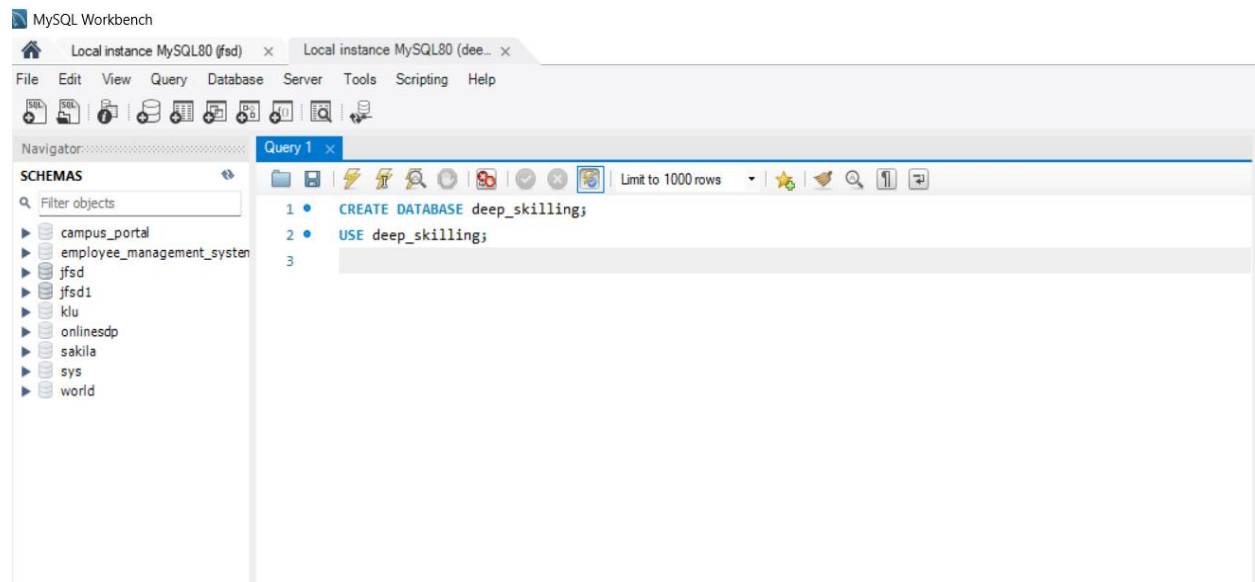


Exercise 1: Control Structures

FROM File PLSQL_Exercises

Creating a database

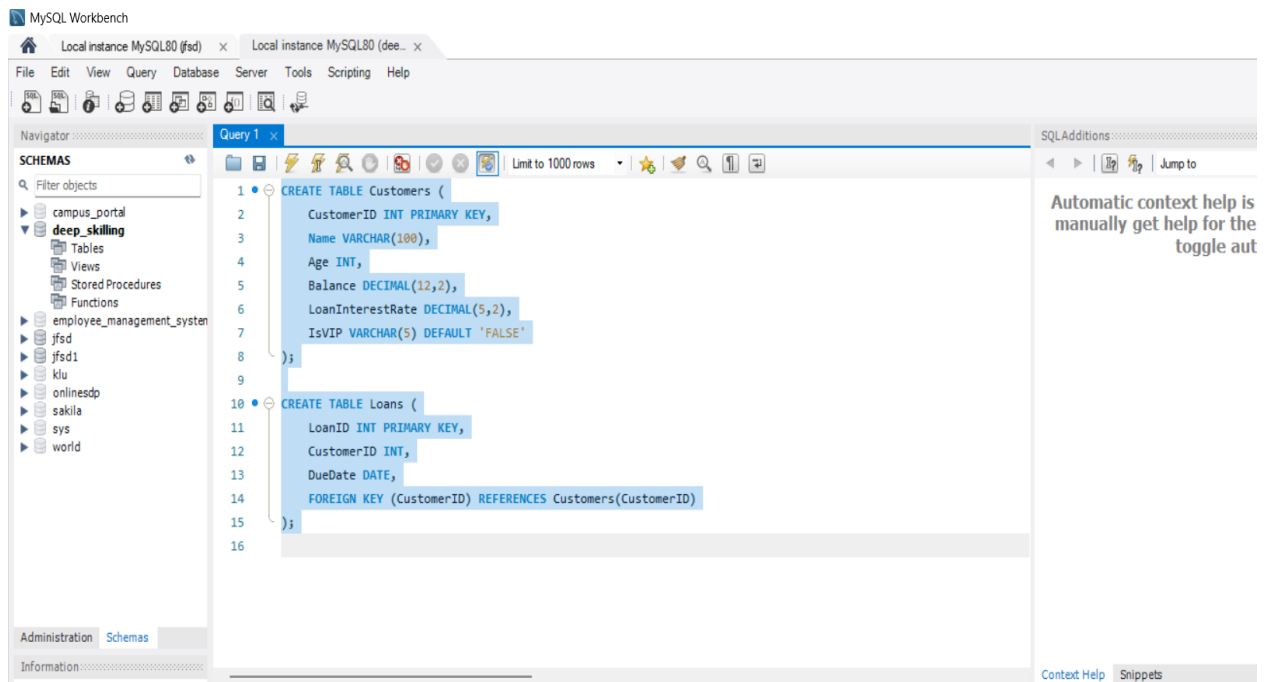


Output:

The screenshot shows the 'Output' pane in MySQL Workbench, displaying the results of the SQL queries. The output is organized