School Ranking Analysis.

DESCRIPTION

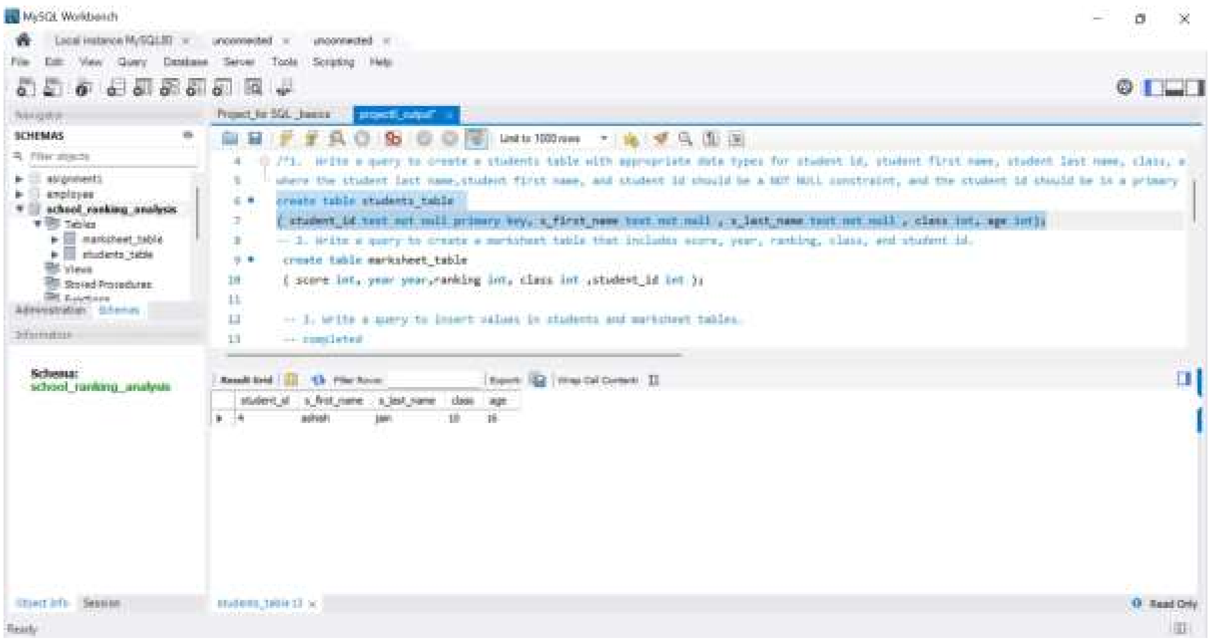
Consider an institution that wants to store the students' details and their marks records to track their progress. The database would contain the students' information, marks of the students with the rank that can be viewed, updated, and evaluated for the performance evaluation.

Objective:

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

Task to be performed:

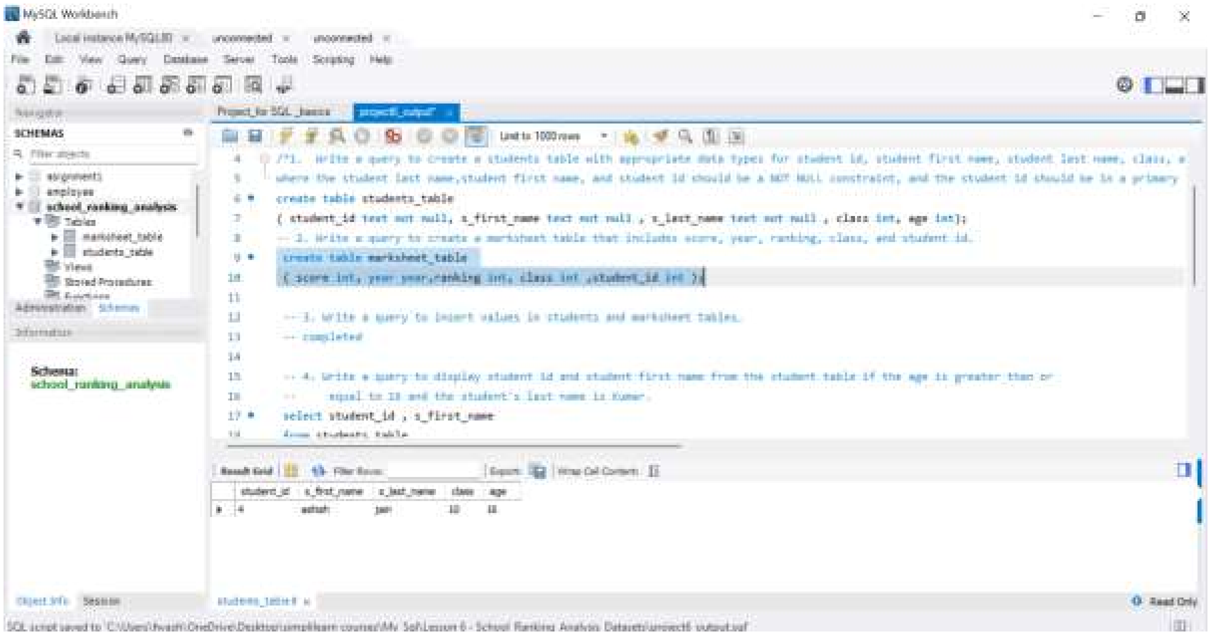
- Write a query to create a **students** table with appropriate data types for student id, student first name, student last name, class, and age where the student last name, student first name, and student id should be a **NOT NULL constraint**, and the student id should be in a **primary key**.



```
/*1.  Write a query to create a students table with appropriate data types for student id, student first name, student last name, class, a
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
create table students_table
( student_id text not null primary key, s_first_name text not null , s_last_name text not null , class int, age int);
-- 2. Write a query to create a marksheet table that includes score, year, ranking, class, and student id.
 create table marksheet_table
 ( score int, year year,ranking int, class int ,student_id int );

 -- 3. Write a query to insert values in students and marksheet tables.
 -- completed
```

