Question 1

- Write and execute a SQL query to list the school names, community names and average attendance for communities with a hardship index of 98.

```sql
SELECT CPS.NAME_OF_SCHOOL,CPS.COMMUNITY_AREA_NAME,CPS.AVERAGE_STUDENT_ATTENDANCE, CSE.HARDSHIP_INDEX
FROM chicago_public_schools CPS LEFT OUTER JOIN
chicago_socioeconomic_data CSE
ON CPS.COMMUNITY_AREA_NUMBER = CSE.COMMUNITY_AREA_NUMBER
WHERE CSE.HARDSHIP_INDEX = 98;
```

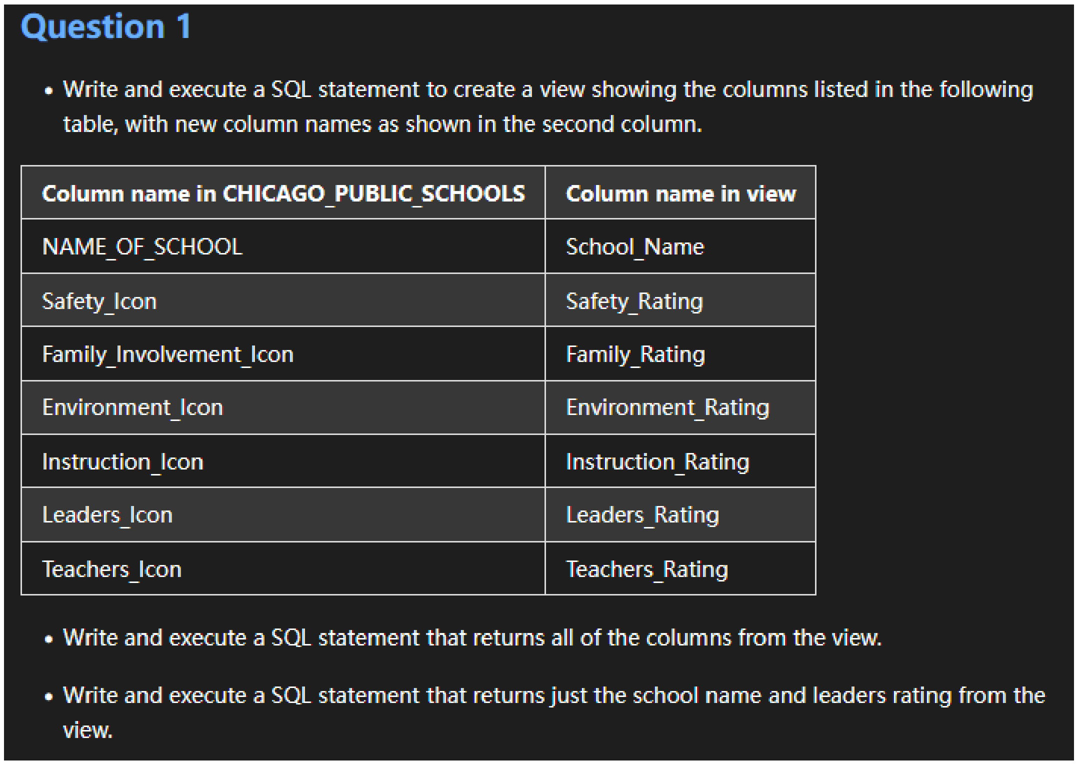
NAME_OF_SCHOOL	COMMUNITY_AREA_NAME	AVERAGE_STUDENT_ATTENDANCE	HARDSHIP_INDEX
George Washington Carver Military Academy High Sch...	RIVERDALE	91.60%	98
George Washington Carver Primary School	RIVERDALE	90.90%	98
Ira F Aldridge Elementary School	RIVERDALE	92.90%	98
William E B Dubois Elementary School	RIVERDALE	93.30%	98

