**Database Design & Implementation**

**Sport Club Management System**

**Student Name**

**Contents**

Objective………………………………………………………………………………………………3

Assumptions………………...……………….………………………………………………………3-4

System Design………………...……………………………..………….……………………..…….4-6

A UC Diagram for Sport Club………………...….………..……….……………………..…….4-6

Design Database.……...……………...……………………….…………………………...………...6-8

(ERM)……...……………...…………………………………………………….……………...7-8

Database Implementation in SQLite…...…………………………………………………………...9-16

Tables Schema Implementation….………………………………………………………...….9-11

Data Insertion Queries…........………………………..………………………………………11-13

Queries of Three SQL …..……...…………..…………………………………………….…13-16

Brief Summary……..…………………………………………………………………………………17

References……………………………………………………………………………………...……..17

**Objective**

Traditional method of record management for any organization is tedious process which takes lot of efforts and time. Also traditional method is not safe and does not offers longer period storage of records. This is also not enabling the backup and recovery of lost records in case of failure. Therefore, an alternative is required to address the limitations of traditional method. Though, the concept of database system is adapts for the record management. Database system (Buxton, 2009) is collections of elements which provide creation, deletion, modifications and monitoring of data records as well database schema of an organization. Database system advantages over the traditional method in many contexts such as efficient searching of records, permanent storage of data, backup & recovery and data access policy. Here, sport club depicts design for how users are perform different activities to access the automated booking system such as sport club members are required to sign up if they are new for system or login to access any service of sport club if they are existing member. Similarly sport club staffs play different roles for different purpose such as teacher who takes sport class of members and record attendance of class and so on. The user interaction interface design depicts by use case diagram (Gomaa, 2011). The objective of the work is to presents system design, database design and implementation for given case scenario i.e. sport club record system. In this work, sport club record system wants to present users interaction interface through unified modeling language, database design by entity relationship diagram and database implementation using structured query language in SQLite (Newman, 2005) software. Further, database schema design also represented for the sport club entities and their relationship type which offers storage data for sport club activities. The database design is represented through entity relationship model (ERM). After the design, database for sport club records is implemented in SQLite (Newman, 2005) using SQL queries. At last, design challenges, brief summary and references list are presented.

**Assumptions**

Database system for the sport club record management is able to meets various assumptions that formulated according to market business expectations. It also answers of certain queries raised from the internal and external users of sport club system such as sport club members, owners and staff. It is able to:

* Obtain the record of activity booking done by member on particular day.
* Find the number of booking done by member in a week.
* Find the number of teachers that name start with letter j or s.
* Find the number of sport activities class taken by a teacher.
* Retrieve information of members who have maximum attendance in a sport activity.
* Retrieve detail of sport activities those booked on weekend day.

**System Design**

To make automate system for sport club record management, system design is phase of development life cycle that ensure modeling of system components, process flow, data flow, user interactions, activities and more. There-fore, system design is accomplish via UML. UML (Gomaa, 2011) is a powerful language to model the system in different contexts. Unified Modeling Language offers various type of diagram that model number of things according to requirements. In this work, user interactions modeling is presents through diagram of use case. Diagram of Use case (Pender, 2003) of UML draw the possible actors which are interacts with system through different use cases. This section introduces a diagram of the application status of a sports club records management system.

**A Diagram of Use Case**

To create a usecase diagram (Pender, 2003) , initially all the actors (including external and internal) and use cases are identified. Then, it is also identifying use case by which an actor is interacting with system. In context of sport club record system, sport club members, staffs are actors and booking, search, update, view etc. are use cases. More about use cases and actors are given below. The name descriptions about use cases and actors defined in the tabular manner.

**Actor Table**

|  |  |
| --- | --- |
| Actor Name | Description |
| Sport\_Club\_Member | This actor is user of the sport club management system who wants avail set of system services. This is classified as new member and existing member that describe further. |
| New\_Member | It is subtype of sport\_club\_member who is new user that require enrolling or registering themselves with system by sign up process. |
| Existing\_Member | Existing\_member is another subtype of sport\_club\_member actor who is already users of system that require only sign in to avail system services. |
| Sport\_Club\_Staff | Sport\_Club\_Staff is an internal users who manage the different activities of system such as booking, searching, updating and recording attendance. This is also classified as club\_teacher and club\_instructor subactors. |
| Club\_Teacher | This actor is subtype of sport\_club\_staff who take sport activities classes as well as record the attendance of class. |
| Club\_Instructor | Club\_Instructor monitor booking of sport activities through search and view booking records as well as they may update their personal data. |

