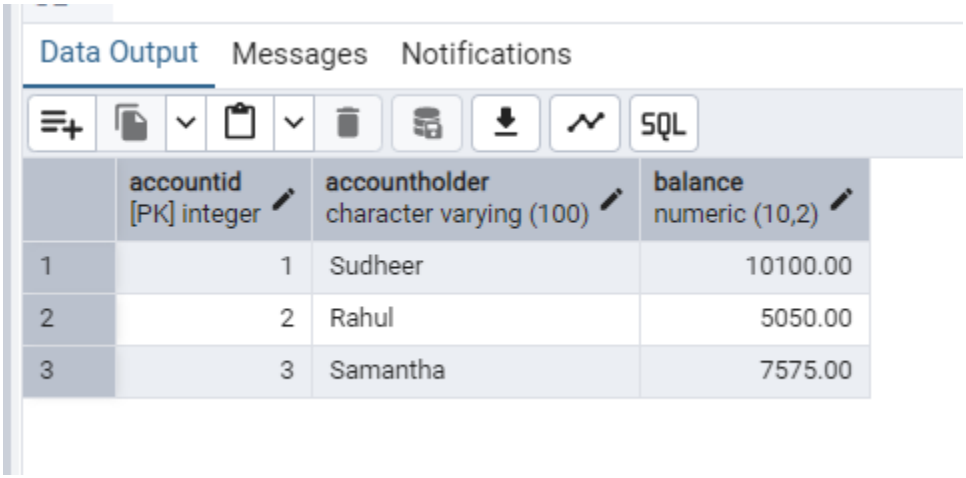
EXERCISE – 3 STORED PROCEDURES

SCENARIO 1

```
CREATE TABLE SavingsAccounts (
   AccountID SERIAL PRIMARY KEY,
   AccountHolder VARCHAR(100),
   Balance DECIMAL(10,2)
);
INSERT INTO SavingsAccounts (AccountHolder, Balance) VALUES
('Sudheer', 10000.00),
('Rahul', 5000.00),
('Samantha', 7500.00);
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest()
LANGUAGE plpgsql
AS $$
BEGIN
   UPDATE SavingsAccounts
   SET Balance = Balance + (Balance * 0.01);
END;
$$;
CALL ProcessMonthlyInterest();

SELECT * FROM SavingsAccounts;
```

Output



Data Output | Messages | Notifications

SQL

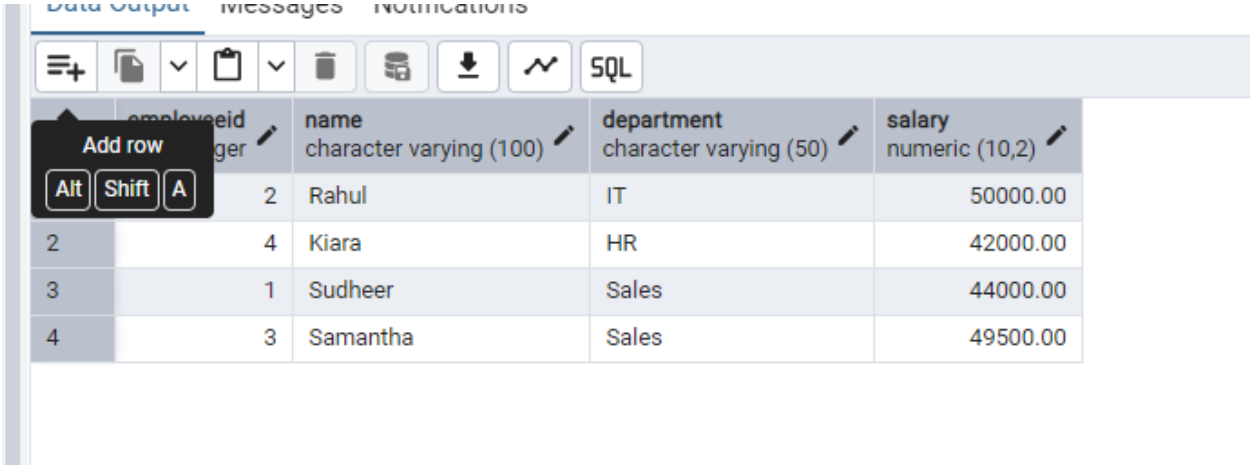
	accountid [PK] integer	accountholder character varying (100)	balance numeric (10,2)
1	1	Sudheer	10100.00
2	2	Rahul	5050.00
3	3	Samantha	7575.00

