10,2) NOT NULL,  
    gross_margin_pct FLOAT(11,9),  
    gross_income DECIMAL(12, 4),  
    rating FLOAT(2, 1)  
);
```

**-- Data cleaning**

```
SELECT * FROM sales;
```

**-- Add the time\_of\_day column**

```
SELECT
    time,
    (CASE
        WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"
        WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"
        ELSE "Evening"
    END) AS time_of_day
FROM sales;
```

```
ALTER TABLE sales ADD COLUMN time_of_day VARCHAR(20);
```

```
UPDATE sales
```

```
SET time_of_day = (
    CASE
        WHEN `time` BETWEEN "00:00:00" AND "12:00:00" THEN "Morning"
        WHEN `time` BETWEEN "12:01:00" AND "16:00:00" THEN "Afternoon"
        ELSE "Evening"
    END
);
```

**-- Add day\_name column**

```
ALTER TABLE sales ADD COLUMN day_name VARCHAR(10);
```

```
UPDATE sales
```

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

-- Add month\_name column

SELECT

date, MONTHNAME(date)

FROM sales;

ALTER TABLE sales ADD COLUMN month\_name VARCHAR(10);

UPDATE sales

SET month\_name = MONTHNAME(date);

---

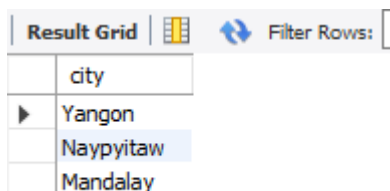
## -- GENERIC QUESTIONS

-- 1 How many unique cities does the data have?

SELECT DISTINCT city

FROM sales;

RESULT:



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query for distinct cities. The first row is the header 'city'. The subsequent rows are 'Yangon', 'Naypyitaw', and 'Mandalay'. The 'Naypyitaw' row is currently selected.

city
Yangon
Naypyitaw
Mandalay

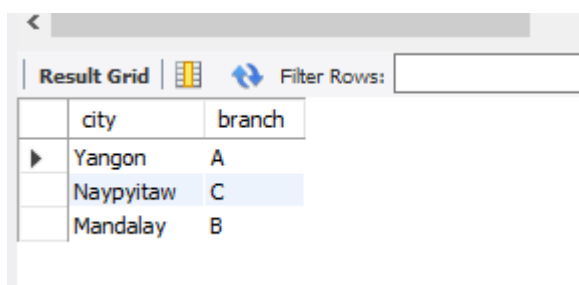
-- 2 In which city is each branch?

SELECT

DISTINCT city,branch

FROM sales;

RESULT:



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query for distinct city and branch combinations. The first row is the header 'city, branch'. The subsequent rows are 'Yangon, A', 'Naypyitaw, C', and 'Mandalay, B'. The 'Naypyitaw, C' row is currently selected.

city	branch
Yangon	A
Naypyitaw	C
Mandalay	B

---

## -- PRODUCT QUESTIONS

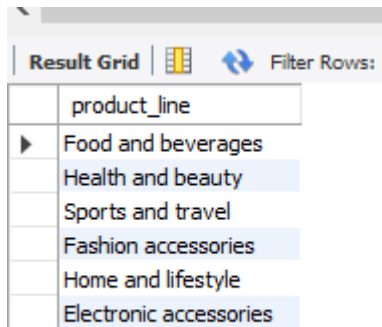
-- 1 How many unique product lines does the data have?

SELECT

DISTINCT product\_line

FROM sales;

RESULT:



The screenshot shows a database query result grid with a header row and six data rows. The header row has a single column labeled 'product\_line'. The data rows list the following product lines: 'Food and beverages', 'Health and beauty', 'Sports and travel', 'Fashion accessories', 'Home and lifestyle', and 'Electronic accessories'. The 'Health and beauty' and 'Electronic accessories' rows are highlighted in blue.