Question 2

- Write and execute a SQL query to list all crimes that took place at a school. Include case number, crime type and community name.

```sql
SELECT
CC.CASE_NUMBER,CC.PRIMARY_TYPE,CSE.COMMUNITY_AREA_NAME,CC.LOCATION_DESCRIPTION
FROM chicago_crime CC LEFT JOIN chicago_socioeconomic_data CSE
ON CC.COMMUNITY_AREA_NUMBER = CSE.COMMUNITY_AREA_NUMBER
WHERE CC.LOCATION_DESCRIPTION LIKE '%SCHOOL%';
```

CASE_NUMBER	PRIMARY_TYPE	COMMUNITY_AREA_NAME	LOCATION_DESCRIPTION
HL353697	BATTERY	South Shore	SCHOOL, PUBLIC, GROUNDS
HL725506	BATTERY	Lincoln Square	SCHOOL, PUBLIC, BUILDING
HP716225	BATTERY	Douglas	SCHOOL, PUBLIC, BUILDING
HH639427	BATTERY	Austin	SCHOOL, PUBLIC, BUILDING
JA460432	BATTERY	Ashburn	SCHOOL, PUBLIC, GROUNDS
HS200939	CRIMINAL DAMAGE	Austin	SCHOOL, PUBLIC, GROUNDS
HK577020	NARCOTICS	Rogers Park	SCHOOL, PUBLIC, GROUNDS
HS305355	NARCOTICS	Brighton Park	SCHOOL, PUBLIC, BUILDING
HT315369	ASSAULT	East Garfield Park	SCHOOL, PUBLIC, GROUNDS
HR585012	CRIMINAL TRESPASS	Ashburn	SCHOOL, PUBLIC, GROUNDS
HH292682	PUBLIC PEACE VIOLATION	CHICAGO	SCHOOL, PRIVATE, BUILDING
G635735	PUBLIC PEACE VIOLATION	CHICAGO	SCHOOL, PUBLIC, BUILDING

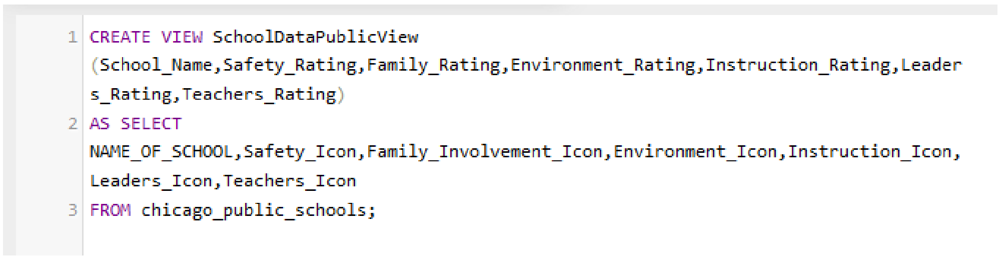


Question 1

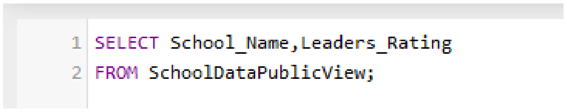
- Write and execute a SQL statement to create a view showing the columns listed in the following table, with new column names as shown in the second column.

Column name in CHICAGO_PUBLIC_SCHOOLS	Column name in view
NAME_OF_SCHOOL	School_Name
Safety_Icon	Safety_Rating
Family_Involvement_Icon	Family_Rating
Environment_Icon	Environment_Rating
Instruction_Icon	Instruction_Rating
Leaders_Icon	Leaders_Rating
Teachers_Icon	Teachers_Rating

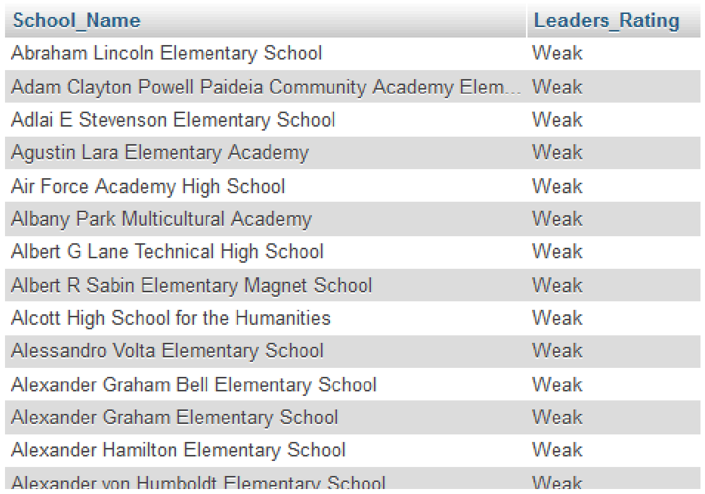
- Write and execute a SQL statement that returns all of the columns from the view.
- Write and execute a SQL statement that returns just the school name and leaders rating from the view.



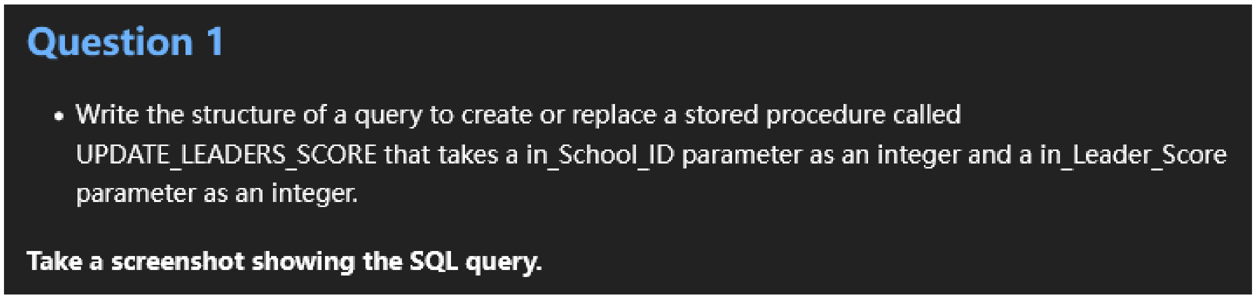
```
CREATE VIEW SchoolDataPublicView
(School_Name,Safety_Rating,Family_Rating,Environment_Rating,Instruction_Rating,Leader
s_Rating,Teachers_Rating)
AS SELECT
NAME_OF_SCHOOL,Safety_Icon,Family_Involvement_Icon,Environment_Icon,Instruction_Icon,
Leaders_Icon,Teachers_Icon
FROM chicago_public_schools;
```



```
SELECT School_Name,Leaders_Rating
FROM SchoolDataPublicView;
```



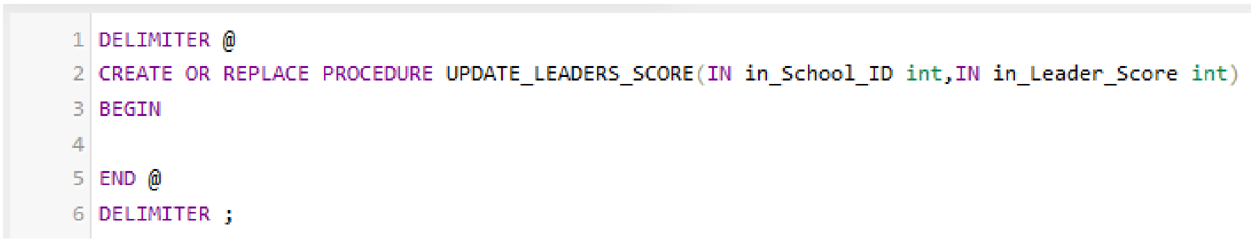
School_Name	Leaders_Rating
Abraham Lincoln Elementary School	Weak
Adam Clayton Powell Paideia Community Academy Elem...	Weak
Adlai E Stevenson Elementary School	Weak
Agustin Lara Elementary Academy	Weak
Air Force Academy High School	Weak
Albany Park Multicultural Academy	Weak
Albert G Lane Technical High School	Weak
Albert R Sabin Elementary Magnet School	Weak
Alcott High School for the Humanities	Weak
Alessandro Volta Elementary School	Weak
Alexander Graham Bell Elementary School	Weak
Alexander Graham Elementary School	Weak
Alexander Hamilton Elementary School	Weak
Alexander von Humboldt Elementary School	Weak



