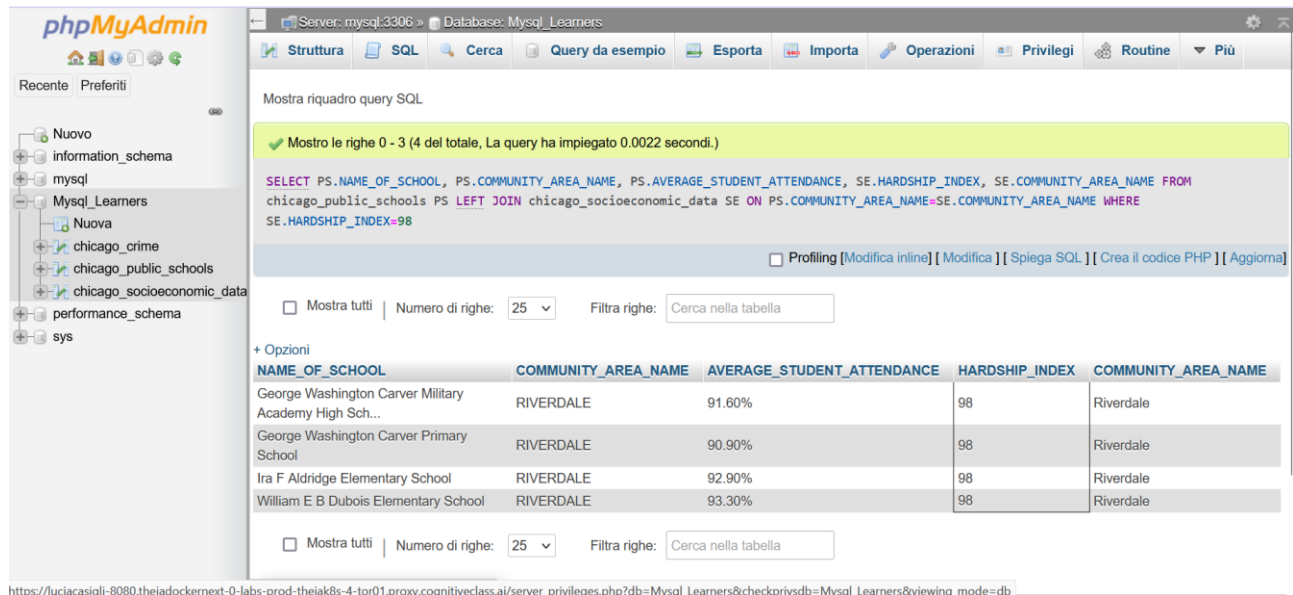


Exercise 1: Using Joins

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.



