CIS108-6 Data Modelling, Management and Governance

**Assessment:** Report (Design) and Implementation

**Student Name**: Pratap Shingane

**Student Number**:7219205988

Table of Contents

[Database Solution 3](#_Toc122467701)

[ER DIAGRAM 5](#_Toc122467702)

[USE CASE Diagram 6](#_Toc122467703)

[SQL Queries 7](#_Toc122467704)

# Database Solution

Database solution is a system that allows organizations to manage, store, and retrieve data efficiently. It includes both the database management system software that manages the data as well as the physical database, where all the data is stored.

A database solution typically provides feature such as data indexing, modelling. Query optimization, data security, and backup and recovery to help organizations manage and use their data effectively. It can be used for a wide range of applications, including customer relationship management, financial reporting, inventory management and more.

There are many different types of databases. Solutions available including relational databases, no SQL databases. And in memory databases. Each type has its own strengths and its suitable for different types of data and use cases.

In this project this have been used query to face data and to get data e.g., staff data, member data at etc.

Author also used grant query for permission according to their requirement. This has to make use anyone of the database solution like my SQL and NoSQL.

**Sports Club Record System: -**

Sports Club record system is a database or system that is used to track and manage various types of information related to a Sports Club. This may include information on players, teams, schedules, results, finances and more.

Sports Club record systems may include features to manage finances such as tracking membership fees, merchandise sales and other sources of income. They may also include tools for communication, such as email or messaging capabilities for coaches and players to stay in touch.

Overall, Sports Club record system is a valuable tool for sports clubs to menage and track important information related to their operations. It helps clubs stay organized, keep track of, keep performance matrix and make informed decisions to improve their operations and support the success of their players.

**Requirement: -**

In this project author Basic requirement is structured query language.

We may use whole process time primary key or may use at the end with the alter query to give unique ID for every member of the Sports Club.

Another stuff is that we have to be careful for parsing the data and retrieve data when needed.

Author men also used as data manipulation language, which means we can use it for maintaining an already existing database.

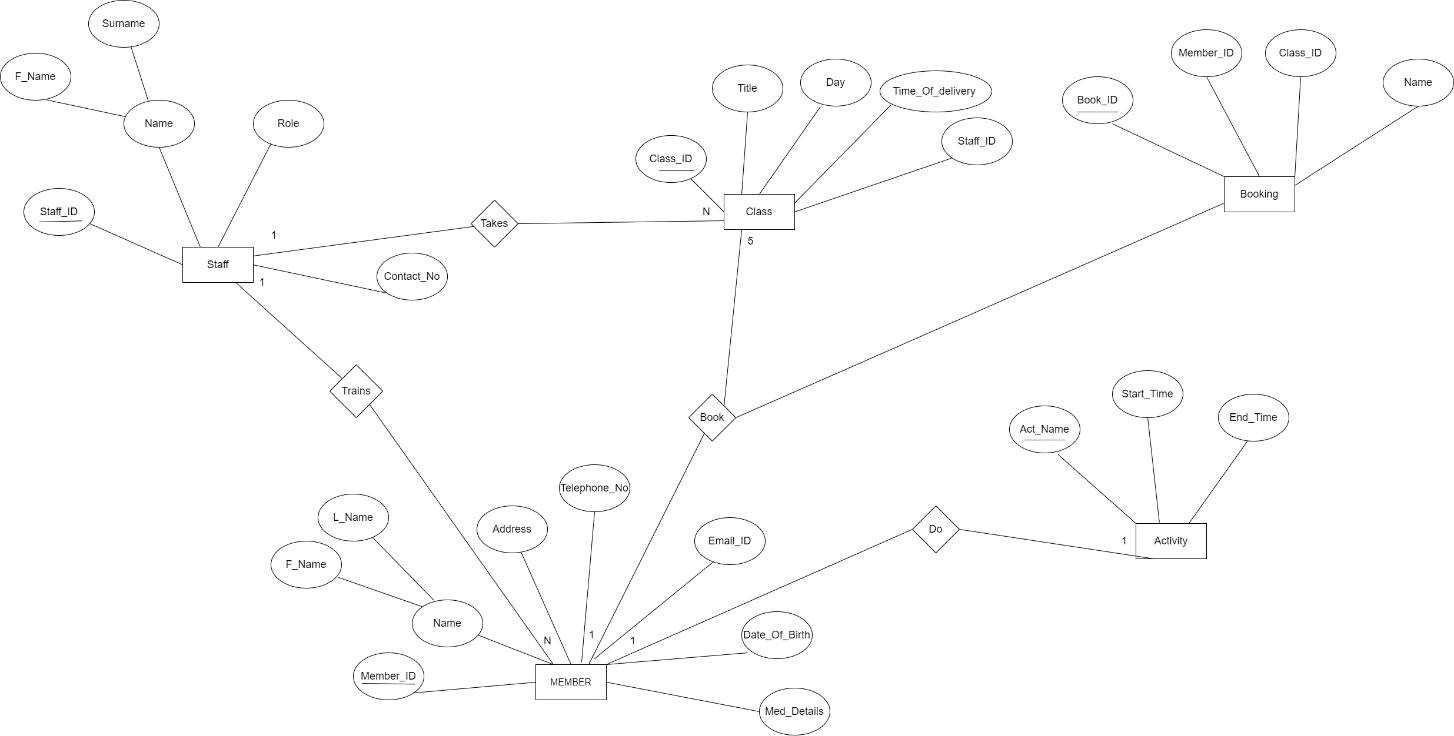
**Advantages of Sports Club Record System: -**

