BSCCS2001: Mock Quiz with Solutions
Weeks 1 to 4

1 Objective Questions

1. Which of the following option aptly describes the scalability property of DBMS?

[Bhaskar:MCQ:2points]

- √ The system can efficiently adapt to a rapidly growing data set.
- ◯ The data survives even if the system crashes abruptly.
- ◯ The business logic associated with the data is always kept intact.
- ◯ None of the above

Solution:
In the context of DBMS, **scalability** means the efficient adaptation of performance of the system to a rapidly growing data set.

2. Consider the entity set given in Figure 1. [Arup: MCQ: 3 points]

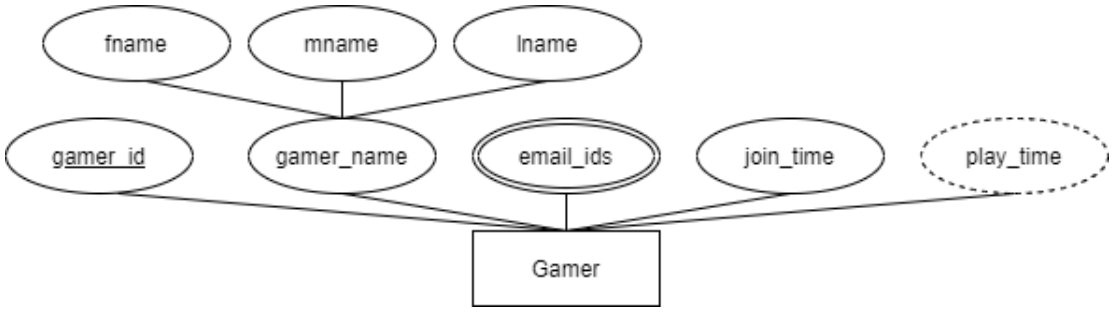


Figure 1: Entity set

Identify the statement which CORRECTLY describes the attributes of the entity set **Gamer**.

- √ gamer_id: primary key
  gamer_name: composite attribute
  email_ids: multivalued attribute
  play_time: derived attribute

◯ gamer_id: primary key
gamer_name: composite attribute
play_time: multivalued attribute
join_time: derived attribute

◯ play_time: primary key
gamer_name: composite attribute
gamer_id: multivalued attribute
fname: derived attribute

◯ email_ids: primary key
fname: composite attribute
email_ids: multivalued attribute
gamer_name: derived attribute

Solution: The different attributes are identified as shown in Figure 2.

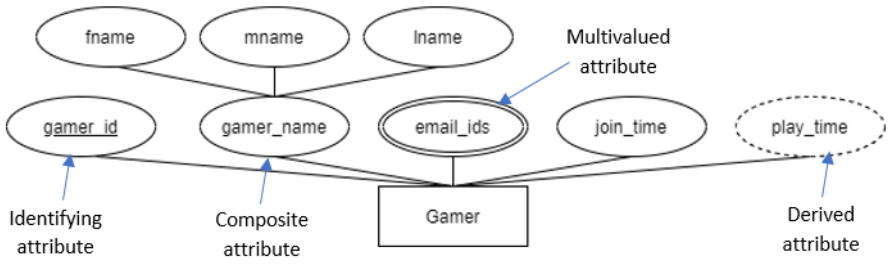


Figure 2: Identification of different types of attributes

3. Consider the following statements:

- SQL can be used to perform complex arithmetic operations.
- File based data management systems ensure consistency of data all the time.

The number of correct statements is

[Bhaskar:MCQ:2 point]

Ans: 0

Solution:
Performing complex arithmetic operations are difficult using SQL.
File based data management systems may not be able to ensure consistency of data all the time, since the relationships and constraints are enforced via file handling

programs. On the other hand, a DBMS has in-built mechanisms to ensure all-time consistency of data.
Therefore, none of the given statements are true.

4. Using the table **Students** in Figure 3, choose the correct SQL statement that will return the table shown in Figure 4.

Students			
Name	Age	Country	Score
Tom	13	Australia	70
Lucy	15	Scotland	95
Frank	16	Germany	76
Jane	13	Australia	49
Robert	16	Germany	93
Ryan	18	Ireland	56
Mike	13	Germany	84

Figure 3: Table Students

Country	Sum
Australia	119
Germany	253

Figure 4: Resultant table

[Harshit: MCQ: 4 points]

◯
```
SELECT Country, SUM(Score) AS Sum
FROM Students
GROUP BY Age HAVING AVG(Score)>100 AND AVG(Age)>12;
```

◯
```
SELECT Country, SUM(Score) AS Sum
FROM Students
GROUP BY Age HAVING SUM(Score)>150 AND AVG(Age)>13;
```

◯
```
SELECT Country, SUM(Score) AS Sum
FROM Students
GROUP BY Country HAVING AVG(Score)>70 AND AVG(Age)>13;
```

√
```
SELECT Country, SUM(Score) AS Sum
FROM Students
GROUP BY Country HAVING SUM(Score)>100 AND AVG(Age)>12;
```

Solution: The GROUP BY clause is used to group data based on specific values of the given attribute.
The SUM function returns the total sum of a numeric column.
The resultant table will be fetched by applying GROUP BY clause on Country that

together will have sum of score greater than 100 and the average age is more than 12.

5. Use the tables in Figure 5 to answer the question that follows.

suppliers

sup_num	sup_name
1001	Able
1002	Peter
1003	Molina
1004	Nikki

parts

part_num	sup_num	part_qty
301	1001	32
301	1004	17
301	1002	41
302	1002	11
302	1003	36
302	1001	16
303	1004	25
304	1002	35
304	1003	40

Figure 5: Table **suppliers** and table **parts**

Let $\{sup\_num\}$ be the primary key of table **suppliers** and $\{part\_num, sup\_num\}$ be the primary key of table **parts**. [Arup: NAT: 4 points]
Consider the SQL query given below:

```
SELECT s.sup_num, sum(p.part_qty)
FROM suppliers s, parts p
WHERE p.part_qty > 30
GROUP BY s.sup_num
```

How many rows will be returned by the above SQL query?
Answer: 4

Solution: As per the given SQL statement, it first obtains the Cartesian product between **suppliers** and **parts**, which include all possible combinations from both the tables.
The part of the statement:
`WHERE p.part_qty > 30`
eliminates the rows which do not satisfy the condition. The output is as shown below:

s.sup_num	s.sup_name	p.part_num	p.sup_num	p.part_qty
1001	Able	301	1001	32
1001	Able	301	1002	41
1001	Able	302	1003	36
1001	Able	304	1002	35
1001	Able	304	1003	40
1002	Peter	301	1001	32
1002	Peter	301	1002	41
1002	Peter	302	1003	36
1002	Peter	304	1002	35
1002	Peter	304	1003	40
1003	Molina	301	1001	32
1003	Molina	301	1002	41
1003	Molina	302	1003	36
1003	Molina	304	1002	35
1003	Molina	304	1003	40
1004	Nikki	301	1001	32
1004	Nikki	301	1002	41
1004	Nikki	302	1003	36
1004	Nikki	304	1002	35
1004	Nikki	304	1003	40

Finally, the part of the statement:
`SELECT s.sup_num, sum(p.part_qty) ...GROUP BY s.sup_num`
results in:

s.sup_num	p.part_qty
1001	184
1002	184
1003	184
1004	184

Thus, the result has 4 rows.

6. Consider relations as given below: [Dhannya: MCQ: 2 points]

```
userMaster(userid,userName,email,phone)
passwHistory(userid,password,status)
```

Which among the following SQL statements will return the valid password(s) of user(s) whose `userid` ends with "001"?

◯ `SELECT password FROM userMaster OUTER JOIN passwHistory WHERE userMaster.userid LIKE "%001" AND status = "VALID";`

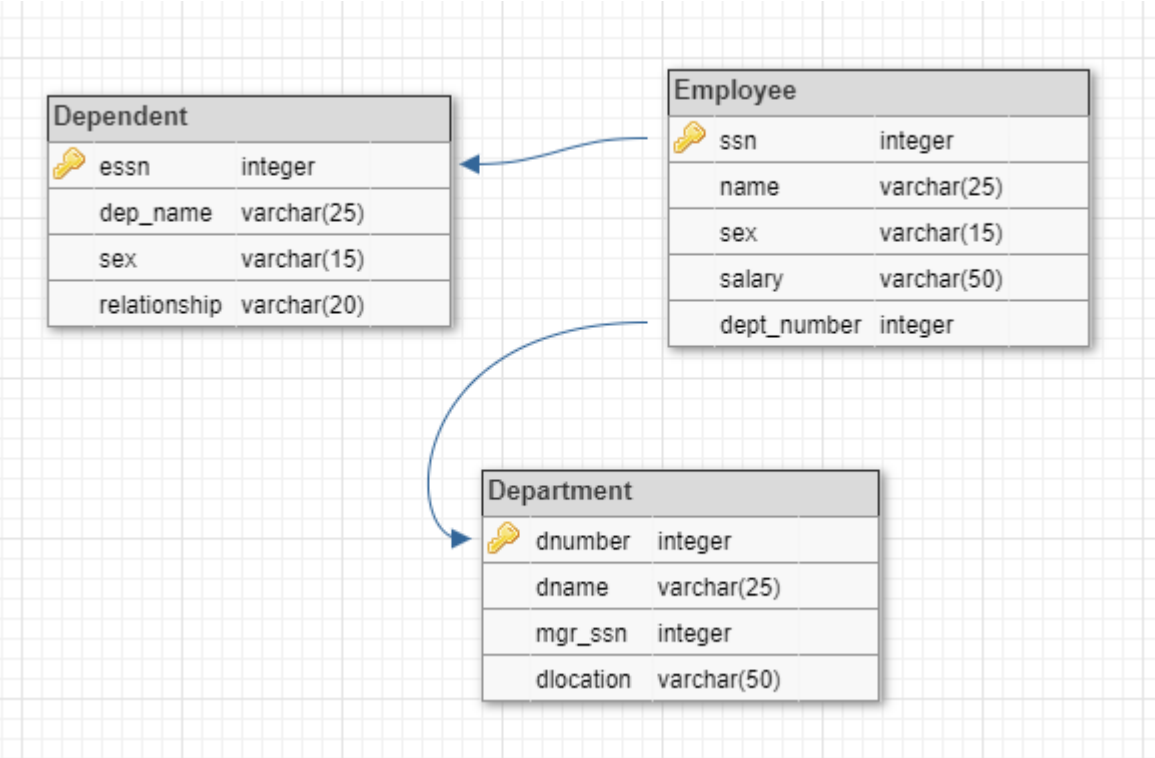
◯ `SELECT password FROM userMaster, passwHistory WHERE userMaster.userid LIKE "%001" AND status = "VALID";`

√ `SELECT password FROM userMaster NATURAL JOIN passwHistory WHERE userMaster.userid LIKE "%001" AND status = "VALID";`

◯ `SELECT password FROM userMaster INNER JOIN passwHistory WHERE userMaster.userid LIKE "%001" AND status = "VALID";`

Solution: `NATURAL JOIN` is a join operation that joins two tables using the common columns of the two tables. Here, the common column is `userid` and hence the `NATURAL JOIN` finds rows with same values for `userid`. The other two constraints are added as conditions under the `WHERE` clause.

7. Consider tables **Employee**, **Department** and **Dependent**, and answer the question that follows. [Harshit: MCQ: 3 points]



Choose the correct query to retrieve the name of each employee who has a dependent with the same name and the same sex as that of the employee.

◯ `SELECT name FROM Dependent as E WHERE EXISTS ( SELECT * FROM Employee as D WHERE E.ssn = D.essn OR E.sex = D.sex AND E.name = D.dep_name );`

◯ `SELECT name FROM Employee as E WHERE NOT IN ( SELECT * FROM Employee as D WHERE E.ssn = D.essn OR E.sex = D.sex AND E.name = D.dep_name );`

√ `SELECT name FROM Employee as E WHERE EXISTS ( SELECT * FROM Dependent as D WHERE E.ssn = D.essn AND E.sex = D.sex AND E.name = D.dep_name );`

◯ `SELECT name FROM Dependent as D WHERE NOT EXISTS ( SELECT * FROM Employee as E WHERE E.ssn = D.essn AND E.sex = D.sex AND E.name = D.dep_name );`

Solution: The EXISTS condition in SQL is used to check whether the result of a correlated nested query is empty (contains no tuples) or not.
SQL NOT IN operator is used to filter the result if the values that are mentioned as part of the IN operator is not satisfied.
As we have to retrieve the name of each employee who has a dependent with certain given condition, the outer query needs to fetch data from the table Employee and inner query from the table Dependent. Hence Option 1 and 4 are incorrect.

For option 3 - The inner query will retrieve all the data from the table dependent, where the name, sex and ssn of each employee is matched with the dependent. Hence, it will return something and the EXISTS condition will become TRUE, and based on that, the outer query will fetch the name(s) of those employees who has a dependent with the same first name and is of the same sex as that of the employee.

For option 2 - In the inner query, the condition of ssn and sex are separated by OR (it has to be 'AND' as per the question's requirement). Also, the NOT IN keyword will look for the values which is not in the set returned by the inner query. Hence the desired result will not be fetched.

