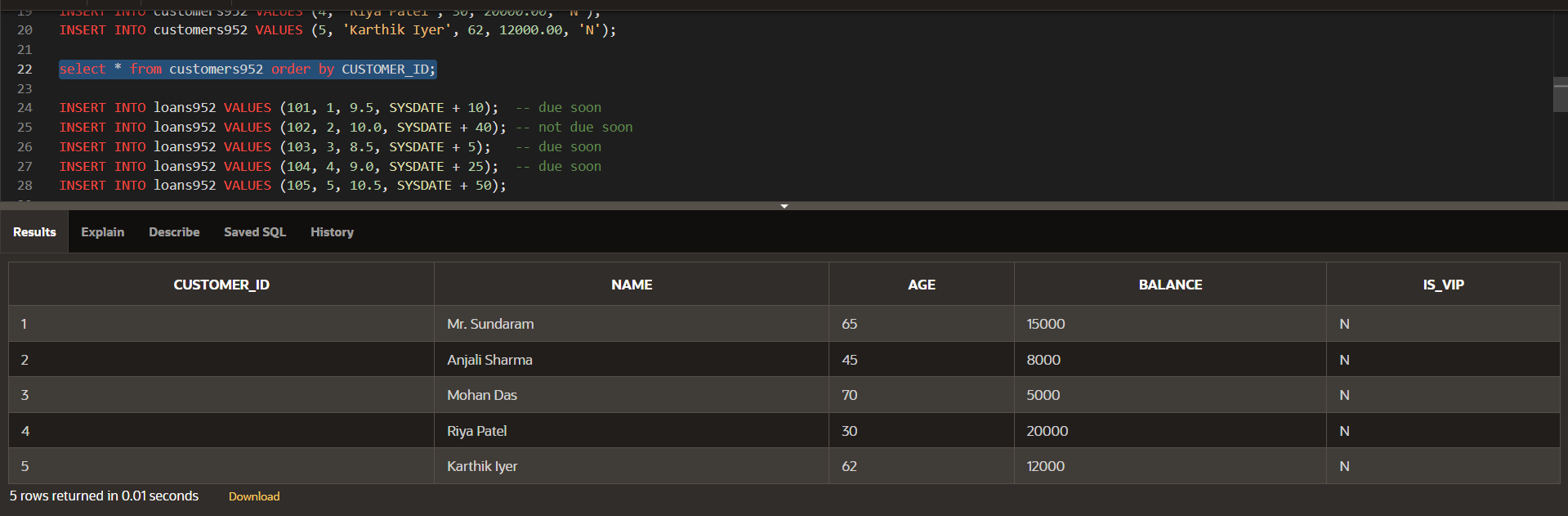
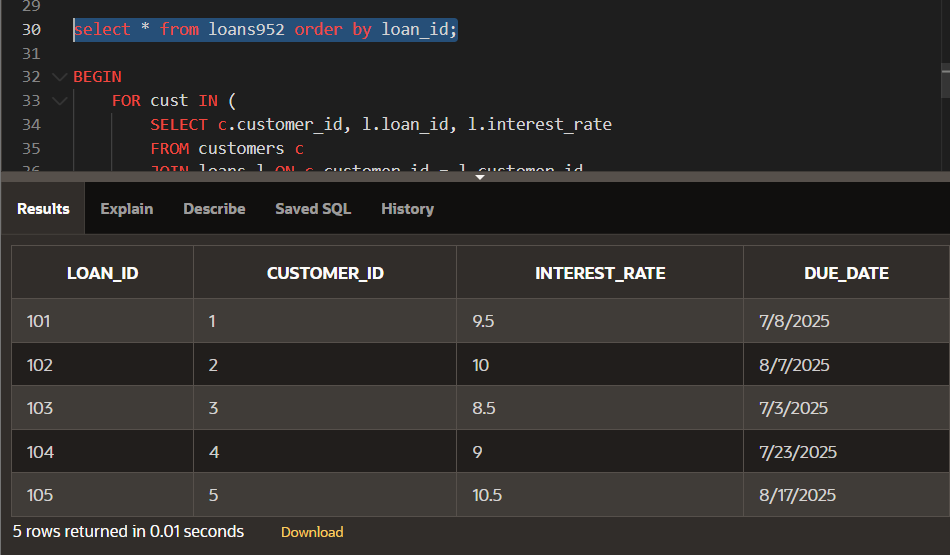
**WEEK 2 – PL/SQL - 6420952**

