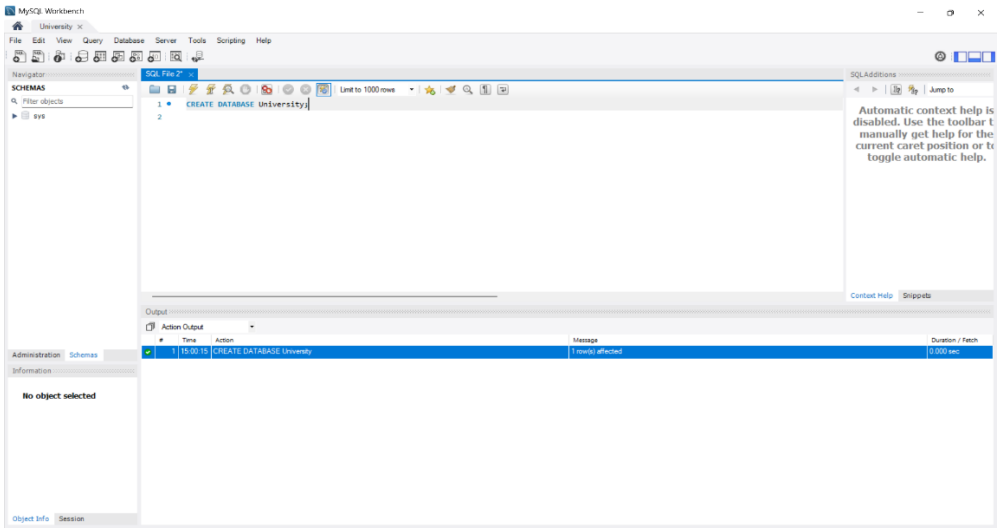
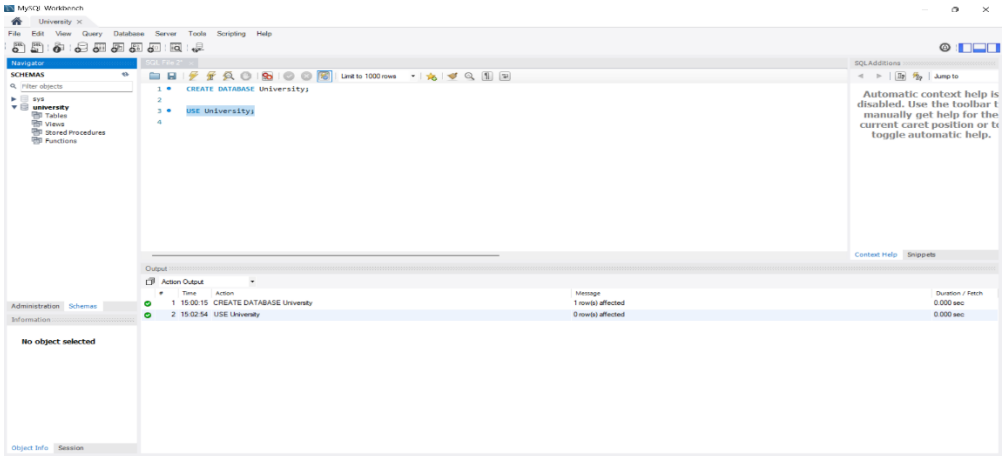
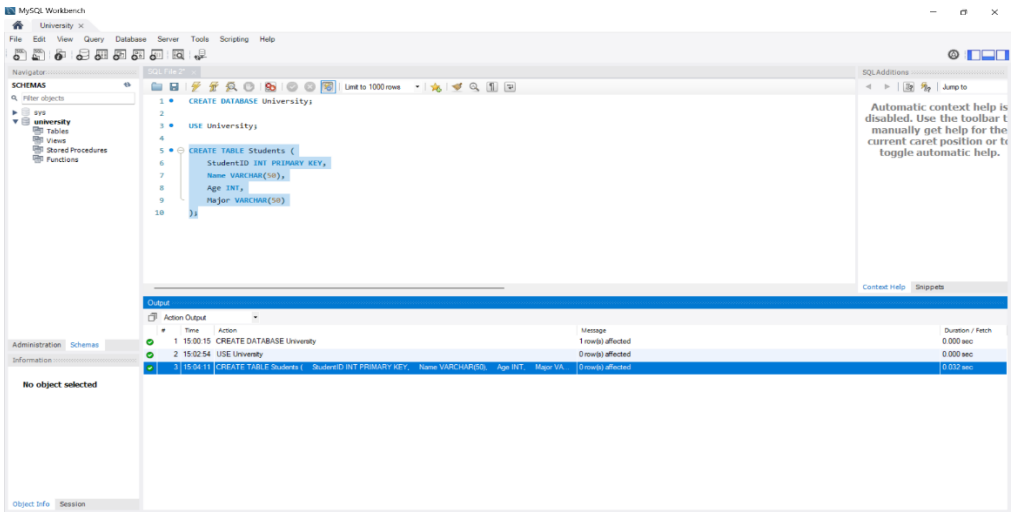
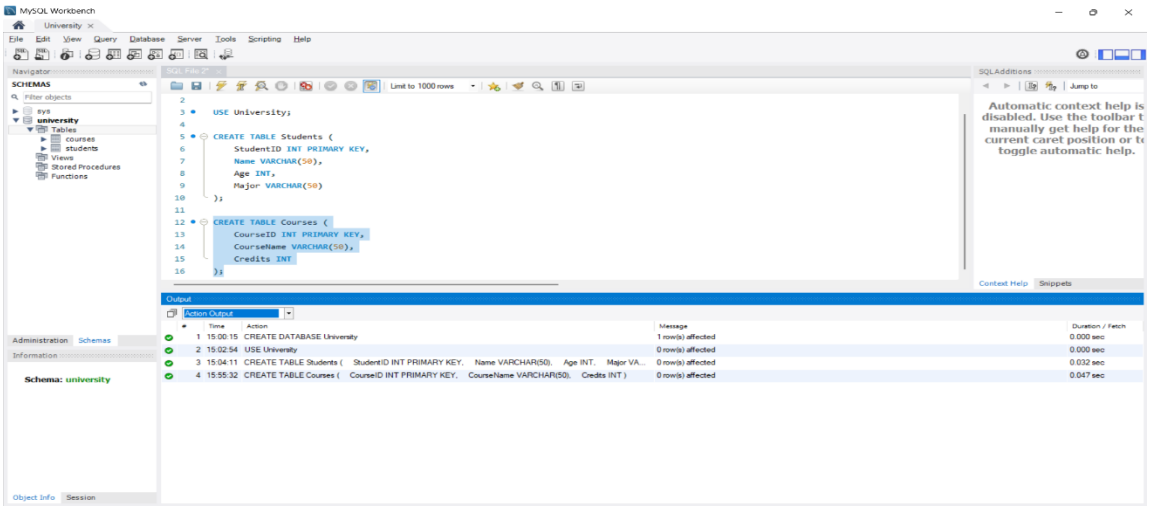
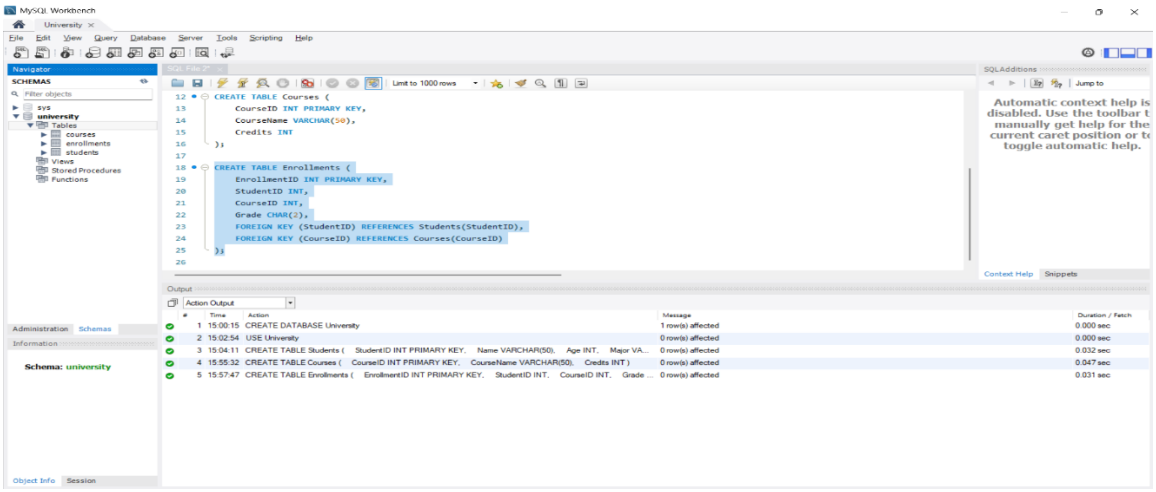
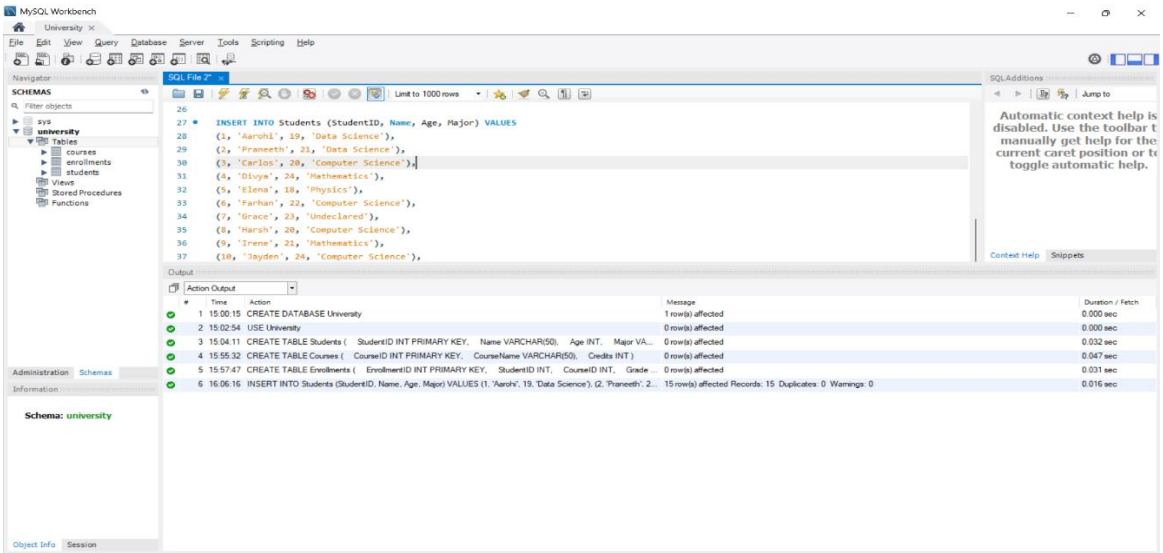

