SQL database on healthy minds project

YUANXUE MA

August 2023

1 Database structure

.

The database developed for the Healthy Mind app comprises five distinct tables. The 't1 cleaed' table captures scores for awareness, connection, insight, and purpose prior to using the app. Conversely, the 't2 cleaned' table records these scores after app utilization. Participants' demographic information is consolidated in the 'demographic' table. Additionally, stress scores of participants before and after using the app are stored in the 't1 stress' and 't2 stress' tables, respectively. Data is organized in a wide format, with columns for the four aforementioned scores and ten columns representing individual stress-related questions.

2 Database connection

The database is hosted on an Amazon RDS. Amazon RDS is a managed relational database service by AWS, which makes it easier to set up, operate, and scale a relational database in the cloud.To connect to the database,it needs correct host name,database name,and credentials. All these can be setup when using amazon RDS.

```
def create_connection():
    try:
        connection = mysql.connector.connect(host='project.
    cwfrcwlbmlxn.us-east-2.rds.amazonaws.com',
                                             database='sql',
                                             user='',
                                             password='')
        if connection.is_connected():
            db_Info = connection.get_server_info()
            print("Connected to MySQL Server version ", db_Info)
            return connection
    except Error as e:
        print("Error while connecting to MySQL", e)
        return None
```

Listing 1: connect to my database

For database security, i will not provide username and password. But the code above demonstrate how to connect to amazon server.

```
def query(q):
    connection = create_connection()
    cursor = None
    results = None
    if connection:
        try:
            cursor = connection.cursor()
            cursor.execute(q)
            results = cursor.fetchall()
        except Error as e:
            print("Error executing the query:", e)
        finally:
            connection.close()
    return results, cursor
```

Listing 2: query data from my database

This function establishes a connection to the database using create connection(). If the connection is successful, it tries to execute the SQL query provided as the parameter q.If the query is successfully executed, it fetches all the results and stores them in the results variable.It returns the results and the cursor object. If there's an error in executing the query, the error is printed. The connection to the database is closed after executing the query.

```
def select(q):
    connection = create_connection()
    if connection:
        try:
            df = read_sql_query(q, connection)
            return df
        except Error as e:
            print("Error executing the query:", e)
        finally:
            connection.close()
    return None
```

Listing 3: sql data to pandas dataframe

This function also establishes a connection to the database using create connection().It attempts to read the SQL query results into a dataframe using the read sql query function from the pandas library.If successful, it returns the dataframe. Otherwise, it prints an error message.After executing the query, the connection to the database is closed.

In conclusion, the query function can be used for executing a variety of SQL queries on the database, like INSERT, UPDATE, or DELETE statements. The select function is specifically tailored for retrieving data from the database. It returns the results as a dataframe, which is useful for data analysis and plotting in Python using libraries like pandas,matplotlib and seaborn.

3 Database upload on Amazon cloud

Amazon RDS (Relational Database Service) is a managed relational database service provided by AWS (Amazon Web Services). You can create, run, and manage multiple types of database engines, including MySQL, PostgreSQL, MariaDB, Oracle, and Microsoft SQL Server.

Here's a step-by-step guide on how to create and upload a database to Amazon RDS:

3.1 Accessing AWS Management Console

Begin by logging into the AWS Console. If you don't have an account, you need to create one. Once inside:

- Navigate to the Amazon RDS Dashboard. Click Service on the upper left
- Look for "RDS" listed under the "Database" category and select it.

3.2 Initiating a New Database Instance

1. Click on the "Create database" option.
2. Opt for your preferred database engine (like MySQL, PostgreSQL).
3. Tailor the database's settings:
    - **Edition:** Choose your desired database engine version.
    - **DB Instance Class:** Select a size commensurate with your needs.
    - **Multi-AZ Deployment:** Enable for robust availability and failover capabilities.
    - **Storage:** Designate both the type and size of the desired storage.
4. Configure the Master username and password for authentication. This becomes your main administrative account.

3.3 Advanced Configuration

- Assign a name to your Database.
- Adjust settings like VPC, subnet group, public access, and the database port.In this case, change CIDR/IP - Inbound and CIDR/IP - Outbound to 0.0.0.0 to gain public access
- Define backup and monitoring preferences as necessary.

3.4 Starting the Database Instance

Activate the "Create database" button located at the bottom. This will initiate the database instance provisioning.

3.5 Uploading or Migrating Data

After the RDS instance becomes operational:

- Connect using mysqlworkbench.
- Enter the previously set hostname, username, and password to establish a connection to the Amazon server.

3.6 RDS Database Creation

Once connected, you can proceed to create your desired database using DataGrip.

4 Connection database in DataGrip

Before initializing a database or inserting data, establish a connection to your database server:

1. Launch DataGrip and select the '+' button in the 'Database' view, commonly found on the interface's right side.
2. Identify your database type (e.g., PostgreSQL, MySQL).
3. Input the connection details, comprising the hostname, port, username, and password you set up in amazon cloud.
4. Validate the connection via the 'Test Connection' button.
5. If accurate, press 'OK' to finalize the connection.

5 Creating a New Database

1. In the 'Database' view, right-click on your connection.
2. Navigate to 'New' and choose 'Database'.
3. Assign a name to your database and confirm with 'OK'.
4. The fresh database will manifest in the list under your connection.

5.1 Generating a New Table

1. Right-click on the recently crafted database within the 'Database' view.
2. Select 'New' followed by 'Table'.
3. Determine a name for your table.
4. Use the table design interface to add columns, assign data types, and stipulate other attributes.
5. Store the table.

5.2 Data Insertion

1. Proceed to the table designated for data insertion.
2. Right-click and opt for ‘Import/Export‘.
3. Insert your csv file
4. choose ‘Submit‘ to update the table with the data.

Conclusion

This document offers a comprehensive, step-by-step guide on database creation using DataGrip. It details the procedures for uploading databases to Amazon servers and provides an overview of the current database architecture.