**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.

CREATE OR REPLACE FUNCTION CalculateAge(dob DATE)

RETURN NUMBER IS

age NUMBER;

BEGIN

age := FLOOR(MONTHS\_BETWEEN(SYSDATE, dob) / 12);

RETURN age;

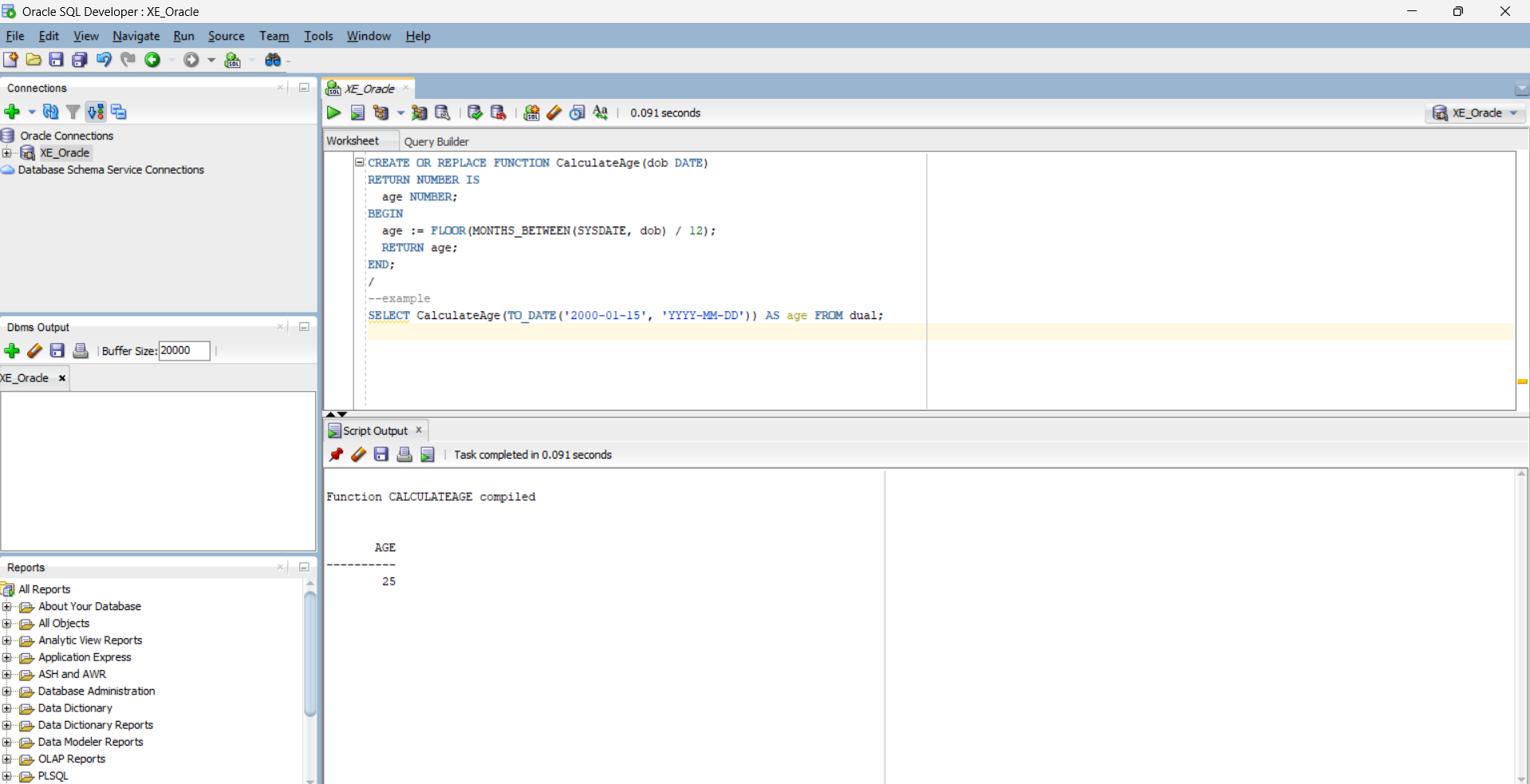
END;

/

--example

SELECT CalculateAge(TO\_DATE('2000-01-15', 'YYYY-MM-DD')) AS age FROM dual;

