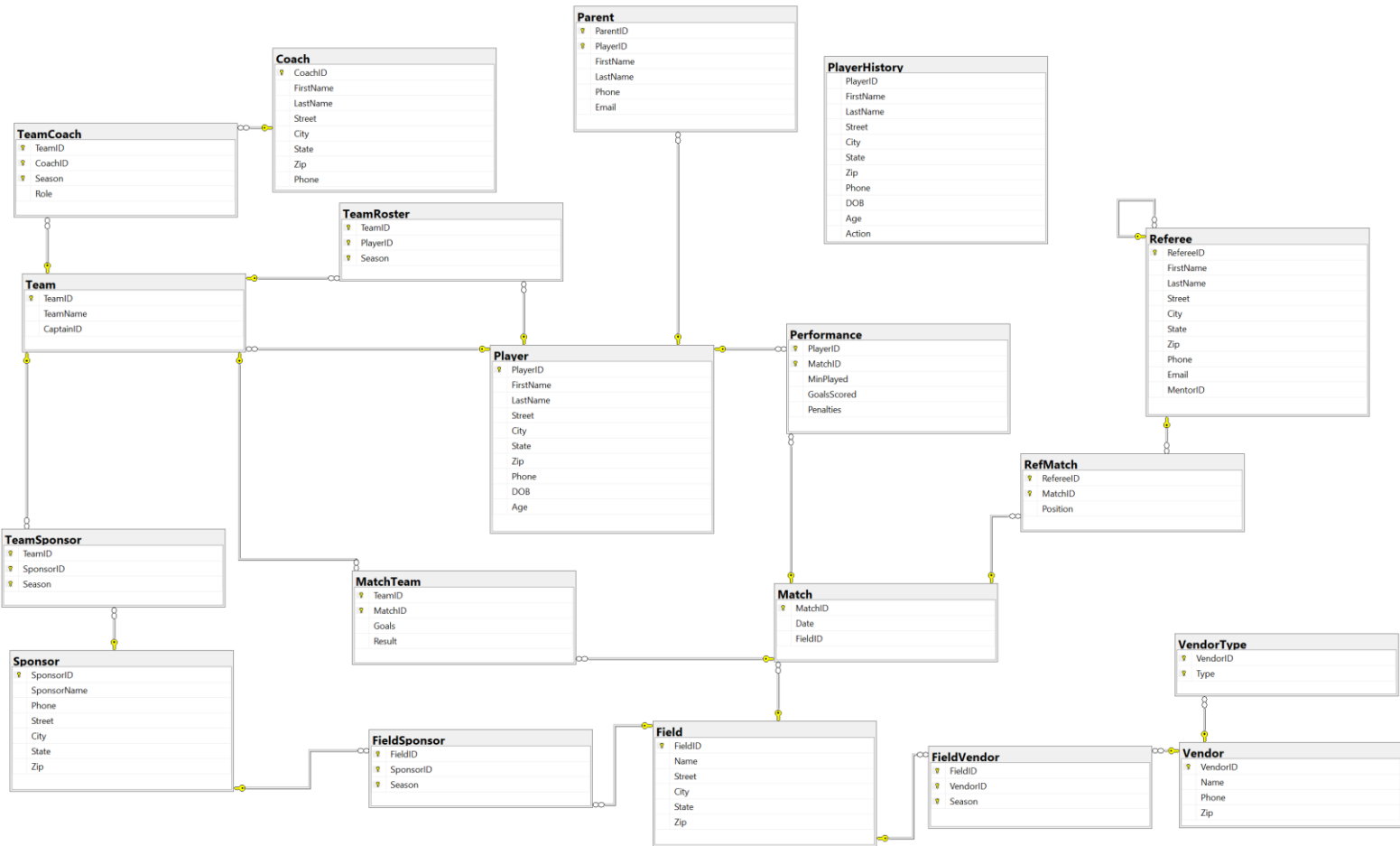


2) Soccer League Database Diagram



CHRISTIAN\SQLSERVER (SQL Server 15.0.2000 - CHRISTIAN\chris)

