SQL STATEMENTS

```
SET GLOBAL local_infile=1;

DROP DATABASE IF EXISTS dash;
CREATE DATABASE IF NOT EXISTS dash;
USE dash;

SHOW TABLES;

-- Create User table if it does not exist
CREATE TABLE IF NOT EXISTS Users (
    user_id INT PRIMARY KEY,
    name VARCHAR(255),
    subscription_type VARCHAR(255),
    date_created DATE
);

-- Create Subscription table if it does not exist
CREATE TABLE IF NOT EXISTS Subscriptions (
    sub_id INT PRIMARY KEY,
    user_id INT,
    payment_interval VARCHAR(255),
    payment_cost INT,
    purchase_date DATE NULL,
    end_date DATE NULL,
    status VARCHAR(255),
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Create Activities table if it does not exist
CREATE TABLE IF NOT EXISTS Activities (
    actv_id INT PRIMARY KEY,
    user_id INT,
    actv_name VARCHAR(255),
    actv_type VARCHAR(255),
    avg_speed INT,
    duration INT,
    heartrate INT,
    upload_date DATE,  -- Added missing comma here
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

load data local infile "data/users.csv" into table Users
            fields terminated by ','
    lines terminated by '\n'
    ignore 1 lines
    ;

load data local infile "data/subscriptions.csv" into table Subscriptions
            fields terminated by ','
```

```
   lines terminated by '\n'
   ignore 1 lines
   SET end_date = NULLIF(@end_date, '')
   ;

load data local infile "data/activities.csv" into table Activities
            fields terminated by ','
   lines terminated by '\n'
   ignore 1 lines
   ;

SHOW TABLES;

SELECT 'running describe Users' AS '';
DESCRIBE Users;

SELECT 'running describe Subscriptions' AS '';
DESCRIBE Subscriptions;

SELECT 'running describe Activities' AS '';
DESCRIBE Activities;

SELECT 'running select * from Users LIMIT 10' AS '';
SELECT * FROM Users LIMIT 10;

SELECT ' running SELECT COUNT(*) FROM Users;' AS '';
SELECT COUNT(*) FROM Users;

SELECT 'running select * from Subscriptions LIMIT 10' AS '';
SELECT * FROM Subscriptions LIMIT 10;

SELECT 'running SELECT COUNT(*) FROM Subscriptions;' AS '';
SELECT COUNT(*) FROM Subscriptions;

SELECT 'running select * from Activities LIMIT 10' AS '';
SELECT * FROM Activities LIMIT 10;

SELECT 'running SELECT COUNT(*) FROM Activities;' AS '';
SELECT COUNT(*) FROM Activities;

SELECT 'running select * FROM Users U, Subscriptions S WHERE U.user_id = S.user_id LIMIT 10' AS '';
SELECT * FROM Users U, Subscriptions S WHERE U.user_id = S.user_id LIMIT 10;

SELECT 'running select * from Users U, Activities A where U.user_id = A.user_id LIMTI 10' AS '';
SELECT * FROM Users U, Activities A WHERE U.user_id = A.user_id LIMIT 10;
-------------------------------------------------------------------   ------------------------------------------------------------------------   ------------
-----------------------------------------------------------   ------------------------------------------------------------------
```

TERMINAL OUTPUT

```
Last login: Fri Feb  2 20:08:27 on ttys018
You have new mail.