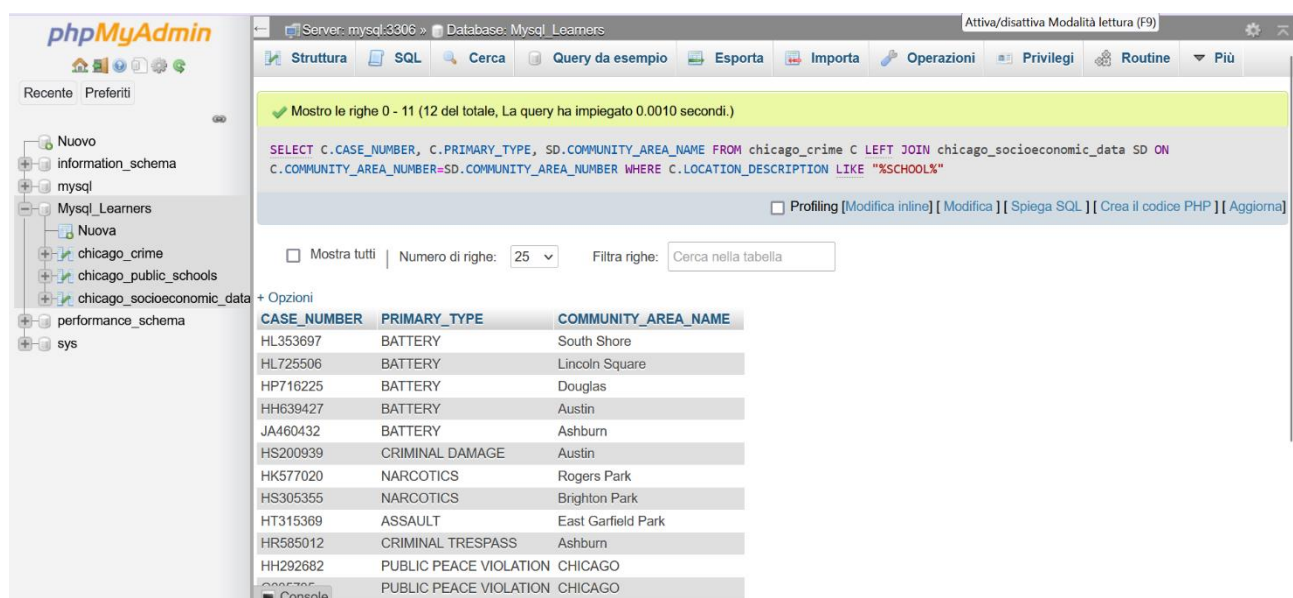
The screenshot shows the phpMyAdmin interface with the MySQL database 'Mysql_Learners' selected. A SQL query is executed, displaying the results in a table. The query filters for schools in the 'RIVERDALE' community area with a hardship index of 98.

```
SELECT PS.NAME_OF_SCHOOL, PS.COMMUNITY_AREA_NAME, PS.AVERAGE_STUDENT_ATTENDANCE, SE.HARDSHIP_INDEX, SE.COMMUNITY_AREA_NAME FROM
chicago_public_schools PS LEFT JOIN chicago_socioeconomic_data SE ON PS.COMMUNITY_AREA_NAME=SE.COMMUNITY_AREA_NAME WHERE
SE.HARDSHIP_INDEX=98
```

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

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.



The screenshot shows the phpMyAdmin interface with the MySQL database 'Mysql_Learners' selected. A SQL query is executed, displaying the results in a table. The query filters for crimes that occurred at a school location.

```
SELECT C.CASE_NUMBER, C.PRIMARY_TYPE, SD.COMMUNITY_AREA_NAME FROM chicago_crime C LEFT JOIN chicago_socioeconomic_data SD ON
C.COMMUNITY_AREA_NUMBER=SD.COMMUNITY_AREA_NUMBER WHERE C.LOCATION_DESCRIPTION LIKE "%SCHOOL%"
```

CASE_NUMBER	PRIMARY_TYPE	COMMUNITY_AREA_NAME
HL353697	BATTERY	South Shore
HL725506	BATTERY	Lincoln Square
HP716225	BATTERY	Douglas
HH639427	BATTERY	Austin
JA460432	BATTERY	Ashburn
HS200939	CRIMINAL DAMAGE	Austin
HK577020	NARCOTICS	Rogers Park
HS305355	NARCOTICS	Brighton Park
HT315369	ASSAULT	East Garfield Park
HR585012	CRIMINAL TRESPASS	Ashburn
HH292682	PUBLIC PEACE VIOLATION	CHICAGO
CH292682	PUBLIC PEACE VIOLATION	CHICAGO

Exercise 2: Creating a View

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 just the school name and leaders rating from the view.

