Atom Indoor Stadium

Brief Contents

Atom Indoor Stadium

The project idea is mainly designed to involve youth and families in active living instead of addiction to video games or engaging in some other toxic ways to deplete dopamine.

Especially, young people are the targeted customers or users in this project. Our mission is to provide an Indoor games/activities facility to residents which will enhance recreation and leisure activities among the community members.
Providing a well-structured database for organizing time slots, courts, and equipment for playing is the goal of the project. As the project is fully based on data provided by users, there is no need for surveying or data collection from third parties. Firstly, using application software, people can register as a team or individual. Then, the user should be able to select a game or sport. For example, football, volleyball, swimming, athletics, badminton, tennis, etc. After Collecting details about players such as name, contact details, and age, the Player will select a time slot and court to play. Finally, Provide the ID number after successfully completing the payment process. Managing employees of each court are additional features that can be inserted into the database finally. This helps to function the stadiums efficiently, and it can also improve discipline among people.

Purpose of Database

The primary objective of preparing a database model is to enable the users to create, update, and maintain data through a variety of tools. Database systems are usually designed to handle large volumes of data that are difficult to manage manually. A flawlessly designed database makes access to data efficient and effective. The data is collected and maintained in a logical and structured way that can be easily analyzed.

At Atom Indoor Stadium, we provide many amenities like Fitness, Recreation, and Aquatics and aim at promoting healthy and active living. In order to better serve our community members, we require an operational database to enhance our data storage and support user ability to search programs and activities online.

We need an efficient procedure to store information regarding membership types, rates, services, and programs that we offer, membership profiles and staff directories etc.
We would like to enable members to register themselves online for various programs, book available courts, and to online pay membership dues. We would also like to maintain records of members' medical conditions (If any) and emergency contacts.

The Atom Indoor Stadium database should satisfy the following requirements.

First and foremost, members must be given an opportunity to go through the ***membership types*** and services offered at Atom Indoor Stadium. They should be able to pick the membership type that suits them best in terms of fees and offerings. Here, we offer different types of memberships such as Family, kids, youth, and fitness memberships. Each type has a definite rate and caters to diverse age groups.

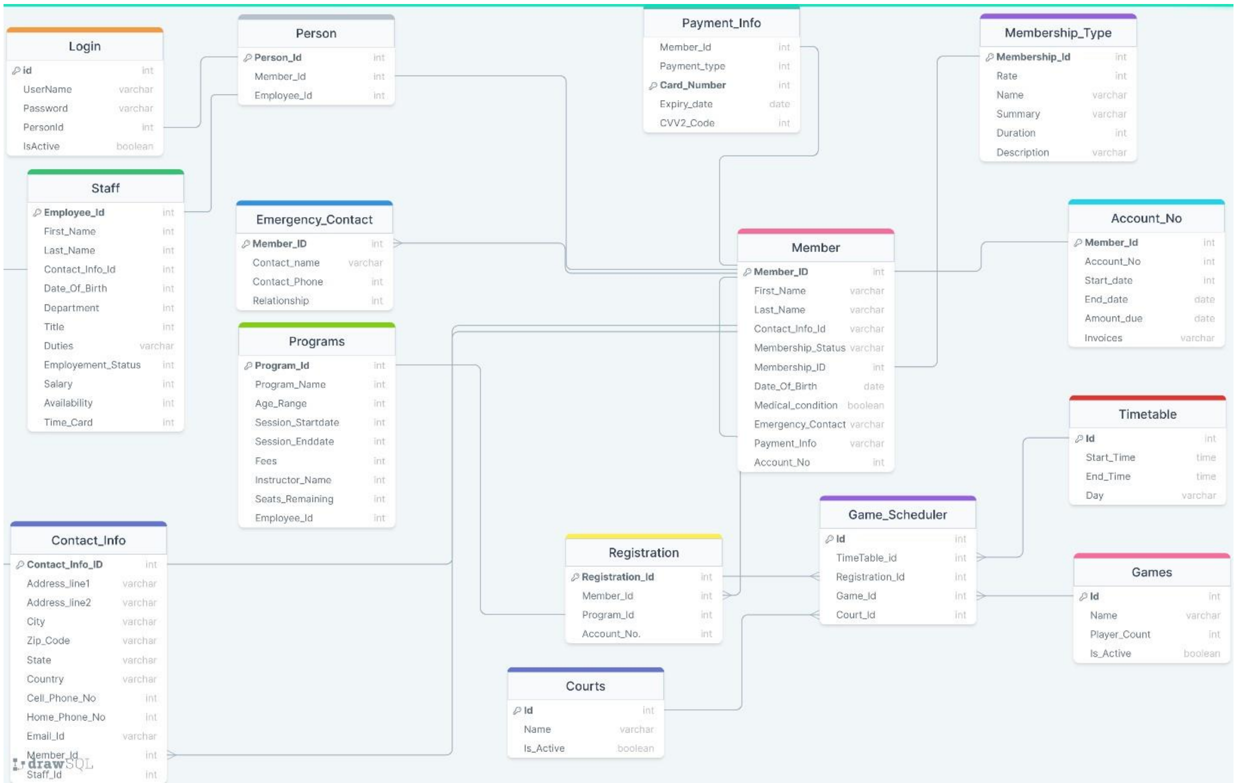
Then, there is a call to gather and sustain all the details regarding our ***members***, whether active, inactive, or day pass members. Member details should include their names, contact info, DOBs, emergency contacts, and medical history if any. Member details must indicate the active period, or any fees pending. A member should be notified via text when their due date for repayment arrives to renew the membership. To enable our members to drop in bookings online for various sports like badminton, basketball, picket balls, etc., we intend to ensure a proper structure to provide the availability of courts and the limitations.

