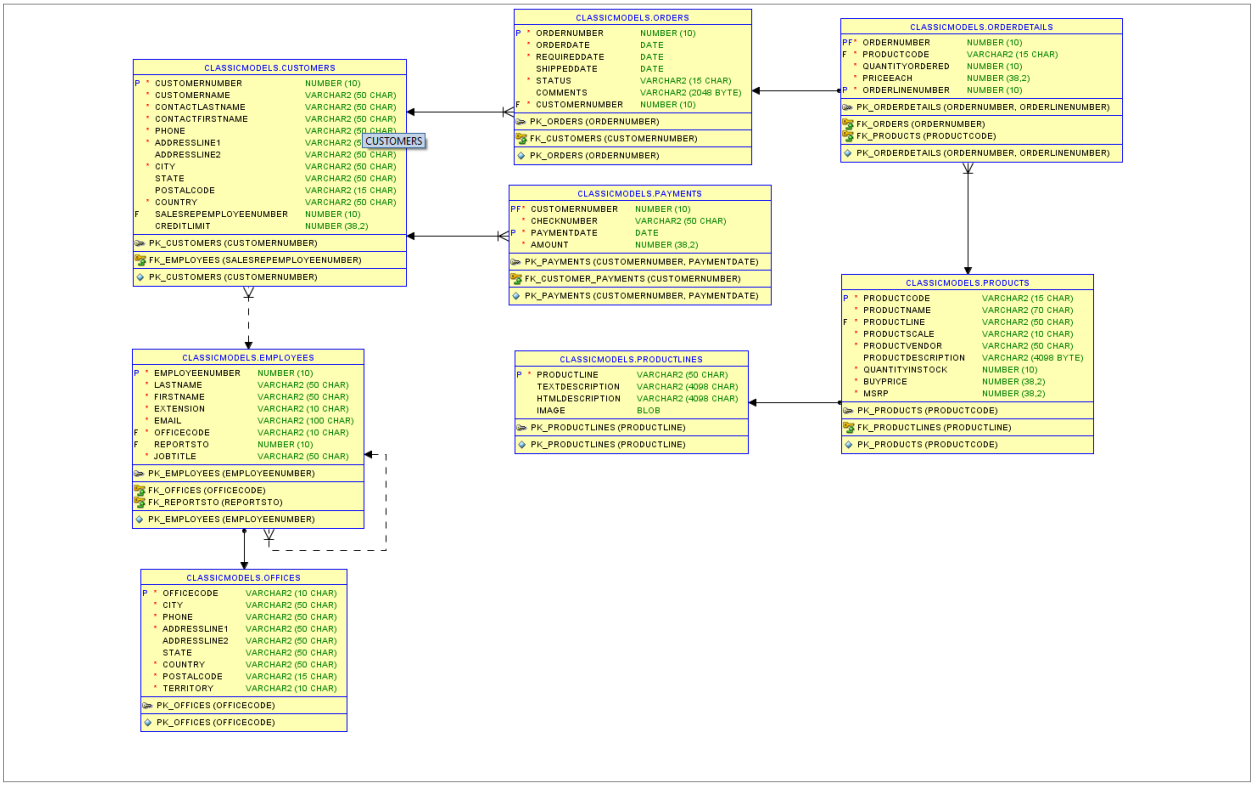
ANIL POONAI

CIS 4400 Assignment 1

3.



4.

```
SELECT CUSTOMERNAME, STATE, SUM(QUANTITYORDERED*PRICEEACH)

FROM CLASSICMODELS.CUSTOMERS INNER JOIN CLASSICMODELS.ORDERS ON
CLASSICMODELS.CUSTOMERS.CUSTOMERNUMBER = CLASSICMODELS.ORDERS.CUSTOMERNUMBER

    INNER JOIN CLASSICMODELS.ORDERDETAILS ON CLASSICMODELS.ORDERS.ORDERNUMBER =
CLASSICMODELS.ORDERDETAILS.ORDERNUMBER

    INNER JOIN CLASSICMODELS.PRODUCTS ON CLASSICMODELS.ORDERDETAILS.PRODUCTCODE =
CLASSICMODELS.PRODUCTS.PRODUCTCODE

WHERE CREDITLIMIT > (SELECT AVG(CREDITLIMIT) FROM CLASSICMODELS.CUSTOMERS) AND
PRODUCTNAME='1962 Volkswagen Microbus'

GROUP BY CUSTOMERNAME,STATE;
```

QUERY RESULTS

Did not know how to copy and paste the query so I took a screenshot.