**Exercise 1: Control Structures**





**Scenario 1:** The bank wants to apply a discount to loan interest rates for customers above 60 years old.

* + **Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

**ANS**

BEGIN

FOR cust IN (

SELECT c.customer\_id, l.loan\_id, l.interest\_rate

FROM customers952 c

JOIN loans952 l ON c.customer\_id = l.customer\_id

WHERE c.age > 60

) LOOP

UPDATE loans952

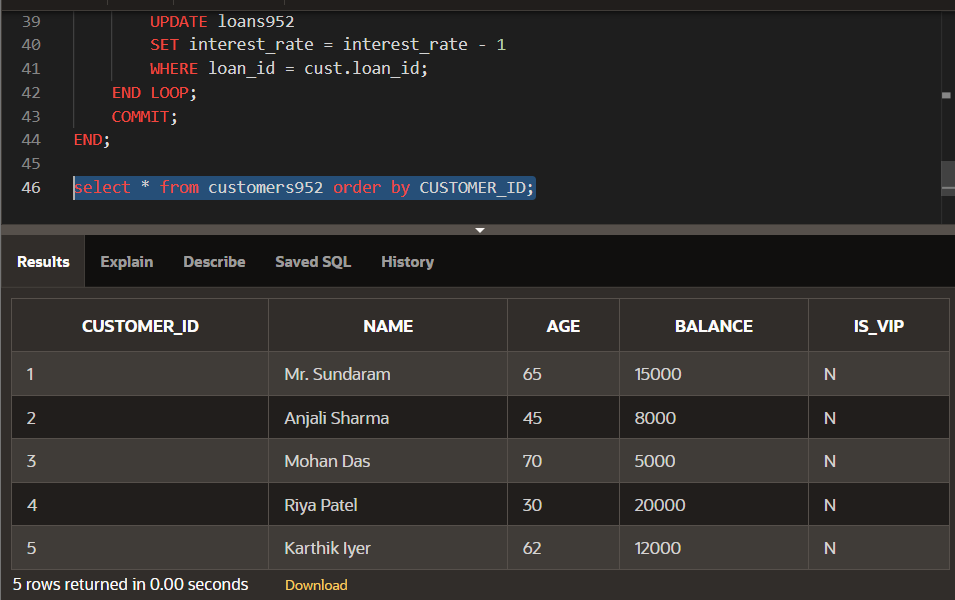
SET interest\_rate = interest\_rate - 1

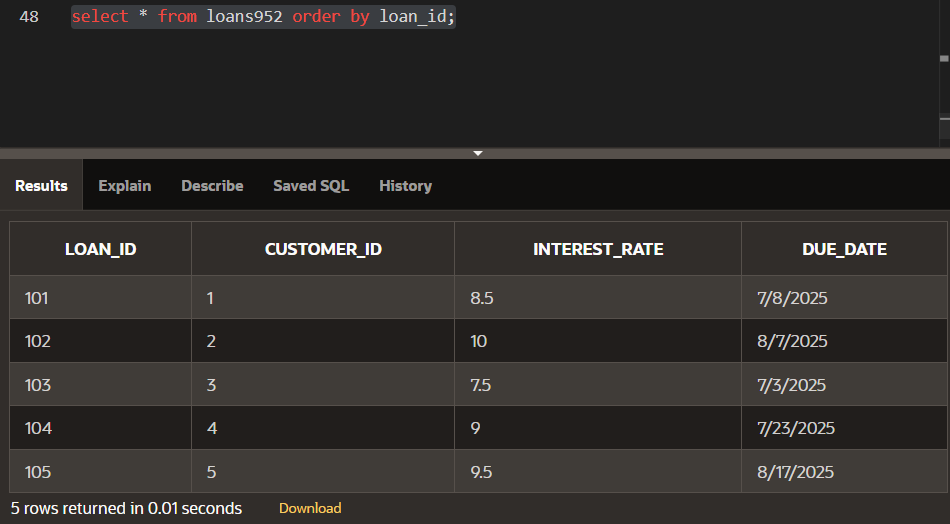
WHERE loan\_id = cust.loan\_id;

