page2image568

SQL Training

Lesson-End Project Solution



**School Ranking Analysis**

1. 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

**SQL code:**

CREATE TABLE lep\_5.students (

s\_id INT NOT NULL,

s\_fname varchar(45) NOT NULL,

s\_lname varchar(45) NOT NULL,

class varchar(45) NULL,

age INT NOT NULL,

PRIMARY KEY(s\_id));

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

**SQL code:**

CREATE TABLE lep\_5.marksheet (

score INT NOT NULL,

year INT NULL,

class varchar(45) NULL,

ranking varchar(45) NULL,

s\_id INT NOT NULL);

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

**SQL code: Students table**

INSERT INTO lep\_5. students (s\_id,s\_fname,s\_lname,class,age) VALUES ('01','krishna','gee','10','18');

**SQL code: Marksheet table**

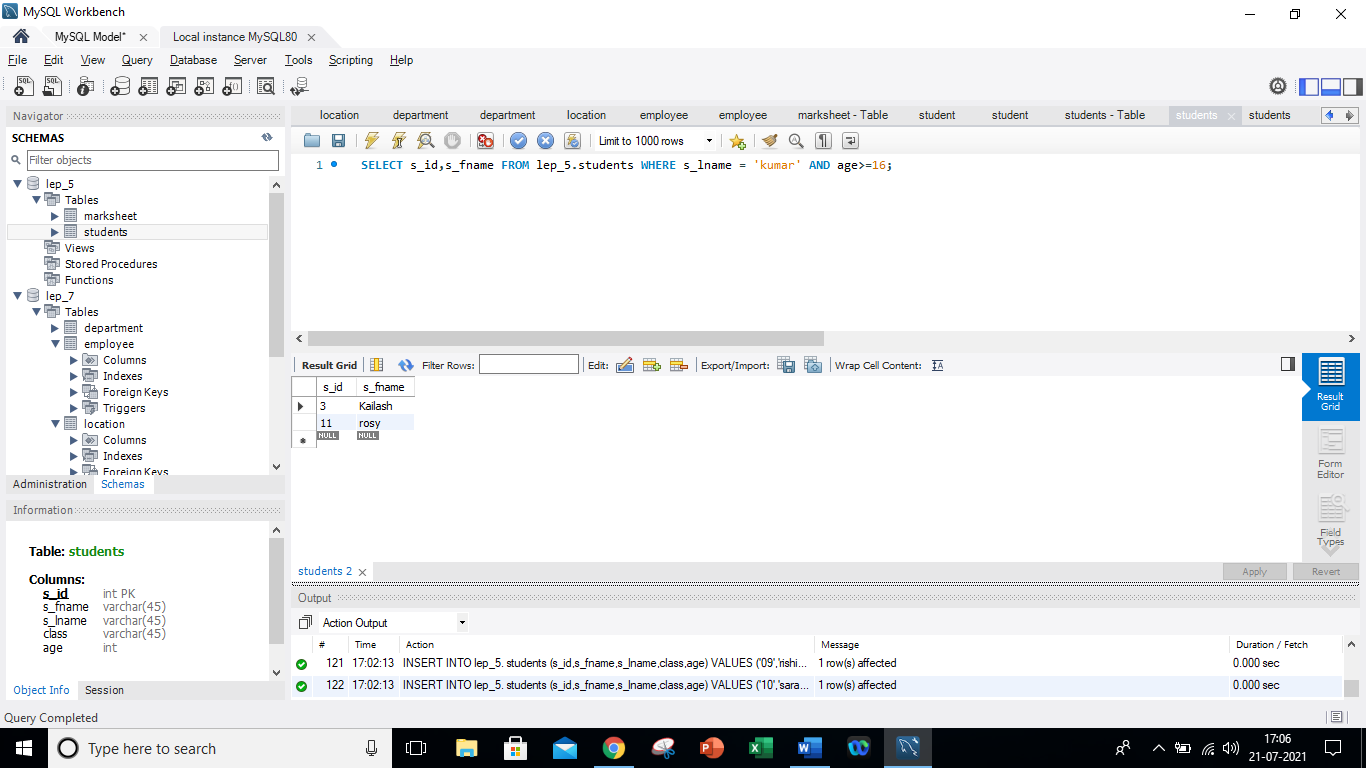
INSERT INTO lep\_5. marksheet (score,year,class,ranking,s\_id) VALUES ('989','2014','10','1','1');