There are several advantages to using a Sports Club record system:

1. **Improved organization**: A sports club record system helps sports clubs stay organized and keep track of important information related to their operations. This includes player profiles, results, schedules, and finances.
2. **Enhanced performance tracking**: Sports Club record system allows clubs to track and analyses key performance matrix such as scores, statistics and team performance. This helps clubs identify areas for improvement and make informed decisions to support player development.
3. **Increased efficiency**: By automating many of tasks involved in managing a Sports Club, a Sports Club record system can help clubs operate more efficiently disinclined scheduling games and practices, managing finances, and communicating with players and coaches.
4. **Better communication**: A Sports Club record system can provide tools for coaches and players to stand touch, such as email or messaging capabilities. This helps ensure that everyone is on the same page. Can easily share important information.
5. **Cost savings**: By streamlining operation and automating many tasks. Hey, Sports Club. Record system can help sport clubs save time and money. This can allow clubs to allocate their resources more effectively and focus on their area of their operation.

# ER DIAGRAM

ER diagram means Entity Relation diagram. ER diagram is widely used to describe the conceptual design of database. It helps to the client and developer to preview the structure of database before implementing the database. Client easily visualized the data that describes how the data related to each other.



<https://drive.google.com/file/d/1ndGzno46sHng88T5ONlFmfsB8frk81p2/view?usp=sharing>

**Challenges I have faced while making ER Diagram:**

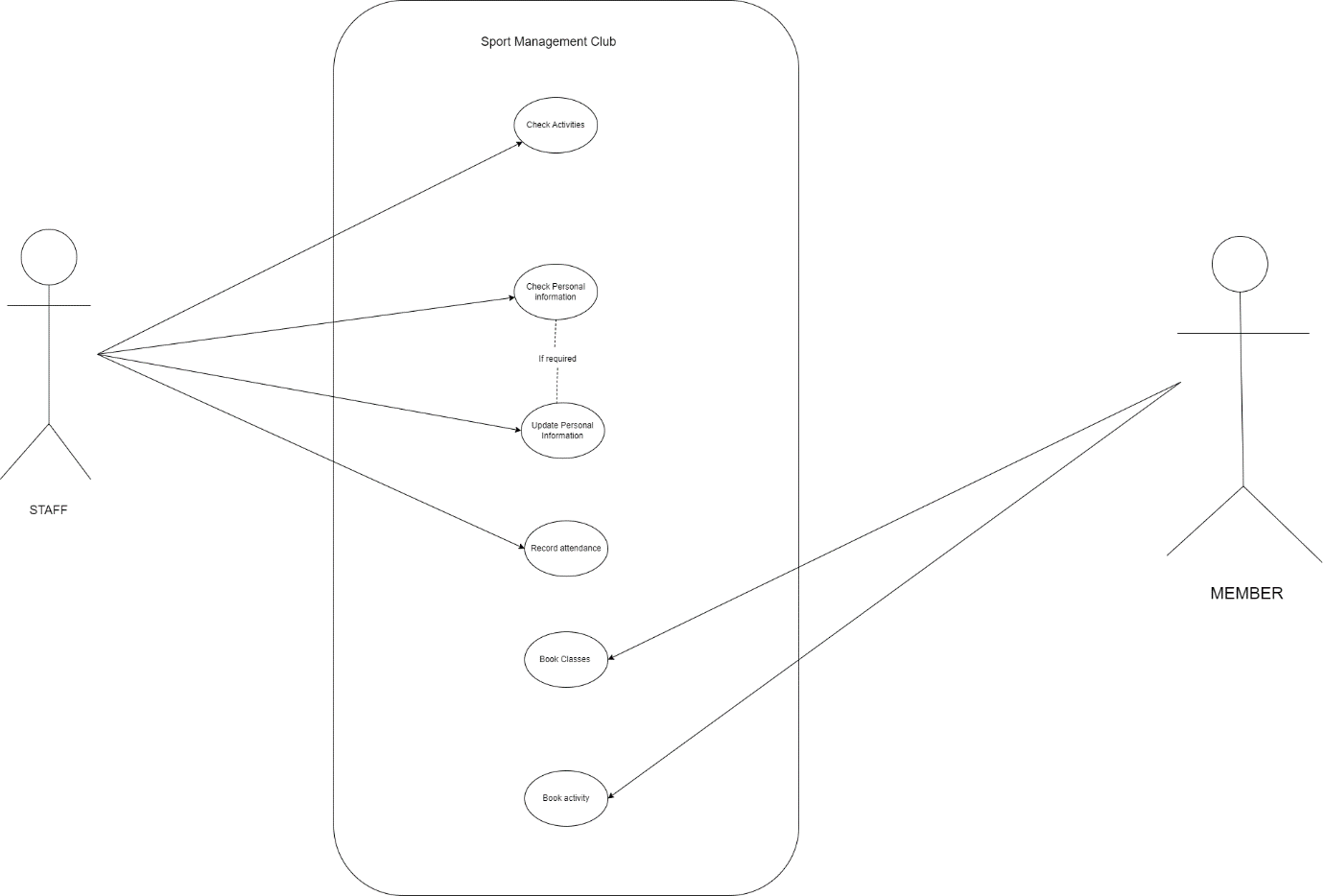
1. To decide weather there is a need to use attributes of one entity as key of another when it is not necessarily mentioned in the statement.
2. Weather to add extra entities which are not there in the problem statement but required for completing the ER.