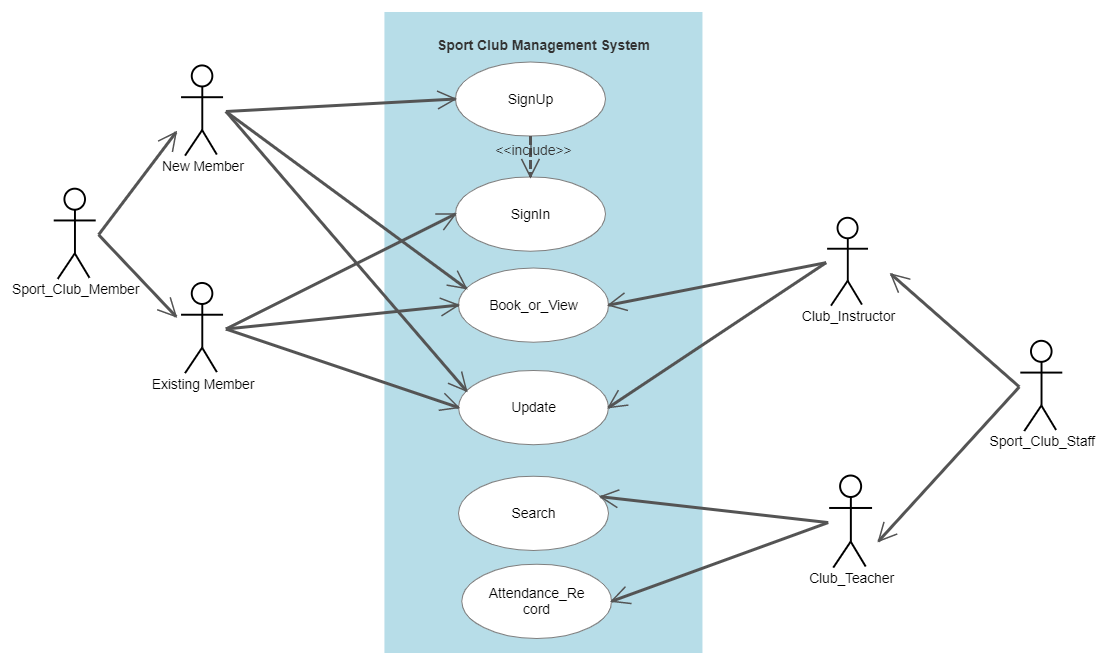
**Use Cases Table**

|  |  |
| --- | --- |
| Use Case Name | Description |
| SignUp | When new user wants to enroll or register with the system then SignUp use case is invoked. By this use case new users become member of sport club management system. After this users may avail any service. |
| SignIn | If user is already enrolled or registered with the system then they directly access service via login process. When user wants to enter in the system service then login actor is involved. |
| Book\_or\_View | This use case works when member of sport club either book new sport activity or view existing booking. It also used by club\_instructor to view booking of sport activity by a member. |
| Update | When sport club members as well as club instructors want to update their personal data then this use case works. |
| Search\_Activity | Search\_Activity use case works when club member want to search sport activities. |
| Attendance\_Record | When club teacher take sport activity class and wants to data attendance of class then this use case is invoked. |

In the above table, actors and uses cases for sport club management system are listed and described which identified from the information given in the sport\_club store system case scenario. With the help of these actors and use cases, a diagram of use case diagram is created. A visual form of sport club management system use case diagram is shown below.

**A Diagram of Use Case**

A diagram of use case (Pender, 2003)  shown below which represents several things about actors and use cases roles such as new users who want to become member of the sport-club they require to signup process. Suppose if users are already member then they may login with the system and may book or view sport activities as well as update their personal information. Similarly sport club teacher may search record attendance of class and sport activity. Sport club instructor may view activities booking. A use case diagram is below.



**Design of Database**

Database design (Chilson, 1983) is back end design phase of an information management system. This phase focuses on identification and representation of all entities that records are required to store in database. For database design, several models are exists such as entity relational model, relational model, network model, object oriented based model and many more. Each database model is used according to application and requirement of the database system. In this, section database design for sport club record system named as sport club management system is presented. Here, entity relationship model (ERM) is used to represent database design for given case scenario.

**Entity Relationship Model (ERM)**

Entity relationship model (ERM) is one among of several database design models. ERM is visual representation of database schema where it use sort of visual-symbols such as single line rectangle for strong entity, oval for attributes, diamond for relationship type and many more. Though all the components of database like entity, their attributes and relationship type is represents by the well-defined symbols. An entity relationship diagram for the management system of sport club is depicted below that comprises of four entities which are related to each other on the basis of some role or job. A complete entity relationship diagram describes in context of entity name and relationship name.

**Sport\_Class\_Activity Entity:**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Attribute Type | Integrity Constraints |
| Class\_Code | Varchar | Atomic | Primary Key |
| Class\_Title | Character | Atomic | - |
| Day | Character | Atomic | - |
| Time | Time | Atomic | - |

**Sport\_Club Entity :**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Attribute Type | Integrity Constraints |
| Club\_No | Varchar | Atomic | Primary Key |
| Name | Character | Atomic | - |
| Location | Character | Atomic | - |
| Contact\_No | Integer | Multi-valued | - |

**Sport\_Club\_Member Entity :**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Data type | Attribute Type | Integrity Constraints |
| Member\_Ref\_No | Varchar | Atomic | Primary Key |
| FirstName | Character | Atomic | - |
| Surname | Character | Atomic | - |
| Address | Character | Composite | - |
| Telephone\_Number | Integer | Multi-valued | - |
| Email | Character | Atomic | - |
| DoB | Date | Atomic | - |
| MedicalStatus | Character | Atomic | - |

**Sport\_Club\_Staff Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute Name | Datatype | Attribute Type | Integrity Constraints |
| Staff\_No | Varchar | Atomic | Primary Key |
| FirstName | Character | Atomic | - |
| Surname | Character | Atomic | - |
| Role | Character | Atomic | - |
| Staff\_No | Varchar | Atomic | Foreign Key |

**Relationship Name Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Relationship Name | Entity Name 1 | Entity Name 2 | Cardinality Ratio | Participation |
| Book\_or\_view | Sport\_Club\_Member | Sport\_Class\_Activity | M:M | Partial |
| Sign Up | Sport\_Club\_Member | Sport\_Club | M:1 | Partial |
| Search | Sport\_Club\_Staff | Sport\_Class\_Activity | 1:M | Partial |
| AttendanceRecord | Sport\_Club\_Staff | Sport\_Class\_Activity | M:M | Partial |
| Work\_for | Sport\_Club\_Staff | Sport\_Club | M:1 | Full |

**Entity Relationship Diagram (ERD)**

Entity relationship diagrams for the sport club management system shows sport club member, sport club, sport activity class and sport club staff entities and their attributes as well as relationship type. In the ERD, primary key of each entity also depicted that ensure unique integrity constraints. Along with this, cardinality ratio and participation of entities for a relationship also represent in the ERD. Below figure show ERD for the sport club management system.

