

# Q1. Retrieve all students enrolled in 'Computer Science' course

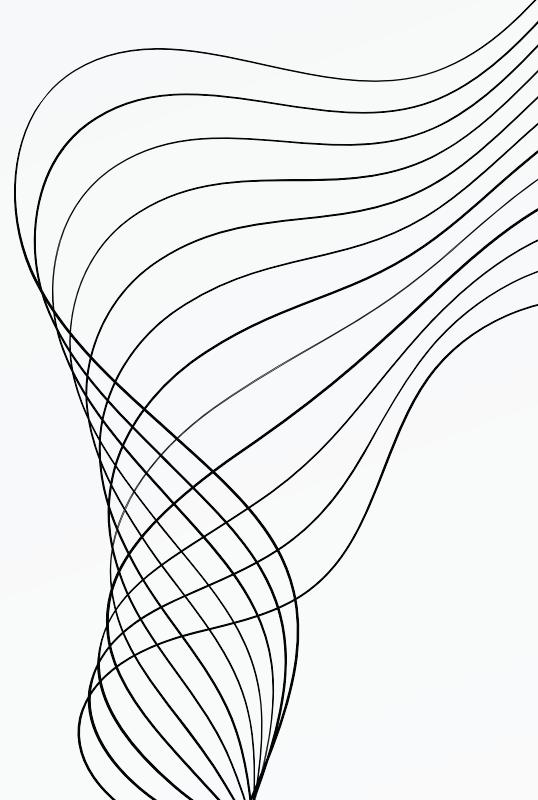


LARANA, INC.

```
SELECT S.SNAME, C.CNAME, E.ENROLL_DATE, E.GRADE  
      FROM STUDENT S  
  JOIN ENROLLMENT E ON S.SID = E.SID  
  JOIN COURSE C ON E.CID = C.CID  
 WHERE C.CNAME = 'Computer Science';
```

SNAME	CNAME	ENROLL_DATE	GRADE
Charlie	Computer Science	15-MAR-23	A
Alice	Computer Science	10-JAN-23	A

[Download CSV](#)



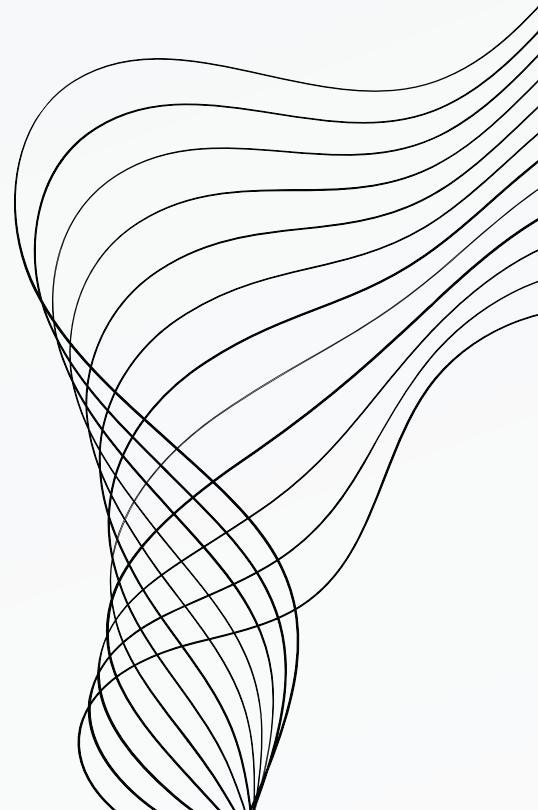
## 2. LIST STUDENTS AND THEIR GRADES FOR A SPECIFIC COURSE (E.G., 'PHYSICS')



LARANA, INC.

```
SELECT S.SNAME, E.GRADE  
FROM STUDENT S  
JOIN ENROLLMENT E ON S.SID = E.SID  
JOIN COURSE C ON E.CID = C.CID  
WHERE C.CNAME = 'PHYSICS';
```

SNAME	GRADE
Alice	A
Bob	A



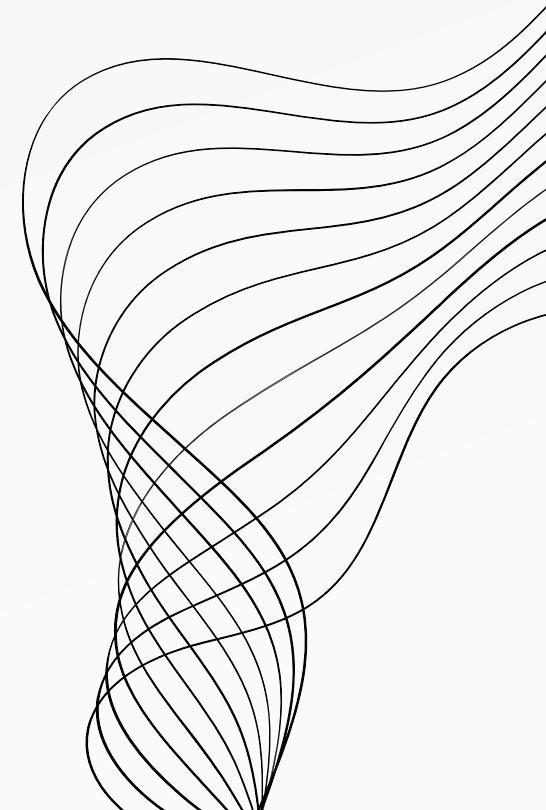
### 3. FIND THE TOTAL NUMBER OF STUDENTS ENROLLED IN EACH DEPARTMENT



LARANA, INC.

```
SELECT D.DNAME, COUNT(E.SID) AS STUDENT_COUNT  
      FROM DEPARTMENT D  
  JOIN COURSE C ON D.DEPT_ID = C.DEPT_ID  
  JOIN ENROLLMENT E ON C.CID = E.CID  
 GROUP BY D.DNAME;
```

DNAME	STUDENT_COUNT
Mathematics	1
Engineering	4
Arts	1
Science	2



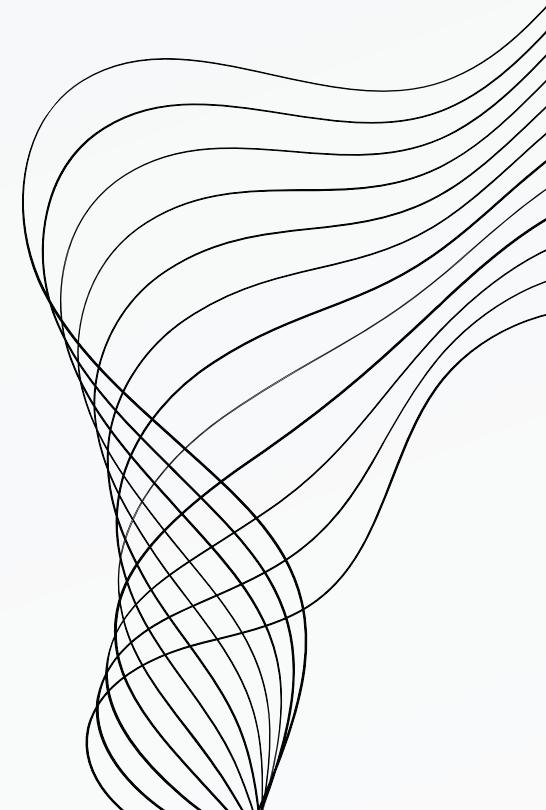
#### 4. FIND THE COURSES WITH THE HIGHEST ENROLLMENT



LARANA, INC.

```
SELECT C.CNAME, COUNT(E.SID) AS TOTAL_ENROLLMENTS  
      FROM COURSE C  
  JOIN ENROLLMENT E ON C.CID = E.CID  
 GROUP BY C.CNAME  
 ORDER BY TOTAL_ENROLLMENTS DESC;
```

CNAME	TOTAL_ENROLLMENTS
Physics	2
Computer Science	2
Data Structures	1
History	1
Calculus	1
Artificial Intelligence	1



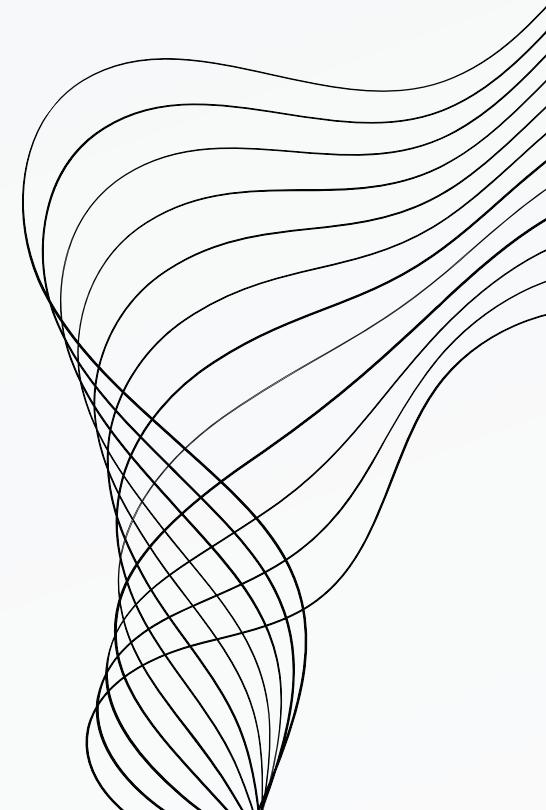
## 5. RETRIEVE THE TOP-PERFORMING STUDENTS FOR EACH COURSE



LARANA, INC.

```
SELECT C.CNAME, S.SNAME, E.GRADE  
FROM COURSE C  
JOIN ENROLLMENT E ON C.CID = E.CID  
JOIN STUDENT S ON E.SID = S.SID  
WHERE E.GRADE = 'A';
```

CNAME	SNAME	GRADE
Physics	Alice	A
Physics	Bob	A
Computer Science	Alice	A
Computer Science	Charlie	A
Calculus	Ethan Fox	A
Artificial Intelligence	George Harris	A



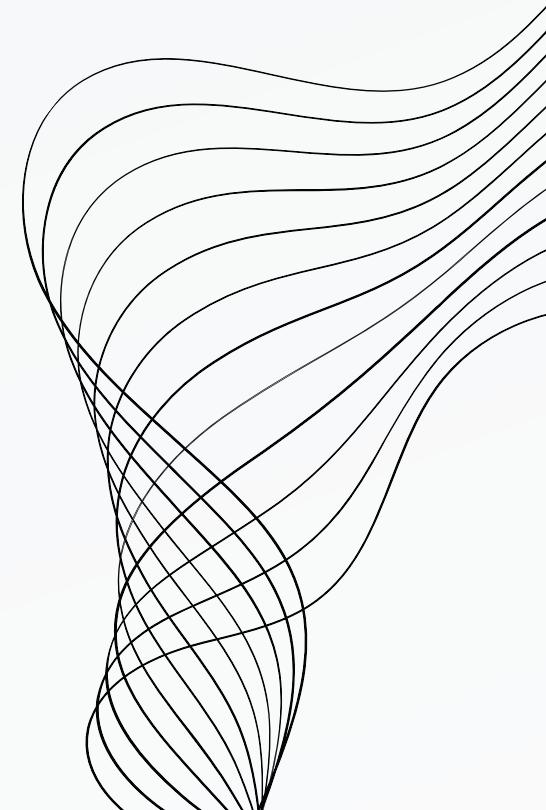
## 6. LIST ALL COURSES WITH STUDENTS WHO HAVE FAILED (ASSUME GRADE 'F')



LARANA, INC.

```
SELECT C.CNAME, S.SNAME, E.GRADE
FROM COURSE C
JOIN ENROLLMENT E ON C.CID = E.CID
JOIN STUDENT S ON E.SID = S.SID
WHERE E.GRADE = 'F';
```

CNAME	SNAME	GRADE
Physics	Alice	A
Physics	Bob	A
Computer Science	Alice	A
Computer Science	Charlie	A
Calculus	Ethan Fox	A
Artificial Intelligence	George Harris	A



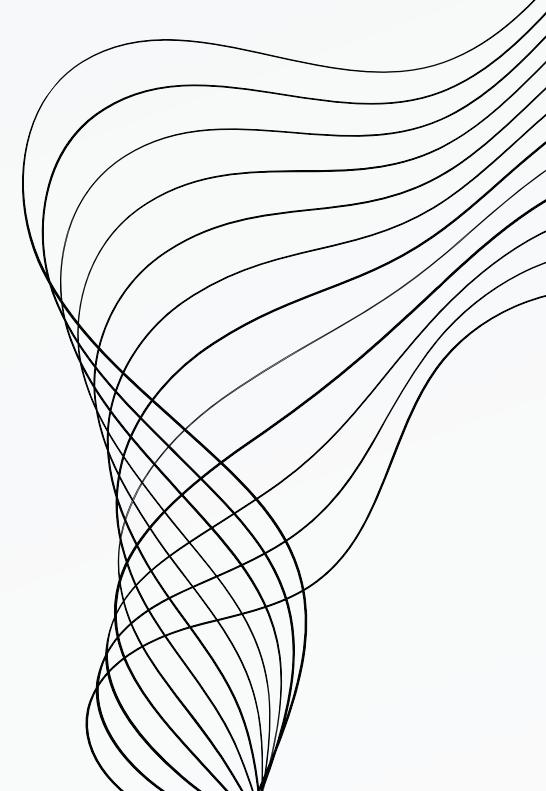
## 7. RETRIEVE DETAILS OF STUDENTS WHO HAVE NOT YET ENROLLED IN ANY COURSE



LARANA, INC.

```
SELECT S.SNAME, S.EMAIL  
FROM STUDENT S  
LEFT JOIN ENROLLMENT E ON S.SID = E.SID  
WHERE E.SID IS NULL;
```

SNAME	EMAIL
Diana Evans	diana.evans@example.com



## 8. UPDATE THE GRADE OF A SPECIFIC STUDENT IN A COURSE



LARANA, INC.

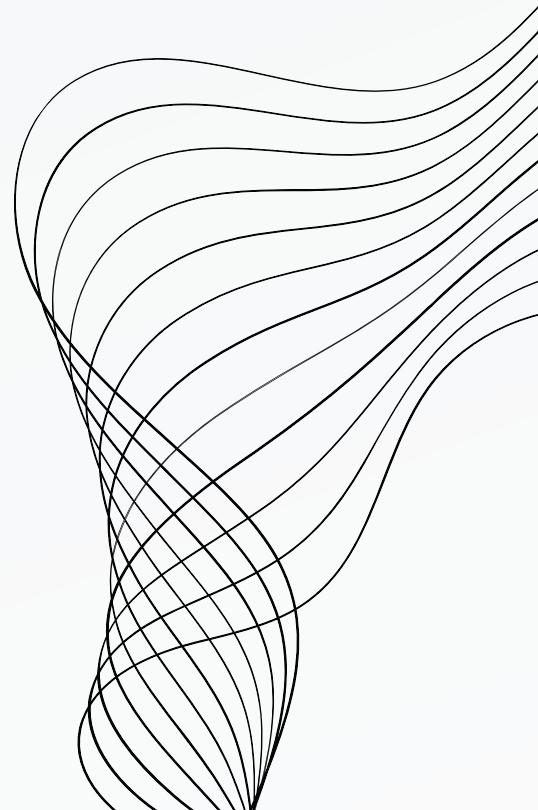
**UPDATE ENROLLMENT**

**SET GRADE = 'B'**

**WHERE SID = 1**

**AND CID = 101;**

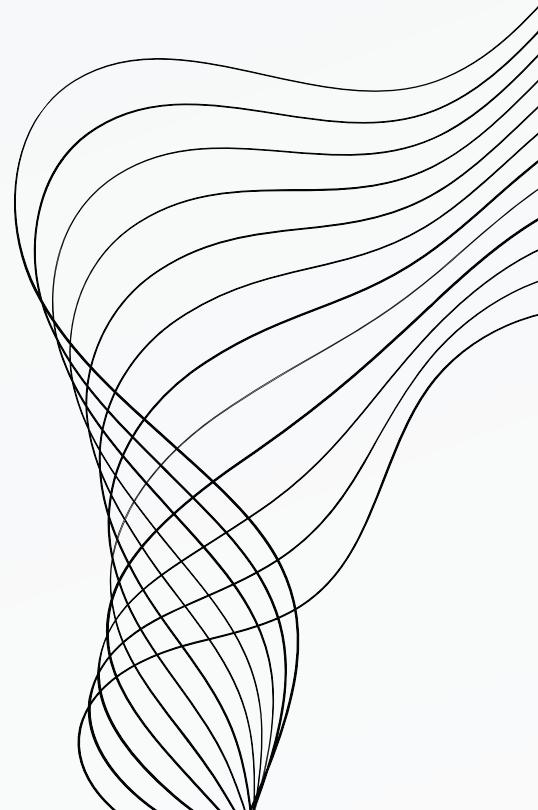
1 row(s) updated.



## 9. CALCULATE THE AVERAGE GRADE FOR EACH COURSE

```
SELECT C.CNAME, AVG(CASE  
    WHEN GRADE = 'A' THEN 4  
    WHEN GRADE = 'B' THEN 3  
    WHEN GRADE = 'C' THEN 2  
    WHEN GRADE = 'D' THEN 1  
    ELSE 0  
END) AS AVG_GRADE  
FROM COURSE C  
JOIN ENROLLMENT E ON C.CID = E.CID  
GROUP BY C.CNAME;
```

Calculus	4
History	3
Artificial Intelligence	4
Data Structures	2
Physics	3.5
Computer Science	4



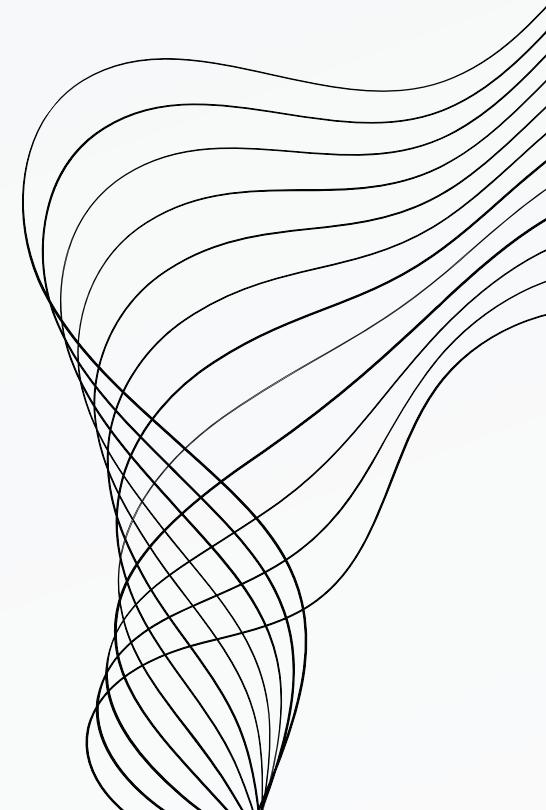
## 10. FIND THE NUMBER OF STUDENTS IN EACH DEPARTMENT



LARANA, INC.

```
SELECT D.DNAME, COUNT(DISTINCT S.SID) AS STUDENT_COUNT  
      FROM DEPARTMENT D  
  JOIN COURSE C ON D.DEPT_ID = C.DEPT_ID  
  JOIN ENROLLMENT E ON C.CID = E.CID  
  JOIN STUDENT S ON E.SID = S.SID  
 GROUP BY D.DNAME  
 ORDER BY STUDENT_COUNT DESC;
```

DNAME	STUDENT_COUNT
Engineering	4
Science	2
Arts	1
Mathematics	1

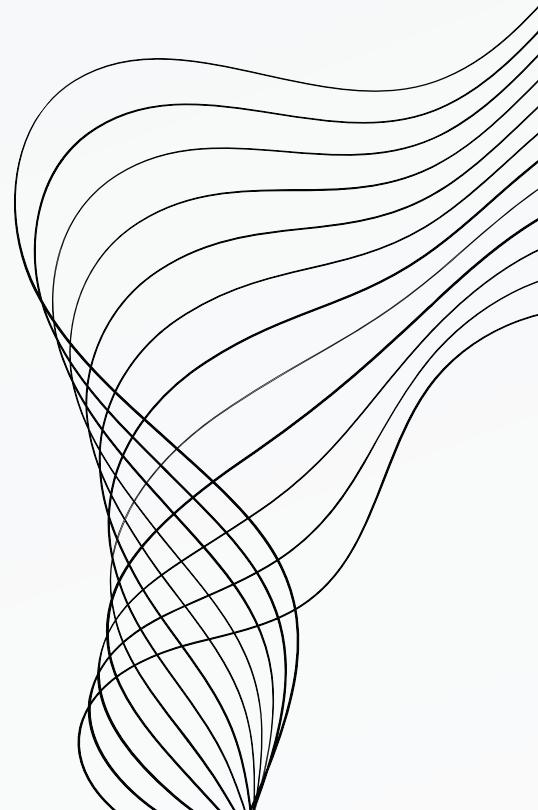


## 11. RETRIEVE STUDENTS ENROLLED IN MORE THAN 3 COURSES



LARANA, INC.

```
SELECT S.SNAME, COUNT(E.CID) AS COURSE_COUNT  
      FROM STUDENT S  
  JOIN ENROLLMENT E ON S.SID = E.SID  
  GROUP BY S.SNAME  
 HAVING COUNT(E.CID) > 3;
```



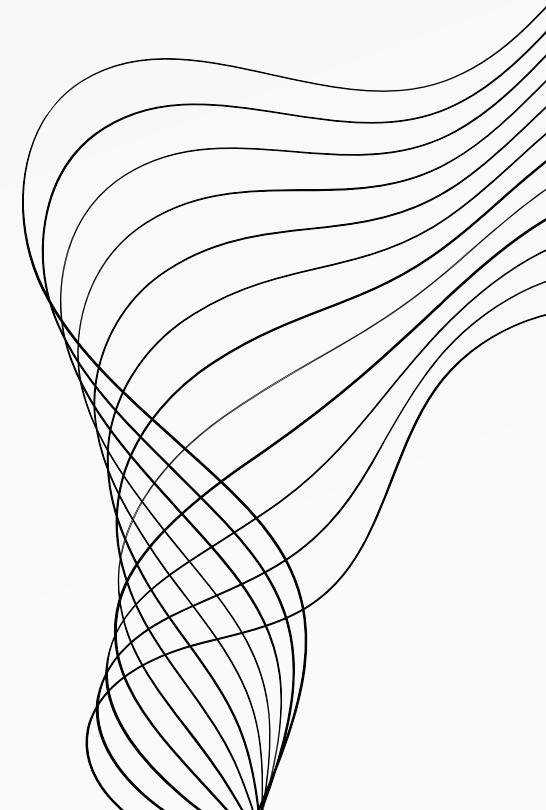
## 12. FIND ALL STUDENTS ENROLLED IN COURSES FROM TWO OR MORE DIFFERENT DEPARTMENTS



LARANA, INC.

```
SELECT S.SNAME, COUNT(DISTINCT D.DEPT_ID) AS  
      DEPARTMENT_COUNT  
  FROM STUDENT S  
 JOIN ENROLLMENT E ON S.SID = E.SID  
 JOIN COURSE C ON E.CID = C.CID  
 JOIN DEPARTMENT D ON C.DEPT_ID = D.DEPT_ID  
 GROUP BY S.SNAME  
 HAVING COUNT(DISTINCT D.DEPT_ID) >= 2;
```

SNAME	DEPARTMENT_COUNT
Alice	3



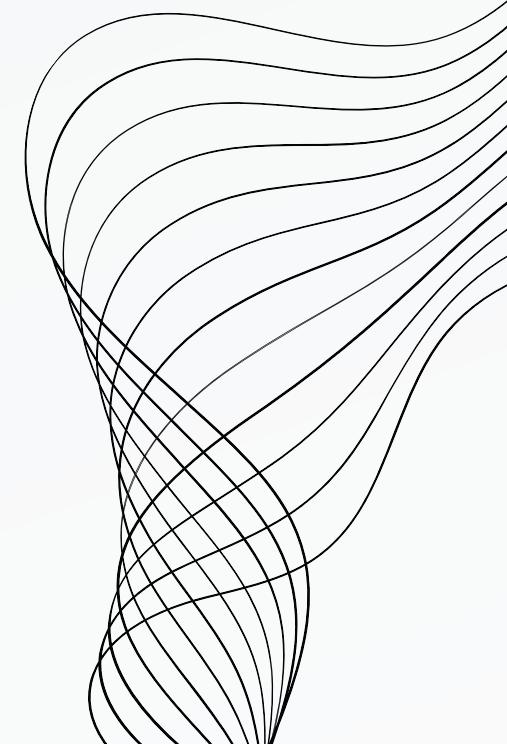
### 13. LIST ALL STUDENTS WHO HAVE TAKEN ALL THE COURSES OFFERED BY A SPECIFIC DEPARTMENT (E.G., 'SCIENCE')



LARANA, INC.

```
SELECT S.SNAME
  FROM STUDENT S
 JOIN ENROLLMENT E ON S.SID = E.SID
   JOIN COURSE C ON E.CID = C.CID
 JOIN DEPARTMENT D ON C.DEPT_ID = D.DEPT_ID
 WHERE D.DNAME = 'SCIENCE'
 GROUP BY S.SNAME
 HAVING COUNT(C.CID) = (SELECT COUNT(*)
                           FROM COURSE
                          WHERE DEPT_ID = (SELECT DEPT_ID
                                            FROM DEPARTMENT
                                           WHERE DNAME = 'SCIENCE'));
```

SNAME
Bob
Alice

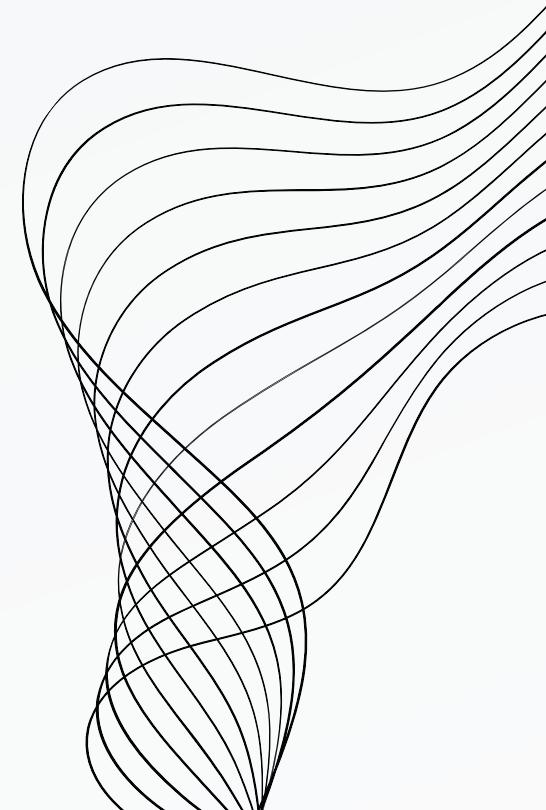


#### 14. FIND STUDENTS WHO HAVE FAILED AT LEAST ONE COURSE (ASSUME GRADE 'F' AS FAILURE)



LARANA, INC.

```
SELECT DISTINCT S.SNAME  
FROM STUDENT S  
JOIN ENROLLMENT E ON S.SID = E.SID  
WHERE E.GRADE = 'F';
```



## 15. CALCULATE THE AVERAGE GRADE OF EACH STUDENT ACROSS ALL THEIR COURSES



LARANA, INC.

```
SELECT S.SNAME, AVG(CASE
    WHEN E.GRADE = 'A' THEN 4
    WHEN E.GRADE = 'B' THEN 3
    WHEN E.GRADE = 'C' THEN 2
    WHEN E.GRADE = 'D' THEN 1
    ELSE 0
END) AS AVG_GRADE
FROM STUDENT S
JOIN ENROLLMENT E ON S.SID = E.SID
GROUP BY S.SNAME;
```

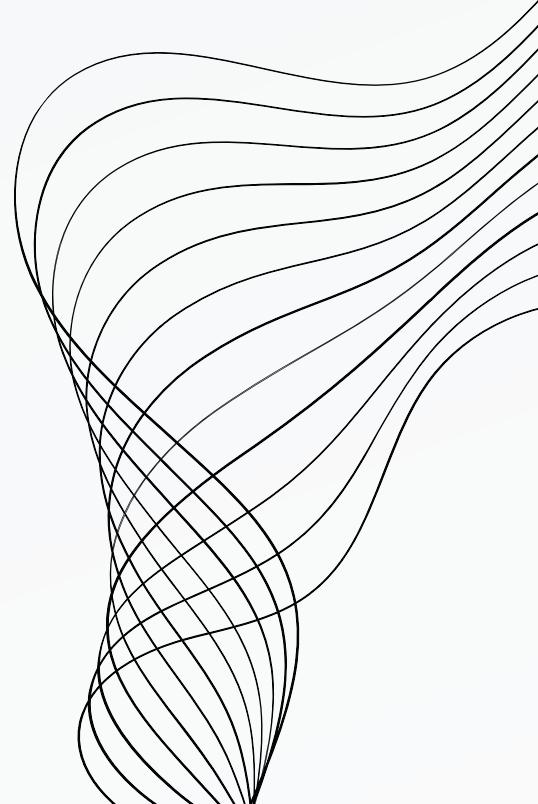
## 16. RETRIEVE COURSES WITH NO ENROLLMENTS (UNUSED COURSES)



LARANA, INC.

```
SELECT C.CNAME  
FROM COURSE C  
LEFT JOIN ENROLLMENT E ON C.CID = E.CID  
WHERE E.CID IS NULL;
```

CNAME
Literature
Business Management



## 17. IDENTIFY STUDENTS ENROLLED IN THE SAME COURSES AS A SPECIFIC STUDENT (E.G., 'ALICE BROWN')



LARANA, INC.

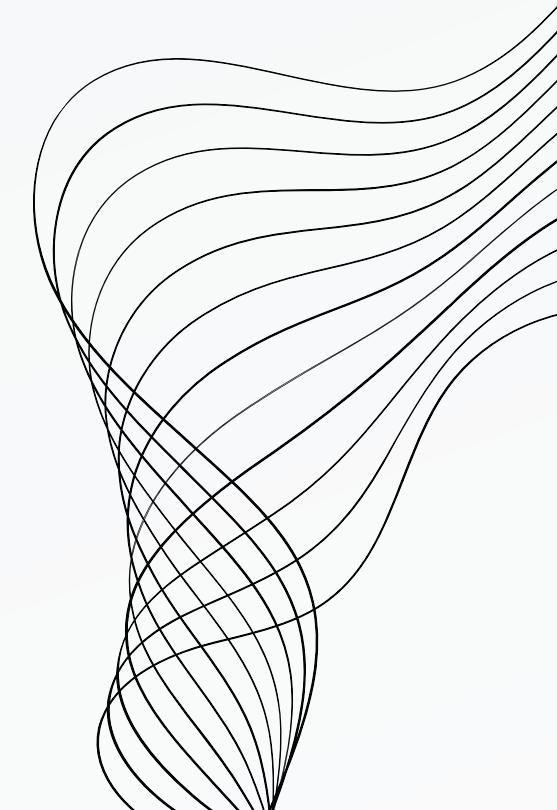
```
SELECT DISTINCT S2.SNAME  
FROM STUDENT S1  
JOIN ENROLLMENT E1 ON S1.SID = E1.SID  
JOIN ENROLLMENT E2 ON E1.CID = E2.CID  
JOIN STUDENT S2 ON E2.SID = S2.SID  
WHERE S1.SNAME = 'ALICE BROWN'  
AND S2.SNAME != 'ALICE BROWN';
```

## 18. LIST STUDENTS WITH THE SAME COURSE ENROLLMENT PATTERN (ENROLLED IN THE SAME SET OF COURSES)

```
SELECT S1.SNAME AS STUDENT_1, S2.SNAME AS STUDENT_2
      FROM ENROLLMENT E1
      JOIN ENROLLMENT E2 ON E1.CID = E2.CID
      JOIN STUDENT S1 ON E1.SID = S1.SID
      JOIN STUDENT S2 ON E2.SID = S2.SID
      GROUP BY S1.SNAME, S2.SNAME
HAVING COUNT(DISTINCT E1.CID) = (SELECT COUNT(DISTINCT
                                         CID)
                                         FROM ENROLLMENT
                                         WHERE SID = S1.SID)
      AND S1.SNAME != S2.SNAME;
```



LARANA, INC.



