

→ Making the database:

```
create database leisureCentre;
```

```
use leisureCentre;
```

```
create table course (  
courseID int auto_increment primary key,  
level varchar(30),  
sessions varchar(30),  
instructorName varchar(30),  
startDate date,  
lessonTime time  
);
```

```
insert into course values  
(1, 'Advanced', 'Morning', 'Peter', '2023-07-05', '09:30:00'),  
(2, 'Intermediate', 'Morning', 'Devon', '2023-07-09', '10:30:00'),  
(3, 'Advanced', 'Afternoon', 'Leila', '2023-06-10', '13:30:00'),  
(4, 'Beginner', 'Evening', 'Leila', '2023-05-01', '18:30:10'),  
(5, 'Intermediate', 'Evening', 'Peter', '2023-06-11', '18:30:10');
```

```
select * from course;
```

```
explain course;
```

```
create table members (  
memberID int auto_increment primary key,  
firstName varchar(30),  
surname varchar(30),  
DOB date,  
address varchar(50),
```

```
city varchar(20)
```

```
);
```

```
insert into members values
```

```
(102, 'Sampha', 'Miles', '1998-01-09', '33 Goodwin Close Mictam CR4 3QB', 'London'),
```

```
(108, 'Dijon', 'Daniels', '1971-02-19', '306 Church Road Merton SW19 2QF', 'London'),
```

```
(259, 'Carter', 'Preston', '1982-08-11', '78 Jadewood Farms New Brunswick NJ1 2PS', 'New Jersey'),
```

```
(781, 'Raemon', 'McGuire', '2000-11-18', '23 Elmsend Road Chessington CR0 1WE', 'Chessington');
```

```
select * from members;
```

```
explain members;
```

```
create table lessons (
```

```
lessonID int auto_increment primary key,
```

```
courseID int,
```

```
memberID int,
```

```
FOREIGN KEY (courseID) REFERENCES course(courseID),
```

```
FOREIGN KEY (memberID) REFERENCES members(memberID)
```

```
);
```

```
-- ALTER TABLE lessons ADD CONSTRAINT PK_Lessons PRIMARY KEY (lessonID);
```

```
insert into lessons values
```

```
(234, 1, 102),
```

```
(212, 2, 108),
```

```
(298, 3, 259),
```

```
(278, 4, 781);
```

```
select * from lessons;
```

```
explain lessons;
```

**-- Queries:**

**-- A. Use the SQL AND, OR and NOT Operators in your query (The WHERE clause can be combined with AND, OR, and NOT operators)**

**-- 1. Where courseID is equal to a number below 5 and the first name of any of the instructors**

```
select * from course where courseID < 5 and instructorName = 'Leila';
```

**-- 2. Where courseID is equal to a number above 5 and the lesson time is in the morning or afternoon.**

```
select * from course where courseID > 5 and lessonTime = 'Morning' or 'Afternoon';
```

**-- B. Order by the above results by:**

**-- 1. startDate in the "course" table**

```
select * from course order by startDate desc;
```

```
select * from course order by startDate asc;
```

**-- 2. MemberID in the "members" table**

```
select * from members order by memberID desc;
```

```
select * from members order by memberID asc;
```

**-- C. UPDATE the following:**

**-- 1. Members table, change the addresses of any three members. {??}**

```
UPDATE members
```

```
SET address = '18 Willow Lane Surrey SE9 2QW'
```

```
WHERE memberID = 781;
```

```
select * from members;
```

**-- 2. Course table, change the startDate and lesson time for three of the sessions.**

```
UPDATE course
```

```
SET startDate = '2022-12-10', lessonTime = '09:00:00'
```

```
WHERE sessions = 'Morning';
```

```
UPDATE course
```

```
SET startDate = '2022-12-10', lessonTime = '13:00:00'
```

```
WHERE sessions = 'Afternoon';
```

```
UPDATE course
```

```
SET startDate = '2022-12-10', lessonTime = '18:00:00'
```

```
WHERE sessions = 'Evening';
```

```
select * from course;
```

**-- D. Use the SQL MIN () and MAX () functions to return the smallest and largest value**

**-- 1. Of the LessonID column in the "lesson" table**

```
select * from lessons;
```

```
SELECT min(lessonID) FROM lessons;
```

```
SELECT max(lessonID) FROM lessons;
```

**-- 2. Of the membersID column in the "members" table**

```
select * from members;
```

```
SELECT min(memberID) FROM members;
```

```
SELECT max(memberID) FROM members;
```

**-- E. Use the SQL COUNT (), AVG () and SUM () functions for these:**

**-- 1. Count the total number of members in the "members" table**

```
SELECT COUNT(memberID) FROM members;
```

**-- 2. Count the total number of sessions in the "course" table**

```
SELECT COUNT(sessions) FROM course;
```

**-- 3. Find the average session time for all sessions in the "course" table**

```
SELECT AVG(lessonTime) FROM course;
```

**-- F. WILDCARD queries (like operator)**

--

**-- a. Find all the people from the "members" table whose last name starts with A.**

```
select * from members where surname like 'A%';
```

```
-- extra by me: starts with M
```

```
select * from members where surname like 'M%';
```

**-- b. Find all the people from the "members" table whose last name ends with A.**

```
select * from members;
```

```
select * from members where surname like '%A';
```

-- extra by me: ends with S

```
select * from members where surname like '%S';
```

**-- c. Find all the people from the "members" table that have "ab" in any position in the last name.**

-- locate = returns the position of the first occurrence of the character

```
select *, locate('ab', surname) AS FirstOccurrenceOfAB FROM members;
```

-- other attempts

```
select locate('mc', surname) AS FirstOccurrenceOfMC FROM members;
```

```
select *, locate('an', surname) AS FirstOccurrenceOfAN FROM members;
```

**-- d. Find all the people from the "members" table that have "b" in the second position in their first name.**

```
select firstName, locate('b', 2) FROM members;
```

**-- e. Find all the people from the "members" table whose last name starts with "a" and are at least 3 characters in length:**

```
select * from members where surname like 'A%' and length(surname)>3;
```

**-- f. Find all the people from the "members" table whose last name starts with "a" and ends with "y"**

```
select * from members;
```

```
SELECT * from members WHERE surname like 'A%Y';
```

-- another query with data I have

```
SELECT * from members WHERE surname like 'D%S';
```

**-- g. Find all the people from the "members" table whose last name does not start with "a" and ends with "y"**

```
SELECT * from members WHERE surname not like 'A%Y';
```

**-- G. What do you understand by LEFT and RIGHT joins? Explain with an example.**

```
SELECT c.courseID
```

```
FROM course as c
```

```
LEFT JOIN lessons as l
```

```
ON c.courseID = l.lessonID;
```