student_id	s_first_name	s_last_name	class	age
4	ashish	jain	10	16

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

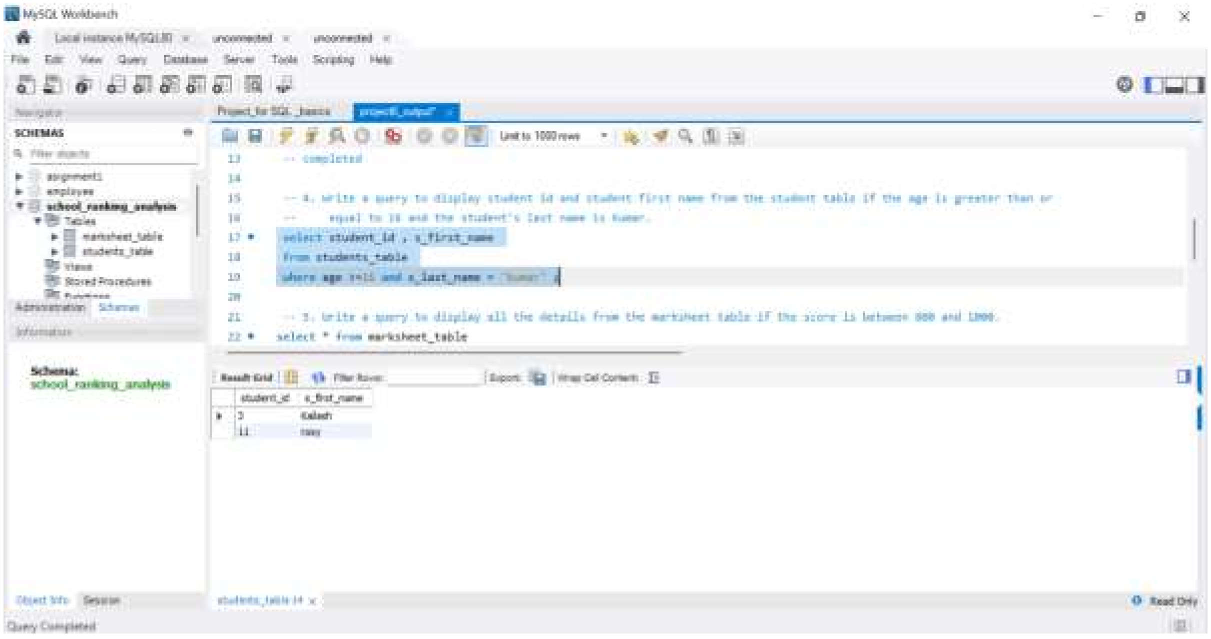


```
/*1.  Write a query to create a students table with appropriate data types for student id, student first name, student last name, class, a
where the student last name,student first name, and student id should be a NOT NULL constraint, and the student id should be in a primary
create table students_table
( student_id text not null, s_first_name text not null , s_last_name text not null , class int, age int);
-- 2. Write a query to create a marksheet table that includes score, year, ranking, class, and student id.
 create table marksheet_table
 ( score int, year year,ranking int, class int ,student_id int );

 -- 3. Write a query to insert values in students and marksheet tables.
 -- completed

 -- 4. 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 , s_first_name
 from students_table
```

```
   student_id   s_first_name   s_last_name   class   age
▶  4            ashish         jain          10      16
```

