```
+--------------------------+
| CSCI3287 Database Systems |
+--------------------------+
| Project 1 - SQL           |
+--------------------------+
| David Skrenta             |
+--------------------------+
| Bryce Desbrisay           |
+--------------------------+
```

1. List all the information in the Offices table. (7)

```
mysql> select * from offices;
+------------+---------------+------------------
+--------------------------+-------------+-----------+----------
+------------+-----------+
| officeCode | city          | phone             | addressLine1
| addressLine2 | state      | country   | postalCode | territory |
+------------+---------------+------------------
+--------------------------+-------------+-----------+----------
+------------+-----------+
| 1          | San Francisco | +1 650 219 4782  | 100 Market Street
| Suite 300    | CA         | USA       | 94080      | NA        |
| 2          | Boston        | +1 215 837 0825  | 1550 Court Place
| Suite 102    | MA         | USA       | 02107      | NA        |
| 3          | NYC           | +1 212 555 3000  | 523 East 53rd Street
| apt. 5A      | NY         | USA       | 10022      | NA        |
| 4          | Paris         | +33 14 723 4404  | 43 Rue Jouffroy
D'abbans | NULL         | NULL       | France    | 75017      | EMEA
|
| 5          | Tokyo         | +81 33 224 5000  | 4-1 Kioicho
| NULL         | Chiyoda-Ku | Japan     | 102-8578   | Japan     |
| 6          | Sydney        | +61 2 9264 2451  | 5-11 Wentworth
Avenue    | Floor #2     | NULL       | Australia | NSW 2010   | APAC
|
| 7          | London        | +44 20 7877 2041 | 25 Old Broad Street
| Level 7      | NULL       | UK        | EC2N 1HN   | EMEA      |
+------------+---------------+------------------
+--------------------------+-------------+-----------+----------
+------------+-----------+
7 rows in set (0.00 sec)
```

2. List the EmployeeNumber, LastName, FirstName, Extension for all employees working out of the
Paris, France office. (5)

```
mysql> select employeeNumber, lastName, firstName, extension from
employees where officeCode = 4;
+---------------+-----------+-----------+-----------+
| employeeNumber | lastName  | firstName | extension |
```

```
+----------------+-----------+-----------+-----------+
|           1102 | Bondur    | Gerard    | x5408     |
|           1337 | Bondur    | Loui      | x6493     |
|           1370 | Hernandez | Gerard    | x2028     |
|           1401 | Castillo  | Pamela    | x2759     |
|           1702 | Gerard    | Martin    | x2312     |
+----------------+-----------+-----------+-----------+
5 rows in set (0.00 sec)
```

3. List the ProductCode, ProductName, ProductVendor, QuantityInStock for all products in the “Classic Cars” product line with a QuantityInStock between 5000 and 7000. (7)

```
mysql> select ProductCode, ProductName, ProductVendor, QuantityInStock
from products where productLine = 'Classic Cars' and (QuantityInStock
> 3000 and QuantityInStock < 5000;
+-------------+-------------------------------
+-------------------------+-----------------+
| ProductCode | ProductName                   | ProductVendor
| QuantityInStock |
+-------------+-------------------------------
+-------------------------+-----------------+
| S10_4757    | 1972 Alfa Romeo GTA           | Motor City Art
Classics  |            3252 |
| S12_1108    | 2001 Ferrari Enzo             | Second Gear Diecast
|            3619 |
| S18_1129    | 1993 Mazda RX-7               | Highway 66 Mini
Classics |            3975 |
| S18_2238    | 1998 Chrysler Plymouth Prowler | Gearbox Collectibles
|            4724 |
| S18_4933    | 1957 Ford Thunderbird         | Studio M Art Models
|            3209 |
| S24_1444    | 1970 Dodge Coronet            | Highway 66 Mini
Classics |            4074 |
| S24_3191    | 1969 Chevrolet Camaro Z28     | Exoto Designs
|            4695 |
+-------------+-------------------------------
+-------------------------+-----------------+
7 rows in set (0.00 sec)
```

