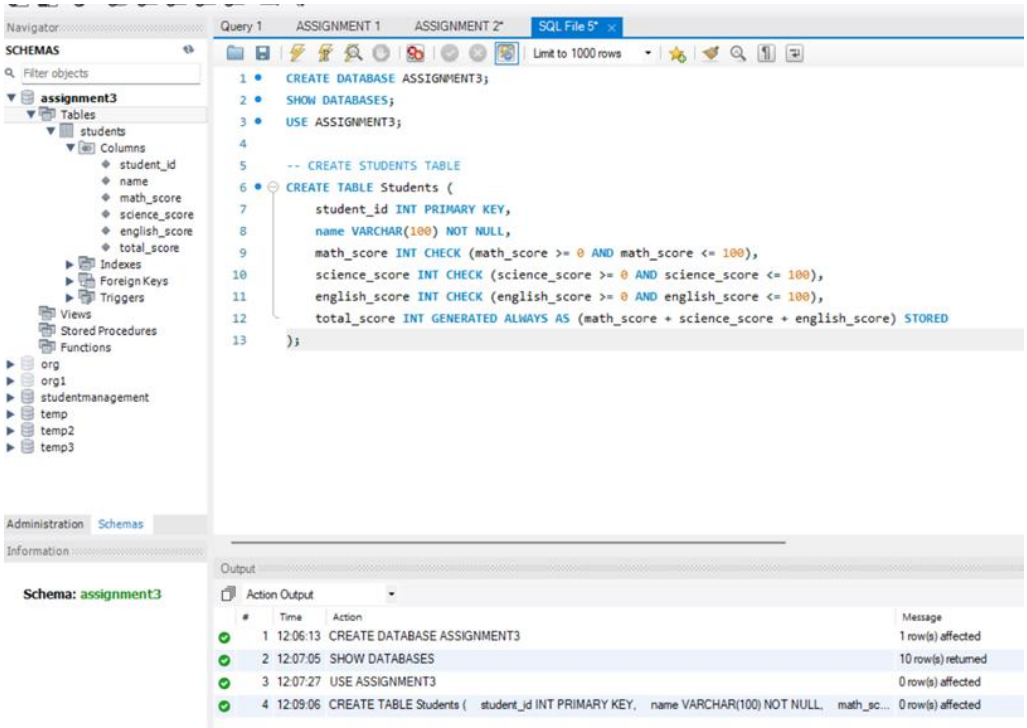
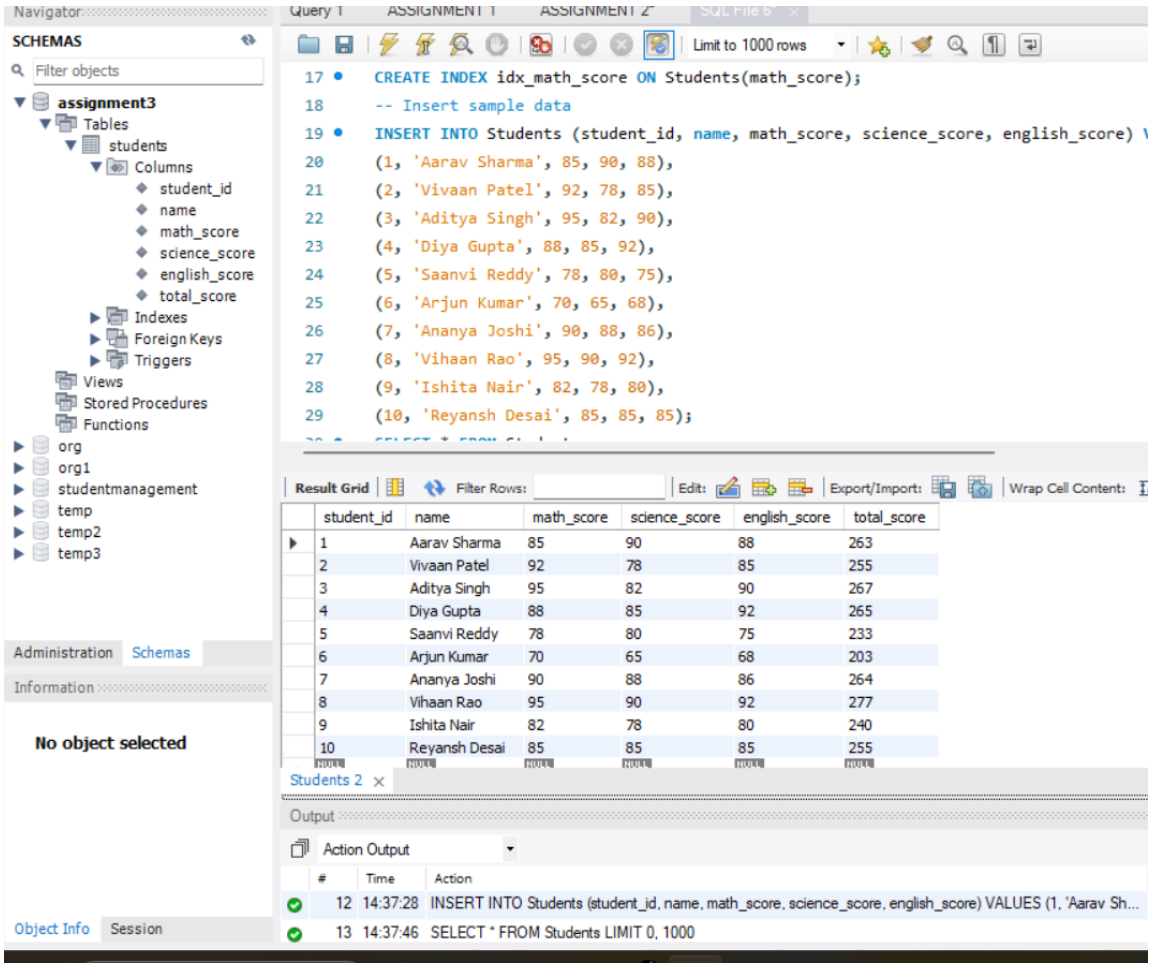
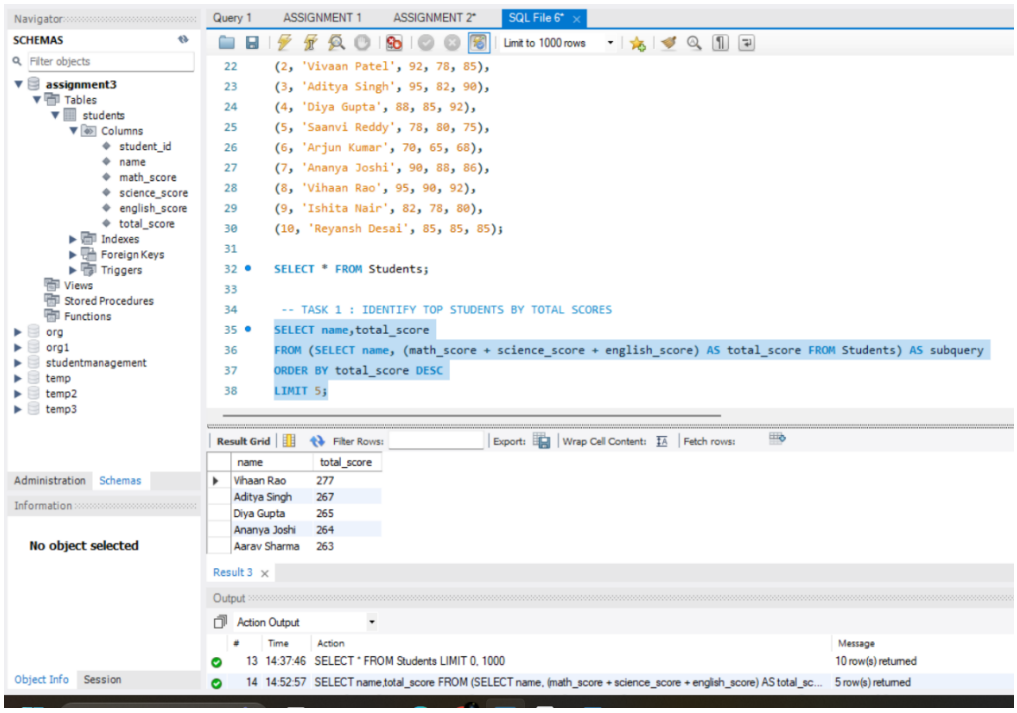
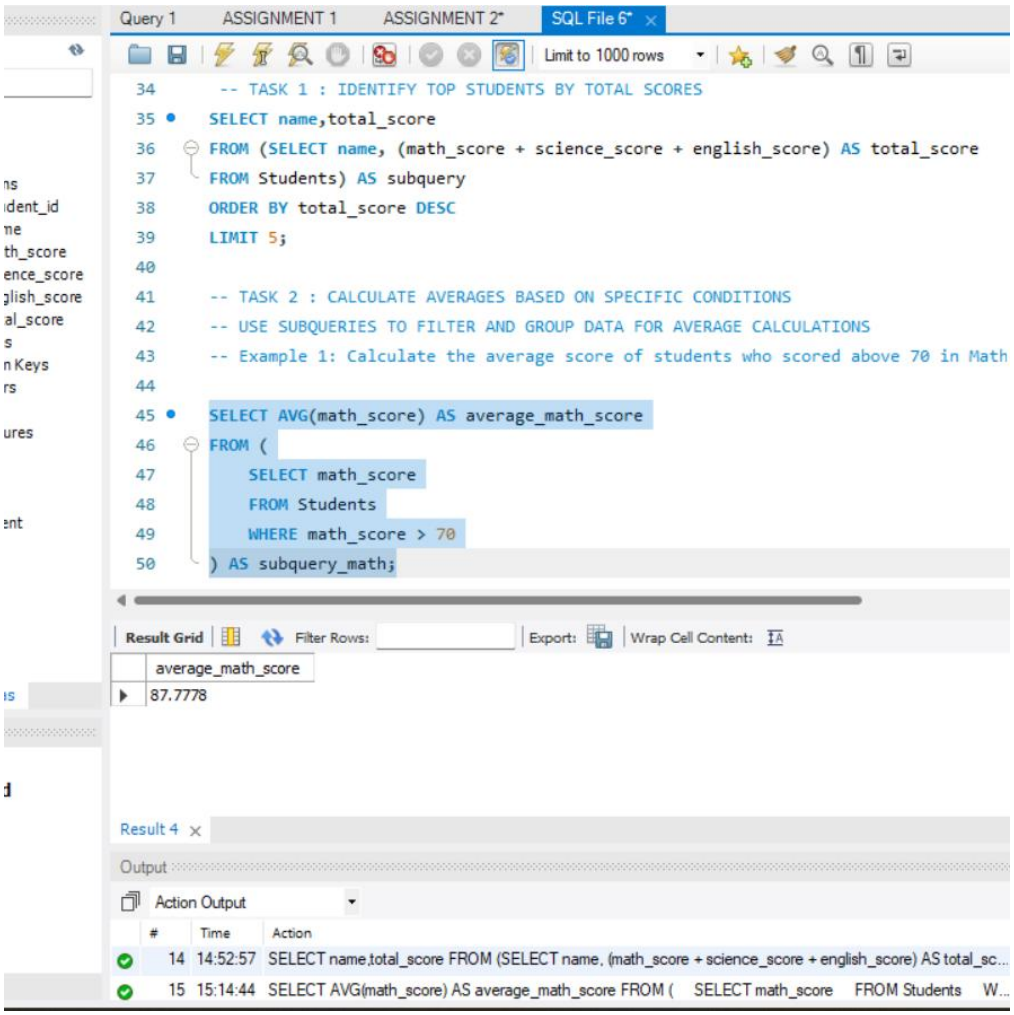
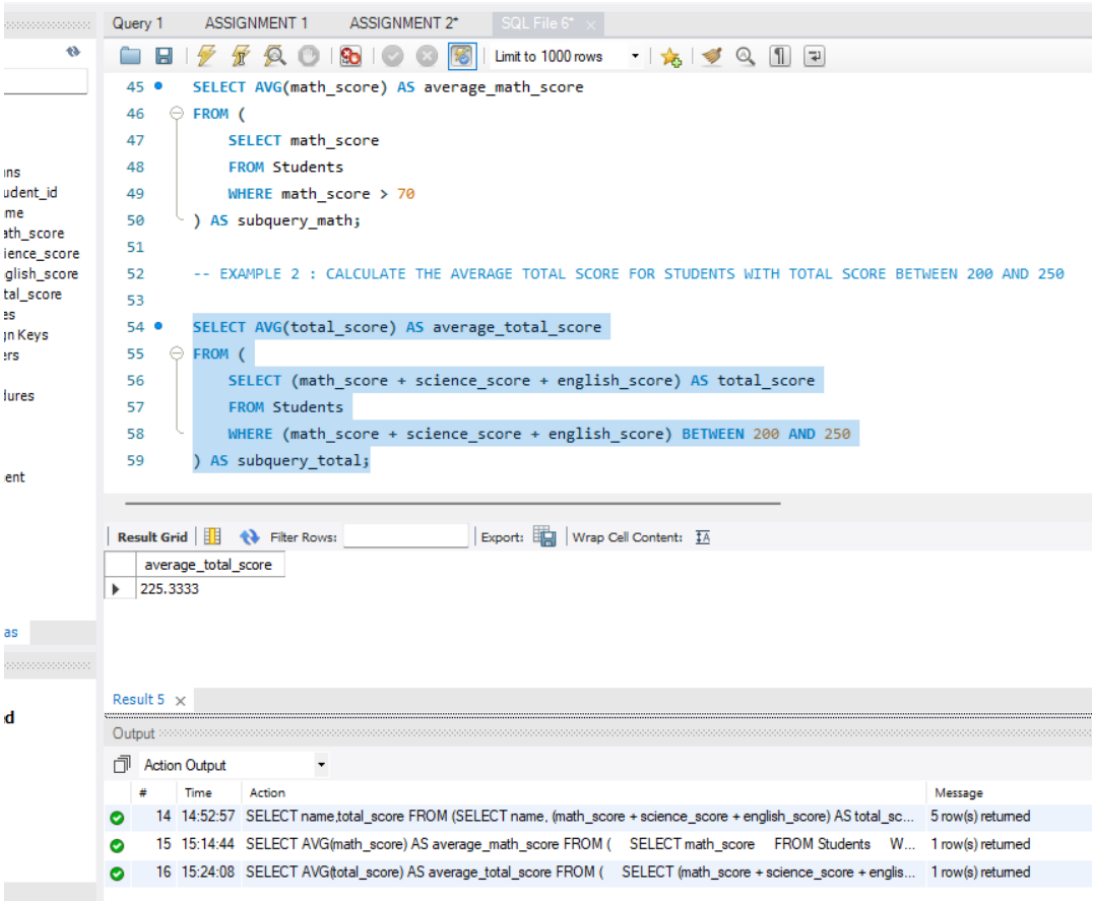
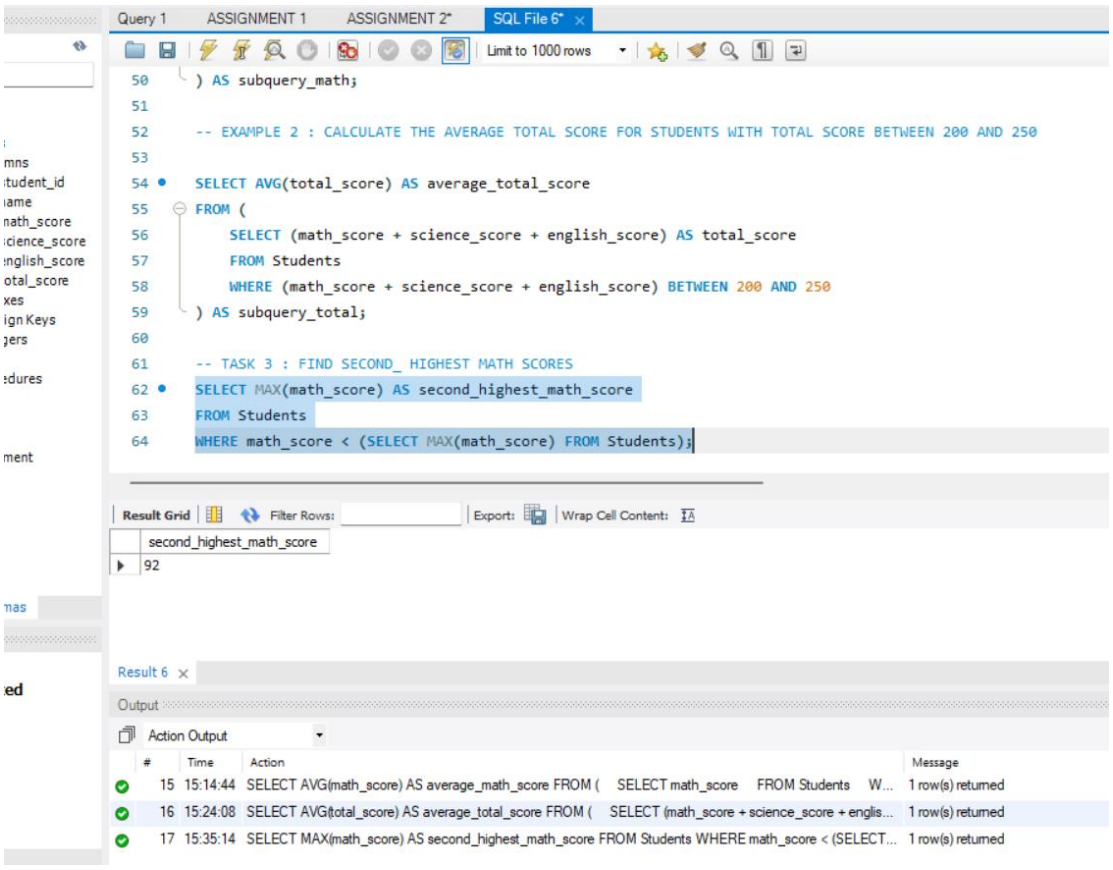

Navigator
SCHEMAS
Filter objects
assignment3
Tables
students
Columns
student_id
name
math_score
science_score
english_score
total_score
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
org
org1
studentmanagement
temp
temp2
temp3
Administration
Schemas
Information
Schema: assignment3
Query 1
ASSIGNMENT 1
ASSIGNMENT 2*
SQL File 5*
Limit to 1000 rows
1 CREATE DATABASE ASSIGNMENT3;
2 SHOW DATABASES;
3 USE ASSIGNMENT3;
4
5 -- CREATE STUDENTS TABLE
6 CREATE TABLE Students (
7 student_id INT PRIMARY KEY,
8 name VARCHAR(100) NOT NULL,
9 math_score INT CHECK (math_score >= 0 AND math_score <= 100),
10 science_score INT CHECK (science_score >= 0 AND science_score <= 100),
11 english_score INT CHECK (english_score >= 0 AND english_score <= 100),