SQL script saved to 'C:\Users\hvash\OneDrive\Desktop\simplilearn courses\My Sql\Lesson 6 - School Ranking Analysis Datasets\project6_output.sql'

4. . 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.**

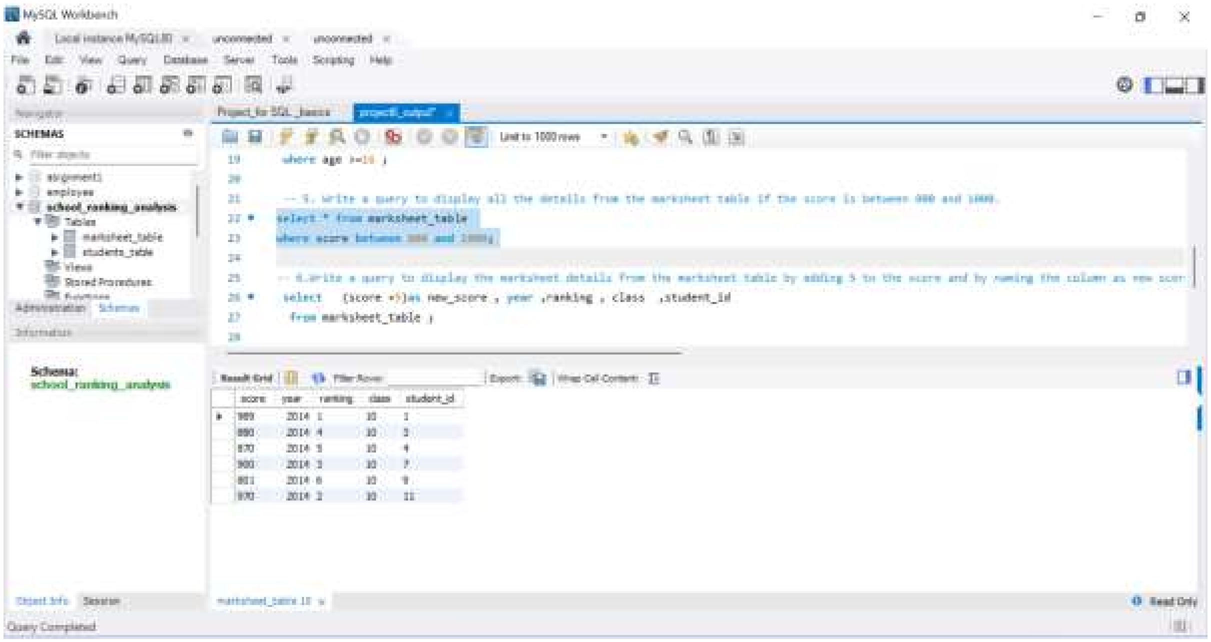


```
 -- completed

 -- 4. 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 , s_first_name
 from students_table
 where age >=16 and s_last_name = 'Kumar' ;

 -- 5. Write a query to display all the details from the marksheet table if the score is between 800 and 1000.
select * from marksheet_table
```

```
   student_id   s_first_name
▶  3            Kailash
   11           ravi
```

