MySQL Assignment-2

1. Write a query to calculate the total number of products in the database.

```
mysql> use gradious;
Database changed
mysql> select count(productCode) as totalProducts from products;
+---------------+
| totalProducts |
+---------------+
|           110 |
+---------------+
1 row in set (0.00 sec)
```

2. Write a query to find the average buy price of all products.

```
mysql> select avg(buyPrice) as avgProducts from products;
+--------------+
| avgProducts |
+--------------+
|   54.395182 |
+--------------+
1 row in set (0.00 sec)
```

3. Write a query to determine the maximum quantity in stock across all products.

```
mysql> select productCode, quantityInStock as maxQuality from products
    -> where quantityInStock = (select max(quantityInStock) from products);
+-------------+------------+
| productCode | maxQuality |
+-------------+------------+
| S12_2823    |       9997 |
+-------------+------------+
1 row in set (0.00 sec)
```

4. Write a query to calculate the total sales revenue for each line.

```
mysql> select ps.productLine, sum( ods.quantityOrdered * ods.priceEach ) as totalSalesRevenue
    -> from orderdetails ods inner join products ps
    -> on ps.productCode = ods.productCode
    -> group by ps.productLine;
+------------------+-------------------+
| productLine      | totalSalesRevenue |
+------------------+-------------------+
| Classic Cars     |        3853922.49 |
| Motorcycles      |        1121426.12 |
| Planes           |         954637.54 |
| Ships            |         663998.34 |
| Trains           |         188532.92 |
| Trucks and Buses |        1024113.57 |
| Vintage Cars     |        1797559.63 |
+------------------+-------------------+
7 rows in set (0.01 sec)
```

5. Write a query to determine the average credit limit for all customers.

```
mysql> select avg(creditLimit) as avgCreditLimit from customers;
+-----------------+
| avgCreditLimit |
+-----------------+
|   67659.016393 |
+-----------------+
1 row in set (0.00 sec)
```

6. Write a query to find the highest payment amount made by a customer.

```
mysql> select max(amount) as highestPayment from payments;
+----------------+
| highestPayment |
+----------------+
|      120166.58 |
+----------------+
1 row in set (0.00 sec)
```

7. write a query to calculate the total quantity ordered for each product.

```
mysql> select productCode, sum(quantityOrdered) as totalProducts
    -> from orderdetails group by productCode;
+-------------+---------------+
| productCode | totalProducts |
+-------------+---------------+
| S10_1678    |          1057 |
| S10_1949    |           961 |
| S10_2016    |           999 |
| S10_4698    |           985 |
| S10_4757    |          1030 |
| S10_4962    |           932 |
| S12_1099    |           933 |
| S12_1108    |          1019 |
| S12_1666    |           972 |
| S12_2823    |          1028 |
| S12_3148    |           963 |
| S12_3380    |           925 |
| S12_3891    |           965 |
| S12_3990    |           900 |
| S12_4473    |          1056 |
| S12_4675    |           992 |
| S18_1097    |           999 |
| S18_1129    |           947 |
| S18_1342    |          1111 |
| S18_1367    |           960 |
| S18_1589    |           914 |
| S18_1662    |          1040 |
| S18_1749    |           918 |
| S18_1889    |           972 |
| S18_1984    |           917 |
| S18_2238    |           986 |
| S18_2248    |           832 |
| S18_2319    |          1053 |
| S18_2325    |           957 |
| S18_2432    |           998 |
| S18_2581    |           917 |
| S18_2625    |           945 |
| S18_2795    |           880 |
| S18_2870    |           855 |
| S18_2949    |          1038 |
```

S24_3151	991
S24_3191	870
S24_3371	969
S24_3420	884
S24_3432	894
S24_3816	923
S24_3856	1052
S24_3949	1051
S24_3969	824
S24_4048	867
S24_4258	983
S24_4278	1009
S24_4620	941
S32_1268	911
S32_1374	1014
S32_2206	906
S32_2509	955
S32_3207	934
S32_3522	988
S32_4289	972
S32_4485	898
S50_1341	1074
S50_1392	979
S50_1514	966
S50_4713	992
S700_1138	934
S700_1691	894
S700_1938	898
S700_2047	897
S700_2466	984
S700_2610	1020
S700_2824	997
S700_2834	973
S700_3167	1047
S700_3505	952
S700_3962	896
S700_4002	1085
S72_1253	960
S72_3212	958