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(

amount NUMBER,

annual\_rate NUMBER,

years NUMBER

)

RETURN NUMBER IS

monthly\_rate NUMBER;

n NUMBER;

emi NUMBER;

BEGIN

monthly\_rate := annual\_rate / 12 / 100;

n := years \* 12;

emi := (amount \* monthly\_rate \* POWER(1 + monthly\_rate, n)) /

(POWER(1 + monthly\_rate, n) - 1);

RETURN ROUND(emi, 2);

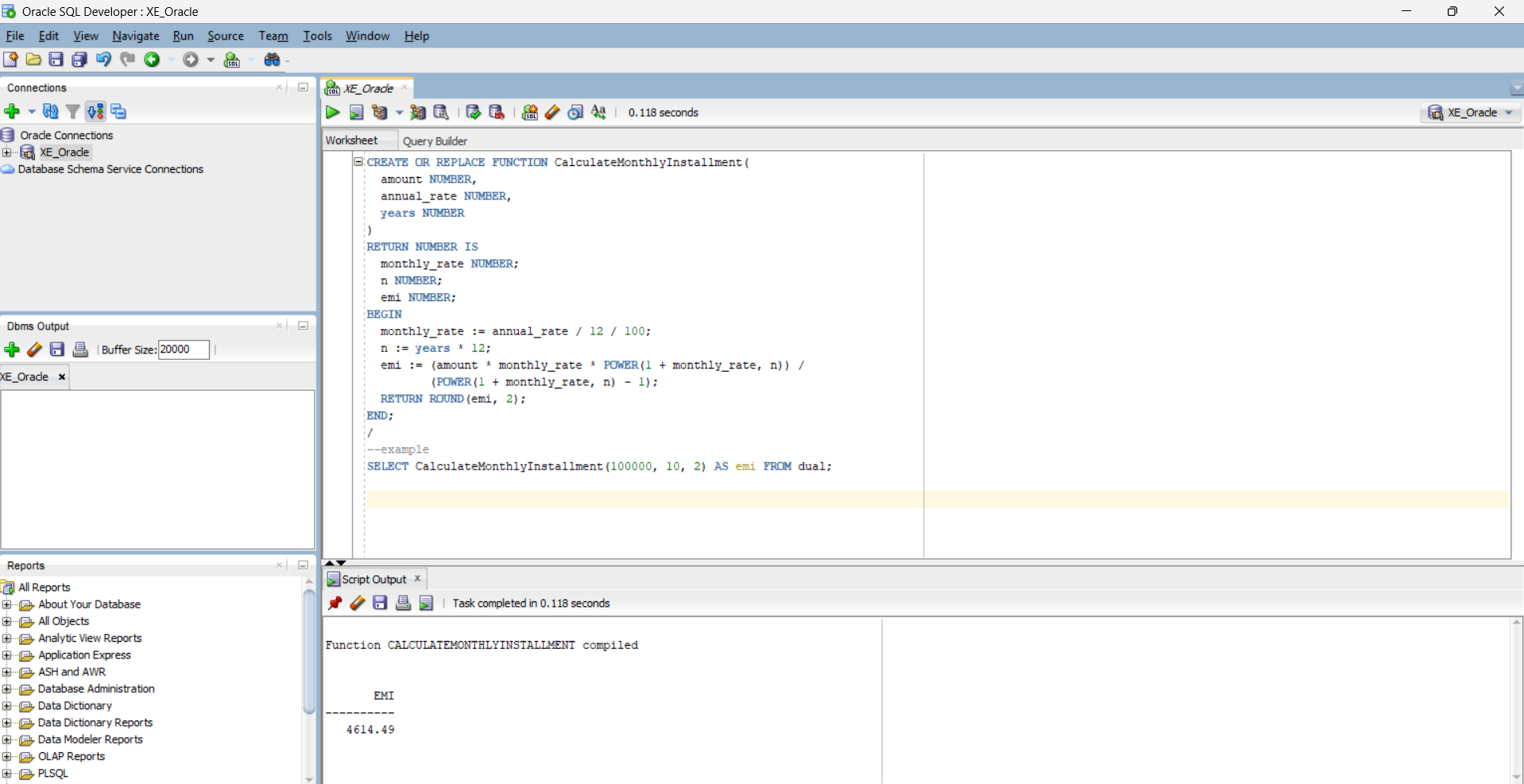
END;