into a table with columns for '#', 'Time', 'Action', 'Message', and 'Duration / Fetch'.

#	Time	Action	Message	Duration / Fetch
1	18:08:14	CREATE DATABASE deep_skilling	1 row(s) affected	0.078 sec
2	18:08:14	USE deep_skilling	0 row(s) affected	0.000 sec

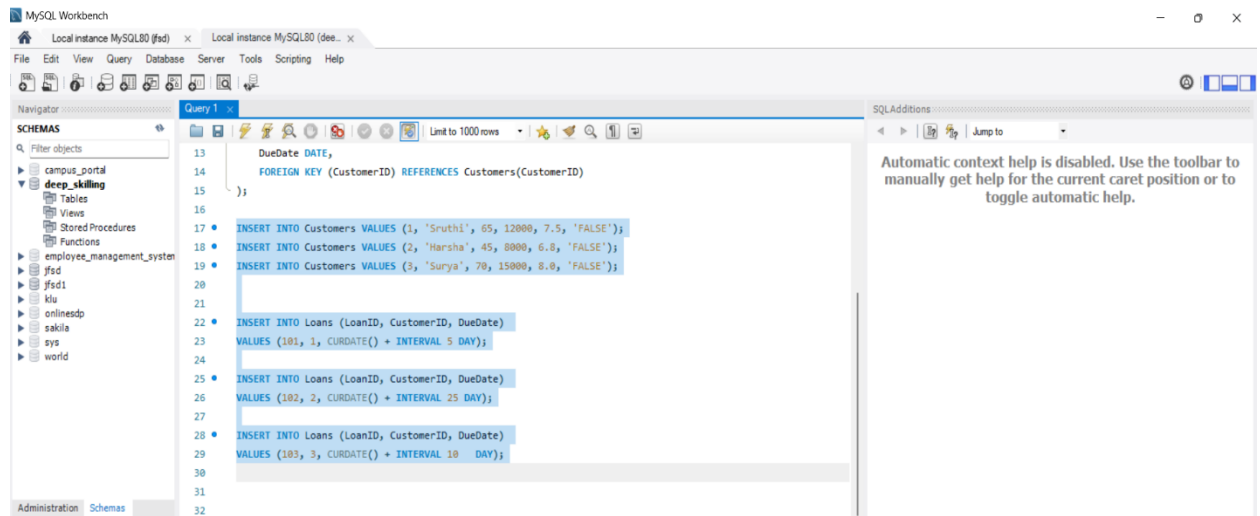
2.



