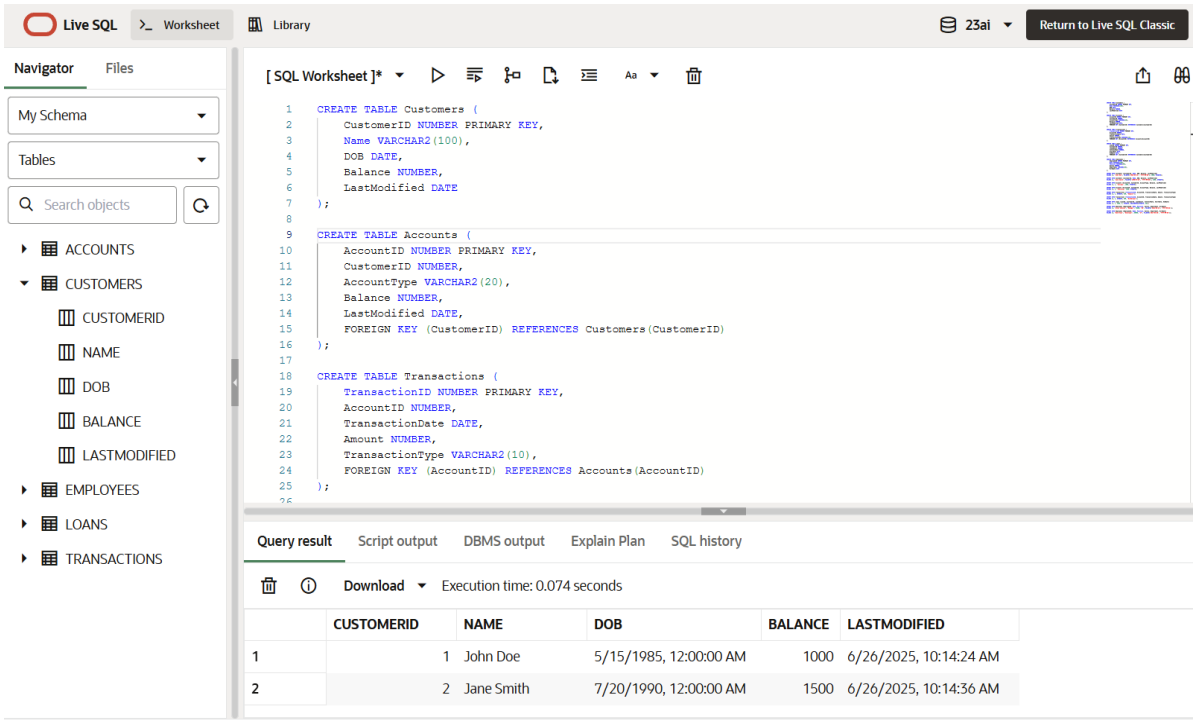
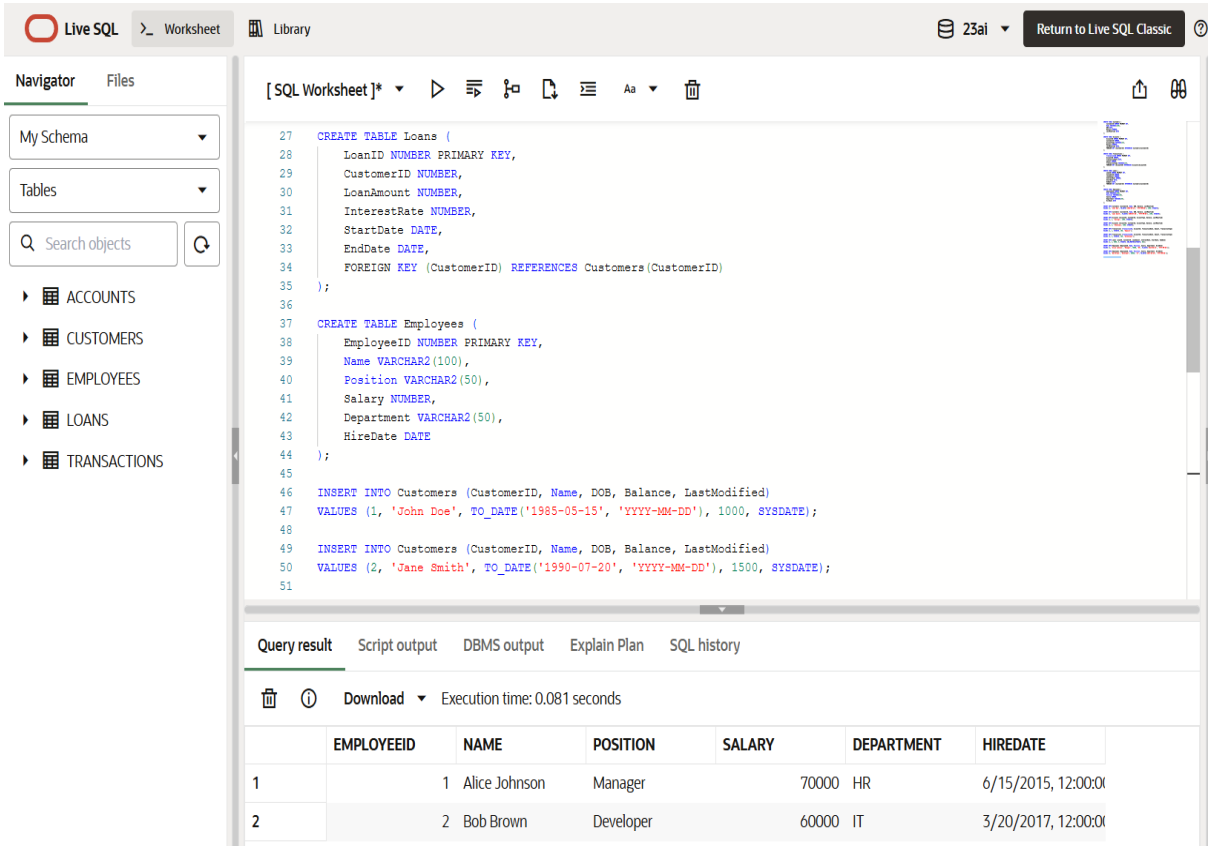