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

-- 2 What is the most selling product line

SELECT

SUM(quantity) as qty,

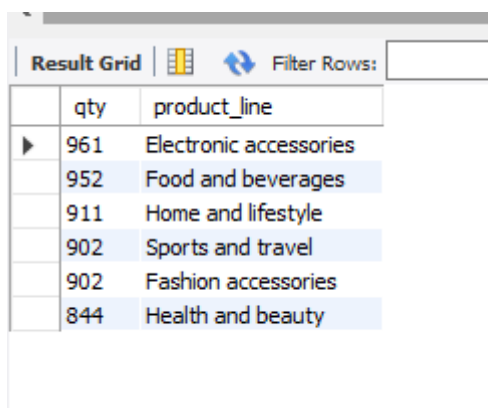
product\_line

FROM sales

GROUP BY product\_line

ORDER BY qty DESC;

RESULT:



The screenshot shows a database query result grid with a header row and seven data rows. The header row has two columns: 'qty' and 'product\_line'. The data rows show the following quantities and product lines: 961 for 'Electronic accessories', 952 for 'Food and beverages', 911 for 'Home and lifestyle', 902 for 'Sports and travel', 902 for 'Fashion accessories', and 844 for 'Health and beauty'. The 'Food and beverages', 'Sports and travel', 'Fashion accessories', and 'Health and beauty' rows are highlighted in blue.

qty	product_line
961	Electronic accessories
952	Food and beverages
911	Home and lifestyle
902	Sports and travel
902	Fashion accessories
844	Health and beauty

-- 3 What is the total revenue by month

SELECT

month\_name AS month,

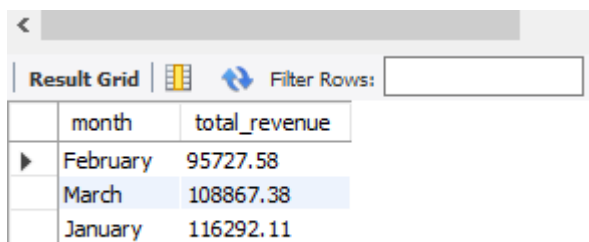
SUM(total) AS total\_revenue

FROM sales

GROUP BY month\_name

ORDER BY total\_revenue;

RESULT:



The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'month' and 'total\_revenue'. The data is sorted by total revenue in descending order. The rows are: February (95727.58), March (108867.38), and January (116292.11). The March row is highlighted in blue.

	month	total_revenue
▶	February	95727.58
	March	108867.38
	January	116292.11

-- 4 What month had the largest COGS?

SELECT

month\_name AS month,

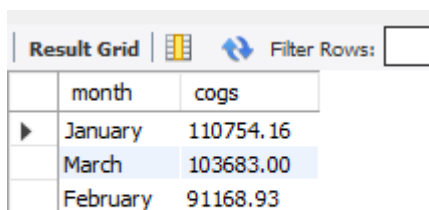
SUM(cogs) AS cogs

FROM sales

GROUP BY month\_name

ORDER BY cogs;

RESULT:



The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'month' and 'cogs'. The data is sorted by COGS in descending order. The rows are: January (110754.16), March (103683.00), and February (91168.93). The March row is highlighted in blue.

	month	cogs
▶	January	110754.16
	March	103683.00
	February	91168.93

-- 5 What product line had the largest revenue?

SELECT

product\_line,

SUM(total) as total\_revenue

FROM sales

GROUP BY product\_line

ORDER BY total\_revenue DESC;

RESULT:

Result Grid	Filter Rows:
product_line	total_revenue
Food and beverages	56144.96
Fashion accessories	54306.03
Sports and travel	53936.30
Home and lifestyle	53861.96
Electronic accessories	53783.34
Health and beauty	48854.48

-- 6 What is the city with the largest revenue?

SELECT

branch,

city,



SUM(total) AS total\_revenue

FROM sales

GROUP BY city, branch

ORDER BY total\_revenue;

RESULT:

Result Grid				Filter Rows:	
	branch	city	total_revenue		
▶	B	Mandalay	104534.93		
	A	Yangon	105861.21		
	C	Naypyitaw	110490.93		

-- 7 What product line had the largest VAT?