Output:

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
1	18:08:14	CREATE DATABASE deep_skilling	1 row(s) affected	0.078 sec
2	18:08:14	USE deep_skilling	0 row(s) affected	0.000 sec
3	18:10:16	CREATE TABLE Customers (CustomerID INT PRIMARY KEY, Name VARCHAR(100), Age INT, Balan...	0 row(s) affected	0.093 sec
4	18:10:16	CREATE TABLE Loans (LoanID INT PRIMARY KEY, CustomerID INT, DueDate DATE, FOREIGN KE...	0 row(s) affected	0.063 sec

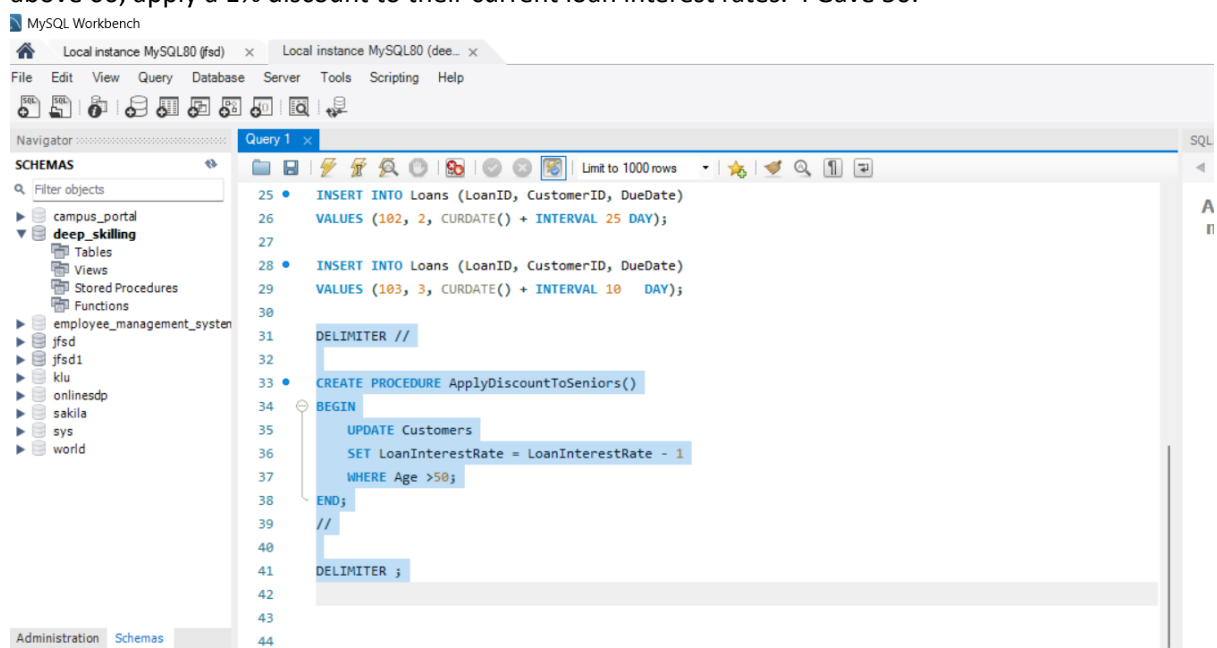
3.



Output screen:

#	Time	Action	Message	Duration / Fetch
5	18:18:05	INSERT INTO Customers VALUES (1, 'Sruthi', 65, 12000, 7.5, 'FALSE')	1 row(s) affected	0.016 sec
6	18:18:05	INSERT INTO Customers VALUES (2, 'Harsha', 45, 8000, 6.8, 'FALSE')	1 row(s) affected	0.016 sec
7	18:18:05	INSERT INTO Customers VALUES (3, 'Surya', 70, 15000, 8.0, 'FALSE')	1 row(s) affected	0.000 sec
8	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (101, 1, CURDATE() + INTERVAL 5 DAY)	1 row(s) affected	0.015 sec
9	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (102, 2, CURDATE() + INTERVAL 25 DAY)	1 row(s) affected	0.000 sec
10	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (103, 3, CURDATE() + INTERVAL 10 DAY)	1 row(s) affected	0.016 sec