	CUSTOMERNAME	STATE	SUM(QUANTITYORDERED*PRICEEACH)
1	Vitachrome Inc.	NY	3327.54
2	La Rochelle Gifts	(null)	6261.71
3	Saveley and Henriot, Co.	(null)	5510.54
4	Scandinavian Gift Ideas	(null)	5275.2
5	Canadian Gift Exchange Network	BC	4293.66
6	Euro+ Shopping Channel	(null)	3446.36
7	Corrida Auto Replicas, Ltd	(null)	7283.94
8	Technics Stores Inc.	CA	5585.48
9	Oulu Toy Supplies, Inc.	(null)	4615.8
10	Diecast Classics Inc.	PA	2811.5
11	Dragon Souveniers, Ltd.	(null)	3762.24
12	Stylish Desk Decors, Co.	(null)	3399.2
13	Handji Gifts Co	(null)	8604.08
14	Muscle Machine Inc	NY	7693
15	Baane Mini Imports	(null)	9103.63
16	La Corne D'abondance, Co.	(null)	2146.83
17	Diecast Collectables	MA	5878.34
18	Down Under Souveniers, Inc	(null)	3711.18
19	Marta's Replicas Co.	MA	2576.16
20	Mini Gifts Distributors Ltd.	CA	15728.32

5.

```
SELECT COUNTRY, TO_CHAR(ORDERDATE, 'YYYY-MM') AS Month,
SUM(QUANTITYORDERED*PRICEEACH),

    RANK() OVER(ORDER BY SUM(QUANTITYORDERED*PRICEEACH) DESC) Rank

FROM CLASSICMODELS.CUSTOMERS INNER JOIN CLASSICMODELS.ORDERS ON
CLASSICMODELS.CUSTOMERS.CUSTOMERNUMBER = CLASSICMODELS.ORDERS.CUSTOMERNUMBER

    INNER JOIN CLASSICMODELS.ORDERDETAILS ON CLASSICMODELS.ORDERS.ORDERNUMBER =
CLASSICMODELS.ORDERDETAILS.ORDERNUMBER

    INNER JOIN CLASSICMODELS.PRODUCTS ON CLASSICMODELS.ORDERDETAILS.PRODUCTCODE =
CLASSICMODELS.PRODUCTS.PRODUCTCODE

GROUP BY COUNTRY,TO_CHAR(ORDERDATE, 'YYYY-MM');
```

QUERY RESULTS

197 Rows

	COUNTRY	MONTH	SUM(QUANTITYORDERED*PRICEEACH)	RANK
1	USA	2019-11	377124.68	1
2	USA	2018-11	290655.54	2
3	USA	2019-08	215256.19	3
4	USA	2020-05	197576.85	4
5	USA	2018-10	193241.34	5
6	USA	2020-02	173722.9	6
7	USA	2019-10	168723.81	7
8	USA	2020-03	160860.33	8
9	USA	2019-05	159429.06	9
10	USA	2018-12	149660.85	10
11	USA	2018-08	147412.05	11
12	USA	2020-01	137915.19	12
13	Spain	2018-11	130137.12	13
14	USA	2020-04	127509.93	14
15	USA	2019-12	116603.76	15
16	France	2018-11	116314.25	16
17	New Zealand	2020-04	108122.9	17
18	France	2019-01	98779.88	18
19	Australia	2018-11	97989.35	19

