Task 3: SQL for Data Analysis

❖ Setting-up the Environment in MySql

1. Database and Table creation as per data.csv structure

```
mysql> create database customer_db;
Query OK, 1 row affected (0.00 sec)

mysql> use customer_db;
Database changed
mysql> create table data(
    -> email VARCHAR(100),
    -> address VARCHAR(100),
    -> avatar VARCHAR(20),
    -> timeonapp INT,
    -> timeonwebsite INT,
    -> lengthofmembership INT,
    -> yearlyamountspent INT);
Query OK, 0 rows affected (0.92 sec)
```

2. Enable Local File Loading and exit

```
mysql> set global local_infile = 1;
Query OK, 0 rows affected (0.17 sec)

mysql> Ctrl-C -- exit!
Bye
```

3. Re-login with local file option and load the data

```
C:\Users\Admin>mysql --local-infile=1 -u root -p
```

```
mysql> use customer_db;
Database changed
```

```
mysql> load data local infile 'D:/Elevate_Labs/Task3/data.csv'
    -> into table data
    -> fields terminated by ','
    -> enclosed by '"'
    -> lines terminated by '\n'
    -> ignore 1 rows;
Query OK, 500 rows affected (0.24 sec)
Records: 500  Deleted: 0  Skipped: 0  Warnings: 0
```

❖ Start Analyzing the Data

1.Count of rows and maxing yearly amount spent

```
mysql> select count(*) from data;
+----------+
| count(*) |
+----------+
|      500 |
+----------+
1 row in set (0.63 sec)

mysql> select MAX(yearlyamountspent) from data;
+------------------------+
| MAX(yearlyamountspent) |
+------------------------+
|                 766.00 |
+------------------------+
1 row in set (0.58 sec)
```

2. Top 10 Customers by Yearly Spending

```
mysql> select email, yearlyamountspent
    -> from data
    -> order by yearlyamountspent desc
    -> limit 10;
+---------------------------------+-------------------+
| email                           | yearlyamountspent |
+---------------------------------+-------------------+
| kyang@diaz.org                  |            766.00 |
| asilva@yahoo.com                |            744.00 |
| william82@gmail.com             |            726.00 |
| jeffrey54@mcdonald-williams.com |            712.00 |
| rhonda01@gmail.com              |            709.00 |
| youngbarbara@yahoo.com          |            701.00 |
| randyrobinson@hotmail.com       |            690.00 |
| susanibarra@yahoo.com           |            689.00 |
| alicia85@lee.com                |            684.00 |
| waltonkaren@gmail.com           |            670.00 |
+---------------------------------+-------------------+
10 rows in set (0.17 sec)
```

3. Average Time on App vs Website

```
mysql> select
    -> avg(timeonapp) as avg_app_time,
    -> avg(timeonwebsite) as avg_website_time
    -> from data;
+--------------+------------------+
| avg_app_time | avg_website_time |
+--------------+------------------+
|      12.0640 |          37.0740 |
+--------------+------------------+
1 row in set (0.17 sec)
```

4. Relationship Between Membership Length and Spending

```
mysql> select lengthofmembership, avg(yearlyamountspent) as avg_spending
    -> from data
    -> group by lengthofmembership
    -> order by lengthofmembership;
+-------------------+-------------+
| lengthofmembership | avg_spending |
+-------------------+-------------+
|                 0 |     314.0000 |
|                 1 |     340.6000 |
|                 2 |     414.4286 |
|                 3 |     470.3905 |
|                 4 |     524.6012 |
|                 5 |     578.3636 |
|                 6 |     687.6250 |
|                 7 |     744.0000 |
+-------------------+-------------+
8 rows in set (0.02 sec)
```

5. Who Spends More: App Users or Website Users?

```
mysql> select
    -> case
    -> when timeonapp > timeonwebsite then 'App preferred'
    -> else 'Website preferred'
    -> end as user_type,
    -> avg(yearlyamountspent) as avg_spent
    -> from data
    -> group by user_type;
+-------------------+-----------+
| user_type         | avg_spent |
+-------------------+-----------+
| Website preferred |  499.3240 |
+-------------------+-----------+
1 row in set (0.47 sec)
```

6. Create a View for Ongoing Analysis

```
mysql> create view high_spender as
    -> select email, yearlyamountspent
    -> from data
    -> where yearlyamountspent > 800;
Query OK, 0 rows affected (3.33 sec)

mysql> select * from high_spender
    -> ;
Empty set (1.07 sec)
```

7. Use a Subquery: Customers Above Average Spending

```
mysql> select yearlyamountspent
    -> from data
    -> where yearlyamountspent > (
    -> select avg(yearlyamountspent) from data
    -> );
+--------------------+
| yearlyamountspent |
+--------------------+
|                588 |
|                582 |
|                599 |
|                637 |
|                522 |
|                550 |
|                570 |
|                522 |
|                573 |
```

8. Optimize Queries with Indexes

```
mysql> create index idx_email on data(email);
Query OK, 0 rows affected (0.19 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> create index idx_spending on data(yearlyamountspent);
Query OK, 0 rows affected (0.12 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

9. Show full table

