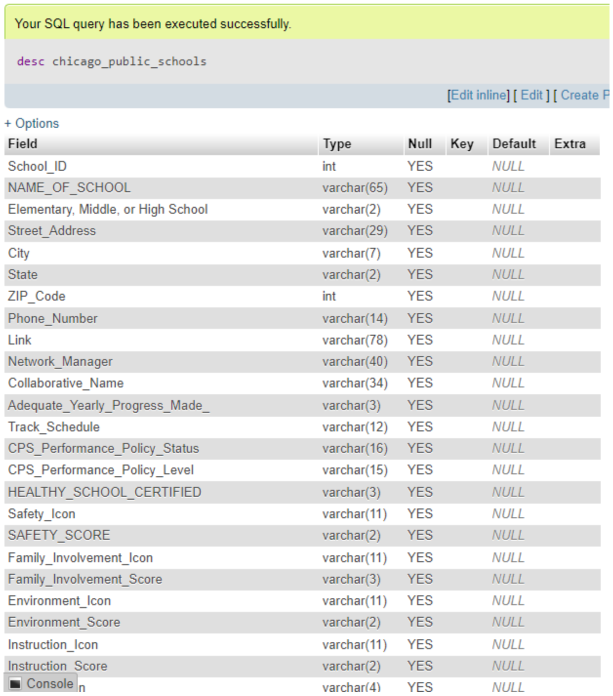
Final Project: Advanced SQL Techniques

Introduction

This project is the ‘Optional Honors’ part of the course [SQL: A Practical Introduction for Querying Databases](). Here is the source. I will be answering the questions asked with the help of **MySQL**.

1. Chicago Public School



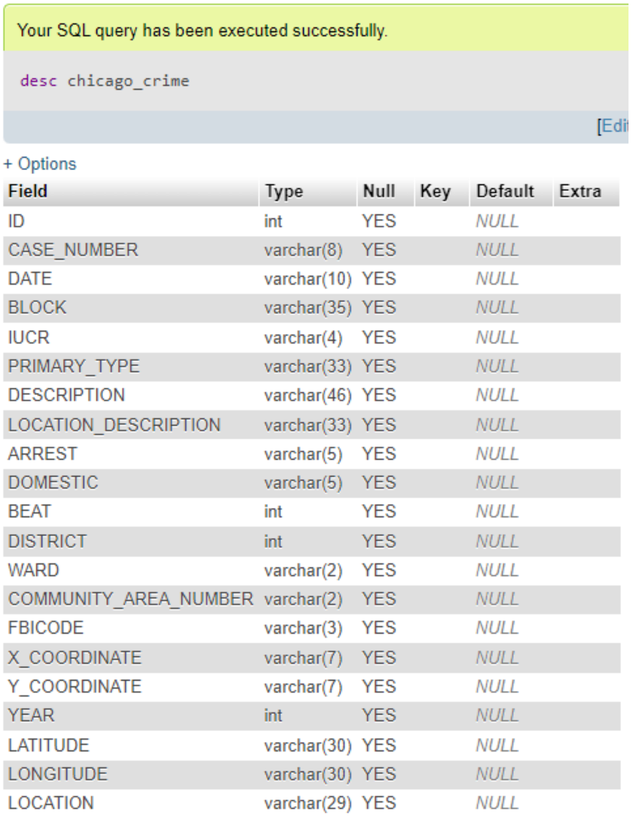
Your SQL query has been executed successfully.

```
desc chicago_public_schools
```

