Control Structures in PL/SQL

PL/SQL control structures allow you to control the flow of your program.  
They include:

* Conditional Statements: IF, CASE
* Looping Statements: LOOP, WHILE, FOR

In this exercise, we use:

* FOR loops to iterate over rows
* IF conditions to check and apply logic based on column values

**2. Table Structure and Attributes**

**🔹 Table: bank\_customers**

* customer\_id: NUMBER (Primary Key)
* name: VARCHAR2(50)
* age: NUMBER
* balance: NUMBER
* loan\_intrest: NUMBER(5,2)
* is\_vip: CHAR(1)

**🔹 Table: loans**

* loan\_id: NUMBER (Primary Key)
* customer\_id: NUMBER (Foreign Key → bank\_customers.customer\_id)
* due\_date: DATE

**3. Table Creation Script**

CREATE TABLE bank\_customers (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(50),

age NUMBER,

balance NUMBER,

loan\_intrest NUMBER(5,2),

is\_vip CHAR(1)

);

CREATE TABLE loans (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

due\_date DATE,

CONSTRAINT fk\_customer

FOREIGN KEY (customer\_id)

REFERENCES bank\_customers(customer\_id)

);

**4. Inserting Sample Data**

INSERT INTO bank\_customers VALUES (1, 'Chaitanya',  23,  5000,   9.50, NULL);

INSERT INTO bank\_customers VALUES (2, 'Varma',      67, 15000,  10.00, NULL);

INSERT INTO bank\_customers VALUES (3, 'Bunny',      45,  8000,  11.00, NULL);

INSERT INTO bank\_customers VALUES (4, 'Pavani',     70, 12000,   8.75, NULL);

INSERT INTO bank\_customers VALUES (5, 'Bhanu',      62,  9500,   9.25, NULL);

INSERT INTO bank\_customers VALUES (6, 'Raj',        31, 20000,  10.50, NULL);

INSERT INTO bank\_customers VALUES (7, 'Sneha',      55, 11000,   9.00, NULL);

INSERT INTO bank\_customers VALUES (8, 'Teja',       65,  7500,  10.25, NULL);

INSERT INTO bank\_customers VALUES (9, 'Meena',      28,  6000,   9.75, NULL);

INSERT INTO bank\_customers VALUES (10, 'Karthik',   72, 16000,   8.50, NULL);

INSERT INTO loans VALUES (101, 2,  TO\_DATE('05-JUL-2025', 'DD-MON-YYYY'));

INSERT INTO loans VALUES (102, 4,  TO\_DATE('15-JUL-2025', 'DD-MON-YYYY'));

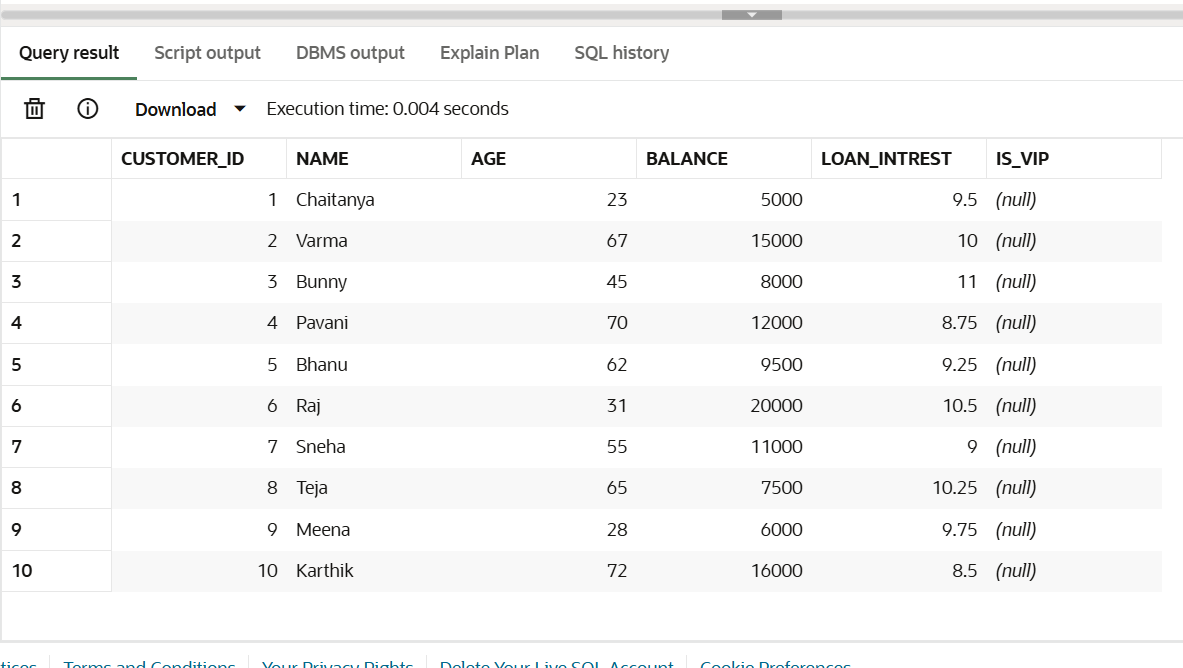
INSERT INTO loans VALUES (103, 6,  TO\_DATE('25-JUL-2025', 'DD-MON-YYYY'));