(base) jairusmartinez@Jairuss-MacBook-Pro week4 % mysql --local-infile=1 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source martinez_jairus_04.sql
Query OK, 0 rows affected (0.01 sec)

Query OK, 3 rows affected (0.05 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Empty set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 10000 rows affected (0.06 sec)
Records: 10000  Deleted: 0  Skipped: 0  Warnings: 0

Query OK, 3546 rows affected, 3546 warnings (0.04 sec)
Records: 3546  Deleted: 0  Skipped: 0  Warnings: 3546

Query OK, 498884 rows affected (4.68 sec)
Records: 498884  Deleted: 0  Skipped: 0  Warnings: 0

+----------------+
| Tables_in_dash |
+----------------+
| Activities     |
| Subscriptions  |
| Users          |
+----------------+
3 rows in set (0.00 sec)

+------------------------+
|                        |
+------------------------+
| running describe Users |
+------------------------+
1 row in set (0.00 sec)

+------------------+--------------+------+-----+---------+-------+
```

```
| Field             | Type         | Null | Key | Default | Extra |
+-------------------+--------------+------+-----+---------+-------+
| user_id           | int          | NO   | PRI | NULL    |       |
| name              | varchar(255) | YES  |     | NULL    |       |
| subscription_type | varchar(255) | YES  |     | NULL    |       |
| date_created      | date         | YES  |     | NULL    |       |
+-------------------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)

+--------------------------------+
|                                |
+--------------------------------+
| running describe Subscriptions |
+--------------------------------+
1 row in set (0.00 sec)

+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| sub_id           | int          | NO   | PRI | NULL    |       |
| user_id          | int          | YES  | MUL | NULL    |       |
| payment_interval | varchar(255) | YES  |     | NULL    |       |
| payment_cost     | int          | YES  |     | NULL    |       |
| purchase_date    | date         | YES  |     | NULL    |       |
| end_date         | date         | YES  |     | NULL    |       |
| status           | varchar(255) | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

+----------------------------+
|                            |
+----------------------------+
| running describe Activities |
```

```
+----------------------------+
1 row in set (0.00 sec)

+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| actv_id    | int          | NO   | PRI | NULL    |       |
| user_id    | int          | YES  | MUL | NULL    |       |
| actv_name  | varchar(255) | YES  |     | NULL    |       |
| actv_type  | varchar(255) | YES  |     | NULL    |       |
| avg_speed  | int          | YES  |     | NULL    |       |
| duration   | int          | YES  |     | NULL    |       |
| heartrate  | int          | YES  |     | NULL    |       |
| upload_date | date        | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

+-------------------------------------+
|                                     |
+-------------------------------------+
| running select * from Users LIMIT 10 |
+-------------------------------------+
1 row in set (0.00 sec)

+---------+--------+-------------------+--------------+
| user_id | name   | subscription_type | date_created |
+---------+--------+-------------------+--------------+
|       1 | user1  | free              | 2020-06-03   |
|       2 | user2  | premium           | 2020-11-29   |
|       3 | user3  | premium           | 2019-09-26   |
|       4 | user4  | premium           | 2019-09-02   |
|       5 | user5  | premium           | 2021-01-02   |
|       6 | user6  | free              | 2020-11-06   |
```

```
|       7 | user7  | free             | 2018-07-18   |
|       8 | user8  | free             | 2017-04-08   |
|       9 | user9  | free             | 2020-10-21   |
|      10 | user10 | free             | 2018-02-23   |
+---------+--------+------------------+--------------+
10 rows in set (0.00 sec)

+-------------------------------------+
|                                     |
+-------------------------------------+
|  running SELECT COUNT(*) FROM Users; |
+-------------------------------------+
1 row in set (0.00 sec)

+----------+
| COUNT(*) |
+----------+
|    10000 |
+----------+
1 row in set (0.00 sec)

+---------------------------------------------+
|                                             |
+---------------------------------------------+
| running select * from Subscriptions LIMIT 10 |
+---------------------------------------------+
1 row in set (0.00 sec)

+--------+---------+------------------+--------------+---------------+----------+--------+
| sub_id | user_id | payment_interval | payment_cost | purchase_date | end_date | status |
+--------+---------+------------------+--------------+---------------+----------+--------+
