MIS 431 Spring 2023
Sample Answers (there could be other ways to write a query)

1. What are the product names, product lines, and product description of all products whose scale is 1:18?

```
SELECT productname, productline, productdescription
FROM CM_products
WHERE productscale = '1:18';
```

2. What products can be purchased for prices between $95 and $105?

```
SELECT productName
FROM CM_products
WHERE BuyPrice between 95 and 105;
```

3. List the product code, product name, quantity in stock, MSRP, and buy price for the lowest priced motorcycle.

```
SELECT productcode, quantityinstock, MSRP, buyprice
FROM CM_products
WHERE Buyprice= (Select min(buyprice) FROM CM_products where productline = 'Motorcycles')
AND productline = 'Motorcycles';
```

4. List the product name and buy price for products that are Motorcycles and 1:10 Product Scale that are greater than the average buy price for products that are Motorcycles and 1:10 product scale. Please format your average to two decimal places.

```
SELECT productName, round(buyprice,2)
FROM CM_products
WHERE ProductLine = 'Motorcycles'
AND ProductScale = '1:10'
And buyprice > (select avg(buyprice) from CM_products
WHERE ProductLine = 'Motorcycles'
AND ProductScale = '1:10');
```

5. Which employees have a last name starting with the letter T? Format so the output is only one column with a comma and space between the last name and the first name.

```
SELECT CONCAT(FirstName, ' ', LastName) AS Employee
FROM CM_employees
WHERE LastName LIKE 'T%';
```

6. List the product name and buy price of those products with the same buy price; only include buy prices under $70.

```
SELECT a. ProductName, b.ProductName, a.BuyPrice
FROM CM_products a join CM_products b on b.BuyPrice=a.BuyPrice
WHERE a.ProductCode > b.ProductCode AND a.BuyPrice < 70;
```

7. List the customer contact name, productline, orderdate, and product name for the sales that occurred in January. Customer name should be formatted to one column, with a space between first name and last name and then labeled as customer in the heading.

```
SELECT CONCAT(ContactFirstName,' ', ContactLastName) as 'Customer', Productline, OrderDate,
productName
FROM CM_customers c JOIN CM_orders o ON c.Customernumber = o.Customernumber
JOIN CM_orderdetails od ON od.OrderNumber = o.OrderNumber
JOIN CM_products p ON od.Productcode = p.Productcode
WHERE MONTH(OrderDate) = 1;
```

8. List the customers with the Sales Reps who have the employee numbers 1370 or 1165. Please use the In keyword and not a subquery.

```
SELECT ContactFirstName, ContactLastName
FROM CM_customers
WHERE SalesRepEmployeeNumber IN ('1370', '1165');
```

9. List the customer name, the contact first and last name and if they have a sales rep, the sales rep first and last name. Concatinate the first and last name of the contact, and also the first and last name of the sales rep and name the headers appropriately.

```
Select customername, concat(contactfirstname,' ' ,contactlastname) as CustomerContact,
concat(firstname,' ' ,lastname) as SalesRep from CM_customers left join CM_Employees on
salesrepemployeenumber =employeeNumber;
```

10. List the Employees last and first name separated by a comma, and the country they are working in and list the Customer contact last and first name separated by a comma and the country they are in. Name the headers Employees_And_Customers, and Location. Your result should be in two columns and sort the output by country.

```
(select concat(lastname,', ',firstname) as Employees_And_Customers, country as Location
from CM_employees join CM_offices on CM_employees.officecode=CM_offices.officecode)
union
(select concat(contactlastname,', ', contactfirstname), country
from CM_customers)
order by Location;
```

11. List the product description and number of products that are Classic Cars.

```
SELECT ProductDescription, count(productline)
FROM CM_products
WHERE ProductLine= 'Classic Cars'
Group by ProductDescription;
```

12. Using the case keyword, for each Product, display the name, product line, product vendor, and if the product is product scale 1:10, display “Small Size”, if 1:12, display “Mid Size”, else display “Other Size”.

```
SELECT Productname, ProductLine, ProductVendor, case productscale when '1:10' then 'Small Size'
when '1:12' then 'Mid Size' else 'Other Size' End as Product_Size FROM CM_products;
```