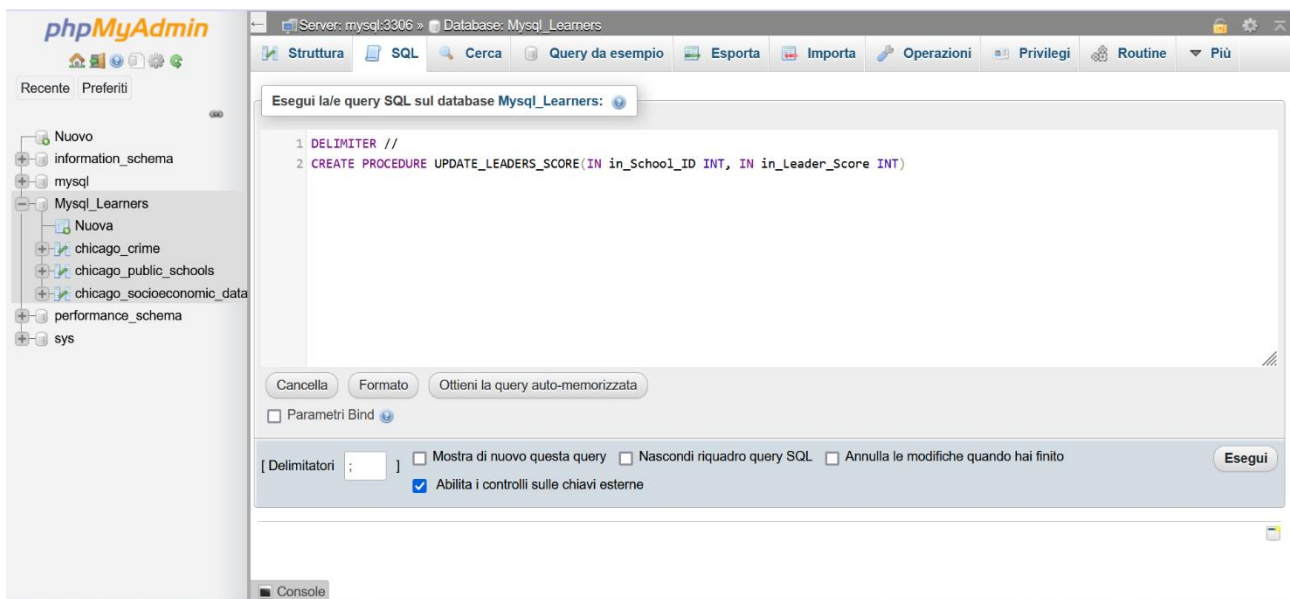
The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible, including the 'Mysql_Learners' database and the 'PRIVATE' view. The main panel displays a SQL query: `select School_Name,Leaders_Rating from PRIVATE`. Below the query, the results are shown in a table with two columns: 'School_Name' and 'Leaders_Rating'. The table contains 25 rows of data, all showing a 'Weak' rating for the 'Leaders_Rating' column. The interface also shows a status bar indicating that the query was executed successfully and took 0.0004 seconds.

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
on Humboldt Elementary School	Weak