Field	Type	Null	Key	Default	Extra
School_ID	int	YES		*NULL*	
NAME_OF_SCHOOL	varchar(65)	YES		*NULL*	
Elementary, Middle, or High School	varchar(2)	YES		*NULL*	
Street_Address	varchar(29)	YES		*NULL*	
City	varchar(7)	YES		*NULL*	
State	varchar(2)	YES		*NULL*	
ZIP_Code	int	YES		*NULL*	
Phone_Number	varchar(14)	YES		*NULL*	
Link	varchar(78)	YES		*NULL*	
Network_Manager	varchar(40)	YES		*NULL*	
Collaborative_Name	varchar(34)	YES		*NULL*	
Adequate_Yearly_Progress_Made_	varchar(3)	YES		*NULL*	
Track_Schedule	varchar(12)	YES		*NULL*	
CPS_Performance_Policy_Status	varchar(16)	YES		*NULL*	
CPS_Performance_Policy_Level	varchar(15)	YES		*NULL*	
HEALTHY_SCHOOL_CERTIFIED	varchar(3)	YES		*NULL*	
Safety_Icon	varchar(11)	YES		*NULL*	
SAFETY_SCORE	varchar(2)	YES		*NULL*	
Family_Involvement_Icon	varchar(11)	YES		*NULL*	
Family_Involvement_Score	varchar(3)	YES		*NULL*	
Environment_Icon	varchar(11)	YES		*NULL*	
Environment_Score	varchar(2)	YES		*NULL*	
Instruction_Icon	varchar(11)	YES		*NULL*	
Instruction_Score	varchar(2)	YES		*NULL*	
n	varchar(4)	YES		*NULL*	

Final Project: Advanced SQL Techniques

2. Chicago Crime data

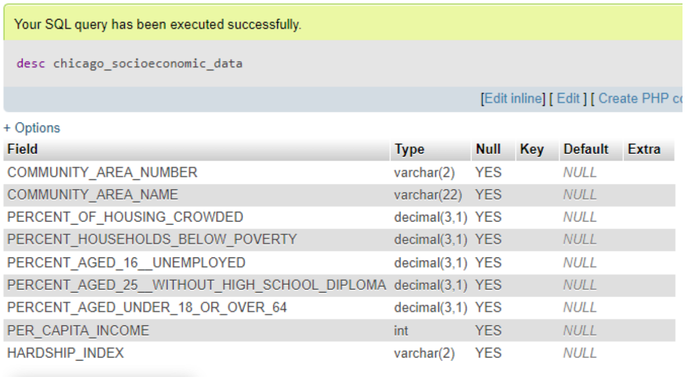


Your SQL query has been executed successfully.

```
desc chicago_crime
```

Field	Type	Null	Key	Default	Extra
ID	int	YES		*NULL*	
CASE_NUMBER	varchar(8)	YES		*NULL*	
DATE	varchar(10)	YES		*NULL*	
BLOCK	varchar(35)	YES		*NULL*	
IUCR	varchar(4)	YES		*NULL*	
PRIMARY_TYPE	varchar(33)	YES		*NULL*	
DESCRIPTION	varchar(46)	YES		*NULL*	
LOCATION_DESCRIPTION	varchar(33)	YES		*NULL*	
ARREST	varchar(5)	YES		*NULL*	
DOMESTIC	varchar(5)	YES		*NULL*	
BEAT	int	YES		*NULL*	
DISTRICT	int	YES		*NULL*	
WARD	varchar(2)	YES		*NULL*	
COMMUNITY_AREA_NUMBER	varchar(2)	YES		*NULL*	
FBICODE	varchar(3)	YES		*NULL*	
X_COORDINATE	varchar(7)	YES		*NULL*	
Y_COORDINATE	varchar(7)	YES		*NULL*	
YEAR	int	YES		*NULL*	
LATITUDE	varchar(30)	YES		*NULL*	
LONGITUDE	varchar(30)	YES		*NULL*	
LOCATION	varchar(29)	YES		*NULL*	

3. Socioeconomic table named Census Data



Your SQL query has been executed successfully.

```
desc chicago_socioeconomic_data
```

Field	Type	Null	Key	Default	Extra
COMMUNITY_AREA_NUMBER	varchar(2)	YES		*NULL*	
COMMUNITY_AREA_NAME	varchar(22)	YES		*NULL*	
PERCENT_OF_HOUSING_CROWDED	decimal(3,1)	YES		*NULL*	
PERCENT_HOUSEHOLDS_BELOW_POVERTY	decimal(3,1)	YES		*NULL*	
PERCENT_AGED_16__UNEMPLOYED	decimal(3,1)	YES		*NULL*	
PERCENT_AGED_25__WITHOUT_HIGH_SCHOOL_DIPLOMA	decimal(3,1)	YES		*NULL*	
PERCENT_AGED_UNDER_18_OR_OVER_64	decimal(3,1)	YES		*NULL*	
PER_CAPITA_INCOME	int	YES		*NULL*	
HARDSHIP_INDEX	varchar(2)	YES		*NULL*	