**Benefits of ERM and how it is beneficial for designing this system:**

1. ER diagram is the base of any Database. It is very difficult to design a database without ER diagram and contains high risk of issues with design constraints.
2. For the given system it helped both user and developers to preview the structure of the database without actually implement it. It will help remove all the flaws with the database design constraints.
3. This ER diagram can be easily converted into the relational database model.

# USE CASE Diagram

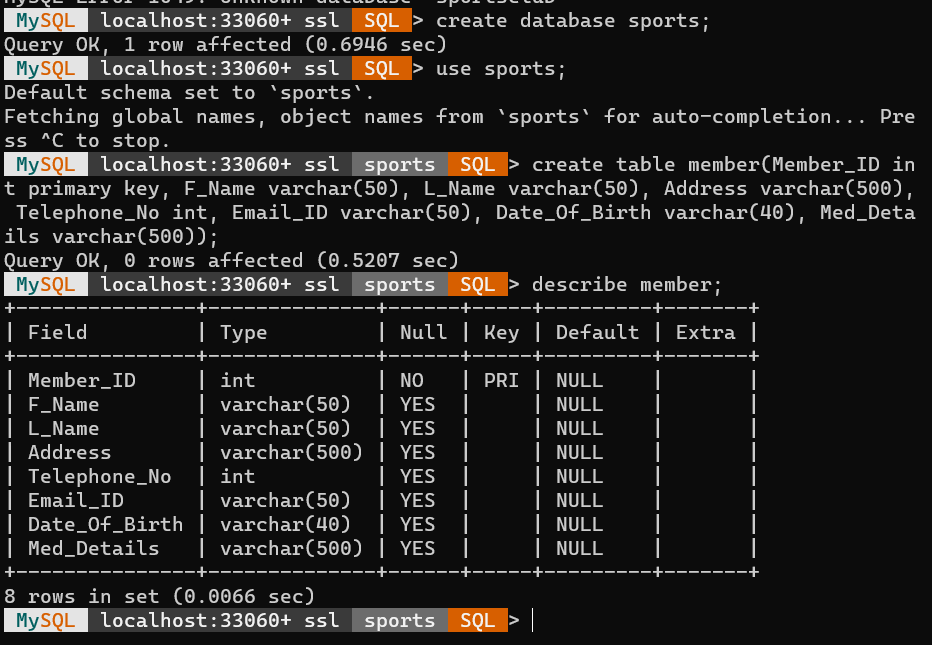
Use case diagram also used as graphical depiction of a client, possible interactions with a system.