Additionally, to ***register for programs***, we should give features concerning programs to our members online which include program name, age range, session start to end date, fees, session duration, weekly schedule, Instructor information, capacity or seats remaining, payment options, etc. We plan to provide many activities for several age groups throughout the year like Zumba, karate, Yoga, and Swimming lessons.

There is also a necessity to maintain information about ***staff*** members. We need their **department** name, title, and contact info in the staff directory. Besides the directory, staff information should be collected while **onboarding** and for their weekly **scheduling**. Each file should contain necessary information like employment status, salary, duties, DOB, virtual timecard, and availability. Staff scheduling will facilitate the manager to adjust timetables and track overtime.

In the future, we have plans to introduce a mobile app as well to make the process more accessible.

The conceptual model with identifying entities, associated attributes, and entity relationships (Phase 1)



Login	
id	int
UserName	varchar
Password	varchar
PersonId	int
IsActive	boolean

Person	
Person_Id	int
Member_Id	int
Employee_Id	int

Payment_Info	
Member_Id	int
Payment_type	int
Card_Number	int
Expiry_date	date
CVV2_Code	int

Membership_Type	
Membership_Id	int
Rate	int
Name	varchar
Summary	varchar
Duration	int
Description	varchar

Staff	
Employee_Id	int
First_Name	int
Last_Name	int
Contact_Info_Id	int
Date_Of_Birth	int
Department	int
Title	int
Duties	varchar
Employement_Status	int
Salary	int
Availability	int
Time_Card	int

Emergency_Contact	
Member_ID	int
Contact_name	varchar
Contact_Phone	int
Relationship	int

Member	
Member_ID	int
First_Name	varchar
Last_Name	varchar
Contact_Info_Id	varchar
Membership_Status	varchar
Membership_ID	int
Date_Of_Birth	date
Medical_condition	boolean
Emergency_Contact	varchar
Payment_Info	varchar
Account_No	int

Account_No	
Member_Id	int
Account_No	int
Start_date	int
End_date	date
Amount_due	date
Invoices	varchar

Programs	
Program_Id	int
Program_Name	int
Age_Range	int
Session_Startdate	int
Session_Enddate	int
Fees	int
Instructor_Name	int
Seats_Remaining	int
Employee_Id	int

Timetable	
Id	int
Start_Time	time
End_Time	time
Day	varchar

Game_Scheduler	
Id	int
TimeTable_id	int
Registration_Id	int
Game_Id	int
Court_Id	int

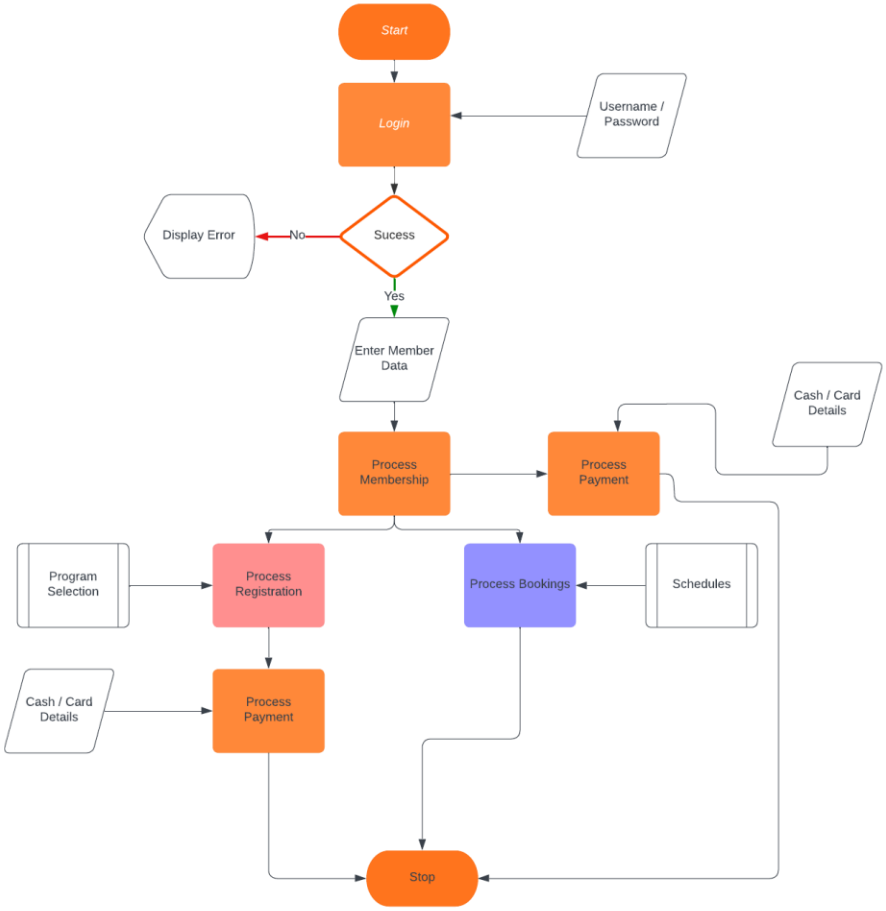
Contact_Info	
Contact_Info_ID	int
Address_line1	varchar
Address_line2	varchar
City	varchar
Zip_Code	varchar
State	varchar
Country	varchar
Cell_Phone_No	int
Home_Phone_No	int
Email_Id	varchar
Member_Id	int
Staff_Id	int