****

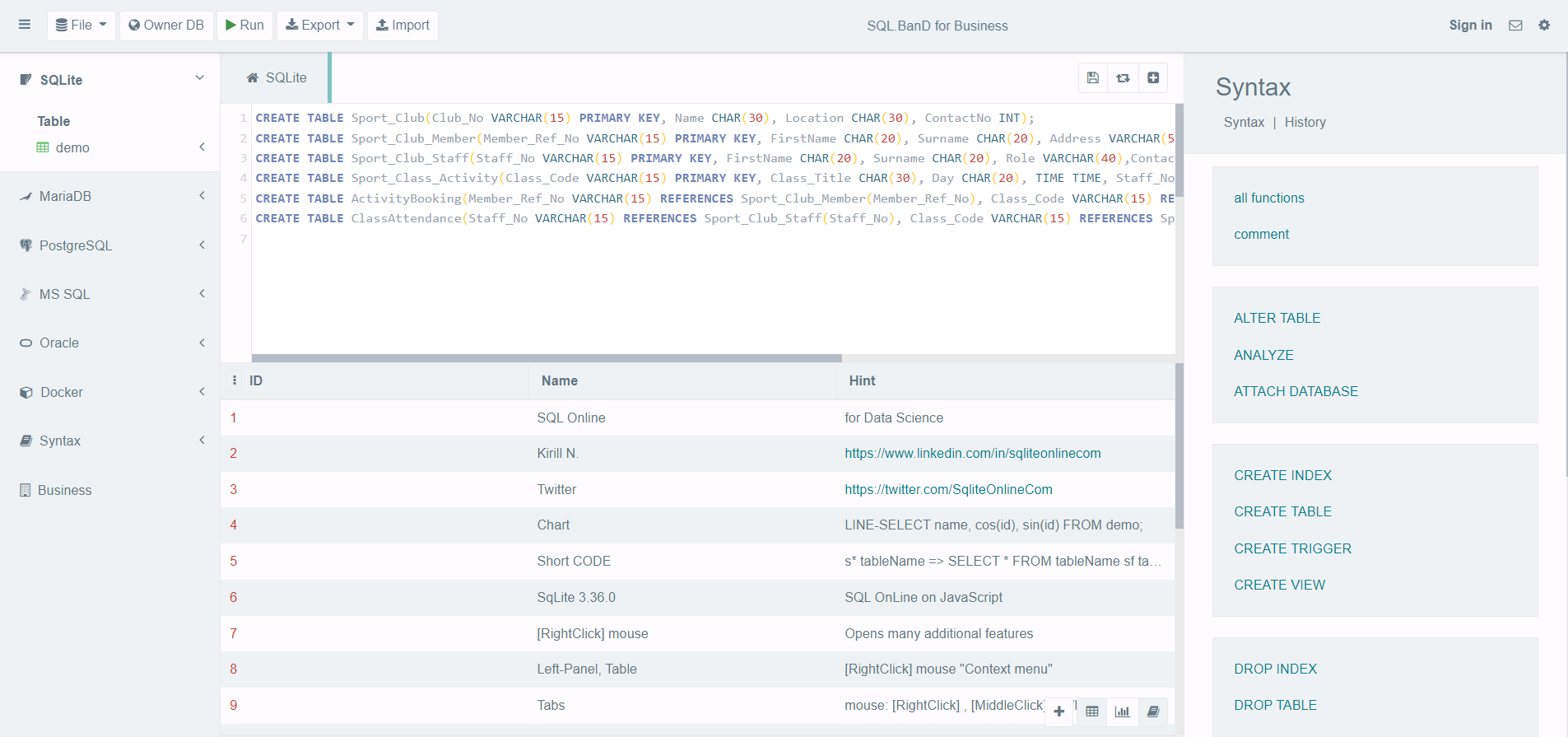
**Database Implementations**

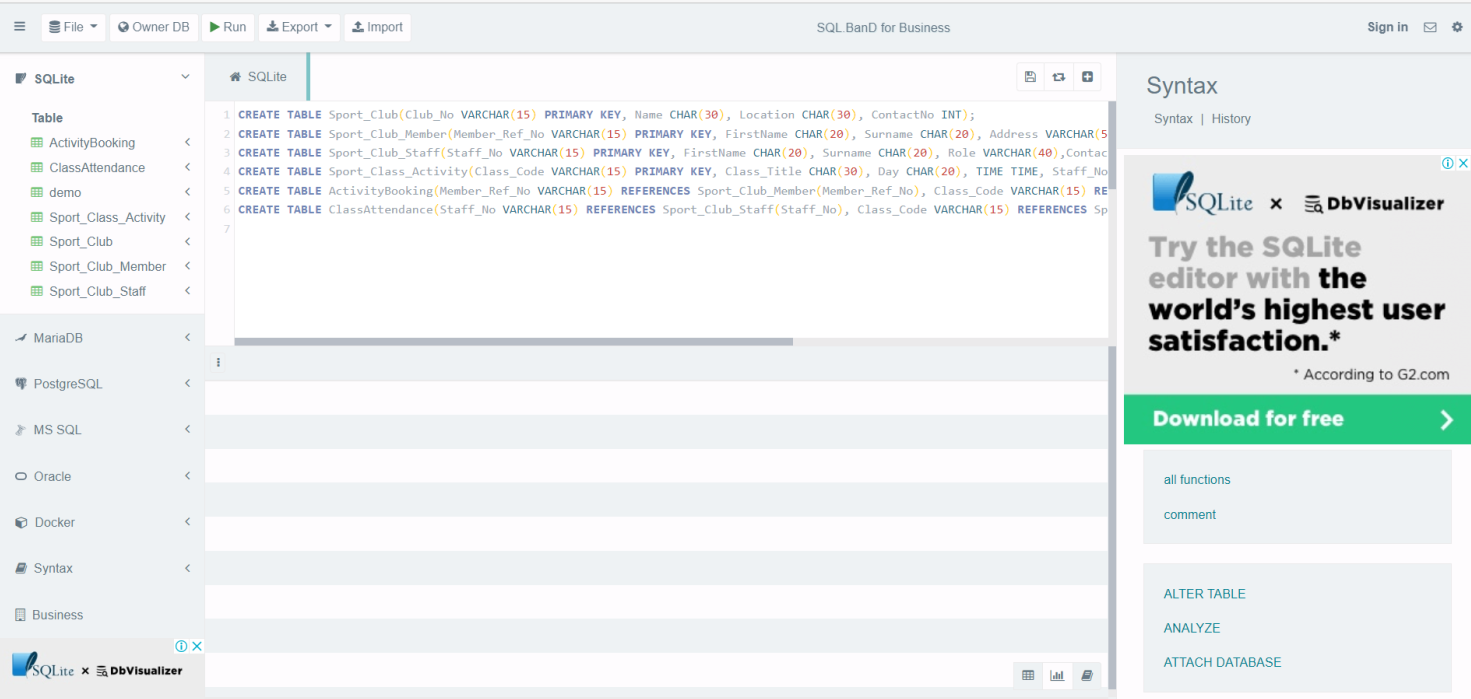
After the database design, implementation is initiated using DBMS (Pathak, 2007) software. DBMS software offers construction, storage, modifications and management of database schema as well as records. For the sport club management system, SQLite (Newman, 2005)  online platform is used to implement database. To do this, structured query language (SQL) statements are used which enable to create, update and delete database tables and storing of data in the tables. In this, DDL and data view language (DVL) components of SQL is used. For creating tables, CREATE statement is uses and for inserting data INSERT statement is used. Further to retrieve the data records, SELECT statement is used.

**Table Schema Implementations**

CREATE command is used. For creating table schema and integrity constraint such unique integrity like primary key as well referential integrity like foreign key. Later then schema creation, data are inserted into tables using INSERT command. The SQL command for both is shows below.