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
SQL File 2*
Limit to 1000 rows
CREATE DATABASE University;
SQL Additions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
Administration Schemas
Information
No object selected
Object Info
Session

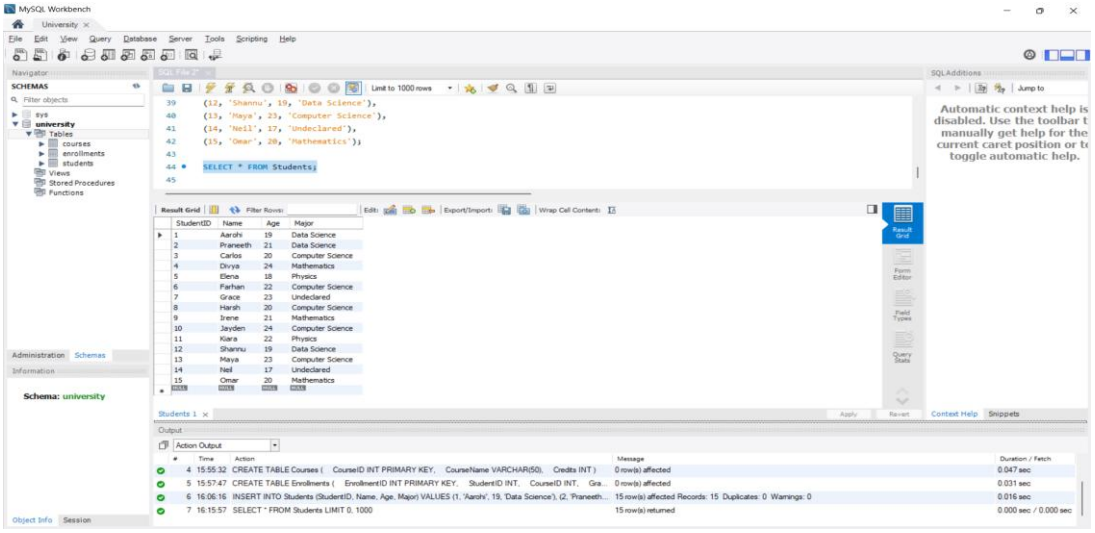

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
Views
Stored Procedures
Functions
SQL File 2*
Limit to 1000 rows
CREATE DATABASE University;
USE University;
SQL Additions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
2 15:02:54 USE University 0 row(s) affected 0.000 sec
Administration Schemas
Information
No object selected
Object Info
Session

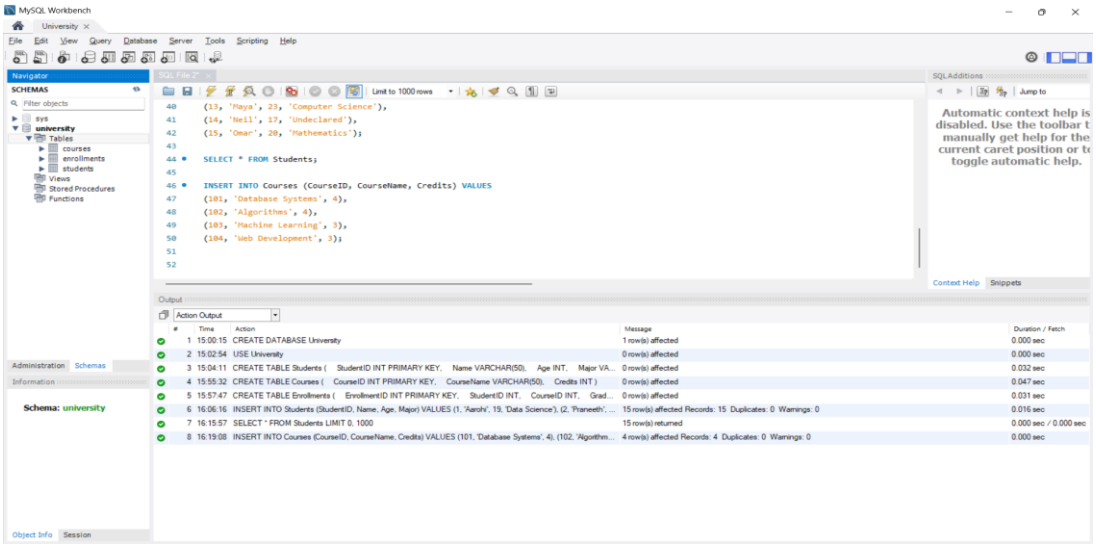

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
Views
Stored Procedures
Functions
SQL File 2*
Limit to 1000 rows
CREATE DATABASE University;
USE University;
CREATE TABLE Students (
StudentID INT PRIMARY KEY,
Name VARCHAR(50),
Age INT,
Major VARCHAR(50)
);
SQL Additions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
2 15:02:54 USE University 0 row(s) affected 0.000 sec
3 15:04:11 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VA... 0 row(s) affected 0.032 sec
Administration Schemas
Information
No object selected
Object Info
Session

