**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.

CREATE OR REPLACE FUNCTION CalculateAge(

p\_DOB IN DATE

) RETURN NUMBER IS

v\_age NUMBER;

BEGIN

v\_age := FLOOR(MONTHS\_BETWEEN(SYSDATE, p\_DOB) / 12);

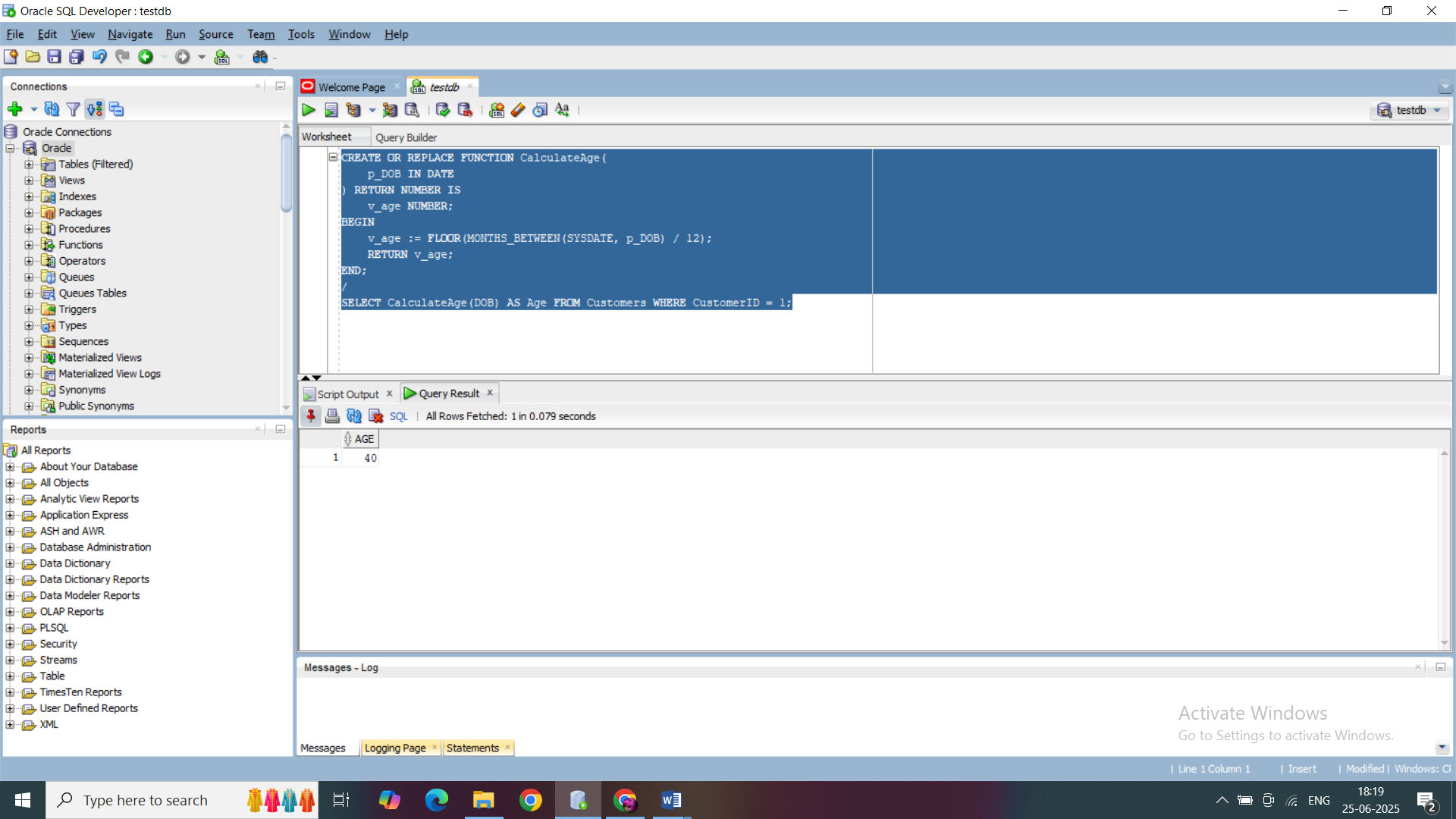
RETURN v\_age;

END;

/

SELECT CalculateAge(DOB) AS Age FROM Customers WHERE CustomerID = 1;

OUTPUT:



**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.

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_LoanAmount IN NUMBER,

p\_InterestRate IN NUMBER, -- annual rate in %

p\_Years IN NUMBER

) RETURN NUMBER IS

v\_monthlyRate NUMBER := p\_InterestRate / 12 / 100;

v\_months NUMBER := p\_Years \* 12;

v\_emi NUMBER;

BEGIN

IF v\_monthlyRate = 0 THEN

v\_emi := p\_LoanAmount / v\_months;