* **create table Sport\_Club(Club\_No varchar(15) primary key, Name char(30), Location char(30), ContactNo int);**
* **create table Sport\_Club\_Member(Member\_Ref\_No varchar(15) PRIMARY key, FirstName char(20), Surname char(20), Address varchar(50),Telephone\_No int, Email varchar(20),DoB date, MedicalStatus varchar(15));**
* **create table Sport\_Club\_Staff(Staff\_No varchar(15) PRIMARY key, FirstName char(20), Surname char(20), Role varchar(40),Contact\_Number int);**
* **create table Sport\_Class\_Activity(Class\_Code varchar(15) PRIMARY key, Class\_Title char(30), Day char(20), Time time, Staff\_No varchar(15) REFERENCES Sport\_Club\_Staff(Staff\_No));**
* **create table ActivityBooking(Member\_Ref\_No varchar(15) REFERENCES Sport\_Club\_Member(Member\_Ref\_No), Class\_Code varchar(15) REFERENCES Sport\_Class\_Activity(Class\_Code), Book\_View char(15));**
* **create table ClassAttendance(Staff\_No varchar(15) REFERENCES Sport\_Club\_Staff(Staff\_No), Class\_Code varchar(15) REFERENCES Sport\_Class\_Actiity(Class\_Code));**



