SELECT table_schema, table_name, table_rows FROM information_schema.tables
WHERE TABLE_SCHEMA LIKE 'classic%';
-- 1. List in alphabetical order the names of the cities in Australia where ClassicModels has customers. (5)
SELECT city AS 'Cities in Austrailia' FROM classicmodels.customers WHERE country = 'Australia'
ORDER BY city asc;

Cities in Austrailia
Chatswood
Glen Waverly
Melbourne
North Sydney
South Brisbane

-- ===
-- 2. List the EmployeeNumber, LastName, FirstName, Extension for all employees working out of the Sydney office. (4)
-- sydney office is code: 6, if you want to check the office code just put E.officeCode
SELECT employeeNumber, LastName, FirstName, Extension
FROM employees E
INNER JOIN offices O
ON E.officeCode = O.officeCode
WHERE E.officeCode = 6;

employeeNumb...	LastName	FirstName	Extension
1088	Patterson	William	x4871
1611	Fixter	Andy	x101
1612	Marsh	Peter	x102
1619	King	Tom	x103

-- ==
-- 3. List the ProductCode, ProductName, ProductVendor, QuantityInStock and ProductLine for all products
-- with a QuantityInStock greater than 4000 and less than 5000. (8)
SELECT productCode, productName, productVendor, quantityinstock, productLine FROM products
WHERE (quantityinstock > 4000 AND quantityinstock < 5000);

productCode	productName	productVendor	quantityinsto...	productLine
S18_2238	1998 Chrysler Plymouth Prowler	Gearbox Collectibles	4724	Classic Cars
S18_2625	1936 Harley Davidson El Knucklehead	Welly Diecast Productions	4357	Motorcycles
S18_2949	1913 Ford Model T Speedster	Carousel DieCast Legends	4189	Vintage Cars
S18_3029	1999 Yamaha Speed Boat	Min Lin Diecast	4259	Ships
S24_1444	1970 Dodge Coronet	Highway 66 Mini Classics	4074	Classic Cars
S24_3191	1969 Chevrolet Camaro Z28	Exoto Designs	4695	Classic Cars
S24_4258	1936 Chrysler Airflow	Second Gear Diecast	4710	Vintage Cars
S72_1253	Boeing X-32A JSF	Motor City Art Classics	4857	Planes

-- ==
-- 4. (Use a SUBQUERY) List the ProductCode, ProductName, ProductVendor, BuyPrice and MSRP for the most
-- expensive (highest MSRP) product sold by ClassicModels. ("MSRP" is the Manufacturer's Suggested Retail Price.) (1)
SELECT productCode, productName, productVendor, buyPrice, MSRP
FROM products
WHERE MSRP = (
SELECT MAX(MSRP) FROM products);

productCode	productName	productVendor	buyPrice	MSRP
S10_1949	1952 Alpine Renault 1300	Classic Metal Creations	98.58	214.30
NULL	NULL	NULL	NULL	NULL

-- ==
-- 5. List the ProductName, MSRP, BuyPrice, and Margin of the product that has the lowest Margin (Margin = MSRP minus BuyPrice). (1)
SELECT productName,MSRP, BuyPrice, MSRP - buyPrice AS 'Margin'
FROM products
WHERE MSRP - BuyPrice =
(SELECT Min(msrp - buyprice) FROM products);

productName	MSRP	BuyPrice	Margin
1939 Chevrolet Deluxe Coupe	33.19	22.57	10.62

-- ==
-- 6. List the country and the number of customers from that country for all countries with more than five customers.
-- List the countries in order from most to least customers. Title the column heading for the count of customers as "Customers". (4)
-- COUNT(e1) , this method will group all the contents of e1 and print it as a count of how many items were grouped
-- need to group all customers with the same country and then you need to filter those out to where only the
-- countries who have 4 or more customers
SELECT country, COUNT(customerName) AS 'Customers' FROM customers GROUP BY country HAVING COUNT(customerName) > 5;

country	Customers
France	12
USA	36
Germany	13
Spain	7

-- ==
-- 7. List the ProductCode, ProductName, and number of orders for the product with the most orders.
-- Title the column heading for the count of orders as "OrderCount". (1)
SELECT productName, orderDetails.productCode, SUM(quantityOrdered) AS "OrderCount" FROM products JOIN orderDetails
WHERE orderDetails.productCode = products.productCode
GROUP BY productCode ORDER BY 3 DESC LIMIT 1;

