

Name: SASHMITHA R

Reg No: 727722EUCD042

Exercise 1: Control Structures

Table Creation: Customers

The screenshot shows the Live SQL interface with a worksheet containing the following SQL code:

```
1 CREATE TABLE Customers (  
2   CustomerID    NUMBER PRIMARY KEY,  
3   Name          VARCHAR2(100),  
4   Age           NUMBER,  
5   Balance       NUMBER(10, 2),  
6   IsVIP         VARCHAR2(5)  
7 );  
8
```

The interface also shows a sidebar with 'My Schema' and 'Tables' sections, and a 'CUSTOMERS' table listed. Below the code editor, the 'Script output' tab is active, displaying the execution result:

```
SQL> CREATE TABLE Customers (  
CustomerID    NUMBER PRIMARY KEY,  
Name          VARCHAR2(100),  
Age           NUMBER,...  
Show more...
```

Below the script output, it states: 'Table CUSTOMERS created.' and 'Elapsed: 00:00:00.019'.

Inserting values in Customers table

The screenshot shows the Live SQL interface with a worksheet containing the following SQL code:

```
1 INSERT INTO Customers (CustomerID, Name, Age, Balance, IsVIP) VALUES (1, 'Sam', 65, 15000, 'FALSE');  
2 INSERT INTO Customers (CustomerID, Name, Age, Balance, IsVIP) VALUES (2, 'Ram', 45, 9000, 'FALSE');  
3 INSERT INTO Customers (CustomerID, Name, Age, Balance, IsVIP) VALUES (3, 'Ravi', 70, 12000, 'FALSE');  
4  
5 select * from Customers;
```

The interface also shows a sidebar with 'My Schema' and 'Tables' sections, and a 'CUSTOMERS' table listed. Below the code editor, the 'Query result' tab is active, displaying the execution result:

```
Execution time: 0.081 seconds
```

	CUSTOMERID	NAME	AGE	BALANCE	ISVIP
1	1	Sam	65	15000	FALSE
2	2	Ram	45	9000	FALSE
3	3	Ravi	70	12000	FALSE

Table Creation: Loans

NavigatorFiles

My Schema

Tables

Search objects

CUSTOMERS

LOANS

[SQL Worksheet]*

1

2

3

4

5

6

7

8

CREATE TABLE Loans (

LoanID

NUMBER PRIMARY KEY,

CustomerID

NUMBER,

InterestRate

NUMBER(5, 2),

DueDate

DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

Query resultScript outputDBMS outputExplain PlanSQL history

SQL> CREATE TABLE Loans (

LoanID

NUMBER PRIMARY KEY,

CustomerID

NUMBER,

InterestRate

NUMBER(5, 2),...

Show more...

Table LOANS created.

Elapsed: 00:00:00.014

Inserting values in Loans table

NavigatorFiles

My Schema

Tables

Search objects

CUSTOMERS

LOANS

[SQL Worksheet]*

1

2

3

4

5

INSERT INTO Loans (LoanID, CustomerID, InterestRate, DueDate) VALUES (101, 1, 8.5, SYSDATE + 10);

INSERT INTO Loans (LoanID, CustomerID, InterestRate, DueDate) VALUES (102, 2, 9.0, SYSDATE + 40);

INSERT INTO Loans (LoanID, CustomerID, InterestRate, DueDate) VALUES (103, 3, 7.5, SYSDATE + 25);

select * from Loans;

Query resultScript outputDBMS outputExplain PlanSQL history

Download

Execution time: 0.089 seconds

	LOANID	CUSTOMERID	INTERESTRATE	DUEDATE
1	101	1	8.5	7/9/2025, 4:48:23 A
2	102	2	9	8/8/2025, 4:48:23 A
3	103	3	7.5	7/24/2025, 4:48:23

Scenario 1: Apply 1% discount to loan interest rates for customers above 60 years old

```
BEGIN

FOR rec IN (SELECT c.CustomerID, l.LoanID, l.InterestRate
            FROM Customers c
            JOIN Loans l ON c.CustomerID = l.CustomerID
            WHERE c.Age > 60)



LOOP
    UPDATE Loans
    SET InterestRate = InterestRate - 1
    WHERE LoanID = rec.LoanID;
END LOOP;

COMMIT;

END;

SELECT * FROM Loans;
```

Output :



Query result Script output DBMS output Explain Plan SQL history					
  Download ▾ Execution time: 0.005 seconds					
	LOANID	CUSTOMERID	INTERESTRATE	DUE DATE	
1	101	1	7.5	7/9/2025, 4:48:23 A	
2	102	2	9	8/8/2025, 4:48:23 A	
3	103	3	6.5	7/24/2025, 4:48:23	