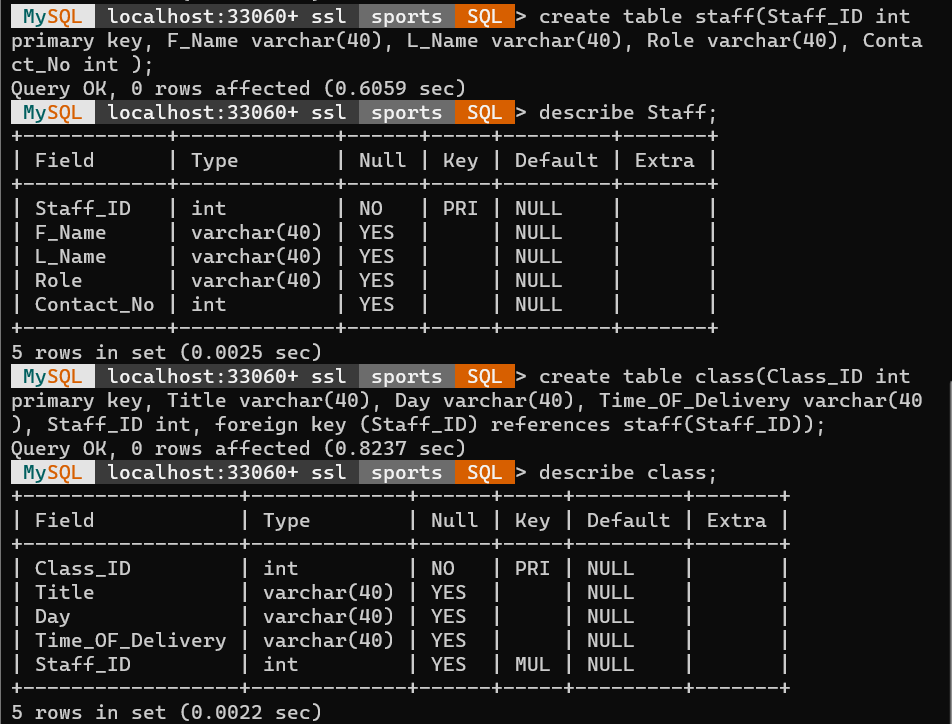


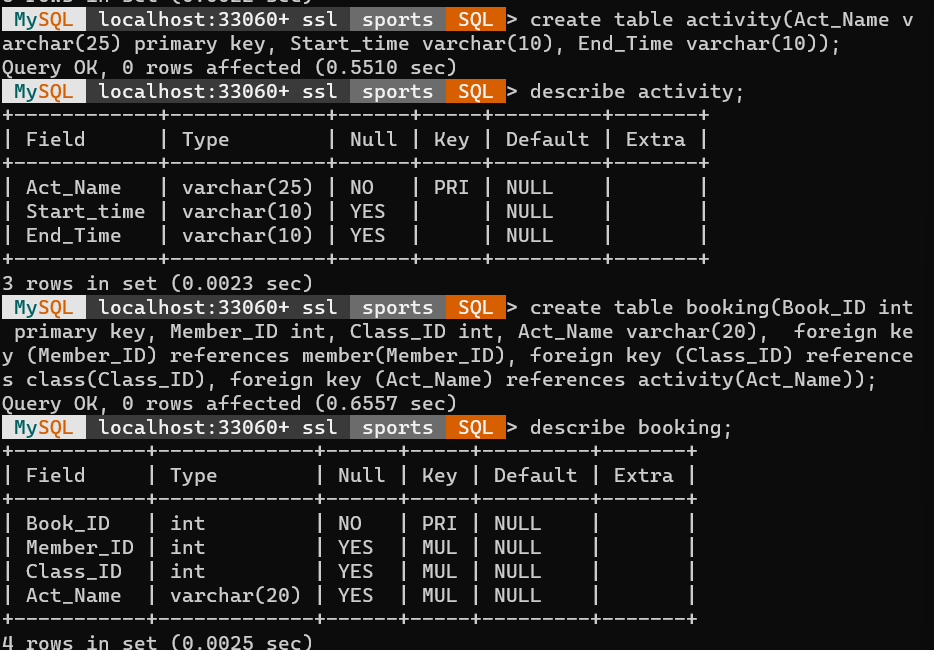
# SQL Queries

SQL-> Structured Query Language -> Used to perform operation on the data and give out the appropriate answer (data)

1. Created a new database sports.
2. Created a new table member with attributes of member mentioned in the statement.
3. Created a new table staff with attributes of member mentioned in the statement.
4. Created a table Class, activity and booking for performing operations mentioned in the statement.