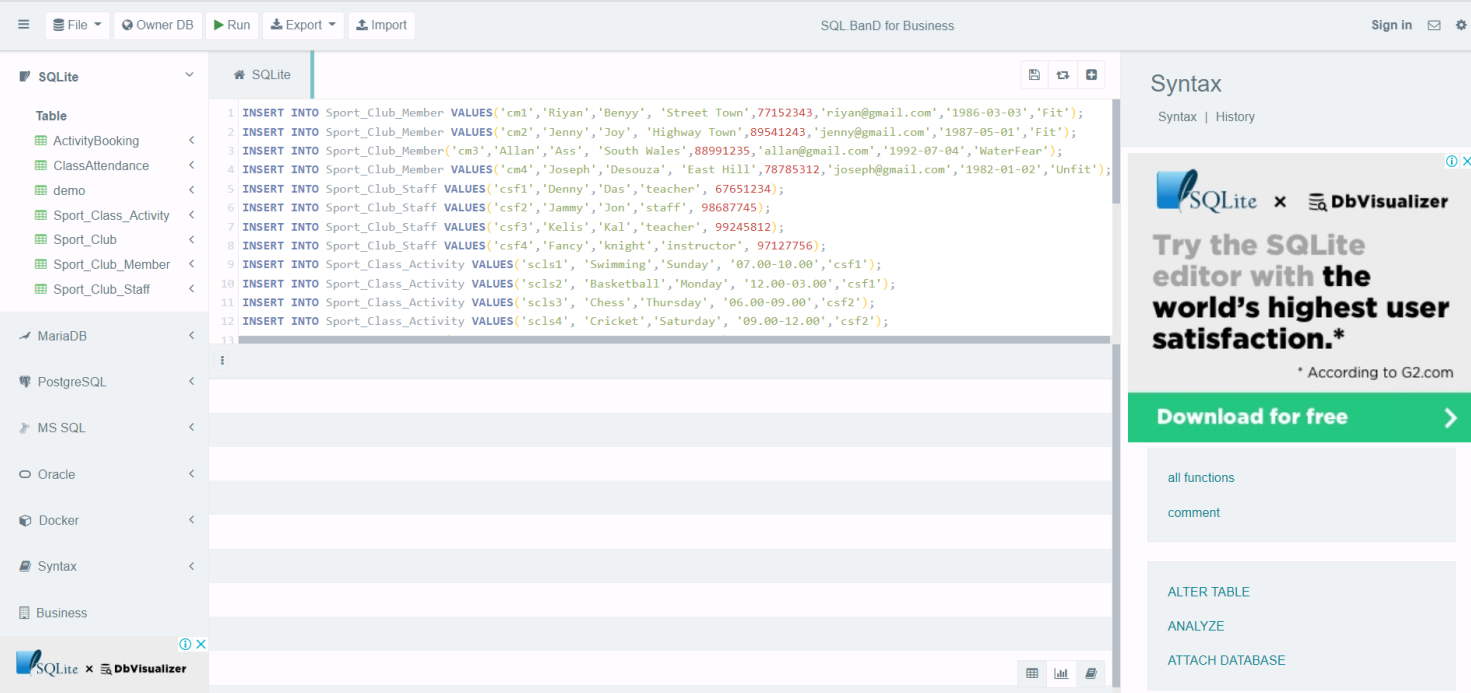


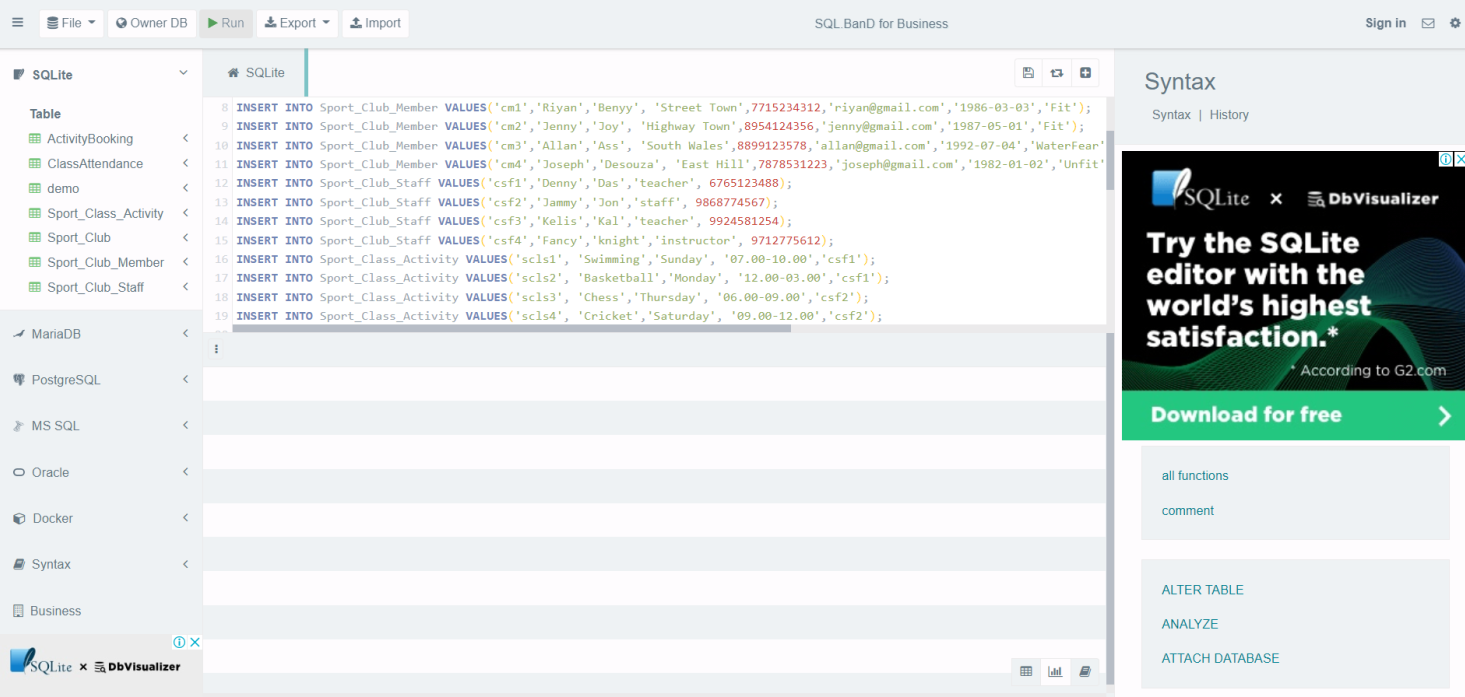
**Data Insertion Queries**

To answers of assumption formulated for the market need by the database design that requires inserting data in the tables using INSERT command. The query(SQL) for inserting data is written below.

