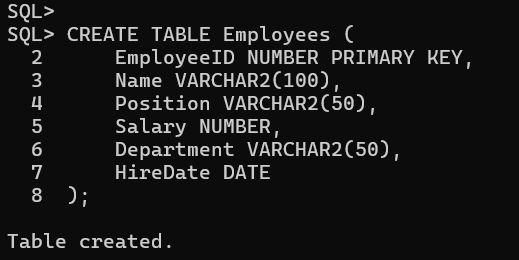
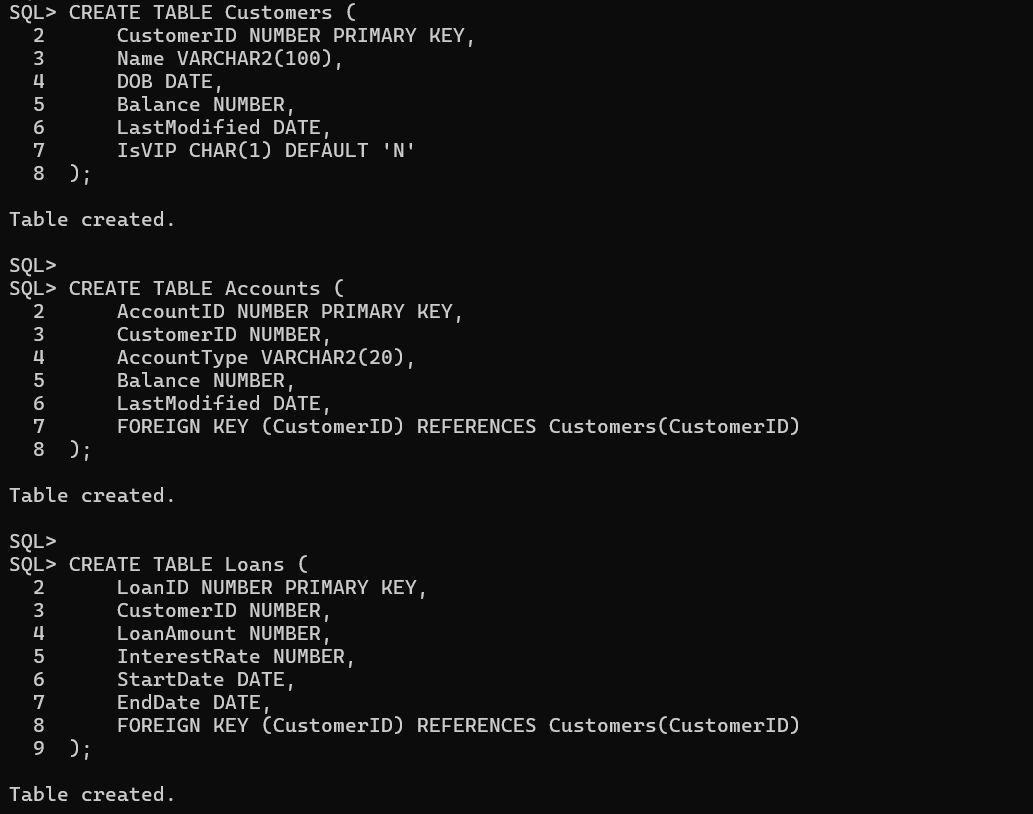
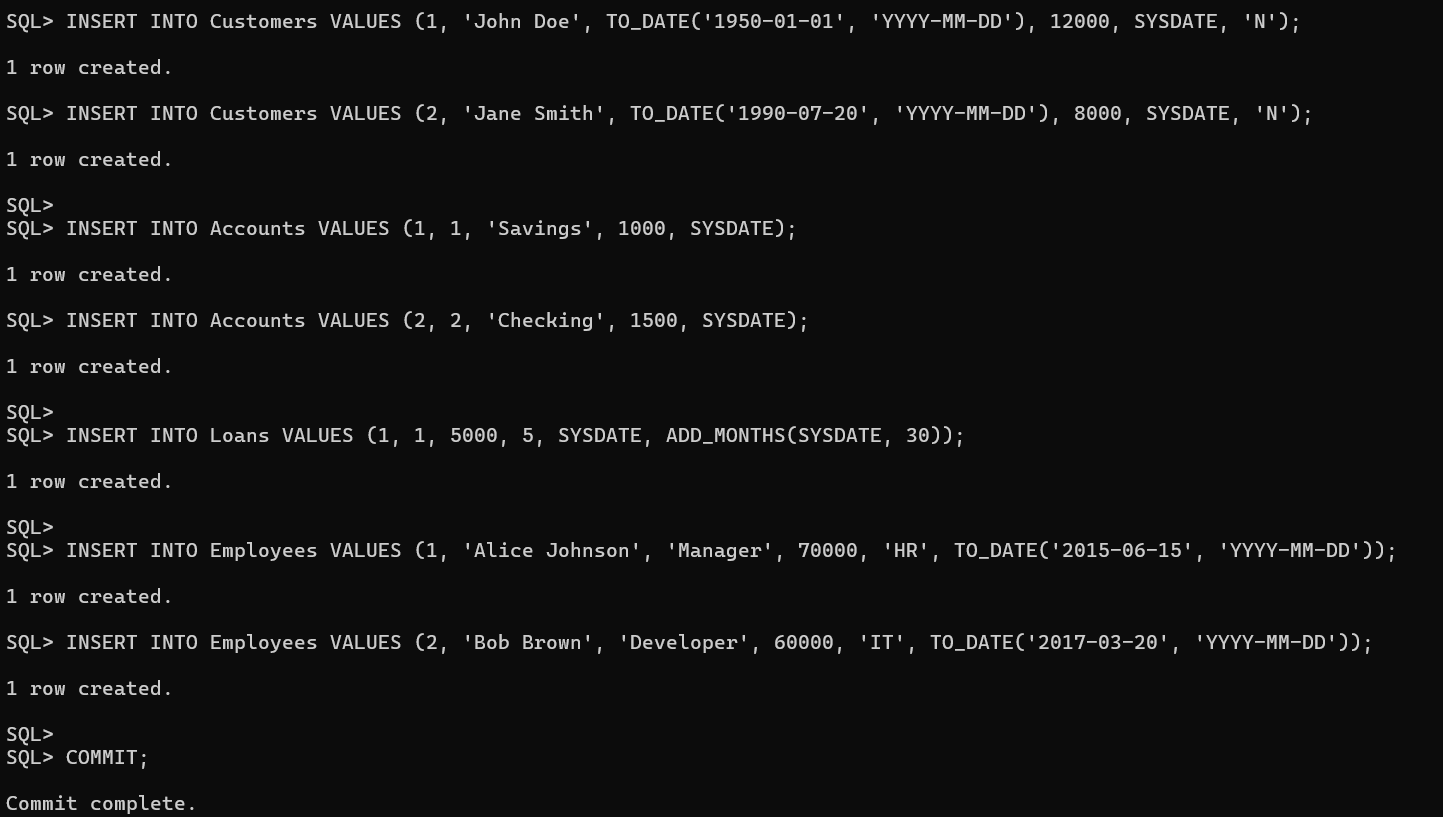
**Module 1 : PL/SQL programming**