Challenges Faced

**Creating Table for Member’s data: -**

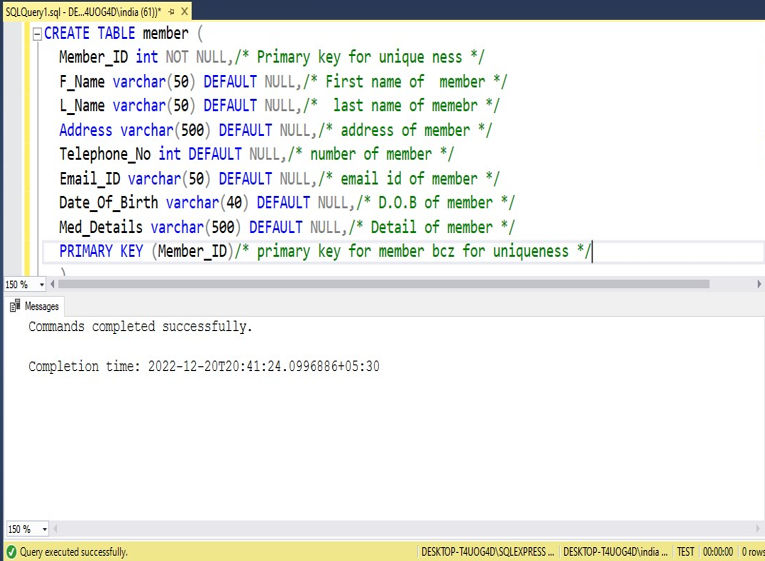
Firstly, create member id which is mandatory i.e. primary key constraint used.

Creating first name and last name using varchar datatype i.e. variable length

Creating other stuff of the member data e.g.: - address, email id etc.

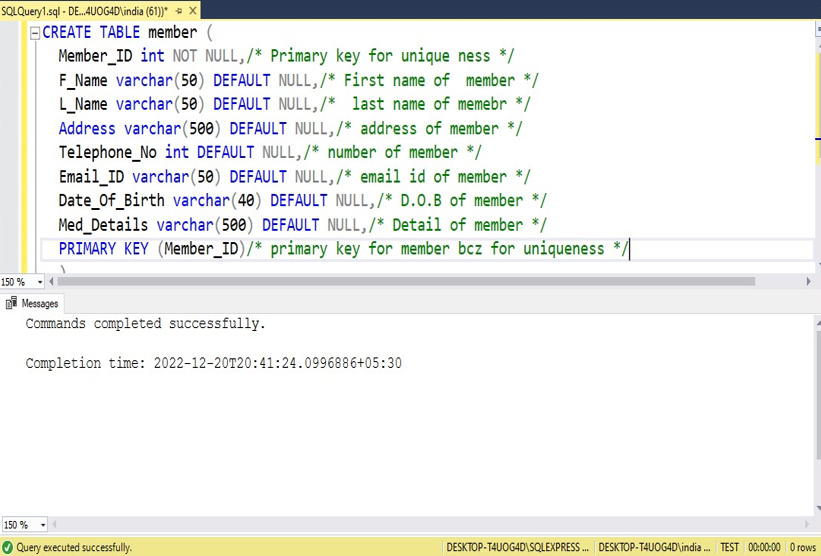
Member id should be unique so author used primary key to identify each record

Expand the explanation and Add challenges



**Creating Table for Activity data**

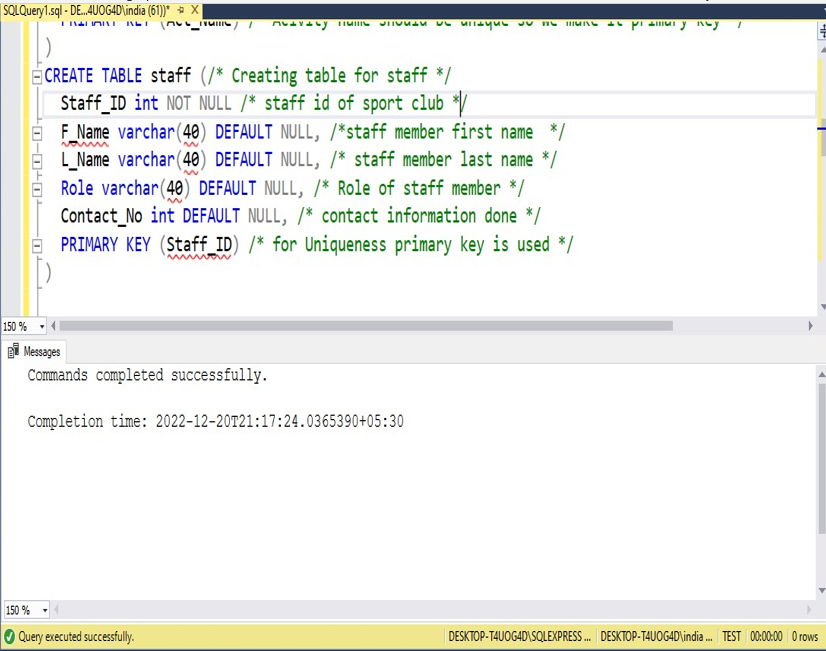
Creating data member for activity table. Number one is activity name that should be unique. After that define other stuff data member with appropriate data type



