CREATION OF TABLE

-- User table

CREATE TABLE [User] (

UserId INT PRIMARY KEY,

UserEmail NVARCHAR(255) NOT NULL,

FirstName NVARCHAR(100),

LastName NVARCHAR(100),

Password NVARCHAR(100) NOT NULL,

StreetName NVARCHAR(255),

State NVARCHAR(100),

Country NVARCHAR(100),

PostalCode NVARCHAR(20),

UserType NVARCHAR(50)

);

GO

-- Team table

CREATE TABLE Team (

TeamId INT PRIMARY KEY,

TeamName NVARCHAR(100),

TeamLocation NVARCHAR(100),

Sponsorship NVARCHAR(100)

);

GO

-- Skills table

CREATE TABLE Skills (

SkillId INT PRIMARY KEY,

SkillName NVARCHAR(100),

SkillDescription NVARCHAR(255)

);

GO

-- Stadium table

CREATE TABLE Stadium (

StadiumId INT PRIMARY KEY,

StadiumName NVARCHAR(100),

Location NVARCHAR(100),

Capacity INT,

AdminId INT

);

GO

-- Match table

CREATE TABLE Match (

MatchId INT PRIMARY KEY,

MatchDate DATETIME,

MatchLocation NVARCHAR(100),

HomeTeamId INT,

AwayTeamId INT,

StadiumId INT,

WinningTeam INT

);

GO

-- Viewer table

CREATE TABLE Viewer (

ViewerId INT PRIMARY KEY,

FavoriteTeam NVARCHAR(100),

LanguagePreference NVARCHAR(50),

FOREIGN KEY (ViewerId) REFERENCES [User](UserId)

);

GO

-- Team Staff table

CREATE TABLE TeamStaff (

StaffId INT PRIMARY KEY,

PlayingCountry NVARCHAR(100),

StaffType NVARCHAR(50),

FOREIGN KEY (StaffId) REFERENCES [User](UserId)

);

GO

-- Admin table

CREATE TABLE Admin (

AdminId INT PRIMARY KEY,

AdminRole NVARCHAR(100),

AdminPermissions NVARCHAR(100),

JoiningDate DATE,

YearsOfExperience INT,

LastLogin DATETIME,

FOREIGN KEY (AdminId) REFERENCES [User](UserId)

);

GO

-- Coach table

CREATE TABLE Coach (

CoachId INT PRIMARY KEY,

CoachingExperience INT,

Specialization NVARCHAR(100),

CoachingPhilosophy NVARCHAR(255),

FOREIGN KEY (CoachId) REFERENCES [User](UserId)

);

GO

-- Player table

CREATE TABLE Player (

PlayerId INT PRIMARY KEY,

Position NVARCHAR(50),

IsSubstitute BIT,

MinutesPlayed INT,

FOREIGN KEY (PlayerId) REFERENCES [User](UserId)

);

GO

-- Player Statistics table

CREATE TABLE PlayerStatistics (

PlayerStatsId INT PRIMARY KEY,

MatchId INT,

PlayerId INT,

Score INT,

FOREIGN KEY (PlayerId) REFERENCES Player(PlayerId),

FOREIGN KEY (MatchId) REFERENCES Match(MatchId)

);

GO

-- Player Skills table

CREATE TABLE PlayerSkills (

PlayerId INT,

SkillId INT,

PRIMARY KEY (PlayerId, SkillId),

FOREIGN KEY (PlayerId) REFERENCES Player(PlayerId),

FOREIGN KEY (SkillId) REFERENCES Skills(SkillId)

);

GO

-- Contract table

CREATE TABLE Contract (

ContractId INT PRIMARY KEY,

Description NVARCHAR(255),

StartDate DATE,

EndDate DATE,

TeamId INT,

IsTeamCaptain BIT,

FOREIGN KEY (TeamId) REFERENCES Team(TeamId)

);

GO

-- Transaction table

CREATE TABLE [Transaction] (

TransactionId INT PRIMARY KEY,

MatchId INT,

ViewerId INT,

Amount MONEY,

PaymentMode NVARCHAR(50),

TransactionTimestamp DATETIME,

FOREIGN KEY (MatchId) REFERENCES Match(MatchId),

FOREIGN KEY (ViewerId) REFERENCES Viewer(ViewerId)

);

GO

-- Now, let's add foreign keys that reference other tables, which have to be created after the referenced table exists