END LOOP;

COMMIT;

END;

****

****

**Scenario 2:** A customer can be promoted to VIP status based on their balance.

* + **Question:** Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

**ANS**

BEGIN

FOR cust IN (

SELECT customer\_id

FROM customers952

WHERE balance > 10000

) LOOP

UPDATE customers952

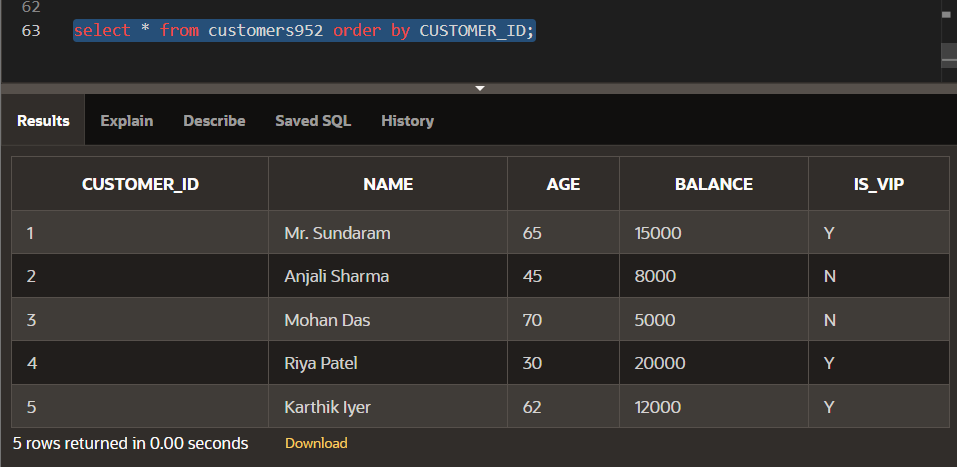
SET is\_vip = 'Y'

WHERE customer\_id = cust.customer\_id;

END LOOP;

COMMIT;

END;

****

**Scenario 3:** The bank wants to send reminders to customers whose loans are due within the next 30 days.

* + **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.

**Ans:**

BEGIN

FOR loan\_rec IN (

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

FROM loans952 l

JOIN customers952 c ON l.customer\_id = c.customer\_id

WHERE l.due\_date <= SYSDATE + 30

) LOOP

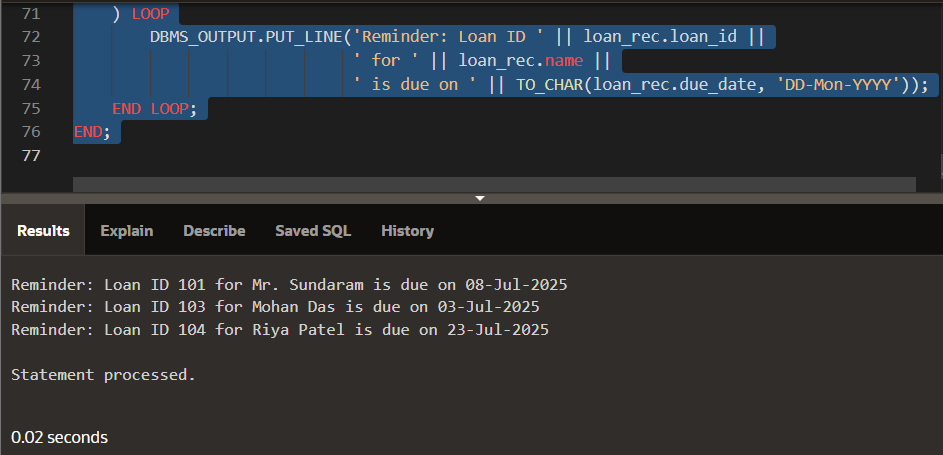
DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan\_rec.loan\_id ||