**Creating Table for Staff data**

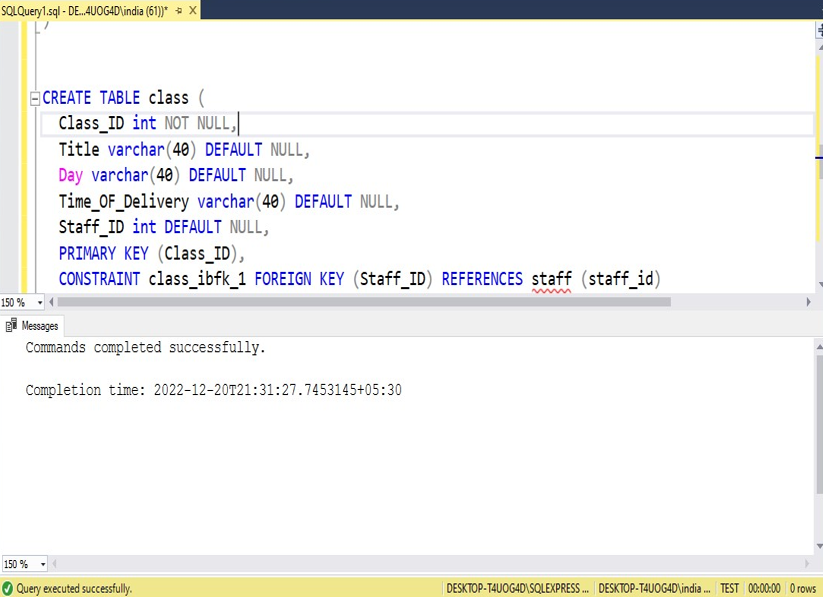
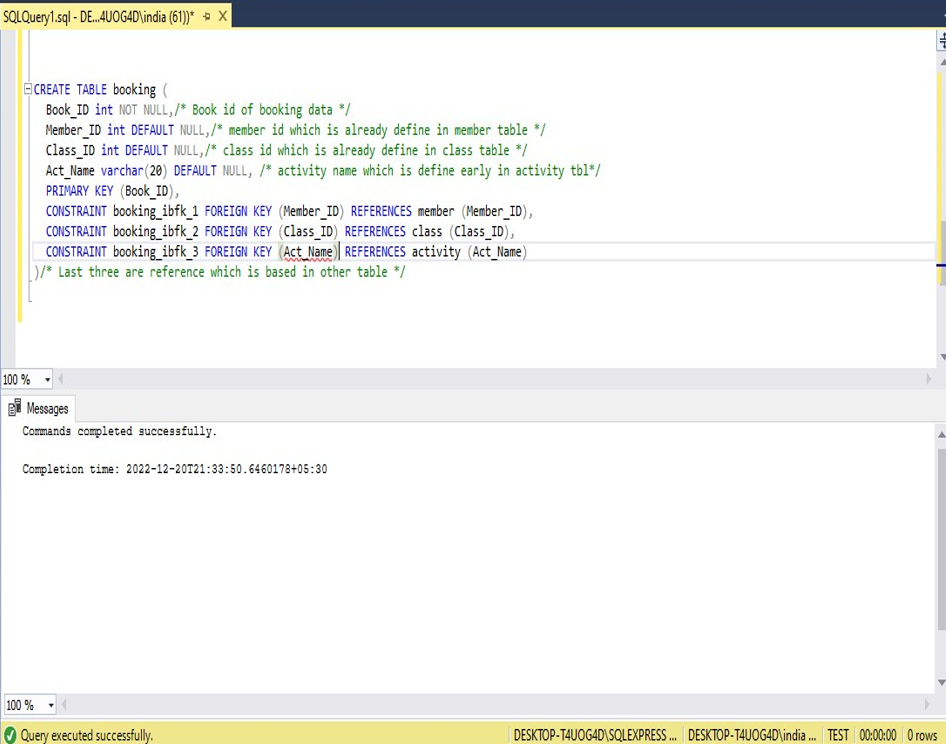
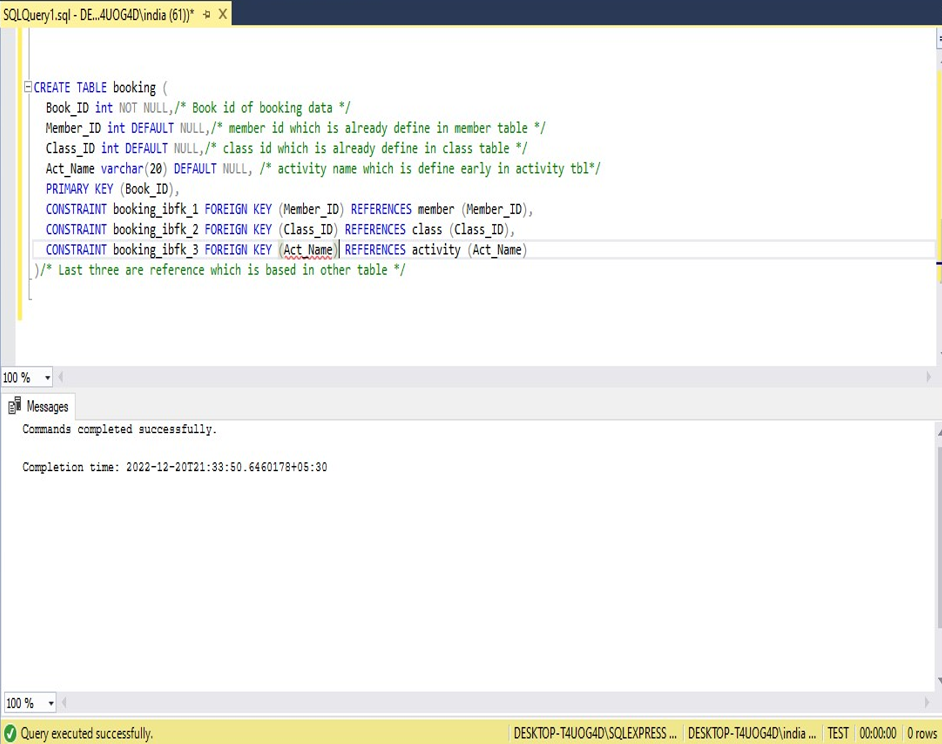
Staff data contain the staff id which is unique and some other stuff data of staff which shown below.