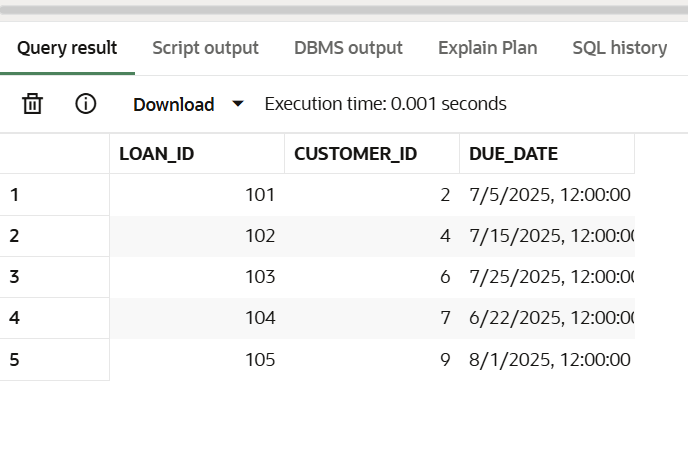
INSERT INTO loans VALUES (104, 7,  TO\_DATE('22-JUN-2025', 'DD-MON-YYYY'));

INSERT INTO loans VALUES (105, 9,  TO\_DATE('01-AUG-2025', 'DD-MON-YYYY'));

COMMIT;

Tables :





**Scenario 1: Senior Citizen Interest Discount**

**📘 Description:**

Reduce loan interest by 1% for all customers older than 60 years.

**💻 Code:**

BEGIN

FOR cust IN (

SELECT customer\_id, age, loan\_intrest

FROM bank\_customers

) LOOP

IF cust.age > 60 THEN

UPDATE bank\_customers

SET loan\_intrest = cust.loan\_intrest - 1

WHERE customer\_id = cust.customer\_id;

END IF;

END LOOP;

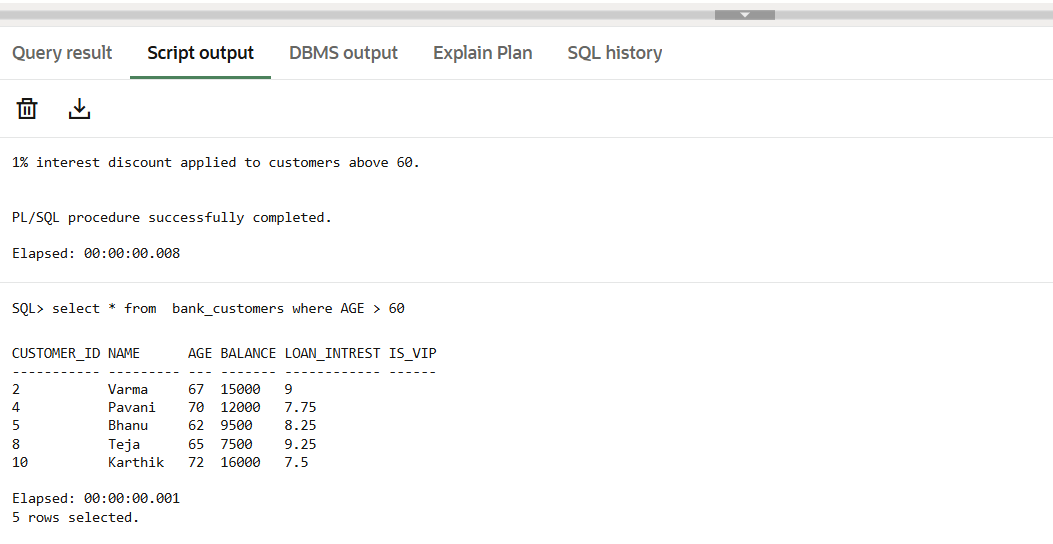
COMMIT;