6.

```
SELECT PRODUCTLINE, TO_CHAR(ORDERDATE, 'YYYY-MM') AS Month,
SUM(QUANTITYORDERED*PRICEEACH) AS TOTAL, lag(SUM(QUANTITYORDERED*PRICEEACH)) OVER (
PARTITION BY PRODUCTLINE ORDER BY TO_CHAR(ORDERDATE, 'YYYY-MM'))

AS LAG,SUM(QUANTITYORDERED*PRICEEACH)/lag(SUM(QUANTITYORDERED*PRICEEACH)) OVER (
PARTITION BY PRODUCTLINE ORDER BY TO_CHAR(ORDERDATE, 'YYYY-MM')) AS MOVERM

FROM CLASSICMODELS.ORDERS INNER JOIN CLASSICMODELS.ORDERDETAILS ON
CLASSICMODELS.ORDERS.ORDERNUMBER = CLASSICMODELS.ORDERDETAILS.ORDERNUMBER

    INNER JOIN CLASSICMODELS.PRODUCTS ON CLASSICMODELS.ORDERDETAILS.PRODUCTCODE =
CLASSICMODELS.PRODUCTS.PRODUCTCODE

GROUP BY PRODUCTLINE,TO_CHAR(ORDERDATE, 'YYYY-MM');
```

QUERY RESULTS

182 Rows

I filtered it so that the quarter was set to '2020-05', the latest one, and made the Month-Over-Month sales in descending order. That is how I came up with the answer: Ships has the largest percentage Month-Over-Month sales.

	PRODUCTLINE	M...	TOTAL	LAG	MOVERM
1	Ships	2020-05	25432.22	6034.22	4.214665690014616636450112856342659034 64
2	Trucks an...	2020-05	80733.59	35635.57	2.265533847220628153274944107811380595 29
3	Trains	2020-05	8190.09	6489.69	1.262015597047008408722142351945932702 49
4	Classic Cars	2020-05	177674.87	145358.39	1.222322770636080930725773723828394081 69
5	Vintage Cars	2020-05	70964.85	69839.89	1.016107700055083133722003284942172732 52
6	Planes	2020-05	29307.94	39870.99	0.735069282202423365961065927883907572 8995
7	Motorcycles	2020-05	49171.38	83717.13	0.587351477529150844038728991306797067 6969

7.

```
SELECT TO_CHAR(ORDERDATE, 'YYYY-MM') AS Month,SUM(QUANTITYORDERED*PRICEEACH-
QUANTITYORDERED*BUYPRICE)

FROM CLASSICMODELS.ORDERS INNER JOIN CLASSICMODELS.ORDERDETAILS ON
CLASSICMODELS.ORDERS.ORDERNUMBER = CLASSICMODELS.ORDERDETAILS.ORDERNUMBER

    INNER JOIN CLASSICMODELS.PRODUCTS ON CLASSICMODELS.ORDERDETAILS.PRODUCTCODE =
CLASSICMODELS.PRODUCTS.PRODUCTCODE

WHERE ORDERDATE LIKE '%19'

GROUP BY TO_CHAR(ORDERDATE, 'YYYY-MM');
```

QUERY RESULTS

So, I chose November as the most profitable month of 2019 on the basis of the amount of money they sold products for multiplied by the amount of products sold minus the amount of money spent on products multiplied by the amount of products sold. Now there is a different a way of doing this, I could've took the inventory of one month and subtracted it by the inventory the month before but this doesn't work as the price of inventory doesn't hold the same amount of weight as actual cash. So, I used the amount of products sold since those aren't standing items anymore and they actually got paid money for them.

	MONTH	SUM(QUANTITYORDERED*PRICEEACH-QUANTITYORDERED*BUYPRI...
1	2019-11	392370.92
2	2019-10	202491.52
3	2019-12	167207.98
4	2019-08	166758.58
5	2019-06	139022.15
6	2019-07	128195.79
7	2019-01	119453.03
8	2019-09	114683.65
9	2019-02	113543.35
10	2019-05	103150.02
11	2019-03	86600.36
12	2019-04	75903.79