8. Consider the relational schema given in Figure 6.

capital
countryID(varchar)
capitalID(varchar)
capitalName(varchar)

country
countryID(varchar)
continent(varchar)
countryName(varchar)

Figure 6: Country Capitals Relational Schema

Choose the correct option(s) to fill in the blanks in the query given below to find the capitals of all those countries that belong to both Asia and Europe.

[Dhannya: MSQ: 4 points]

```
SELECT capitalName FROM capital
WHERE _______________ (
        (SELECT countryID FROM country
        WHERE continent ='Asia'
        AND capital.countryid = country.countryid)
        ---------------
        (SELECT countryID FROM country
        WHERE continent ='Europe'
        AND capital.countryid = country.countryid));
```

√ EXISTS, INTERSECT

◯ NOT EXISTS, UNION

◯ EXISTS, UNION

◯ NOT EXISTS, INTERSECT

Solution: The inner query of the first option returns all countries which belong to Asia and Europe. Hence it is correct.
The inner query of the second option returns all countries that are not in the set of countries that belong to either Asia or Europe. This is incorrect.
The inner query of the third option returns all countries that belong to either Asia or Europe. We need countries that belong to both and hence this option is also incorrect.
The inner query of the fourth option filters out those countries that belong to both Asia and Europe. This is also incorrect.

9. Consider the relational schema given in Figure 7 and answer the question that follows.

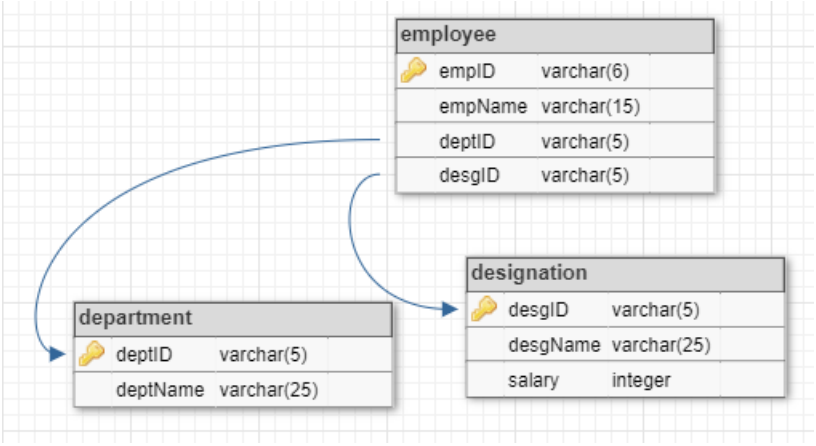


Figure 7: Employee Schema

If the relations **employee, designation** and **department** have 100, 6, 5 rows respectively, what is the difference between the maximum and the minimum number of rows returned by the following query? [Dhannya: NAT: 3 points]

```
SELECT * FROM employee LEFT OUTER JOIN designation
ON employee.desgID = designation.desgID;
```

Answer: 0

Solution: Left outer join (also known as Left join) returns all tuples returned by natural join along with those tuples in the left table (here, `employee`) that does not have matching entry in the right table. In the given question, however, `desgID` is the foreign key in Table `employee` that references Table `designation`. Therefore, there will not be any tuple in the left table that does not have a matching entry in the right table. Thus, the maximum number of rows returned by the left join in the given example is 100.
The case when `employee` table has no rows is the case when left outer join will have the minimum number of rows. In this case, however, the `employee` table has 100 rows. So, there will be at least 100 rows returned by the left join.
The answer is $100 - 100 = 0$.

10. Which of the following options is equivalent to the SQL query shown below?

[Piyush/Dhannya:MCQ:2points]

```
SELECT DISTINCT T.name
FROM instructor as T, instructor as S
WHERE T.salary < S.salary and S.dept_name = 'Finance';
```

√
```
SELECT DISTINCT name
FROM instructor
WHERE salary < SOME (SELECT salary
FROM instructor
WHERE dept_name = 'Finance');
```

◯
```
SELECT DISTINCT name
FROM instructor
WHERE salary < ALL (SELECT salary
FROM instructor
WHERE dept_name = 'Finance');
```

◯
```
SELECT DISTINCT name
FROM instructor
WHERE salary <= ALL (SELECT salary
FROM instructor
WHERE dept_name = 'Finance');
```

