

BSCCS2003: Graded Questions with Solutions

Week 4

1. Consider the following two tables: [MCQ : 1 point]

Table 1: Instructors

Instructor_ID	Instructor_Name	Department_Name
101	Amit Dubey	Physics
102	Sarthak Gaur	English
103	Neha Sharma	English
104	Sumit Kumar	Chemistry
105	Himanshi Mehra	Mathematics

Table 2: Departments

Department_Name	Building_Name
English	B1
Mathematics	B2
Physics	B3
Chemistry	B2
Finance	B5

Which of the following queries will list the names of those instructors whose department is “English” or whose building is either ‘B1’ or ‘B2’?

- ☐ SELECT Instructor_Name from Instructors i, Departments d
WHERE d.Department_Name = i.Department_Name
and (i.Department_Name = ‘English’ or d.Building_Name = ‘B1’) and d.Building_Name = ‘B2’;
- ☐ SELECT Instructor_Name from Instructors i, Departments d
WHERE (i.Department_Name = ‘English’ or d.Building_Name in (‘B1’, ‘B2’));
- ✓ ☒ SELECT Instructor_Name from Instructors i, Departments d
WHERE d.Department_Name = i.Department_Name
and (i.Department_Name = ‘English’ or d.Building_Name in(‘B1’, ‘B2’));
- ☐ All of the above

Solution:

Option 1: The first query will result in no records.

Option 2: The second query will result in a Cartesian product and generate wrong output.

Option 3: The third query will result in the correct result, i.e., instructors names who belong to “English” department or whose building is either ‘B1’ or ‘B2’.

2. Consider the following two tables: [MCQ : 1 point]

Table 1: Instructors

Instructor_ID	Instructor_Name	Department_Name
101	Amit Dubey	Physics
102	Sarthak Gaur	English
103	Neha Sharma	English
104	Sumit Kumar	Chemistry
105	Himanshi Mehra	Mathematics

Table 2: Departments

Department_Name	Building_Name
English	B1
Mathematics	B2
Physics	B3
Chemistry	B2
Finance	B5

Which of the following queries will list the names of those instructors whose department is in the building 'B1'?

- ☒ `SELECT Instructor_Name from Instructors i INNER JOIN Departments d
ON i.Department_Name = d.Department_Name
WHERE d.Building_Name = 'B1';`
- ☐ `SELECT Instructor_Name from Instructors i INNER JOIN Departments d
ON i.Department_Name = d.Building_Name
WHERE d.Building_Name = 'B1';`
- ☐ `SELECT Instructor_Name from Instructors i INNER JOIN Departments d
ON i.Department_Name = d.Department_Name
WHERE i.Building_Name = 'B1';`
- ☐ All of the above

Solution:

Option 1: The first query will result "Sarthak Gaur" and "Neha Sharma" as an inner join will be performed over a common column named "Department_Name" in both of the tables.

Option 2: The second query will show no results, as there is no entry in the "Instructors" table whose "Department_Name" matches with the column "Building_Name" in table "Departments".

Option 3: The third query will generate an error, as there is no attribute called "Building_Name" in the table "Instructors".

3. Is it mandatory for a table to have a primary key? [MCQ: 1 point]

- ☐ Yes, because each table is uniquely identified by its primary key.
- ✓ ☒ No, because databases can have tables with all non-unique fields, they can exist even without a primary key.
- ☐ Yes, because each table should have a field with unique entries.
- ☐ No, as the primary key is automatically set by the table if not given explicitly.

Solution: It is not necessary for a table to have a primary key, as tables can be designed with no primary key set. A primary key does not define any table uniquely in the database, but can define a field or a group of fields collectively within the table. Primary keys are set by the user or the creator of the table, and is not automatically set by the table in case it is not set by the creator.

4. Consider the following 2 tables:

Table 1: Employee	Employee_ID	Employee_Name	Age	PhoneNo
	1	Saurav Lokesh	23	9871768118
	2	Garima Das	25	9987712611
	3	Swaroop Kumar	21	9863674830
	4	Sumita Verma	25	9872618731

Table 2 : Project	Project_ID	Employee_ID	Client_ID	Project_Name
	101	1	4	Project1
	102	2	8	Project2
	103	3	3	Project3
	104	4	7	Project4

Which of the following queries will pull out all the Project names and Project id with their corresponding Employees names only?

[MCQ : 3 Points]

- ☐ SELECT Employee.Employee_ID,
Employee.Employee_Name, Project.Project_ID,
Project.Project_Name
FROM Employee
INNER JOIN Project
ON Employee.Employee_ID=Project.Employee_ID;
- ✓ ☒ SELECT Project.Project_Name,
Project.Project_ID, Employee.Employee_Name

```
FROM Employee
INNER JOIN Project
ON Employee.Employee_ID=Project.Employee_ID;
```

- ☐ SELECT Employee.Employee_ID,
Employee.Employee_Name, Project.Project_ID,
Project.Project_Name,
FROM Employee,
INNER JOIN Project ,
ON Employee.Employee_ID=Project.Employee_ID;
- ☐ None of the above

Solution: The correct INNER JOIN syntax is :
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;

In option (i) and (iii) this syntax is violated.