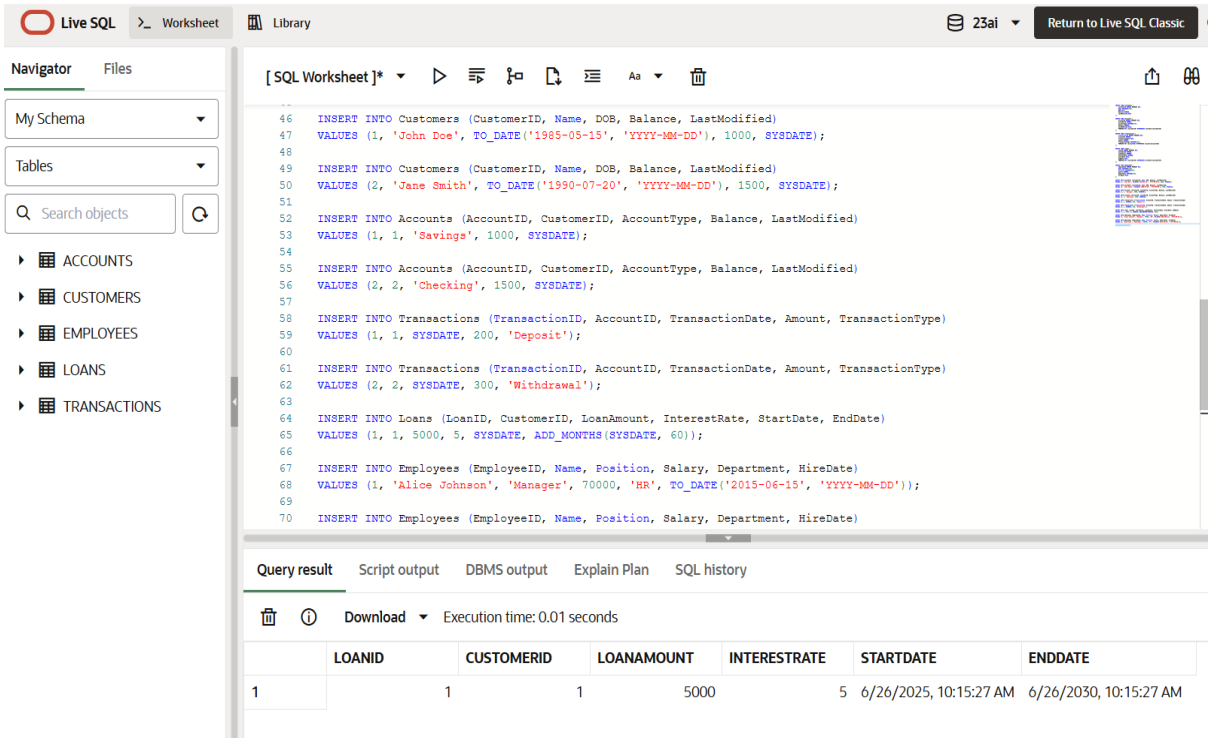
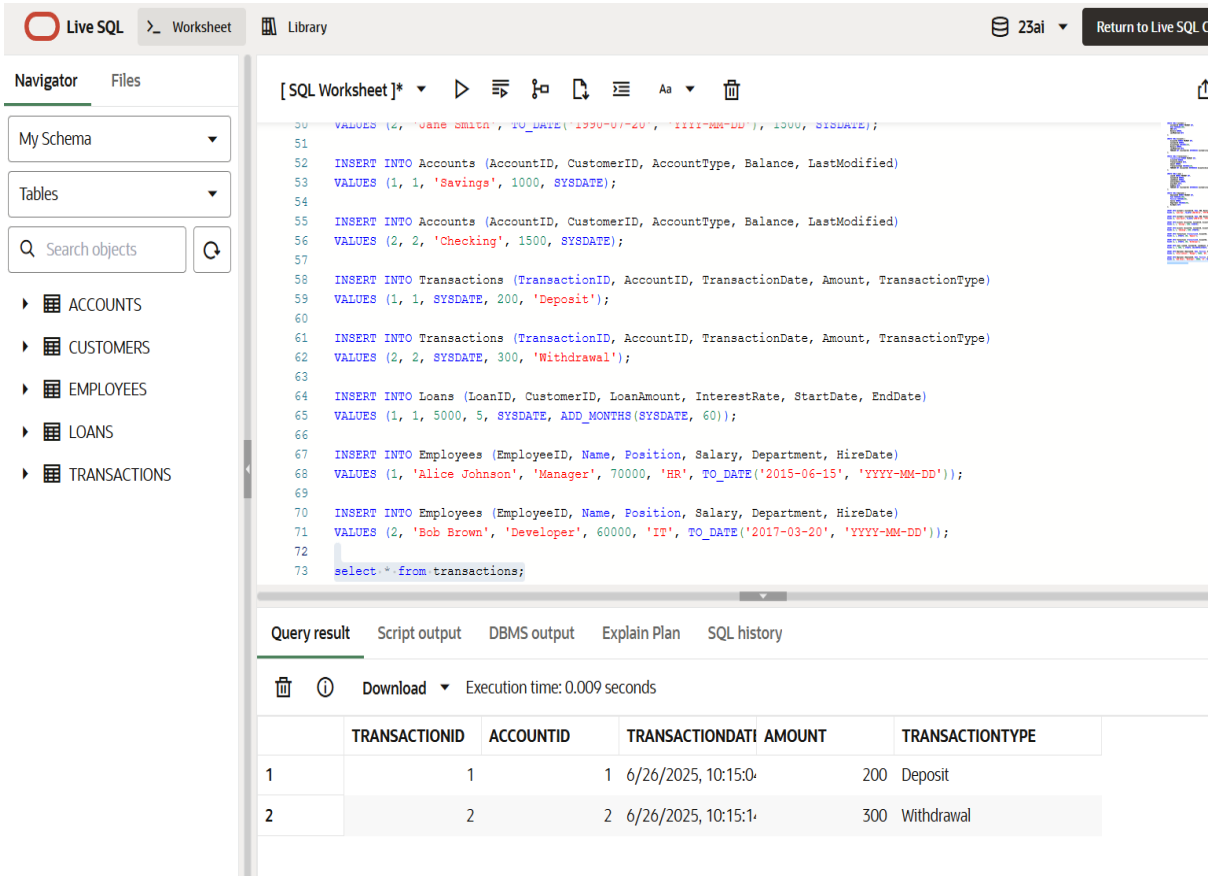
Week-2