◯
```
SELECT DISTINCT name
FROM instructor
WHERE salary <= SOME (SELECT salary
FROM instructor
WHERE dept_name = 'Finance');
```

11. Given the relations R_1, R_2 as shown in Figure 8, choose the relation that results from $R_1 \div R_2$.

[Dhannya:MCQ:4 points]

A	B
a_1	b_1
a_1	b_2
a_2	b_1
a_3	b_1
a_3	b_2
a_1	b_3
a_2	b_3

Relation R_1

A
a_1
a_2
a_3

Relation R_2

Figure 8: Relations R_1 and R_2

○

B
b_1
b_2
b_3

○

B
a_1
a_2
a_3

√

B
b_1

○

A
a_1

12. Consider the entity sets **MOVIE** and **PERSON** with a many-to-many relationship set **WATCHES** as shown in Figure 9 and answer the question that follows.

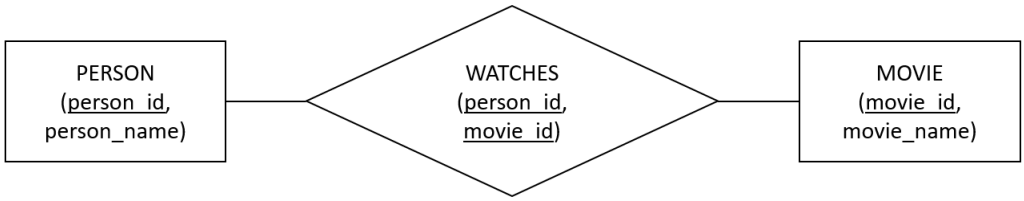


Figure 9: E-R diagram

If we wish to store the ticket number of a ticket purchased by a person for a movie, then in which among the following must this information appear?

[Harshit/Arup/Dhannya: MCQ: 3 points]

- ◯ PERSON
- √ WATCHES
- ◯ MOVIE
- ◯ WATCHES or PERSON
- ◯ WATCHES or MOVIE
- ◯ PERSON or MOVIE
- ◯ WATCHES or PERSON or MOVIE

Solution: Since there is a many-to-many relationship between PERSON and MOVIE, it follows that the ticket payment information must be added as a descriptive attribute of **WATCHES** relationship set.

13. Let X(A, B) and Y(C, D) be two relations with instances as shown in Figure 10.

[Piyush:MCQ:4 points]

X	
A	B
4	5
5	4
6	6

Y	
C	D
4	5
6	7
6	8

Figure 10: Relations **X** and **Y**

Find the number of tuples returned by the following query:
$\Pi_A(\sigma_{B=C}(X \times Y))$ - $\Pi_A(\sigma_{A=D}(X \times Y))$

- ◯ 9
- ◯ 3
- ◯ 2
- √ None of the above

Solution: $(X \times Y)$

A	B	C	D
4	5	4	5
4	5	6	7
4	5	6	8
5	4	4	5
5	4	6	7
5	4	6	8
6	6	4	5
6	6	6	7
6	6	6	8

$\Pi_A(\sigma_{B=C}(X \times Y))$

A
5
6

$\Pi_A(\sigma_{A=D}(X \times Y))$

A
5

$\Pi_A(\sigma_{B=C}(X \times Y))$ - $\Pi_A(\sigma_{A=D}(X \times Y))$

A
6

So,the number of tuples is 1. Thus, option 4 is correct.

