School Ranking Analysis

Problem statement:

You are a database administrator in an institution, and you have been asked to store the students' details and their marks to track their progress. The database helps to view the students' marks with a rank that can be viewed, updated, and evaluated to evaluate their performance.

Objective:

The objective is to design a database to retrieve the information of a student as needed for the records.

1. **-- create a database named SQL basics:**

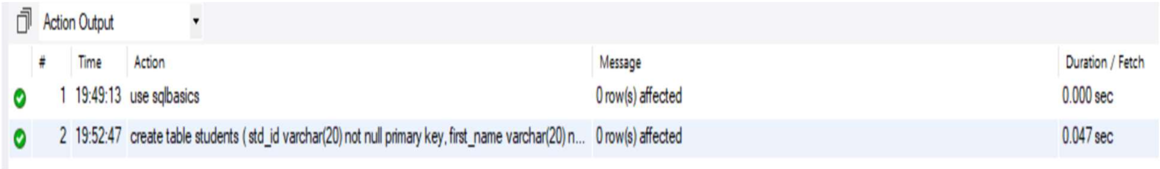
   ```
   create database sqlbasics;
   ```

2. **-- select SQL basics:**

   ```
   use sqlbasics;
   ```

3. **-- Write a query to create a students table with the student ID, first name, last name, class, and age fields and ensure that the last name, first name, and student ID fields have the NOT NULL constraint and that the student ID field is a primary key**

   ```
   create table students (
   std_id varchar(20) not null primary key,
   first_name varchar(20) not null,
   last_name varchar(20) not null,
   class varchar(10),
   age int);
   ```



	#	Time	Action	Message	Duration / Fetch
✓	1	19:49:13	use sqlbasics	0 row(s) affected	0.000 sec
✓	2	19:52:47	create table students (std_id varchar(20) not null primary key, first_name varchar(20) n...	0 row(s) affected	0.047 sec

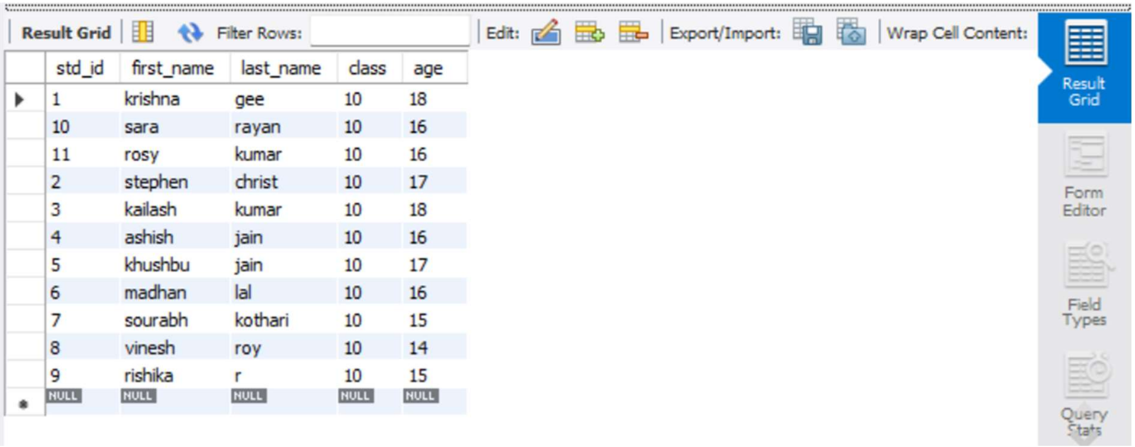
4. **-- Write a query to create a marksheet table with score, year, ranking, class, and student ID fields:**

   ```
   create table marksheet (
   score int,
   year int,
   ranking varchar(20),
   class varchar(10),
   std_id varchar(20) not null);
   ```

5. **-- Write a query to insert values into the students table:**

```
insert into students(std_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,'khushbu','jain',10,17),
(6,'madhan','lal',10,16),
(7,'sourabh','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 students;
```

