Week 2:

**Scenario 1:**

**Code:**

-- Enable output

SET SERVEROUTPUT ON;

-- Drop tables if they already exist (optional, for reruns)

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE loans\_new';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE customers\_new';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

-- Step 1: Create tables

CREATE TABLE customers\_new (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER(10, 2),

IsVIP VARCHAR2(5) DEFAULT 'FALSE'

);

CREATE TABLE loans\_new (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER REFERENCES customers\_new(customer\_id),

interest\_rate NUMBER(5, 2),

due\_date DATE

);

-- Step 2: Insert sample data

INSERT INTO customers\_new VALUES (1, 'John Doe', 65, 10000, 'FALSE');

INSERT INTO customers\_new VALUES (2, 'Jane Smith', 45, 15000, 'FALSE');

INSERT INTO loans\_new VALUES (101, 1, 7.5, TO\_DATE('2025-12-31', 'YYYY-MM-DD'));

INSERT INTO loans\_new VALUES (102, 2, 8.0, TO\_DATE('2025-11-30', 'YYYY-MM-DD'));

COMMIT;

-- Step 3: PL/SQL block to apply 1% interest discount for customers above 60

BEGIN

FOR rec IN (SELECT customer\_id FROM customers\_new WHERE age > 60) LOOP

UPDATE loans\_new

SET interest\_rate = interest\_rate - (interest\_rate \* 0.01)

WHERE customer\_id = rec.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount for customer\_id: ' || rec.customer\_id);

END LOOP;

COMMIT;

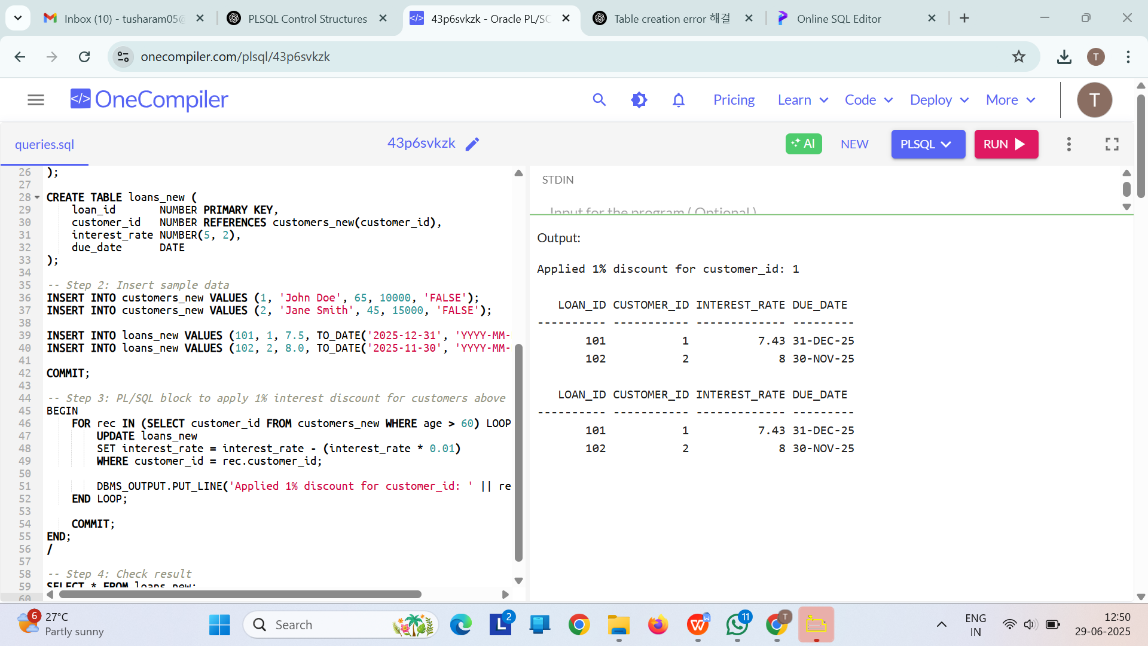
END;

/

-- Step 4: Check result

SELECT \* FROM loans\_new;