Live SQL
Worksheet
Library
23ai
Return to Live SQL Classic
Navigator
Files
My Schema
Tables
Search objects
ACCOUNTS
CUSTOMERS
CUSTOMERID
NAME
DOB
BALANCE
LASTMODIFIED
EMPLOYEES
LOANS
TRANSACTIONS
[SQL Worksheet]*
CREATE TABLE Customers (
CustomerID NUMBER PRIMARY KEY,
Name VARCHAR2(100),
DOB DATE,
Balance NUMBER,
LastModified DATE
);
CREATE TABLE Accounts (
AccountID NUMBER PRIMARY KEY,
CustomerID NUMBER,
AccountType VARCHAR2(20),
Balance NUMBER,
LastModified DATE,
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
CREATE TABLE Transactions (
TransactionID NUMBER PRIMARY KEY,
AccountID NUMBER,
TransactionDate DATE,
Amount NUMBER,
TransactionType VARCHAR2(10),
FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)
);
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.074 seconds
CUSTOMERID
NAME
DOB
BALANCE
LASTMODIFIED
1 1 John Doe 5/15/1985, 12:00:00 AM 1000 6/26/2025, 10:14:24 AM
2 2 Jane Smith 7/20/1990, 12:00:00 AM 1500 6/26/2025, 10:14:36 AM


