School Ranking Analysis.

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:

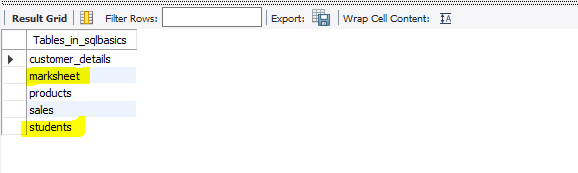
The design of the database helps to retrieve thousands of student records easily.

Task to be performed:

1. 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 TABLE **STUDENTS**( STUDENT\_ID INTEGER PRIMARY KEY NOT NULL, FIRST\_NAME VARCHAR(20),LAST\_NAME VARCHAR(20),CLASS VARCHAR(20),AGE INTEGER);

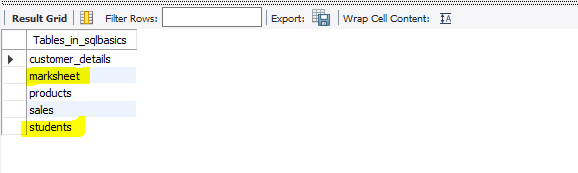
SHOW TABLES;



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

CREATE TABLE **MARKSHEET**( SCORE INTEGER, YEAR INTEGER,RANKING INTEGER,CLASS VARCHAR(20),STUDENT\_ID INTEGER);

SHOW TABLES;



1. Write a query to insert values in students and marksheet tables.

INSERT INTO STUDENTS (STUDENT\_ID, FIRST\_NAME, LAST\_NAME, 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),

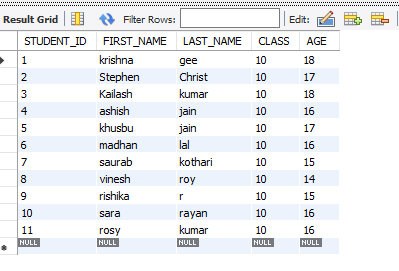
(7, 'saurab', 'kothari', '10', 15),

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

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

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

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



INSERT INTO MARKSHEET (SCORE, YEAR, CLASS, RANKING, STUDENT\_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),

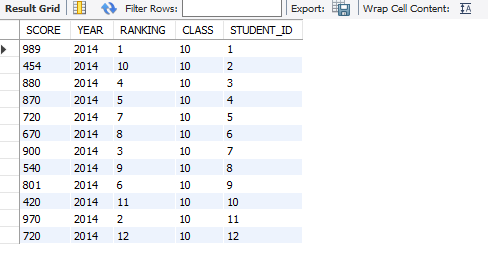
(540, 2014, '10', 9, 8),

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

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

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

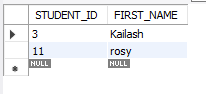
(720, 2014, '10', 12, 12);



1. 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.

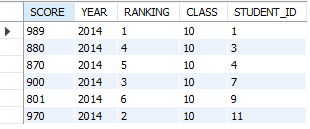
SELECT STUDENT\_ID, FIRST\_NAME FROM STUDENTS

WHERE AGE>=16 AND LAST\_NAME = 'KUMAR';



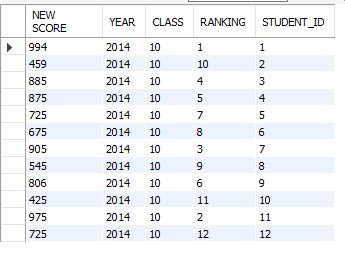
1. Write a query to display all the details from the mark sheet table if the score is between 800 and 1000.

SELECT \* FROM MARKSHEET WHERE SCORE BETWEEN 800 AND 1000;



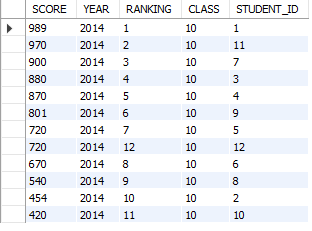
1. Write a query to display the mark sheet details from the mark sheet table by adding 5 to the score and by naming the column as a new score.

SELECT SCORE + 5 AS 'NEW SCORE', YEAR, CLASS, RANKING, STUDENT\_ID FROM MARKSHEET;



1. Write a query to display the mark sheet table in descending order of the score.

SELECT \* FROM MARKSHEET ORDER BY SCORE DESC;



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

SELECT \* FROM STUDENTS WHERE FIRST\_NAME LIKE “a%”;