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

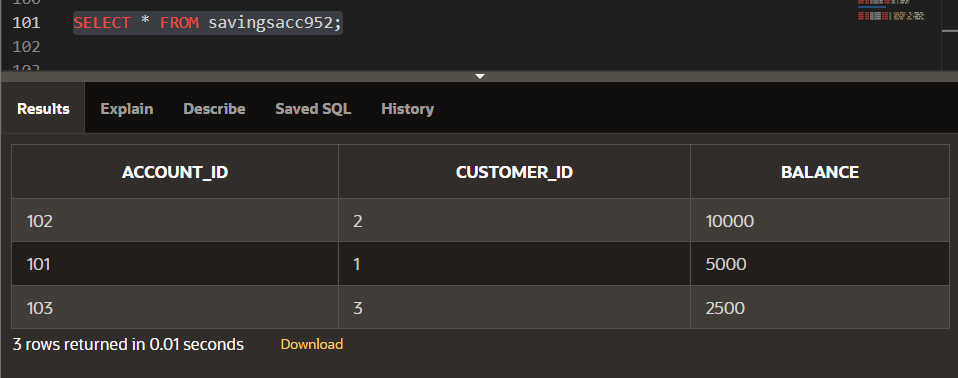
' is due on ' || TO\_CHAR(loan\_rec.due\_date, 'DD-Mon-YYYY'));

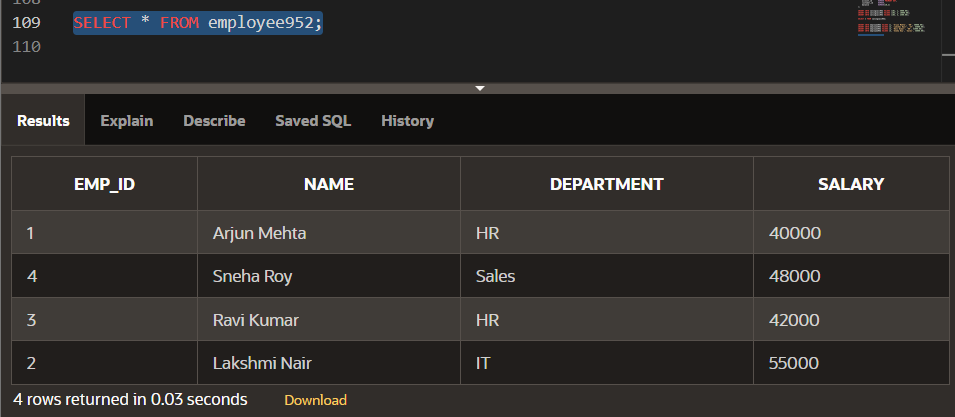
END LOOP;

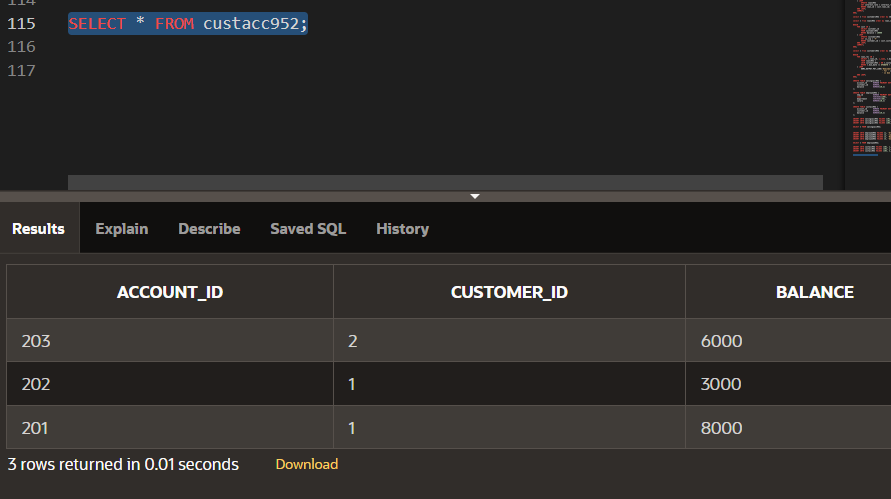
END;

****

**Exercise 3: Stored Procedures**







**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

* + **Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest952 AS

BEGIN

FOR acc IN (SELECT account\_id, balance FROM savingsacc952) LOOP

UPDATE savingsacc952

SET balance = balance + (balance \* 0.01)

WHERE account\_id = acc.account\_id;