SELECT

product\_line,

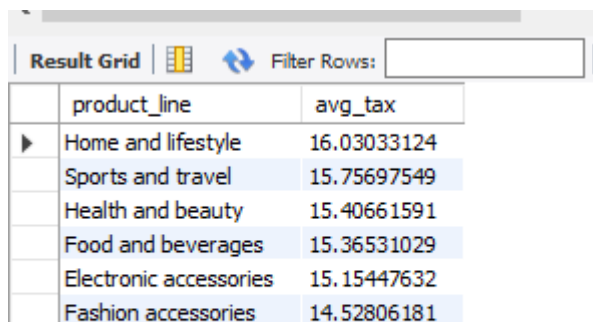
AVG(tax\_pct) as avg\_tax

FROM sales

GROUP BY product\_line

ORDER BY avg\_tax DESC;

RESULT:



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. Below the header, there are six rows of data. The first row is 'Home and lifestyle' with an 'avg\_tax' of 16.03033124. The subsequent rows are 'Sports and travel' (15.75697549), 'Health and beauty' (15.40661591), 'Food and beverages' (15.36531029), 'Electronic accessories' (15.15447632), and 'Fashion accessories' (14.52806181). The rows are alternatingly highlighted in white and light blue.

	product_line	avg_tax
▶	Home and lifestyle	16.03033124
	Sports and travel	15.75697549
	Health and beauty	15.40661591
	Food and beverages	15.36531029
	Electronic accessories	15.15447632
	Fashion accessories	14.52806181

-- 8 Fetch each product line and add a column to those product

-- line showing "Good", "Bad". Good if its greater than average sales

SELECT

AVG(quantity) AS avg\_qnty

FROM sales;

SELECT

product\_line,

CASE

WHEN AVG(quantity) > 6 THEN "Good"

ELSE "Bad"

END AS remark

FROM sales

GROUP BY product\_line;

RESULT:

<		
Result Grid		
Filter Rows:		
	product_line	remark
▶	Food and beverages	Bad
	Health and beauty	Bad
	Sports and travel	Bad
	Fashion accessories	Bad
	Home and lifestyle	Bad
	Electronic accessories	Bad

-- 9 Which branch sold more products than average product sold?

SELECT

branch,

SUM(quantity) AS qnty

FROM sales

GROUP BY branch

HAVING SUM(quantity) > (SELECT AVG(quantity) FROM sales);

RESULT:

<		
Result Grid		
Filter Rows:		
	branch	qnty
▶	A	1849
	C	1828
	B	1795



-- 10 What is the most common product line by gender

SELECT

gender,

product\_line,

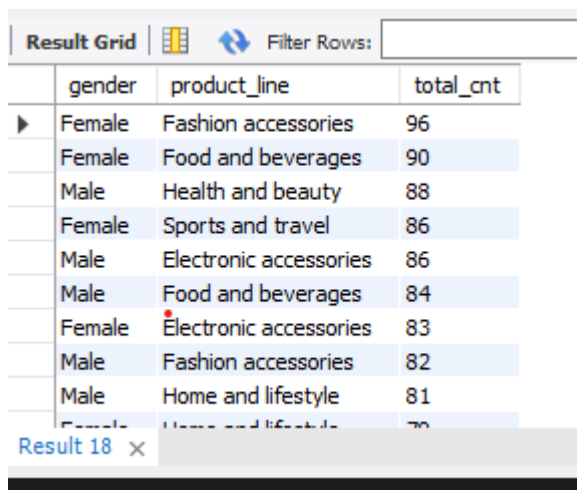
COUNT(gender) AS total\_cnt

FROM sales

GROUP BY gender, product\_line

ORDER BY total\_cnt DESC;

RESULT:



The screenshot shows a SQL query result grid with the following data:

	gender	product_line	total_cnt
▶	Female	Fashion accessories	96
	Female	Food and beverages	90
	Male	Health and beauty	88
	Female	Sports and travel	86
	Male	Electronic accessories	86
	Male	Food and beverages	84
	Female	Electronic accessories	83
	Male	Fashion accessories	82
	Male	Home and lifestyle	81
	Female	Home and lifestyle	70

Result 18 x

-- 11 What is the average rating of each product line