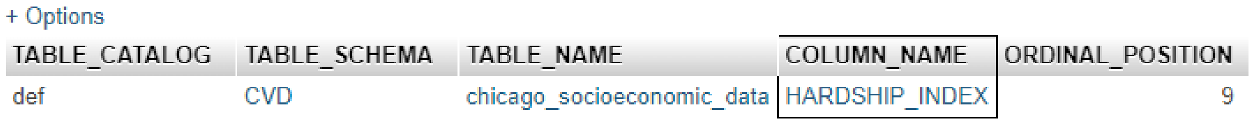
Final Project: Advanced SQL Techniques

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.***

To find those columns in the data base for which I am not sure which table it belongs I use this query:

```
SELECT * FROM INFORMATION_SCHEMA.COLUMNS
WHERE COLUMN_NAME LIKE '%HARD%'
ORDER BY TABLE_NAME;
```

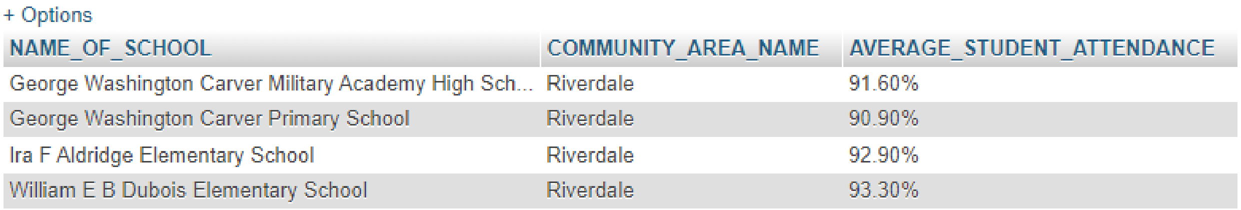


+ Options

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION
def	CVD	chicago_socioeconomic_data	HARDSHIP_INDEX	9

```
SELECT CPS.NAME_OF_SCHOOL, CSED.COMMUNITY_AREA_NAME,
CPS.AVERAGE_STUDENT_ATTENDANCE
```

```
FROM chicago_public_schools AS CPS

LEFT JOIN chicago_socioeconomic_data AS CSED

ON CPS.COMMUNITY_AREA_NUMBER = CSED.COMMUNITY_AREA_NUMBER

WHERE CSED.HARDSHIP_INDEX = 98;
```



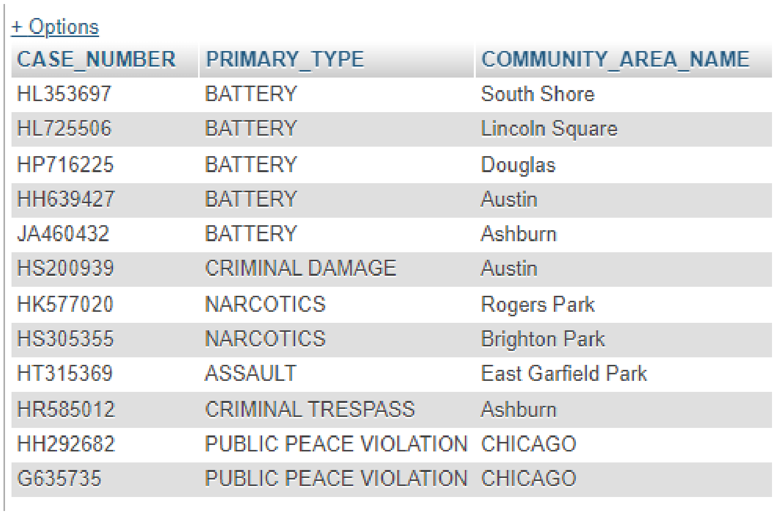
NAME_OF_SCHOOL	COMMUNITY_AREA_NAME	AVERAGE_STUDENT_ATTENDANCE
George Washington Carver Military Academy High Sch...	Riverdale	91.60%
George Washington Carver Primary School	Riverdale	90.90%
Ira F Aldridge Elementary School	Riverdale	92.90%
William E B Dubois Elementary School	Riverdale	93.30%

