# LeagueLore

## Apresentação Final

Base de Dados 2023

| | | |
|---|---|---|
| Ana Loureiro | 104063 | LEI |
| Guilherme Lopes | 93393 | LECI |

# Introdução

Porquê uma database de League of Legends Lore

# O que podes encontrar…

Regiões

Champions

Universos

Histórias

# SQL Code – Stored Procedures

## ADD Procedures

- AddChampion
- AddUniverse
- AddStory

- AddExists_On
- AddParticipates

## UPDATE Procedures

- UpdateChampion
- UpdateRegion
- UpdateUniverse
- UpdateStory

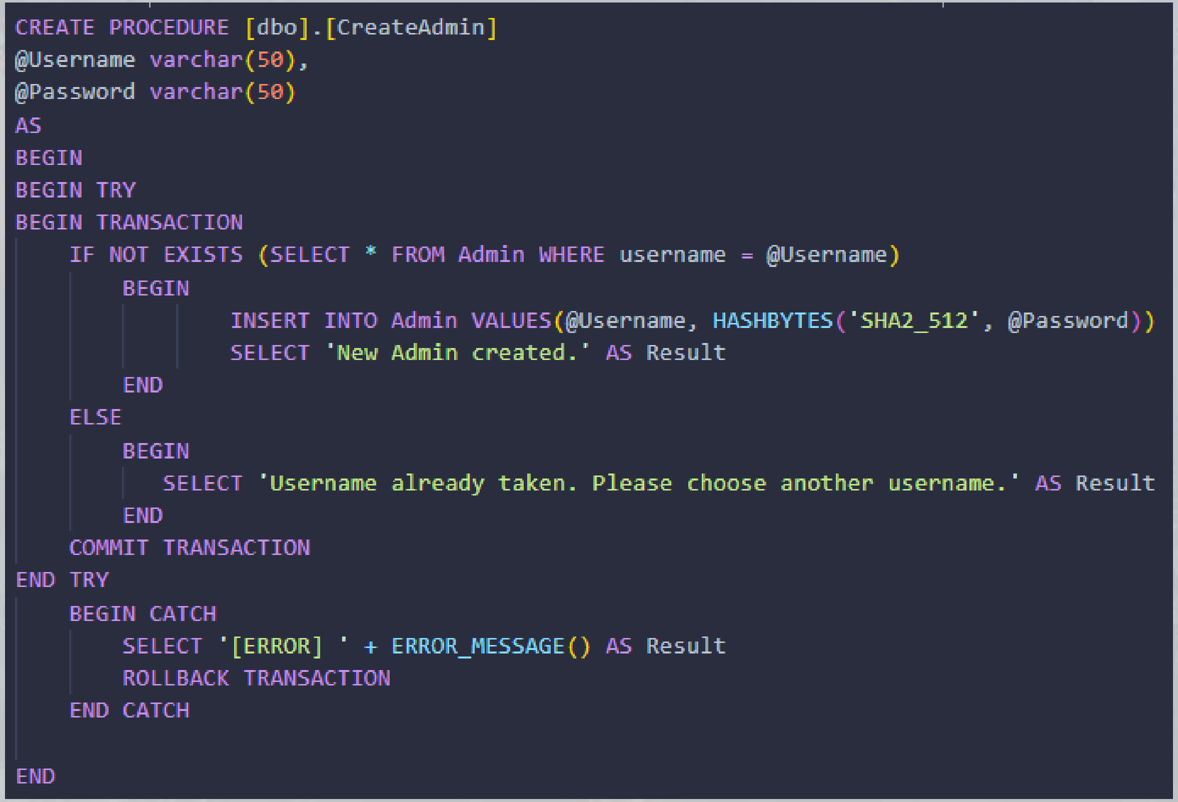
## DELETE Procedures

- DeleteChampion
- DeleteUniverse
- DeleteStory

- DeleteExists_On
- DeleteParticipates

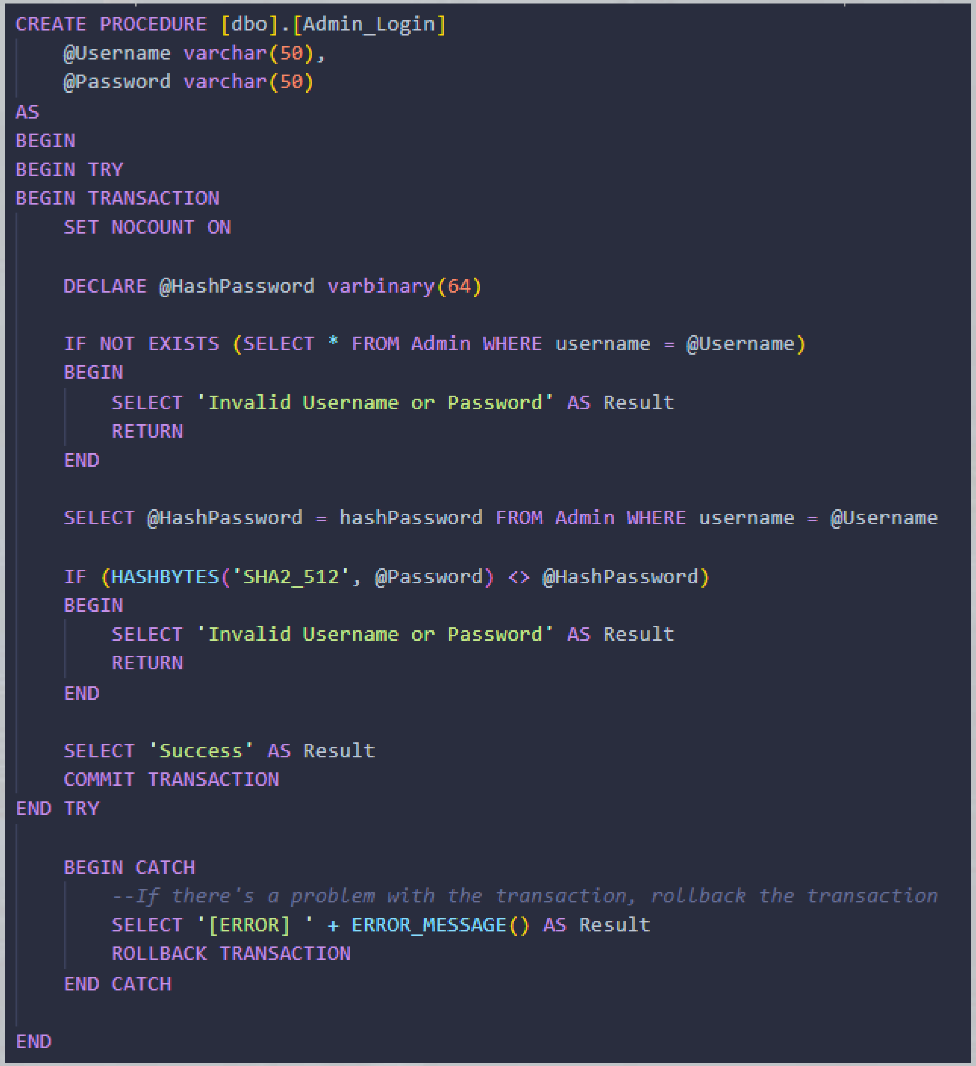
# SQL Code – Stored Procedures

CreateAdmin



```sql
CREATE PROCEDURE [dbo].[CreateAdmin]
@Username varchar(50),
@Password varchar(50)
AS
BEGIN
BEGIN TRY
BEGIN TRANSACTION
    IF NOT EXISTS (SELECT * FROM Admin WHERE username = @Username)
        BEGIN
                INSERT INTO Admin VALUES(@Username, HASHBYTES('SHA2_512', @Password))
                SELECT 'New Admin created.' AS Result
        END
    ELSE
        BEGIN
           SELECT 'Username already taken. Please choose another username.' AS Result
        END
    COMMIT TRANSACTION
END TRY
    BEGIN CATCH
        SELECT '[ERROR] ' + ERROR_MESSAGE() AS Result
        ROLLBACK TRANSACTION
    END CATCH

END
```