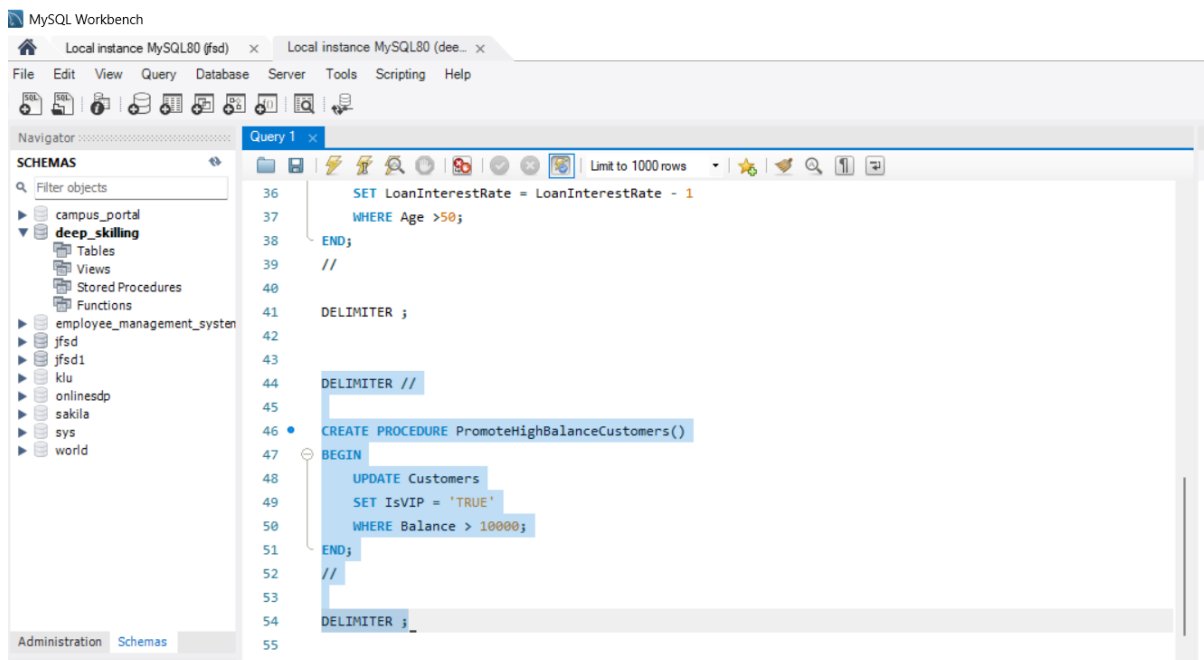
4. **Question:** Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates. I Gave 50.



Output Screen:

Output				
Action Output				
#	Time	Action	Message	
✓ 6	18:18:05	INSERT INTO Customers VALUES (2, 'Harsha', 45, 8000, 6.8, 'FALSE')	1 row(s) affected	
✓ 7	18:18:05	INSERT INTO Customers VALUES (3, 'Surya', 70, 15000, 8.0, 'FALSE')	1 row(s) affected	
✓ 8	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (101, 1, CURDATE() + INTERVAL 5 DAY)	1 row(s) affected	
✓ 9	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (102, 2, CURDATE() + INTERVAL 25 DAY)	1 row(s) affected	
✓ 10	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (103, 3, CURDATE() + INTERVAL 10 DAY)	1 row(s) affected	
✓ 11	18:22:00	CREATE PROCEDURE ApplyDiscountToSeniors() BEGIN UPDATE Customers SET LoanInterestRate = L...	0 row(s) affected	

5. **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.