ELSE

v\_emi := (p\_LoanAmount \* v\_monthlyRate \* POWER(1 + v\_monthlyRate, v\_months)) /

(POWER(1 + v\_monthlyRate, v\_months) - 1);

END IF;

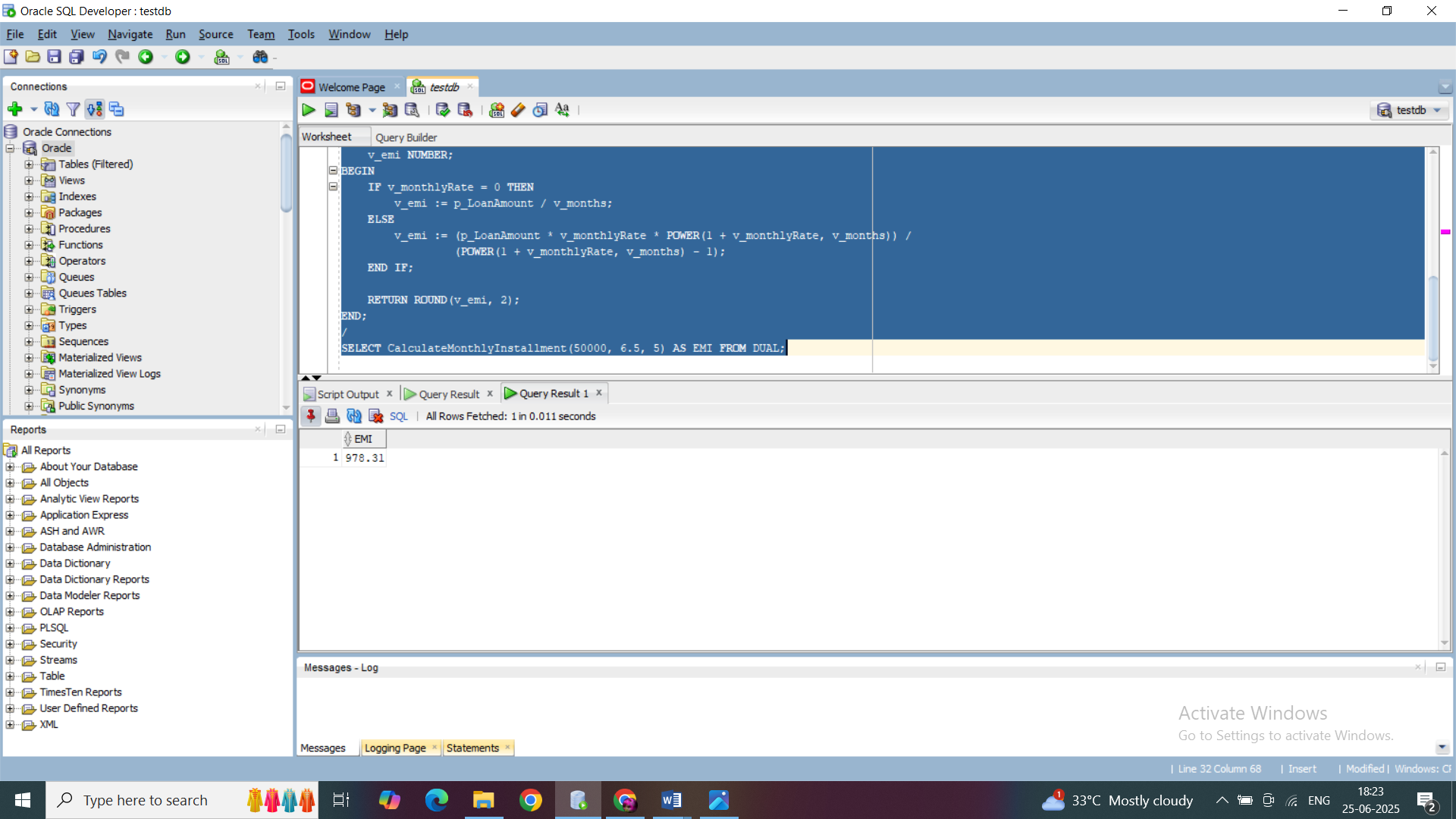
RETURN ROUND(v\_emi, 2);

END;

/

SELECT CalculateMonthlyInstallment(50000, 6.5, 5) AS EMI FROM DUAL;

OUTPUT:



**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.

CREATE OR REPLACE FUNCTION HasSufficientBalance(

p\_AccountID IN NUMBER,

p\_Amount IN NUMBER

) RETURN BOOLEAN IS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = p\_AccountID;

RETURN v\_balance >= p\_Amount;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

WHEN OTHERS THEN

RETURN FALSE;

END;

/

DECLARE

v\_result BOOLEAN;

BEGIN

v\_result := HasSufficientBalance(1, 300);

IF v\_result THEN

DBMS\_OUTPUT.PUT\_LINE('Sufficient balance.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Insufficient balance.');

END IF;

END;

/

OUTPUT:

