1. **Write a SQL statement to find the names of teams that have played more than 3 matches.[Database: FLIS]**

Ans :

SELECT t.name

FROM teams t

JOIN matches m

ON t.team\_id = m.host\_team\_id OR t.team\_id = m.guest\_team\_id

GROUP BY t.name

HAVING COUNT(m.match\_num) > 3;

1. **Write an SQL statement to find the first name, last name of the faculty of the department having department code as 'ME' and who have issued at least one book, such that there are no duplicate firstname-lastname pairs.[Database: LIS]**

SELECT DISTINCT f.faculty\_fname, f.faculty\_lname

FROM faculty f

JOIN members m ON f.id = m.id

JOIN book\_issue bi ON m.member\_no = bi.member\_no

WHERE f.department\_code = 'ME';

1. **Write an SQL statement to find the number of book-titles issued on 11th August 2021.[Database: LIS]**

SELECT COUNT(DISTINCT book\_catalogue.title) AS number\_of\_titles\_issued

FROM book\_issue

JOIN book\_copies ON book\_issue.accession\_no = book\_copies.accession\_no

JOIN book\_catalogue ON book\_copies.ISBN\_no = book\_catalogue.ISBN\_no

WHERE book\_issue.doi = '2021-08-11';

1. **Write a SQL statement to find the names of faculty (faculty\_fname, faculty\_lname) who did not issue any book.[Database: LIS]**

SELECT faculty.faculty\_fname, faculty.faculty\_lname

FROM faculty

LEFT JOIN members ON faculty.id = members.id

LEFT JOIN book\_issue ON members.member\_no = book\_issue.member\_no

WHERE book\_issue.member\_no IS NULL

**( please cross check its retrieving correctly,see if any student names are coming in the list)**

1. **Write a SQL statement to find the unique book titles which are issued to 'PG' students but not to 'UG' students .[Database: LIS]**

**Method 1 : ( using subquery)**

SELECT DISTINCT book\_catalogue.title

FROM book\_issue

JOIN members ON book\_issue.member\_no = members.member\_no

JOIN book\_copies ON book\_issue.accession\_no = book\_copies.accession\_no

JOIN book\_catalogue ON book\_copies.isbn\_no = book\_catalogue.isbn\_no

WHERE members.member\_type = 'PG'

AND book\_catalogue.title NOT IN (

SELECT DISTINCT book\_catalogue.title

FROM book\_issue

JOIN members ON book\_issue.member\_no = members.member\_no

JOIN book\_copies ON book\_issue.accession\_no = book\_copies.accession\_no

JOIN book\_catalogue ON book\_copies.isbn\_no = book\_catalogue.isbn\_no

WHERE members.member\_type = 'UG'

);

**Method 2: (using EXCEPT)**

SELECT DISTINCT book\_catalogue.title

FROM book\_issue

JOIN members ON book\_issue.member\_no = members.member\_no

JOIN book\_copies ON book\_issue.accession\_no = book\_copies.accession\_no

JOIN book\_catalogue ON book\_copies.isbn\_no = book\_catalogue.isbn\_no

WHERE members.member\_type = 'PG'

EXCEPT

SELECT DISTINCT book\_catalogue.title

FROM book\_issue

JOIN members ON book\_issue.member\_no = members.member\_no

JOIN book\_copies ON book\_issue.accession\_no = book\_copies.accession\_no

JOIN book\_catalogue ON book\_copies.isbn\_no = book\_catalogue.isbn\_no

WHERE members.member\_type = 'UG';

1. **Write a SQL statement to find the name of the manager of the team: 'All Stars'.[Database: FLIS]**

Ans : select name from managers,teams where managers.team\_id=(select team\_id from teams where name='All Stars')

1. **Write an SQL statement to find the names of all teams.[Database: FLIS]**

Ans : select name from teams

1. **Write an SQL statement to find the titles of books authored by an author having first name as 'Joh Paul' and last name as 'Mueller'.[Database: LIS**

**]**

Ans : select title from book\_catalogue where ISBN\_no in(select ISBN\_no from book\_authors where author\_fname='Joh Paul' and author\_lname='Mueller')

1. **Write a SQL statement to find the titles of books published by 'McGraw Hill Education'.[Database: LIS]**

Ans : select title from book\_catalogue where publisher='McGraw Hill Education'

1. **Write a SQL statement to display the first name and the last name of students (student\_fname, student\_lname) pursuing 'PG' courses.[Database: LIS]**

Ans :

select student\_fname,student\_lname

from students

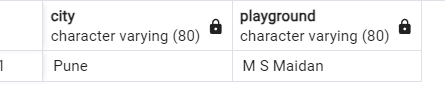
WHERE roll\_no in

(select roll\_no from

members

WHERE member\_type = 'PG');

### Write an SQL statement to find the city and playground of the teams whose away-jersey color(jersey\_away\_color) is 'Pink'. ( 1 mark )

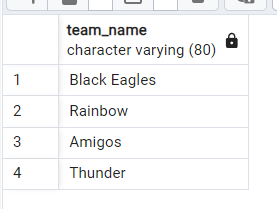


SELECT city, playground

FROM teams

WHERE jersey\_away\_color = 'Pink';

### Write a SQL statement to find the names of teams that have played more than 3 matches ( 1 mark )



SELECT t.name

FROM teams t

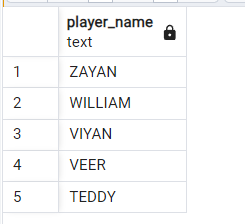
JOIN matches m

ON t.team\_id = m.host\_team\_id OR t.team\_id = m.guest\_team\_id

GROUP BY t.name

HAVING COUNT(m.match\_num) > 3;

### Write an SQL query to retrieve a unique list of player names in uppercase, who were born on or after '2000-01-01'. Sort the results alphabetically in descending order and only display the first 5 names. ( 2 mark)



SELECT DISTINCT UPPER(name) AS player\_name

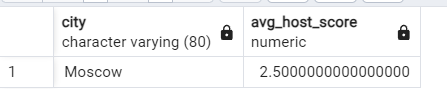
FROM players

WHERE dob >= '2000-01-01'

ORDER BY player\_name DESC

LIMIT 5;

### Find the average score of host teams for each city. Only include cities where the average host team score is at least 2 and there are more than 3 matches. ( 2 marks )



SELECT t.city, AVG(m.host\_team\_score) AS avg\_score, COUNT(\*) AS match\_count

FROM matches m

JOIN teams t ON m.host\_team\_id = t.team\_id

GROUP BY t.city

HAVING AVG(m.host\_team\_score) >= 2 AND COUNT(\*) > 3;