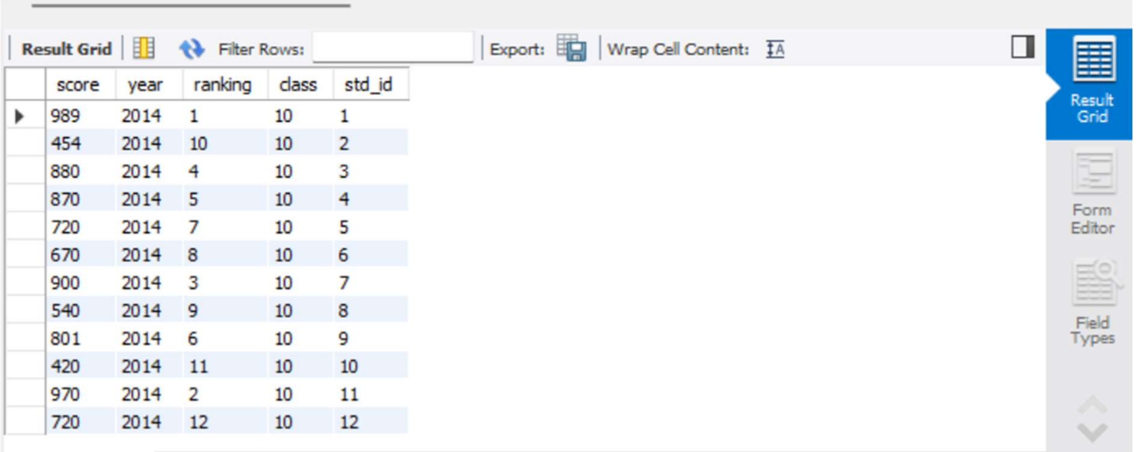


	std_id	first_name	last_name	class	age
▶	1	krishna	gee	10	18
	10	sara	rayan	10	16
	11	rosy	kumar	10	16
	2	stephen	christ	10	17
	3	kailash	kumar	10	18
	4	ashish	jain	10	16
	5	khushbu	jain	10	17
	6	madhan	lal	10	16
	7	sourabh	kothari	10	15
	8	vinesh	roy	10	14
	9	rishika	r	10	15
*	NULL	NULL	NULL	NULL	NULL

6. **-- Write a query to insert values into the marksheet table:**

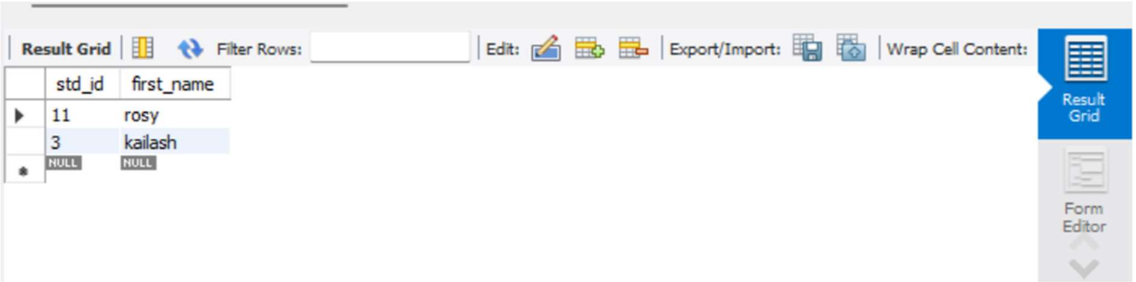
```
insert into marksheet (score, year, ranking, class, std_id)
values (989,2014,1,10,1),
(454,2014,10,10,2),
(880,2014,4,10,3),
(870,2014,5,10,4),
(720,2014,7,10,5),
(670,2014,8,10,6),
(900,2014,3,10,7),
(540,2014,9,10,8),
(801,2014,6,10,9),
(420,2014,11,10,10),
(970,2014,2,10,11),
(720,2014,12,10,12);

Select*from marksheet;
```



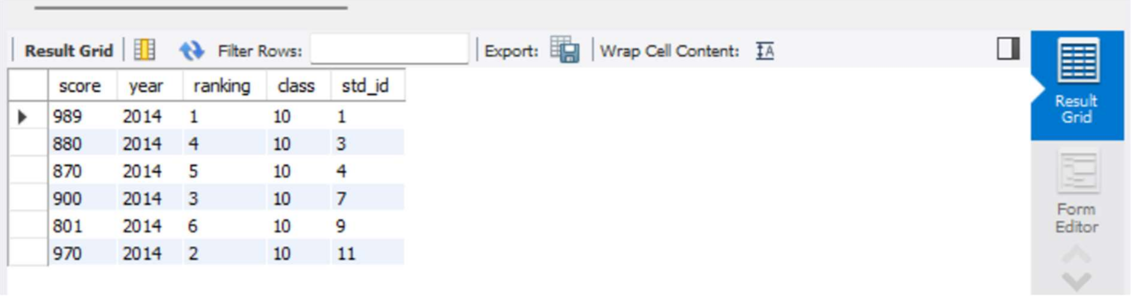
score	year	ranking	class	std_id
989	2014	1	10	1
454	2014	10	10	2
880	2014	4	10	3
870	2014	5	10	4
720	2014	7	10	5
670	2014	8	10	6
900	2014	3	10	7
540	2014	9	10	8
801	2014	6	10	9
420	2014	11	10	10
970	2014	2	10	11
720	2014	12	10	12

7. **-- Write a query to display the student ID and first name of every student in the students table whose age is greater than or equal to 16 and whose last name is Kumar:**
   select std_id, first_name from students where age>=16 and last_name='kumar';



std_id	first_name
11	rosy
3	kailash
NULL	NULL

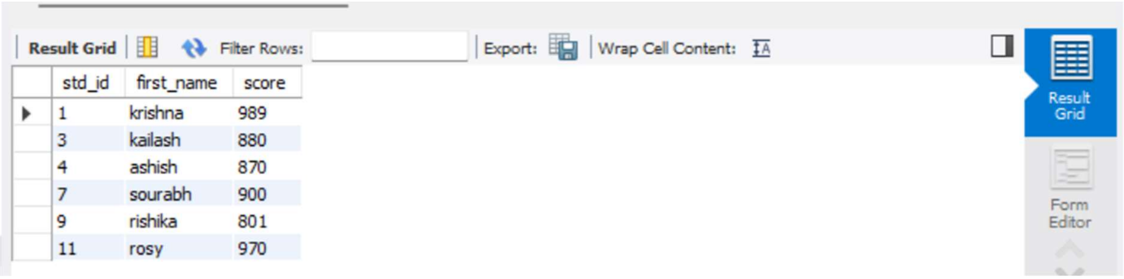
8. **-- Write a query to display the details of every student from the marksheet table whose score is between 800 and 1000:**
   select*from marksheet where score between 800 and 1000;



score	year	ranking	class	std_id
989	2014	1	10	1
880	2014	4	10	3
870	2014	5	10	4
900	2014	3	10	7
801	2014	6	10	9
970	2014	2	10	11

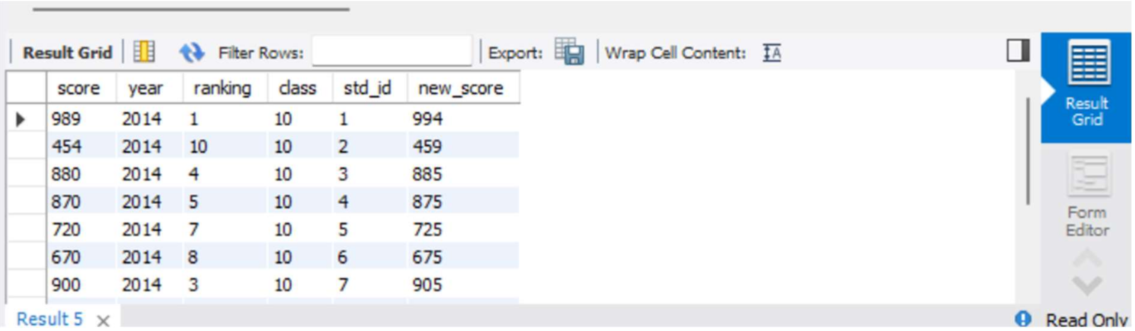
OR

select s.std_id, s.first_name, m.score from students s left join marksheet m on s.std_id=m.std_id where score between 800 and 1000;



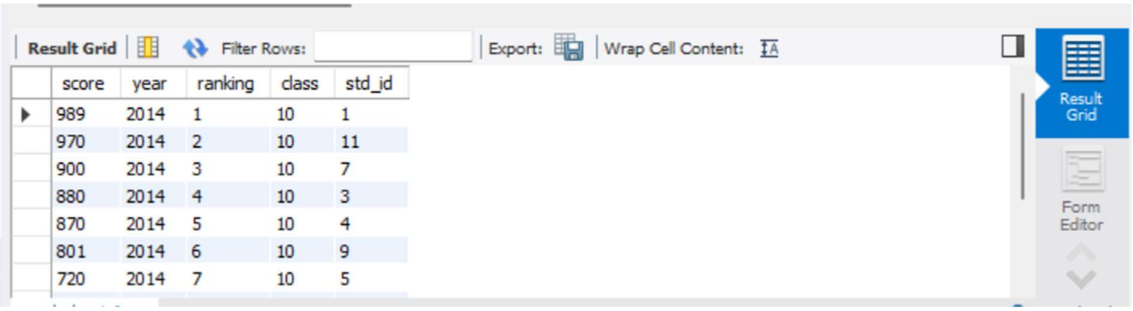
std_id	first_name	score
1	krishna	989
3	kailash	880
4	ashish	870
7	sourabh	900
9	rishika	801
11	rosy	970

9. **-- Write a query to increase the score in the marksheet table by five and create a new score column to display this new score:**
select*,score+5 as new_score from marksheet;