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

```
BEGIN  
  
  FOR rec IN (SELECT CustomerID  
              FROM Customers  
              WHERE Balance > 10000)  
  
  LOOP  
    UPDATE Customers  
    SET IsVIP = 'TRUE'  
    WHERE CustomerID = rec.CustomerID;  
  
  END LOOP;  
  
  COMMIT;  
  
END;
```

```
SELECT * FROM Customers;
```

Output :

Query result Script output DBMS output Explain Plan SQL history					
  Download ▾ Execution time: 0.001 seconds					
	CUSTOMERID	NAME	AGE	BALANCE	ISVIP
1	1	Sam	65	15000	TRUE
2	2	Ram	45	9000	FALSE
3	3	Ravi	70	12000	TRUE

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

```
DECLARE
v_due_date Loans.DueDate%TYPE;
v_name Customers.Name%TYPE;
BEGIN
FOR rec IN (
    SELECT c.Name, l.DueDate
    FROM Loans l
    JOIN Customers c ON c.CustomerID = l.CustomerID
    WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30
)
LOOP
    DBMS_OUTPUT.PUT_LINE('Reminder: Loan for ' || rec.Name || ' is due on ' ||
TO_CHAR(rec.DueDate, 'DD-MON-YYYY'));
END LOOP;
END;
```

Output :

Query result

Script output

DBMS output

Explain Plan

SQL history



Reminder: Loan for Sam is due on 09-JUL-2025
Reminder: Loan for Ravi is due on 24-JUL-2025

Exercise 3: Stored Procedures

Table Creation: SavingsAccounts

The screenshot shows the SQL Developer interface. On the left, the Navigator pane shows 'My Schema' and 'Tables'. The main editor shows a SQL script to create a table named 'SavingsAccounts' with columns 'AccountID' (NUMBER PRIMARY KEY), 'CustomerName' (VARCHAR2(100)), and 'Balance' (NUMBER(10, 2)). The script is executed, and the 'Script output' pane shows the successful execution of the CREATE TABLE statement. The 'Query result' pane shows the table structure.

```
1 CREATE TABLE SavingsAccounts (  
2   AccountID    NUMBER PRIMARY KEY,  
3   CustomerName VARCHAR2(100),  
4   Balance      NUMBER(10, 2)  
5 );
```

Query result Script output DBMS output Explain Plan SQL history

SQL> CREATE TABLE SavingsAccounts (
AccountID NUMBER PRIMARY KEY,
CustomerName VARCHAR2(100),
Balance NUMBER(10, 2)...
Show more...

Table SAVINGSACCOUNTS created.

Elapsed: 00:00:00.013

Inserting values in SavingsAccounts table

The screenshot shows the SQL Developer interface. On the left, the Navigator pane shows 'My Schema' and 'Tables'. The main editor shows a SQL script to insert four rows into the 'SavingsAccounts' table. The script is executed, and the 'Script output' pane shows the successful execution of the INSERT statements. The 'Query result' pane shows the resulting table data.

```
1 INSERT INTO SavingsAccounts VALUES (1, 'Alice', 1000);  
2 INSERT INTO SavingsAccounts VALUES (2, 'Bob', 2000);  
3 INSERT INTO SavingsAccounts VALUES (3, 'Charlie', 3000);  
4 INSERT INTO SavingsAccounts VALUES (4, 'Diana', 4000);  
5  
6 select * from SavingsAccounts;
```

Query result Script output DBMS output Explain Plan SQL history

Download Execution time: 0.081 seconds

	ACCOUNTID	CUSTOMERNAME	BALANCE
1	1	Alice	1000
2	2	Bob	2000
3	3	Charlie	3000
4	4	Diana	4000

Table Creation: Employees

NavigatorFiles

My Schema

Tables

EMPLOYEES

SAVINGSACCOUNTS

[SQL Worksheet]*

1

2

3

4

5

6

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Department VARCHAR2(50),

Salary NUMBER(10, 2)

)

;

Query resultScript outputDBMS outputExplain PlanSQL history

SQL> CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

Department VARCHAR2(50),...

Show more...

Table EMPLOYEES created.