|     52 |       2 | monthly          |           11 | 2021-06-06    | NULL     | active |
|     53 |       3 | monthly          |           11 | 2020-09-20    | NULL     | active |
```

```
|    54 |      4 | semi-annual     |          65 | 2020-02-28   | NULL     | active |
|    55 |      5 | semi-annual     |          65 | 2021-12-16   | NULL     | active |
|   511 |     11 | monthly         |          11 | 2018-12-27   | NULL     | active |
|   516 |     16 | monthly         |          11 | 2020-11-15   | NULL     | active |
|   518 |     18 | monthly         |          11 | 2019-11-27   | NULL     | active |
|   519 |     19 | monthly         |          11 | 2019-04-15   | NULL     | active |
|   521 |     21 | monthly         |          11 | 2018-11-30   | NULL     | active |
|   522 |     22 | monthly         |          11 | 2019-04-27   | NULL     | active |
+-------+--------+-----------------+--------------+--------------+----------+--------+
10 rows in set (0.00 sec)

+-----------------------------------------+
|                                         |
+-----------------------------------------+
| running SELECT COUNT(*) FROM Subscriptions; |
+-----------------------------------------+
1 row in set (0.00 sec)

+----------+
| COUNT(*) |
+----------+
|     3546 |
+----------+
1 row in set (0.00 sec)

+-----------------------------------------+
|                                         |
+-----------------------------------------+
| running select * from Activities LIMIT 10 |
+-----------------------------------------+
1 row in set (0.00 sec)

+---------+---------+-------------+--------------+-----------+----------+-----------+-------------+
```

| actv_id | user_id | actv_name | actv_type | avg_speed | duration | heartrate | upload_date |
+---------+---------+-------------+--------------+-----------+----------+-----------+-------------+
100	1	Activity_84	RockClimbing	0	32	144	2020-11-15
101	1	Activity_47	Swimming	2	15	145	2021-12-31
102	1	Activity_79	Gym	0	75	145	2022-08-24
103	1	Activity_29	Gym	0	27	148	2020-05-14
200	2	Activity_51	Cycling	10	192	115	2022-02-26
201	2	Activity_94	Hiking	2	18	144	2022-05-20
202	2	Activity_43	RockClimbing	0	109	127	2021-03-30
203	2	Activity_21	RockClimbing	0	28	139	2021-07-31
204	2	Activity_93	RockClimbing	0	92	158	2022-04-08
205	2	Activity_86	Gym	0	80	146	2023-12-16
+---------+---------+-------------+--------------+-----------+----------+-----------+-------------+
10 rows in set (0.00 sec)

+--+
| |
+--+
| running SELECT COUNT(*) FROM Activities; |
+--+
1 row in set (0.00 sec)

+----------+
| COUNT(*) |
+----------+
| 498884 |
+----------+
1 row in set (0.01 sec)

+---+
| |
+---+
| running select * FROM Users U, Subscriptions S WHERE U.user_id = S.user_id LIMIT 10 |

```
+--------------------------------------------------------------------------------+
1 row in set (0.00 sec)

+---------+--------+------------------+-------------+--------+---------+------------------+-------------+--------------+---------
-+--------+
| user_id | name   | subscription_type | date_created | sub_id | user_id | payment_interval | payment_cost |
purchase_date | end_date | status |
+---------+--------+------------------+-------------+--------+---------+------------------+-------------+--------------+---------
-+--------+
|       2 | user2  | premium          | 2020-11-29   |     52 |       2 | monthly          |           11 | 2021-06-06   |
NULL     | active |
|       3 | user3  | premium          | 2019-09-26   |     53 |       3 | monthly          |           11 | 2020-09-20   |
NULL     | active |
|       4 | user4  | premium          | 2019-09-02   |     54 |       4 | semi-annual      |           65 | 2020-02-28   |
NULL     | active |
|       5 | user5  | premium          | 2021-01-02   |     55 |       5 | semi-annual      |           65 | 2021-12-16   |
NULL     | active |
|      11 | user11 | premium          | 2018-01-30   |    511 |      11 | monthly          |           11 | 2018-12-27   |
NULL     | active |
|      16 | user16 | premium          | 2020-05-23   |    516 |      16 | monthly          |           11 | 2020-11-15   |
NULL     | active |
|      18 | user18 | premium          | 2019-11-16   |    518 |      18 | monthly          |           11 | 2019-11-27   |
NULL     | active |
|      19 | user19 | premium          | 2019-03-12   |    519 |      19 | monthly          |           11 | 2019-04-15   |
NULL     | active |
|      21 | user21 | premium          | 2018-11-26   |    521 |      21 | monthly          |           11 | 2018-11-30   |
NULL     | active |
|      22 | user22 | premium          | 2018-12-07   |    522 |      22 | monthly          |           11 | 2019-04-27   |
NULL     | active |
+---------+--------+------------------+-------------+--------+---------+------------------+-------------+--------------+---------
-+--------+
10 rows in set (0.00 sec)