OUTPUT-



**Scenario 2:**

**Code:**

-- Enable output

SET SERVEROUTPUT ON;

-- Optional: Drop tables if they exist

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE loans\_new';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE customers\_new';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

-- Step 1: Create tables

CREATE TABLE customers\_new (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER(10, 2),

IsVIP VARCHAR2(5) DEFAULT 'FALSE'

);

CREATE TABLE loans\_new (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER REFERENCES customers\_new(customer\_id),

interest\_rate NUMBER(5, 2),

due\_date DATE

);

-- Step 2: Insert sample data

INSERT INTO customers\_new VALUES (1, 'John Doe', 65, 10000, 'FALSE');

INSERT INTO customers\_new VALUES (2, 'Jane Smith', 45, 15000, 'FALSE');

INSERT INTO customers\_new VALUES (3, 'Alice Brown', 70, 12000, 'FALSE');

INSERT INTO loans\_new VALUES (101, 1, 7.5, TO\_DATE('2025-12-31', 'YYYY-MM-DD'));

INSERT INTO loans\_new VALUES (102, 2, 8.0, TO\_DATE('2025-11-30', 'YYYY-MM-DD'));

INSERT INTO loans\_new VALUES (103, 3, 9.0, TO\_DATE('2025-10-15', 'YYYY-MM-DD'));

COMMIT;

-- Step 3: Promote to VIP where balance > 10000

BEGIN

FOR rec IN (SELECT customer\_id, balance FROM customers\_new WHERE balance > 10000) LOOP

UPDATE customers\_new

SET IsVIP = 'TRUE'

WHERE customer\_id = rec.customer\_id;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || rec.customer\_id || ' promoted to VIP (Balance: $' || rec.balance || ')');

END LOOP;

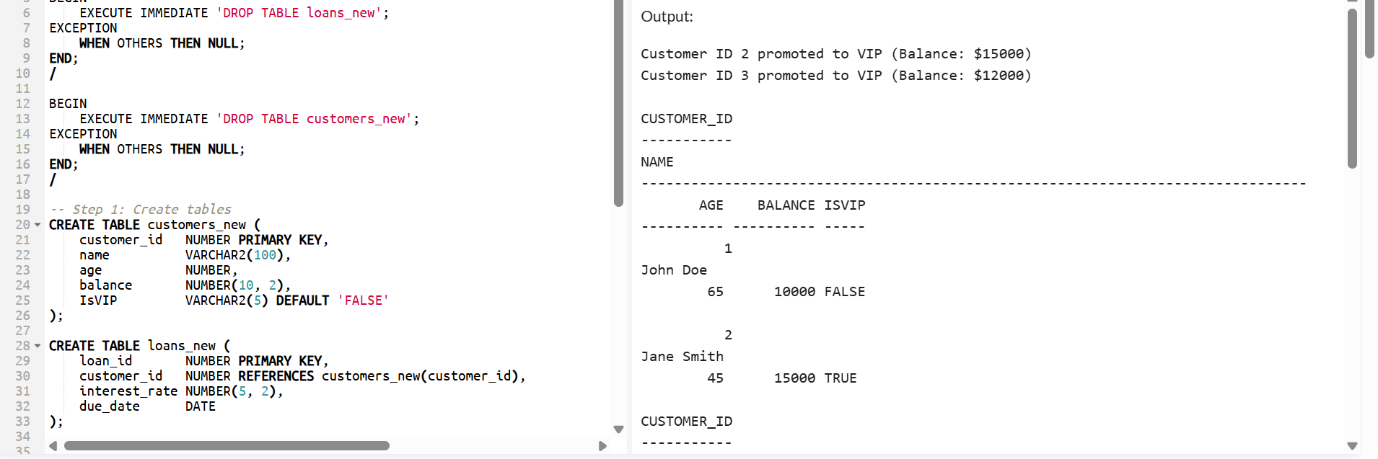
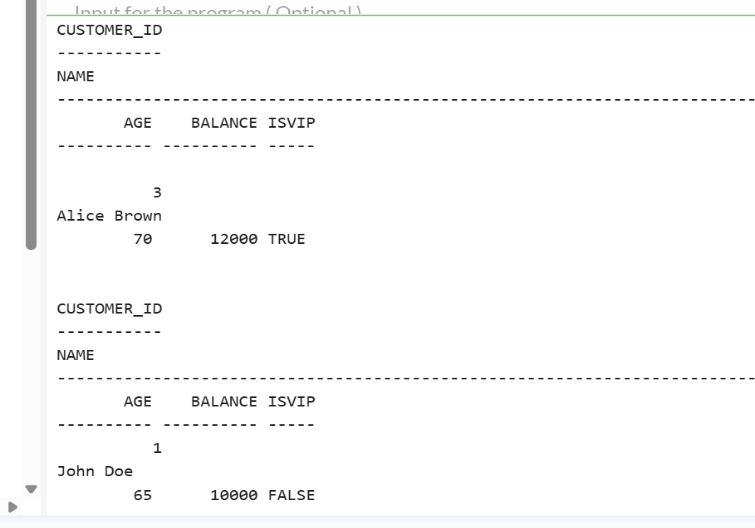
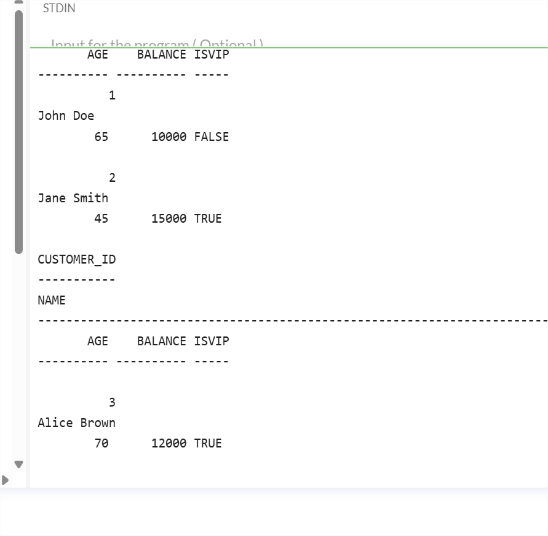
COMMIT;