109 rows in set (0.01 sec)

8. write a query to determine the of employees in each office.

```
mysql> select officeCode, count(*) as totalEmployees from employees group by officeCode;
+------------+----------------+
| officeCode | totalEmployees |
+------------+----------------+
| 1          |              6 |
| 2          |              2 |
| 3          |              2 |
| 4          |              5 |
| 5          |              2 |
| 6          |              4 |
| 7          |              2 |
+------------+----------------+
7 rows in set (0.00 sec)
```

9. write a query to calculate the average price for each order.

```
mysql> select orderNumber, avg(quantityOrdered * priceEach) as avgPrice
    -> from orderdetails group by orderNumber;
+-------------+-------------+
| orderNumber | avgPrice    |
+-------------+-------------+
|       10100 | 2555.957500 |
|       10101 | 2637.252500 |
|       10102 | 2747.390000 |
|       10103 | 3138.684375 |
|       10104 | 3092.784615 |
|       10105 | 3597.280667 |
|       10106 | 2897.322778 |
|       10107 | 2786.577500 |
|       10108 | 3187.576250 |
|       10109 | 4305.523333 |
|       10110 | 3026.605625 |
|       10111 | 2756.308333 |
|       10112 | 3837.470000 |
|       10113 | 2761.075000 |
|       10114 | 3338.314000 |
|       10115 | 4333.196000 |
|       10116 | 1627.560000 |
|       10117 | 3698.345833 |
|       10118 | 3101.400000 |
|       10119 | 2559.023571 |
|       10120 | 3057.602000 |
|       10121 | 3340.094000 |
|       10122 | 2989.685882 |
|       10123 | 3642.860000 |
|       10124 | 2510.921538 |
|       10125 | 3782.540000 |
|       10126 | 3360.701176 |
|       10127 | 3922.756667 |
|       10128 | 3471.247500 |
|       10129 | 3269.904444 |
|       10130 | 3018.480000 |
|       10131 | 2129.036250 |
|       10132 | 2880.000000 |
|       10133 | 2795.755000 |
|       10134 | 3345.638571 |
```

```
|       10387 | 3516.040000 |
|       10388 | 3786.721250 |
|       10389 | 3495.817500 |
|       10390 | 3493.906250 |
```

```
mysql> select cs.country, sum( ods.quantityOrdered * ods.priceEach ) as totalSalesRevenue
    -> from orders os inner join customers cs
    -> on  os.customerNumber = cs.customerNumber
    -> inner join orderdetails ods
    -> on os.orderNumber = ods.orderNumber
    -> group by cs.country;
+-------------+--------------------+
| country     | totalSalesRevenue |
+-------------+--------------------+
| France      |        1007374.02 |
| USA         |        3273280.05 |
| Australia   |         562582.59 |
| Norway      |         270846.30 |
| Germany     |         196470.99 |
| Spain       |        1099389.09 |
| Sweden      |         187638.35 |
| Denmark     |         218994.92 |
| Singapore   |         263997.78 |
| Japan       |         167909.95 |
| Finland     |         295149.35 |
| UK          |         436947.44 |
| Ireland     |          49898.27 |
| Canada      |         205911.86 |
| Hong Kong   |          45480.79 |
| Italy       |         360616.81 |
| Switzerland |         108777.92 |
| Belgium     |         100068.76 |
| New Zealand |         476847.01 |
| Austria     |         188540.06 |
| Philippines |          87468.30 |
+-------------+--------------------+
21 rows in set (0.01 sec)
```