```

```
+----------------------------------------------------------------------------+
|                                                                            |
+----------------------------------------------------------------------------+
| running select * from Users U, Activities A where U.user_id = A.user_id LIMTI 10 |
+----------------------------------------------------------------------------+
1 row in set (0.00 sec)

+---------+-------+------------------+-------------+---------+---------+------------+-------------+-----------+----------+-------------+
| user_id | name  | subscription_type | date_created | actv_id | user_id | actv_name   | actv_type    | avg_speed | duration | heartrate | upload_date |
+---------+-------+------------------+-------------+---------+---------+------------+-------------+-----------+----------+-------------+
|       1 | user1 | free             | 2020-06-03   |     100 |       1 | Activity_84 | RockClimbing |         0 |       32 |       144 | 2020-11-15  |
|       1 | user1 | free             | 2020-06-03   |     101 |       1 | Activity_47 | Swimming     |         2 |       15 |       145 | 2021-12-31  |
|       1 | user1 | free             | 2020-06-03   |     102 |       1 | Activity_79 | Gym          |         0 |       75 |       145 | 2022-08-24  |
|       1 | user1 | free             | 2020-06-03   |     103 |       1 | Activity_29 | Gym          |         0 |       27 |       148 | 2020-05-14  |
|       2 | user2 | premium          | 2020-11-29   |     200 |       2 | Activity_51 | Cycling      |        10 |      192 |       115 | 2022-02-26  |
|       2 | user2 | premium          | 2020-11-29   |     201 |       2 | Activity_94 | Hiking       |         2 |       18 |       144 | 2022-05-20  |
|       2 | user2 | premium          | 2020-11-29   |     202 |       2 | Activity_43 | RockClimbing |         0 |      109 |       127 | 2021-03-30  |
|       2 | user2 | premium          | 2020-11-29   |     203 |       2 | Activity_21 | RockClimbing |         0 |       28 |       139 | 2021-07-31  |
|       2 | user2 | premium          | 2020-11-29   |     204 |       2 | Activity_93 | RockClimbing |         0 |       92 |       158 | 2022-04-08  |
|       2 | user2 | premium          | 2020-11-29   |     205 |       2 | Activity_86 | Gym          |         0 |       80 |       146 | 2023-12-16  |
```

```
+---------+-------+-------------------+--------------+---------+---------+-------------+--------------+-----------+----------+------
-----+-------------+
```

10 rows in set (0.00 sec)

mysql>

PYTHON CODE

```
"""Module to create synthetic data for COMP 3421 PDA"""

from random import randint, choices, choice
from datetime import datetime, timedelta
import logging
import pandas as pd

# basic logging and formatting
logging.basicConfig(format='fakeData - %(asctime)s - %(levelname)s - %(message)s', level=logging.INFO)

def timing_decorator(func):
    """Times any function that it's wrapped around"""
    def wrapper(*args, **kwargs):
        start_time = datetime.now()
        result = func(*args, **kwargs)
        end_time = datetime.now()

        elapsed_time = end_time - start_time
        logging.info(f"{func.__name__}(): {elapsed_time.total_seconds():.2f}s.")
        return result

    return wrapper

@timing_decorator
def create_users(num_records: int, year_start: int, year_end:int , percent_free: float):
    """
    Creates synthetic users for Users table within PDA.

    SQL Schema:
        CREATE TABLE IF NOT EXISTS Users (
        user_id INT PRIMARY KEY,
        name VARCHAR(255),
        subscription_type VARCHAR(255),
        date_created DATE
        );

    Params:
```

```
    num_records: how many unique users
    year_start: year to start date for range of creation
    year_end: year to end date for range of creation
    percent_free: percent of users who hava a 'free' subscription
  Returns:
    df_users : pd.DataFrame
  """
  users_data = []

  # generate and insert num_records users into the DataFrame
  for user_id in range(1, num_records + 1):
    name = f'user{user_id}'

    # determine subscription type based on the specified percentages
    subscription_type = 'free' if randint(1, 100) <= percent_free else 'premium'

    # generate a random date between 2017 and 2024
    date_created = datetime(year_start, 1, 1) + timedelta(days=randint(0, (year_end - year_start) * 365))

    # append the generated data to a temporary DataFrame
    users_entry = {
      'user_id': user_id,
      'name': name,
      'subscription_type': subscription_type,
      'date_created': date_created.strftime('%Y-%m-%d')
    }

    users_data.append(users_entry)

  df_users = pd.DataFrame(users_data)
  return df_users

@timing_decorator
def create_subsciptions(subscribed_users: pd.DataFrame):
  """
  Creates synthetic subscriptions data for Subscriptions table wihtin PDA.