- Databases
 - System Databases
 - Database Snapshots
 - castilloc_CMD
 - Hospital
 - ScottsBar_CC
 - Soccer League
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Coach
 - Columns
 - Keys
 - Constraints
 - CK_Coach_Phone
 - CK_Coach_State
 - CK_Coach_Zip
 - DF_Coach_Phone
 - DF_Coach_State
 - Triggers
 - Indexes
 - PK_Coach (Clustered)
 - UN_Coach_Phone (Unique, Non-Clustered)
 - Statistics
 - dbo.Field
 - dbo.FieldSponsor
 - dbo.FieldVendor
 - dbo.Match
 - dbo.MatchTeam
 - dbo.Parent
 - dbo.Performance
 - dbo.Player
 - Columns
 - Keys
 - Constraints
 - CK_Player_Phone
 - CK_Player_State
 - CK_Player_Zip
 - DF_Player_Phone
 - DF_Player_State
 - Triggers
 - DML_Player_Delete_Trigger
 - DML_Player_Update_Trigger
 - Indexes
 - CIX_Player_Name_DOB (Non-Unique, Non-Clustered)
 - PK_Player (Clustered)
 - Statistics
 - dbo.PlayerHistory
 - dbo.Referee
 - dbo.RefMatch
 - dbo.Sponsor
 - dbo.Team
 - dbo.TeamCoach
 - Columns
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - CPK_TeamCoach (Clustered)
 - FI_TeamCoach_Role (Non-Unique, Non-Clustered, Filtered)

3) Screen captures of your expanded navigation panel showing your CKs, DFs, Indexes and UN Key for each table.