* **insert into Sport\_Club\_Member values('cm1','Riyan','Benyy', 'Street Town',7715234312,'riyan@gmail.com','1986-03-03','Fit');**
* **insert into Sport\_Club\_Member values('cm2','Jenny','Joy', 'Highway Town',8954124356,'jenny@gmail.com','1987-05-01','Fit');**
* **insert into Sport\_Club\_Member values('cm3','Allan','Ass', 'South Wales',8899123578,'allan@gmail.com','1992-07-04','WaterFear');**
* **insert into Sport\_Club\_Member values('cm4','Joseph','Desouza', 'East Hill',7878531223,'joseph@gmail.com','1982-01-02','Unfit');**
* **insert into Sport\_Club\_Staff values('csf1','Denny','Das','teacher', 6765123488);**
* **insert into Sport\_Club\_Staff values('csf2','Jammy','Jon','staff', 9868774567);**
* **insert into Sport\_Club\_Staff values('csf3','Kelis','Kal','teacher', 9924581254);**
* **insert into Sport\_Club\_Staff values('csf4','Fancy','knight','instructor', 9712775612);**
* **insert into Sport\_Class\_Activity values('scls1', 'Swimming','Sunday', '07.00-10.00','csf1');**
* **insert into Sport\_Class\_Activity values('scls2', 'Basketball','Monday', '12.00-03.00','csf1');**
* **insert into Sport\_Class\_Activity values('scls3', 'Chess','Thursday', '06.00-09.00','csf2');**
* **insert into Sport\_Class\_Activity values('scls4', 'Cricket','Saturday', '09.00-12.00','csf2');**
* **insert into Sport\_Club values('sclb1', 'Fitfat','Town Side', 9999886622);**
* **insert into Sport\_Club values('sclb2', '24x7 Open','South Street', 7982331177);**
* **insert into Sport\_Club values('sclb3', 'Newyork club','Newyork', 89565827899);**
* **insert into ActivityBooking values('cm1','scls1','Book');**
* **insert into ActivityBooking values('cm2','scls2','Book');**
* **insert into ActivityBooking values('cm1','scls1','Book');**
* **insert into ActivityBooking values('cm2','scls2','Book');**
* **insert into ActivityBooking values('cm3','scls3','view');**
* **insert into ActivityBooking values('cm4','scls4','view');**
* **insert into ActivityBooking values('cm2','scls3','Book');**
* **insert into ActivityBooking values('cm3','scls4','view');**
* **insert into ActivityBooking values('cm4','scls1','Book');**
* **insert into ClassAttendance values ('csf1','scls1');**
* **insert into ClassAttendance values ('csf2','scls2');**
* **insert into ClassAttendance values ('csf1','scls1');**
* **insert into ClassAttendance values ('csf3','scls3');**
* **insert into ClassAttendance values ('csf2','scls2');**
* **insert into ClassAttendance values ('csf3','scls3');**

INSERT statements that written above have inserted few data records into each table of database. The execution of insert command in SQLite software is shows below.

****

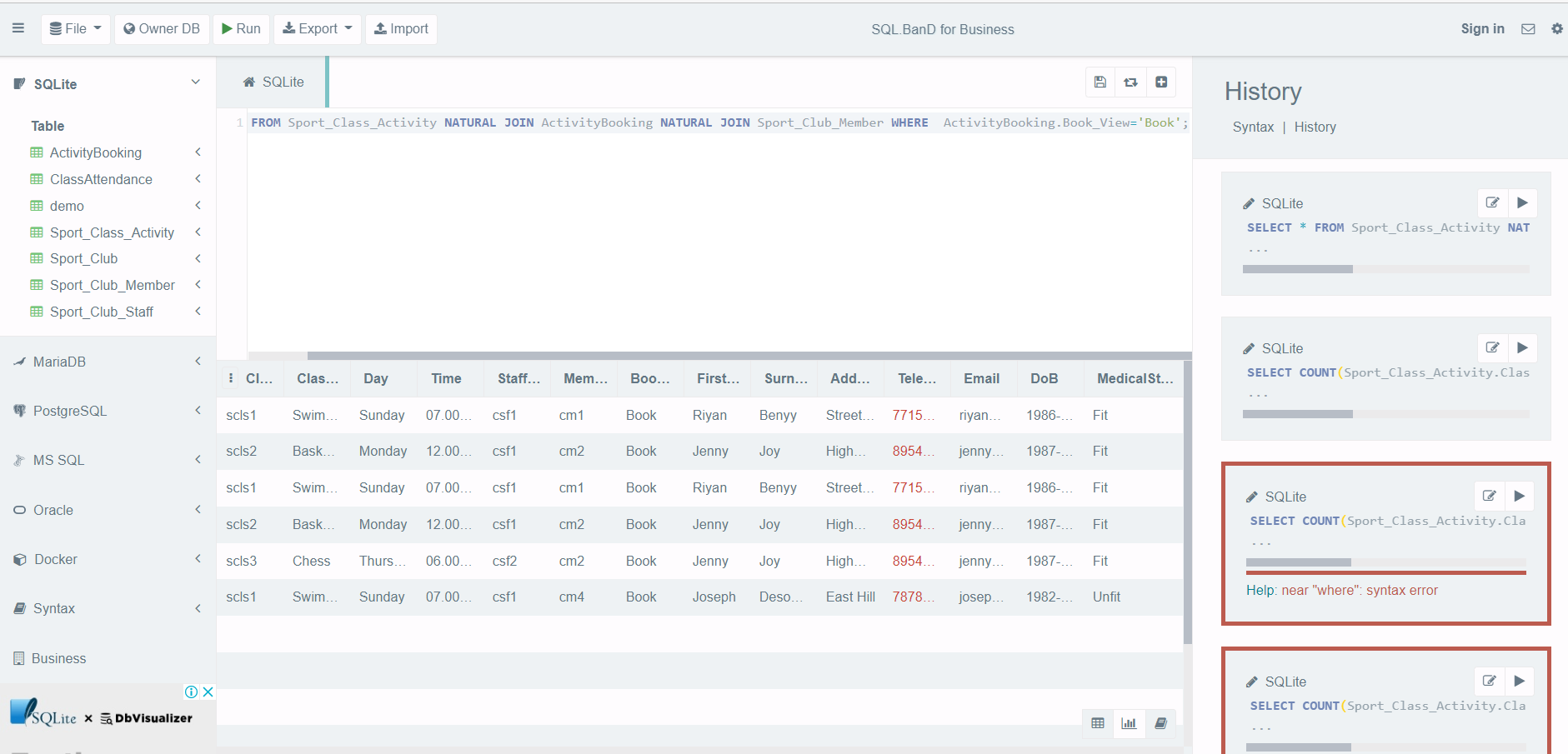
****

**Three SQL Queries**

After the creation of tables and insertions of data into tables, test whether designed and implemented database satisfy formulated assumptions according to market needs or not. Though, three SQL (Pathak, 2007) queries were written that used select statement to retrieve needed information from the sport club database. These queries answer some of business assumptions which are mentioned in previous section.

