Selects

1. List all the data in the classic models database:

a) Product Lines (7) SELECT * FROM productlines;

b) Product (110); SELECT * FROM products;

c) Employees (23) SELECT * FROM employees;

d) Offices (7) SELECT * FROM offices;

e) Customers (122) SELECT * FROM customers;

f) Orders (326) SELECT * FROM orders;

g) Orderdetails (2996) SELECT * FROM orderdetails;

h) Payments (273) SELECT * FROM payments;

2. Select customer name from customer. Sort by customer name (122) - SELECT customername FROM customers ORDER BY customername ;

3. List each of the different status that an order may be in (6) Cancelled, Disputed, In Process, On Hold, Resolved, Shipped

4. List firstname and lastname for each employee. Sort by lastname then firstname (23)
SELECT lastname, firstname FROM employees ORDER BY lastname, firstname;

5. List all the employee job titles (7)
SELECT DISTINCT jobtitle FROM employees ORDER BY jobtitle;

6. List all products along with their product scale (110)
SELECT productname, productscale FROM products;

7. List all the territories where we have offices (4)
SELECT DISTINCT territory FROM offices;

Where Clause

8. select contact firstname, contact lastname and credit limit for all customers where credit limit > 50000 (85)
SELECT contactfirstname, contactlastname, creditlimit FROM customers WHERE creditlimit > 50000;

9. select customers who do not have a credit limit (0.00) (24)
SELECT customername FROM customers WHERE creditlimit = 0.00;

10. List all offices not in the USA (4)
SELECT city, country FROM offices WHERE country != 'USA' ;

11. List orders made between June 16, 2014 and July 7, 2014 (8)
SELECT orderdate FROM orders WHERE orderdate >= '2014-06-16' AND orderdate <= '2014-07-07' ;

12. List products that we need to reorder (quantityinstock < 1000) (12)
SELECT productname FROM products WHERE quantityinstock < 1000 ;

13. List all orders that shipped after the required date (1)
SELECT ordernumber FROM orders WHERE shippeddate > requireddate;

14. List all customers who have the word ‘Mini’ in their name (10)

SELECT customername FROM customers WHERE customername LIKE '%Mini%';

15. List all products supplied by ‘Highway 66 Mini Classics’ (9)

SELECT productname FROM products WHERE productvendor = 'Highway 66 Mini Classics';

16. List all product not supplied by ‘Highway 66 Mini Classics’ (101)

SELECT productname FROM products WHERE productvendor != 'Highway 66 Mini Classics';

17. List all employees that don't have a manager (1)

SELECT firstname, lastname, reportsto FROM employees WHERE reportsto IS NULL ;

Natural Join

18. Display every order along with the details of that order for order numbers 10270, 10272, 10279 (23)

Hint: this can be done two ways. Try both of them. Which is easier if you have a large number of selection criteria?

19. List of productlines, text description of the product line and vendors that supply the products in that productline. (65)

SELECT DISTINCT productline, textdescription, productvendor FROM products NATURAL JOIN productlines ;

Inner Join

20. select customers that live in the same state as one of our offices (26)

SELECT customername, customers.state, offices.state FROM offices INNER JOIN customers on customers."STATE" = offices."STATE" ;

21. select customers that live in the same state as their employee representative works (26)

```
SELECT  customername, customers.state, offices.state
FROM ((customers
INNER JOIN employees on customers.SALESREPEMPLOYEENUMBER =
employees.EMPLOYEENUMBER)
INNER JOIN offices ON customers."STATE" = offices."STATE") ;
```

Multi-join

22. select customerName, orderDate, quantityOrdered, productLine, productName for all orders made and shipped in 2015 (444)

```
SELECT  customername, orderdate, quantityordered, productline, productname, shippeddate
FROM (((orders
INNER JOIN customers on customers.CUSTOMERNUMBER = orders.CUSTOMERNUMBER)
INNER JOIN orderdetails ON orderdetails.ORDERNUMBER = orders.ORDERNUMBER)
INNER JOIN products ON products.PRODUCTCODE = orderdetails.PRODUCTCODE)
WHERE orderdate >= '2015-01-01' AND orderdate <= '2015-12-31'
AND shippeddate >= '2015-01-01' AND shippeddate <= '2015-12-31';
```