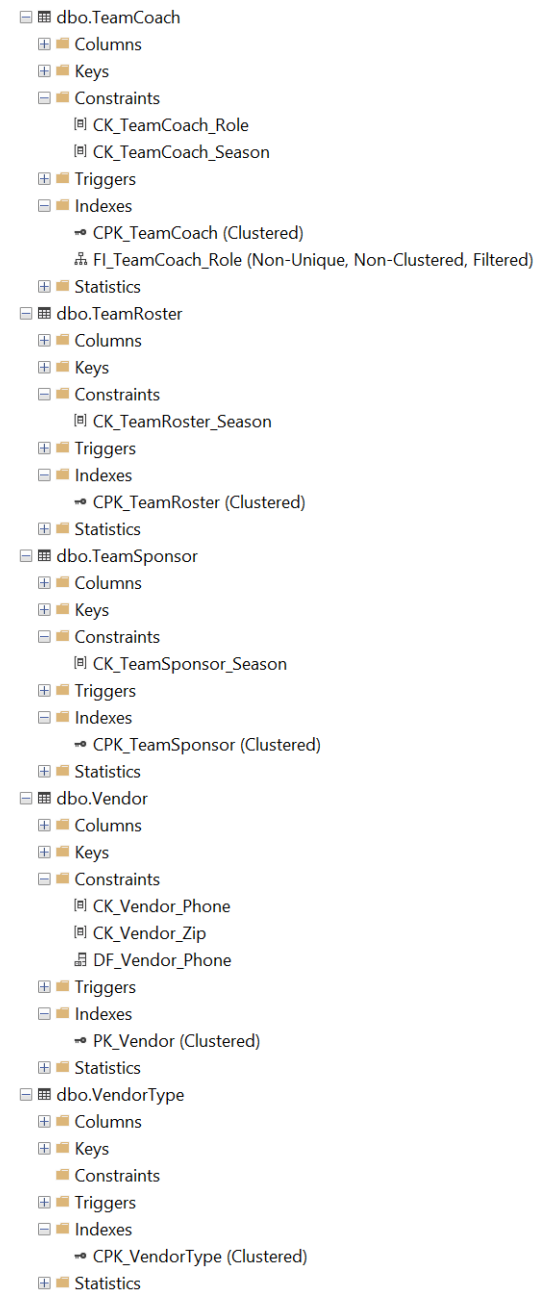
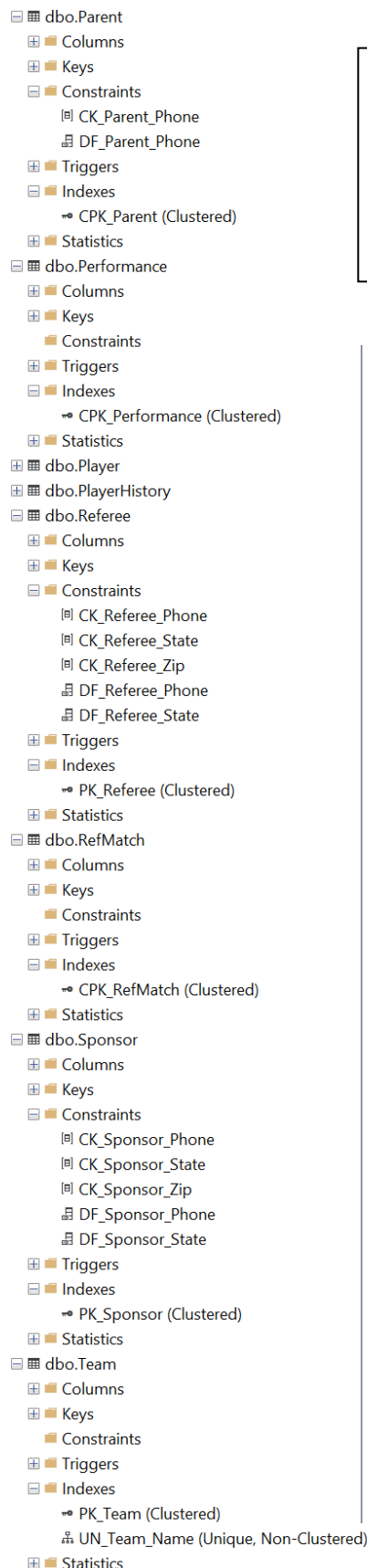
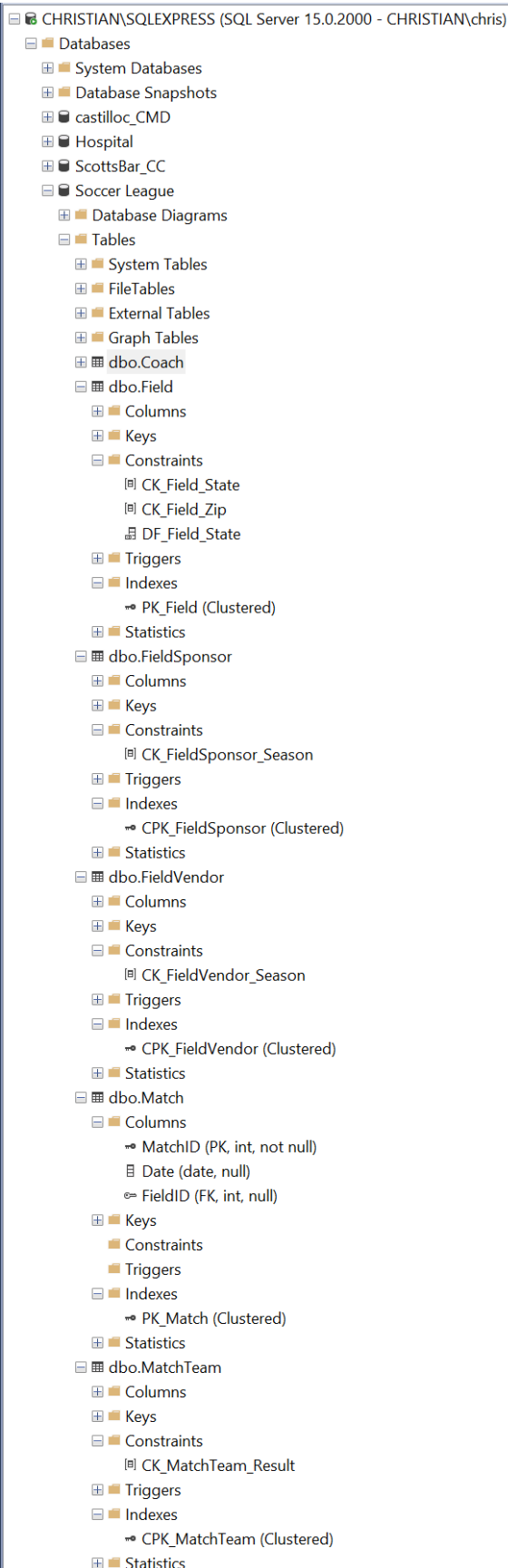
[Coach, Player, TeamCoach]

