Big data Management Assignment

Jeyadev L

G23AI2071

Connection to Redis

Assignment-5-JD.py > main

```python
import redis as rd
import os
import logging
import pandas as pd
import re

# Configure logging
logging.basicConfig(
    level=logging.DEBUG,
    format='%(asctime)s - %(levelname)s - %(name)s - %(message)s',
    handlers=[
        logging.StreamHandler()
    ]
)
logger = logging.getLogger("RedisConnection")

def connect_to_redis_cloud():
    """Connects to Redis cloud and returns the connection object with status."""

    logger.info("Attempting to connect to Redis Cloud...")
    try:
        redis_conn = rd.Redis(
            host=os.getenv('Redis_Host'),
            port=13330,
            username="default",
            password=os.getenv('Redis_password'),
            decode_responses=True
        )
        # Test connection
        if redis_conn.ping():
            logger.info("Successfully connected to Redis Cloud.")
        return redis_conn, True
    except rd.ConnectionError as conn_err:
        logger.error("ConnectionError: Unable to connect to Redis Cloud.", exc_info=True)
    except rd.AuthenticationError as auth_err:
        logger.error("AuthenticationError: Invalid credentials for Redis Cloud.", exc_info=True)
    except Exception as e:
        logger.exception("An unexpected error occurred while connecting to Redis Cloud.")
    return None, False
```

Reading Data from CSV and TXT files

```python
def read_data():
    """This function reads data from the userscores.csv and users.txt files
    and returns them as Pandas DataFrames."""

    logger.info("Reading Dataset from local...")
    try:
        user_score_data = pd.read_csv('data/userscores.csv')
    except FileNotFoundError:
        raise FileNotFoundError("The file 'data/userscores.csv' does not exist.")
    except pd.errors.ParserError:
        raise ValueError("Error parsing 'data/userscores.csv'. Please check the file format.")
    logger.info("Reading users.txt from local")
    try:
        with open('data/users.txt', 'r', encoding='utf-8') as file:
            lines = file.readlines()
        pattern = r'"([^"]+)"'
        records = []
        for line in lines:
            line = line.strip()
            parts = re.findall(pattern, line)
            record = {}
            i = 0
            while i < len(parts):
                if ':' in parts[i]:
                    key, value = parts[i].split(':', 1)
                    record[key] = value
                    i += 1  # Move to the next part
                else:
                    if i + 1 < len(parts):
                        key = parts[i]
                        value = parts[i + 1]
                        record[key] = value
                        i += 2
            records.append(record)
        user_data = pd.DataFrame(records)
    except FileNotFoundError:
        raise FileNotFoundError("The file 'data/users.txt' does not exist.")
    except Exception as e:
        raise ValueError(f"An error occurred while processing 'data/users.txt': {e}")
    return user_score_data, user_data
```

Write Function to Redis.

I use pandas dataframe to write into Redis which eliminates the need to specify any data type or schema

Assignment-5-JD.py

Assignment-5-JD.py > write_dataframe_to_redis

```python

def write_dataframe_to_redis(redis_conn, df, key_prefix):
    """Writes the Pandas DataFrame to Redis as hashes using a pipeline for bulk write."""
    logger.info("Bulk writing DataFrame to Redis...")
    pipeline = redis_conn.pipeline()
    for idx, row in df.iterrows():
        key = f"{key_prefix}:{idx}"
        data = row.to_dict()
        pipeline.hset(key, mapping=data)
        if 'user' in data:
            secondary_index_key = f"user_lookup:{data['user']}"
            pipeline.set(secondary_index_key, key)
    try:
        pipeline.execute()
        logger.info(f"Successfully bulk written {len(df)} records to Redis.")
    except Exception as e:
        logger.error(f"An error occurred while bulk writing to Redis: {e}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

Python

```
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5> & C:/Users/GOD/AppData/Local/Programs/Python/Python311/python.exe c:/Users/GOD/Desktop/IIT/BDM/Assig-5/Assignment-5-JD.py
2024-11-29 12:01:43,329 - INFO - RedisConnection - Attempting to connect to Redis Cloud...
2024-11-29 12:01:43,456 - INFO - RedisConnection - Successfully connected to Redis Cloud.
2024-11-29 12:01:43,456 - INFO - RedisConnection - Reading Dataset from local...
2024-11-29 12:01:43,489 - INFO - RedisConnection - Reading users.txt from local
2024-11-29 12:01:43,530 - INFO - RedisConnection - Bulk writing DataFrame to Redis...
2024-11-29 12:01:44,180 - INFO - RedisConnection - Successfully bulk written 5996 records to Redis.
2024-11-29 12:01:44,180 - INFO - RedisConnection - Bulk writing DataFrame to Redis...
2024-11-29 12:01:44,458 - INFO - RedisConnection - Successfully bulk written 5996 records to Redis.
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5>
```

Query 1

Assignment-5-JD.py

Assignment-5-JD.py > query_by_user

```python
def query_by_user(redis_conn, user_value):
    """
    Queries Redis to return all attributes of a user by the 'user' column value.