SCENARIO 2

```
CREATE TABLE Employees (
    EmployeeID SERIAL PRIMARY KEY,
    Name VARCHAR(100),
    Department VARCHAR(50),
    Salary DECIMAL(10,2)
);

INSERT INTO Employees (Name, Department, Salary) VALUES
('Sudheer', 'Sales', 40000),
('Rahul', 'IT', 50000),
('Samantha', 'Sales', 45000),
('Kiara', 'HR', 42000);
```

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(
    dept TEXT,
    bonus_percent DECIMAL
)
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE Employees
    SET Salary = Salary + (Salary * bonus_percent / 100)
    WHERE Department = dept;
END;
$$;
CALL UpdateEmployeeBonus('Sales', 10);
SELECT * FROM Employees;
```

Output



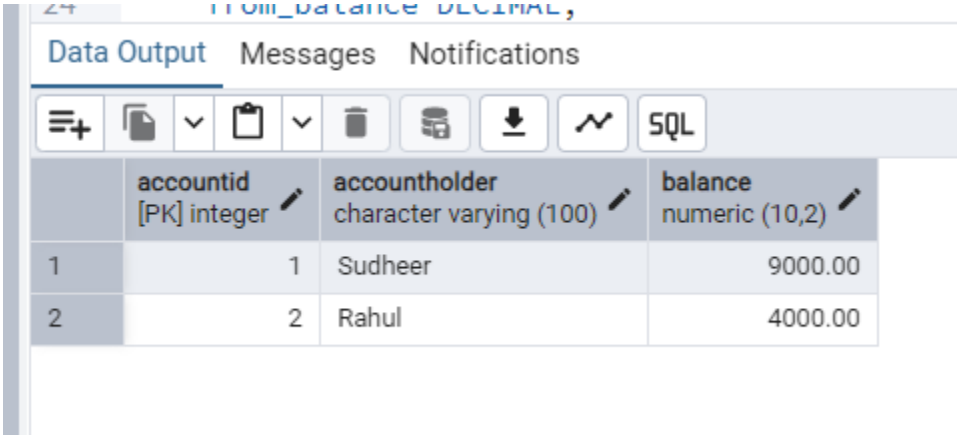
	employeeid	name character varying (100)	department character varying (50)	salary numeric (10,2)
1	2	Rahul	IT	50000.00
2	4	Kiara	HR	42000.00
3	1	Sudheer	Sales	44000.00
4	3	Samantha	Sales	49500.00

SCENARIO 3

```
CREATE TABLE Accounts (
   AccountID SERIAL PRIMARY KEY,
   AccountHolder VARCHAR(100),
   Balance DECIMAL(10,2)
);

INSERT INTO Accounts (AccountHolder, Balance) VALUES
('Sudheer', 10000.00),
('Rahul', 3000.00);
CREATE OR REPLACE PROCEDURE TransferFunds(
   from_account INT,
   to_account INT,
   amount DECIMAL
)
LANGUAGE plpgsql
AS $$
DECLARE
   from_balance DECIMAL;
BEGIN
   SELECT Balance INTO from_balance FROM Accounts WHERE AccountID = from_account;

   IF from_balance IS NULL THEN
      RAISE EXCEPTION 'Source account % does not exist', from_account;
```

```
    ELSIF from_balance < amount THEN
        RAISE EXCEPTION 'Insufficient funds in account %', from_account;
    ELSE
        UPDATE Accounts SET Balance = Balance - amount WHERE AccountID = from_account;
        UPDATE Accounts SET Balance = Balance + amount WHERE AccountID = to_account;
    END IF;
END;
$$;
CALL TransferFunds(1, 2, 1000);


SELECT * FROM Accounts;
```

Output



	accountid [PK] integer	accountholder character varying (100)	balance numeric (10,2)
1	1	Sudheer	9000.00
2	2	Rahul	4000.00