SQL Practice Solutions

1. Find the first names and last names of authors, having the author's first name as a single character. (Database: LIS)

Solution:

```
SELECT author_fname, author_lname
FROM book_authors
WHERE author_fname LIKE ‘_’
```

2. Find the titles and publishers of all books, except the ones published in year ‘2015' or ‘2017'. (Database: LIS)

Solution:

```
SELECT title, publisher
FROM book_catalogue
EXCEPT
SELECT title, publisher
FROM book_catalogue
WHERE year IN (‘2015’, ‘2017’)
```

3. Find the first names and last names of the students whose birthday is in May 2002 or in May 2003. (Database: LIS)

Solution:

```
SELECT student_fname, student_lname
FROM students
WHERE dob BETWEEN ’2003-05-01’ AND ’2003-05-30’
UNION
SELECT student_fname, student_lname
FROM students
WHERE dob BETWEEN ’2002-05-01’ AND ’2002-05-30’
```

4. Find out the total number of members in the UG with alias name or column header as ‘total member’. (Database: LIS)

Solution:

```
SELECT COUNT(member_type) AS total_member
FROM members
WHERE member_type=‘UG’
```

5. Find out the number of female students in each department. Display department_code and number of female students. (Database: LIS)

Solution:

```
SELECT department_code, COUNT(gender) AS no_of_females
FROM students
WHERE gender=‘F’
GROUP BY department_code
```

6. Find the names and date-of-births of those managers who have joined in years 2019 and 2020. (Database: FLIS)

Solution:

```
SELECT name, dob
FROM managers
WHERE since BETWEEN ’2019-01-01’ AND ’2020-12-31’
```

7. Find the names of all those teams where the last name of the team starts with the letter S. Use the name attribute of the teams table to answer this question. (Database: FLIS)

Solution:

```
SELECT *
FROM teams
WHERE name LIKE ‘_% S%’
```

8. Print the name, dept name, salary of instructors, where the department name should be sorted in descending order and within each department, the instructor names should be sorted in ascending order of salary within that department. (Database: University)

Solution:

```
SELECT name,dept_name,salary
FROM instructor
ORDER BY dept_name DESC, salary ASC
```

9. Write a SQL statement to find the names of players that start with ‘S’ but does not end with ‘n’. (Database: FLIS)

Solution:

```
SELECT name
FROM players
WHERE name LIKE ‘S%’
EXCEPT
SELECT name
FROM players
WHERE name LIKE ‘%n’
```

10. Find out the total number of players who are playing from the team id ‘T0001’. (Database: FLIS)

Solution:

```
SELECT COUNT(player_id)
FROM players
GROUP BY team_id
HAVING team_id = ‘T0001’
```

11. Write a query to obtain the natural join between the tables, students and departments. (Database: LIS)

Solution:

```
SELECT *
FROM students NATURAL JOIN departments
```

12. Find details of those instructors of the Accounting department who have more salary than at least one instructor of the Psychology department. (Database: University)

Solution:

```
SELECT *
FROM instructor
WHERE dept_name = ‘Accounting’
AND salary > ANY (SELECT salary
                  FROM instructor
                  WHERE dept_name = ‘Psychology’)
```

13. Write a SQL statement to find out the manager’s date of birth(dob) of the team for which “Shlok” plays. (Database: FLIS)

Solution:

```
SELECT dob
FROM managers
WHERE team_id IN (SELECT team_id
                  FROM players
                  WHERE name = ‘Shlok’)
```

14. Write a SQL statement to find out match number(match num) played in the playground “Emirates Stadium”. (Database: FLIS)

Solution:

```
SELECT match_num
FROM matches
WHERE host_team_id = ANY (SELECT team_id
                          FROM teams
                          WHERE playground = ‘Emirates Stadium’)
```

15. Find the name, player id, date of birth and city of all players who played for team ‘Rainbow’. (Database: FLIS)

Solution:

```
SELECT p.name, p.player_id, p.dob, t.city
FROM players AS p, teams AS t
WHERE p.team_id = t.team_id
AND t.name = ‘Rainbow’
```

16. Find the name of the teams which belong to the same city as the team ‘Amigos’. (Database: FLIS)

Solution:

```
SELECT t2.name
FROM teams AS t1, teams AS t2
WHERE t1.name = ‘Amigos’
AND t2.name <> ‘Amigos’
AND t1.city = t2.city
```

17. Find the name of the department in which Gita Das is studying. (Database: LIS)

Solution:

```
SELECT department_name
FROM departments NATURAL JOIN students
WHERE student_fname = ‘Gita’
AND student_lname = ‘Das’
```

18. Find the roll number of all male students, having their department building in ‘Block 2’. (Database: LIS)

Solution:

```
SELECT roll_no
FROM departments NATURAL JOIN students
WHERE department_building = ‘Block_2’
AND gender = ‘M’
```

19. Find the first name, last name and the roll number of students having their department building in ‘Block 1’. (Database: LIS)

Solution:

```
SELECT student_fname, student_lname, roll_no
FROM departments NATURAL JOIN students
WHERE department_building = ‘Block_1’
```