- Views
 - System Views
 - dbo.All_Teams_Captains_Coaches_VW
 - dbo.Teams_Coaches_Contact_VW
 - dbo.VendorName_Type_VW_index
 - Columns
 - vendor_name (nvarchar(30), not null)
 - vendor_type (nvarchar(30), not null)
 - Triggers
 - Indexes
 - CIX_VendorName_Type_VW (Clustered)
 - Statistics

4) Screen capture of your index on your view, showing columns of index.

3) Screen captures of your expanded navigation panel showing your CKs, DFs, Indexes and UN Key for each table.

-All other columns



```
/* 2. VIEW- List of all teams, all captains, and all coaches*/  
CREATE VIEW All_Teams_Captains_Coaches_VW  
AS  
SELECT --All teams  
    'Team' AS 'Category',  
    TeamName AS 'Name'  
FROM Team  
UNION  
SELECT --All captains  
    'Captain' AS 'Category',  
    CONCAT(P.FirstName, ' ', P.LastName) AS 'Name'  
FROM  
    Team T  
JOIN  
    Player P ON P.PlayerID=T.CaptainID  
UNION  
SELECT --All coaches  
    'Coach' AS 'Category',  
    CONCAT(FirstName, ' ', LastName) AS 'Name'  
FROM  
    Coach
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T04:04:29.7430547-04:00

```
/* 2. VIEW- List of all teams with their coaches and contact number*/  
CREATE VIEW Teams_Coaches_Contact_VW  
AS  
SELECT  
    T.TeamName AS 'team',  
    TC.Season AS 'season',  
    CONCAT(C.FirstName, ' ', C.LastName) AS 'team_coach',  
    C.Phone AS 'coach_contact_number'  
FROM  
    Team T  
JOIN  
    TeamCoach TC ON T.TeamID= TC.TeamID  
JOIN  
    Coach C ON TC.CoachID= C.CoachID
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T03:38:50.0428117-04:00

```
/* 1. SPROC- Player roster for a single team using team name as parameter.*/
```

```
CREATE PROC Player_Team_Roster_Para @TeamName varchar(30)
AS
BEGIN
SELECT
    T.TeamName AS 'team_name',
    (P.FirstName + ' ' + P.LastName) AS 'player_name'
FROM
    Team T
JOIN
    TeamRoster TR ON T.TeamID= TR.TeamID
JOIN
    Player P ON TR.PlayerID= P.PlayerID
WHERE
    (T.TeamName LIKE @TeamName + '%')
END
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T04:30:53.0940369-04:00

```
/* 2. SPROC- Teams and their coaches using 2 parameters, filtered for team name and for coach role.*/
```