/

--example

SELECT CalculateMonthlyInstallment(100000, 10, 2) 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 TABLE accounts (

acc\_no NUMBER PRIMARY KEY,

holder\_name VARCHAR2(100),

balance NUMBER

);

CREATE OR REPLACE FUNCTION HasSufficientBalance(

acc\_id IN NUMBER,

amount IN NUMBER

)

RETURN BOOLEAN IS

acc\_balance NUMBER;

BEGIN

SELECT balance INTO acc\_balance FROM accounts WHERE acc\_no = acc\_id;

RETURN acc\_balance >= amount;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN FALSE;

END;

/

--example usage

DECLARE

result BOOLEAN;

BEGIN

result := HasSufficientBalance(401, 2000);

IF result THEN

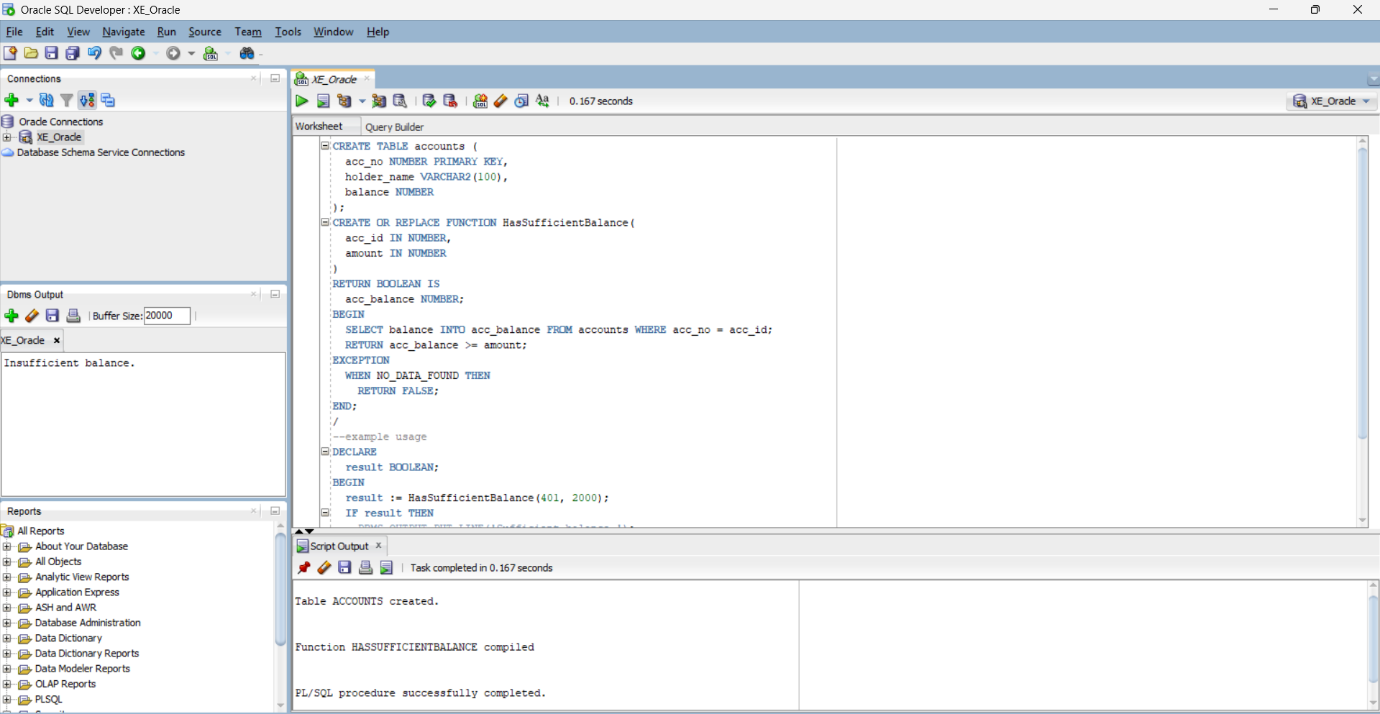
DBMS\_OUTPUT.PUT\_LINE('Sufficient balance.');

ELSE

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

END IF;

END;

Output: