

Write syntax to create the following tables so they look like this:

A] students Table

Student_ID	Forename	Surname

B] exams Table

Exam_ID	Exam_Name	Max_Mark

C] results Table

Result_ID	Student_ID	Exam_ID	Mark

Example Query

```
CREATE TABLE students(  
    Student_ID integer PRIMARY KEY,  
    Forename varchar(50) NOT NULL,  
    Surname varchar(50) NOT NULL  
);  
  
CREATE TABLE exams(  
    Exam_ID integer PRIMARY KEY,  
    Exam_Name varchar(50) NOT NULL,  
    Max_Mark integer NOT NULL  
);  
  
CREATE TABLE results(  
    Result_ID integer PRIMARY KEY,  
    Student_ID integer NOT NULL,  
    Exam_ID integer NOT NULL,  
    Mark integer NOT NULL  
);
```

Write a query to list students' forenames and surnames where they scored higher than 60 and the respective exam.

Example Query

```
SELECT s.Forename, s.Surname, e.Exam_Name
```

```

FROM results as r
JOIN students as s
JOIN exams as e
WHERE r.Mark > 60
AND r.student_id = s.student_id
AND r.exam_id = e.exam_id;

```

Expected Output

	Forename	Surname	Exam_Name
▶	Belen	Badillo	Algorithms
	Ciara	Connelly	Algorithms
	Everly	Evans	Algorithms
	Belen	Badillo	Cyber Security
	Everly	Evans	Cyber Security
	Fabia	Fahim	Cyber Security

Write a query that checks for suspected collusion in an exam where students receive the same mark, and returns the students' full names, the suspected exam, and their mark.

*For simplicity, you can assume there won't be identical marks across different exams.
So if 62 is a mark in Exam 1, there won't be a 62 in Exam 2.*

Example Query

```

SELECT s.Forename, s.Surname, e.Exam_Name, r.mark
FROM results AS r
JOIN students AS s
JOIN exams AS e
WHERE r.mark IN (
    SELECT mark FROM results
    GROUP BY mark HAVING COUNT(*) > 1
)
AND r.student_id = s.student_id AND r.exam_id = e.exam_id;

```

Expected Output

	Forename	Surname	Exam_Name	mark
▶	Belen	Badillo	Algorithms	62
	Ciara	Connelly	Algorithms	62
	Belen	Badillo	Cyber Security	68
	Everly	Evans	Cyber Security	68