Database Project Week

# Introduction

You have been asked to develop a database for a local leisure centre, as they are in the process of upgrading their existing paper base record keeping system building for their swimming pool. They would like to use the database to help them manage their swimming lessons. Below are the entities and their attributes which were extracted from their paper-based system by a database developer who never got to implement the database for the local leisure centre. Use the **entities** and their **attributes** provided to create the respective tables. In addition, the leisure centre has now decided that they want two databases. A database implemented using SQL and a second NoSQL database, preferably using MySQL workbench and MongoDB respectively. It is the view that once the databases are successfully implemented, the leisure centre would then decide which of the databases they would possibly integrate with a front-end Python application.

* Course (courseID, Level, Sessions, Instructor, startDate, LessonTime)
  + Lessons (LessonID, CourseID, MemberID)
  + Members (MemberID, Firstname, Surname, DOB, Address, City)

The above highlighted in green are the primary keys.

EXERCISES:

1. **Use the SQL AND, OR and NOT Operators in your query (The WHERE clause can be combined with AND, OR, and NOT operators)**
2. Where courseID is equals to a number below 5 and the first name of any of the instructors
3. Where courseID is equals to a number above 5 and the lesson time is in the morning or afternoon.
4. **Order by the above results by:**
5. startDate in “*course*” table
6. MemberID in “*members*” table
7. **UPDATE the following:**
8. Members table, change the addresses of any three members.
9. Course table, change the startDate and lesson time for three of the sessions.
10. **Use the SQL MIN () and MAX () Functions to return the smallest and largest value**
11. Of the LessonID column in the “*lesson*” table
12. Of the membersID column in the “*members*” table
13. **Use the SQL COUNT (), AVG () and SUM () Functions for these:**
14. Count the total number of members in the “*members*” table
15. Count the total number of sessions in the” members” table
16. Find the average session time for all “*sessions”* in course table
17. **WILDCARD queries (like operator)**
18. Find all the people from the “*members*” table whose last name starts with A.
19. Find all the people from the “*members*” table whose last name ends with A.
20. Find all the people from the “*members*” table that have "ab" in any position in the last name.
21. Find all the people from the “*members*” table that that have "b" in the second position in their first name.
22. Find all the people from the “*members*” table whose last name starts with "a" and are at least 3 characters in length:
23. Find all the people from the “*members*” table whose last name starts with "a" and ends with "y"
24. Find all the people from the “*members*” table whose last name **does not** starts with "a" and ends with "y"
25. **What do you understand by LEFT and RIGHT join? Explain with an example.**

# 