Live SQL
Worksheet
Library
23ai
Return to Live SQL Classic
Navigator
Files
My Schema
Tables
Search objects
ACCOUNTS
CUSTOMERS
EMPLOYEES
LOANS
TRANSACTIONS
[SQL Worksheet]*
CREATE TABLE Loans (
LoanID NUMBER PRIMARY KEY,
CustomerID NUMBER,
LoanAmount NUMBER,
InterestRate NUMBER,
StartDate DATE,
EndDate DATE,
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
CREATE TABLE Employees (
EmployeeID NUMBER PRIMARY KEY,
Name VARCHAR2(100),
Position VARCHAR2(50),
Salary NUMBER,
Department VARCHAR2(50),
HireDate DATE
);
INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)
VALUES (1, 'John Doe', TO_DATE('1985-05-15', 'YYYY-MM-DD'), 1000, SYSDATE);
INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)
VALUES (2, 'Jane Smith', TO_DATE('1990-07-20', 'YYYY-MM-DD'), 1500, SYSDATE);
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.081 seconds
EMPLOYEEID
NAME
POSITION
SALARY
DEPARTMENT
HIREDATE
1 1 Alice Johnson Manager 70000 HR 6/15/2015, 12:00:0
2 2 Bob Brown Developer 60000 IT 3/20/2017, 12:00:0

