Exercise 1: Create a Stored Procedure

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Departments;

CREATE TABLE Departments (

DepartmentID INTEGER PRIMARY KEY,

DepartmentName TEXT

);

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

JoinDate TEXT,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

INSERT INTO Departments VALUES

(1, 'HR'), (2, 'Finance'), (3, 'IT'), (4, 'Marketing');

INSERT INTO Employees VALUES

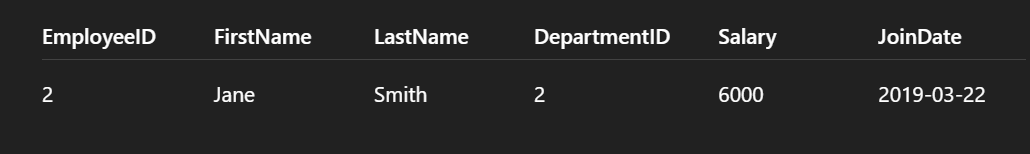
(1, 'John', 'Doe', 1, 5000.00, '2020-01-15'),

(2, 'Jane', 'Smith', 2, 6000.00, '2019-03-22'),

(3, 'Michael', 'Johnson', 3, 7000.00, '2018-07-30'),

(4, 'Emily', 'Davis', 4, 5500.00, '2021-11-05');

SELECT \* FROM Employees WHERE DepartmentID = 2;



Exercise 2: Modify a Stored Procedure

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Departments;

CREATE TABLE Departments (

DepartmentID INTEGER PRIMARY KEY,

DepartmentName TEXT

);

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

JoinDate TEXT,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

INSERT INTO Departments VALUES

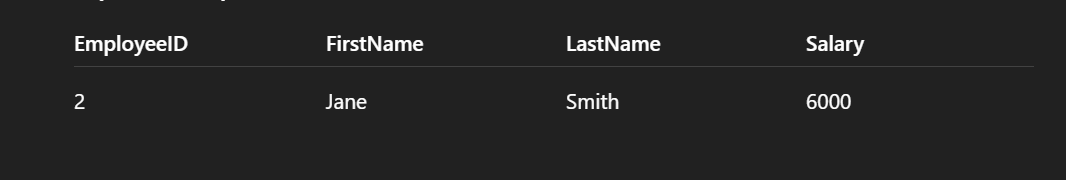
(1, 'HR'), (2, 'Finance'), (3, 'IT'), (4, 'Marketing');

INSERT INTO Employees VALUES

(1, 'John', 'Doe', 1, 5000.00, '2020-01-15'),

(2, 'Jane', 'Smith', 2, 6000.00, '2019-03-22');

SELECT EmployeeID, FirstName, LastName, Salary FROM Employees WHERE DepartmentID = 2;



Exercise 4: Execute a Stored Procedure

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Departments;

CREATE TABLE Departments (

DepartmentID INTEGER PRIMARY KEY,

DepartmentName TEXT

);

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

JoinDate TEXT,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

INSERT INTO Departments VALUES

(1, 'HR'), (2, 'Finance'), (3, 'IT'), (4, 'Marketing');

INSERT INTO Employees VALUES

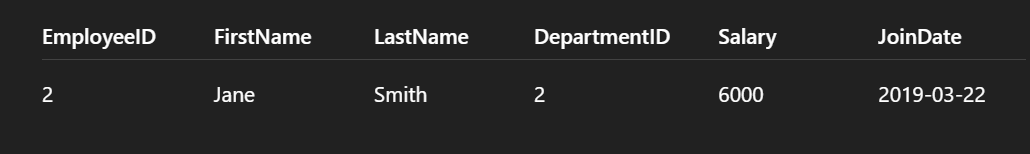
(1, 'John', 'Doe', 1, 5000.00, '2020-01-15'),

(2, 'Jane', 'Smith', 2, 6000.00, '2019-03-22'),

(3, 'Michael', 'Johnson', 3, 7000.00, '2018-07-30'),

(4, 'Emily', 'Davis', 4, 5500.00, '2021-11-05');

SELECT \* FROM Employees WHERE DepartmentID = 2;



Exercise 5: Return Data from a Stored Procedure

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Departments;

CREATE TABLE Departments (

DepartmentID INTEGER PRIMARY KEY,

DepartmentName TEXT

);

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

JoinDate TEXT,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

INSERT INTO Departments VALUES

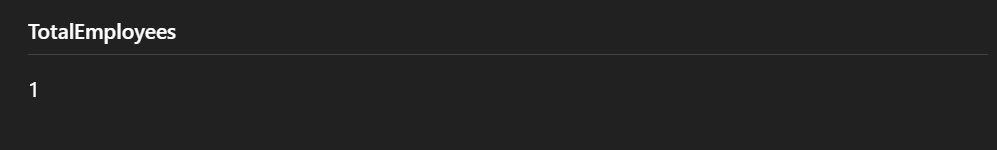
(1, 'HR'), (2, 'Finance');

INSERT INTO Employees VALUES

(1, 'John', 'Doe', 1, 5000.00, '2020-01-15'),

(2, 'Jane', 'Smith', 2, 6000.00, '2019-03-22');

SELECT COUNT(\*) AS TotalEmployees FROM Employees WHERE DepartmentID = 2;



Exercise 6: Use Output Parameters in a Stored Procedure

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Departments;

CREATE TABLE Departments (

DepartmentID INTEGER PRIMARY KEY,

DepartmentName TEXT

);