```
CREATE PROC Team_Coaches_Role_Para @TeamName varchar(30), @CoachRole char(2)
AS
BEGIN
SELECT
    T.TeamName AS 'team_name',
    (C.FirstName + ' ' + C.LastName) AS 'coach_name',
    TC.Role AS 'coach_role'
FROM
    Team T
JOIN
    TeamCoach TC ON T.TeamID= TC.TeamID
JOIN
    Coach C ON TC.CoachID= C.CoachID
WHERE
    (T.TeamName LIKE @TeamName + '%') AND TC.role = @CoachRole
END
```

Programmability
Stored Procedures
System Stored Procedures
dbo.Player_Team_Roster_Para
dbo.Team_Coaches_Role_Para

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T04:48:23.2937670-04:00

```
/* 1. TRIGGER- For delete player, insert into Player history table.*/
```

```
CREATE TRIGGER Player_Delete_Trigger
ON Player
FOR DELETE
AS
BEGIN
    INSERT INTO PlayerHistory
    (PlayerID, FirstName, LastName, Street, City, State, Zip, Phone, DOB, Action)
    SELECT
        PlayerID, FirstName, LastName, Street, City, State, Zip, Phone, DOB, 'DELETED'
    FROM
        deleted;
END;
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T05:09:34.4682719-04:00

```
-- 2. TRIGGERS- Instead of delete team, print "you do not have permission to delete a team record."
```

```
CREATE TRIGGER DML_Delete_Team
ON Team
INSTEAD OF DELETE
AS
PRINT 'You do not have permission to delete a team record.'
ROLLBACK
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T05:22:40.4740494-04:00

```
/* 3. TRIGGER- For updated player, insert into Player history table.*/
```

```
CREATE TRIGGER DML_Player_Update_Trigger
ON Player
FOR UPDATE
AS
BEGIN
    INSERT INTO PlayerHistory
    (PlayerID, FirstName, LastName, Street, City, State, Zip, Phone, DOB, Action)
    SELECT
        PlayerID, FirstName, LastName, Street, City, State, Zip, Phone, DOB, 'UPDATED'
    FROM
        deleted;
END;
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T05:15:48.7637316-04:00

```
-- 4. TRIGGERS- PlayerHistory table has an updated/deleted column [action]
```

```
SELECT *
FROM PlayerHistory
```

100 %

Results Messages

PlayerID	FirstName	LastName	Street	City	State	Zip	Phone	DOB	Age	Action
----------	-----------	----------	--------	------	-------	-----	-------	-----	-----	--------

5) Screen capture of your deleted and updated rows in your history tables. [under action column]

▼ (General)	
Columns	FirstName (ASC), LastName (ASC), DOB (ASC)
Is Unique	No
Type	Index
▼ Identity	
(Name)	CIX_Player_Name_DOB
Description	
▼ Table Designer	
Create As Clustered	No
> Data Space Specification	PRIMARY
> Fill Specification	
Ignore Duplicate Keys	No
Included Columns	

```
-- 2. Index- Coach Assignment Filtered Index
CREATE NONCLUSTERED INDEX FI_TeamCoach_Role
ON TeamCoach(Role)
WHERE Role = 'AC'
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T05:47:21.0139196-04:00

▼ (General)	
Columns	Phone (ASC)
Is Unique	Yes
Type	Index
▼ Identity	
(Name)	UN_Coach_Phone
Description	
▼ Table Designer	
Create As Clustered	No
> Data Space Specification	PRIMARY
> Fill Specification	
Ignore Duplicate Keys	No
Included Columns	

```
-- 4. Index- One of your views must be an indexed view with "_index" at end of view name.
```

```
CREATE VIEW VendorName_Type_VW_index with schemabinding
AS
SELECT
    V.Name 'vendor_name', VT.Type 'vendor_type'
FROM
    dbo.Vendor V
JOIN
    dbo.VendorType VT ON V.VendorID= VT.VendorID
```

100 %

Messages

Commands completed successfully.

Completion time: 2024-03-14T06:08:53.9284634-04:00