

STORED PROCEDURE

--CREATING A TABLE USERS

```
CREATE TABLE Users(User_id INT,Username VARCHAR(25),Password VARCHAR(25),Email  
VARCHAR(25));
```

--SHOWING THE TABLE STRUCTURE

```
EXEC sp_help Users;
```

--CREATED STORED PROCEDURE FOR CREATING USER

```
CREATE PROCEDURE CreateNewUser
```

```
    @Userid INT,
```

```
    @Username VARCHAR(25),
```

```
    @Password VARCHAR(25),
```

```
    @Email VARCHAR(25)
```

```
AS
```

```
BEGIN
```

```
    INSERT INTO Users (User_id,Username,Password,Email)
```

```
    VALUES (@Userid,@Username,@Password,@Email)
```

```
END;
```

```
GO
```

--ADDING VALUES TO THE STORED PROCEDURE

--EXECUTING STORED PROCEDURES

```
EXEC CreateNewUser
```

```
    @Userid = 101,
```

```
    @Username = 'Adithya',
```

```
    @Password = 'aDithya123',
```

```
    @Email = 'adithya555@gmail.com';
```

--STORED PROCEDURE FOR DELETING USER

CREATE PROCEDURE DeleteUser

 @UserId INT

AS

BEGIN

 DELETE FROM Users WHERE User_id = @UserId;

END;

--EXECUTING

EXEC DeleteUser @UserId = 101;

--STORED PROCEDURE FOR SELECTING USER

CREATE PROCEDURE getUserInfo

 @UserId INT

AS

BEGIN

 SELECT * FROM Users WHERE UserId = @UserId

END;

--EXECUTING

EXEC getUserInfo @UserId = 101;

--DROPPING PROCEDURE

DROP PROCEDURE DeleteUser

--DROPPING AND IF THERE EXISTS OR NOT

DROP PROCEDURE IF EXISTS DeleteUser

```

-- Create stored procedure for Update

CREATE PROCEDURE UpdateUser
    @UserId INT,
    @Username NVARCHAR(50),
    @Password NVARCHAR(50),
    @Email NVARCHAR(100)
AS
BEGIN
    UPDATE Users
    SET Username = @Username, Password = @Password, Email = @Email
    WHERE User_id = @UserId;
END;

--ALTER PROCEDURE

ALTER PROCEDURE UpdateUser
    @UserId INT,
    @Username NVARCHAR(50),
    @Password NVARCHAR(50),
    @Email NVARCHAR(100)
AS
BEGIN
    UPDATE Users
    SET Username = @Username, Password = @Password, Email = @Email
    WHERE User_id = @UserId;
END;

--EXECUTING

EXEC UpdateUser
    @UserId = 101 ,
    @Username = 'Adithyan',
    @Password = 'Adithyan123',
    @Email = 'adithyan555@gmail.com';

```

CREATING ANOTHER TABLE FOR IF ELSE INSIDE STORED PROCEDURE

CREATE TABLE STUDENTS (

Student_id INT PRIMARY KEY,

first_name VARCHAR(25),

last_Name VARCHAR(25),

dob DATE,

admission_date DATE);

-- Create stored procedure for CRUD operations

CREATE PROCEDURE ManageStudent

@Action NVARCHAR(10),

@StudentId INT = NULL,

@FirstName NVARCHAR(50) = NULL,

@LastName NVARCHAR(50) = NULL,

@DOB DATE = NULL,

@AdmissionDate DATE = NULL

--USED NULL BECAUSE NOT MANDATORY TO GIVE VALUE FOR ALL PARAMETER

AS

BEGIN

IF @Action = 'CREATE'

BEGIN

INSERT INTO Students (Student_id,first_name,last_Name,dob,admission_date)

VALUES (@StudentId, @FirstName, @LastName, @DOB, @AdmissionDate);

END

ELSE IF @Action = 'READ'

BEGIN

SELECT * FROM Students WHERE Student_id = @StudentId;

END

ELSE IF @Action = 'UPDATE'

BEGIN

```
UPDATE Students

SET first_name = @FirstName, last_Name = @LastName, dob = @DOB, admission_date =
@AdmissionDate

WHERE Student_id = @StudentId;

END

ELSE IF @Action = 'DELETE'

BEGIN

DELETE FROM Students WHERE Student_id = @StudentId;

END

END;
```

EXEC ManageStudent

```
@Action = 'CREATE',
    @StudentId = 101,
@FirstName = 'John',
@LastName = 'Doe',
@DOB = '2000-01-01',
@AdmissionDate = '2024-09-01';
```

EXEC ManageStudent

```
@Action = 'READ',
@StudentId = 101;

-- Only StudentId is required for reading
```

EXEC ManageStudent

```
@Action = 'UPDATE',
@StudentId = 101,
@FirstName = 'Jane';

-- Update only the first name
```

EXEC ManageStudent

@Action = 'DELETE',

@StudentId = 101;

-- Only StudentId is needed for deletion

SELECT * FROM Students