10. Write
each cou

11. Writ
for each

```
|       10411 | 3230.042222 |
|       10412 | 4263.225455 |
|       10413 | 4750.130000 |
|       10414 | 3629.060714 |
|       10415 | 2189.052000 |
|       10416 | 2525.875714 |
|       10417 | 4762.483333 |
|       10418 | 2625.271111 |
|       10419 | 3744.290714 |
|       10420 | 3250.116154 |
|       10421 | 3819.550000 |
|       10422 | 2924.720000 |
|       10423 | 1719.546000 |
|       10424 | 4885.050000 |
|       10425 | 3201.803077 |
+-------------+-------------+
326 rows in set (0.00 sec)
```

```
mysql> select productLine, avg(quantityInStock) as avgQuantityInStock
    -> from products group by productLine;
+------------------+--------------------+
| productLine      | avgQuantityInStock |
+------------------+--------------------+
| Classic Cars     |          5767.9737 |
| Motorcycles      |          5338.5385 |
| Planes           |          5190.5833 |
| Ships            |          2981.4444 |
| Trains           |          5565.3333 |
| Trucks and Buses |          3259.1818 |
| Vintage Cars     |          5203.3333 |
+------------------+--------------------+
7 rows in set (0.00 sec)
```

12. Write a query to determine the total number of orders placed by each customer.

```
mysql> select cs.customerName, count(os.orderNumber ) as totalSalesRevenue
    -> from customers cs left join orders os
    -> on  os.customerNumber = cs.customerNumber
    -> group by cs.customerName;
+-----------------------------------+--------------------+
| customerName                      | totalSalesRevenue |
+-----------------------------------+--------------------+
| Atelier graphique                 |                  3 |
| Signal Gift Stores                |                  3 |
| Australian Collectors, Co.        |                  5 |
| La Rochelle Gifts                 |                  4 |
| Baane Mini Imports                |                  4 |
| Mini Gifts Distributors Ltd.      |                 17 |
| Havel & Zbyszek Co                |                  0 |
| Blauer See Auto, Co.              |                  4 |
| Mini Wheels Co.                   |                  3 |
| Land of Toys Inc.                 |                  4 |
| Euro+ Shopping Channel            |                 26 |
| Volvo Model Replicas, Co          |                  4 |
| Danish Wholesale Imports          |                  5 |
| Saveley & Henriot, Co.            |                  3 |
| Dragon Souveniers, Ltd.           |                  5 |
| Muscle Machine Inc                |                  4 |
| Diecast Classics Inc.             |                  4 |
| Technics Stores Inc.              |                  4 |
| Handji Gifts& Co                  |                  4 |
| Herkku Gifts                      |                  3 |
| American Souvenirs Inc            |                  0 |
| Porto Imports Co.                 |                  0 |
| Daedalus Designs Imports          |                  2 |
| La Corne D'abondance, Co.         |                  3 |
| Cambridge Collectables Co.        |                  2 |
| Gift Depot Inc.                   |                  3 |
| Osaka Souveniers Co.              |                  2 |
| Vitachrome Inc.                   |                  3 |
| Toys of Finland, Co.              |                  3 |
| AV Stores, Co.                    |                  3 |
| Clover Collections, Co.           |                  2 |
| Auto-Moto Classics Inc.           |                  3 |
| UK Collectables, Ltd.             |                  3 |
```

Kommission Auto	0
Gifts4AllAges.com	3
Online Diecast Creations Co.	3
Lisboa Souveniers, Inc	0
Precious Collectables	0
Collectables For Less Inc.	3
Royale Belge	4
Salzburg Collectables	4
Cruz & Sons Co.	3
L'ordine Souveniers	3
Tokyo Collectables, Ltd	4
Auto Canal+ Petit	3
Stuttgart Collectable Exchange	0
Extreme Desk Decorations, Ltd	3
Bavarian Collectables Imports, Co.	1
Classic Legends Inc.	3
Feuer Online Stores, Inc	0
Gift Ideas Corp.	3
Scandinavian Gift Ideas	3
The Sharp Gifts Warehouse	4
Mini Auto Werke	3
Super Scale Inc.	2
Microscale Inc.	2
Corrida Auto Replicas, Ltd	3
Warburg Exchange	0
FunGiftIdeas.com	3
Anton Designs, Ltd.	0
Australian Collectables, Ltd	3
Frau da Collezione	2
West Coast Collectables Co.	2
Mit VergnÃ¾gen & Co.	0
Kremlin Collectables, Co.	0
Raanan Stores, Inc	0
Iberia Gift Imports, Corp.	2
Motor Mint Distributors Inc.	3
Signal Collectibles Ltd.	2
Double Decker Gift Stores, Ltd	2
Diecast Collectables	2
Kelly's Gift Shop	4