ALTER TABLE Stadium ADD CONSTRAINT FK\_Stadium\_Admin FOREIGN KEY (AdminId) REFERENCES Admin(AdminId);

GO

ALTER TABLE Match ADD CONSTRAINT FK\_Match\_HomeTeam FOREIGN KEY (HomeTeamId) REFERENCES Team(TeamId);

GO

ALTER TABLE Match ADD CONSTRAINT FK\_Match\_AwayTeam FOREIGN KEY (AwayTeamId) REFERENCES Team(TeamId);

GO

ALTER TABLE Match ADD CONSTRAINT FK\_Match\_Stadium FOREIGN KEY (StadiumId) REFERENCES Stadium(StadiumId);

GO

Queries insert data in the TABLE

-- Insert multiple rows into User table

INSERT INTO [User] (UserId, UserEmail, FirstName, LastName, Password, StreetName, State, Country, PostalCode, UserType)

VALUES

(1, 'john.doe@example.com', 'John', 'Doe', 'password123', '123 Main St', 'SomeState', 'SomeCountry', '12345', 'Viewer'),

(2, 'jane.smith@example.com', 'Jane', 'Smith', 'password124', '124 Main St', 'SomeState', 'SomeCountry', '12345', 'Staff'),

(3, 'mike.jones@example.com', 'Mike', 'Jones', 'password125', '125 Main St', 'SomeState', 'SomeCountry', '12345', 'Admin'),

(4, 'lisa.white@example.com', 'Lisa', 'White', 'password126', '126 Main St', 'SomeState', 'SomeCountry', '12345', 'Coach'),

(5, 'gary.black@example.com', 'Gary', 'Black', 'password127', '127 Main St', 'SomeState', 'SomeCountry', '12345', 'Player');

-- Insert multiple rows into Viewer table

INSERT INTO Viewer (ViewerId, FavoriteTeam, LanguagePreference)

VALUES

(1, 'Team A', 'English'),

(6, 'Team B', 'Spanish'),

(7, 'Team C', 'French'),

(8, 'Team D', 'German'),

(9, 'Team E', 'Italian');

-- Insert multiple rows into Team Staff table

INSERT INTO TeamStaff (StaffId, PlayingCountry, StaffType)

VALUES

(2, 'Country A', 'Manager'),

(10, 'Country B', 'Coach'),

(11, 'Country C', 'Physio'),

(12, 'Country D', 'Scout'),

(13, 'Country E', 'Data Analyst');

-- Insert multiple rows into Admin table

INSERT INTO Admin (AdminId, AdminRole, AdminPermissions, JoiningDate, YearsOfExperience, LastLogin)

VALUES

(3, 'Role A', 'Permissions A', '2023-01-01', 5, '2023-11-18T08:00:00'),

(14, 'Role B', 'Permissions B', '2022-01-01', 10, '2023-11-17T08:00:00'),

(15, 'Role C', 'Permissions C', '2021-01-01', 8, '2023-11-16T08:00:00'),

(16, 'Role D', 'Permissions D', '2020-01-01', 15, '2023-11-15T08:00:00'),

(17, 'Role E', 'Permissions E', '2019-01-01', 20, '2023-11-14T08:00:00');

-- Insert multiple rows into Coach table

INSERT INTO Coach (CoachId, CoachingExperience, Specialization, CoachingPhilosophy)

VALUES

(4, 10, 'Offense', 'Philosophy A'),

(18, 8, 'Defense', 'Philosophy B'),

(19, 12, 'Special Teams', 'Philosophy C'),

(20, 5, 'Youth Development', 'Philosophy D'),

(21, 7, 'Fitness', 'Philosophy E');

-- Insert multiple rows into Player table

INSERT INTO Player (PlayerId, Position, IsSubstitute, MinutesPlayed)

VALUES

(5, 'Forward', 0, 90),

(22, 'Midfielder', 0, 85),

(23, 'Defender', 1, 60),

(24, 'Goalkeeper', 0, 90),

(25, 'Striker', 0, 80);

-- Insert multiple rows into Player Statistics table

INSERT INTO PlayerStatistics (PlayerStatsId, MatchId, PlayerId, Score)

VALUES

(1, 1, 5, 1),

(2, 2, 22, 0),

(3, 3, 23, 0),

(4, 4, 24, 0),

(5, 5, 25, 2);

-- Insert multiple rows into Player Skills table

-- Assuming SkillIds are 1 through 5

INSERT INTO PlayerSkills (PlayerId, SkillId)