**Step1: Creating the required tables**



**Step 2: Insert sample data**



**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.

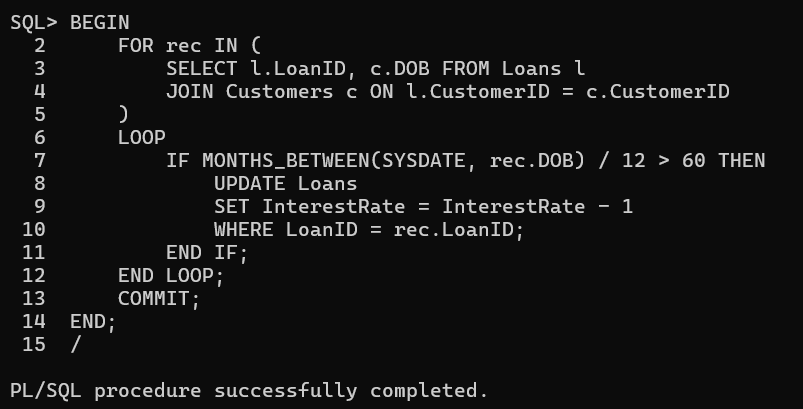
**Solution :**

**Step 1:**

**Description:** This block goes through all loans and checks if the customer is older than 60. If yes, it reduces their loan interest rate by 1%.