Registration	
Registration_Id	int
Member_Id	int
Program_Id	int
Account_No.	int

Games	
Id	int
Name	varchar
Player_Count	int
Is_Active	boolean

Courts	
Id	int
Name	varchar
Is_Active	boolean

Flow chart



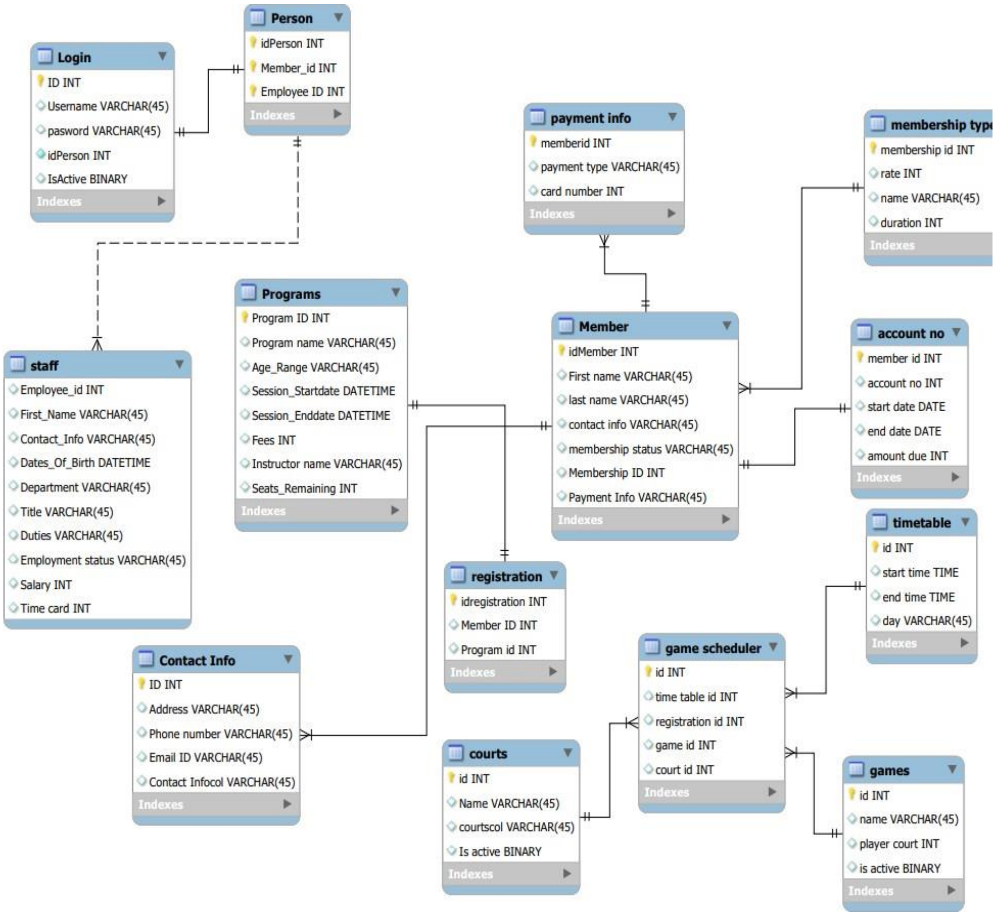
We are planning to prepare a database model, which enables users to create, update, and maintain their data through a variety of tools. Numbers, text, and dates are all the predominant data used in this system, so ***structured data*** are collected and maintained in a flawlessly designed database which helps to access data in more effective ways. Data models are classified as Enterprise Data Models and Project Data Models. Project Data Model is a more detailed view, so we are using this type of data model. We are using Relational Database technology, which involves tables (relations), also called entities and primary/foreign keys for executing relationships between entities.

Identified Attributes

Attributes can be defined as the properties or characteristics of an entity. It often corresponding to a field in a table. Metadata is nothing but data which describes the properties and context of user data. According to the business rules, properties are programmed for each attribute which helps to maintain the data quality. In our project, we have entities named as, Login, person, staff, payment info, member, membership type, Timetable, Game Scheduler, Registration, courts, programs, etc. These entities have relationships through possessing primary and foreign keys. Attributes and data types of the attributes are mentioned in the below mentioned diagram.

For example, Login is one of the entities in the project which consists of,

Attributes	**Data type**	**Properties**
ID	INT	It is primary key & not null.
Username	VARCHAR(45)	It accepts only 45 characters
password	VARCHAR(45)	It accepts only 45 characters
ID Person	INT	It is foreign key & not null
Is Active	BINARY	1- Active, 0- not active



Information stored/retrieved

Information provided by users and information about the Employee, Timetable, Courts are used in this system. Firstly, using application software, people can register as a team or individual. Those who are new to the system can enroll themselves by purchasing membership. User should select the game or sport. For example, football, volleyball, swimming, athletics, badminton, tennis, etc. Collecting details about players such as name, contact details, age, and emergency contact. Player will select time slot and court to play. Provide ID number after successfully completing the payment process.

So personal details of user, interest of the user (name of the sport), and details about the needed time slot, trainer and courts of the user are the information stored in the data base. This information can be updated / modified by the user with the help of user interface application.

Scaling Database

We are planning to be prepared for handling an increased workload that is beyond the capabilities of a single database instance. To make it more efficient, we have decided to leverage the benefits of scaling out our database instances by implementing Vertical scaling. The main advantages of choosing Vertical scaling are:

1. Simplicity and changing database instance size is easy in cloud information
2. Reduced server costs
3. Relatively minimal code changes and will be fast to implement
4. Fewer data consistency issues

But vertical scaling comes with its fair share of challenges, and we are designing our database in such a way to conquer such challenges easily.