VALUES

(5, 1),

(22, 2),

(23, 3),

(24, 4),

(25, 5);

-- Insert multiple rows into Skills table

INSERT INTO Skills (SkillId, SkillName, SkillDescription)

VALUES

(1, 'Passing', 'Ability to pass the ball accurately'),

(2, 'Shooting', 'Ability to shoot the ball accurately'),

(3, 'Tackling', 'Ability to tackle the ball away from opponents'),

(4, 'Dribbling', 'Ability to dribble the ball past opponents'),

(5, 'Goalkeeping', 'Ability to save shots on goal');

-- Insert multiple rows into Team table

INSERT INTO Team (TeamId, TeamName, TeamLocation, Sponsorship)

VALUES

(1, 'Team A', 'Location A', 'Sponsor A'),

(2, 'Team B', 'Location B', 'Sponsor B'),

(3, 'Team C', 'Location C', 'Sponsor C'),

(4, 'Team D', 'Location D', 'Sponsor D'),

(5, 'Team E', 'Location E', 'Sponsor E');

-- Insert multiple rows into Contract table

-- Assuming StartDate and EndDate are for year 2023

INSERT INTO Contract (ContractId, Description, StartDate, EndDate, TeamId, IsTeamCaptain)

VALUES

(1, 'Contract A', '2023-01-01', '2023-12-31', 1, 1),

(2, 'Contract B', '2023-02-01', '2023-12-31', 2, 0),

(3, 'Contract C', '2023-03-01', '2023-12-31', 3, 0),

(4, 'Contract D', '2023-04-01', '2023-12-31', 4, 0),

(5, 'Contract E', '2023-05-01', '2023-12-31', 5, 0);

-- Insert multiple rows into Transaction table

-- Assuming amounts and timestamps

INSERT INTO Transaction (TransactionId, MatchId, ViewerId, Amount, PaymentMode, TransactionTimestamp)

VALUES

(1, 1, 1, 100.00, 'Credit Card', '2023-11-18T08:30:00'),

(2, 2, 6, 150.00, 'PayPal', '2023-11-18T09:30:00'),

(3, 3, 7, 200.00, 'Debit Card', '2023-11-18T10:30:00'),

(4, 4, 8, 250.00, 'Credit Card', '2023-11-18T11:30:00'),

(5, 5, 9, 300.00, 'Bank Transfer', '2023-11-18T12:30:00');

-- Insert multiple rows into Stadium table

-- Assuming capacities and admin IDs

INSERT INTO Stadium (StadiumId, StadiumName, Location, Capacity, AdminId)

VALUES

(1, 'Stadium A', 'Location A', 50000, 3),

(2, 'Stadium B', 'Location B', 40000, 14),

(3, 'Stadium C', 'Location C', 30000, 15),

(4, 'Stadium D', 'Location D', 20000, 16),

(5, 'Stadium E', 'Location E', 10000, 17);

-- Insert multiple rows into Match table

-- Assuming dates and team IDs

INSERT INTO Match (MatchId, MatchDate, MatchLocation, HomeTeamId, AwayTeamId, StadiumId, WinningTeam)

VALUES

(1, '2023-11-18T15:00:00', 'Stadium Location A', 1, 2, 1, 1),

(2, '2023-11-19T15:00:00', 'Stadium Location B', 2, 3, 2, 2),

(3, '2023-11-20T15:00:00', 'Stadium Location C', 3, 4, 3, 3),

(4, '2023-11-21T15:00:00', 'Stadium Location D', 4, 5, 4, 4),

(5, '2023-11-22T15:00:00', 'Stadium Location E', 5, 1, 5, 5);

STORED PROCEDURE

-- Stored Procedure to get user details

CREATE PROCEDURE GetUserDetails

@UserId INT,

@UserType NVARCHAR(50) OUTPUT

AS

BEGIN

SELECT @UserType = UserType FROM [User] WHERE UserId = @UserId;

END;

GO

-- Stored Procedure to update player minutes

CREATE PROCEDURE UpdatePlayerMinutes

@PlayerId INT,

@Minutes INT

AS

BEGIN

UPDATE Player SET MinutesPlayed = MinutesPlayed + @Minutes WHERE PlayerId = @PlayerId;

END;