122 rows in set (0.00 sec)

13. Write a query to find the maximum credit limit among all customers.

```
mysql> select max(creditLimit) as maxCreditLimit from customers;
+----------------+
| maxCreditLimit |
+----------------+
|      227600.00 |
+----------------+
1 row in set (0.00 sec)
```

14. Write a query to count the number of offices in each country.

```
mysql> select country, count(officeCode) as totalOffices
    -> from offices group by country;
+-----------+--------------+
| country   | totalOffices |
+-----------+--------------+
| USA       |            3 |
| France    |            1 |
| Japan     |            1 |
| Australia |            1 |
| UK        |            1 |
+-----------+--------------+
5 rows in set (0.00 sec)
```

15. Write a query to calculate the average payment amount for each customer.

```
mysql> select customerNumber, avg(amount) as avgPayment
    -> from payments group by customerNumber;
+----------------+--------------+
| customerNumber | avgPayment   |
+----------------+--------------+
|            103 |  7438.120000 |
|            112 | 26726.993333 |
|            114 | 45146.267500 |
|            119 | 38983.226667 |
|            121 | 26056.197500 |
|            124 | 64909.804444 |
|            128 | 18984.440000 |
|            129 | 22236.853333 |
|            131 | 35879.980000 |
|            141 | 55056.844615 |
|            144 | 21840.325000 |
|            145 | 26861.625000 |
|            146 | 43435.116667 |
|            148 | 39062.757500 |
|            151 | 44478.487500 |
|            157 | 49254.625000 |
|            161 | 26136.305000 |
|            166 | 35140.190000 |
|            167 | 48781.235000 |
|            171 | 30890.850000 |
|            172 | 28851.173333 |
|            173 | 16099.345000 |
|            175 | 31808.210000 |
|            177 | 31180.610000 |
|            181 | 24165.880000 |
|            186 | 31848.820000 |
|            187 | 49470.030000 |
|            189 | 24949.135000 |
|            198 |  7184.753333 |
|            201 | 30583.590000 |
|            202 | 35061.095000 |
|            204 | 27788.630000 |
|            205 | 31267.766667 |
|            209 | 25286.440000 |
|            211 | 45480.790000 |
|            216 | 22840.156667 |
```

324	26852.243333
328	19140.755000
333	18396.720000
334	34632.246667
339	28969.670000
344	23375.570000
347	20753.095000
350	23849.176667
353	31745.797500
357	28331.190000
362	16766.735000
363	38816.430000
379	24511.216667
381	7304.295000
382	28353.333333
385	29156.100000
386	45071.655000
398	26387.182500
406	28812.323333
412	33352.470000
415	31310.090000
424	23071.443333
447	16655.926667
448	38388.220000
450	59551.380000
452	17019.996667
455	35189.325000
456	14615.215000
458	37480.030000
462	29542.496667
471	22460.380000
473	12679.160000
475	21874.360000
484	25493.925000
486	25908.863333
487	21285.185000
489	14793.075000
495	32770.870000
496	38165.730000

98 rows in set (0.00 sec)

16. write a query to determine the number of products in each product line.

```
mysql> select productLine, count(productCode) as totalProducts
    -> from products group by productline;
+------------------+---------------+
| productLine      | totalProducts |
+------------------+---------------+
| Classic Cars     |            38 |
| Motorcycles      |            13 |
| Planes           |            12 |
| Ships            |             9 |
| Trains           |             3 |
| Trucks and Buses |            11 |
| Vintage Cars     |            24 |
+------------------+---------------+
7 rows in set (0.00 sec)
```