2 SQL Query Questions

14. Using the schema shown in Figure 11, answer the question that follows.

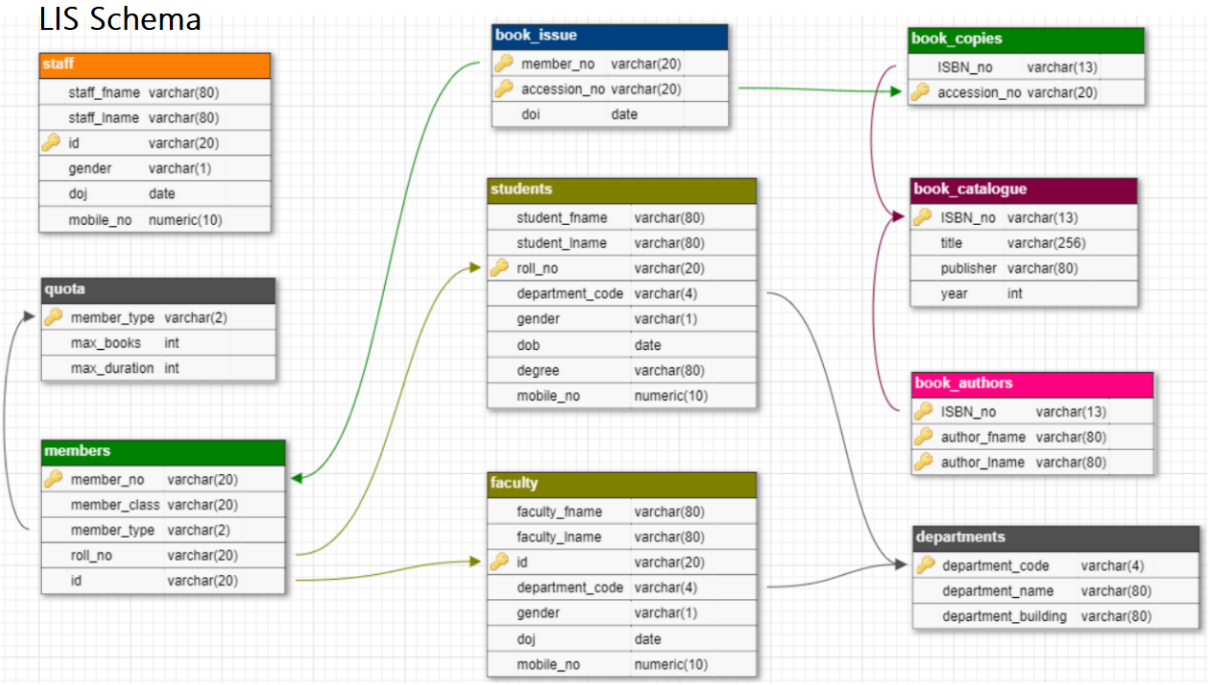


Figure 11: LIS Schema

Write an SQL statement to find the first name of author who has published more than once in the year 2017. [QQ:5 points]

[Database:LIS]

Solution:

```
select author_fname
from book_authors a, book_catalogue b
where a.isbn_no = b.isbn_no and
b.year = 2017
group by a.author_fname having count(year)>1
```

15. Using the schema shown in Figure 12, answer the question that follows.

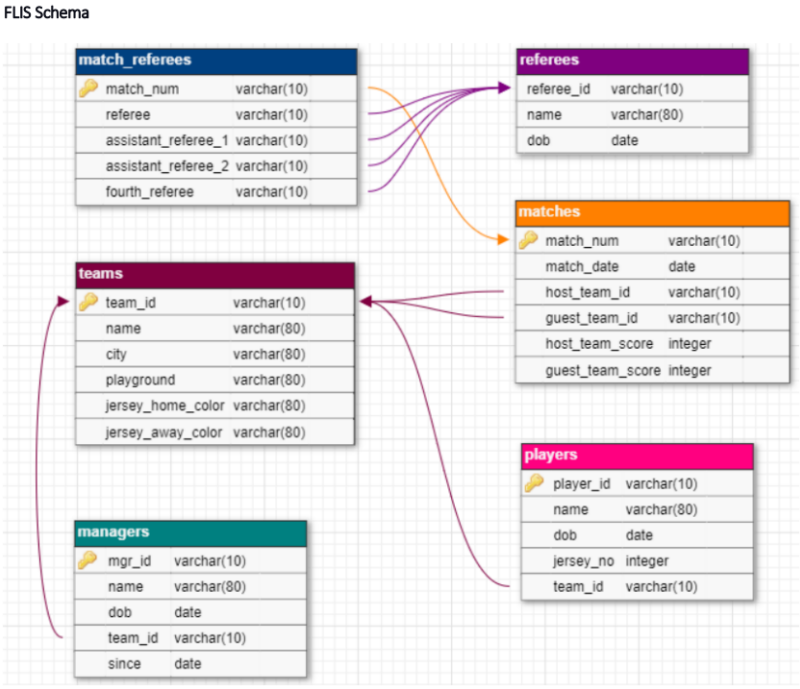


Figure 12: FLIS Schema

Write an SQL statement to find the names of players of teams that have played matches as guests between 10th May 2020 and 12th May 2020 (including both dates).

[QQ:5 points]

[Database:FLIS]

Solution:

```
select name from players p, matches m
where p.team_id = m.guest_team_id and
m.match_date <= '2020-05-12' and m.match_date >= '2020-05-10'
```

OR

```
select name from players p, matches m
where p.team_id = m.guest_team_id and
m.match_date between '2020-05-10' and '2020-05-12'
```