12 total_score INT GENERATED ALWAYS AS (math_score + science_score + english_score) STORED
13);
Output
Action Output
Time Action Message
1 12:06:13 CREATE DATABASE ASSIGNMENT3 1 row(s) affected
2 12:07:05 SHOW DATABASES 10 row(s) returned
3 12:07:27 USE ASSIGNMENT3 0 row(s) affected
4 12:09:06 CREATE TABLE Students (student_id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, math_sc... 0 row(s) affected


Navigator
SCHEMAS
Filter objects
assignment3
Tables
students
Columns
student_id
name
math_score
science_score
english_score
total_score
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
org
org1
studentmanagement
temp
temp2
temp3
Administration
Schemas
Information
No object selected
Object Info
Session
Limit to 1000 rows
17 CREATE INDEX idx_math_score ON Students(math_score);
18 -- Insert sample data
19 INSERT INTO Students (student_id, name, math_score, science_score, english_score) \
20 (1, 'Aarav Sharma', 85, 90, 88),
21 (2, 'Vivaan Patel', 92, 78, 85),
22 (3, 'Aditya Singh', 95, 82, 90),
23 (4, 'Diya Gupta', 88, 85, 92),
24 (5, 'Saanvi Reddy', 78, 80, 75),
25 (6, 'Arjun Kumar', 70, 65, 68),
26 (7, 'Ananya Joshi', 90, 88, 86),
27 (8, 'Vihaan Rao', 95, 90, 92),
28 (9, 'Ishita Nair', 82, 78, 80),
29 (10, 'Reyansh Desai', 85, 85, 85);
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
student_id name math_score science_score english_score total_score
1 Aarav Sharma 85 90 88 263
2 Vivaan Patel 92 78 85 255
3 Aditya Singh 95 82 90 267
4 Diya Gupta 88 85 92 265
5 Saanvi Reddy 78 80 75 233
6 Arjun Kumar 70 65 68 203
7 Ananya Joshi 90 88 86 264
8 Vihaan Rao 95 90 92 277
9 Ishita Nair 82 78 80 240
10 Reyansh Desai 85 85 85 255
Students 2
Output
Action Output
Time Action
12 14:37:28 INSERT INTO Students (student_id, name, math_score, science_score, english_score) VALUES (1, 'Aarav Sh...
13 14:37:46 SELECT * FROM Students LIMIT 0, 1000


Navigator
SCHEMAS
Filter objects
assignment3
Tables
students
Columns
student_id
name
math_score
science_score
english_score
total_score
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
org
org1
studentmanagement
temp
temp2
temp3