20. Find out the details of the members who have not issued any books. (Database: LIS)

Solution:

```
SELECT *
FROM members
WHERE NOT EXISTS (SELECT *
                  FROM book_issue
                  WHERE members.member_no = book_issue.member_no)
```

21. Find out the name of courses which have been taught in both Fall semester and Spring semester. (Database: University)

Solution:

```
SELECT title
FROM course
WHERE course_id IN ((SELECT course_id
                     FROM section
                     WHERE semester = ‘Fall’)
                    INTERSECT
                     (SELECT course_id
                      FROM section
                      WHERE semester = ‘Spring’))
```

22. Write a SQL statement to find out the number of students who have studied in each building from 2005 till 2008 (including 2005, 2008). (Database: University)

Solution:

```
SELECT COUNT(DISTINCT st.id), building
FROM section AS se, takes AS t, student AS st
WHERE st.id = t.id
AND t.course_id = se.course_id
AND t.sec_id = se.sec_id
AND t.semester = se.semester
AND t.year = se.year
AND (t.year >= 2005 AND t.year <= 2008)
GROUP BY building
```

23. Write a SQL statement to find out the dates when one or more copies of the book having the title “Learning with Python” has been issued. (Database: LIS)

Solution:

```
SELECT DISTINCT bi.doi
FROM book_catalogue AS bkcat
INNER JOIN book_copies AS bcp USING (isbn_no)
INNER JOIN book_issue AS bi USING (accession_no)
WHERE bkcat.title = ‘Learning with Python’
```

24. Let D be the set of all departments whose average salary is more than the maximum salary of ‘Psychology’ department. Write a SQL statement to find the name and salary of the instructor(s) who has/have the maximum salary in each department in D. (Database: University)

Solution:

```
WITH temp_table(iname,isalary,idept_name) AS
(SELECT i.name,i.salary,i.dept_name
FROM instructor i
WHERE i.dept_name IN (SELECT dept_name
FROM instructor
GROUP BY dept_name
HAVING AVG(salary) > (SELECT MAX(salary)
FROM instructor
WHERE dept_name = ’Psychology’)))
select iname, isalary from temp_table where isalary in (SELECT max(isalary)
FROM temp_table group by idept_name)
```

25. Write a SQL statement to find out the number of courses which have been taught in Fall semester but never in Spring semester. (Database: University)

Solution:

```
    SELECT COUNT(*)
    FROM ((SELECT course_id
           FROM section
           WHERE semester = ‘Fall’)
          EXCEPT
          (SELECT course_id
           FROM section
           WHERE semester = ‘Spring’)) AS temp_table
```

26. Print name,id,num count of instructor(s) who has taught maximum number of classes on the day ‘W’. (num count is the number of the classes they took on day ‘W’.) (Database: University)

Solution:

```
  with temp_table(iid, icount) AS
     (SELECT i.id, count(*)
      FROM section se, time_slot ts, teaches t, instructor i
      WHERE se.time_slot_id = ts.time_slot_id
      AND se.course_id = t.course_id
      AND se.sec_id = t.sec_id
      AND se.semester = t.semester
      AND se.year = t.year
      AND t.id = i.id
      AND ts.day = ‘W’
      GROUP BY i.id)
    SELECT name, iid, icount
    FROM temp_table, instructor
    WHERE temp_table.iid = instructor.id
    AND icount >= ALL (SELECT icount
                       FROM temp_table)
```

27. Find the names of the fourth referees (fourth referee) and the match number of all matches played on “2020-05-19”. (Database: FLIS)

Solution:

```
SELECT name, match_num
FROM referees
INNER JOIN match_referees
ON referees.referee_id = match_referees.fourth_referee
INNER JOIN matches USING (match_num)
WHERE match_date = ‘2020-05-19’
```

28. Write a SQL statement to find out the names of research scholars (RS) who have not issued any books. (Database: LIS)

Solution:

```
SELECT student_fname, student_lname
FROM students
WHERE roll_no in (SELECT roll_no
                  FROM members
                  WHERE member_no IN ((SELECT member_no
                                      FROM members
                                      WHERE member_type = 'RS')
                                     EXCEPT
                                      (SELECT DISTINCT member_no
                                      FROM book_issue))
                 )
```

29. Write a SQL statement to find out the name of the oldest player from the team named “All Stars”. (Database: FLIS)

Solution:

```
SELECT players.name
FROM players
WHERE players.dob = (SELECT MIN(pl.dob)
                     FROM players pl, teams te
                     WHERE te.team_id = pl.team_id
                     AND te.name = 'All Stars')
```

30. Write an SQL statement to find the unique book titles which are issued to “PG” students but not to “UG” students.(Database: LIS)

Solution:

```
SELECT title
FROM book_catalogue
WHERE isbn_no IN (SELECT isbn_no
                  FROM book_issue NATURAL JOIN book_copies
                  WHERE member_no IN (SELECT member_no
                                      FROM members
                                      NATURAL JOIN book_issue
                                      WHERE member_type = 'PG')
                EXCEPT
                 SELECT isbn_no
                 FROM book_issue
                 NATURAL JOIN book_copies
                 WHERE member_no IN (SELECT member_no
                                     FROM members NATURAL JOIN book_issue
                                     WHERE member_type = 'UG')
                 )
```