8.

```
SELECT
CUSTOMERNAME,SUM(QUANTITYORDERED*PRICEEACH)/COUNT(CLASSICMODELS.ORDERS.ORDERNUMBER) AS AOV,

COUNT(CLASSICMODELS.ORDERS.ORDERNUMBER)/98 AS PF,SUM(QUANTITYORDERED*PRICEEACH)/COUNT(CLASSICMODELS.ORDERS.ORDERNUMBER) * COUNT(CLASSICMODELS.ORDERS.ORDERNUMBER)/98 AS CLV

FROM CLASSICMODELS.CUSTOMERS INNER JOIN CLASSICMODELS.ORDERS ON CLASSICMODELS.CUSTOMERS.CUSTOMERNUMBER = CLASSICMODELS.ORDERS.CUSTOMERNUMBER

    INNER JOIN CLASSICMODELS.ORDERDETAILS ON CLASSICMODELS.ORDERS.ORDERNUMBER = CLASSICMODELS.ORDERDETAILS.ORDERNUMBER

    INNER JOIN CLASSICMODELS.PRODUCTS ON CLASSICMODELS.ORDERDETAILS.PRODUCTCODE = CLASSICMODELS.PRODUCTS.PRODUCTCODE

GROUP BY CUSTOMERNAME;
```

QUERY RESULTS

98 Rows

I used the Customer Lifetime Value to determine this. First I calculated their Average Order Value by dividing the total sales by their total orders. Then I calculated their Purchase Frequency by dividing their total amount of orders by the number of distinct customers that exist. Then I multiplied both the Average Order Value and the Purchase Frequency to get the Customer Lifetime Value. Then I just filtered it by highest to lowest and got Mini Gifts Distributors Ltd. as their best customer.

	CUSTOMERNAME	AOV	PF	CLV
1	Mini Gifts Distributors Ltd.	3287.92966666666666666666666666666666666667	1.836734693877551020408163265306122448898	6039.05448979591836734693877551020408163 4
2	Muscle Machine Inc	3486.25513043478260869565217391304347826 1	1.173469387755102040816326530612244897796	4091.013673469387755102040816326530612245
3	Baane Mini Imports	3125.701212121212121212121212121212121212	1.010204081632653061224489795918367346694	3157.596122448979591836734693877551020408
4	Euro+ Shopping Channel	3088.62	0.969387755102040816326530612244897959183 7	2994.070408163265306122448979591836734694
5	Australian Collectors, Co.	3283.364909090909090909090909090909090909	0.561224489795918367346938775510204081632 7	1842.704795918367346938775510204081632653
6	La Rochelle Gifts	2991.945660377358490566037735849056603774	0.540816326530612244897959183673469387755 1	1618.093061224489795918367346938775510204
7	Dragon Souveniers, Ltd.	3633.744883720930232558139534883720930233	0.438775510204081632653061224489795918367 3	1594.398265306122448979591836734693877551
8	Down Under Souveniers, Inc	3361.349565217391304347826086956521739 13	0.469387755102040816326530612244897959183 7	1577.776326530612244897959183673469387755
9	Land of Toys Inc.	3042.554081632653061224489795918367346939	0.5	1521.277040816326530612244897959183673469
10	Clover Collections, Co.	3237.910652173913043478260869565217391304	0.469387755102040816326530612244897959183 7	1519.835612244897959183673469387755102041
11	AV Stores, Co.	2910.001764705882352941176470588235294118	0.520408163265306122448979591836734693877 6	1514.388673469387755102040816326530612245
12	The Sharp Gifts Warehouse	3588.40675	0.408163265306122448979591836734693877551	1464.655816326530612244897959183673469388
13	Salzburg Collectables	3437.00175	0.408163265306122448979591836734693877551	1402.857857142857142857142857142857142857
14	Kelly's Gift Shop	2863.766458333333333333333333333333333333 3	0.489795918367346938775510204081632653061 2	1402.661122448979591836734693877551020408
15	Anna's Decorations, Ltd	2979.004782608695652173913043478260869565	0.469387755102040816326530612244897959183 7	1398.308367346938775510204081632653061224
16	Souveniers And Things Co.	2911.024347826086956521739130434782608696	0.469387755102040816326530612244897959183 7	1366.399183673469387755102040816326530612
17	Corporate Gift Ideas Co.	3227.823902439024390243902439024390243902	0.418367346938775510204081632653061224489 8	1350.416122448979591836734693877551020408
18	Saveley and Henriot, Co.	3178.179268292682926829268292682926829268	0.418367346938775510204081632653061224489 8	1329.646428571428571428571428571428571429
19	Danish Wholesale Imports	3585.697777777777777777777777777777777778	0.367346938775510204081632653061224489795 9	1317.195102040816326530612244897959183673

9.