DBMS\_OUTPUT.PUT\_LINE('1% interest discount applied to customers above 60.');

END;

**📖 Explanation:**

The loop checks each customer's age. If the age is above 60, their loan\_intrest is reduced by 1%.



**Scenario 2: Promote to VIP Based on Balance**

**📘 Description:**

Customers with a balance greater than 10,000 are marked as VIP (is\_vip = 'Y'), else is\_vip = 'N'.

**💻 Code:**

BEGIN

FOR cust IN (

SELECT customer\_id, balance

FROM bank\_customers

) LOOP

IF cust.balance > 10000 THEN

UPDATE bank\_customers

SET is\_vip = 'Y'

WHERE customer\_id = cust.customer\_id;

ELSE

UPDATE BANK\_CUSTOMERS SET is\_vip ='N'

WHERE CUSTOMER\_ID=cust.customer\_id;

END IF;

END LOOP;

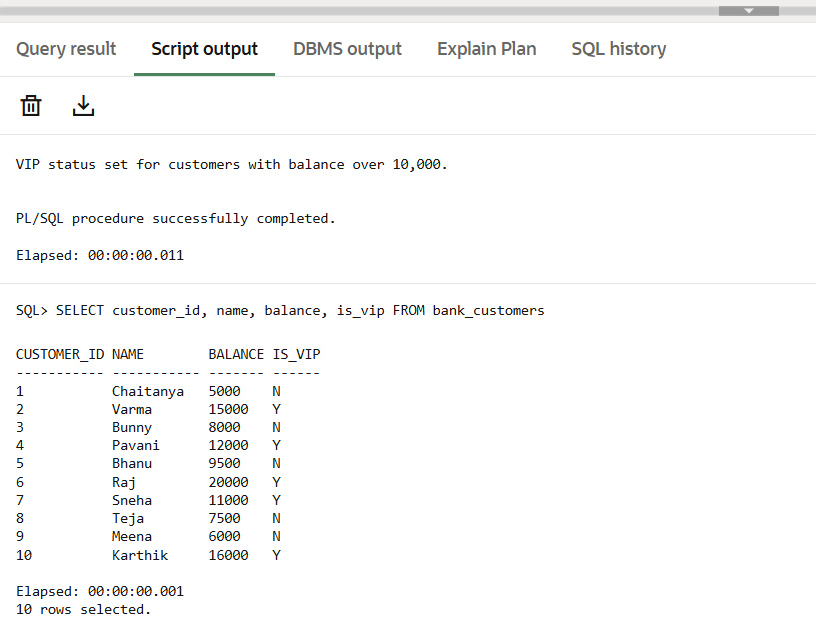
COMMIT;

DBMS\_OUTPUT.PUT\_LINE('VIP status set for customers with balance over 10,000.');

END;

**📖 Explanation:**

This script sets VIP status based on the balance of each customer.



**Scenario 3: Loan Due Reminders**

**📘 Description:**

Print reminder messages for customers whose loan is due within the next 30 days.

**💻 Code:**

BEGIN

FOR loan\_rec IN (

SELECT l.loan\_id, l.due\_date, c.customer\_id, c.name

FROM loans l

JOIN bank\_customers c ON l.customer\_id = c.customer\_id

WHERE l.due\_date <= SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE(

'📢 Reminder: Loan ' || loan\_rec.loan\_id ||

' for ' || loan\_rec.name ||

' (Customer ID: ' || loan\_rec.customer\_id || ')' ||

' is due on ' || TO\_CHAR(loan\_rec.due\_date, 'DD-MON-YYYY')

);

END LOOP;

END;

**📖 Explanation:**

Joins loans and bank\_customers, and prints reminders if the due date is within the next 30 days.