Administration
Schemas
Information
No object selected
Object Info
Session
Query 1
ASSIGNMENT 1
ASSIGNMENT 2*
SQL File 6*
Limit to 1000 rows
(2, 'Vivaan Patel', 92, 78, 85),
(3, 'Aditya Singh', 95, 82, 90),
(4, 'Diya Gupta', 88, 85, 92),
(5, 'Saanvi Reddy', 78, 80, 75),
(6, 'Arjun Kumar', 70, 65, 68),
(7, 'Ananya Joshi', 90, 88, 86),
(8, 'Vihaan Rao', 95, 90, 92),
(9, 'Ishita Nair', 82, 78, 80),
(10, 'Reyansh Desai', 85, 85, 85);
SELECT * FROM Students;
-- TASK 1 : IDENTIFY TOP STUDENTS BY TOTAL SCORES
SELECT name,total_score
FROM (SELECT name, (math_score + science_score + english_score) AS total_score FROM Students) AS subquery
ORDER BY total_score DESC
LIMIT 5;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows:
name
total_score
Vihaan Rao 277
Aditya Singh 267
Diya Gupta 265
Ananya Joshi 264
Aarav Sharma 263
Result 3
Output
Action Output
Time Action Message
13 14:37:46 SELECT * FROM Students LIMIT 0, 1000 10 row(s) returned
14 14:52:57 SELECT name,total_score FROM (SELECT name, (math_score + science_score + english_score) AS total_sc... 5 row(s) returned