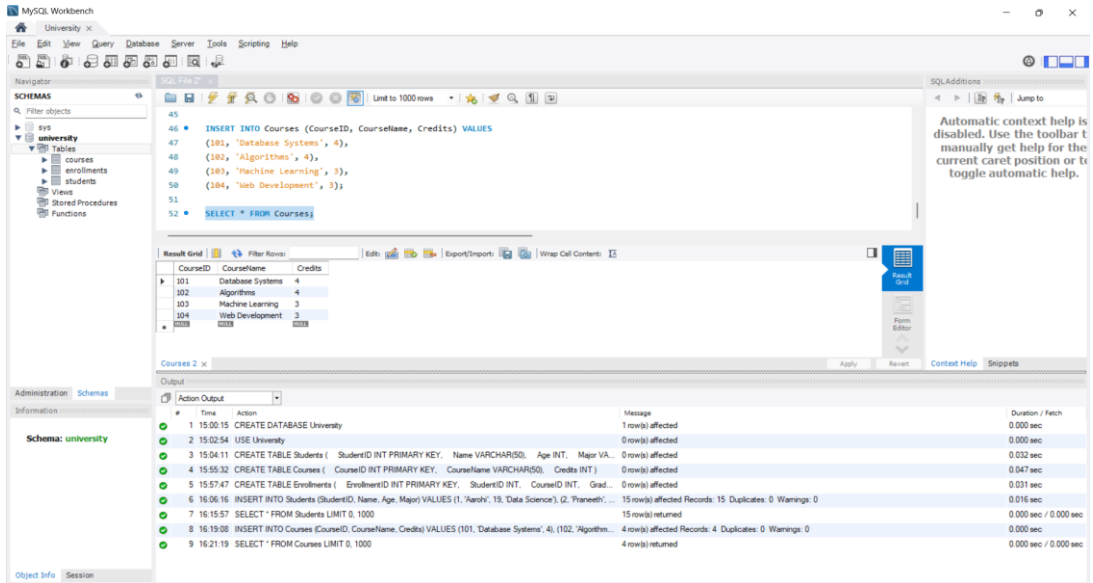

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
USE University;
CREATE TABLE Students (
StudentID INT PRIMARY KEY,
Name VARCHAR(50),
Age INT,
Major VARCHAR(50)
);
CREATE TABLE Courses (
CourseID INT PRIMARY KEY,
CourseName VARCHAR(50),
Credits INT
);
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
2 15:02:54 USE University 0 row(s) affected 0.000 sec
3 15:04:11 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VA... 0 row(s) affected 0.032 sec
4 15:55:32 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT ) 0 row(s) affected 0.047 sec

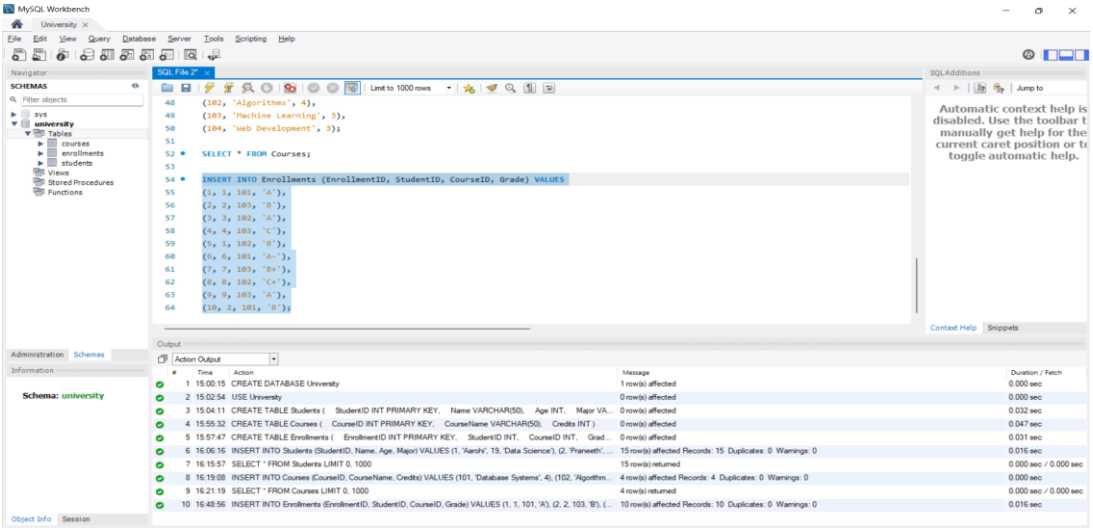

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
CREATE TABLE Courses (
CourseID INT PRIMARY KEY,
CourseName VARCHAR(50),
Credits INT
);
CREATE TABLE Enrollments (
EnrollmentID INT PRIMARY KEY,
StudentID INT,
CourseID INT,
Grade CHAR(2),
FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
);
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
2 15:02:54 USE University 0 row(s) affected 0.000 sec
3 15:04:11 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VA... 0 row(s) affected 0.032 sec
4 15:55:32 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT ) 0 row(s) affected 0.047 sec
5 15:57:47 CREATE TABLE Enrollments ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Grade ... 0 row(s) affected 0.031 sec

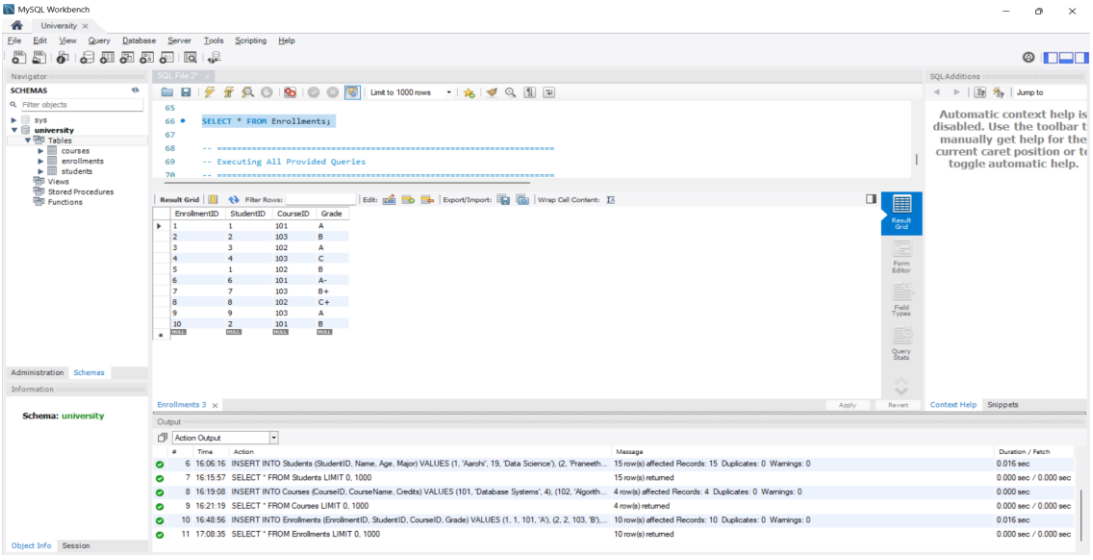

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
INSERT INTO Students (StudentID, Name, Age, Major) VALUES
(1, 'Aarohi', 19, 'Data Science'),
(2, 'Praneeth', 21, 'Data Science'),
(3, 'Carlos', 20, 'Computer Science'),
(4, 'Divya', 24, 'Mathematics'),
(5, 'Elena', 18, 'Physics'),
(6, 'Farhan', 22, 'Computer Science'),
(7, 'Grace', 23, 'Undeclared'),
(8, 'Harsh', 20, 'Computer Science'),
(9, 'Irene', 21, 'Mathematics'),
(10, 'Jayden', 24, 'Computer Science');
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
2 15:02:54 USE University 0 row(s) affected 0.000 sec
3 15:04:11 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VA... 0 row(s) affected 0.032 sec
4 15:55:32 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT ) 0 row(s) affected 0.047 sec
5 15:57:47 CREATE TABLE Enrollments ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Grade ... 0 row(s) affected 0.031 sec
6 16:06:16 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarohi', 19, 'Data Science'), (2, 'Praneeth', 2... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.016 sec