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.

```
SELECT CCD.CASE_NUMBER, CCD.PRIMARY_TYPE, CSED.COMMUNITY_AREA_NAME

FROM chicago_crime AS CCD

LEFT JOIN chicago_socioeconomic_data AS CSED

ON CCD.COMMUNITY_AREA_NUMBER = CSED.COMMUNITY_AREA_NUMBER

WHERE LOWER(CCD.LOCATION_DESCRIPTION) LIKE '%school%';
```

Final Project: Advanced SQL Techniques



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
G635735	PUBLIC PEACE VIOLATION	CHICAGO

Exercise 2: Creating a View

Question 1

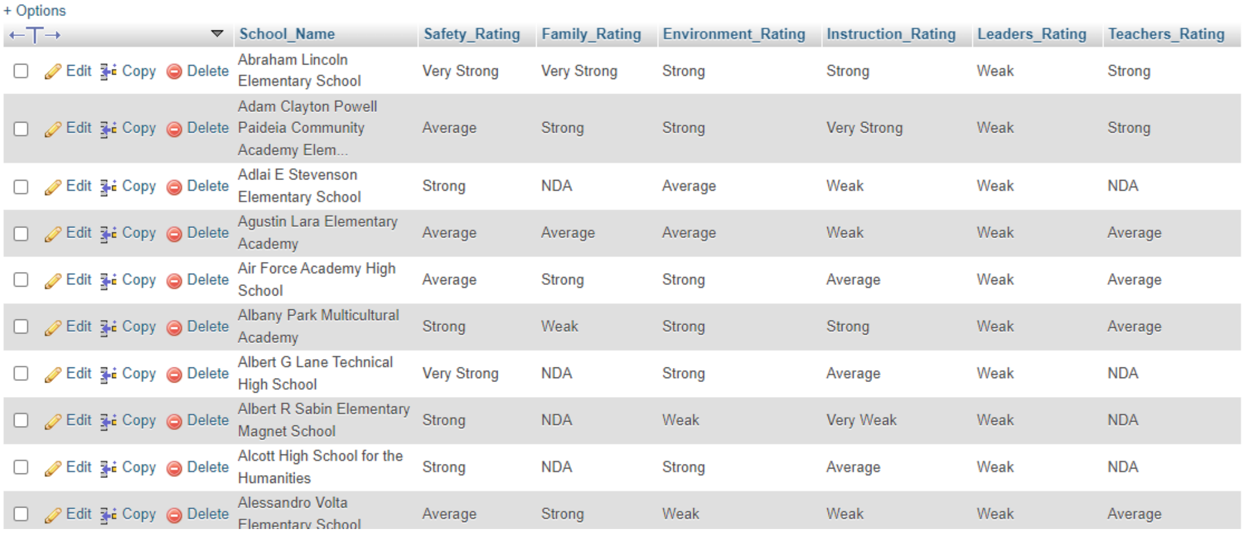
Write and execute a SQL statement to create a view showing the columns listed in the following table, with new column names as shown in the second column. community name.

```
CREATE VIEW chicago_school_info(School_Name, Safety_Rating,
Family_Rating, Environment_Rating, Instruction_Rating, Leaders_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;
```

Final Project: Advanced SQL Techniques

Write and execute a SQL statement that returns all of the columns from the view.

```
SELECT * FROM chicago_school_info;
```