Query 1
ASSIGNMENT 1
ASSIGNMENT 2*
SQL File 6*
Limit to 1000 rows
-- TASK 1 : IDENTIFY TOP STUDENTS BY TOTAL SCORES
SELECT name,total_score
FROM (SELECT name, (math_score + science_score + english_score) AS total_score
FROM Students) AS subquery
ORDER BY total_score DESC
LIMIT 5;
-- TASK 2 : CALCULATE AVERAGES BASED ON SPECIFIC CONDITIONS
-- USE SUBQUERIES TO FILTER AND GROUP DATA FOR AVERAGE CALCULATIONS
-- Example 1: Calculate the average score of students who scored above 70 in Math
SELECT AVG(math_score) AS average_math_score
FROM (
SELECT math_score
FROM Students
WHERE math_score > 70
) AS subquery_math;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
average_math_score
87.7778
Result 4
Output
Action Output
Time Action
14 14:52:57 SELECT name,total_score FROM (SELECT name, (math_score + science_score + english_score) AS total_sc...
15 15:14:44 SELECT AVG(math_score) AS average_math_score FROM (SELECT math_score FROM Students W...


Query 1
ASSIGNMENT 1
ASSIGNMENT 2*
SQL File 6*
Limit to 1000 rows
SELECT AVG(math_score) AS average_math_score
FROM (
SELECT math_score
FROM Students
WHERE math_score > 70
) AS subquery_math;
-- EXAMPLE 2 : CALCULATE THE AVERAGE TOTAL SCORE FOR STUDENTS WITH TOTAL SCORE BETWEEN 200 AND 250
SELECT AVG(total_score) AS average_total_score
FROM (
SELECT (math_score + science_score + english_score) AS total_score
FROM Students
WHERE (math_score + science_score + english_score) BETWEEN 200 AND 250
) AS subquery_total;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
average_total_score
225.3333
Result 5
Output
Action Output
Time Action Message
14 14:52:57 SELECT name,total_score FROM (SELECT name, (math_score + science_score + english_score) AS total_sc... 5 row(s) returned
15 15:14:44 SELECT AVG(math_score) AS average_math_score FROM (SELECT math_score FROM Students W... 1 row(s) returned
16 15:24:08 SELECT AVG(total_score) AS average_total_score FROM (SELECT (math_score + science_score + englis... 1 row(s) returned


Query 1
ASSIGNMENT 1
ASSIGNMENT 2*
SQL File 6*
Limit to 1000 rows
) AS subquery_math;
-- EXAMPLE 2 : CALCULATE THE AVERAGE TOTAL SCORE FOR STUDENTS WITH TOTAL SCORE BETWEEN 200 AND 250
SELECT AVG(total_score) AS average_total_score
FROM (
SELECT (math_score + science_score + english_score) AS total_score
FROM Students
WHERE (math_score + science_score + english_score) BETWEEN 200 AND 250
) AS subquery_total;
-- TASK 3 : FIND SECOND_ HIGHEST MATH SCORES
SELECT MAX(math_score) AS second_highest_math_score
FROM Students
WHERE math_score < (SELECT MAX(math_score) FROM Students);
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
second_highest_math_score
92
Result 6
Output
Action Output
Time Action Message
15 15:14:44 SELECT AVG(math_score) AS average_math_score FROM (SELECT math_score FROM Students W... 1 row(s) returned
16 15:24:08 SELECT AVG(total_score) AS average_total_score FROM (SELECT (math_score + science_score + englis... 1 row(s) returned
17 15:35:14 SELECT MAX(math_score) AS second_highest_math_score FROM Students WHERE math_score < (SELECT... 1 row(s) returned