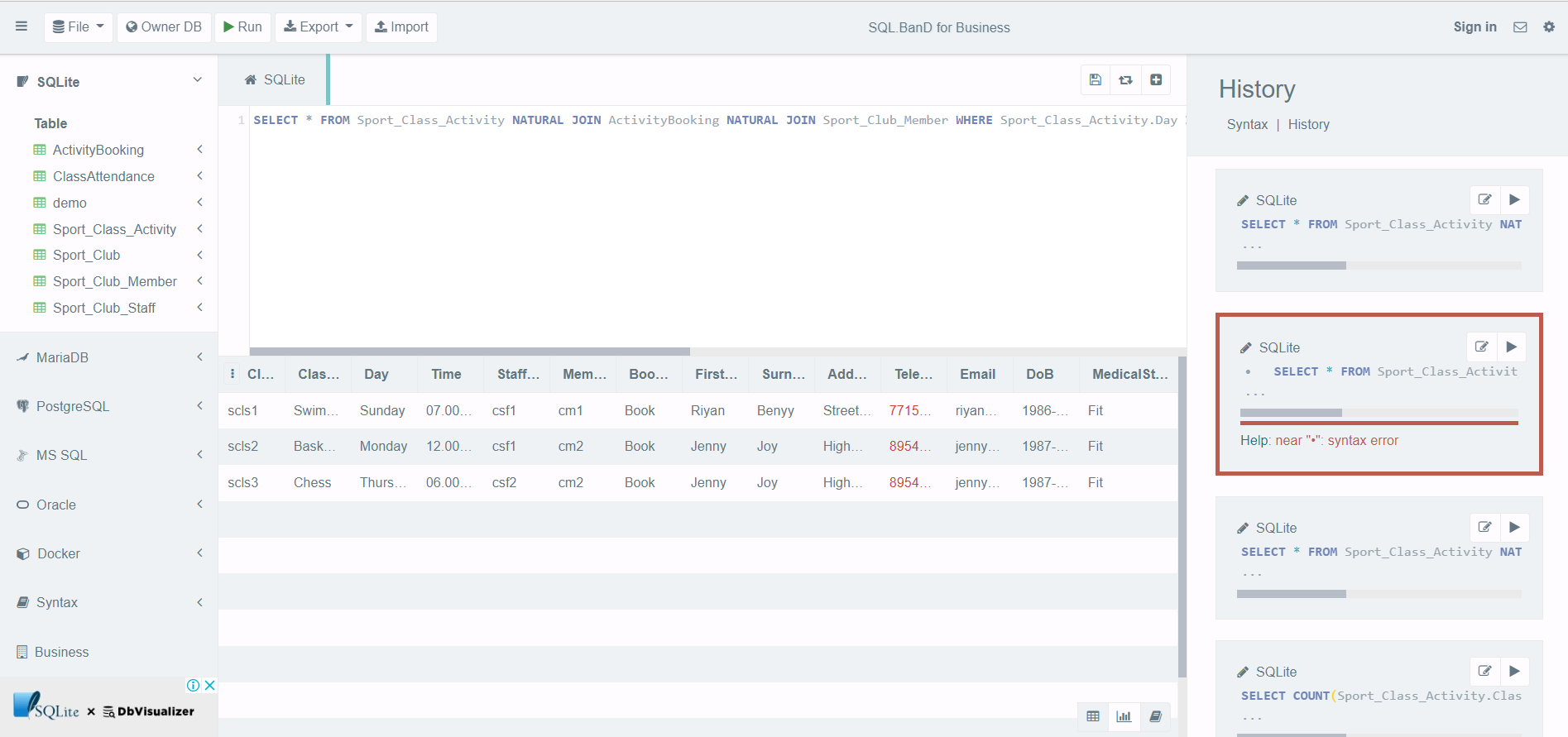
**SQL Query 1:** Write SQL query to one that shows how a sports club member can view their current bookings.

* **select \* from Sport\_Class\_Activity natural join ActivityBooking NATURAL join Sport\_Club\_Member where ActivityBooking.Book\_View='Book';**