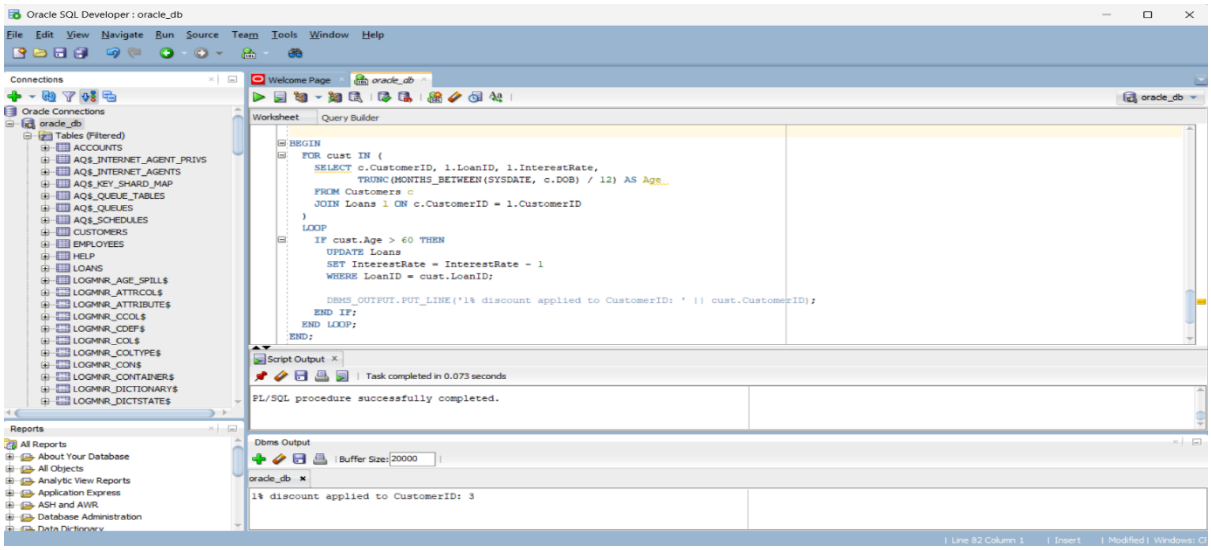

Live SQL
Worksheet
Library
23ai
Return to Live SQL Classic
Navigator
Files
My Schema
Tables
Search objects
ACCOUNTS
CUSTOMERS
EMPLOYEES
LOANS
TRANSACTIONS
[SQL Worksheet]*
INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)
VALUES (1, 'John Doe', TO_DATE('1985-05-15', 'YYYY-MM-DD'), 1000, SYSDATE);
INSERT INTO Customers (CustomerID, Name, DOB, Balance, LastModified)
VALUES (2, 'Jane Smith', TO_DATE('1990-07-20', 'YYYY-MM-DD'), 1500, SYSDATE);
INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)
VALUES (1, 1, 'Savings', 1000, SYSDATE);
INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)
VALUES (2, 2, 'Checking', 1500, SYSDATE);
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
VALUES (1, 1, SYSDATE, 200, 'Deposit');
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
VALUES (2, 2, SYSDATE, 300, 'Withdrawal');
INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)
VALUES (1, 1, 5000, 5, SYSDATE, ADD_MONTHS(SYSDATE, 60));
INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)
VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', TO_DATE('2015-06-15', 'YYYY-MM-DD'));
INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.01 seconds
LOANID
CUSTOMERID
LOANAMOUNT
INTERESTRATE
STARTDATE
ENDDATE
1
1
1
5000
5
6/26/2025, 10:15:27 AM
6/26/2030, 10:15:27 AM


Live SQL
Worksheet
Library
23ai
Navigator
Files
My Schema
Tables
Search objects
ACCOUNTS
CUSTOMERS
EMPLOYEES
LOANS
TRANSACTIONS
[SQL Worksheet]*
INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)
VALUES (1, 1, 'Savings', 1000, SYSDATE);
INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified)
VALUES (2, 2, 'Checking', 1500, SYSDATE);
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
VALUES (1, 1, SYSDATE, 200, 'Deposit');
INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
VALUES (2, 2, SYSDATE, 300, 'Withdrawal');
INSERT INTO Loans (LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate)
VALUES (1, 1, 5000, 5, SYSDATE, ADD_MONTHS(SYSDATE, 60));
INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)
VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', TO_DATE('2015-06-15', 'YYYY-MM-DD'));
INSERT INTO Employees (EmployeeID, Name, Position, Salary, Department, HireDate)
VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', TO_DATE('2017-03-20', 'YYYY-MM-DD'));
select * from transactions;
Query result
Script output
DBMS output
Explain Plan
SQL history
Download
Execution time: 0.009 seconds
TRANSACTIONID
ACCOUNTID
TRANSACTIONDATI
AMOUNT
TRANSACTIONTYPE
1
1
1
6/26/2025, 10:15:0
200
Deposit
2
2
2
6/26/2025, 10:15:1
300
Withdrawal