productName	productCode	OrderCount
1992 Ferrari 360 Spider red	S18_3232	1808

-- ==
-- 8. Create an "Employee Roster" report listing the Manager's name and the name of the employees who report to that manager.
-- Sort by Manager. For both managers and employees, concatenate Firstname + Lastname and refer to them as "ManagerName" and "EmployeeName"
-- in the answer set. (22)
-- assuming this means that anyone who reports to someone else is a there 'manager'
```sql
SELECT CONCAT(m.firstName,' ',m.lastName) AS 'ManagerName' , CONCAT(e.firstName,' ',e.lastName) AS 'EmployeeName'
 FROM employees m
 INNER JOIN employees e ON m.employeeNumber = e.reportsTo
 ORDER BY 1;
```

ManagerName	EmployeeName
Anthony Bow	Leslie Jennings
Anthony Bow	Leslie Thompson
Anthony Bow	Julie Firrelli
Anthony Bow	Steve Patterson
Anthony Bow	Foon Yue Tseng
Anthony Bow	George Vanauf
Diane Murphy	Mary Patterson
Diane Murphy	Jeff Firrelli
Gerard Bondur	Loui Bondur
Gerard Bondur	Gerard Hernandez
Gerard Bondur	Pamela Castillo
Gerard Bondur	Larry Bott
Gerard Bondur	Barry Jones
Gerard Bondur	Martin Gerard
Mami Nishi	Yoshimi Kato
Mary Patterson	William Patterson
Mary Patterson	Gerard Bondur
Mary Patterson	Anthony Bow
Mary Patterson	Mami Nishi
William Patte…	Andy Fixter
William Patte…	Peter Marsh
William Patte…	Tom King

-- ===
-- 9. List the EmployeeNumber, LastName, FirstName of the president of the company (the one employee with no boss.) (1)
SELECT EmployeeNumber, LastName, FirstName FROM employees WHERE jobTitle = 'President';

EmployeeNumber	LastName	FirstName
1002	Murphy	Diane

-- ===
-- 10. List the ProductName for all products in the "Classic Cars" product line from the 1960's. (11)
SELECT productName from products P JOIN productlines PL
ON P.productLine = PL.productLine WHERE P.productLine = 'Classic Cars'
AND P.productName LIKE '196_%';

productName
1962 LanciaA Delta 16V
1968 Ford Mustang
1969 Corvair Monza
1968 Dodge Charger
1969 Ford Falcon
1969 Dodge Charger
1965 Aston Martin DB5
1969 Dodge Super Bee
1966 Shelby Cobra 427 S/C
1969 Chevrolet Camaro Z28
1961 Chevrolet Impala

-- ===

-- 11. List the month name and the year and the order count for the TOP TWO months in which ClassicModels customers

-- placed the highest number of orders. (2)

-- get this one checked probably

```sql
SELECT MONTHNAME(orderDate) AS 'MONTH' , EXTRACT(YEAR FROM orderDate) AS 'YEAR',
COUNT(quantityOrdered) AS 'ORDERS' FROM orders O, orderDetails OD
GROUP BY MONTHNAME(orderDate), EXTRACT(YEAR FROM orderDate) ORDER BY 3 DESC
LIMIT 2;
```

MONTH	YEAR	ORDERS
November	2004	98868
November	2003	89880

-- ===

-- 12. List the firstname, lastname of employees who are Sales Reps who have no assigned customers. (2)

```sql
SELECT firstname, lastname FROM Employees E JOIN Customers C
WHERE E.jobTitle = 'sales rep' AND
NOT(E.employeeNumber IN (select salesRepEmployeeNumber from customers where
salesrepemployeenumber != 'Null'))
GROUP BY firstname, lastname, employeeNumber;
```

firstname	lastname
Tom	King
Yoshimi	Kato

-- ===

-- 13. List the customername of customers from Switzerland with no orders. (2)

-- this was based off of the assumption that you knew no customer had no orders if they had no payments.

