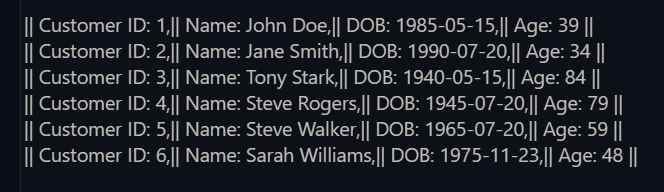
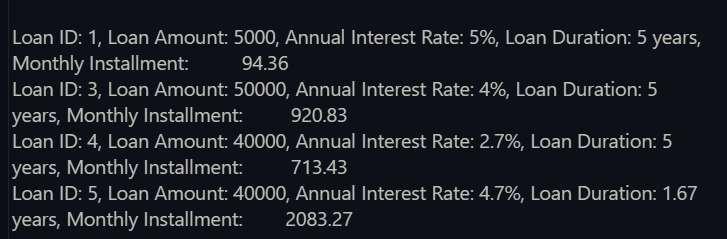
**Exercise 4: Functions**

* **Scenario 1:** Calculate the age of customers for eligibility checks.
* **Question:** Write a function *CalculateAge* that takes a customer's date of birth as input and returns their age in years
* The *CalculateAge* function computes the age by subtracting the birth year from the current year and returns the result.
* The DECLARE block initializes a cursor to select customer details (*CustomerID*, *Name*, *DOB*) from the Customers table.
* A loop fetches each customer's details, calculates their age using the *CalculateAge* function, and prints the details with the calculated age.
* The cursor is opened at the start and closed after processing all records.

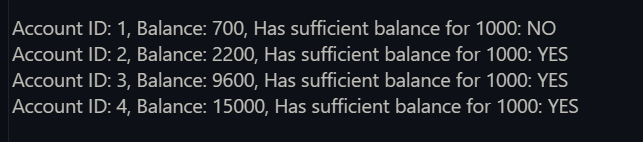
**Here is the output of the code:**

The function prints the all-customer’s age according to the Customers table

* **Scenario 2:** The bank needs to compute the monthly installment for a loan.
* **Question:** Write a function ***CalculateMonthlyInstallment*** that takes the loan amount, interest rate, and loan duration in years as input and returns the monthly installment amount.
* The *CalculateMonthlyInstallment* function calculates monthly loan *installments* based on the loan amount, annual interest rate, and loan duration in years using the EMI formula.
* It converts the annual interest rate to a monthly rate and the loan duration in years to months before applying the EMI formula.
* The block of PL/SQL code opens a cursor to fetch loan details from the Loans table.
* It calculates the loan duration in years and uses the *CalculateMonthlyInstallment* function to compute the monthly *installment* for each loan.
* The results, including loan details and calculated monthly *installments*, are printed to the output with two decimal places.

**Here is the output of the code**:

* **Scenario 3:** Check if a customer has sufficient balance before making a transaction.
* **Question:** Write a function ***HasSufficientBalance*** that takes an account ID and an amount as input and returns a Boolean indicating whether the account has at least the specified amount.
* **Function Definition**: The *HasSufficientBalance* function takes an account ID and an amount as parameters and checks if the account has a balance greater than or equal to the specified amount.
* **Balance Retrieval**: It retrieves the account balance using a SELECT statement and stores it in the variable v\_*Balance*.
* **Balance Check**: The function compares the retrieved balance to the specified amount and returns TRUE if the balance is sufficient, otherwise FALSE.
* **Exception Handling**: If no data is found for the provided account ID (account does not exist), the function catches the NO\_DATA\_FOUND exception and returns FALSE.
* **Testing Block**: A DECLARE block tests the function for all accounts by iterating through them, calling the *HasSufficientBalance* function, and outputting whether each account has sufficient balance for the specified amount.

**The output of the code:**