Expand the explanation



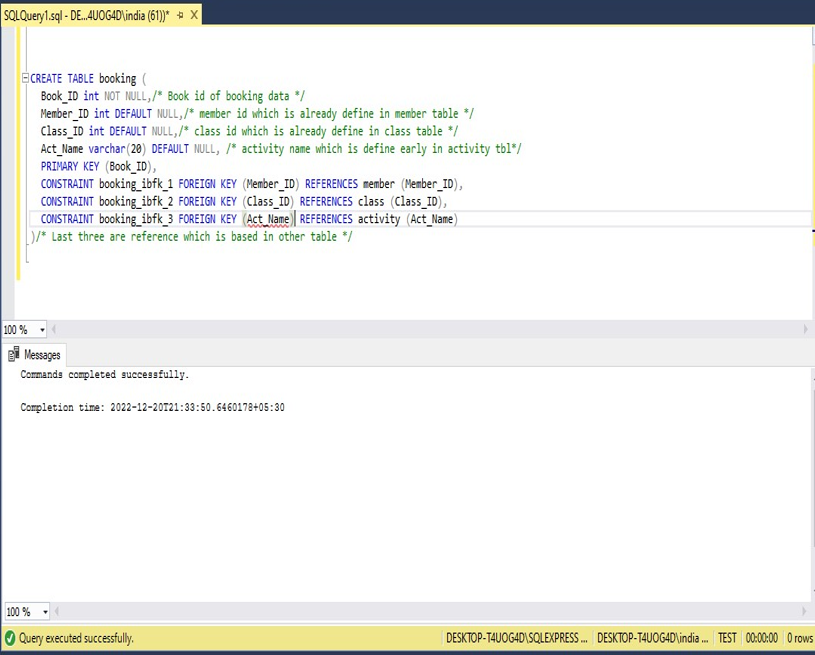
**Creating Table for Class data**

Class table contain personal data as well as other table data as a reference foreign key helper.



**Creating Table for Booking data**

In Booking table other table data also include which is referencing through various table present in data base. It is done by foreign key.



Challenges Faced:

1. To apply the constraints mentioned in the statements
2. To apply the relationships in the database.

**SPORTS.SQL**

-- MySQL dump 10.13  Distrib 8.0.31, for Win64 (x86\_64)

--

-- Host: localhost    Database: sports

-- ------------------------------------------------------

-- Server version 8.0.31

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `activity`

--

DROP TABLE IF EXISTS `activity`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

/\* created activity Table\*/

CREATE TABLE `activity` (

  `Act\_Name` varchar(25) NOT NULL,

  `Start\_time` varchar(10) DEFAULT NULL,

  `End\_Time` varchar(10) DEFAULT NULL,

  PRIMARY KEY (`Act\_Name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `activity`

--

LOCK TABLES `activity` WRITE;

/\*!40000 ALTER TABLE `activity` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `activity` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `booking`

--

DROP TABLE IF EXISTS `booking`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