Exercise 1: Control Structures

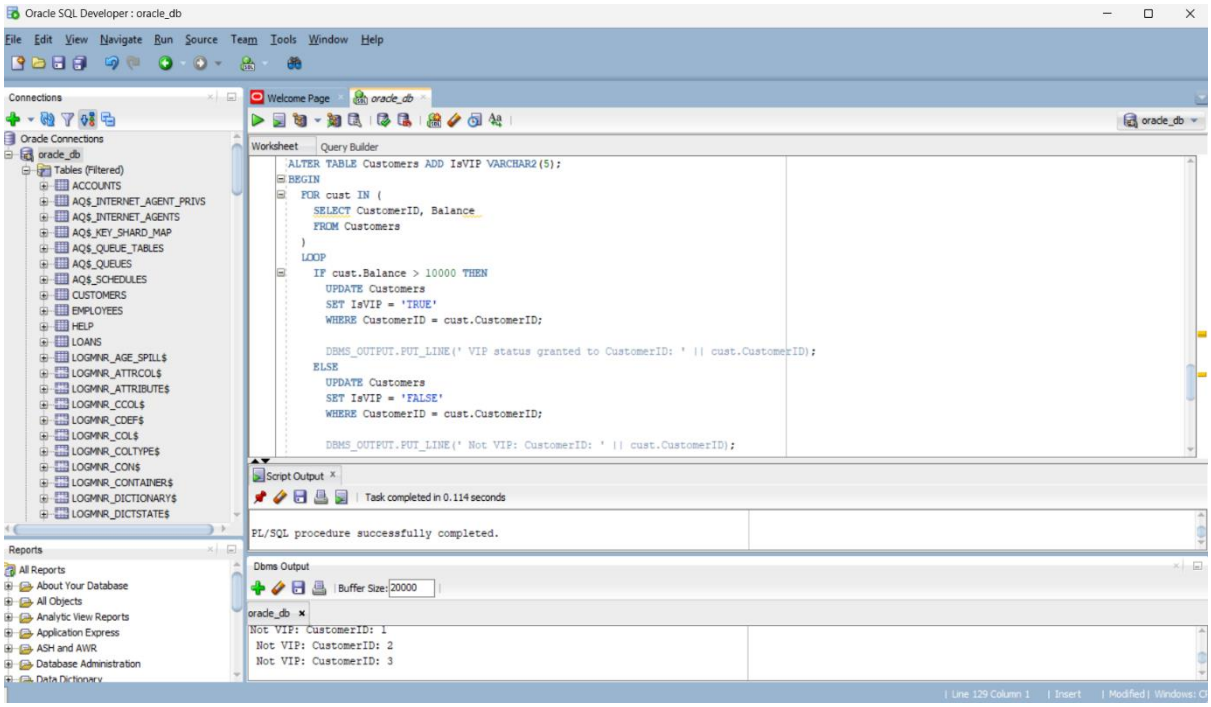
Scenario 1: The bank wants to apply a discount to loan interest rates for customers above 60 years old.

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.