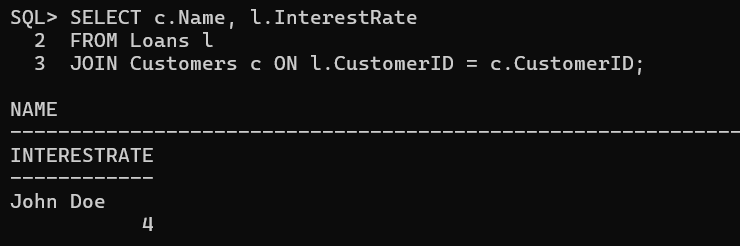
**Step 2:**

**Code:**

****

**Step 3:**

**Sample Output Check:**

****

**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.

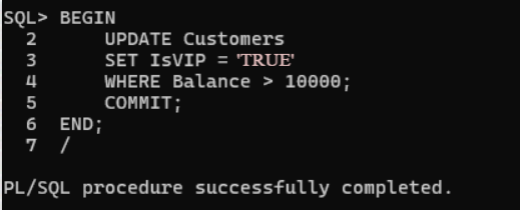
**Solution :**

**Step 1:**

**Description:** This block updates the IsVIP flag to 'TRUE' for all customers with a balance over ₹10,000.

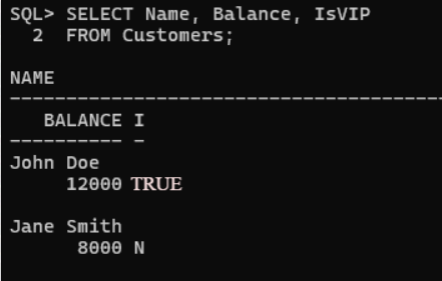
**Step 2:**

**Code:**

****

**Step 3:**

**Sample Output Check:**

****

**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.

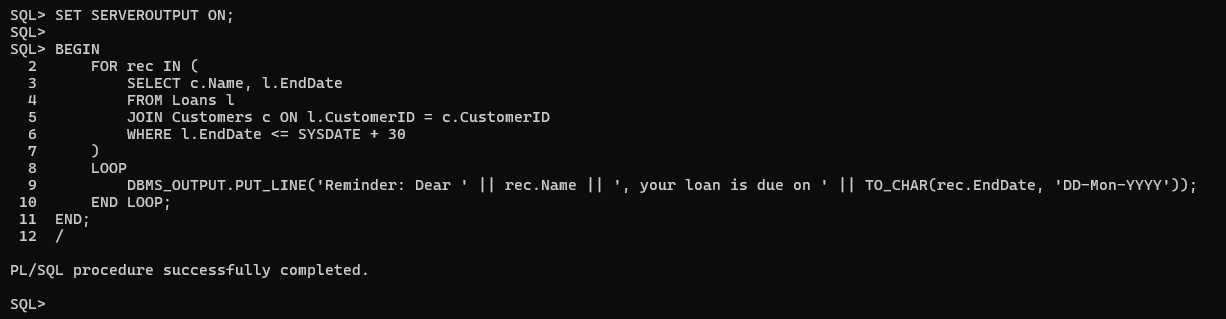
**Solution :**

**Step 1:**

**Description:** This block finds all customers whose loan is due within the next 30 days and prints a reminder message using DBMS\_OUTPUT.

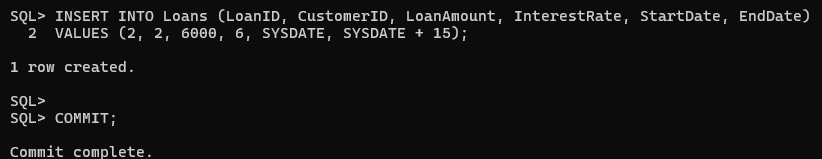
**Step 2:**

**Code:**

****

**Step 3:**

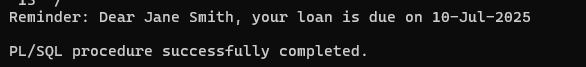
**Inserting a test loan:**

****

**Step 4:**

**Output:**

**The reminder will appear like this:**

****

**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.

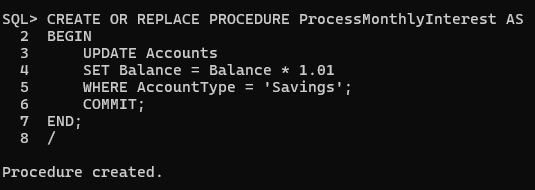
**Solution :**

**Step 1:**

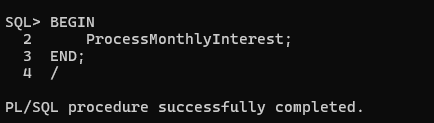
**Description:** This procedure increases the balance of all savings accounts by 1% every month. It uses a simple formula: balance \* 1.01.