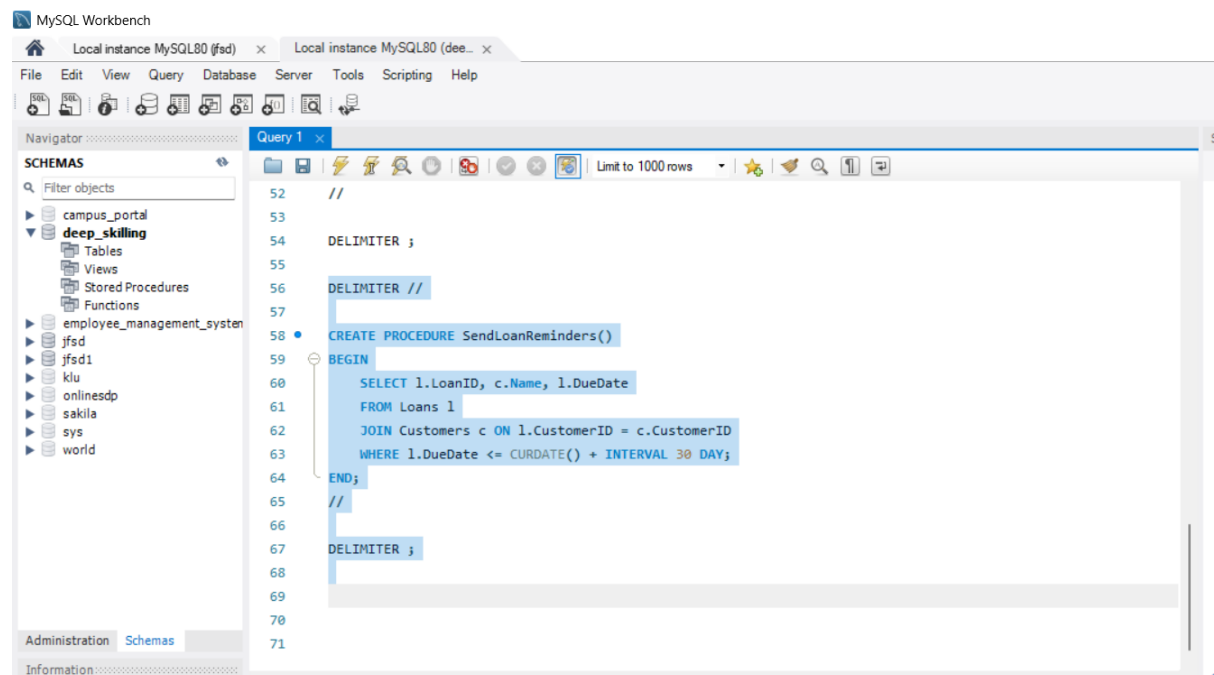


```
36      SET LoanInterestRate = LoanInterestRate - 1
37      WHERE Age > 50;
38  END;
39  //
40
41  DELIMITER ;
42
43
44  DELIMITER //
45
46  CREATE PROCEDURE PromoteHighBalanceCustomers()
47  BEGIN
48      UPDATE Customers
49      SET IsVIP = 'TRUE'
50      WHERE Balance > 10000;
51  END;
52  //
53
54  DELIMITER ;
55
```

Output screen:

Output				
Action Output				
#	Time	Action	Message	
✓ 7	18:18:05	INSERT INTO Customers VALUES (3, 'Surya', 70, 15000, 8.0, 'FALSE')	1 row(s) affected	
✓ 8	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (101, 1, CURDATE() + INTERVAL 5 DAY)	1 row(s) affected	
✓ 9	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (102, 2, CURDATE() + INTERVAL 25 DAY)	1 row(s) affected	
✓ 10	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (103, 3, CURDATE() + INTERVAL 10 DAY)	1 row(s) affected	
✓ 11	18:22:00	CREATE PROCEDURE ApplyDiscountToSeniors() BEGIN UPDATE Customers SET LoanInterestRate = L...	0 row(s) affected	
✓ 12	18:23:13	CREATE PROCEDURE PromoteHighBalanceCustomers() BEGIN UPDATE Customers SET IsVIP = 'TRUE...	0 row(s) affected	

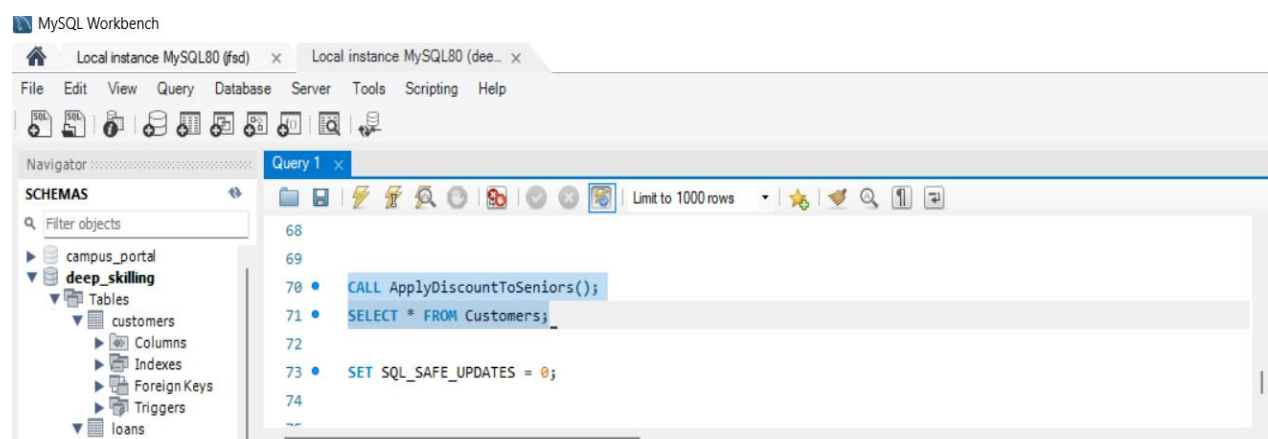
6. **Question:** Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer.



Output screen:

Output				
Action Output				
#	Time	Action	Message	
8	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (101, 1, CURDATE() + INTERVAL 5 DAY)	1 row(s) affected	
9	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (102, 2, CURDATE() + INTERVAL 25 DAY)	1 row(s) affected	
10	18:18:05	INSERT INTO Loans (LoanID, CustomerID, DueDate) VALUES (103, 3, CURDATE() + INTERVAL 10 DAY)	1 row(s) affected	
11	18:22:00	CREATE PROCEDURE ApplyDiscountToSeniors() BEGIN UPDATE Customers SET LoanInterestRate = L...	0 row(s) affected	
12	18:23:13	CREATE PROCEDURE PromoteHighBalanceCustomers() BEGIN UPDATE Customers SET IsVIP = 'TRUE...	0 row(s) affected	
13	18:23:55	CREATE PROCEDURE SendLoanReminders() BEGIN SELECT l.LoanID, c.Name, l.DueDate FROM Loan...	0 row(s) affected	

7.



Output Screen:

MySQL Workbench interface showing a query execution. The query is as follows:

```
68  
69  
70 • CALL ApplyDiscountToSeniors();  
71 • SELECT * FROM Customers;  
72  
73 • SET SQL_SAFE_UPDATES = 0;  
74
```

The result grid displays the following data:

CustomerID	Name	Age	Balance	LoanInterestRate	IsVIP
1	Sruthi	65	12000.00	6.50	FALSE
2	Harsha	45	8000.00	6.80	FALSE
3	Surya	70	15000.00	7.00	FALSE
* NULL	NULL	NULL	NULL	NULL	NULL

8.