Development of application

Installation and management cost is one of the important scales in developing database related system. Database administrators, system developers, data modeling and designing tools, User interface, application programs, Data Base Management System (DBMS), Repository, Database, and End users are the component of the system. Nowadays, speed or velocity of the process of storing and retrieving the data from the system plays vital role in Database Development Process. Managing people involved in the database development is significant factor of the process. As the collecting data from the users, is a major part of the workload of the system, the specified application is read heavy. MySQL is a traditional relational database managing software and it is known for its good performance in ***read-heavy workloads***.

In next phase, we further classified some more entities with their attributes, data types, length and key names.

Tables/Entities

Membership_Type:

Column Name	**Data Type**	**Length**	**Key Name**
Membership_Id	int		Primary Key
Name	varchar	50	
Rate	numeric		
Summary	varchar	200	
Duration	int		
Description	varchar	200	

Member:

Column Name	**Data Type**	**Length**	**Key Name**
Member_Id	int		Primary Key
First_Name	varchar	50	
Last_Name	varchar		
Membership_Status	bit		
Membership_Id	int		Foreign key
Date_Of_Birth	date		
Medical_Condition	bit		

Programs:

Column Name	**Data Type**	**Length**	**Key Name**
Program_Id	int		Primary Key
Program_Name	varchar	50	
Age_Range	int		

Session_Startdate	date		
Session_Enddate	date		
Fees	int		
Instructor_Name	varchar	100	
Seat_Remaining	int		
Employee_Id	int		Foreign key

Games:

Column Name	**Data Type**	**Length**	**Key Name**
Game_Id	int		Primary Key
Name	varchar	50	
Player_Count	int		
Is_Active	bit		

Payment_Info:

Column Name	**Data Type**	**Length**	**Key Name**
Id	int		Primary Key
Member_Id	int		Foreign key
Payment_type	varchar	50	
Card_Number	int		
Expiry_Date	date		
CVV2_Code	int		

Staff:

Column Name	**Data Type**	**Length**	**Key Name**
Employee_Id	int		Primary key
First_Name	varchar	50	
Last_Name	varchar	50	
Date_Of_Birth	date		
Department	varchar	20	
Title	varchar	20	
Duties	varchar	250	
Employment_Status	char	1	
Salary	int		
Availability	int		
Time_Card	int		

Table creation Scripts

Membership_Type:

```
CREATE TABLE [dbo].[Membership_Type](
        [Membership_Id] [int] IDENTITY(1,1) NOT NULL,
        [Name] [varchar](50) NOT NULL,
        [Rate] [numeric](18, 0) NOT NULL,
        [Summary] [varchar](200) NOT NULL,
        [Duration] [int] NOT NULL,
        [Description] [varchar](200) NOT NULL,
PRIMARY KEY CLUSTERED
(
        [Membership_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Member:

```
CREATE TABLE [dbo].[Member](
        [Member_Id] [int] IDENTITY(1,1) NOT NULL,
        [First_Name] [varchar](50) NOT NULL,
        [Last_Name] [varchar](50) NOT NULL,
        [Membership_Status] [bit] NOT NULL,
        [Membership_Id] [int] NOT NULL,
        [Date_Of_Birth] [date] NOT NULL,
        [Medical_Condition] [bit] NOT NULL,
PRIMARY KEY CLUSTERED
(
        [Member_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Member]  WITH CHECK ADD  CONSTRAINT [FK_Membership_Type_Member]
FOREIGN KEY([Membership_Id])
REFERENCES [dbo].[Membership_Type] ([Membership_Id])
GO
```

Programs:

```
CREATE TABLE [dbo].[Programs](
        [Program_Id] [int] IDENTITY(1,1) NOT NULL,
        [Program_Name] [varchar](50) NOT NULL,
        [Age_Range] [int] NOT NULL,
        [Session_Startdate] [date] NOT NULL,
        [Session_Enddate] [date] NOT NULL,
        [Fees] [int] NOT NULL,
        [Instructor_Name] [varchar](100) NOT NULL,
        [Seat_Remaining] [int] NOT NULL,
        [Employee_Id] [int] NOT NULL,
PRIMARY KEY CLUSTERED
 (
        [Program_Id] ASC
 )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
 ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
 [PRIMARY]
 ) ON [PRIMARY]
 GO

 ALTER TABLE [dbo].[Programs] WITH CHECK ADD CONSTRAINT [FK_Staff_Programs] FOREIGN
 KEY([Employee_Id])
 REFERENCES [dbo].[Staff]([Employee_Id])
 GO
```

Games:

```
CREATE TABLE [dbo].[Games](
       [Game_Id] [int] IDENTITY(1,1) NOT NULL,
       [Name] [varchar](50) NOT NULL,
       [Player_Count] [int] NOT NULL,
       [Is_Active] [bit] NOT NULL,
PRIMARY KEY CLUSTERED
(
       [Game_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Payment_Info:

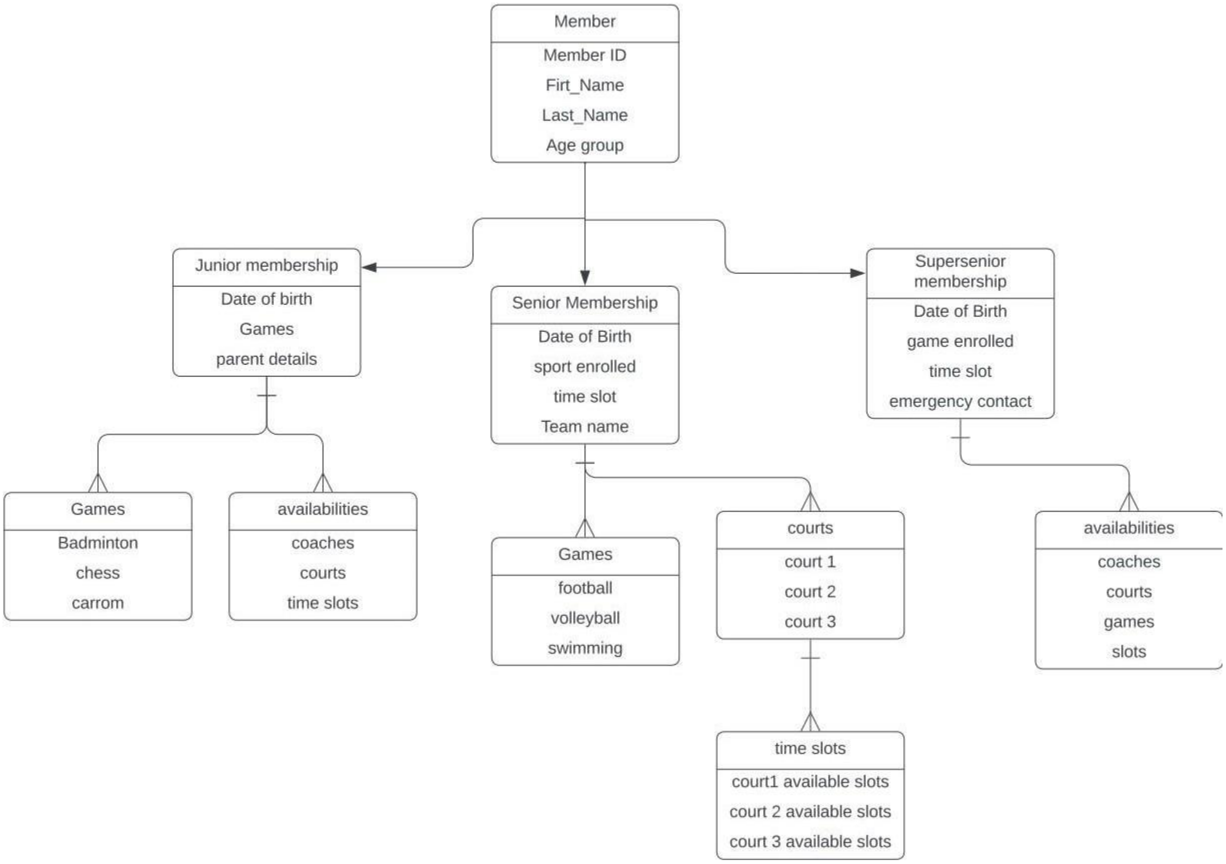
```
CREATE TABLE [dbo].[Payment_Info](
       [Id] [int] IDENTITY(1,1) NOT NULL,
       [Member_Id] [int] NOT NULL,
       [Payment_type] [varchar](50) NOT NULL,
       [Card_Number] [int] NOT NULL,
       [Expiry_Date] [date] NOT NULL,
       [CVV2_Code] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
       [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Payment_Info]  WITH CHECK ADD  CONSTRAINT [FK_Member_Payment_Info]
FOREIGN KEY([Member_Id])
REFERENCES [dbo].[Member] ([Member_Id])
GO
```

Staff:

```
CREATE TABLE [dbo].[Staff](
       [Employee_Id] [int] IDENTITY(1,1) NOT NULL,
       [First_Name] [varchar](50) NOT NULL,
       [Last_Name] [varchar](50) NOT NULL,
       [Date_Of_Birth] [date] NOT NULL,
       [Department] [varchar](20) NOT NULL,
       [Title] [varchar](20) NOT NULL,
       [Duties] [varchar](250) NOT NULL,
       [Employment_Status] [int] NOT NULL,
       [Salary] [int] NOT NULL,
       [Availability] [int] NOT NULL,
       [Time_Card] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
       [Employee_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
```

Enhanced ER Diagram

Sarath .r | October 29, 2022



Business Reports

Business rules are derived from policies, procedures, events, and functions of the business. Business rules are statements that define or constrain some aspect of the business. **Business needs** must be fulfilled by defining good business rules. In "Atom Indoor Stadium", business needs are simple. In this business, firstly system should collect the information (such as Name, Contact detail, interests etc.,) from the players, coaches. Secondly according to the availability of timesheet and courts, system helps the player to choose their court and time slot. Finally, providing the players with unique ID after completing the payment process.

Tables or entities with many columns or attributes are created according to the business needs. Then finding complex attributes in the entities and forming separate entity for each complex attributes, can be known as normalization. Normalization is the process of organizing data in a database, which includes creating tables and establishing relationships between those tables according to business rules. Below mentioned Unnormalized tables contains complex attributes such as Department, Activity, and Timesheet, these attributes can be normalized into separate entities. In other hand, attributes in the contact Info table, such as cell phone no, home phone no, email id, member id, staff id, can be separated or formed as new entity and address details can be normalized as another entity.

Reports needed to make strategic decisions

- Number of registrations occurred during each month of the year.
- To find the most famous membership type among the people, by number of enrollments in that membership type.
- Which time slot got more demand among people, by calculating rate of speed in which that time slot is filling.
- To find the most popular game or activity among the people, with help of number of registrations in particular time duration.
- Best performing coach.