-- so they could have a sales rep but that doesn't mean they went through with any transactions.

```sql
SELECT * FROM customers WHERE NOT(( customerNumber IN (SELECT customerNumber FROM
Payments) ))
AND country = 'switzerland';
```

customerName
BG&E Collectables
Precious Collectables

-- ===
-- 14. List the customername and total quantity of products ordered for customers who have ordered less than 500 products
-- across all their orders. (11)
SELECT customerName AS 'Customer Name' , SUM(quantityOrdered) AS 'SumOfProducts' FROM customers C JOIN orders O JOIN orderDetails OD
ON C.customerNumber = O.customerNumber AND O.ordernumber = OD.ordernumber
GROUP BY customerName HAVING SUM(quantityOrdered) < 500;

Customer Name	SumOfProducts
Atelier graphique	270
Cambridge Collectables Co.	357
Clover Collections, Co.	490
Auto-Moto Classics Inc.	287
Boards & Toys Co.	102
CAF Imports	468
Royale Belge	278
Bavarian Collectables Imports, Co.	401
Microscale Inc.	381
Frau da Collezione	272
Double Decker Gift Stores, Ltd	357

-- ===
-- 15. Create a NEW table named "LowCustomers" with three columns: CustomerNumber (integer), ContactDate (DATE) and OrderTotal
-- (a decimal number with 9 digits in total having two decimal places). None of these columns can be NULL. Include a PRIMARY KEY
-- constraint named "LowCustomer_PK" on CustomerNumber. (no answer set)
CREATE TABLE LowCustomers(
        CustomerNumber INT,
    OrderTotal DECIMAL(9,2),
    ContactDate DATE,
    CONSTRAINT PK_LowCustomers PRIMARY KEY(CustomerNumber)
);
-- ===
-- 16. Populate the new table "LowCustomers" with the CustomerNumber, today's date, and the total value of all their
-- orders (PriceEach * quantityOrdered) for those customers whose order total value is less than $50,000. (inserted 16 rows, no answer set)
INSERT INTO LowCustomers
        (SELECT customerNumber, SUM(PriceEach * quantityOrdered), CURDATE() FROM orders JOIN orderDetails
        ON orders.orderNumber = orderDetails.orderNumber GROUP BY customerNumber HAVING SUM(PriceEach*quantityOrdered) < 50000);

-- ==
-- 17. List the contents of the LowCustomers table in descending OrderTotal sequence. (16)
SELECT * FROM lowcustomers ORDER BY OrderTotal DESC;

CustomerNumb...	OrderTotal	ContactDate
447	49967.78	2020-02-19
189	49898.27	2020-02-19
344	46751.14	2020-02-19
211	45480.79	2020-02-19
475	43748.72	2020-02-19
487	42570.37	2020-02-19
347	41506.19	2020-02-19
173	32198.69	2020-02-19
415	31310.09	2020-02-19
489	29586.15	2020-02-19
456	29230.43	2020-02-19
381	29217.18	2020-02-19
473	25358.32	2020-02-19
103	22314.36	2020-02-19
198	21554.26	2020-02-19
219	7918.60	2020-02-19

-- ==
-- 18. Add a new column to the LowCustomers table called OrderCount (integer). (No answer set)
ALTER TABLE lowcustomers
        ADD COLUMN OrderCount INT;

-- ==
-- 19. Update the LowCustomers table, setting the OrderCount column to a random number (from 0 to
18). (Should update 16 rows)
-- HINT: use the RAND() and FLOOR() functions.
UPDATE lowCustomers
SET ordercount = (SELECT FLOOR(RAND()*(18-0+1)+0));

SET SQL_SAFE_UPDATES = 0; -- needed to do this to bypass warning, probably a bad idea

-- ==
-- 20. List the contents of the LowCustomers table in descending OrderCount sequence. (16)
SELECT * FROM lowcustomers ORDER BY orderCount DESC;

CustomerNumb...	OrderTotal	ContactDate	OrderCount
189	49898.27	2020-02-19	18
211	45480.79	2020-02-19	16
489	29586.15	2020-02-19	16
473	25358.32	2020-02-19	15
173	32198.69	2020-02-19	14
103	22314.36	2020-02-19	12
219	7918.60	2020-02-19	11
198	21554.26	2020-02-19	10
447	49967.78	2020-02-19	8
344	46751.14	2020-02-19	6
415	31310.09	2020-02-19	6
475	43748.72	2020-02-19	5
456	29230.43	2020-02-19	4
487	42570.37	2020-02-19	3
381	29217.18	2020-02-19	1
347	41506.19	2020-02-19	0

-- ==
-- 21. Drop the low customers table (no answer set)
DROP TABLE IF EXISTS LowCustomers;

COLLABORATORS:
Zachary Chommala
Bao Nguyen