**Step 2:**

**Code:**

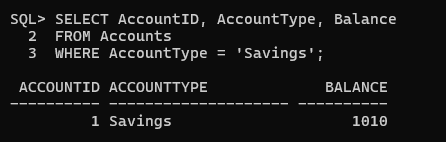
****

**Step 3:**

****

**Step 4:**

**Output:**

****

**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.

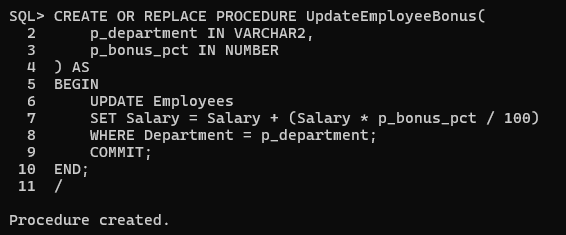
**Solution :**

**Step 1:**

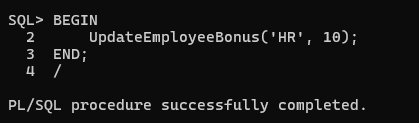
**Description:** This procedure accepts a department name and bonus percentage. It increases the salary of all employees in that department by the given bonus.

**Step 2:**

**Code:**

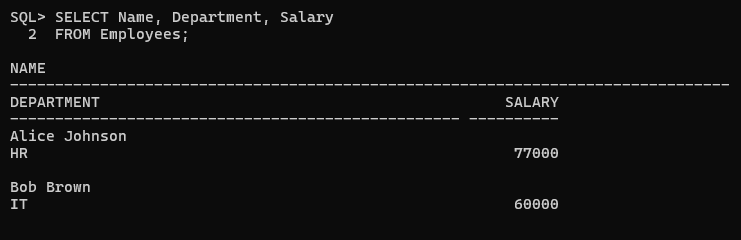
****

**Step 3:**

****

**Step 4:**

**Output:**

****

**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.

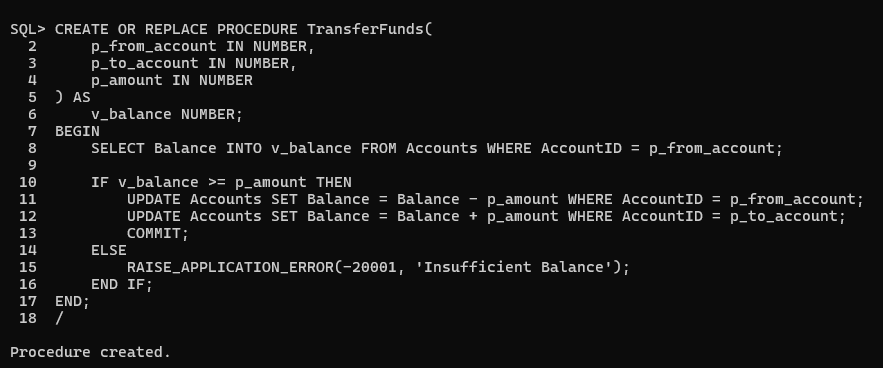
**Solution :**

**Step 1:**

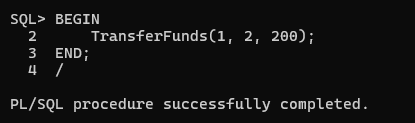
**Description:** This procedure transfers money from one account to another only if the source account has enough balance. If not, it throws an error.

**Step 2:**

**Code:**

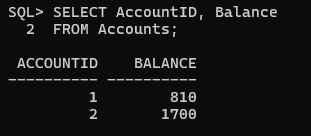
****

**Step 3:**

****

**Step 4:**

**Output:**

****