Outer Join

23. List products that didn't sell (1)

```
SELECT  productname, products.PRODUCTCODE, quantityordered
FROM products
LEFT OUTER JOIN orderdetails
ON products.PRODUCTCODE = orderdetails.PRODUCTCODE
WHERE quantityordered Is NULL;
```

24. List all customers and their sales rep even if they don’t have a sales rep (122)

```
SELECT  customername, lastname, firstname
FROM customers
LEFT OUTER JOIN employees
ON customers.SALESREPEMPLOYEENUMBER = employees.EMPLOYEENUMBER;
```

Aggregate Functions

25. Find the total of all payments made by each customer (98)

```
SELECT customername,  SUM(priceeach * quantityordered) AS totalsales
FROM ((customers
INNER JOIN orders
ON customers.CUSTOMERNUMBER = orders.CUSTOMERNUMBER)
INNER JOIN orderdetails
ON orderdetails.ORDERNUMBER = orders.ORDERNUMBER)
GROUP BY customername;
```

26. Find the largest payment made by a customer (1)

```
SELECT   MAX(priceeach * quantityordered) AS largestpayment
FROM ((customers
INNER JOIN orders
ON customers.CUSTOMERNUMBER = orders.CUSTOMERNUMBER)
INNER JOIN orderdetails
ON orderdetails.ORDERNUMBER = orders.ORDERNUMBER);
```

27. Find the average payment made by a customer (1)

```
SELECT   AVG(priceeach * quantityordered) AS averagepayment
FROM ((customers
INNER JOIN orders
ON customers.CUSTOMERNUMBER = orders.CUSTOMERNUMBER)
INNER JOIN orderdetails
ON orderdetails.ORDERNUMBER = orders.ORDERNUMBER);
```

28. What is the total number of products per product line (7)

```
SELECT  productlines.PRODUCTLINE, COUNT(productname) AS prodperline
FROM products
INNER JOIN productlines
ON products.PRODUCTLINE = productlines.PRODUCTLINE
GROUP BY productlines.productline;
```

29. What is the number of orders per status (6)

```
SELECT  status, COUNT(ordernumber) AS prodperline
```

```
FROM orders
GROUP BY status;
```

30. List all offices and the number of employees working in each office (7)

```
SELECT  offices.officecode, city, COUNT(employeenumber) AS prodperline
FROM offices
INNER JOIN employees
ON employees.OFFICECODE = offices.OFFICECODE
GROUP BY offices.officecode, city;
```

Having

31. List the total number of products per product line where number of products > 3 (6)

```
SELECT  products.productline, COUNT(productcode) AS prodperline
FROM products
INNER JOIN productlines
ON products.PRODUCTLINE = productlines.PRODUCTLINE
GROUP BY products.productline
HAVING COUNT(productcode) > 3;
```

32. List the product lines and number of vendors for product lines that are supported by < 10 vendors (2)

```
SELECT  products.productline, COUNT(productvendor) AS prodperline
FROM products
INNER JOIN productlines
ON products.PRODUCTLINE = productlines.PRODUCTLINE
GROUP BY products.productline
HAVING COUNT(productvendor) < 10;
```

Computations

33. What is the total cost per order for orders > 55000.00 sorted by largest total first. (15)

```
SELECT  orders.ordernumber, SUM(priceeach * quantityordered) AS prodperline
FROM orders
INNER JOIN orderdetails
ON orders.ORDERNUMBER = orderdetails.ORDERNUMBER
GROUP BY orders.ORDERNUMBER
HAVING SUM(priceeach * quantityordered) > 55000
ORDER BY prodperline DESC;
```

34. What is the profit per product (MSRP-buyprice) (110)

```
SELECT  productname, (MSRP - buyprice) AS profperprod
FROM products;
```