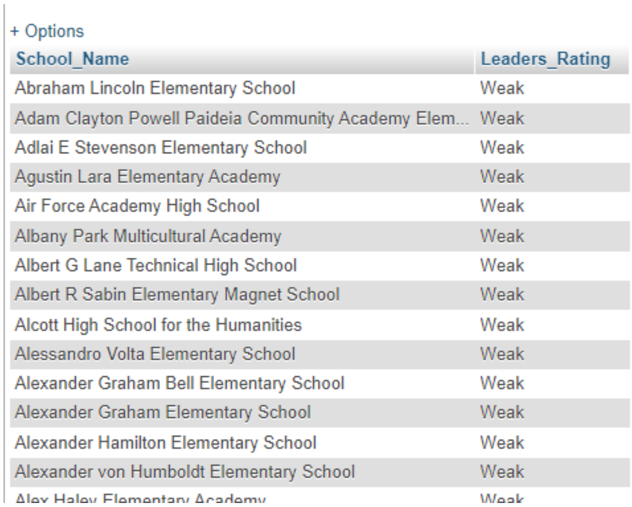
+ Options

←T→ ▼	School_Name	Safety_Rating	Family_Rating	Environment_Rating	Instruction_Rating	Leaders_Rating	Teachers_Rating
□ Edit Copy Delete	Abraham Lincoln Elementary School	Very Strong	Very Strong	Strong	Strong	Weak	Strong
□ Edit Copy Delete	Adam Clayton Powell Paideia Community Academy Elem...	Average	Strong	Strong	Very Strong	Weak	Strong
□ Edit Copy Delete	Adlai E Stevenson Elementary School	Strong	NDA	Average	Weak	Weak	NDA
□ Edit Copy Delete	Agustin Lara Elementary Academy	Average	Average	Average	Weak	Weak	Average
□ Edit Copy Delete	Air Force Academy High School	Average	Strong	Strong	Average	Weak	Average
□ Edit Copy Delete	Albany Park Multicultural Academy	Strong	Weak	Strong	Strong	Weak	Average
□ Edit Copy Delete	Albert G Lane Technical High School	Very Strong	NDA	Strong	Average	Weak	NDA
□ Edit Copy Delete	Albert R Sabin Elementary Magnet School	Strong	NDA	Weak	Very Weak	Weak	NDA
□ Edit Copy Delete	Alcott High School for the Humanities	Strong	NDA	Strong	Average	Weak	NDA
□ Edit Copy Delete	Alessandro Volta Elementary School	Average	Strong	Weak	Weak	Weak	Average

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

```
SELECT School_Name, Leaders_Rating FROM chicago_school_info;
```

Final Project: Advanced SQL Techniques



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
Alex Haley Elementary Academy	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.

```
DELIMITER //

CREATE OR REPLACE PROCEDURE 'UPDATE_LEADERS_SCORE' (in_School_ID int,
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.

```
DELIMITER //

CREATE OR REPLACE PROCEDURE UPDATE_LEADERS_SCORE (in_School_ID int,
in_Leader_Score int)

BEGIN

      UPDATE chicago_public_schools

      SET Leader_Score = in_Leader_Score

      WHERE School_ID = in_School_ID;

END //

DELIMITER ;
```

Question 3

Final Project: Advanced SQL Techniques

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.

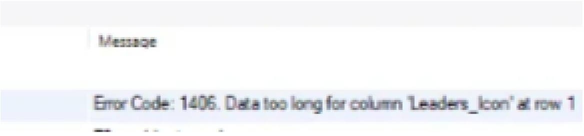
```
DELIMITER //

CREATE PROCEDURE UPDATE_LEADERS_SCORE (in_School_ID int, 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 >0 AND in_Leader_Score <20

            THEN UPDATE chicago_public_schools

            SET Leaders_Icon ='Very Weak'

            WHERE School_ID = in_School_ID;

      ELSEIF in_Leader_Score < 40

            THEN UPDATE chicago_public_schools

            SET Leaders_Icon ='Weak'

            WHERE School_ID = in_School_ID;

      ELSEIF in_Leader_Score < 60

            THEN UPDATE chicago_public_schools

            SET Leaders_Icon ='Average'

            WHERE School_ID = in_School_ID;
```

```
     ELSEIF in_Leader_Score < 80
           THEN UPDATE chicago_public_schools
           SET Leaders_Icon ='Strong'
           WHERE School_ID = in_School_ID;
     ELSEIF in_Leader_Score < 100
           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.

Data type of leaders icon is varchar(4) so an error was given if character longer than 4 was entered. We can see below:



```
Message
Error Code: 1406. Data too long for column 'Leaders_Icon' at row 1
```

Hence I first modified the data type.

Final Project: Advanced SQL Techniques

```
ALTER TABLE chicago_public_schools MODIFY COLUMN leaders_icon
varchar(15);


CALL UPDATE_LEADERS_SCORE(610084,50);


SELECT School_ID, leaders_icon, Leaders_Score FROM
chicago_public_schools WHERE School_ID=610084;
```