SELECT

ROUND(AVG(rating), 2) as avg\_rating,

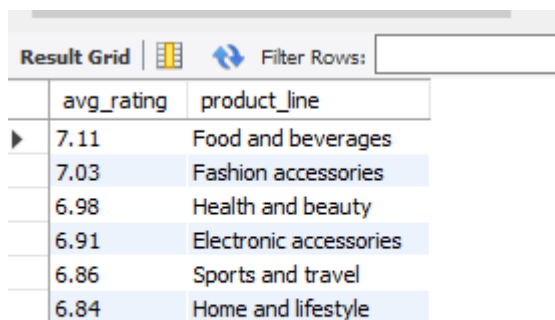
product\_line

FROM sales

GROUP BY product\_line

ORDER BY avg\_rating DESC;

RESULT:



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains two columns: 'avg\_rating' and 'product\_line'. The data is sorted by 'avg\_rating' in descending order. The rows are: Food and beverages (7.11), Fashion accessories (7.03), Health and beauty (6.98), Electronic accessories (6.91), Sports and travel (6.86), and Home and lifestyle (6.84).

avg_rating	product_line
7.11	Food and beverages
7.03	Fashion accessories
6.98	Health and beauty
6.91	Electronic accessories
6.86	Sports and travel
6.84	Home and lifestyle

-----

--CUSTOMER QUESTIONS

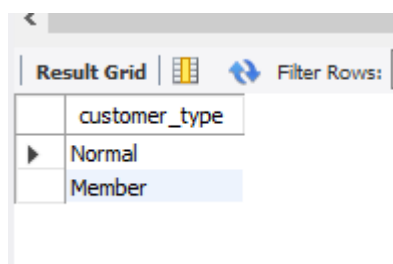
-- 1 How many unique customer types does the data have?

SELECT

DISTINCT customer\_type

FROM sales;

RESULT:



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains one column: 'customer\_type'. The data is sorted alphabetically. The rows are: Normal and Member.

customer_type
Normal
Member

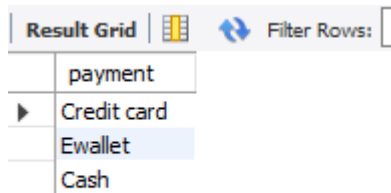
-- 2 How many unique payment methods does the data have?

SELECT

    DISTINCT payment

FROM sales;

RESULT:



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid contains a single column named 'payment' with three rows: 'Credit card', 'Ewallet', and 'Cash'. The 'Ewallet' row is highlighted in blue.

payment
Credit card
Ewallet
Cash

-- 3 What is the most common customer type?

SELECT

    customer\_type,

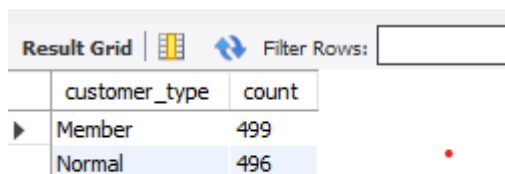
    count(\*) as count

FROM sales

GROUP BY customer\_type

ORDER BY count DESC;

RESULT:



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid has two columns: 'customer\_type' and 'count'. There are two rows: 'Member' with a count of 499, and 'Normal' with a count of 496. The 'Normal' row is highlighted in blue.

customer_type	count
Member	499
Normal	496

-- 5 Which customer type buys the most?

SELECT

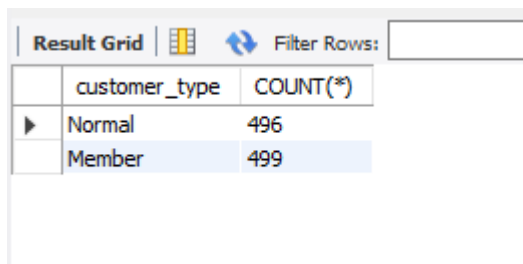
customer\_type,

COUNT(\*)

FROM sales

GROUP BY customer\_type;

RESULT:



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains two columns: 'customer\_type' and 'COUNT(\*)'. There are two rows: 'Normal' with a count of 496, and 'Member' with a count of 499. The 'Member' row is highlighted in blue.