Query Completed

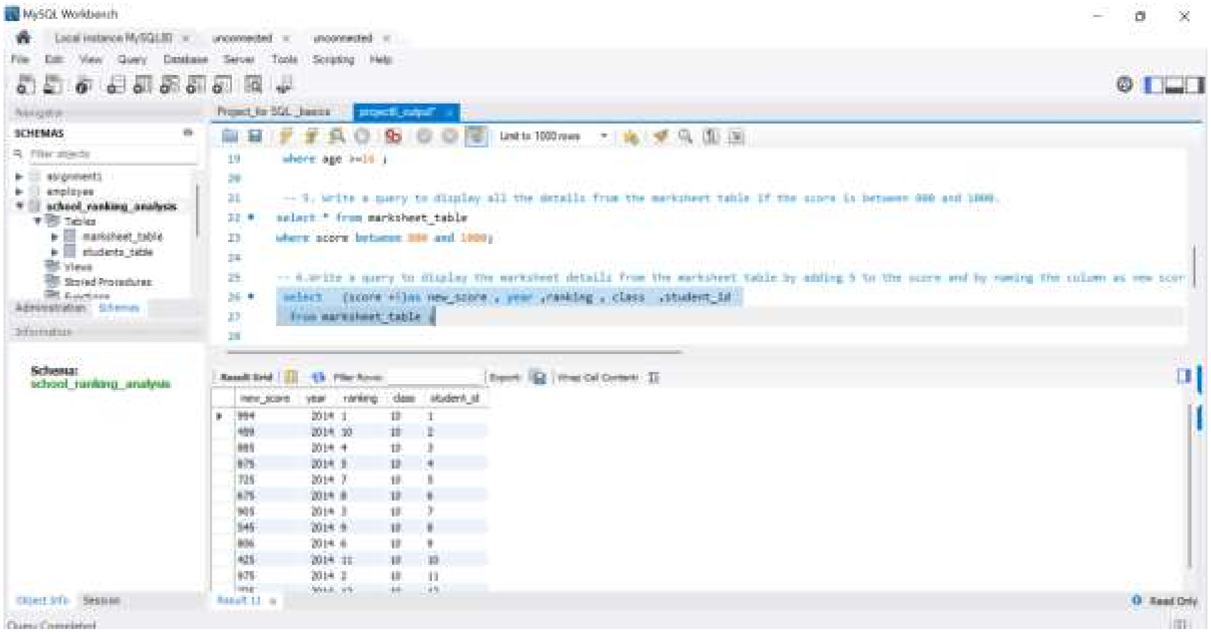
5. Write a query to display all the details from the marksheet table **if the score is between 800 and 1000.**



```
    where age >=16 ;

    -- 5. write a query to display all the details from the marksheet table if the score is between 800 and 1000.
select * from marksheet_table
where score between 800 and 1000;

-- 6.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 scor
 select   (score +5)as new_score , year ,ranking , class  ,student_id
  from marksheet_table ;
```

score	year	ranking	class	student_id
989	2014	1	10	1
890	2014	4	10	3
870	2014	5	10	4
900	2014	3	10	7
801	2014	6	10	9
870	2014	2	10	11

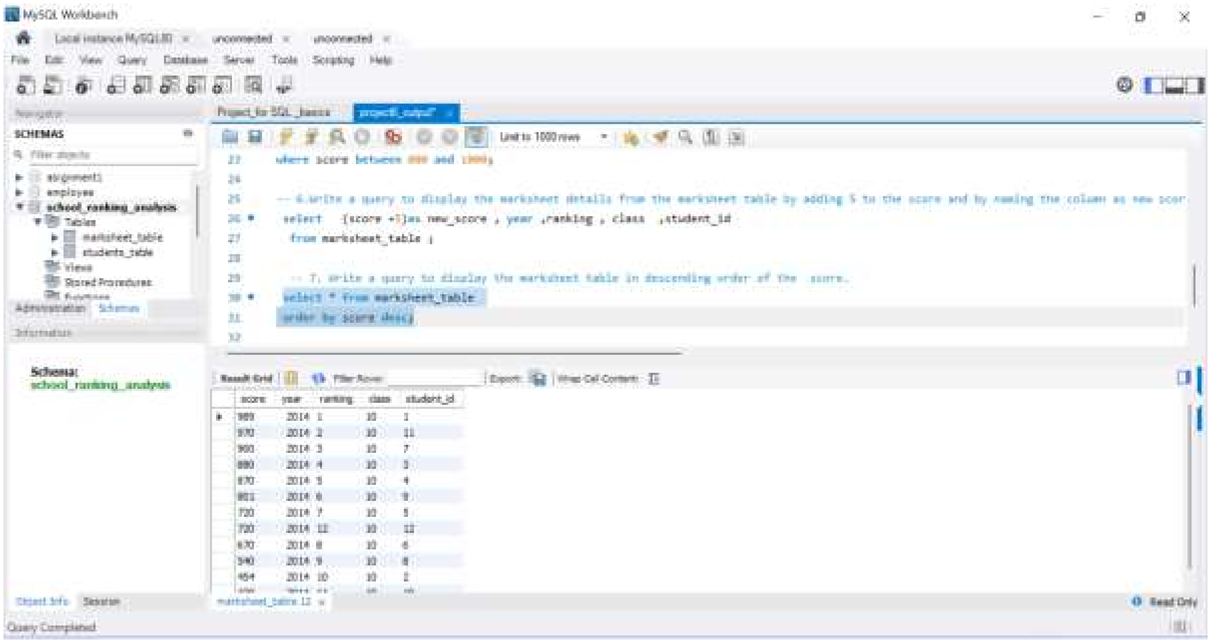
- 6. Write a query to display the marksheet details from the marksheet table by **adding 5 to the score** and by naming the **column** as **new score**.



```
    where age >=16 ;

    -- 5. write a query to display all the details from the marksheet table if the score is between 800 and 1000.
select * from marksheet_table
where score between 800 and 1000;

-- 6.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 scor
 select   (score +5)as new_score , year ,ranking , class  ,student_id
  from marksheet_table ;
```