Elapsed: 00:00:00.018

Inserting values in Employees table

NavigatorFiles

My Schema

Tables

EMPLOYEES

SAVINGSACCOUNTS

[SQL Worksheet]*

1

2

3

4

5

6

INSERT INTO Employees VALUES (101, 'John', 'HR', 5000);

INSERT INTO Employees VALUES (102, 'Sara', 'IT', 6000);

INSERT INTO Employees VALUES (103, 'Mike', 'Finance', 7000);

INSERT INTO Employees VALUES (104, 'Lisa', 'IT', 5500);

select * from Employees;

Query resultScript outputDBMS outputExplain PlanSQL history

Download Execution time: 0.009 seconds

	EMPLOYEEID	NAME	DEPARTMENT	SALARY
1	101	John	HR	5000
2	102	Sara	IT	6000
3	103	Mike	Finance	7000
4	104	Lisa	IT	5500

Table Creation: Accounts

NavigatorFiles

My Schema

Tables

Q Search objects

ACCOUNTS

EMPLOYEES

SAVINGSACCOUNTS

[SQL Worksheet]*

Aa

1 CREATE TABLE Accounts (

2 AccountID NUMBER PRIMARY KEY,

3 CustomerName VARCHAR2(100),

4 Balance NUMBER(10, 2)

5);

Query resultScript outputDBMS outputExplain PlanSQL history

SQL> CREATE TABLE Accounts (
AccountID NUMBER PRIMARY KEY,
CustomerName VARCHAR2(100),
Balance NUMBER(10, 2)...
Show more...

Table ACCOUNTS created.

Elapsed: 00:00:00.020

Inserting values in Accounts table

NavigatorFiles

My Schema

Tables

Q Search objects

ACCOUNTS

EMPLOYEES

SAVINGSACCOUNTS

[SQL Worksheet]*

Aa

1 INSERT INTO Accounts VALUES (201, 'Alice', 3000);

2 INSERT INTO Accounts VALUES (202, 'Bob', 2500);

3 INSERT INTO Accounts VALUES (203, 'Charlie', 1500);

4 INSERT INTO Accounts VALUES (204, 'Diana', 500);

5

6 select * from Accounts;

Query resultScript outputDBMS outputExplain PlanSQL history

i







Download

Execution time: 0.01 seconds

	ACCOUNTID	CUSTOMERNAME	BALANCE
1	201	Alice	3000
2	202	Bob	2500
3	203	Charlie	1500
4	204	Diana	500




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

```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS
BEGIN
    FOR rec IN (SELECT AccountID, Balance FROM SavingsAccounts) LOOP
        UPDATE SavingsAccounts
        SET Balance = Balance + (Balance * 0.01)
        WHERE AccountID = rec.AccountID;
    END LOOP;
    COMMIT;
END;
```

[SQL Worksheet]*      Aa 

```
1 BEGIN
2   ProcessMonthlyInterest;
3 END;
4
5 select * from SavingsAccounts;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

  Download  Execution time: 0.004 seconds

	ACCOUNTID	CUSTOMERNAME	BALANCE
1	1	Alice	1020.1
2	2	Bob	2040.2
3	3	Charlie	3060.3
4	4	Diana	4080.4


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

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (  
    p_department IN VARCHAR2,  
    p_bonus_percent IN NUMBER  
) IS  
BEGIN  
    UPDATE Employees  
    SET Salary = Salary + (Salary * p_bonus_percent / 100)  
    WHERE Department = p_department;  
    COMMIT;  
END;
```

[SQL Worksheet]*      Aa  

```
1 BEGIN  
2 | UpdateEmployeeBonus('IT', 10);  
3 END;  
4  
5 select * from Employees;
```







Query result Script output DBMS output Explain Plan SQL history

  Download  Execution time: 0.001 seconds

	EMPLOYEEID	NAME	DEPARTMENT	SALARY
1	101	John	HR	5000
2	102	Sara	IT	6600
3	103	Mike	Finance	7000
4	104	Lisa	IT	6050




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

```
CREATE OR REPLACE PROCEDURE TransferFunds (  
    p_from_account IN NUMBER, p_to_account IN NUMBER,  
    p_amount IN NUMBER ) IS v_balance NUMBER;  
  
BEGIN  
  
    SELECT Balance INTO v_balance FROM Accounts WHERE AccountID = p_from_account;  
  
    IF v_balance < p_amount THEN  
        RAISE_APPLICATION_ERROR(-20001, 'Insufficient balance in source account');  
    END IF;  
  
    UPDATE Accounts SET Balance = Balance - p_amount WHERE AccountID = p_from_account;  
  
    UPDATE Accounts SET Balance = Balance + p_amount WHERE AccountID = p_to_account;  
  
    COMMIT;
```

[SQL Worksheet]*      Aa 

```
1 BEGIN  
2   TransferFunds(201, 202, 500);  
3 END;  
4  
5 select * from Accounts;
```

Query result | Script output | DBMS output | Explain Plan | SQL history

  Download  Execution time: 0.001 seconds

	ACCOUNTID	CUSTOMERNAME	BALANCE
1	201	Alice	2500
2	202	Bob	3000
3	203	Charlie	1500
4	204	Diana	500