4. List the ProductCode, ProductName, ProductVendor, BuyPrice and MSRP for the least expensive
(lowest MSRP) product sold by ClassicModels. (“MSRP” is the Manufacturer’s Suggested Retail Price.)
(1)

```
NOTE: No ClassicModels vendor found in data
mysql> select x.ProductCode, x.ProductName, x.ProductVendor,
x.BuyPrice, min(x.MSRP) as MSRP from (select * from products where
productVendor = 'Classic Metal Creations') as x;
```

```
+-------------+-------------------------+-----------------------
+----------+-------+
| productCode | productName               | productVendor           |
buyPrice | MSRP  |
+-------------+-------------------------+-----------------------
+----------+-------+
| S10_1949    | 1952 Alpine Renault 1300 | Classic Metal Creations |
98.58 | 44.80 |
+-------------+-------------------------+-----------------------
+----------+-------+
1 row in set (0.00 sec)
```

5. What is the ProductName and Profit of the product that has the highest profit (profit = MSRP minus
BuyPrice). (1)

```
mysql> select productName, max(MSRP - buyPrice) as profit from
products;
+--------------------------------------+--------+
| productName                          | profit |
+--------------------------------------+--------+
| 1969 Harley Davidson Ultimate Chopper | 115.72 |
+--------------------------------------+--------+
1 row in set (0.00 sec)
```

6. List the country and the number of customers from that country for all countries having five or more
customers. List the countries sorted in descending order from highest to lowest number of customers.
(6)

```
mysql> select country from (select country, count(*) as customerCount
from customers group by country) as x where customerCount >= 5 order
by customerCount desc;
+-----------+
| country   |
+-----------+
| USA       |
| Germany   |
| France    |
| Spain     |
| Australia |
| UK        |
+-----------+
6 rows in set (0.00 sec)
```

7. List the ProductCode, ProductName, and number of orders for the product with the most orders. (1)

```
mysql> select x.productCode, max(x.productCount) as totalOrders,
```

```
products.productName from (select productCode, count(*) as
productCount from orderdetails group by productCode) as x join
products on products.productCode = x.productCode;
+-------------+-------------+-----------------------------------+
| productCode | totalOrders | productName                       |
+-------------+-------------+-----------------------------------+
| S10_1678    |          53 | 1969 Harley Davidson Ultimate Chopper |
+-------------+-------------+-----------------------------------+
1 row in set (0.00 sec)
```

8. List the EmployeeNumber, Firstname + Lastname (concatenated into one column in the answer set,
separated by a blank) for all the employees reporting to Anthony Bow. (6)

```
mysql> select employeeNumber, concat(firstName, ' ', lastName) as
firstLast from employees where reportsTo = 1143;
+----------------+-----------------+
| employeeNumber | firstLast       |
+----------------+-----------------+
|           1165 | Leslie Jennings |
|           1166 | Leslie Thompson |
|           1188 | Julie Firrelli  |
|           1216 | Steve Patterson |
|           1286 | Foon Yue Tseng  |
|           1323 | George Vanauf   |
+----------------+-----------------+
6 rows in set (0.00 sec)
```

9. List the EmployeeNumber, LastName, FirstName of the president of the company (the one employee
with no boss.) (1)

```
mysql> select employeeNumber, firstName, lastName from employees where
jobTitle = 'President';
+----------------+-----------+----------+
| employeeNumber | firstName | lastName |
+----------------+-----------+----------+
|           1002 | Diane     | Murphy   |
+----------------+-----------+----------+
1 row in set (0.00 sec)
```

10. List the ProductName for all products in the “Classic Cars” product line from the 1950’s. (6)

```
mysql> select productName from (select productName,
substring(productName, 1, 4) as year from products where productLine =
'Classic Cars') as x where year >= '1950' and year < '1960';
+----------------------------------+
| productName                      |
```

```
+-----------------------------------+
| 1952 Alpine Renault 1300          |
| 1957 Corvette Convertible         |
| 1957 Ford Thunderbird             |
| 1958 Chevy Corvette Limited Edition |
| 1952 Citroen-15CV                 |
| 1956 Porsche 356A Coupe           |
+-----------------------------------+
6 rows in set (0.00 sec)
```

11. List the month name and the total number of orders for the month in 2003 in which ClassicModels
customers placed the most orders. (1)

```
mysql> select max(totalMonthOrders) as totalMonthOrders, orderMonth
from (select *, count(*) as totalMonthOrders from (select *,
substring(orderDate, 1, 4) as orderYear, substring(orderDate, 6, 2) as
orderMonth from orders) as x where orderYear = '2003' group by
orderMonth) as y;
+-----------------+-----------+
| totalMonthOrders | orderMonth |
+-----------------+-----------+
|               30 | 01        |
+-----------------+-----------+
1 row in set (0.01 sec)
```