Set Operations

35. List all customers who didn't order in 2015 (78)

```
SELECT customername FROM customers
   WHERE customernumber
   NOT IN (SELECT customernumber FROM orders WHERE orderdate BETWEEN
'2015-01-01' AND '2015-12-30');
```

36. List all people that we deal with (employees and customer contacts). Display first name, last name, company name (or employee) (145)

```
SELECT customername AS companyname, contactfirstname AS firstname, contactlastname AS
lastname FROM customers   UNION         SELECT 'Employee' AS companyname, firstname
AS firstname, lastname AS lastname FROM employees;
```

37. List the states and the country that the state is part of that have customers but not offices, offices but not customers, or both one or more customers and one or more offices all in one query. Designate which state is which with the string 'Customer', 'Office', or 'Both'. If a state falls into the “Both” category, do not list it as a Customer or an Office state. Order by the country, then the state. Give the category column (where you list ‘Customer’, ‘Office’, or ‘Both’) a header of “Category” and exclude any entries in which the state is null. (19)

```
SELECT DISTINCT 'customer' AS “Category”, state, country FROM customers
   WHERE state IS NOT NULL
UNION
SELECT DISTINCT 'office' AS “Category”, state, country FROM offices
   WHERE state IS NOT NULL
UNION
SELECT DISTINCT 'both' AS “Category”, o.state, o.country FROM offices o
   INNER JOIN customers c ON o.state = c.state
EXCEPT
SELECT DISTINCT 'customer' AS“Category”, o.state, o.country FROM offices o
   INNER JOIN customers c ON o.state = c.state
EXCEPT
SELECT DISTINCT 'office' AS “Category”, o.state, o.country FROM offices o
   INNER JOIN customers c ON o.state = c.state
   ORDER BY country, state;
```

38. List the Product Code and Product name of every product that has never been in on order in which the customer asked for more than 48 of them. Order by the Product Name. (8)

```
SELECT productcode, productname FROM products
EXCEPT
SELECT o.productcode, p.productname FROM orderdetails o
   INNER JOIN products p ON p.PRODUCTCODE = o.PRODUCTCODE
   WHERE quantityordered > 48
   ORDER BY productname;
```

39. List the last name, first name, and employee number of all of the employees who do not have any customers. Order by last name first, then the first name. (8).

```
SELECT lastname, firstname, employeenumber FROM employees
EXCEPT
SELECT DISTINCT lastname, firstname, employeenumber FROM employees e
   INNER JOIN customers c ON e.EMPLOYEENUMBER = c.SALESREPEMPLOYEENUMBER;
```

Subqueries

40. Find the first name and last name of all customer contacts whose customer is located in the same state as the San Francisco office. (11)

```
SELECT contactlastname, contactfirstname FROM customers
   WHERE state = (SELECT state FROM offices WHERE city = 'San Francisco');
```

41. Which products have an MSRP within 5% of the average MSRP across all products? List the Product Name, the MSRP, and the average MSRP ordered by the product MSRP. (14)

```
SELECT productName, MSRP, AVG(MSRP) FROM products
WHERE MSRP < (( SELECT AVG(MSRP) FROM products) + 0.05*( SELECT AVG(MSRP)
FROM products))
AND MSRP > (( SELECT AVG(MSRP) FROM products) - 0.05*( SELECT AVG(MSRP) FROM
products)); //doesnt work

SELECT productname, msrp, productline FROM products
   WHERE (msrp BETWEEN ((SELECT AVG(msrp) FROM products) - (SELECT AVG(msrp) *
.05 FROM products))
   AND ((SELECT AVG(msrp) FROM products) + (SELECT AVG(msrp) * .05 FROM products)))
   ORDER BY msrp;
```

42. Which customer made the largest individual payment. (1)

```
SELECT * FROM customers c
   INNER JOIN payments p ON c.CUSTOMERNUMBER = p.CUSTOMERNUMBER
   WHERE p.amount = (SELECT MAX(amount) FROM payments);
```