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

JoinDate TEXT,

FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)

);

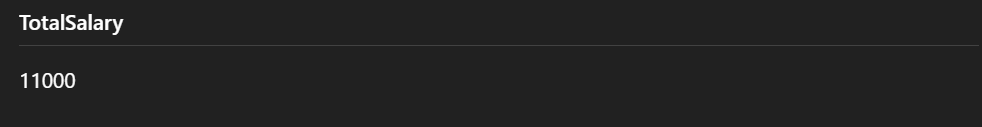
INSERT INTO Departments VALUES (1, 'HR'), (2, 'Finance');

INSERT INTO Employees VALUES

(1, 'John', 'Doe', 2, 5000.00, '2020-01-15'),

(2, 'Jane', 'Smith', 2, 6000.00, '2019-03-22');

SELECT SUM(Salary) AS TotalSalary FROM Employees WHERE DepartmentID = 2;



Exercise 7: Create a Stored Procedure with Multiple Parameters

DROP TABLE IF EXISTS Employees;

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

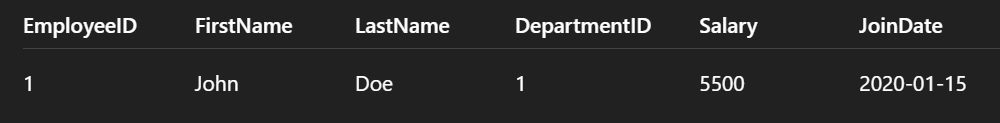
JoinDate TEXT

);

INSERT INTO Employees VALUES (1, 'John', 'Doe', 1, 5000.00, '2020-01-15');

UPDATE Employees SET Salary = 5500.00 WHERE EmployeeID = 1;

SELECT \* FROM Employees WHERE EmployeeID = 1;



Exercise 8: Create a Stored Procedure with Conditional Logic

DROP TABLE IF EXISTS Employees;

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

JoinDate TEXT

);

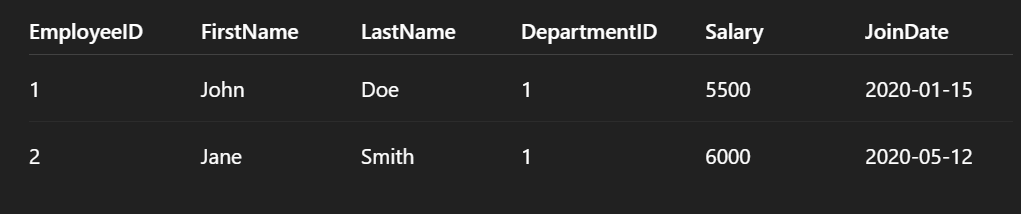
INSERT INTO Employees VALUES

(1, 'John', 'Doe', 1, 5000.00, '2020-01-15'),

(2, 'Jane', 'Smith', 1, 5500.00, '2020-05-12');

UPDATE Employees SET Salary = Salary + 500 WHERE DepartmentID = 1;

SELECT \* FROM Employees WHERE DepartmentID = 1;



Exercise 9: Use Transactions in a Stored Procedure

DROP TABLE IF EXISTS Employees;

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

JoinDate TEXT

);

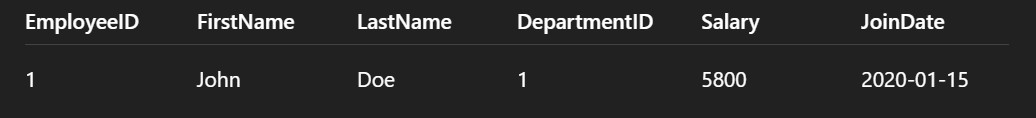
INSERT INTO Employees VALUES (1, 'John', 'Doe', 1, 5000.00, '2020-01-15');

BEGIN TRANSACTION;

UPDATE Employees SET Salary = 5800.00 WHERE EmployeeID = 1;

COMMIT;

SELECT \* FROM Employees WHERE EmployeeID = 1;



Exercise 10: Use Dynamic SQL in a Stored Procedure

DROP TABLE IF EXISTS Employees;

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

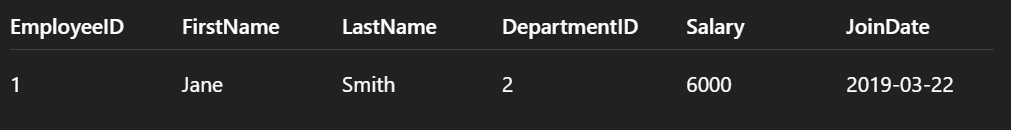
Salary REAL,

JoinDate TEXT

);

INSERT INTO Employees VALUES (1, 'Jane', 'Smith', 2, 6000.00, '2019-03-22');

SELECT \* FROM Employees WHERE LastName = 'Smith';



Exercise 11: Handle Errors in a Stored Procedure

DROP TABLE IF EXISTS Employees;

CREATE TABLE Employees (

EmployeeID INTEGER PRIMARY KEY,

FirstName TEXT,

LastName TEXT,

DepartmentID INTEGER,

Salary REAL,

JoinDate TEXT

);

INSERT INTO Employees VALUES (1, 'John', 'Doe', 1, 5000.00, '2020-01-15');

UPDATE Employees SET Salary = 6000.00 WHERE EmployeeID = 999;

SELECT \* FROM Employees;