1. 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

**SQL code:**

SELECT s\_id,s\_fname FROM lep\_5.students WHERE s\_lname = 'kumar' AND age>=16;

**Output:**



1. Write a query to display the details of every student from the **marksheet** table whose score is between 800 and 1000

**SQL code:**

SELECT \* FROM lep\_5.marksheet WHERE score BETWEEN 800 AND 1000 ;

**Output:**

Graphical user interface, application

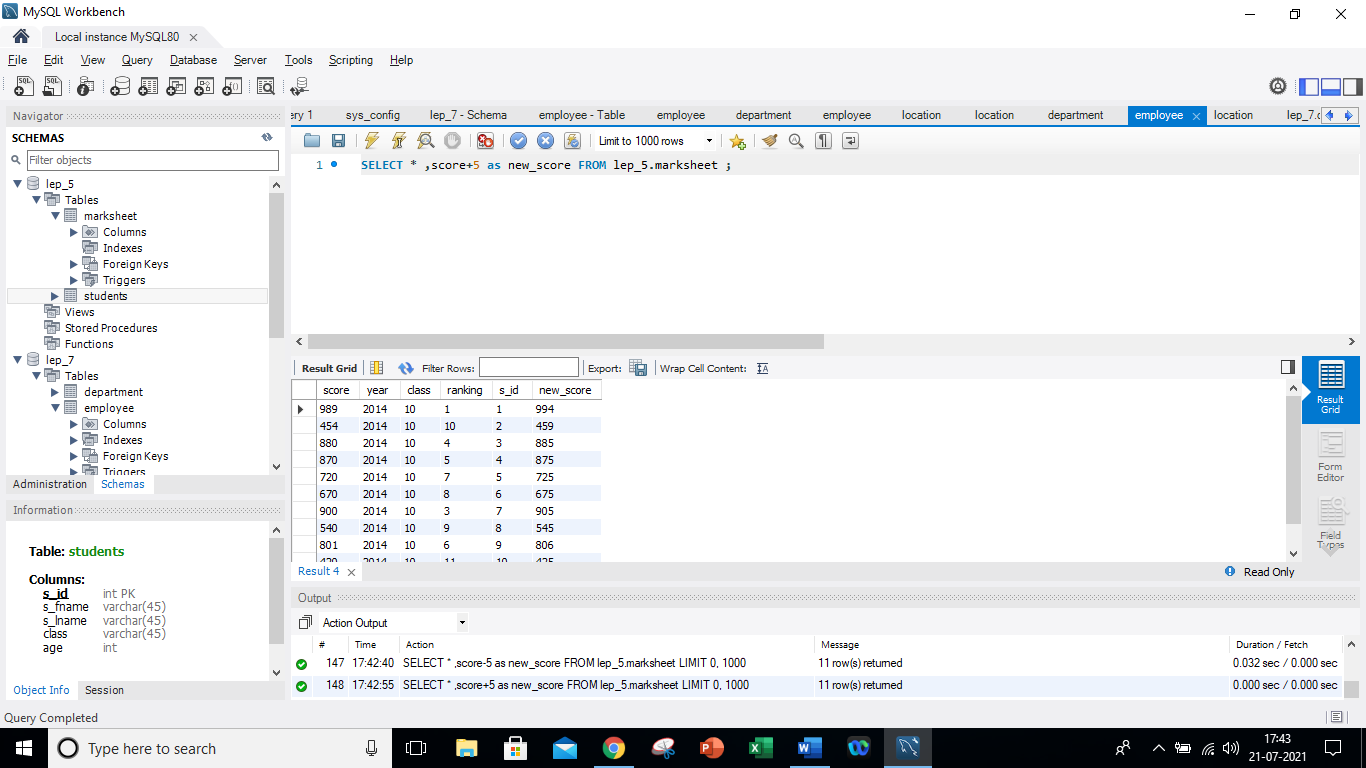
Description automatically generated

1. Write a query to increase the score in the **marksheet** table by five and create a new score column to display this new score

**SQL code:**

SELECT \* ,score+5 as new\_score FROM lep\_5.marksheet ;

**Output:**



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

**SQL code:**

SELECT \* FROM lep\_5.marksheet ORDER BY score DESC ;

**Output:**

A picture containing table

Description automatically generated

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

**SQL code:**

SELECT \* FROM lep\_5.students WHERE s\_fname LIKE 's%';

**Output:**

A screenshot of a computer

Description automatically generated with medium confidence