43. Which customers made an individual payment where half of the payment is more than average payment. List their name. (8)

```
SELECT DISTINCT contactlastname, contactfirstname FROM customers c
   INNER JOIN payments p ON c.CUSTOMERNUMBER = p.CUSTOMERNUMBER
   WHERE (amount/2) > (SELECT AVG(amount) FROM payments);
```

44. Which orderline is the smallest in terms of money. List the order number, orderline number and the money involved on that orderline. (1)

```
SELECT ordernumber, orderlinenumber, (quantityordered * priceeach) AS totalcost FROM
orderdetails
   WHERE (quantityordered * priceeach) = (SELECT MIN(quantityordered * priceeach) FROM
orderdetails);
```

Recursion

45. List all employ

```
SELECT EMPLOYEENUMBER FROM employees
WHERE reportsto IN
(SELECT reportsto FROM employees
   GROUP BY reportsto
   HAVING COUNT(employeenumber) =
      (SELECT MAX(c.employeecount)
      FROM (SELECT reportsto, COUNT(employeenumber) AS employeecount
      FROM employees
      GROUP BY reportsto) c));
```

ees that have the same last name. Make sure each combination is listed only once (5)

```
Select L.lastName, L.employeeNumber, R.lastName, R.employeeNumber
from employees L
inner join employees R
ON( L.lastName  = R.lastName and L.employeeNumber > R.employeeNumber);
```

46. List all the first and last name of all employees and their managers. Order by employee last name, then first name. (22)

```
Select L.lastName, L.firstname, l.reportsto, r.employeenumber, R.lastName, R.firstname
from employees L
inner join employees R
ON( L.reportsto  = R.employeenumber )
order by l.lastname, l.firstname;
```

Extra

47. What product that makes us the most money (qty*price) (1)

```
SELECT p.productcode, p.productname, quantityordered * priceeach FROM orderdetails o
   INNER JOIN products p ON o.PRODUCTCODE = p.PRODUCTCODE
   WHERE (quantityordered * priceeach) = (SELECT MAX(quantityordered * priceeach) FROM
orderdetails);
```

48. List the products in the product line with the most number of products (38)

```
SELECT productname FROM products WHERE productline = (SELECT productline
FROM products
GROUP BY productline
HAVING COUNT(productline) = (SELECT MAX(c.productcount)
FROM (SELECT productline, COUNT(productline) AS productcount
FROM products
GROUP BY productline) c));
```

49. What is the customer and sales person of the highest priced order? (1)

```
SELECT DISTINCT od.ordernumber, c.CUSTOMERNAME, e.LASTNAME, e.FIRSTNAME
FROM orderdetails od
   INNER JOIN orders o ON od.ORDERNUMBER = o.ORDERNUMBER
   INNER JOIN customers c ON o.CUSTOMERNUMBER = c.CUSTOMERNUMBER
   INNER JOIN employees e ON c.SALESREPEMPLOYEENUMBER = e.EMPLOYEENUMBER
   WHERE od.ordernumber = (SELECT ordernumber FROM orderdetails
      WHERE (quantityordered * priceeach) = (SELECT MAX(quantityordered * priceeach) AS
total FROM orderdetails));
```

50. What is the manager who manages the greatest number of employees (2)

```
SELECT reportsto, COUNT(employeenumber) AS employeecount FROM employees
GROUP BY reportsto
HAVING COUNT(employeenumber) =
   (SELECT MAX(c.employeecount) FROM (SELECT reportsto, COUNT(employeenumber) AS
employeecount FROM employees GROUP BY reportsto) c);
```

51. Select all employees who work for the manager that manages the greatest number of employees (12)

```
SELECT * FROM employees
WHERE reportsto IN (SELECT reportsto FROM employees
   GROUP BY reportsto
   HAVING COUNT(employeenumber) =
      (SELECT MAX(c.employeecount) FROM (SELECT reportsto, COUNT(employeenumber)
AS employeecount FROM employees GROUP BY reportsto) c));
```

```
3. SELECT p.productname, quantityinstock * MSRP FROM products P
```

```
WHERE (quantityinstock * msrp) = (SELECT min(quantityinstock * msrp) FROM products);
```