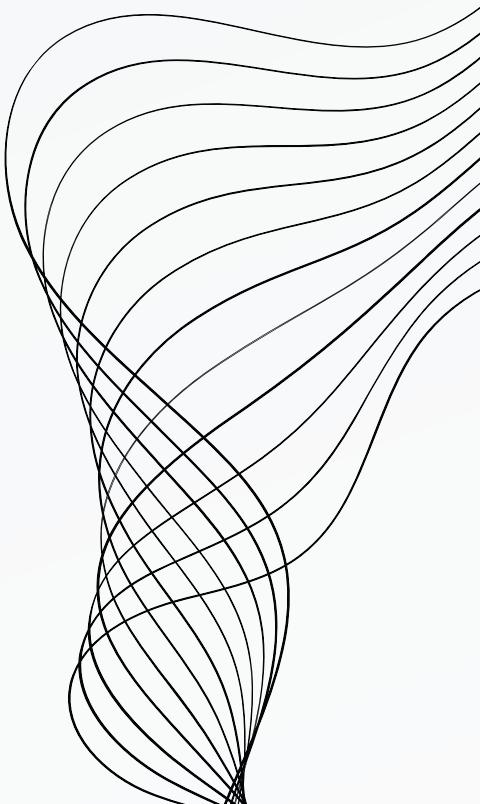
# 19. FIND THE HIGHEST AND LOWEST GRADE RECEIVED BY STUDENTS IN EACH DEPARTMENT



LARANA, INC.

```
SELECT D.DNAME, MAX(E.GRADE) AS HIGHEST_GRADE, MIN(E.GRADE) AS LOWEST_GRADE  
      FROM DEPARTMENT D  
  JOIN COURSE C ON D.DEPT_ID = C.DEPT_ID  
  JOIN ENROLLMENT E ON C.CID = E.CID  
 GROUP BY D.DNAME;
```

DNAME	HIGHEST_GRADE	LOWEST_GRADE
Mathematics	A	A
Engineering	C	A
Arts	B	B
Science	B	A



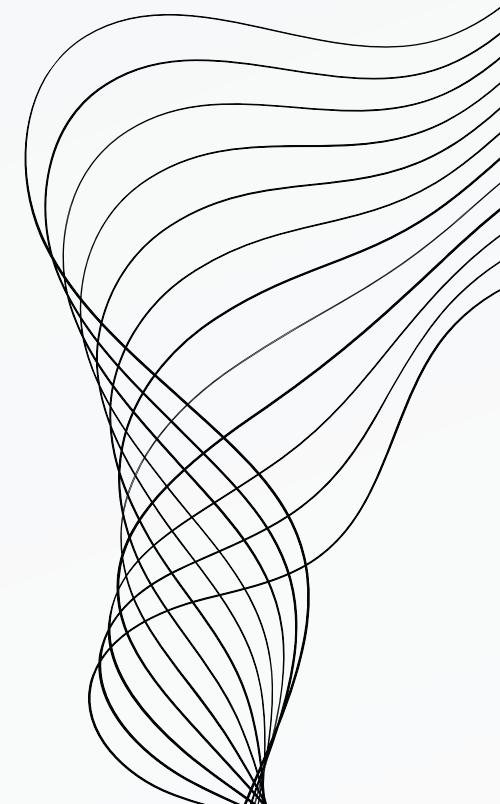
## --20. RETRIEVE THE LIST OF COURSES THAT OFFER MORE CREDITS THAN THE AVERAGE COURSE



LARANA, INC.

```
SELECT C.CNAME, C.CREDITS  
      FROM COURSE C  
 WHERE C.CREDITS > (SELECT AVG(CREDITS) FROM COURSE);
```

CNAME	CREDITS
Computer Science	5
Data Structures	5
Artificial Intelligence	5



## 18. LIST STUDENTS WITH THE SAME COURSE ENROLLMENT PATTERN (ENROLLED IN THE SAME SET OF COURSES)

```
SELECT S1.SNAME AS STUDENT_1, S2.SNAME AS STUDENT_2
      FROM ENROLLMENT E1
      JOIN ENROLLMENT E2 ON E1.CID = E2.CID
      JOIN STUDENT S1 ON E1.SID = S1.SID
      JOIN STUDENT S2 ON E2.SID = S2.SID
      GROUP BY S1.SNAME, S2.SNAME
HAVING COUNT(DISTINCT E1.CID) = (SELECT COUNT(DISTINCT
                                         CID)
                                         FROM ENROLLMENT
                                         WHERE SID = S1.SID)
      AND S1.SNAME != S2.SNAME;
```



LARANA, INC.