2996 Rows

I realize I could have just made some temporary columns and get rid of the columns that were duplicates but at the time just typing it all out seemed easier to do.

```
CREATE VIEW Number9 AS

SELECT CLASSICMODELS.CUSTOMERS.CUSTOMERNUMBER, CUSTOMERNAME, CONTACTLASTNAME,
CONTACTFIRSTNAME, PHONE, ADDRESSLINE1, ADDRESSLINE2, CITY, STATE, POSTALCODE, COUNTRY,
SALESREPEMPLOYEENUMBER, CREDITLIMIT,

LASTNAME, FIRSTNAME, EXTENSION, EMAIL, OFFICECODE, REPORTSTO, JOBTITLE,

CLASSICMODELS.ORDERS.ORDERNUMBER, ORDERDATE, REQUIREDDATE, SHIPPEDDATE, STATUS,
COMMENTS,

CLASSICMODELS.ORDERDETAILS.PRODUCTCODE, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER,

PRODUCTNAME, PRODUCTLINE, PRODUCTSCALE, PRODUCTVENDOR, PRODUCTDESCRIPTION,
QUANTITYINSTOCK, BUYPRICE, MSRP

FROM CLASSICMODELS.CUSTOMERS, CLASSICMODELS.ORDERS , CLASSICMODELS.EMPLOYEES ,
CLASSICMODELS.ORDERDETAILS , CLASSICMODELS.PRODUCTS

WHERE
CLASSICMODELS.CUSTOMERS.CUSTOMERNUMBER=CLASSICMODELS.ORDERS.CUSTOMERNUMBER AND
CLASSICMODELS.CUSTOMERS.SALESREPEMPLOYEENUMBER=CLASSICMODELS.EMPLOYEES.EMPLOYEEN
UMBER

AND CLASSICMODELS.ORDERS.ORDERNUMBER=CLASSICMODELS.ORDERDETAILS.ORDERNUMBER AND
CLASSICMODELS.ORDERDETAILS.PRODUCTCODE=CLASSICMODELS.PRODUCTS.PRODUCTCODE;
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE
1	CUSTOMERNAME	VARCHAR2(50 CHAR)	No	(null)	2	(null)	NO	NO	NO
2	CONTACTLASTNAME	VARCHAR2(50 CHAR)	No	(null)	3	(null)	NO	NO	NO
3	CONTACTFIRSTNAME	VARCHAR2(50 CHAR)	No	(null)	4	(null)	NO	NO	NO
4	CUSTOMERNUMBER	NUMBER(10)	No	(null)	1	(null)	NO	NO	NO
5	PHONE	VARCHAR2(50 CHAR)	No	(null)	5	(null)	NO	NO	NO
6	ADDRESSLINE1	VARCHAR2(50 CHAR)	No	(null)	6	(null)	NO	NO	NO
7	ADDRESSLINE2	VARCHAR2(50 CHAR)	Yes	(null)	7	(null)	NO	NO	NO
8	CITY	VARCHAR2(50 CHAR)	No	(null)	8	(null)	NO	NO	NO
9	STATE	VARCHAR2(50 CHAR)	Yes	(null)	9	(null)	NO	NO	NO
10	POSTALCODE	VARCHAR2(15 CHAR)	Yes	(null)	10	(null)	NO	NO	NO
11	COUNTRY	VARCHAR2(50 CHAR)	No	(null)	11	(null)	NO	NO	NO
12	SALESREPEMPLOYEENUMBER	NUMBER(10)	Yes	(null)	12	(null)	NO	NO	NO
13	CREDITLIMIT	NUMBER(38,2)	Yes	(null)	13	(null)	NO	NO	NO
14	LASTNAME	VARCHAR2(50 CHAR)	No	(null)	14	(null)	NO	NO	NO
15	FIRSTNAME	VARCHAR2(50 CHAR)	No	(null)	15	(null)	NO	NO	NO
16	EXTENSION	VARCHAR2(10 CHAR)	No	(null)	16	(null)	NO	NO	NO
17	EMAIL	VARCHAR2(100 CHAR)	No	(null)	17	(null)	NO	NO	NO
18	OFFICECODE	VARCHAR2(10 CHAR)	No	(null)	18	(null)	NO	NO	NO
19	REPORTSTO	NUMBER(10)	Yes	(null)	19	(null)	NO	NO	NO
20	JOBTITLE	VARCHAR2(50 CHAR)	No	(null)	20	(null)	NO	NO	NO