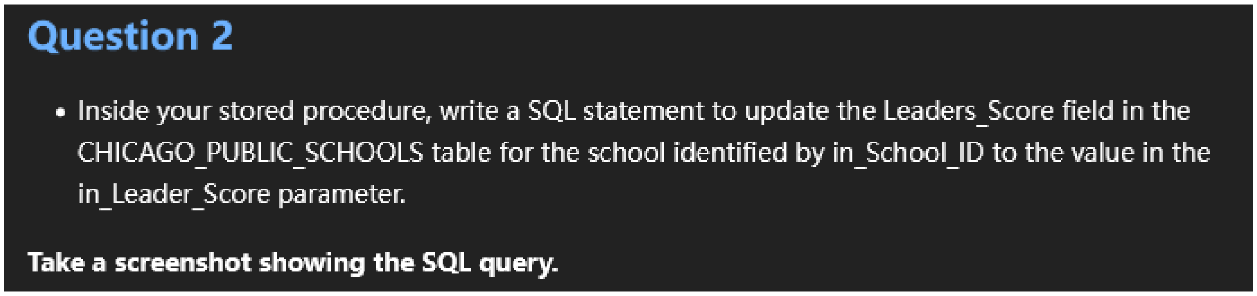
Question 1

- Write the structure of a query to create or replace a stored procedure called UPDATE_LEADERS_SCORE that takes a in_School_ID parameter as an integer and a in_Leader_Score parameter as an integer.

Take a screenshot showing the SQL query.



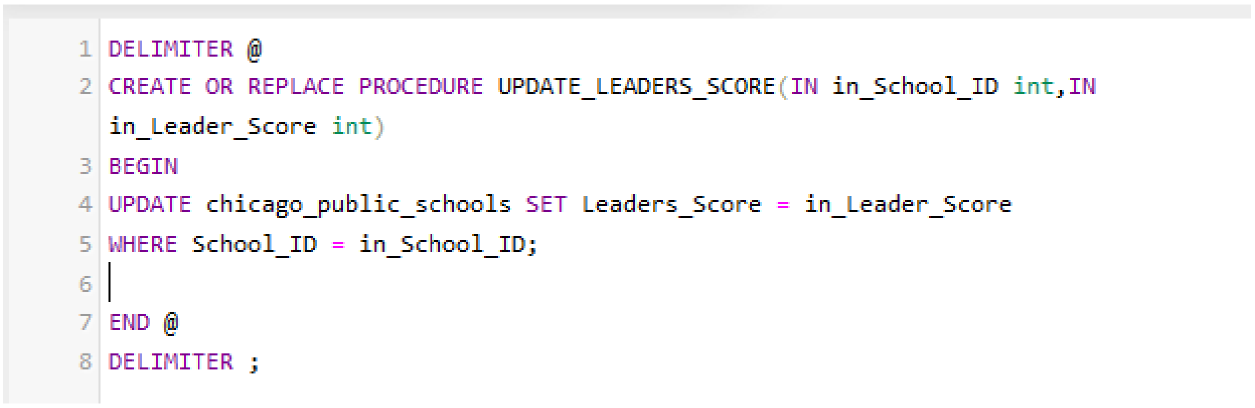
```
DELIMITER @
CREATE OR REPLACE PROCEDURE UPDATE_LEADERS_SCORE(IN in_School_ID int,IN in_Leader_Score int)
BEGIN

END @
DELIMITER ;
```



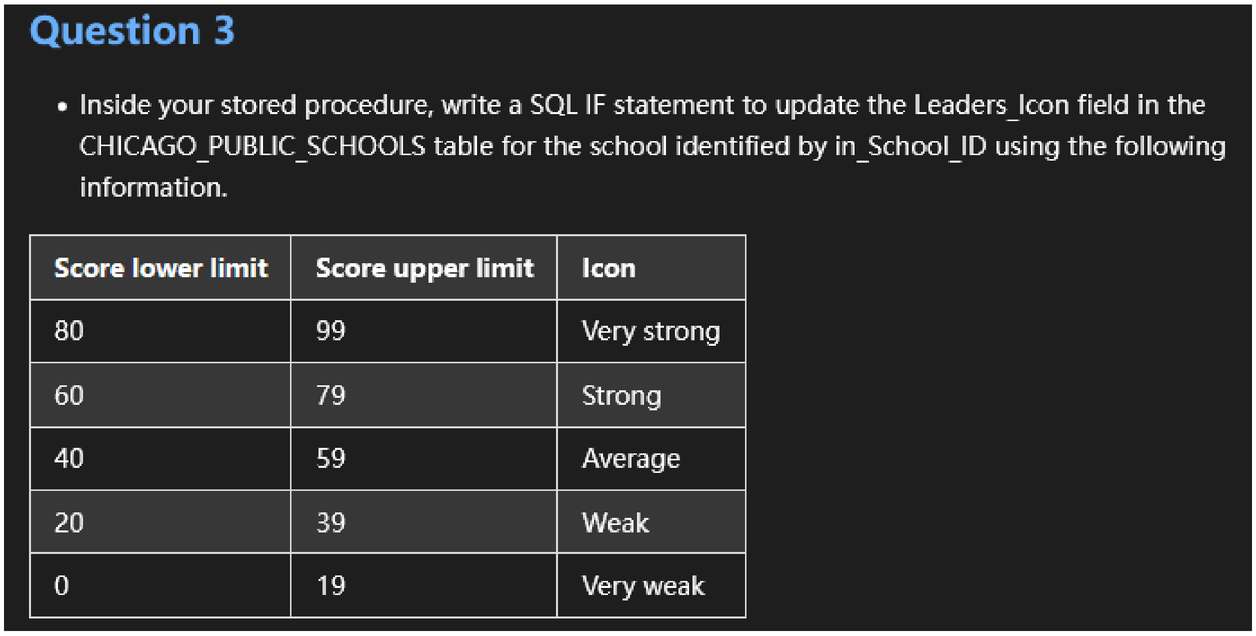
Question 2

- Inside your stored procedure, write a SQL statement to update the Leaders_Score field in the CHICAGO_PUBLIC_SCHOOLS table for the school identified by in_School_ID to the value in the in_Leader_Score parameter.

Take a screenshot showing the SQL query.



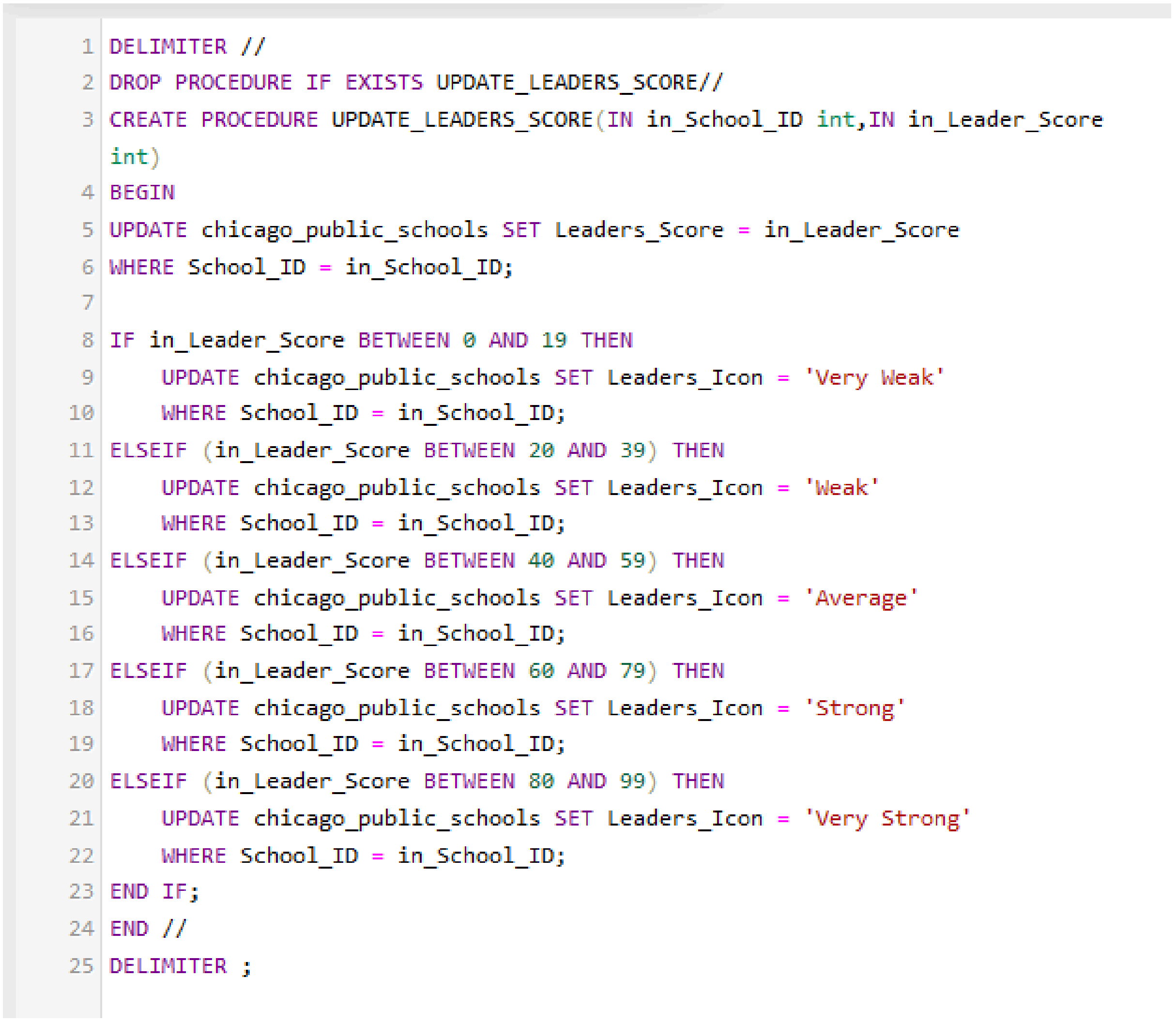
```
DELIMITER @
CREATE OR REPLACE PROCEDURE UPDATE_LEADERS_SCORE(IN in_School_ID int,IN in_Leader_Score int)
BEGIN
UPDATE chicago_public_schools SET Leaders_Score = in_Leader_Score
WHERE School_ID = in_School_ID;

END @
DELIMITER ;
```



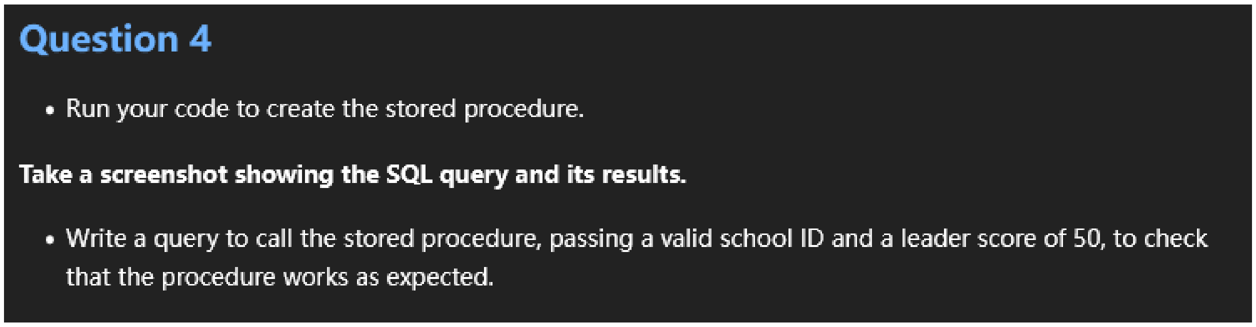
Question 3

- Inside your stored procedure, write a SQL IF statement to update the Leaders_Icon field in the CHICAGO_PUBLIC_SCHOOLS table for the school identified by in_School_ID using the following information.

Score lower limit	Score upper limit	Icon
80	99	Very strong
60	79	Strong
40	59	Average
20	39	Weak
0	19	Very weak



```sql
DELIMITER //
DROP PROCEDURE IF EXISTS UPDATE_LEADERS_SCORE//
CREATE PROCEDURE UPDATE_LEADERS_SCORE(IN in_School_ID int,IN in_Leader_Score int)
BEGIN
UPDATE chicago_public_schools SET Leaders_Score = in_Leader_Score
WHERE School_ID = in_School_ID;

IF in_Leader_Score BETWEEN 0 AND 19 THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Very Weak'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 20 AND 39) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Weak'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 40 AND 59) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Average'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 60 AND 79) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Strong'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 80 AND 99) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Very Strong'
    WHERE School_ID = in_School_ID;
END IF;
END //
DELIMITER ;
```

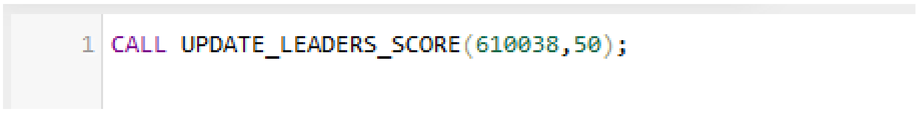


Question 4

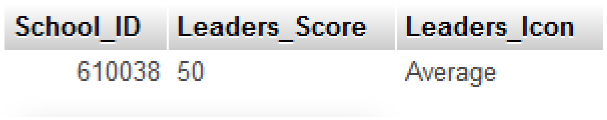
- Run your code to create the stored procedure.

Take a screenshot showing the SQL query and its results.

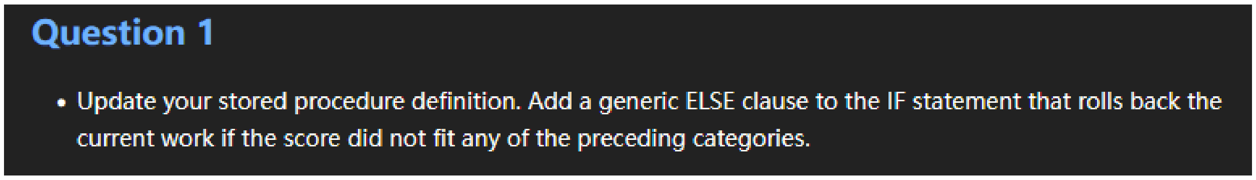
- Write a query to call the stored procedure, passing a valid school ID and a leader score of 50, to check that the procedure works as expected.



```sql
CALL UPDATE_LEADERS_SCORE(610038,50);
```