Exercise 3: Creating a Stored Procedure

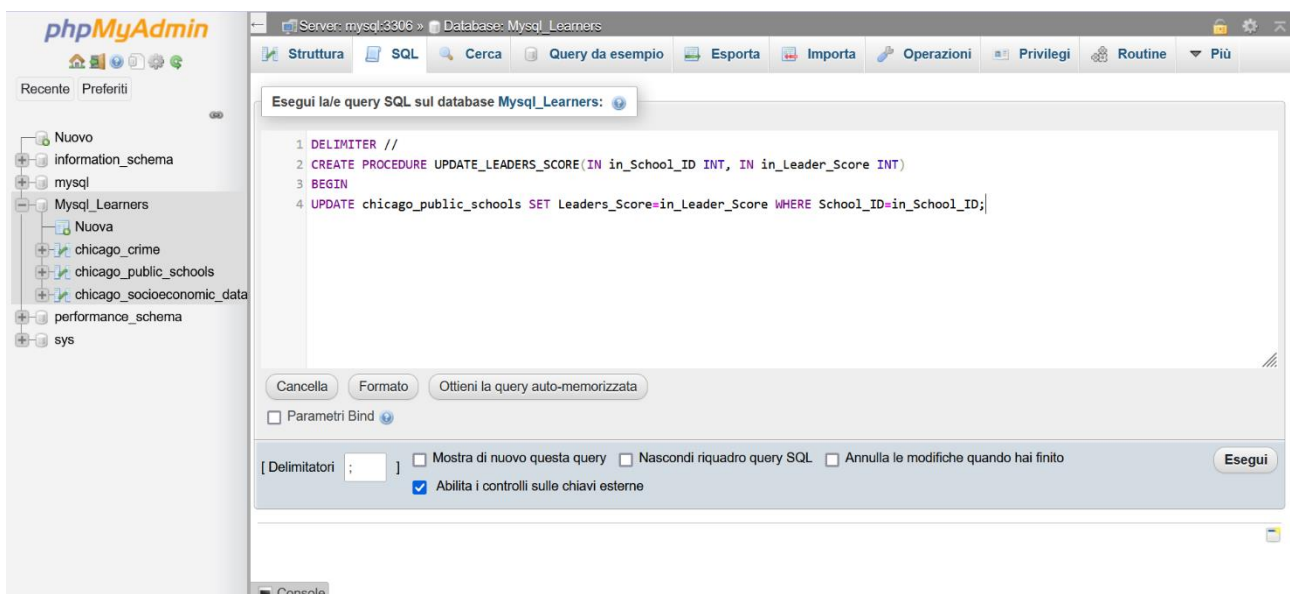
Question 1

- 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.



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.



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

The screenshot shows the phpMyAdmin interface with the 'Mysql_Learners' database selected. The SQL editor contains the following code:

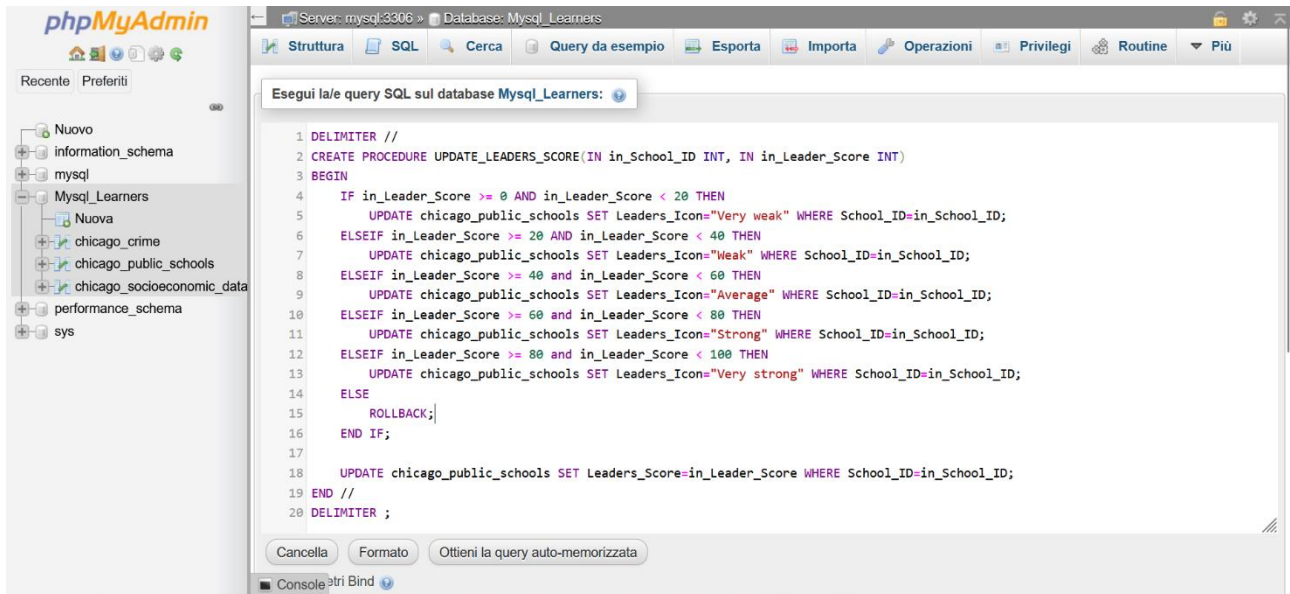
```
1 DELIMITER //
2 CREATE PROCEDURE UPDATE_LEADERS_SCORE(IN in_School_ID INT, IN in_Leader_Score INT)
3 BEGIN
4     IF in_Leader_Score >= 0 AND in_Leader_Score < 20 THEN
5         UPDATE chicao_public_schools SET Leaders_Icon="Very weak" WHERE School_ID=in_School_ID;
6     ELSEIF in_Leader_Score >= 20 AND in_Leader_Score < 40 THEN
7         UPDATE chicao_public_schools SET Leaders_Icon="Weak" WHERE School_ID=in_School_ID;
8     ELSEIF in_Leader_Score >= 40 and in_Leader_Score < 60 THEN
9         UPDATE chicao_public_schools SET Leaders_Icon="Average" WHERE School_ID=in_School_ID;
10    ELSEIF in_Leader_Score >= 60 and in_Leader_Score < 80 THEN
11        UPDATE chicao_public_schools SET Leaders_Icon="Strong" WHERE School_ID=in_School_ID;
12    ELSEIF in_Leader_Score >= 80 and in_Leader_Score < 100 THEN
13        UPDATE chicao_public_schools SET Leaders_Icon="Very strong" WHERE School_ID=in_School_ID;
14    END IF;
15
16    UPDATE chicao_public_schools SET Leaders_Score=in_Leader_Score WHERE School_ID=in_School_ID;
17 END //
18 DELIMITER ;
```