21	ORDERNUMBER	NUMBER(10)	No	(null)	21	(null)	NO	NO	NO
22	ORDERDATE	DATE	No	(null)	22	(null)	NO	NO	NO
23	REQUIREDDATE	DATE	No	(null)	23	(null)	NO	NO	NO
24	SHIPPEDDATE	DATE	Yes	(null)	24	(null)	NO	NO	NO
25	STATUS	VARCHAR2(15 CHAR)	No	(null)	25	(null)	NO	NO	NO
26	COMMENTS	VARCHAR2(2048)	Yes	(null)	26	(null)	NO	NO	NO
27	PRODUCTCODE	VARCHAR2(15 CHAR)	No	(null)	27	(null)	YES	YES	YES
28	QUANTITYORDERED	NUMBER(10)	No	(null)	28	(null)	YES	YES	YES
29	PRICEEACH	NUMBER(38,2)	No	(null)	29	(null)	YES	YES	YES
30	ORDERLINENUMBER	NUMBER(10)	No	(null)	30	(null)	YES	YES	YES
31	PRODUCTNAME	VARCHAR2(70 CHAR)	No	(null)	31	(null)	NO	NO	NO
32	PRODUCTLINE	VARCHAR2(50 CHAR)	No	(null)	32	(null)	NO	NO	NO
33	PRODUCTSCALE	VARCHAR2(10 CHAR)	No	(null)	33	(null)	NO	NO	NO
34	PRODUCTVENDOR	VARCHAR2(50 CHAR)	No	(null)	34	(null)	NO	NO	NO
35	PRODUCTDESCRIPTION	VARCHAR2(4098)	Yes	(null)	35	(null)	NO	NO	NO
36	QUANTITYINSTOCK	NUMBER(10)	No	(null)	36	(null)	NO	NO	NO
37	BUYPRICE	NUMBER(38,2)	No	(null)	37	(null)	NO	NO	NO
38	MSRP	NUMBER(38,2)	No	(null)	38	(null)	NO	NO	NO

10.

So below is the pivot table ordered by PRODUCTLINE and then again by COUNTRY. I made a new column in the spreadsheet by multiplying the QUANTITYORDERED by the PRICEEACH for every product in every order. This gave me a column with the amount of dollar worth, which when added to the pivot table shows the amount by country and product line.

3	Row Labels	Sum of Total Dollars Ordered
4	**Classic Cars**	**3853922.49**
5	Australia	187965.47
6	Austria	101526.38
7	Belgium	18459.9
8	Canada	59659.68
9	Denmark	140725.84
10	Finland	143593.36
11	France	394961.42
12	Germany	132046.93
13	Ireland	86244.29
14	Italy	120419.99
15	Japan	41138.72
16	New Zealand	151699.02
17	Norway	157384.94
18	Norway	91868.43
19	Philippines	51985.52
20	Singapore	118486.14
21	Spain	199697.58
22	Sweden	66590.19
23	Switzerland	108777.92
24	UK	147172.77
25	USA	1333518
26	**Motorcycles**	**1121426.12**
27	**Planes**	**954637.54**
28	**Ships**	**663998.34**
29	**Trains**	**188532.92**
30	**Trucks and Buses**	**1024113.57**
31	**Vintage Cars**	**1797559.63**
32	**Grand Total**	**9604190.61**

11.

So, I ended up converting CREDITLIMIT to a dimension instead of a calculated field. Then I put PRODUCTLINE as a row and added CREDITLIMT as a column. This way we could see the differences in the TOTAL DOLLARS ORDERED by each PRODUCTLINE as it moved along the different CREDITLIMIT's. I ended up placing TOTAL DOLLARS ORDERED as a row and found that a bar graph best showed the differences.