END;

/

-- Step 4: View updated customers

SELECT \* FROM customers\_new;

OUTPUT  

**Scenario 3:**

**Code:**

-- Enable DBMS output

SET SERVEROUTPUT ON;

-- Optional: Drop existing tables if rerunning

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE loans\_new';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE customers\_new';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

-- Step 1: Create required tables

CREATE TABLE customers\_new (

customer\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

age NUMBER,

balance NUMBER(10, 2),

IsVIP VARCHAR2(5) DEFAULT 'FALSE'

);

CREATE TABLE loans\_new (

loan\_id NUMBER PRIMARY KEY,

customer\_id NUMBER REFERENCES customers\_new(customer\_id),

interest\_rate NUMBER(5, 2),

due\_date DATE

);

-- Step 2: Insert sample data

INSERT INTO customers\_new VALUES (1, 'John Doe', 65, 10000, 'FALSE');

INSERT INTO customers\_new VALUES (2, 'Jane Smith', 45, 15000, 'FALSE');

INSERT INTO customers\_new VALUES (3, 'Alice Brown', 70, 12000, 'FALSE');

-- Loan due in 15 days

INSERT INTO loans\_new VALUES (101, 1, 7.5, SYSDATE + 15);

-- Loan already overdue (won't be included)

INSERT INTO loans\_new VALUES (102, 2, 8.0, SYSDATE - 10);

-- Loan due in 45 days (won’t be included)

INSERT INTO loans\_new VALUES (103, 3, 9.0, SYSDATE + 45);

COMMIT;

-- Step 3: PL/SQL Block – Send reminders for loans due within 30 days

BEGIN

FOR rec IN (

SELECT c.name, l.due\_date

FROM customers\_new c

JOIN loans\_new l ON c.customer\_id = l.customer\_id

WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Dear ' || rec.name ||

', your loan is due on ' || TO\_CHAR(rec.due\_date, 'DD-MON-YYYY') || '. Please make the payment on time.');

END LOOP;

END;

/

-- Optional: View loans

SELECT \* FROM loans\_new;