Normalization

Table structure before normalization

Staff:

Column Name	**Data Type**	**Length**	**Key Name**
Employee_Id	int		Primary key
First_Name	varchar	50	
Last_Name	varchar	50	
Date_Of_Birth	date		
Department	varchar	20	
Title	varchar	20	

Duties	varchar	250	
Employment_Status	char	1	
Salary	int		
Activity	Varchar	250	
Time_Sheet	Varchar	250	

Contact_Info

Column Name	Data Type	Length	Key Name
Contact_Info_Id	int		Primary key
Address_Line1	varchar	50	
Address_Line2	varchar	50	
City	varchar	50	
Zip_Code	varchar	10	
State	varchar	50	
Country	varchar	50	
Cell_Phone_No	char	10	
Home_Phone_No	char	10	
Email_Id	varchar	50	
Member_Id	int		Foreign Key
Staff_Id	int		Foreign Key

Highlighted columns in the above tables are going to be normalized as shown below -

Normalized Tables:

Staff

Column Name	Data Type	Length	Key Name
Employee_Id	int		Primary key
First_Name	varchar	50	
Last_Name	varchar	50	
Date_Of_Birth	date		
Title	varchar	20	
Duties	varchar	250	
Employment_Status	char	1	
Salary	int		

About Entities and Attributes

Here, **Entities** of the “Atom Indoor Stadium database system” are stated as, Staff, Address, Contact Info, Staff address, Department, Activity, Timesheet, etc. Entities are classified into three types, namely, Strong entity, weak entity, Associative entity. **Attributes** are characteristics of an entity. They are columns of the entity, for an instance, attributes of the above mentioned “Staff” entity are Employee ID, First Name, Last Name, Date of Birth, Title, Duties, Employment Status, Salary. Properties of an Attribute can be classified as Identifier, Partial identifier, Composite or Multivalued attribute, Required and optional attribute, for an instance, Length, Data Type are metadata or properties of the attributes. Identifiers are helpful in building relationship among the entities according to the business needs.

Address

Column Name	**Data Type**	**Length**	**Key Name**
Adress_Id	int		Primary key
Address_Line1	varchar	50	
Address_Line2	varchar	50	
City	varchar	50	
Zip_Code	varchar	10	
State	varchar	50	
Country	varchar	50	

Contact_Info

Column Name	**Data Type**	**Length**	**Key Name**
Contact_Info_Id	int		Primary key
Cell_Phone_No	char	10	
Home_Phone_No	char	10	
Email_Id	varchar	50	
Member_Id	int		Foreign Key
Staff_Id	int		Foreign Key

Staff_Address

Column Name	**Data Type**	**Length**	**Key Name**
Staff_Address_Id	int		Primary key
Address_Id	int		Foreign Key
Staff_Id	int		Foreign Key
Is_Active	bit		

Department

Column Name	**Data Type**	**Length**	**Key Name**
Dept_Id	int		Primary key
[Dept_Name]	varchar	20	Foreign Key
Is_Active	bit		