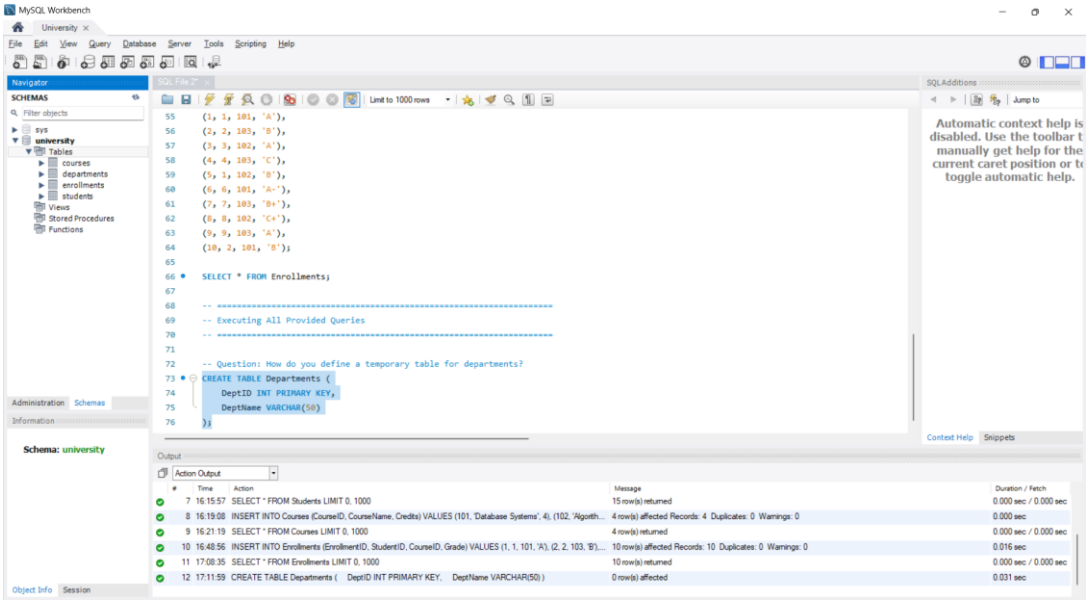

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
39 (12, 'Shannu', 19, 'Data Science'),
40 (13, 'Maya', 23, 'Computer Science'),
41 (14, 'Neil', 17, 'Undeclared'),
42 (15, 'Omar', 20, 'Mathematics');
43
44 • SELECT * FROM Students;
45
Result Grid
Filter Rows:
Edit: Export/Import: Wrap Cell Content:
StudentID Name Age Major
1 Aarohi 19 Data Science
2 Praneeth 21 Data Science
3 Carlos 20 Computer Science
4 Divya 24 Mathematics
5 Elena 18 Physics
6 Farhan 22 Computer Science
7 Grace 23 Undeclared
8 Harsh 20 Computer Science
9 Irene 21 Mathematics
10 Jayden 24 Computer Science
11 Kiara 22 Physics
12 Shannu 19 Data Science
13 Maya 23 Computer Science
14 Neil 17 Undeclared
15 Omar 20 Mathematics
Students 1
Apply Revert
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
4 15:55:32 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT ) 0 row(s) affected 0.047 sec
5 15:57:47 CREATE TABLE Enrollments ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Gra... 0 row(s) affected 0.031 sec
6 16:06:16 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarohi', 19, 'Data Science'), (2, 'Praneeth'... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.016 sec
7 16:15:57 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec

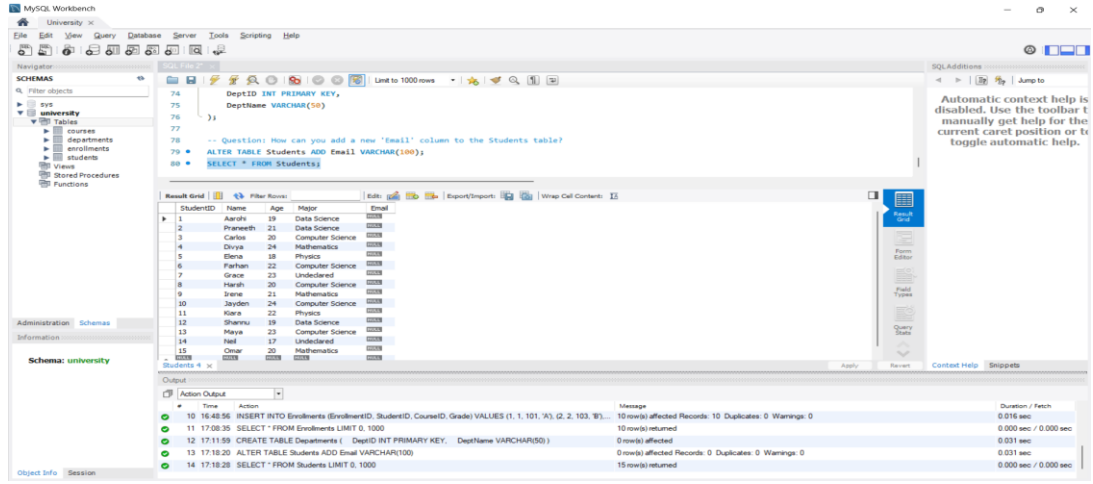

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
40 (13, 'Maya', 23, 'Computer Science'),
41 (14, 'Neil', 17, 'Undeclared'),
42 (15, 'Omar', 20, 'Mathematics');
43
44 • SELECT * FROM Students;
45
46 • INSERT INTO Courses (CourseID, CourseName, Credits) VALUES
47 (101, 'Database Systems', 4),
48 (102, 'Algorithms', 4),
49 (103, 'Machine Learning', 3),
50 (104, 'Web Development', 3);
51
52
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
2 15:02:54 USE University 0 row(s) affected 0.000 sec
3 15:04:11 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VA... 0 row(s) affected 0.032 sec
4 15:55:32 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT ) 0 row(s) affected 0.047 sec
5 15:57:47 CREATE TABLE Enrollments ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Grad... 0 row(s) affected 0.031 sec
6 16:06:16 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarohi', 19, 'Data Science'), (2, 'Praneeth',... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.016 sec
7 16:15:57 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
8 16:19:08 INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorithm... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec

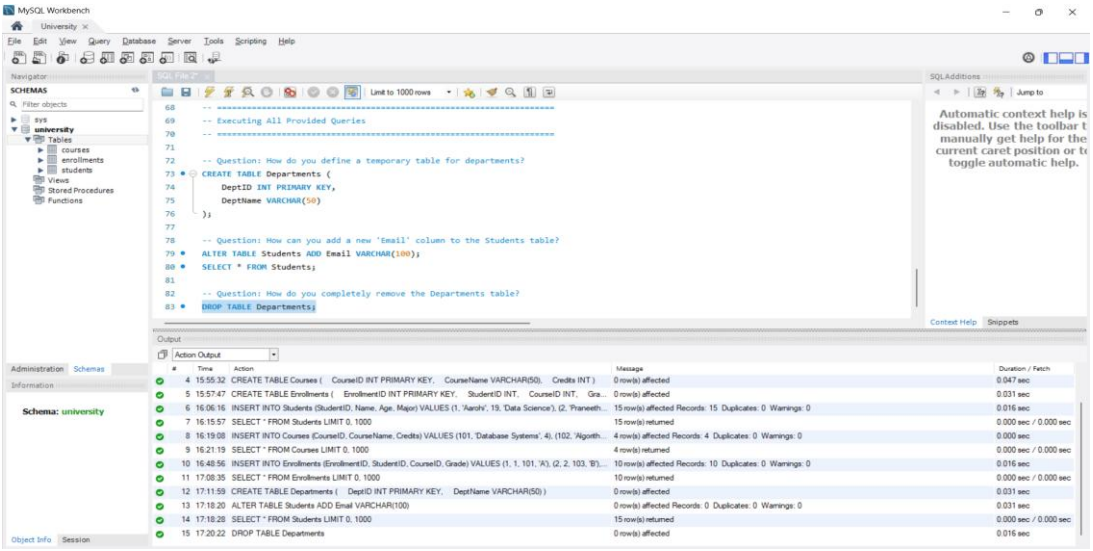

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
45
46 • INSERT INTO Courses (CourseID, CourseName, Credits) VALUES
47 (101, 'Database Systems', 4),
48 (102, 'Algorithms', 4),
49 (103, 'Machine Learning', 3),
50 (104, 'Web Development', 3);
51
52 • SELECT * FROM Courses;
Result Grid
Filter Rows:
Edit: Export/Import: Wrap Cell Content:
CourseID CourseName Credits
101 Database Systems 4
102 Algorithms 4
103 Machine Learning 3
104 Web Development 3
Courses 2
Apply Revert
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
2 15:02:54 USE University 0 row(s) affected 0.000 sec
3 15:04:11 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VA... 0 row(s) affected 0.032 sec
4 15:55:32 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT ) 0 row(s) affected 0.047 sec
5 15:57:47 CREATE TABLE Enrollments ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Grad... 0 row(s) affected 0.031 sec
6 16:06:16 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarohi', 19, 'Data Science'), (2, 'Praneeth',... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.016 sec
7 16:15:57 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
8 16:19:08 INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorithm... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec
9 16:21:19 SELECT * FROM Courses LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec

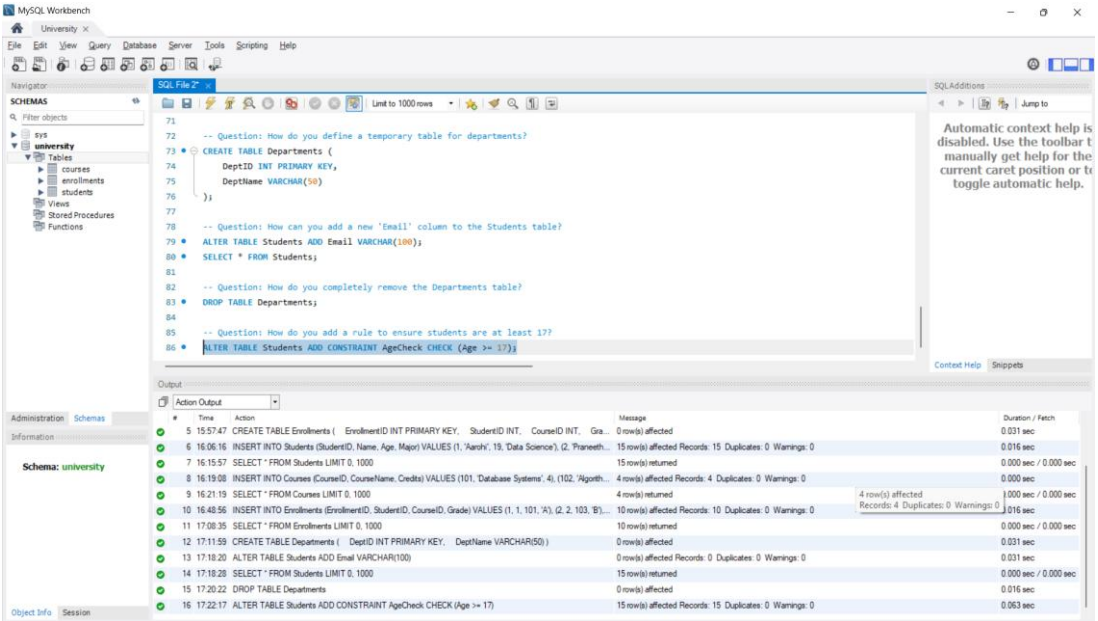

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
48 (102, 'Algorithms', 4),
49 (103, 'Machine Learning', 3),
50 (104, 'Web Development', 3);
51
52 • SELECT * FROM Courses;
53
54 • INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES
55 (1, 1, 101, 'A'),
56 (2, 2, 103, 'B'),
57 (3, 3, 102, 'A'),
58 (4, 4, 103, 'C'),
59 (5, 1, 102, 'B'),
60 (6, 6, 101, 'A-'),
61 (7, 7, 103, 'B+'),
62 (8, 8, 102, 'C+'),
63 (9, 9, 103, 'A'),
64 (10, 2, 101, 'B');
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 15:00:15 CREATE DATABASE University 1 row(s) affected 0.000 sec
2 15:02:54 USE University 0 row(s) affected 0.000 sec
3 15:04:11 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VA... 0 row(s) affected 0.032 sec
4 15:55:32 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT ) 0 row(s) affected 0.047 sec
5 15:57:47 CREATE TABLE Enrollments ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Grad... 0 row(s) affected 0.031 sec
6 16:06:16 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarohi', 19, 'Data Science'), (2, 'Praneeth',... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.016 sec
7 16:15:57 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
8 16:19:08 INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorithm... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec
9 16:21:19 SELECT * FROM Courses LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
10 16:48:56 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (1, 1, 101, 'A'), (2, 2, 103, 'B'), (... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.016 sec

