# Disclaimer: This output contains AI-generated content; user is advised to review it before consumption.  
  
#\*Start of AI Generated Content\*

# **PostgreSQL Database Interaction Codebase**

## **Introduction**

This document provides an in-depth explanation of a Python codebase designed to interact with a PostgreSQL database. The code facilitates two primary functionalities: calculating the sum of values in a specified column and determining the age of an individual based on their birthdate. The document outlines the problem statement, solution approach, code functionality, input/output formats, and concludes with a summary of the code's purpose.

## **Problem Statement**

* **Database Connectivity Issues**: Establishing a reliable connection to a PostgreSQL database.
* **Data Aggregation**: Calculating the sum of values in a specific column within a database table.
* **Data Transformation**: Computing the age of an individual based on their birthdate stored in the database.

## **Solution Approach**

* **Database Connection**: Utilize the psycopg2 library to establish a secure connection to the PostgreSQL database.
* **SQL Query Execution**: Craft and execute parameterized SQL queries to aggregate data (sum of values) and transform data (calculate age).
* **Error Handling**: Implement try-except blocks to catch and handle potential database connection errors, invalid table names, and invalid column names.

## **Functionality of Code**

### **1. Constants and Static String Values**

* **Database Constants**: Define database host, name, user, and password as constants for easy configuration.
* **SQL Queries**: Store SQL queries as string constants for calculating the sum of values and determining age.
* **Error Messages**: Define error messages for database connection failures, invalid table names, and invalid column names.

### **2. PostgreSQL Database Connection**

* **establish\_db\_connection Function**:
* Establishes a connection to the PostgreSQL database using psycopg2.
* Returns the connection object upon success; otherwise, returns None and prints the error message.

### **3. PostgreSQL Functions**

#### **a. Sum of Values**

* **sum\_of\_values Function**:
* Calculates the sum of values in a specified column within a given table.
* Takes table\_name and column\_name as input parameters.
* Returns the sum of values as a float; otherwise, returns None and prints the error message.

#### **b. Calculate Age**

* **calculate\_age Function**:
* Computes the age of an individual based on their birthdate.
* Accepts table\_name, birthdate\_column, and person\_id as input parameters.
* Returns the age as a string; otherwise, returns None and prints the error message.

## **Input and Output Format**

### **Input Format**

|  |  |  |
| --- | --- | --- |
| **Function** | **Input Parameters** | **Data Type** |
| sum\_of\_values | table\_name , column\_name | str , str |
| calculate\_age | table\_name , birthdate\_column , person\_id | str , str , int |
| establish\_db\_connection | **None** (uses predefined constants) | **-** |

### **Output Format**

|  |  |  |
| --- | --- | --- |
| **Function** | **Output** | **Data Type** |
| sum\_of\_values | Sum of values in the specified column | float (or None on error) |
| calculate\_age | Age of the individual | str (or None on error) |
| establish\_db\_connection | PostgreSQL database connection object | psycopg2.extensions.connection (or None on error) |

## **Conclusion**

This Python codebase provides a structured approach to interacting with a PostgreSQL database, enabling the calculation of sum of values in a column and the determination of an individual's age based on their birthdate. By utilizing the psycopg2 library and implementing robust error handling, the code ensures a reliable and efficient database interaction experience.

#\*End of AI Generated Content\*