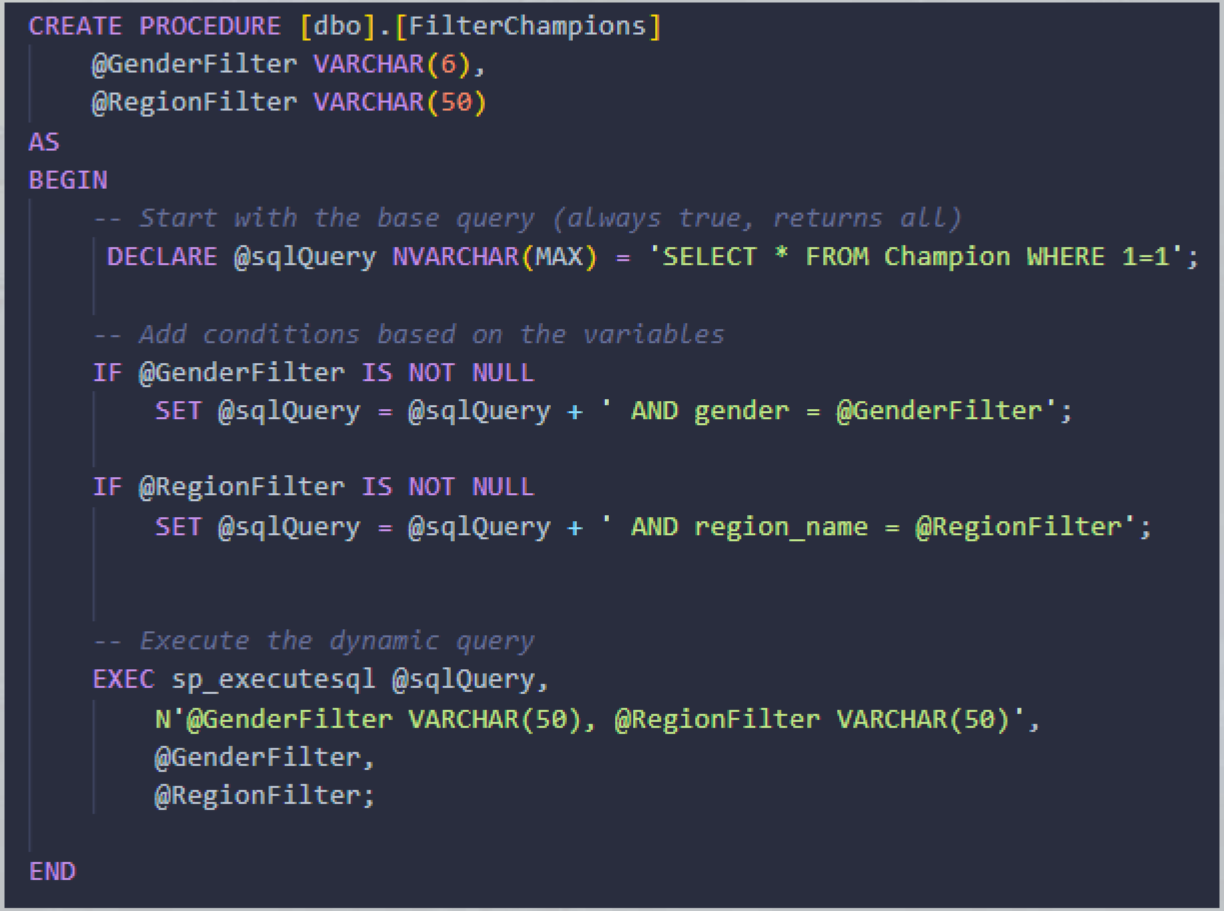
AdminLogin



```sql
CREATE PROCEDURE [dbo].[Admin_Login]
    @Username varchar(50),
    @Password varchar(50)
AS
BEGIN
BEGIN TRY
BEGIN TRANSACTION
    SET NOCOUNT ON

    DECLARE @HashPassword varbinary(64)

    IF NOT EXISTS (SELECT * FROM Admin WHERE username = @Username)
    BEGIN
        SELECT 'Invalid Username or Password' AS Result
        RETURN
    END

    SELECT @HashPassword = hashPassword FROM Admin WHERE username = @Username

    IF (HASHBYTES('SHA2_512', @Password) <> @HashPassword)
    BEGIN
        SELECT 'Invalid Username or Password' AS Result
        RETURN
    END

    SELECT 'Success' AS Result
    COMMIT TRANSACTION
END TRY

    BEGIN CATCH
        --If there's a problem with the transaction, rollback the transaction
        SELECT '[ERROR] ' + ERROR_MESSAGE() AS Result
        ROLLBACK TRANSACTION
    END CATCH

END
```

# SQL Code – Stored Procedures

FilterChampions →



```sql
CREATE PROCEDURE [dbo].[FilterChampions]
    @GenderFilter VARCHAR(6),
    @RegionFilter VARCHAR(50)
AS
BEGIN
    -- Start with the base query (always true, returns all)
     DECLARE @sqlQuery NVARCHAR(MAX) = 'SELECT * FROM Champion WHERE 1=1';

    -- Add conditions based on the variables
    IF @GenderFilter IS NOT NULL
        SET @sqlQuery = @sqlQuery + ' AND gender = @GenderFilter';

    IF @RegionFilter IS NOT NULL
        SET @sqlQuery = @sqlQuery + ' AND region_name = @RegionFilter';


    -- Execute the dynamic query
    EXEC sp_executesql @sqlQuery,
        N'@GenderFilter VARCHAR(50), @RegionFilter VARCHAR(50)',
        @GenderFilter,
        @RegionFilter;

END
```