17. write a query to count the number of customers in each state.

```
mysql> select state, count(customerNumber) as totalcustomers
    -> from customers group by state;
+---------------+----------------+
| state         | totalcustomers |
+---------------+----------------+
| NULL          |             73 |
| NV            |              1 |
| Victoria      |              2 |
| CA            |             11 |
| NY            |              6 |
| PA            |              3 |
| CT            |              4 |
| MA            |              9 |
| Osaka         |              1 |
| BC            |              2 |
| QuÃ©bec       |              1 |
| Isle of Wight |              1 |
| NSW           |              2 |
| NJ            |              1 |
| Queensland    |              1 |
| Co. Cork      |              1 |
| Pretoria      |              1 |
| NH            |              1 |
| Tokyo         |              1 |
+---------------+----------------+
19 rows in set (0.00 sec)
```

18. Write a query to find the minimum payment amount among all customers.

```
mysql> select min(amount) as minPayment from payments;
+------------+
| minPayment |
+------------+
|     615.45 |
+------------+
1 row in set (0.00 sec)
```

19. Write a query to calculate the average sales revenue per order.

```
mysql> select orderNumber, avg( quantityOrdered * priceEach ) as avgSalesRevenue
    -> from orderdetails group by orderNumber;
+-------------+-----------------+
| orderNumber | avgSalesRevenue |
+-------------+-----------------+
|       10100 |     2555.957500 |
|       10101 |     2637.252500 |
|       10102 |     2747.390000 |
|       10103 |     3138.684375 |
|       10104 |     3092.784615 |
|       10105 |     3597.280667 |
|       10106 |     2897.322778 |
|       10107 |     2786.577500 |
|       10108 |     3187.576250 |
|       10109 |     4305.523333 |
|       10110 |     3026.605625 |
|       10111 |     2756.308333 |
|       10112 |     3837.470000 |
|       10113 |     2761.075000 |
|       10114 |     3338.314000 |
|       10115 |     4333.196000 |
|       10116 |     1627.560000 |
|       10117 |     3698.345833 |
|       10118 |     3101.400000 |
|       10119 |     2559.023571 |
|       10120 |     3057.602000 |
|       10121 |     3340.094000 |
|       10122 |     2989.685882 |
|       10123 |     3642.860000 |
|       10124 |     2510.921538 |
|       10125 |     3782.540000 |
|       10126 |     3360.701176 |
|       10127 |     3922.756667 |
|       10128 |     3471.247500 |
|       10129 |     3269.904444 |
|       10130 |     3018.480000 |
|       10131 |     2129.036250 |
|       10132 |     2880.000000 |
|       10133 |     2795.755000 |
|       10134 |     3345.638571 |
```

10387	3516.040000
10388	3786.721250
10389	3495.817500
10390	3493.906250
10391	2984.852000
10392	2935.706667
10393	3053.938182
10394	2586.105714
10395	4482.022500
10396	3461.942500
10397	2486.464000
10398	2592.052222
10399	3781.718750
10400	3528.371111
10401	3627.086667
10402	4063.616667
10403	4139.882222
10404	5178.351250
10405	7031.550000
10406	7212.873333
10407	4352.462500
10408	615.450000
10409	1163.090000
10410	4049.148889
10411	3230.042222
10412	4263.225455
10413	4750.130000
10414	3629.060714
10415	2189.052000
10416	2525.875714
10417	4762.483333
10418	2625.271111
10419	3744.290714
10420	3250.116154
10421	3819.550000
10422	2924.720000
10423	1719.546000
10424	4885.050000
10425	3201.803077

326 rows in set (0.01 sec)

20. Write a query to determine the total quantity ordered for each product line.

```
mysql> select ps.productLine, sum(ods.quantityOrdered)
as totalQuantityOrdered
    -> from orderdetails ods inner join products ps
    -> on ps.productCode = ods.productCode group by ps.
productLine;
+------------------+-----------------------+
| productLine      | totalQuantityOrdered  |
+------------------+-----------------------+
| Classic Cars     |                 35582 |
| Motorcycles      |                 12778 |
| Planes           |                 11872 |
| Ships            |                  8532 |
| Trains           |                  2818 |
| Trucks and Buses |                 11001 |
| Vintage Cars     |                 22933 |
+------------------+-----------------------+
7 rows in set (0.01 sec)
```