MySQL Workbench interface showing a query execution. The query is as follows:

```
72  
73 • SET SQL_SAFE_UPDATES = 0;  
74  
75  
76 • CALL PromoteHighBalanceCustomers();  
77 • SELECT * FROM Customers;  
78
```

The result grid displays the following data:

CustomerID	Name	Age	Balance	LoanInterestRate	IsVIP
1	Sruthi	65	12000.00	6.50	TRUE
2	Harsha	45	8000.00	6.80	FALSE
3	Surya	70	15000.00	7.00	TRUE
* NULL	NULL	NULL	NULL	NULL	NULL

Output screen:

MySQL Workbench

Local instance MySQL80 (fsd) x Local instance MySQL80 (dee... x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

campus_portal

deep_skilling

Tables

customers

Columns

Indexes

Foreign Keys

Triggers

loans

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

employee_management_sys

fsd

fsd1

klu

onlinesdp

sakila

Administration Schemas

Information

Schema: jfsd

Object Info Session

Query 1 x

Limit to 1000 rows

72

73 • SET SQL_SAFE_UPDATES =0;

74

75

76 • CALL PromoteHighBalanceCustomers();

77 • SELECT * FROM Customers;

78

Result Grid

Filter Rows:

CustomerID Name Age Balance LoanInterestRate IsVIP

1	Sruthi	65	12000.00	6.50	TRUE
2	Harsha	45	8000.00	6.80	FALSE
3	Surya	70	15000.00	7.00	TRUE
NULL	NULL	NULL	NULL	NULL	NULL

Customers 2 x

Apply Revert Context

Output

Action Output

#	Time	Action	Message
17	18:26:46	SELECT * FROM Customers LIMIT 0, 1000	3 row(s) returned
18	18:27:55	SET SQL_SAFE_UPDATES = 1	0 row(s) affected
19	18:28:33	CALL PromoteHighBalanceCustomers()	Error Code: 1175. You are using safe update mode a
20	18:28:51	SET SQL_SAFE_UPDATES =0	0 row(s) affected
21	18:28:56	CALL PromoteHighBalanceCustomers()	2 row(s) affected
22	18:28:56	SELECT * FROM Customers LIMIT 0, 1000	3 row(s) returned

9.

MySQL Workbench

Local instance MySQL80 (fsd) x Local instance MySQL80 (dee... x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

campus_portal

deep_skilling

Tables

customers

Columns

Indexes

Foreign Keys

Triggers

loans

Columns

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

Functions

employee_management_sys

fsd

fsd1

klu

onlinesdp

sakila

Administration Schemas

Information

Schema: jfsd

Object Info Session

Query 1 x

Limit to 1000 rows

75

76 • CALL PromoteHighBalanceCustomers();

77 • SELECT * FROM Customers;

78

79

80 • CALL SendLoanReminders();

81

Result Grid

Filter Rows:

LoanID Name DueDate

101	Sruthi	2025-06-30
102	Harsha	2025-07-20
103	Surya	2025-07-05

Result 3 x

Read Only

Output Screen:

MySQL Workbench

Local instance MySQL80 (jfsd) x Local instance MySQL80 (dee... x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- campus_portal
- deep_skilling
 - Tables
 - customers
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - loans
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions
 - employee_management_sys
 - jfsd
 - jfsd1
 - klu
 - onlinesdp
 - sakila

Administration Schemas

Information

Schema: jfsd

Object Info Session

Query 1 x

Limit to 1000 rows

```
75
76 • CALL PromoteHighBalanceCustomers();
77 • SELECT * FROM Customers;
78
79
80 • CALL SendLoanReminders();
81
```

Result Grid

LoanID	Name	DueDate
101	Sruthi	2025-06-30
102	Harsha	2025-07-20
103	Surya	2025-07-05

Result 3 x

Read Only

Output

Action Output

#	Time	Action	Message
✓ 18	18:27:55	SET SQL_SAFE_UPDATES = 1	0 row(s) affected
✗ 19	18:28:33	CALL PromoteHighBalanceCustomers()	Error Code: 1175. You are using safe update
✓ 20	18:28:51	SET SQL_SAFE_UPDATES = 0	0 row(s) affected
✓ 21	18:28:56	CALL PromoteHighBalanceCustomers()	2 row(s) affected
✓ 22	18:28:56	SELECT * FROM Customers LIMIT 0, 1000	3 row(s) returned
✓ 23	18:29:51	CALL SendLoanReminders()	3 row(s) returned