5. Consider the following tables.

[MCQ : 1 point]

1) Table 1: records

record_id	matches	runs	highest_score	format	average	player_id
1	39	2679	212	test	46.19	1
2	227	9206	264	odi	48.96	1
3	92	7547	254	test	52.05	2
4	254	12169	183	odi	59.07	2
5	114	8765	278	test	50.66	3
6	228	9577	176	odi	53.55	3
7	77	7540	239	test	61.8	4
8	128	4378	164	odi	43.35	4
9	86	7311	335	test	48.1	5
10	128	5455	179	odi	45.08	5

2) Table 2: players

player_id	country	name	role
1	India	Rohit Sharma	opening_batsman
2	India	Virat Kohli	middle_order_batsman
3	South Africa	AB de Villiers	middle_order_batsman
4	Australia	Steven Smith	middle_order_batsman
5	Australia	David Warner	opening_batsman

What will be the output of the following code?

```
SELECT players.country
FROM players
INNER JOIN records ON records.player_id = players.player_id
GROUP BY country
ORDER BY SUM(records.runs) DESC
LIMIT 1;
```

☒ India

☐ Australia

☐ South Africa

☐ Code will produce an error

Solution: The query will produce the name of the country with the highest aggregate runs. In this case, it is India.

6. Consider the following tables.

[MCQ : 2 points]

1) Table 1: records

record_id	matches	runs	highest_score	format	average	player_id
1	39	2679	212	test	46.19	1
2	227	9206	264	odi	48.96	1
3	92	7547	254	test	52.05	2
4	254	12169	183	odi	59.07	2
5	114	8765	278	test	50.66	3
6	228	9577	176	odi	53.55	3
7	77	7540	239	test	61.8	4
8	128	4378	164	odi	43.35	4
9	86	7311	335	test	48.1	5
10	128	5455	179	odi	45.08	5

2) Table 2: players

player_id	country	name	role
1	India	Rohit Sharma	opening_batsman
2	India	Virat Kohli	middle_order_batsman
3	South Africa	AB de Villiers	middle_order_batsman
4	Australia	Steven Smith	middle_order_batsman
5	Australia	David Warner	opening_batsman

What will be the output of following code?

```
SELECT players.name
FROM players
INNER JOIN records ON records.player_id = players.player_id
WHERE players.role = "opening_batsman" and records.format = "odi"
ORDER BY records.highest_score DESC
LIMIT 1;
```

✓ Rohit Sharma

- ☐ AB de Villiers
- ☐ Virat Kohli
- ☐ David warner

Solution: The query will produce the name of the player with highest “highest_score” in odi and who is also an opening batsman.

7. Consider the following tables.

[MCQ : 3 points]

1) Table 1: records

record_id	matches	runs	highest_score	format	average	player_id
1	39	2679	212	test	46.19	1
2	227	9206	264	odi	48.96	1
3	92	7547	254	test	52.05	2
4	254	12169	183	odi	59.07	2
5	114	8765	278	test	50.66	3
6	228	9577	176	odi	53.55	3
7	77	7540	239	test	61.8	4
8	128	4378	164	odi	43.35	4
9	86	7311	335	test	48.1	5
10	128	5455	179	odi	45.08	5

2) Table 2: players

player_id	country	name	role
1	India	Rohit Sharma	opening_batsman
2	India	Virat Kohli	middle_order_batsman
3	South Africa	AB de Villiers	middle_order_batsman
4	Australia	Steven Smith	middle_order_batsman
5	Australia	David Warner	opening_batsman

Which of the following queries will output the names of batsmen with the top three averages in test matches and has played more than 75 test matches?

- ☐

```
SELECT players.name
FROM players
INNER JOIN records ON records.player_id = players.player_id
ORDER BY records.average
```

```
WHERE records.matches > 75 and records.format = "test"
LIMIT 3;
```

- ```
SELECT players.name
FROM players
INNER JOIN records ON records.player_id = players.player_id
WHERE records.matches > 75 and records.format = "test"
ORDER BY records.average
LIMIT 3;
```
- ```
SELECT players.name
FROM players
INNER JOIN records ON records.player_id = players.player_id
ORDER BY records.average DESC
WHERE records.matches > 75 and records.format = "test"
LIMIT 3;
```
- ✓

```
SELECT players.name
FROM players
INNER JOIN records ON records.player_id = players.player_id
WHERE records.matches > 75 and records.format = "test"
ORDER BY records.average DESC
LIMIT 3;
```

Solution: option 1 and 3 will produce error because WHERE clause should come before the ORDER BY clause.

option 2 will produce the top 3 batsmen with lower average in test matches and has played more than 75 test matches.

option 4 will produce the top 3 batsmen with higher average in test matches and has played more than 75 test matches.