```sql
-- School Ranking Analysis Project --
-/* DESCRIPTION : Consider an institution that wants to store the students' details and their marks records to track their progress.
                  The database would contain the students' information, marks of the students with the rank that can be viewed, updated, and evaluated for the performance evaluation.*/
-/* Objective   : The design of the database helps to easily retrieve thousands of student records.*/


-- Task to be performed:
-- TASK01:
/* Write a query to create a students table with appropriate data types for student id, student first name, student last name, class, and age where the student last name, student first name, and student id should be a NOT NULL constraint, and the student id should be in a primary key. */

CREATE DATABASE student_datasets;
USE student_datasets;
CREATE TABLE student_datasets.students_table
(
s_id INT NOT NULL,
s_fname VARCHAR (200) NOT NULL,
s_lname VARCHAR (200) NOT NULL,
class INT,
age INT,
PRIMARY KEY (s_id)
);
DESCRIBE student_datasets.students_table;

-- TASK02:
-- Write a query to create a marksheet table that includes score, year, ranking, class, and student id.

CREATE DATABASE marksheet_datasets;
USE marksheet_datasets;
CREATE TABLE marksheet_datasets.marksheet_table
(
score INT NOT NULL,
year year NOT NULL,
class INT NOT NULL,
ranking INT NOT NULL,
s_id INT NOT NULL PRIMARY KEY
);
DESCRIBE marksheet_datasets.marksheet_table;

-- TASK03:
-- Write a query to insert values in students and marksheet tables.

-- insert values in students table
INSERT INTO student_datasets.students_table (s_id,s_fname,s_lname,class,age)
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");
SELECT*FROM student_datasets.students_table;

-- insert values in marksheet table
INSERT INTO marksheet_datasets.marksheet_table (score,year,class,ranking,s_id)
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 marksheet_datasets.marksheet_table;

-- TASK04:
-- 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 s_id, s_fname FROM student_datasets.students_table WHERE
age >= 16 AND s_lname = "KUMAR";

-- TASK 05:
-- Write a query to display all the details from the marksheet table if the score is between 800 and 1000.
SELECT*FROM marksheet_datasets.marksheet_table
 WHERE score BETWEEN 800 AND 1000;

-- TASK 06:
-- 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_datasets.marksheet_table;

 -- TASK 07:
-- Write a query to display the marksheet table in descending order of the score.
SELECT * FROM marksheet_datasets.marksheet_table
order by score DESC;

-- TASK 08:
-- Write a query to display details of the students whose first name starts with a.
SELECT*FROM student_datasets.students_table
WHERE s_fname like "a%";
```