END LOOP;

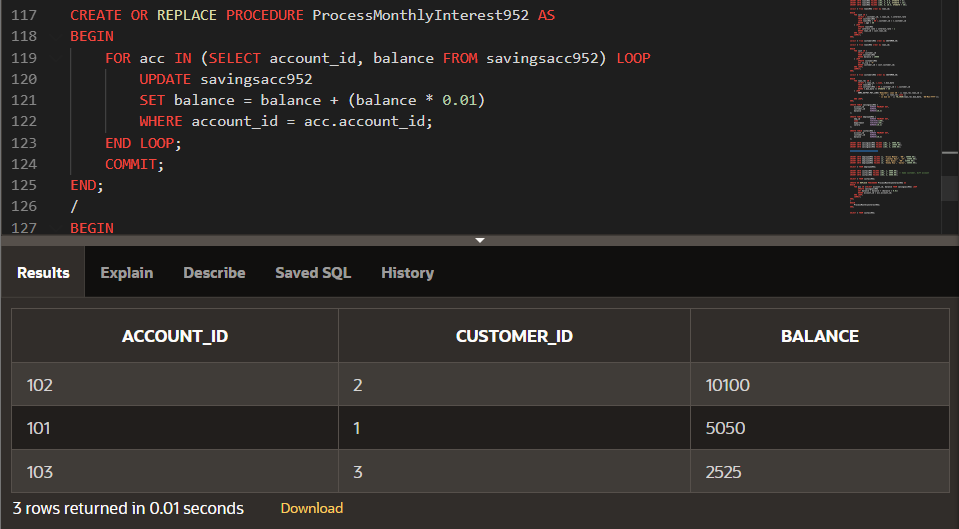
COMMIT;

END;

BEGIN

ProcessMonthlyInterest952;

END;



**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

* + **Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus952(

p\_department IN VARCHAR2,

p\_bonus\_percent IN NUMBER

) AS

BEGIN

FOR emp IN (

SELECT emp\_id, salary FROM employee952

WHERE department = p\_department

) LOOP

UPDATE employee952

SET salary = salary + (salary \* (p\_bonus\_percent / 100))

WHERE emp\_id = emp.emp\_id;

END LOOP;

COMMIT;

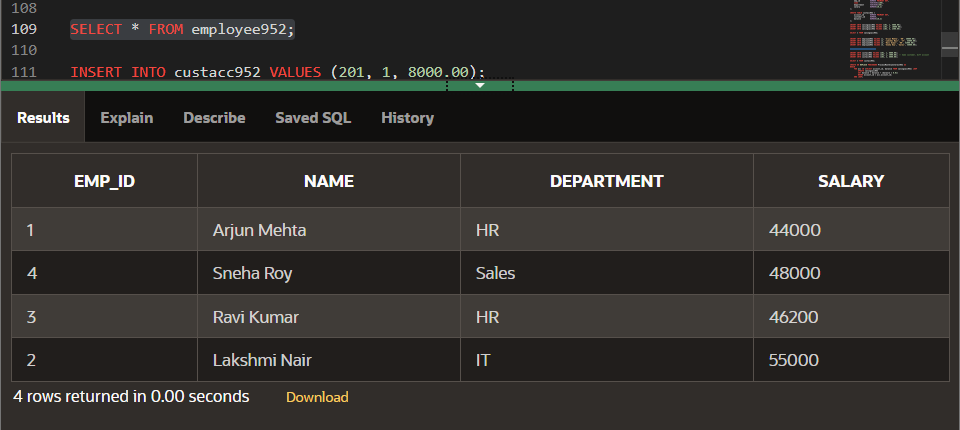
END;

/

BEGIN

UpdateEmployeeBonus952('HR', 10); -

END;



**Scenario 3:** Customers should be able to transfer funds between their accounts.

* + **Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

CREATE OR REPLACE PROCEDURE TransferFunds952(

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

SELECT balance INTO v\_balance

FROM custacc952

WHERE account\_id = p\_from\_account\_id;

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account');

ELSE

UPDATE custacc952

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_account\_id;

UPDATE custacc952

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_account\_id;

COMMIT;

END IF;

END;

BEGIN

TransferFunds952(201, 202, 2000);

END;

