-- School Ranking Analysis --

use sql\_basics;

CREATE TABLE students (

student\_id INT PRIMARY KEY NOT NULL,

student\_first\_name VARCHAR(50) NOT NULL,

student\_last\_name VARCHAR(50) NOT NULL,

class VARCHAR(20),

age INT

);

CREATE TABLE marksheet (

score INT,

year INT,

ranking INT,

class VARCHAR(20),

student\_id INT

);

INSERT INTO students VALUES (1,'krishna','gee',10,18),

(2,'Stephen','Christ',10,17),

(3,'Kailash','kumar',10,18),

(4,'ashish','jain',10,16),

(5,'khusbu','jain',10,17),

(6,'madhan','lal',10,16),

(7,'saurab','kothari',10,15),

(8,'vinesh','roy',10,14),

(9,'rishika','r',10,15),

(10,'sara','rayan',10,16),

(11,'rosy','kumar',10,16);

INSERT INTO marksheet VALUES (989,2014,10,1,1),

(454,2014,10,10,2),

(880,2014,10,4,3),

(870,2014,10,5,4),

(720,2014,10,7,5),

(670,2014,10,8,6),

(900,2014,10,3,7),

(540,2014,10,9,8),

(801,2014,10,6,9),

(420,2014,10,11,10),

(970,2014,10,2,11),

(720,2014,10,12,12);

select \* from students;

select \* from marksheet;

-- Write a query to display student id and student first name from the student table if the age is greater than or equal to 16 and the student's last name is Kumar --

SELECT student\_id, student\_first\_name

FROM students

WHERE age >= 16 AND student\_last\_name = 'Kumar';

-- Write a query to display all the details from the marksheet table if the score is between 800 and 1000 --

SELECT \*

FROM marksheet

WHERE score BETWEEN 800 AND 1000;

-- Write a query to display the marksheet details from the marksheet table by adding 5 to the score and by naming the column as new score --

SELECT \*, score + 5 AS new\_score

FROM marksheet;

-- Write a query to display the marksheet table in descending order of the score --

SELECT \*

FROM marksheet

ORDER BY score DESC;

-- Write a query to display details of the students whose first name starts with a --

SELECT \*

FROM students

WHERE student\_first\_name LIKE 'a%';