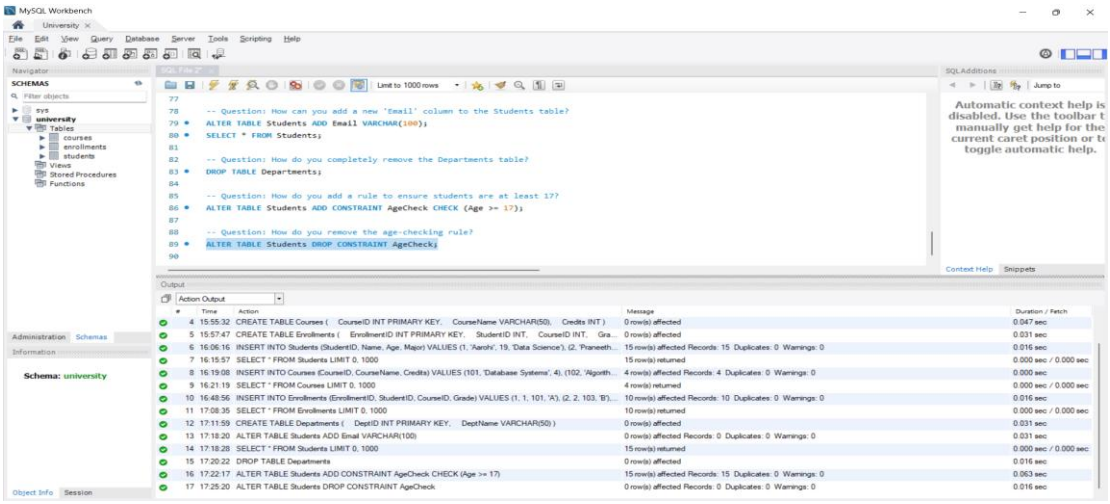

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
65
66 • SELECT * FROM Enrollments;
67
68 -- ================================================================
69 -- Executing All Provided Queries
70 -- ================================================================
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
EnrollmentID StudentID CourseID Grade
1 1 101 A
2 2 103 B
3 3 102 A
4 4 103 C
5 1 102 B
6 6 101 A-
7 7 103 B+
8 8 102 C+
9 9 103 A
10 2 101 B
Enrollments 3
Apply
Revert
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
6 16:06:16 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarohi', 19, 'Data Science'), (2, 'Praneeth... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.016 sec
7 16:15:57 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
8 16:19:08 INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorith... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec
9 16:21:19 SELECT * FROM Courses LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
10 16:48:56 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (1, 1, 101, 'A'), (2, 2, 103, 'B'),... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.016 sec
11 17:08:35 SELECT * FROM Enrollments LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec

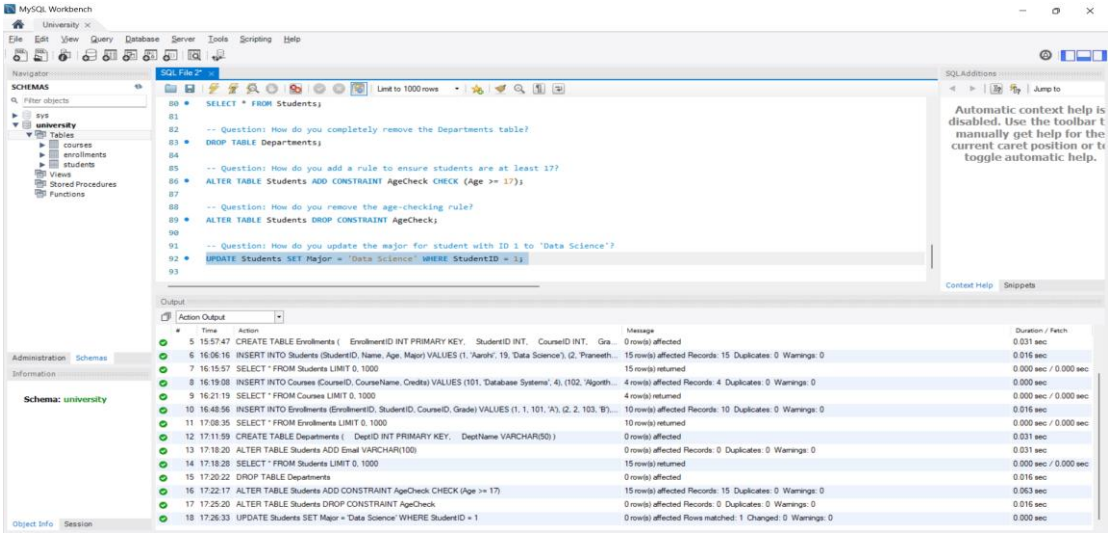

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
departments
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Limit to 1000 rows
55 (1, 1, 101, 'A'),
56 (2, 2, 103, 'B'),
57 (3, 3, 102, 'A'),
58 (4, 4, 103, 'C'),
59 (5, 1, 102, 'B'),
60 (6, 6, 101, 'A-'),
61 (7, 7, 103, 'B+'),
62 (8, 8, 102, 'C+'),
63 (9, 9, 103, 'A'),
64 (10, 2, 101, 'B');
65
66 • SELECT * FROM Enrollments;
67
68 -- ================================================================
69 -- Executing All Provided Queries
70 -- ================================================================
71
72 -- Question: How do you define a temporary table for departments?
73 • CREATE TABLE Departments (
74 DeptID INT PRIMARY KEY,
75 DeptName VARCHAR(50)
76 );
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
7 16:15:57 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
8 16:19:08 INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorith... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec
9 16:21:19 SELECT * FROM Courses LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
10 16:48:56 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (1, 1, 101, 'A'), (2, 2, 103, 'B'),... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.016 sec
11 17:08:35 SELECT * FROM Enrollments LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
12 17:11:59 CREATE TABLE Departments ( DeptID INT PRIMARY KEY, DeptName VARCHAR(50) ) 0 row(s) affected 0.031 sec

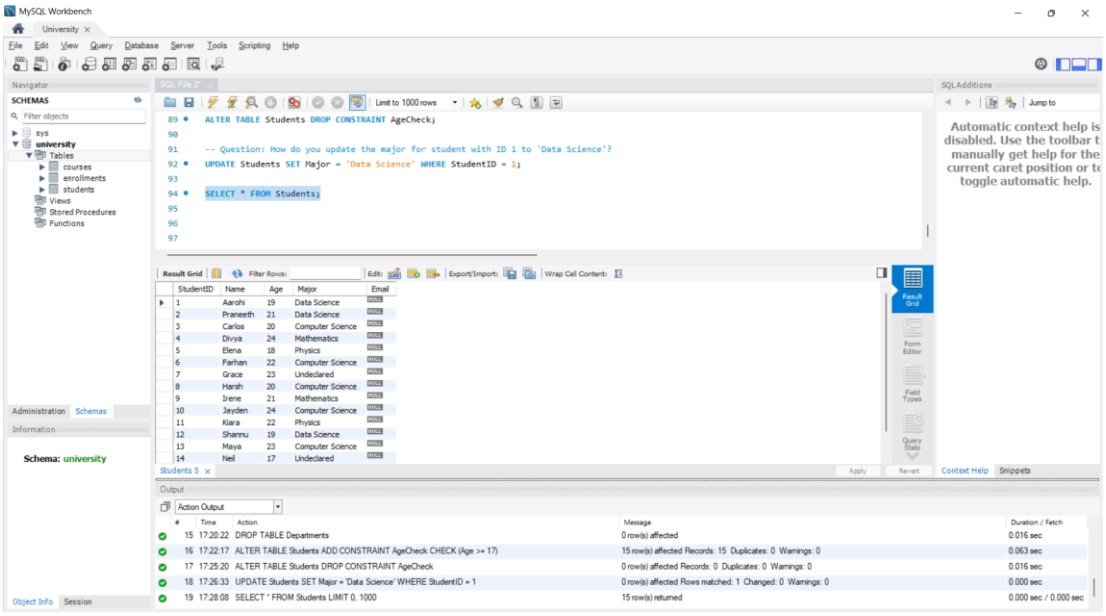

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
university
Tables
courses
departments
enrollments
students
Views
Stored Procedures
Functions
Schema: university
SQL File 2*
Limit to 1000 rows
DeptID INT PRIMARY KEY,
DeptName VARCHAR(50)
);
-- Question: How can you add a new 'Email' column to the Students table?
ALTER TABLE Students ADD Email VARCHAR(100);
SELECT * FROM Students;
Result Grid
StudentID Name Age Major Email
1 Aarohi 19 Data Science
2 Praneeth 21 Data Science
3 Carlos 20 Computer Science
4 Divya 24 Mathematics
5 Elena 18 Physics
6 Farhan 22 Computer Science
7 Grace 23 Undeclared
8 Harsh 20 Computer Science
9 Irene 21 Mathematics
10 Jayden 24 Computer Science
11 Kiara 22 Physics
12 Shannu 19 Data Science
13 Maya 23 Computer Science
14 Neil 17 Undeclared
15 Omar 20 Mathematics
Output
Action Output
10 16:48:56 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (1, 1, 101, 'A'), (2, 2, 103, 'B')... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.016 sec
11 17:08:35 SELECT * FROM Enrollments LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
12 17:11:59 CREATE TABLE Departments ( DeptID INT PRIMARY KEY, DeptName VARCHAR(50) ) 0 row(s) affected 0.031 sec
13 17:18:20 ALTER TABLE Students ADD Email VARCHAR(100) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.031 sec
14 17:18:28 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec

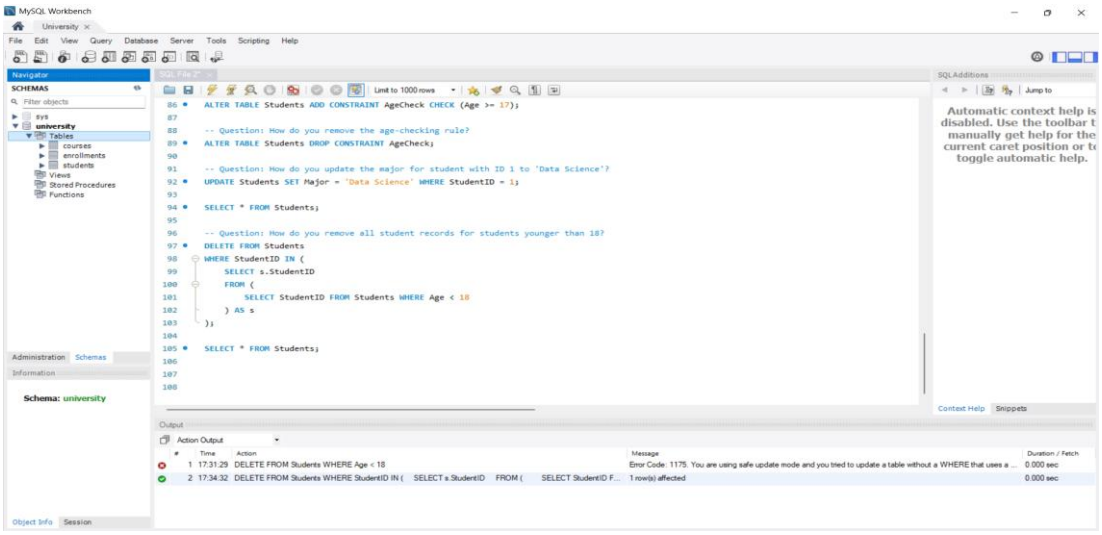

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Schema: university
-- Executing All Provided Queries
-- Question: How do you define a temporary table for departments?
CREATE TABLE Departments (
DeptID INT PRIMARY KEY,
DeptName VARCHAR(50)
);
-- Question: How can you add a new 'Email' column to the Students table?
ALTER TABLE Students ADD Email VARCHAR(100);
SELECT * FROM Students;
-- Question: How do you completely remove the Departments table?
DROP TABLE Departments;
Output
Action Output
4 15:55:32 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT ) 0 row(s) affected 0.047 sec
5 15:57:47 CREATE TABLE Enrollments ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Gra... 0 row(s) affected 0.031 sec
6 16:06:16 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarohi', 19, 'Data Science'), (2, 'Praneeth... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.016 sec
7 16:15:57 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
8 16:19:08 INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorith... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec
9 16:21:19 SELECT * FROM Courses LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
10 16:48:56 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (1, 1, 101, 'A'), (2, 2, 103, 'B')... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.016 sec
11 17:08:35 SELECT * FROM Enrollments LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
12 17:11:59 CREATE TABLE Departments ( DeptID INT PRIMARY KEY, DeptName VARCHAR(50) ) 0 row(s) affected 0.031 sec
13 17:18:20 ALTER TABLE Students ADD Email VARCHAR(100) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.031 sec
14 17:18:28 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
15 17:20:22 DROP TABLE Departments 0 row(s) affected 0.016 sec

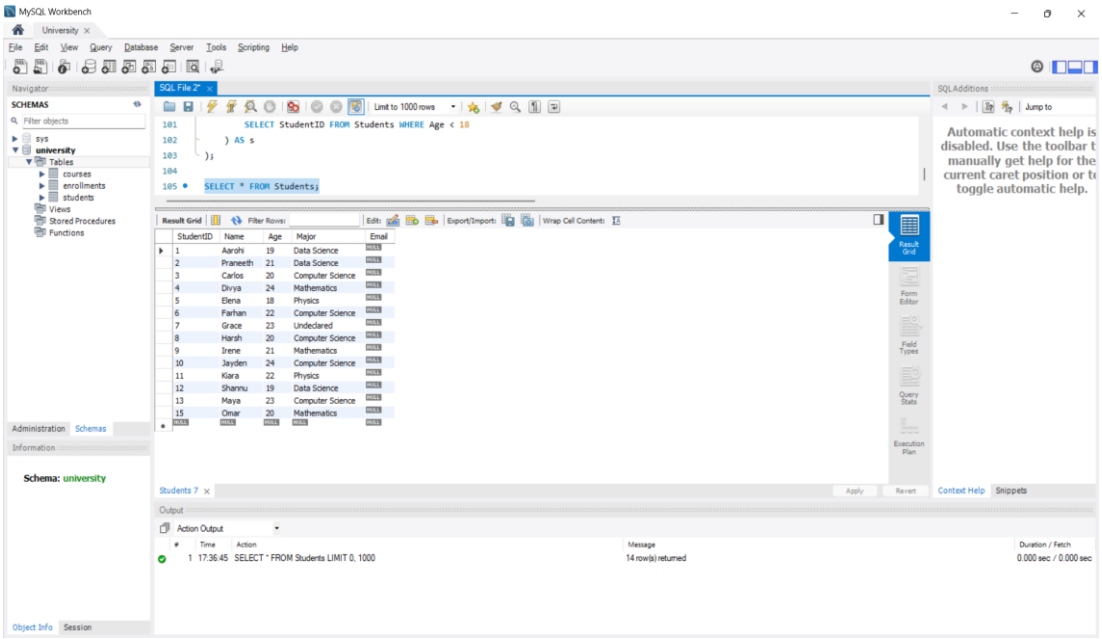

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Schema: university
SQL File 2*
-- Question: How do you define a temporary table for departments?
CREATE TABLE Departments (
DeptID INT PRIMARY KEY,
DeptName VARCHAR(50)
);
-- Question: How can you add a new 'Email' column to the Students table?
ALTER TABLE Students ADD Email VARCHAR(100);
SELECT * FROM Students;
-- Question: How do you completely remove the Departments table?
DROP TABLE Departments;
-- Question: How do you add a rule to ensure students are at least 17?
ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);
Output
Action Output
5 15:57:47 CREATE TABLE Enrollments ( EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Gra... 0 row(s) affected 0.031 sec
6 16:06:16 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarohi', 19, 'Data Science'), (2, 'Praneeth... 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.016 sec
7 16:15:57 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
8 16:19:08 INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorith... 4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0 0.000 sec
9 16:21:19 SELECT * FROM Courses LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
10 16:48:56 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (1, 1, 101, 'A'), (2, 2, 103, 'B')... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.016 sec
11 17:08:35 SELECT * FROM Enrollments LIMIT 0, 1000 10 row(s) returned 0.000 sec / 0.000 sec
12 17:11:59 CREATE TABLE Departments ( DeptID INT PRIMARY KEY, DeptName VARCHAR(50) ) 0 row(s) affected 0.031 sec
13 17:18:20 ALTER TABLE Students ADD Email VARCHAR(100) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.031 sec
14 17:18:28 SELECT * FROM Students LIMIT 0, 1000 15 row(s) returned 0.000 sec / 0.000 sec
15 17:20:22 DROP TABLE Departments 0 row(s) affected 0.016 sec
16 17:22:17 ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17) 15 row(s) affected Records: 15 Duplicates: 0 Warnings: 0 0.063 sec