    Parameters:
    - redis_conn: Redis connection object
    - user_value: The value in the 'user' column to query for

    Returns:
    - A dictionary of user attributes or a message indicating the user was not found
    """
    try:
        lookup_key = f"user_lookup:{user_value}"
        primary_key = redis_conn.get(lookup_key)

        if primary_key:
            user_data = redis_conn.hgetall(primary_key)
            if user_data:
                logger.info(f"Successfully retrieved data for user '{user_value}'.")
                logger.info(user_data)
            else:
                logger.warning(f"No data found for user '{user_value}'.")
                # return f"No data found for user '{user_value}'."
        else:
            logger.warning(f"User '{user_value}' not found.")
            # return f"User '{user_value}' not found."
    except Exception as e:
        logger.error(f"An error occurred while querying user '{user_value}': {e}")
        # return None
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

Python

```
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5> & C:/Users/GOD/AppData/Local/Programs/Python/Python311/python.exe c:/Users/GOD/Desktop/IIT/BDM/Assig-5/Assi
gnment-5-JD.py
2024-11-29 12:05:34,102 - INFO - RedisConnection - Attempting to connect to Redis Cloud...
2024-11-29 12:05:34,228 - INFO - RedisConnection - Successfully connected to Redis Cloud.
2024-11-29 12:05:34,279 - INFO - RedisConnection - Successfully retrieved data for user '1'.
2024-11-29 12:05:34,280 - INFO - RedisConnection - {'user': '1', 'first_name': 'Mohammed', 'last_name': 'Ahern', 'email': 'mahern0@amazon.com', 'gen
der': 'male', 'ip_address': '180.132.241.207', 'country': 'China', 'country_code': 'CN', 'city': 'Yuanjue', 'longitude': '105.324979', 'latitude': '
29.55451', 'last_login': '1581151007'}
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5>
```

Query 2

Assignment-5-JD.py > query_by_users_for_coordinates

```python
def query_by_users_for_coordinates(redis_conn, user_value):
    """
    Queries Redis to return the coordinates (longitude and latitude) of a user by 'user' value.

    Parameters:
    - redis_conn: Redis connection object
    - user_value: The value in the 'user' column to query for (e.g., user ID)

    Returns:
    - A dictionary containing 'longitude' and 'latitude' or a message indicating the user was not found
    """
    try:
        # Lookup the key using the secondary index
        lookup_key = f"user_lookup:{user_value}"
        primary_key = redis_conn.get(lookup_key)

        if primary_key:
            # Use hget to get longitude and latitude fields of the user hash
            longitude = redis_conn.hget(primary_key, "longitude")
            latitude = redis_conn.hget(primary_key, "latitude")

            if longitude is not None and latitude is not None:
                logger.info(f"Successfully retrieved coordinates for user '{user_value}'.")
                logger.info(f"The longitude and latitude user {user_value} is {longitude} and {latitude}")
            else:
                logger.warning(f"Coordinates not found for user '{user_value}'.")
                # return f"Coordinates not found for user '{user_value}'."
        else:
            logger.warning(f"User '{user_value}' not found.")
            # return f"User '{user_value}' not found."
    except Exception as e:
        logger.error(f"An error occurred while querying coordinates for user '{user_value}': {e}")
        # return None
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5> & C:/Users/GOD/AppData/Local/Programs/Python/Python311/python.exe c:/Users/GOD/Desktop/IIT/BDM/Assig-5/Assignment-5-JD.py
2024-11-29 12:06:48,109 - INFO - RedisConnection - Attempting to connect to Redis Cloud...
2024-11-29 12:06:48,247 - INFO - RedisConnection - Successfully connected to Redis Cloud.
2024-11-29 12:06:48,326 - INFO - RedisConnection - Successfully retrieved coordinates for user '1'.
2024-11-29 12:06:48,326 - INFO - RedisConnection - The longitude and latitude user 1 is 105.324979 and 29.55451
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5>
```