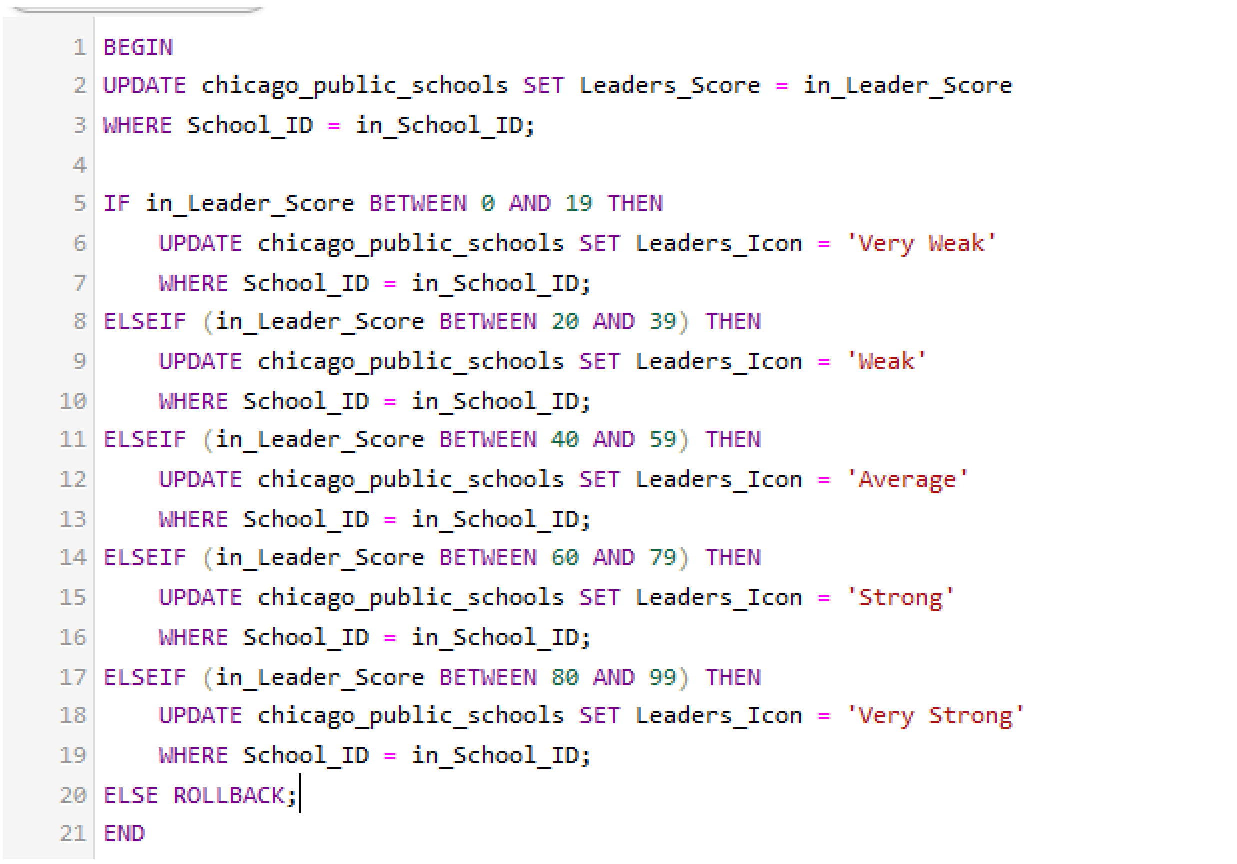


School_ID	Leaders_Score	Leaders_Icon
610038	50	Average



Question 1

- Update your stored procedure definition. Add a generic ELSE clause to the IF statement that rolls back the current work if the score did not fit any of the preceding categories.



```sql
BEGIN
UPDATE chicago_public_schools SET Leaders_Score = in_Leader_Score
WHERE School_ID = in_School_ID;

IF in_Leader_Score BETWEEN 0 AND 19 THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Very Weak'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 20 AND 39) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Weak'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 40 AND 59) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Average'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 60 AND 79) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Strong'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 80 AND 99) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Very Strong'
    WHERE School_ID = in_School_ID;
ELSE ROLLBACK;
END
```

Question 2

- Update your stored procedure definition again. Add a statement to commit the current unit of work at the end of the procedure.

```
BEGIN
UPDATE chicago_public_schools SET Leaders_Score = in_Leader_Score
WHERE School_ID = in_School_ID;

IF in_Leader_Score BETWEEN 0 AND 19 THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Very Weak'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 20 AND 39) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Weak'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 40 AND 59) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Average'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 60 AND 79) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Strong'
    WHERE School_ID = in_School_ID;
ELSEIF (in_Leader_Score BETWEEN 80 AND 99) THEN
    UPDATE chicago_public_schools SET Leaders_Icon = 'Very Strong'
    WHERE School_ID = in_School_ID;
ELSE ROLLBACK;
END IF;
COMMIT;
END
```

- Write and run one query to check that the updated stored procedure works as expected when you use a valid score of 38.
- Write and run another query to check that the updated stored procedure works as expected when you use an invalid score of 101.

```
CALL UPDATE_LEADERS_SCORE(610038,38);
```

School_ID	Leaders_Score	Leaders_Icon
610038	38	Weak