******

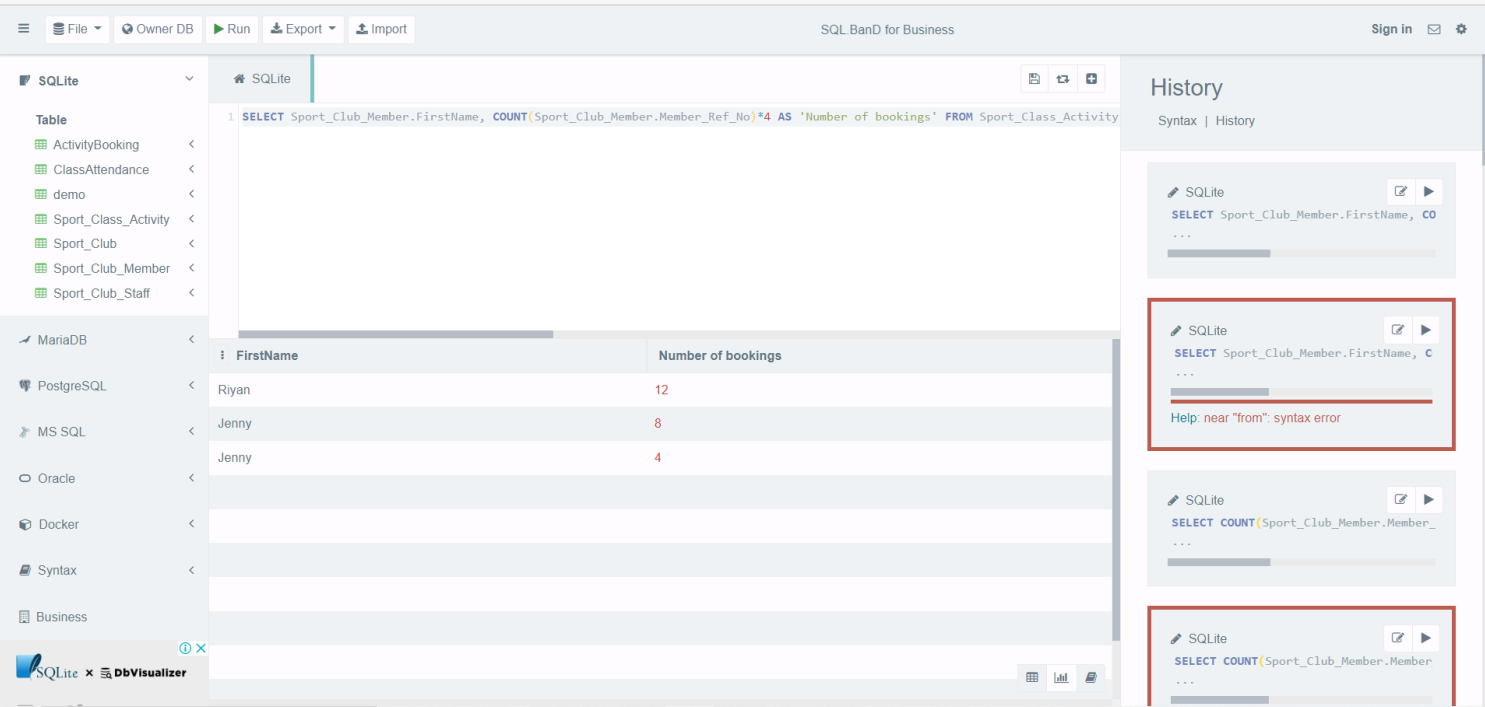
**SQL Query 2**: Write SQL query to one that displays the current weekly activities for the sports club. The SQL query for this is written and executed below.

* **select \* from Sport\_Class\_Activity NATURAL join ActivityBooking NATURAL JOIN Sport\_Club\_Member where Sport\_Class\_Activity.Day in ('Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday') and ActivityBooking.Book\_View='Book' group by Sport\_Class\_Activity.Class\_Code;**

****

**SQL Query 3:** Write SQL query to show most active members monthly (most bookings and most attendance). The SQL query for this is written and executed below.

* **select Sport\_Club\_Member.FirstName, count(Sport\_Club\_Member.Member\_Ref\_No)\*4 as 'Number of bookings' from Sport\_Class\_Activity NATURAL join ActivityBooking NATURAL JOIN Sport\_Club\_Member where Sport\_Class\_Activity.Day in ('Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday') and ActivityBooking.Book\_View='Book' group by Sport\_Class\_Activity.Class\_Code;**

****

**Design Problems or Issues**

Entity Relationship Model (ERM) for database design (Pathak, 2007) represents different types of attributes such as atomic, multi-valued, composite, derived and complex attribute using separate notations. Along with this, it also represents relationship type with different type of cardinality ratio such as one to one (1:1), one to many (1:M), many to one (M:1) and many to many (M:M). The resulted database design using ERM is not normalized. It means ERM leads number of problems or issues due to multi-valued attributes and many to many relationship types among entities. Major problems or issues of ERM database design are to produce duplicate data records, update anomaly problems as well as inconsistent state of database when it is physical implemented. Therefore to overcome or minimize these issues, we require applying normalization (Pathak, 2007) process on ERM database. ERM database is normalized by concepts of decompositions of those entities which have multi-valued attributes and participated into many to many relationships. Further, set of normal form test applied on database to ensure minimum duplicity and consistency of data. The result of normalization process over the ERM is produce new database design that referred as relational model. For the sport club management system, ERM is not normalized but implemented database is normalized.

**Brief Summary**

To overcome the problems of traditional method for managing records, the concept of database system is adapted. Today most of applications either commercial or e governance required to manage their records in efficient and secure manner. For this, database system is demanding exponentially. Database system ensures permanent storage, backup & recovery of data in easy and convenient way. For the managing record of sport club activities, staff and members a database system is designed and implemented in SQLite to meets some market need for the business of sport club owner.

**References**

Buxton, S. (2009) Database design : know it all. Amsterdam: Morgan Kaufmann Publishers/Elsevier. Available at: INSERT-MISSING-URL (Accessed: April 22, 2022).

Chilson, D. W. and Kudlac, M. E. (1983) “Database Design: A Survey of Logical and Physical Design Techniques,” ACM SIGMIS Database: the DATABASE for Advances in Information Systems, 15(1), pp. 11–19. doi: 10.1145/1113500.1113502.

Gomaa, H. (2011) Software modeling and design : uml, use cases, patterns, and software architectures. Cambridge: Cambridge University Press. Available at: INSERT-MISSING-URL (Accessed: April 22, 2022).

Newman, C. (2005) Sqlite. Indianapolis, Ind.: Sams (Developer's library). Available at: INSERT-MISSING-URL (Accessed: April 22, 2022).

Pathak, N. (2007) Database management system. Mumbai: Global Media. Available at: https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=3011378 (Accessed: April 22, 2022).

Pender, T., McSheffrey, E. and Varveris, L. (2003) Uml bible. Indianapolis, Ind.: Wiley Pub. Available at: INSERT-MISSING-URL (Accessed: April 22, 2022).