	customer_type	COUNT(*)
▶	Normal	496
	Member	499

-- 6 What is the gender of most of the customers?

SELECT

gender,

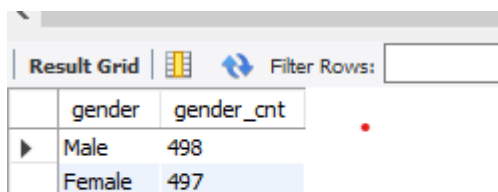
COUNT(\*) as gender\_cnt

FROM sales

GROUP BY gender

ORDER BY gender\_cnt DESC;

RESULT:



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains two columns: 'gender' and 'gender\_cnt'. There are two rows: 'Male' with a count of 498, and 'Female' with a count of 497. The 'Female' row is highlighted in blue.

	gender	gender_cnt
▶	Male	498
	Female	497

-- 7 What is the gender distribution per branch?

SELECT

branch,gender,

COUNT(\*) as gender\_cnt

FROM sales

GROUP BY branch,gender

ORDER BY branch,gender\_cnt DESC;

RESULT:

Result Grid			
	branch	gender	gender_cnt
▶	A	Male	179
	A	Female	160
	B	Male	169
	B	Female	160
	C	Female	177
	C	Male	150

Gender per branch is more or less the same hence, I don't think has an effect of the sales per branch and other factors

-- 8 Which time of the day do customers give most ratings?

SELECT

time\_of\_day,

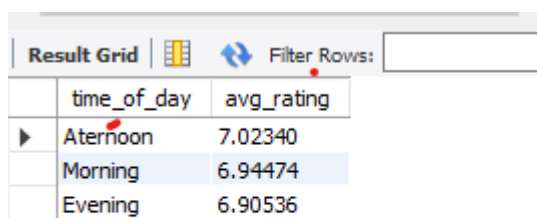
AVG(rating) AS avg\_rating

FROM sales

GROUP BY time\_of\_day

ORDER BY avg\_rating DESC;

RESULT:



The screenshot shows a database interface with a 'Result Grid' tab. It contains a table with two columns: 'time\_of\_day' and 'avg\_rating'. The data is sorted in descending order of average rating. The 'Morning' row is highlighted in blue. There is a small red arrow pointing to the 'Aternoon' row and a red dot above the 'Evening' row.

	time_of_day	avg_rating
▶	Aternoon	7.02340
	Morning	6.94474
	Evening	6.90536

-- Looks like time of the day does not really affect the rating, its

-- more or less the same rating each time of the day.

-- 9 Which time of the day do customers give most ratings per branch?

```
SELECT branch,  
       time_of_day,  
       AVG(rating) AS avg_rating  
FROM sales  
GROUP BY branch,time_of_day  
ORDER BY avg_rating DESC;
```

RESULT:

Result Grid			
Filter Rows:			
	branch	time_of_day	avg_rating
▶	A	Aternoon	7.18889
	C	Evening	7.09859
	C	Aternoon	7.06667
	A	Morning	7.00548
	C	Morning	6.97458
	A	Evening	6.87143
	B	Morning	6.83793
	B	Aternoon	6.81129
	B	Evening	6.75102

-- Branch A and C are doing well in ratings, branch B needs to do a

-- little more to get better ratings.

-- 10 Which day fo the week has the best avg ratings?how many sales are made on these days?

SELECT

day\_name,

AVG(rating) AS avg\_rating,



SUM(total) as total\_sales

FROM sales

GROUP BY day\_name

ORDER BY avg\_rating DESC;

RESULT:

Result Grid		 Filter Rows:	<input type="text"/>
	day_name	avg_rating	total_sales
▶	Monday	7.13065	37344.85
	Friday	7.05507	43848.50
	Tuesday	7.00316	51482.39
	Sunday	6.98864	43937.61
	Saturday	6.90183	56120.86
	Thursday	6.88986	45349.34
	Wednesday	6.76028	42803.52

-- Mon, Tue and Friday are the top best days for good ratings



-- 11 Which day of the week has the best average ratings to branch C?