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

```
mysql> select firstName, lastName from employees where jobTitle =
'Sales Rep' and not exists (select * from customers where
salesRepEmployeeNumber = employeeNumber);
+-----------+----------+
| firstName | lastName |
+-----------+----------+
| Tom       | King     |
| Yoshimi   | Kato     |
+-----------+----------+
2 rows in set (0.00 sec)
```

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

```
mysql> select customers.customerName from customers where country =
'Spain' and not exists (select * from orders where customerNumber =
customers.customerNumber);
+--------------------+
| customerName       |
+--------------------+
| ANG Resellers      |
| Anton Designs, Ltd. |
```

```
+--------------------+
2 rows in set (0.00 sec)
```

14. List the customername and total quantity of products ordered for customers who have ordered
more than 2000 products across all their orders. (2)

```
mysql> select customerName, totalOrders from (select customerNumber,
sum(quantityOrdered) as totalOrders from orderdetails join orders on
orders.orderNumber = orderdetails.orderNumber group by customerNumber)
as x join customers on customers.customerNumber  = x.customerNumber
where totalOrders > 2000;
+-----------------------------+-------------+
| customerName                | totalOrders |
+-----------------------------+-------------+
| Mini Gifts Distributors Ltd. |        6366 |
| Euro+ Shopping Channel       |        9327 |
+-----------------------------+-------------+
2 rows in set (0.00 sec)
```

15. Create a NEW table named “TopCustomers” 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 “TopCustomer_PK” on
CustomerNumber. (no answer set)

```
mysql> create table TopCustomers (
    customerNumber int,
    contractDate date,
    orderTotal decimal(9),
    constraint TopCustomer_PK primary key (customerNumber)
);
```

16. Populate the new table “TopCustomers” with the CustomerNumber, today’s date, and the total
value of all their orders (PriceEach * quantityOrdered) for those customers whose order total value is
greater than $150,000. (inserted 7 rows, no answer set)

```
mysql> insert into TopCustomers (customerNumber, contractDate,
orderTotal)
select customerNumber, now() as currentDate, totalValue from (select
x.customerNumber, sum(x.totalValue) as totalValue from (select
customerNumber, (orderdetails.priceEach *
orderdetails.quantityOrdered) as totalValue from orderdetails join
orders on orders.orderNumber = orderdetails.orderNumber) as x group by
x.customerNumber) as y where y.totalValue > 150000;
```

17. List the contents of the TopCustomers table in descending OrderTotal sequence. (7)

```
mysql> select * from TopCustomers order by orderTotal desc;
+----------------+--------------+------------+
| customerNumber | contractDate | orderTotal |
+----------------+--------------+------------+
|            141 | 2018-10-28   |     820690 |
|            124 | 2018-10-28   |     591827 |
|            114 | 2018-10-28   |     180585 |
|            151 | 2018-10-28   |     177914 |
|            119 | 2018-10-28   |     158573 |
|            148 | 2018-10-28   |     156251 |
|            323 | 2018-10-28   |     154622 |
+----------------+--------------+------------+
7 rows in set (0.00 sec)
```

18. Add a new column to the TopCustomers table called OrderCount (integer). (No answer set)

```
mysql> alter table TopCustomers add OrderCount int;
```

19. Update the Top Customers table, setting the OrderCount column to a random number (from 0 to 20).
(Should update 7 rows) HINT: use the RAND() and FLOOR() functions.

```
mysql> update TopCustomers set OrderCount = floor(rand() *
(20-0+1)+0);
```

20. List the contents of the TopCustomers table in descending OrderCount sequence. (7)

```
mysql> select * from TopCustomers order by OrderCount desc;
+----------------+--------------+------------+------------+
| customerNumber | contractDate | orderTotal | OrderCount |
+----------------+--------------+------------+------------+
|            151 | 2018-10-28   |     177914 |         18 |
|            119 | 2018-10-28   |     158573 |         16 |
|            141 | 2018-10-28   |     820690 |         14 |
|            114 | 2018-10-28   |     180585 |         13 |
|            148 | 2018-10-28   |     156251 |          8 |
|            323 | 2018-10-28   |     154622 |          5 |
|            124 | 2018-10-28   |     591827 |          0 |
+----------------+--------------+------------+------------+
7 rows in set (0.01 sec)
```

21. Drop the TopCustomers table. (no answer set)

```
mysql> drop table TopCustomers;
```