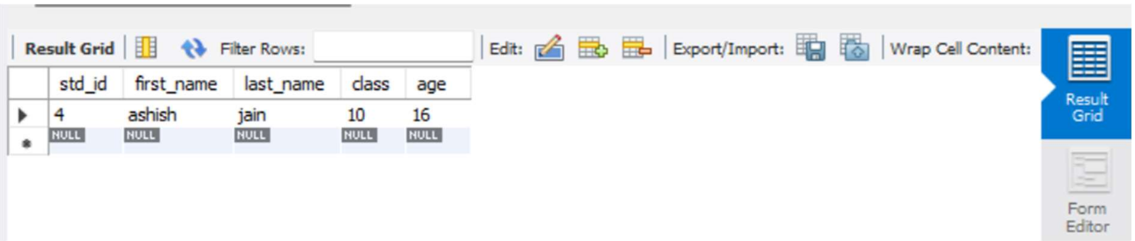
score	year	ranking	class	std_id	new_score
989	2014	1	10	1	994
454	2014	10	10	2	459
880	2014	4	10	3	885
870	2014	5	10	4	875
720	2014	7	10	5	725
670	2014	8	10	6	675
900	2014	3	10	7	905

10. **-- Write a query to display the marksheet table in descending order of the score:**
select*from marksheet order by score desc;



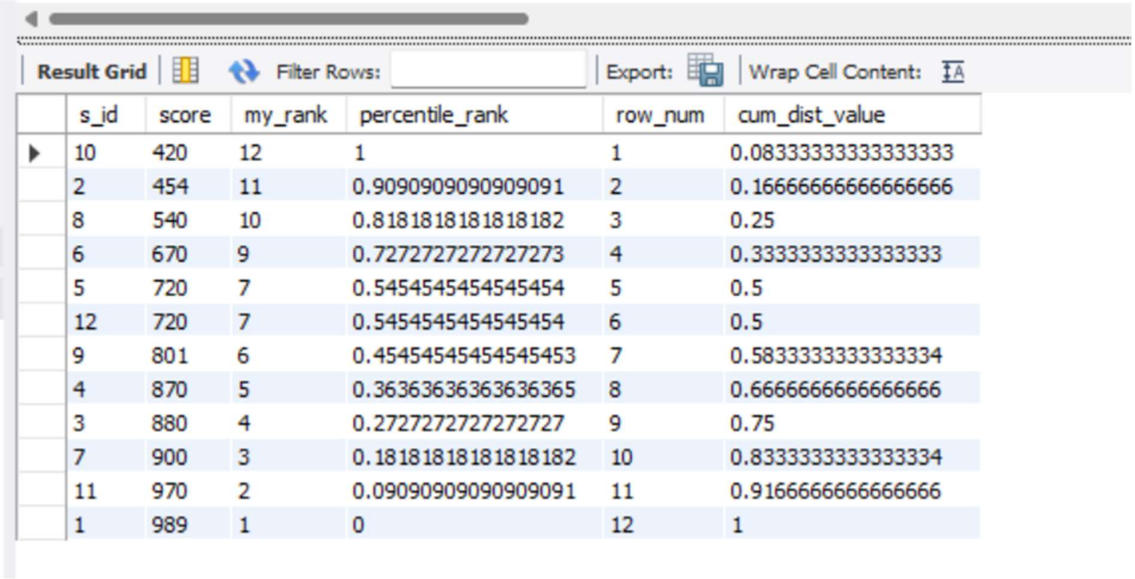
score	year	ranking	class	std_id
989	2014	1	10	1
970	2014	2	10	11
900	2014	3	10	7
880	2014	4	10	3
870	2014	5	10	4
801	2014	6	10	9
720	2014	7	10	5

11. **-- Write a query to display the details of every student whose first name starts with an ‘a':**
select*from students where first_name like 'a%';



std_id	first_name	last_name	class	age
4	ashish	jain	10	16
NULL	NULL	NULL	NULL	NULL

12. **-- You are required to identify the rank and row number and calculate the cumulative distribution and percentile score based on the student score from the marksheet table:**
select s_id, score, rank() over (order by score desc) my_rank, percent_rank() over (order by score desc) percentile_rank, row_number() over (order by score) row_num, cume_dist() over (order by score) cum_dist_value from marksheet_datasets;



s_id	score	my_rank	percentile_rank	row_num	cum_dist_value
10	420	12	1	1	0.0833333333333333
2	454	11	0.9090909090909091	2	0.1666666666666666
8	540	10	0.8181818181818182	3	0.25
6	670	9	0.7272727272727273	4	0.3333333333333333
5	720	7	0.5454545454545454	5	0.5
12	720	7	0.5454545454545454	6	0.5
9	801	6	0.45454545454545453	7	0.5833333333333334
4	870	5	0.36363636363636365	8	0.6666666666666666
3	880	4	0.2727272727272727	9	0.75
7	900	3	0.18181818181818182	10	0.8333333333333334
11	970	2	0.09090909090909091	11	0.9166666666666666
1	989	1	0	12	1