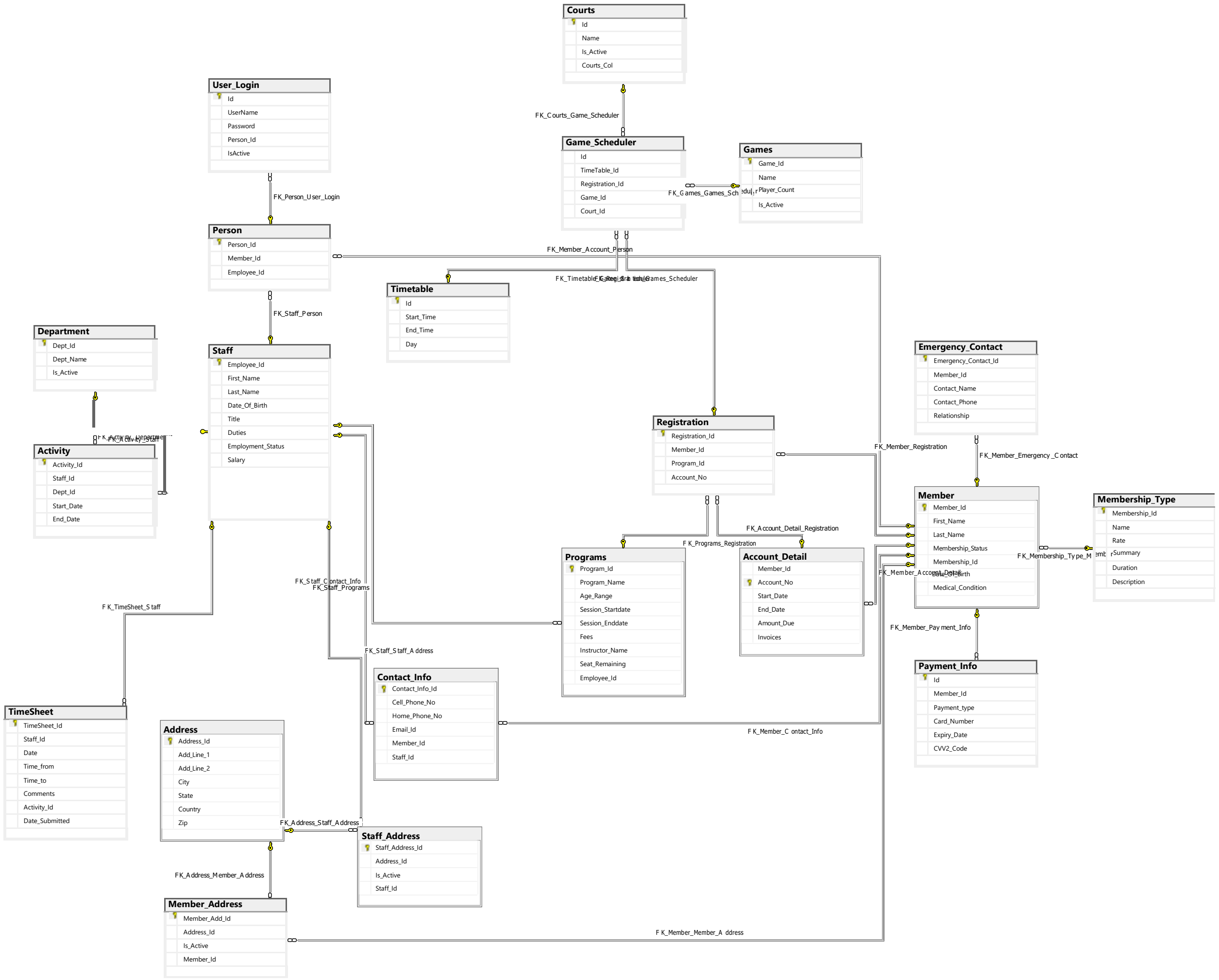
Activity

Column Name	Data Type	Length	Key Name
Activity_Id	int		Primary key
Staff_Id	int		Foreign Key
Dept_Id	int		Foreign Key
Start_Date	date		
End_Date	date		

Timesheet

Column Name	Data Type	Length	Key Name
TimeSheet_Id	int		Primary key
Staff_Id	int		Foreign Key
Activity_Id	int		Foreign Key
Date	date		
Time_from	time		
Time_to	time		
Comments	varchar	50	
Date_Submitted	date		

ER Diagram – SQL Server



In this ER diagram, each box is an entity, and attributes of the entity mentioned inside the box. Relationship among entities is represented by lines between them.

About Relationships

Relationship type is modeled as lines between the entities and the **relationship instance** is between specific entity instances or records or rows. Cardinality of relationships are classified as one-to-one, one-to-many, and many-to-many. Primary Keys and foreign keys are used to establish the relationship among entities, for an instance, from ER diagram consider an attribute “member-id” from the entity “member”. “Member id” is primary key of the entity, which is used as foreign key in many entities to establish relationship among them.

Please find below the scripts of newly created/edited entities as a result of Normalization:

Activity Entity:

```
USE [AtomIndoorStadium]
GO

/****** Object:  Table [dbo].[Activity]    Script Date: 11/7/2022 10:22:40 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Activity](
	[Activity_Id] [int] IDENTITY(1,1) NOT NULL,
	[Staff_Id] [int] NOT NULL,
	[Dept_Id] [int] NOT NULL,
	[Start_Date] [date] NOT NULL,
	[End_Date] [date] NOT NULL,
 CONSTRAINT [PK_Activity] PRIMARY KEY CLUSTERED
(
	[Activity_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Activity] WITH CHECK ADD CONSTRAINT [FK_Activity_Department] FOREIGN
KEY([Dept_Id])
REFERENCES [dbo].[Department] ([Dept_Id])
GO

ALTER TABLE [dbo].[Activity] CHECK CONSTRAINT [FK_Activity_Department]
GO

ALTER TABLE [dbo].[Activity] WITH CHECK ADD CONSTRAINT [FK_Activity_Staff] FOREIGN
KEY([Staff_Id])
REFERENCES [dbo].[Staff] ([Employee_Id])
GO

ALTER TABLE [dbo].[Activity] CHECK CONSTRAINT [FK_Activity_Staff]
GO

USE [AtomIndoorStadium]
GO
```

Department Entity:

```
/****** Object:  Table [dbo].[Department]    Script Date: 11/7/2022 10:23:57 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Department](
	[Dept_Id] [int] IDENTITY(1,1) NOT NULL,
	[Dept_Name] [varchar](20) NOT NULL,
	[Is_Active] [bit] NOT NULL,
 CONSTRAINT [PK_Department] PRIMARY KEY CLUSTERED 
(
	[Dept_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, 
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON 
[PRIMARY]
) ON [PRIMARY]
GO

USE [AtomIndoorStadium]
GO
```

TimeSheet Entity:

```
/****** Object:  Table [dbo].[TimeSheet]    Script Date: 11/7/2022 10:25:02 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[TimeSheet](
	[TimeSheet_Id] [int] IDENTITY(1,1) NOT NULL,
	[Staff_Id] [int] NOT NULL,
	[Date] [date] NOT NULL,
	[Time_from] [time](7) NOT NULL,
	[Time_to] [time](7) NOT NULL,
	[Comments] [varchar](50) NOT NULL,
	[Activity_Id] [int] NOT NULL,
	[Date_Submitted] [date] NOT NULL,
 CONSTRAINT [PK_TimeSheet] PRIMARY KEY CLUSTERED 
(
	[TimeSheet_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, 
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON 
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[TimeSheet] WITH CHECK ADD CONSTRAINT [FK_TimeSheet_Staff] FOREIGN 
KEY([Staff_Id])
REFERENCES [dbo].[Staff] ([Employee_Id])
GO

ALTER TABLE [dbo].[TimeSheet] CHECK CONSTRAINT [FK_TimeSheet_Staff]
GO
```

Address Entity:

```
USE [AtomIndoorStadium]
GO
```

```
/****** Object:  Table [dbo].[Address]    Script Date: 11/7/2022 10:26:18 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Address](
	[Address_Id] [int] IDENTITY(1,1) NOT NULL,
	[Add_Line_1] [varchar](50) NOT NULL,
	[Add_Line_2] [varchar](50) NOT NULL,
	[City] [varchar](20) NOT NULL,
	[State] [varchar](20) NOT NULL,
	[Country] [varchar](10) NOT NULL,
	[Zip] [varchar](10) NOT NULL,
PRIMARY KEY CLUSTERED
(
	[Address_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO


USE [AtomIndoorStadium]
GO
```