Below the code editor are buttons for 'Cancella', 'Formato', and 'Ottieni la query auto-memorizzata'. At the bottom, there are checkboxes for 'Parametri Bind', 'Mostra di nuovo questa query', 'Nascondi riquadro query SQL', and 'Annulla le modifiche quando hai finito', along with an 'Esegui' button.

Exercise 4: Using Transactions

Question 1

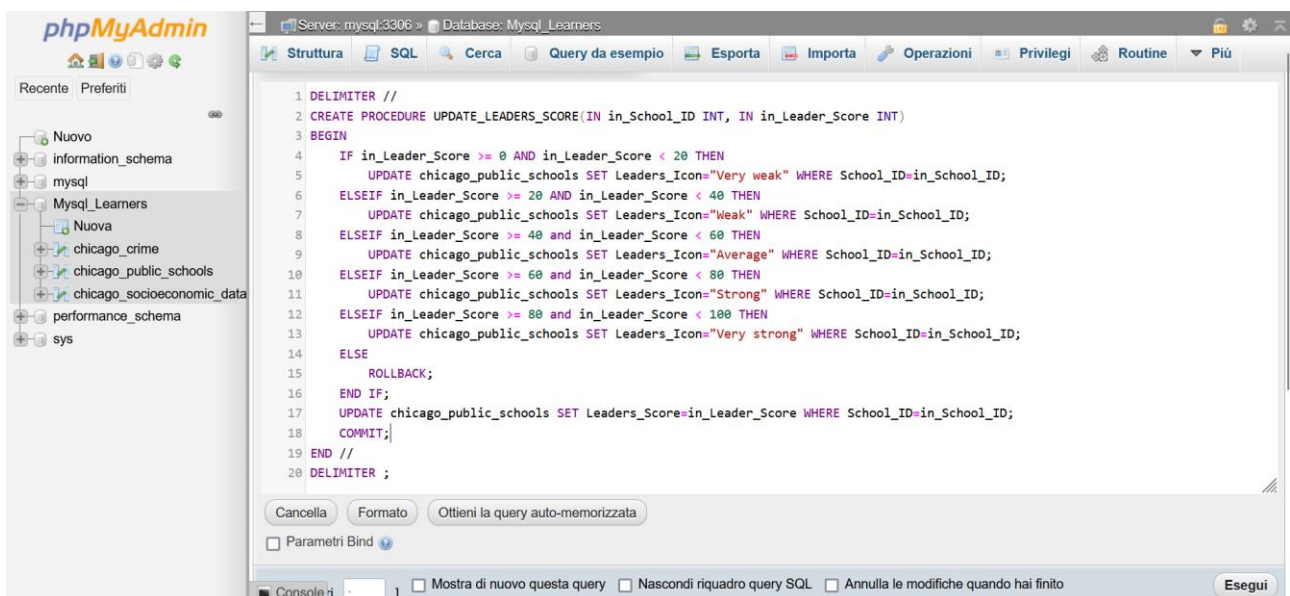
- Update your stored procedure definition. Add a generic ELSE clause to the IF statement that rolls back the current work if the score did not fit any of the preceding categories.



```
1 DELIMITER //  
2 CREATE PROCEDURE UPDATE_LEADERS_SCORE(IN in_School_ID INT, IN in_Leader_Score INT)  
3 BEGIN  
4     IF in_Leader_Score >= 0 AND in_Leader_Score < 20 THEN  
5         UPDATE chicago_public_schools SET Leaders_Icon="Very weak" WHERE School_ID=in_School_ID;  
6     ELSEIF in_Leader_Score >= 20 AND in_Leader_Score < 40 THEN  
7         UPDATE chicago_public_schools SET Leaders_Icon="Weak" WHERE School_ID=in_School_ID;  
8     ELSEIF in_Leader_Score >= 40 AND in_Leader_Score < 60 THEN  
9         UPDATE chicago_public_schools SET Leaders_Icon="Average" WHERE School_ID=in_School_ID;  
10    ELSEIF in_Leader_Score >= 60 AND in_Leader_Score < 80 THEN  
11        UPDATE chicago_public_schools SET Leaders_Icon="Strong" WHERE School_ID=in_School_ID;  
12    ELSEIF in_Leader_Score >= 80 AND in_Leader_Score < 100 THEN  
13        UPDATE chicago_public_schools SET Leaders_Icon="Very strong" WHERE School_ID=in_School_ID;  
14    ELSE  
15        ROLLBACK;  
16    END IF;  
17    UPDATE chicago_public_schools SET Leaders_Score=in_Leader_Score WHERE School_ID=in_School_ID;  
18 END //  
19 DELIMITER ;
```

Question 2

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



```
1 DELIMITER //  
2 CREATE PROCEDURE UPDATE_LEADERS_SCORE(IN in_School_ID INT, IN in_Leader_Score INT)  
3 BEGIN  
4     IF in_Leader_Score >= 0 AND in_Leader_Score < 20 THEN  
5         UPDATE chicago_public_schools SET Leaders_Icon="Very weak" WHERE School_ID=in_School_ID;  
6     ELSEIF in_Leader_Score >= 20 AND in_Leader_Score < 40 THEN  
7         UPDATE chicago_public_schools SET Leaders_Icon="Weak" WHERE School_ID=in_School_ID;  
8     ELSEIF in_Leader_Score >= 40 AND in_Leader_Score < 60 THEN  
9         UPDATE chicago_public_schools SET Leaders_Icon="Average" WHERE School_ID=in_School_ID;  
10    ELSEIF in_Leader_Score >= 60 AND in_Leader_Score < 80 THEN  
11        UPDATE chicago_public_schools SET Leaders_Icon="Strong" WHERE School_ID=in_School_ID;  
12    ELSEIF in_Leader_Score >= 80 AND in_Leader_Score < 100 THEN  
13        UPDATE chicago_public_schools SET Leaders_Icon="Very strong" WHERE School_ID=in_School_ID;  
14    ELSE  
15        ROLLBACK;  
16    END IF;  
17    UPDATE chicago_public_schools SET Leaders_Score=in_Leader_Score WHERE School_ID=in_School_ID;  
18    COMMIT;  
19 END //  
20 DELIMITER ;
```