/\* Created Table Booking\*/

CREATE TABLE `booking` (

  `Book\_ID` int NOT NULL,

  `Member\_ID` int DEFAULT NULL,

  `Class\_ID` int DEFAULT NULL,

  `Act\_Name` varchar(20) DEFAULT NULL,

  PRIMARY KEY (`Book\_ID`),

  KEY `Member\_ID` (`Member\_ID`),

  KEY `Class\_ID` (`Class\_ID`),

  KEY `Act\_Name` (`Act\_Name`),

  CONSTRAINT `booking\_ibfk\_1` FOREIGN KEY (`Member\_ID`) REFERENCES `member` (`Member\_ID`),

  CONSTRAINT `booking\_ibfk\_2` FOREIGN KEY (`Class\_ID`) REFERENCES `class` (`Class\_ID`),

  CONSTRAINT `booking\_ibfk\_3` FOREIGN KEY (`Act\_Name`) REFERENCES `activity` (`Act\_Name`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `booking`

--

LOCK TABLES `booking` WRITE;

/\*!40000 ALTER TABLE `booking` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `booking` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `class`

--

DROP TABLE IF EXISTS `class`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

/\*Created Table class\*/

CREATE TABLE `class` (

  `Class\_ID` int NOT NULL,

  `Title` varchar(40) DEFAULT NULL,

  `Day` varchar(40) DEFAULT NULL,

  `Time\_OF\_Delivery` varchar(40) DEFAULT NULL,

  `Staff\_ID` int DEFAULT NULL,

  PRIMARY KEY (`Class\_ID`),

  KEY `Staff\_ID` (`Staff\_ID`),

  CONSTRAINT `class\_ibfk\_1` FOREIGN KEY (`Staff\_ID`) REFERENCES `staff` (`Staff\_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `class`

--

LOCK TABLES `class` WRITE;

/\*!40000 ALTER TABLE `class` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `class` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `member`

--

DROP TABLE IF EXISTS `member`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

/\*Created Table Mamber\*/

CREATE TABLE `member` (

  `Member\_ID` int NOT NULL,

  `F\_Name` varchar(50) DEFAULT NULL,

  `L\_Name` varchar(50) DEFAULT NULL,

  `Address` varchar(500) DEFAULT NULL,

  `Telephone\_No` int DEFAULT NULL,

  `Email\_ID` varchar(50) DEFAULT NULL,

  `Date\_Of\_Birth` varchar(40) DEFAULT NULL,

  `Med\_Details` varchar(500) DEFAULT NULL,

  PRIMARY KEY (`Member\_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `member`

--

LOCK TABLES `member` WRITE;

/\*!40000 ALTER TABLE `member` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `member` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `staff`

--

DROP TABLE IF EXISTS `staff`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

/\*Created Table Staff\*/

CREATE TABLE `staff` (

  `Staff\_ID` int NOT NULL,

  `F\_Name` varchar(40) DEFAULT NULL,

  `L\_Name` varchar(40) DEFAULT NULL,

  `Role` varchar(40) DEFAULT NULL,

  `Contact\_No` int DEFAULT NULL,

  PRIMARY KEY (`Staff\_ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `staff`

--

LOCK TABLES `staff` WRITE;

/\*!40000 ALTER TABLE `staff` DISABLE KEYS \*/;

/\*!40000 ALTER TABLE `staff` ENABLE KEYS \*/;

UNLOCK TABLES;

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2022-12-13 17:09:04

Will do proof reading and make sure every sentence is in third person

**Google these headings and write them**

testing report

system tech design report

operation manual (typical usage)

“use case diagram”, an “activity diagram”

# Summary

The assignment helps me learn SQL concepts, designing of database, dataflow diagrams and help me learn about how a database system is designed. The positive side of the assignment is that it helps me design the system and code it using SQL concept for its kind of project for me. SQL concepts are really important and it is very important to go deep into SQL to design a good database with less redundancy and meet requirement of all the constraints that the client requirements meet. It was a database designed to track various activities of a sport club. The staff entity of the database contains information about the staff ID which help to identify unique staff, staff’s first name and last name, the role of the staff, along with their contact number. Next entity is the Member entity who the target of the sport club. The sport member has to register for two types of stuff one is activity which include all the sport and its staring time is 6am and ending time is 10 pm, there is no requirement of coaching if member is participating in just the activities. Next is the Classes which offers training, training include the staff training and staring time for coaching is 9 am and ending tine is 5pm. The next and the important entity is the Booking which include booking ID which hlp identify unique booking, it also contains Member ID reference from member table and class ID reference from class table. The next entity is the class entity is the class which include the class ID, Title of the class weather it is the coaching of which sport, schedule of the class which is day and time of start and end of the class. The next entity is the activity entity which has attributes name of the activity, its start time and end time. The start time and end time of the activity and the class is really very important as it is used as a reference for the booking entity. It contains the testing report along with operation manuals and system tech design report.