GO

-- Stored Procedure to insert a new match and output its ID

CREATE PROCEDURE InsertMatch

@MatchDate DATETIME,

@MatchLocation NVARCHAR(100),

@HomeTeamId INT,

@AwayTeamId INT,

@StadiumId INT,

@MatchId INT OUTPUT

AS

BEGIN

INSERT INTO Match (MatchDate, MatchLocation, HomeTeamId, AwayTeamId, StadiumId)

VALUES (@MatchDate, @MatchLocation, @HomeTeamId, @AwayTeamId, @StadiumId);

SET @MatchId = SCOPE\_IDENTITY();

END;

GO

VIEWS

-- View for team details

CREATE VIEW TeamDetails AS

SELECT t.TeamId, t.TeamName, t.TeamLocation, t.Sponsorship, COUNT(p.PlayerId) as PlayerCount

FROM Team t

JOIN Player p ON t.TeamId = p.PlayerId

GROUP BY t.TeamId, t.TeamName, t.TeamLocation, t.Sponsorship;

GO

-- View for match statistics

CREATE VIEW MatchStatistics AS

SELECT m.MatchId, m.MatchDate, m.MatchLocation, SUM(ps.Score) as TotalScore

FROM Match m

JOIN PlayerStatistics ps ON m.MatchId = ps.MatchId

GROUP BY m.MatchId, m.MatchDate, m.MatchLocation;

GO

-- View for transaction summary

CREATE VIEW TransactionSummary AS

SELECT t.TransactionId, t.Amount, t.PaymentMode, t.TransactionTimestamp, u.FirstName, u.LastName

FROM Transaction t

JOIN [User] u ON t.ViewerId = u.UserId;

GO

DML TRIGGER

-- Trigger to update last login after admin logs in

CREATE TRIGGER UpdateLastLogin

ON Admin

AFTER UPDATE

AS

BEGIN

IF UPDATE(LastLogin)

BEGIN

-- Add logic to log or process after an admin's last login time is updated

PRINT 'Admin last login updated';

END

END;

GO

TABLE CHECK CONSTRAINT

-- Check constraint for valid email

ALTER TABLE [User]

ADD CONSTRAINT CHK\_User\_Email

CHECK (UserEmail LIKE '%@%.%');

GO

-- Check constraint for positive amount in transactions

ALTER TABLE Transaction

ADD CONSTRAINT CHK\_Transaction\_Amount

CHECK (Amount > 0);

GO

-- Check constraint for user type

ALTER TABLE [User]

ADD CONSTRAINT CHK\_User\_Type

CHECK (UserType IN ('Viewer', 'Staff', 'Admin', 'Coach', 'Player'));

GO

Computed Columns Based on a User-Defined Function (UDF)

CREATE FUNCTION dbo.GetFullName (@FirstName NVARCHAR(100), @LastName NVARCHAR(100))

RETURNS NVARCHAR(200)

AS

BEGIN

RETURN @FirstName + ' ' + @LastName;

END;

GO

ALTER TABLE [User]

ADD FullName AS dbo.GetFullName(FirstName, LastName);

GO

COLUMN Data Encryption

-- First, create a master key

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'yourStrong(!)Password';

GO

-- Then, create a certificate

CREATE CERTIFICATE UserCertificate WITH SUBJECT = 'User Data Encryption';

GO

-- Next, create a symmetric key

CREATE SYMMETRIC KEY UserDataSymmetricKey

WITH ALGORITHM = AES\_256

ENCRYPTION BY CERTIFICATE UserCertificate;

GO

-- Finally, add the encrypted column

ALTER TABLE [User]

ADD PasswordEncrypted VARBINARY(256);

GO

-- Use the key to encrypt data as it is inserted/updated

OPEN SYMMETRIC KEY UserDataSymmetricKey

DECRYPTION BY CERTIFICATE UserCertificate;

UPDATE [User]

SET PasswordEncrypted = EncryptByKey(Key\_GUID('UserDataSymmetricKey'), Password);

GO

CLOSE SYMMETRIC KEY UserDataSymmetricKey;

Non clustered indexes

-- Non-clustered index on the User table

CREATE NONCLUSTERED INDEX IDX\_User\_LastName

ON [User] (LastName);

GO

-- Non-clustered index on the PlayerStatistics table

CREATE NONCLUSTERED INDEX IDX\_PlayerStatistics\_Score

ON PlayerStatistics (Score);

GO

-- Non-clustered index on the Transaction table

CREATE NONCLUSTERED INDEX IDX\_Transaction\_Amount

ON Transaction (Amount);

GO