  SQL Schema:
    CREATE TABLE IF NOT EXISTS Subscriptions (
    sub_id INT PRIMARY KEY,
    user_id INT,
    payment_interval VARCHAR(255),
    payment_cost INT,
    purchase_date DATE NULL,
    end_date DATE NULL,
    status VARCHAR(255),ßß
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
    );

  Params:
    subscribed_users: dataframe contaiing subscribed users only
  Returns:
    df_subscriptions: pd.Dataframe
```

```
    """
    subscriptions_data = []

    for i in subscribed_users.index:
        user_id = subscribed_users.loc[i, 'user_id']
        sub_id = int(f'5{user_id}')
        payment_interval = choices(['monthly', 'semi-annual', 'annual'],
                           weights=[.70, .10, .20])[0]

        if payment_interval == 'annual':
            payment_cost = 120
        elif payment_interval == 'semi-annual':
            payment_cost = 65
        else:
            payment_cost = 11

        purchase_date = pd.to_datetime(subscribed_users.loc[i, 'date_created']) + timedelta(days=randint(0, 365))

        status = 'active'
        end_date = pd.NA

        # append the generated data to a temporary DataFrame
        subscriptions_entry = {
            'sub_id': sub_id,
            'user_id': user_id,
            'payment_interval': payment_interval,
            'payment_cost': payment_cost,
            'purchase_date': purchase_date.strftime('%Y-%m-%d'),
            'end_date': end_date,
            'status': status
        }

        subscriptions_data.append(subscriptions_entry)

        df_subscribers = pd.DataFrame(subscriptions_data)

    return df_subscribers

@timing_decorator
def create_activities(df_users: pd.DataFrame):
    """
    Create Activities table.

    CREATE TABLE IF NOT EXISTS Activities (
    actv_id INT PRIMARY KEY,
    user_id INT,
    actv_name VARCHAR(255),
    actv_type VARCHAR(255),
    avg_speed INT,
    duration INT,
    heartrate INT,
    upload_date DATE,  -- Added missing comma here
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
```

```
);

Params:
   df_users: dataframe containing all user data
Returns:
   df_activities: pd.DataFrame
"""
activities_data = []

for user_id in df_users['user_id']:
   num_activities = randint(1, 100)

   for i in range(num_activities):
      actv_name = f'Activity_{randint(1, 100)}'
      actv_id = int(f'{user_id}0{i}')
      actv_type = choice(['Running', 'Cycling', 'Hiking', 'Gym', 'Swimming', 'RockClimbing'])

      if actv_type == 'Running':
         avg_speed = randint(1, 7)
      elif actv_type == 'Cycling':
         avg_speed = randint(10, 21)
      elif actv_type == 'RockClimbing' or actv_type == 'Gym':
         avg_speed = 0
      else:
         avg_speed = randint(1, 3)

      if actv_type == 'Running' or actv_type == 'RockClimbing' or actv_type == 'Gym':
         duration = randint(15, 120)
      elif actv_type == 'Cycling':
         duration = randint(60, 360)
      else:
         duration = randint(15, 45)

      heartrate = randint(100, 160)


      upload_date = pd.to_datetime(df_users.loc[i, 'date_created']) + timedelta(days=randint(0, 4 * 365))

      activity_entry = {
         'actv_id': actv_id,
         'user_id': user_id,
         'actv_name': actv_name,
         'actv_type': actv_type,
         'avg_speed': avg_speed,
         'duration': duration,
         'heartrate': heartrate,
         'upload_date': upload_date.strftime('%Y-%m-%d')
      }

      activities_data.append(activity_entry)

df_activities = pd.DataFrame(activities_data)
return df_activities
```

```
@timing_decorator
def main(num_users: int, percent_free: int):
    """
    Exports 3 csv files corresponding to df_users, df_subscriptions,
    and df_activities.
    Params:
        num_users: number of users which dictates Subscribers/Activities record lengths
        percent_free: percent of users with a "free" account
    Returns:
        None
    """
    try:
        logging.info('Creating users table...')
        df_users = create_users(num_users, 2017, 2022, percent_free)
        subscribed_users = df_users.loc[df_users['subscription_type'] == 'premium', ['user_id', 'date_created']]

        logging.info('Creating Subscriptions table...')
        df_subscribers = create_subsciptions(subscribed_users)

        logging.info('Creatng activities table...')
        df_activities = create_activities(df_users.loc[:, ['user_id', 'date_created']])

        logging.info('Exporting data...')
        df_users.to_csv('data/users.csv', index=False)
        df_subscribers.to_csv('data/subscriptions.csv', index=False)
        df_activities.to_csv('data/activities.csv', index=False)

        logging.info('Export complete')
    except Exception as e:
        logging.error(f'Error: {e}')


if __name__ == '__main__':
    main(num_users=10000, percent_free=65)
```