Member_Address Entity:

```
/****** Object:  Table [dbo].[Member_Address]    Script Date: 11/7/2022 10:32:01 AM
******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Member_Address](
	[Member_Add_Id] [int] NOT NULL,
	[Address_Id] [int] NOT NULL,
	[Is_Active] [bit] NOT NULL,
	[Member_Id] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
	[Member_Add_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Member_Address]  WITH CHECK ADD  CONSTRAINT
[FK_Address_Member_Address] FOREIGN KEY([Address_Id])
REFERENCES [dbo].[Address] ([Address_Id])
```

```
GO

ALTER TABLE [dbo].[Member_Address] CHECK CONSTRAINT [FK_Address_Member_Address]
GO

ALTER TABLE [dbo].[Member_Address]  WITH CHECK ADD  CONSTRAINT [FK_Member_Member_Address]
FOREIGN KEY([Member_Id])
REFERENCES [dbo].[Member] ([Member_Id])
GO

ALTER TABLE [dbo].[Member_Address] CHECK CONSTRAINT [FK_Member_Member_Address]
GO

USE [AtomIndoorStadium]
GO
```

Staff_Address Entity:

```
/****** Object:  Table [dbo].[Staff_Address]    Script Date: 11/7/2022 10:34:03 AM
******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Staff_Address](
       [Staff_Address_Id] [int] IDENTITY(1,1) NOT NULL,
       [Address_Id] [int] NOT NULL,
       [Is_Active] [bit] NOT NULL,
       [Staff_Id] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
       [Staff_Address_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Staff_Address]  WITH CHECK ADD  CONSTRAINT [FK_Address_Staff_Address]
FOREIGN KEY([Address_Id])
REFERENCES [dbo].[Address] ([Address_Id])
GO

ALTER TABLE [dbo].[Staff_Address] CHECK CONSTRAINT [FK_Address_Staff_Address]
GO

ALTER TABLE [dbo].[Staff_Address]  WITH CHECK ADD  CONSTRAINT [FK_Staff_Staff_Address]
FOREIGN KEY([Staff_Id])
REFERENCES [dbo].[Staff] ([Employee_Id])
GO

ALTER TABLE [dbo].[Staff_Address] CHECK CONSTRAINT [FK_Staff_Staff_Address]
GO
```

Contact_Info Entity:

```
USE [AtomIndoorStadium]
GO

/****** Object:  Table [dbo].[Contact_Info]    Script Date: 11/7/2022 10:38:15 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Contact_Info](
       [Contact_Info_Id] [int] IDENTITY(1,1) NOT NULL,
       [Cell_Phone_No] [int] NOT NULL,
       [Home_Phone_No] [int] NOT NULL,
       [Email_Id] [varchar](50) NOT NULL,
       [Member_Id] [int] NOT NULL,
       [Staff_Id] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
       [Contact_Info_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Contact_Info]  WITH CHECK ADD  CONSTRAINT [FK_Member_Contact_Info]
FOREIGN KEY([Member_Id])
REFERENCES [dbo].[Member] ([Member_Id])
GO

ALTER TABLE [dbo].[Contact_Info] CHECK CONSTRAINT [FK_Member_Contact_Info]
GO

ALTER TABLE [dbo].[Contact_Info]  WITH CHECK ADD  CONSTRAINT [FK_Staff_Contact_Info]
FOREIGN KEY([Staff_Id])
REFERENCES [dbo].[Staff] ([Employee_Id])
GO

ALTER TABLE [dbo].[Contact_Info] CHECK CONSTRAINT [FK_Staff_Contact_Info]
GO
```

Staff Entity:

```
USE [AtomIndoorStadium]
GO
```

```
/****** Object:  Table [dbo].[Staff]    Script Date: 11/7/2022 10:40:59 AM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Staff](
	[Employee_Id] [int] IDENTITY(1,1) NOT NULL,
	[First_Name] [varchar](50) NOT NULL,
	[Last_Name] [varchar](50) NOT NULL,
	[Date_Of_Birth] [date] NOT NULL,
	[Title] [varchar](20) NOT NULL,
	[Duties] [varchar](250) NOT NULL,
	[Employment_Status] [int] NOT NULL,
	[Salary] [int] NOT NULL,
PRIMARY KEY CLUSTERED
(
	[Employee_Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
```

Script to perform **Default Constraint** to insert a default value into a column(s) statement:

```
USE [AtomIndoorStadium]
GO

SELECT [Address_Id]
      ,[Add_Line_1]
      ,[Add_Line_2]
      ,[City]
      ,[State]
      ,[Country]
      ,[Zip]
  FROM [dbo].[Address]

GO

ALTER TABLE Address
ADD CONSTRAINT df_Country
DEFAULT 'United States' FOR Country;
```

Scripts to perform allocation of primary and foreign keys:

```
ALTER TABLE Member_Address
ADD CONSTRAINT FK_Member_Member_Address
FOREIGN KEY (Member_Id) REFERENCES Member (Member_Id);

ALTER TABLE Member_Address
ADD PRIMARY KEY (Member_Add_Id);
```

```sql
ALTER TABLE Staff_Address
ADD CONSTRAINT FK_Staff_Staff_Address
FOREIGN KEY (Staff_Id) REFERENCES Staff (Employee_Id);

ALTER TABLE Staff_Address
ADD PRIMARY KEY (Staff_Address_Id);
```