Query 3

Assignment-5-JD.py > query3

```python
def query3(redis_conn):
    Queries Redis to return the keys and last names of the users whose IDs do not start with an odd number.

    Parameters:
    - redis_conn: Redis connection object

    Returns:
    - A Pandas DataFrame containing 'key' and 'last_name' for users whose IDs do not start with an odd number.
    """
    logger.info("Querying for users whose IDs do not start with an odd number...")

    try:
        data = []
        for key in redis_conn.scan_iter("user_data:*"):

            user_id = redis_conn.hget(key, "user")

            if user_id and len(user_id) > 0:
                if user_id[0] not in ['1', '3', '5', '7', '9']:
                    # Retrieve the last name
                    last_name = redis_conn.hget(key, "last_name")
                    if last_name:
                        data.append([key, last_name])
        df = pd.DataFrame(data, columns=['key', 'last_name'])

        logger.info(f"Successfully retrieved {len(df)} users whose IDs do not start with an odd number.")
        logger.info(df)
        # return df

    except Exception as e:
        logger.error(f"An error occurred while querying for users whose IDs do not start with an odd number: {e}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5> & C:/Users/GOD/AppData/Local/Programs/Python/Python311/python.exe c:/Users/GOD/Desktop/IIT/BDM/Assig-5/Assignment-5-JD.py
2024-11-29 12:09:23,126 - INFO - RedisConnection - Attempting to connect to Redis Cloud...
2024-11-29 12:09:23,304 - INFO - RedisConnection - Successfully connected to Redis Cloud.
2024-11-29 12:09:23,304 - INFO - RedisConnection - Querying for users whose IDs do not start with an odd number...
2024-11-29 12:13:34,090 - INFO - RedisConnection - Successfully retrieved 2444 users whose IDs do not start with an odd number.
2024-11-29 12:13:34,091 - INFO - RedisConnection -                  key   last_name
0       user_data:66       Rioch
1     user_data:2451      Cannop
2     user_data:4083     Franchi
3     user_data:2305     Verrall
4     user_data:2056        Fabb
...              ...         ...
2439  user_data:4647     Sellors
2440  user_data:2635       Kowal
2441   user_data:818    Starmont
2442  user_data:4136  Moneypenny
2443  user_data:4205     Brosnan

[2444 rows x 2 columns]
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5>
```

Query 4

Assignment-5-JD.py

```python
def query4(redis_conn):
    """
    Queries Redis to return female users in China or Russia with latitude between 40 and 46.
    Parameters:
    - redis_conn: Redis connection object
    Returns:
    - A Pandas DataFrame containing 'user', 'country', 'latitude', 'longitude', and 'last_name'
      for female users in China or Russia with latitude between 40 and 46.
    """
    logger.info("Querying Redis for female users in China or Russia with latitude between 40 and 46...")
    try:
        data = []
        # Scan through all keys that match user_data:*
        for key in redis_conn.scan_iter("user_data:*"):
            # Get required fields to apply filtering
            gender = redis_conn.hget(key, "gender")
            country = redis_conn.hget(key, "country")
            latitude = redis_conn.hget(key, "latitude")
            last_name = redis_conn.hget(key, "last_name")
            user_id = redis_conn.hget(key, "user")
            longitude = redis_conn.hget(key, "longitude")
            if (
                gender == "female" and
                country in ["China", "Russia"] and
                latitude is not None
            ):
                try:
                    latitude_value = float(latitude)
                    if 40 <= latitude_value <= 46:
                        # Append the matching data to the list
                        data.append([user_id, country, latitude, longitude, last_name])
                except ValueError:
                    logger.warning(f"Skipping invalid latitude value: {latitude} for user {user_id}")
        df = pd.DataFrame(data, columns=['user', 'country', 'latitude', 'longitude', 'last_name'])
        logger.info(f"Successfully retrieved {len(df)} female \
            users in China or Russia with latitude between 40 and 46.")
        logger.info(df)
    except Exception as e:
        logger.error(f"An error occurred while querying for female \
            users in China or Russia with latitude between 40 and 46: {e}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
2024-11-29 12:17:49,001 - INFO - RedisConnection - Querying Redis for female users in China or Russia with latitude between 40 and 46...
2024-11-29 12:31:20,479 - INFO - RedisConnection - Successfully retrieved 97 female users in China or Russia with latitude between 40 and 46.
2024-11-29 12:31:20,479 - INFO - RedisConnection -     user country    latitude    longitude last_name
0   3568   China   41.985186   122.836723  MacAless
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
2   5306  Russia  43.2374865   45.0503473    Maddin
3    658   China   42.629452   120.639361   Allanby
4   1143   China   43.817071   125.323544      Ryde
1   5920  Russia  43.8507498  131.8648449    Dollen
2   5306  Russia  43.2374865   45.0503473    Maddin
3    658   China   42.629452   120.639361   Allanby
1   5920  Russia  43.8507498  131.8648449    Dollen
2   5306  Russia  43.2374865   45.0503473    Maddin
1   5920  Russia  43.8507498  131.8648449    Dollen
```

```
2024-11-29 12:31:20,479 - INFO - RedisConnection -     user country    lati
0   3568   China   41.985186   122.836723  MacAless
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
2   5306  Russia  43.2374865   45.0503473    Maddin
3    658   China   42.629452   120.639361   Allanby
4   1143   China   43.817071   125.323544      Ryde
1   5920  Russia  43.8507498  131.8648449    Dollen
2   5306  Russia  43.2374865   45.0503473    Maddin
3    658   China   42.629452   120.639361   Allanby
1   5920  Russia  43.8507498  131.8648449    Dollen
2   5306  Russia  43.2374865   45.0503473    Maddin
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
1   5920  Russia  43.8507498  131.8648449    Dollen
2   5306  Russia  43.2374865   45.0503473    Maddin
2   5306  Russia  43.2374865   45.0503473    Maddin
3    658   China   42.629452   120.639361   Allanby
4   1143   China   43.817071   125.323544      Ryde
..   ...     ...         ...          ...       ...
..   ...     ...         ...          ...       ...
92  1576  Russia  45.8697083   43.3478599     Relfe
93  5162   China   42.016401   121.668489   Rolfini
93  5162   China   42.016401   121.668489   Rolfini
94  4372   China   44.439044  125.1797741   Clawley
94  4372   China   44.439044  125.1797741   Clawley
95  4364  Russia    43.47139     43.84694    Adamek
96  2568   China   44.766541   129.688614    Landre

[97 rows x 5 columns]
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5>
```