SELECT

day\_name,

COUNT(day\_name) total\_sales

FROM sales

WHERE branch = "C"

GROUP BY day\_name

ORDER BY total\_sales DESC;

RESULT:

Result Grid		Filter Rows:	
	day_name	total_sales	
▶	Tuesday	54	
	Saturday	54	
	Wednesday	50	
	Thursday	48	
	Sunday	46	
	Monday	38	
	Friday	37	

---

## -- SALES QUESTION

-- 1 Number of sales made in each time of the day per weekday.

SELECT

day\_name as week\_day,time\_of\_day,

COUNT(\*) AS total\_sales

FROM sales

WHERE day\_name = "Sunday"

GROUP BY time\_of\_day

ORDER BY total\_sales DESC;

RESULT:

	week_day	time_of_day	total_sales
►	Sunday	Evening	58
	Sunday	Aternoon	52
	Sunday	Morning	22

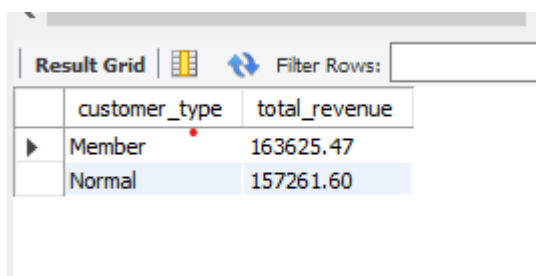
-- Evenings experience most sales, the stores are

-- filled during the evening hours

-- 2 Which of the customer types brings the most revenue?

```
SELECT
    customer_type,
    SUM(total) AS total_revenue
FROM sales
GROUP BY customer_type
ORDER BY total_revenue DESC;
```

RESULT:



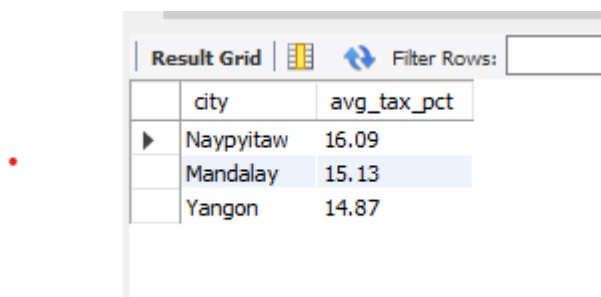
The screenshot shows a 'Result Grid' window with a table containing two columns: 'customer\_type' and 'total\_revenue'. The table has two rows: 'Member' with a total revenue of 163625.47, and 'Normal' with a total revenue of 157261.60. The 'Normal' row is highlighted in blue.

customer_type	total_revenue
Member	163625.47
Normal	157261.60

-- 3 Which city has the largest tax/VAT percent?

```
SELECT
    city,
    ROUND(AVG(tax_pct), 2) AS avg_tax_pct
FROM sales
GROUP BY city
ORDER BY avg_tax_pct DESC;
```

RESULT:



The screenshot shows a 'Result Grid' window with a table containing two columns: 'city' and 'avg\_tax\_pct'. The table has three rows: 'Naypyitaw' with an average tax/VAT percent of 16.09, 'Mandalay' with 15.13, and 'Yangon' with 14.87. The 'Mandalay' row is highlighted in blue.

city	avg_tax_pct
Naypyitaw	16.09
Mandalay	15.13
Yangon	14.87

-- 4 Which customer type pays the most in VAT?

SELECT

customer\_type,

AVG(tax\_pct) AS total\_tax

FROM sales

GROUP BY customer\_type

ORDER BY total\_tax;

RESULT:

Result Grid			Filter Rows:
	customer_type	total_tax	
▶	Normal	15.1	
	Member	15.61	