1.Merit Rewards

1. Merit Rewards

On the basis of merit, a company decides to promote some of its employee in its HR division at the end of the quarter because of their high performance. Write a query to find the employee IDs along with the names of all its employees who work in the HR department who earned a bonus of 5000 dollars or more in the last quarter.

There are two tables in the database: *employee_information* and *last_quarter_bonus*. Their primary keys are *employee_ID*.

▾ Schema

There are 2 tables: employee_information, last_quarter_bonus.

employee_information		
Name	Type	Description
employee_ID	INTEGER	The employee ID of the employee. This is the primary key.
name	STRING	The name of the employee.
division	STRING	The division in which the employee works.

last_quarter_bonus		
Name	Type	Description
employee_ID	INTEGER	The employee ID of the employee. This is the primary key.
bonus	INTEGER	The bonus earned by employee in last quarter (in dollars).

Note: Both tables contain data about all employees working in the company.

▾ Sample Data Tables

employee_information		
employee_ID	name	division
1	Julia	HR

last_quarter_bonus		
Name	Type	Description
employee_ID	INTEGER	The employee ID of the employee. This is the primary key.
bonus	INTEGER	The bonus earned by employee in last quarter (in dollars).

Note: Both tables contain data about all employees working in the company.

▾ Sample Data Tables

employee_information		
employee_ID	name	division
1	Julia	HR
2	Samantha	Tech
3	Richard	HR

last_quarter_bonus	
employee_ID	bonus
1	2000
2	5500
3	6240

Sample Output

```
3 Richard
```

Explanation

- There are two employees working in the HR department, with employee IDs 1 and 3. However, only employee with ID 3 has a bonus greater than equal to 5000, and hence information about only that employee is displayed.
- Employee 2, despite having a bonus of more than 5000 is not displayed because he does not belong to the HR department.

Language MySQL · Environment · Autocomplete Ready

```mysql
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT ei.employee_ID, ei.name
FROM employee_information ei
JOIN last_quarter_bonus lqb ON lqb.employee_ID = ei.employee_ID
WHERE ei.division = "HR" AND lqb.bonus >= 5000;
```

Line: 8 Col:

Test Results — Run Query — Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+-------------+-------------+
| employee_ID | name        |
+-------------+-------------+
|           1 | Kim         |
|           2 | Reginald    |
|           9 | Wendy       |
|          27 | Meredith    |
|          40 | Anna        |
|          79 | Sharon      |
|          82 | Kristin     |
|          96 | Brooke      |
|         101 | Cynthia     |
```

Language MySQL · Autocomplete Loading... · Environment

```mysql
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT ei.employee_ID, ei.name
FROM employee_information ei
JOIN last_quarter_bonus lqb ON lqb.employee_ID = ei.employee_ID
WHERE ei.division = "HR" AND lqb.bonus >= 5000;
```

Test Results — Run Query — Submit

Compiled successfully. Correct answer.

Test case 0

Expected Output — Download

```
1 Kim
2 Reginald
9 Wendy
27 Meredith
40 Anna
79 Sharon
82 Kristin
96 Brooke
101 Cynthia
110 Rhonda
113 Brian
```

2.Student Analysis

2. Student Analysis

A school recently conducted its annual examination and wishes to know the list of academically low performing students to organize extra classes for them. Write a query to return the roll number and names of students who have a total of less than 100 marks including all 3 subjects.

There are two tables: *student_information and examination_marks. Their primary keys are roll_number.*

▾ Schema

You are provided 2 tables: student_information, examination_marks.

student_information		
Name	Type	Description
roll_number	INTEGER	The roll number of the student. This is the primary key.
name	STRING	The name of the student.

examination_marks		
Name	Type	Description
roll_number	INTEGER	The roll number of the student. This is the primary key.
subject_one	INTEGER	The marks of the student in first subject.
subject_two	INTEGER	The marks of the student in second subject.
subject_three	INTEGER	The marks of the student in third subject.

Note: Both tables contain data about all students enrolled in the school.

▾ Sample Data Tables

student_information	
roll_number	name
1	Sheila
2	Rachel
3	Christopher

examination_marks			
roll_number	subject_one	subject_two	subject_three
1	32	48	64
2	24	21	25
3	55	12	10

Sample Output

```
2 Rachel
3 Christopher
```

Explanation

- The cumulative marks of student with roll numbers 1, 2 and 3 are 144, 70 and 77. Since student 2 and 3 have a total of less than 100, their names and roll numbers are displayed.

Language: MySQL | Environment | Autocomplete Ready

```
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT si.roll_number, si.name
FROM student_information si
JOIN examination_marks em ON si.roll_number = em.roll_number
WHERE (em.subject_one + em.subject_two + em.subject_three) < 100;

```

Line: 10 Col: 1

Test Results — Run Query | Submit

Compiled successfully. Correct answer.

Test case 0

Your Output (stdout)

```
+-------------+-----------+
| roll_number | name      |
+-------------+-----------+
|           5 | Mark      |
|          13 | Robert    |
|          18 | Bob       |
|          37 | Melanie   |
|          42 | Albert    |
|          56 | Diane     |
|          58 | Brandon   |
|          67 | Alexander |
|          78 | Gregory   |
```

Language: MySQL | Environment | Autocomplete Ready

```
/*
Enter your query below.
Please append a semicolon ";" at the end of the query
*/
SELECT si.roll_number, si.name
FROM student_information si
JOIN examination_marks em ON si.roll_number = em.roll_number
WHERE (em.subject_one + em.subject_two + em.subject_three) < 100;

```

Line: 10 Col: 1

Test Results — Run Query | Submit

Compiled successfully. Correct answer.

Test case 0

```
172 Richard
178 Jason
183 Ronald
189 Jason
194 Andrew
198 Nichole
205 James
212 William
231 Lori
235 Tonya
237 Jeffery
245 Brittany
247 Amber
```