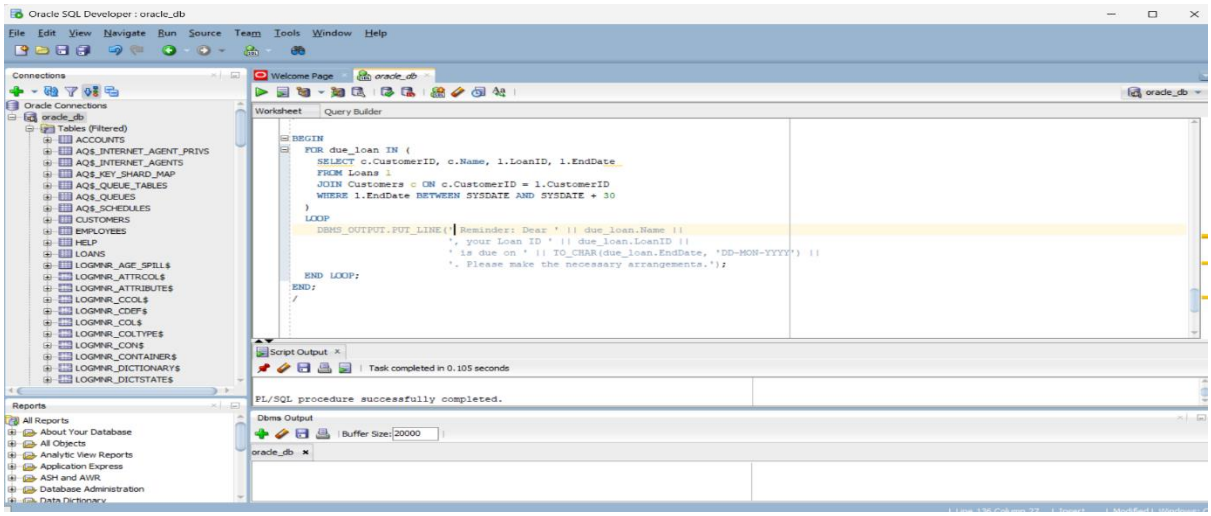
Scenario 2: A customer can be promoted to VIP status based on their balance.

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.



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

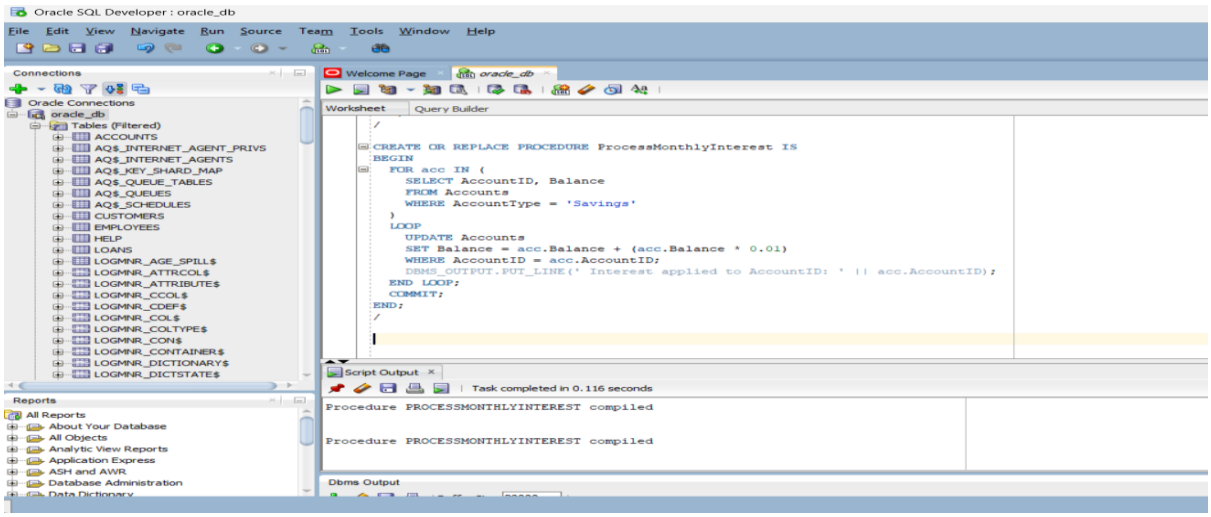
Question: Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.



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.



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.