Query 5

Assignment-5-JD.py > main

```python
def query5(redis_conn):
    try:
        data = []
        for key in redis_conn.scan_iter("user_score_data:*"):
            leaderboard = redis_conn.hget(key, "leaderboard")
            score = redis_conn.hget(key, "score")
            user_id = redis_conn.hget(key, "user:id")
            key, value = user_id.split(":")
            if leaderboard == "2.0" and score is not None:
                try:
                    score_value = float(score)
                    data.append({"user_id": int(value), "score": score_value})
                except ValueError:
                    logger.warning(f"Skipping invalid score value: {score} for user {user_id}")
        df = pd.DataFrame(data)

        if df.empty:
            logger.warning("No data found for leaderboard 2.")
            return pd.DataFrame()
        top_10_players_df = df.sort_values(by='score', ascending=False).head(10)
        logger.info(top_10_players_df)
        email_data = []
        for user_id in top_10_players_df['user_id']:
            user_key = f"user_data:{user_id}"
            email = redis_conn.hget(user_key, "email")
            if email:
                email_data.append({"user_id": user_id, "email": email})

        # Convert email data into a DataFrame
        email_df = pd.DataFrame(email_data)

        logger.info(f"Successfully retrieved email IDs for the top 10 players from leaderboard 2.")
        logger.info(email_df)


    except Exception as e:
        logger.error(f"An error occurred while querying for the top 10 players from leaderboard 2: {e}")
        #return pd.DataFrame()  # Return an empty DataFrame in case of an error

def main():
    redis_connection, status = connect_to_redis_cloud()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

```
2024-11-29 12:48:50,929 - INFO - RedisConnection - Successfully retrieved email IDs for the top 10 players from leaderboard 2.
2024-11-29 12:48:50,929 - INFO - RedisConnection - Successfully retrieved email IDs for the top 10 players from leaderboard 2.
2024-11-29 12:48:50,929 - INFO - RedisConnection -    user_id                          email
0     2468        cpervoed0@macromedia.com
1     4786       diamittiily@webeden.co.uk
2     5861            ahileyo1@plala.or.jp
3     1972      esimonianr0@opensource.org
4      318             dsurcombe8u@tiny.cc
5     2971     msaulqz@cargocollective.com
6     5771             sroydslj@cpanel.net
7     5830        rpitfordn6@artisteer.com
8      501           cdarcodx@omniture.com
9     2491  bmendezdn@pagesperso-orange.fr
PS C:\Users\GOD\Desktop\IIT\BDM\Assig-5>
```