```
mysql> select *
    -> from data
    -> limit 10;
+------------------------------+-----------------------------------------------------------+-----------------+-----------+--------------+-----------------
---+--------------------+
| email                        | address                                                   | avatar          | timeonapp | timeonwebsite | lengthofmembersh
ip | yearlyamountspent |
+------------------------------+-----------------------------------------------------------+-----------------+-----------+--------------+-----------------
---+--------------------+
| mstephenson@fernandez.com    | 835 Frank TunnelWrightmouth, MI 82180-9605                | Violet          |        13 |           40 |
 4 |                588 |
| hduke@hotmail.com            | 4547 Archer CommonDiazchester, CA 06566-8576              | DarkGreen       |        11 |           37 |
 3 |                392 |
| pallen@yahoo.com             | 24645 Valerie Unions Suite 582Cobbborough, DC 99414-7564  | Bisque          |        11 |           37 |
 4 |                488 |
| riverarebecca@gmail.com      | 1414 David ThroughwayPort Jason, OH 22070-1220            | SaddleBrown     |        14 |           37 |
 3 |                582 |
| mstephens@davidson-herman.com | 14023 Rodriguez PassagePort Jacobville, PR 37242-1057    | MediumAquaMarine |        13 |           38 |
 4 |                599 |
| alvareznancy@lucas.biz       | 645 Martha Park Apt. 611Jeffreychester, MN 67218-7250     | FloralWhite     |        12 |           34 |
 5 |                637 |
| katherine20@yahoo.com        | 68388 Reyes Lights Suite 692Josephbury, WV 92213-0247     | DarkSlateBlue   |        11 |           37 |
 5 |                522 |
| awatkins@yahoo.com           | Unit 6538 Box 8980DPO AP 09026-4941                       | Aqua            |        12 |           37 |
 4 |                550 |
| vchurch@walter-martinez.com  | 860 Lee KeyWest Debra, SD 97450-0495                      | Salmon          |        13 |           38 |
```

10. Average time on app based on avatar

```
mysql> select avatar, avg(timeonapp) as avg_time_on_app
    -> from data
    -> group by avatar;
+----------------------+------------------+
| avatar               | avg_time_on_app  |
+----------------------+------------------+
| AliceBlue            |          12.0000 |
| AntiqueWhite         |          11.4000 |
| Aqua                 |          12.1667 |
| Aquamarine           |          12.0000 |
| Azure                |          12.0000 |
| Beige                |          12.0000 |
| Bisque               |          12.0000 |
| Black                |          13.0000 |
| BlanchedAlmond       |          11.8000 |
| Blue                 |          11.7500 |
```

11. Find the Most Engaged Users (High App + Website Time)

```
mysql> select email, (timeonapp + timeonwebsite) as total_time_spent, yearlyamountspent
    -> from data
    -> order by total_time_spent desc
    -> limit 5;
+--------------------------+------------------+-------------------+
| email                    | total_time_spent | yearlyamountspent |
+--------------------------+------------------+-------------------+
| mstephenson@fernandez.com |               53 |               588 |
| pkline@hotmail.com       |               53 |               525 |
| alejandro75@hotmail.com  |               53 |               452 |
| randall85@williams.com   |               53 |               582 |
| russellbaldwin@ferrell.info |               53 |               660 |
+--------------------------+------------------+-------------------+
5 rows in set (0.07 sec)
```

12. Check If Longer Membership Correlates with More App Time

```
mysql> select lengthofmembership, avg(timeonapp) as avg_app_time
    -> from data
    -> group by lengthofmembership
    -> order by lengthofmembership;
+--------------------+--------------+
| lengthofmembership | avg_app_time |
+--------------------+--------------+
|                  0 |      13.0000 |
|                  1 |      12.1333 |
|                  2 |      12.0536 |
|                  3 |      12.0473 |
|                  4 |      12.0636 |
|                  5 |      12.0260 |
|                  6 |      12.6250 |
|                  7 |      12.0000 |
+--------------------+--------------+
8 rows in set (0.00 sec)
```

13. Find Users Spending a Lot Despite Low Time on App

```
mysql> select email, timeonapp, yearlyamountspent
    -> from data
    -> where timeonapp < 20 and yearlyamountspent > 800
    -> group by yearlyamountspent desc;
Empty set (0.08 sec)
```

14. Segment Users into Spending Buckets

```
mysql> select
    -> case
    -> when yearlyamountspent >= 500 then 'high spenders'
    -> when yearlyamountspent between 300 and 499 then 'Medium Spenders'
    -> else 'low spenders'
    -> end as spending_category,
    -> count(*) as user_count
    -> from data
    -> group by spending_category;
+-------------------+------------+
| spending_category | user_count |
+-------------------+------------+
| high spenders     |        249 |
| low spenders      |          5 |
| Medium Spenders   |        246 |
+-------------------+------------+
3 rows in set (0.00 sec)
```

15. Segment Users by Membership Duration

```
mysql> select
    -> case
    -> when lengthofmembership >= 5 then 'loyal Members'
    -> when lengthofmembership between 2 and 4 then 'Regular Members'
    -> else 'New Members'
    -> end as membership_segment,
    -> count(*) as user_count
    -> from data
    -> group by membership_segment;
+--------------------+------------+
| membership_segment | user_count |
+--------------------+------------+
| loyal Members      |         86 |
| New Members        |         16 |
| Regular Members    |        398 |
+--------------------+------------+
3 rows in set (0.00 sec)
```