new_score	year	ranking	class	student_id
994	2014	1	10	1
499	2014	10	10	2
895	2014	4	10	3
875	2014	5	10	4
725	2014	7	10	5
675	2014	8	10	6
905	2014	3	10	7
546	2014	9	10	8
806	2014	6	10	9
425	2014	11	10	10
875	2014	2	10	11
[illegible]	2014	[illegible]	10	[illegible]

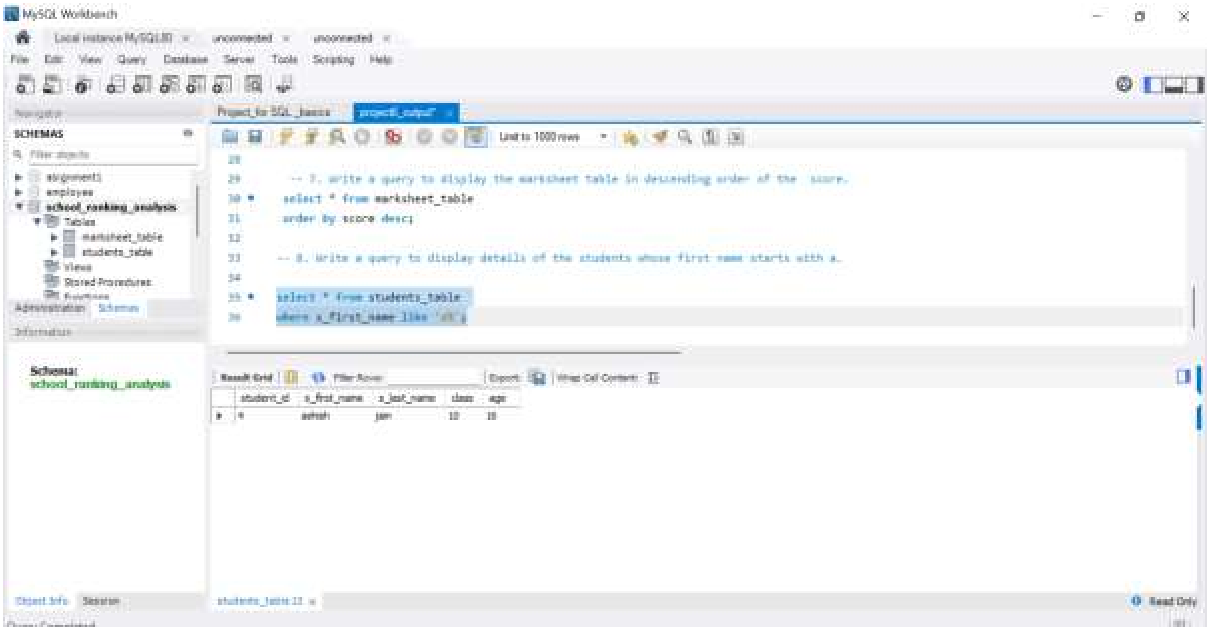
7. Write a query to display the marksheet table in **descending order of the score**.



```
where score between 800 and 1000;

-- 6.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 scor
 select   (score +5)as new_score , year ,ranking , class  ,student_id
  from marksheet_table ;

  -- 7. write a query to display the marksheet table in descending order of the  score.
 select * from marksheet_table
 order by score desc;
```

score	year	ranking	class	student_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
720	2014	11	10	12
670	2014	8	10	6
540	2014	9	10	8
454	2014	10	10	2

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



```
  -- 7. write a query to display the marksheet table in descending order of the  score.
 select * from marksheet_table
 order by score desc;

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

select * from students_table
where s_first_name like 'a%';
```

student_id	s_first_name	s_last_name	class	age
4	ashish	jain	10	15