13. List the Product name, Product Description, and Product Vendor of the cheapest Product using BuyPrice

```
SELECT productname, productdescription, productvendor from CM_products where buyprice = (Select
min(buyPrice) FROM CM_products);
```

14. List every customer contact name that has not purchased any products.

```
SELECT ContactFirstName, ContactLastName
FROM CM_customers
WHERE CustomerNumber not in (select CustomerNumber from CM_orders O JOIN CM_orderdetails OD
on O.OrderNumber = OD.OrderNumber);
```

15. Who made a purchase most recently? Please include name, order date, and days since order in your answer.

```
SELECT CONCAT(ContactFirstName,' ', ContactLastName) as 'Customer', Orderdate,
datediff(current_date, orderdate)
FROM CM_customers c JOIN CM_orders o ON c.Customernumber = o.Customernumber
where datediff(current_date, orderdate) = (select min(datediff(current_date, orderdate)) from
cm_orders)
```

or

```
SELECT CONCAT(ContactFirstName,' ', ContactLastName) as 'Customer', Orderdate,
datediff(current_date, orderdate)
FROM CM_customers c JOIN CM_orders o ON c.Customernumber = o.Customernumber
where orderdate = (select max(orderdate) from cm_orders)
```

16. Display the different products and their quantity in stock. Use the IF function to generate inventory replenishment status messages. Use 4000 as your benchmark to replenish inventory and put the items in alphabetical order. Therefore if it's less than 4000, the message should be Replenish inventory, else it should be Inventory levels are Fine. Also, the column header should be Message.

```
SELECT ProductCode, ProductName, QuantityInStock,
IF (QuantityInStock < 4000, 'Replenish inventory', 'Inventory levels are Fine') AS Message
FROM CM_products
ORDER BY ProductName ;
```

17. List the Customer name, order number, and total cost for the customer who has the most expensive order. Note, a sale may have several items on the order.

```
create view ProductSales (ordernumber, totalsale) as select ordernumber,
sum(quantityordered*priceeach) from CM_orderdetails group by OrderNumber;

select CustomerName, totalsale from CM_customers c Join CM_orders o ON
c.CustomerNumber=o.CustomerNumber join ProductSales ps ON ps.OrderNumber=o.OrderNumber
where totalsale = (select max(totalsale) from ProductSales);
```

Just for your information, and you are not required to know how for this class, but another way to write this query not using a view would be:

```
select  CustomerName, od.ordernumber, sum(quantityordered*priceeach) from  CM_customers c Join
CM_orders o ON c.CustomerNumber=o.CustomerNumber join CM_orderdetails  od  ON
od.OrderNumber=o.OrderNumber
group by CustomerName, OrderNumber
having sum(quantityordered*priceeach) = (select max(totalsale) from (select ordernumber,
sum(quantityordered*priceeach) as totalsale from CM_Orderdetails group by ordernumber) as M)
```

18. List the product name and that has a higher than average MSRP and a lower than average buyPrice.

```
select ProductName from CM_products
where MSRP > (select avg(MSRP) from CM_products) and buyPrice < (select avg(buyPrice) from
CM_products);
```

19. List the product name and MSRP where the MSRP is the minimum amount greater than the average MSRP.

```
select productname, MSRP from CM_products where
MSRP = (select min(MSRP) from CM_products where MSRP > (select avg(MSRP) from CM_products));
```

20. List the Product Number, Name, Scale, Order Date, and number of days since the order date of all orders within the past 20 days. Name any headers appropriately.

```
SELECT p.productcode, productname, productscale, orderdate, datediff(CURRENT_DATE, orderdate) as
TimeSinceOrder
from CM_products p join CM_orderdetails od on p.productcode=od.productcode join CM_orders o on
od.OrderNumber=o.OrderNumber
where datediff(CURRENT_DATE, Orderdate) <= 20
```