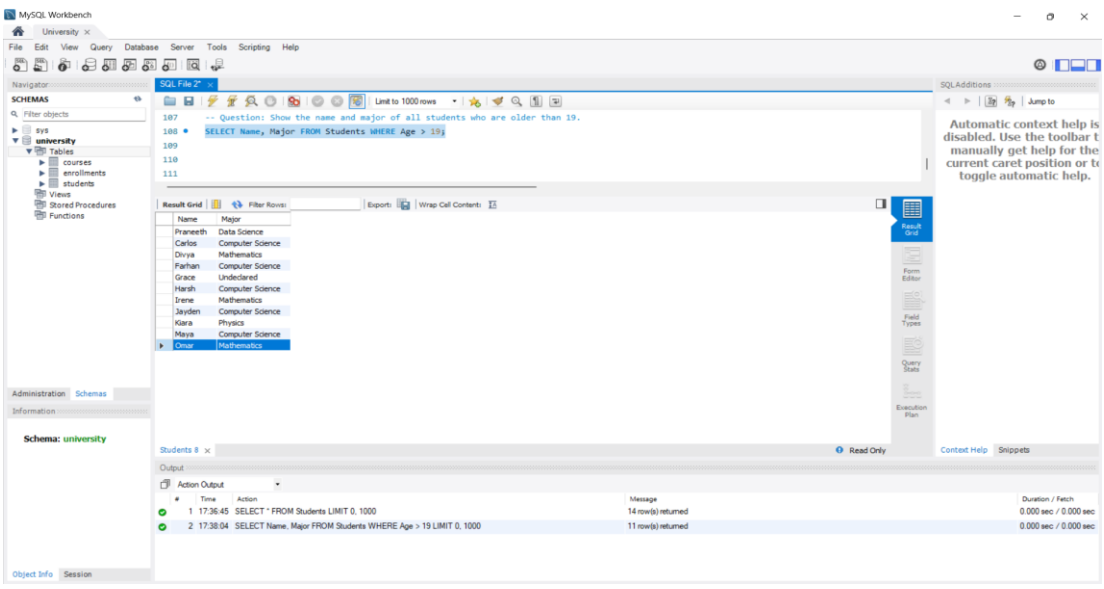

MySQL Workbench
-- Question: How can you add a new 'Email' column to the Students table?
ALTER TABLE Students ADD Email VARCHAR(100);
SELECT * FROM Students;
-- Question: How do you completely remove the Departments table?
DROP TABLE Departments;
-- Question: How do you add a rule to ensure students are at least 17?
ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);
-- Question: How do you remove the age-checking rule?
ALTER TABLE Students DROP CONSTRAINT AgeCheck;
Schema: university

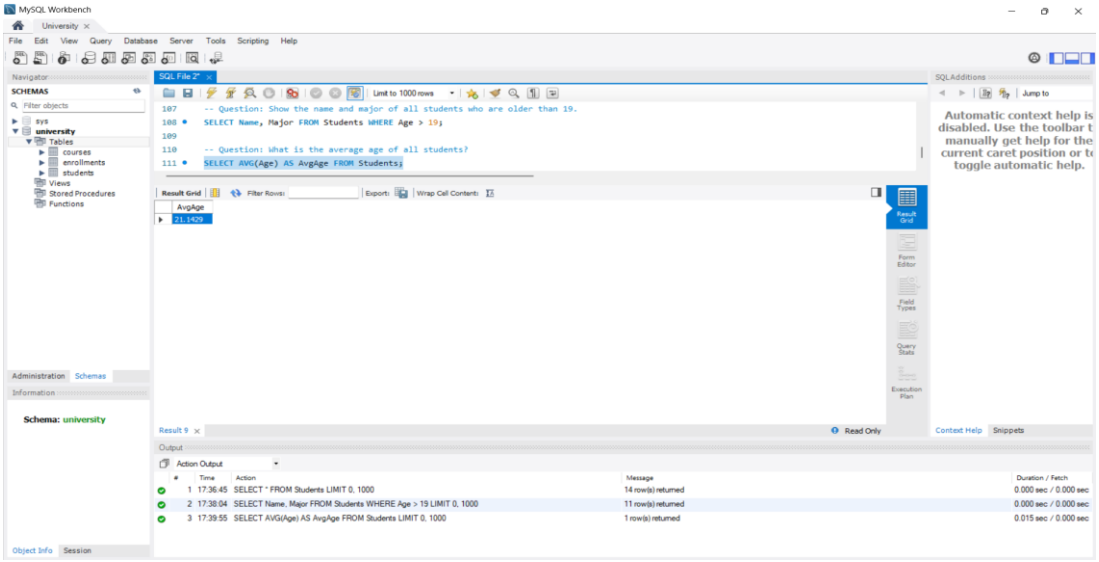

MySQL Workbench
SELECT * FROM Students;
-- Question: How do you completely remove the Departments table?
DROP TABLE Departments;
-- Question: How do you add a rule to ensure students are at least 17?
ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);
-- Question: How do you remove the age-checking rule?
ALTER TABLE Students DROP CONSTRAINT AgeCheck;
-- Question: How do you update the major for student with ID 1 to 'Data Science'?
UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;
Schema: university

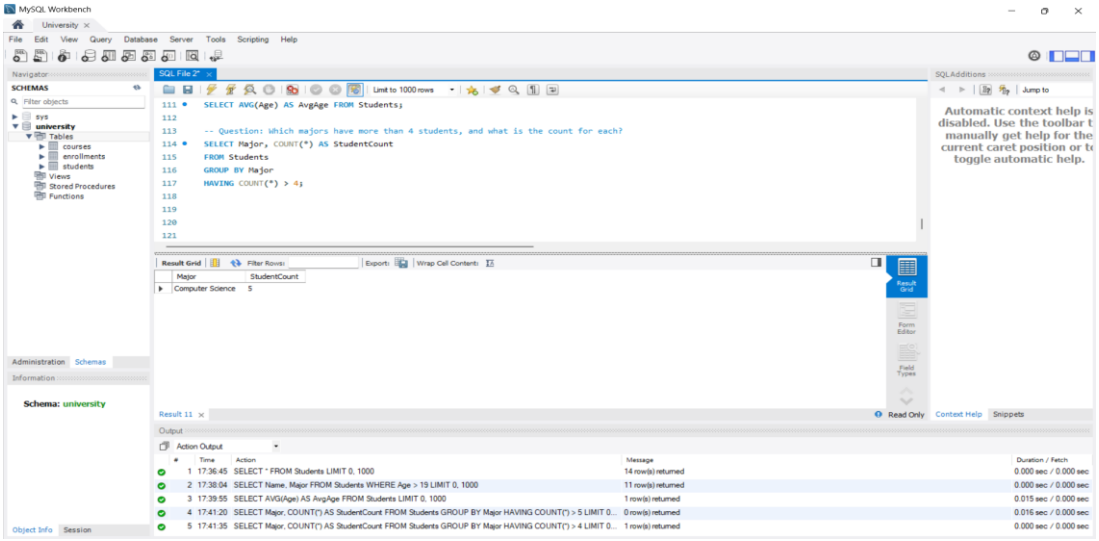

MySQL Workbench
ALTER TABLE Students DROP CONSTRAINT AgeCheck;
-- Question: How do you update the major for student with ID 1 to 'Data Science'?
UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;
SELECT * FROM Students;
Schema: university

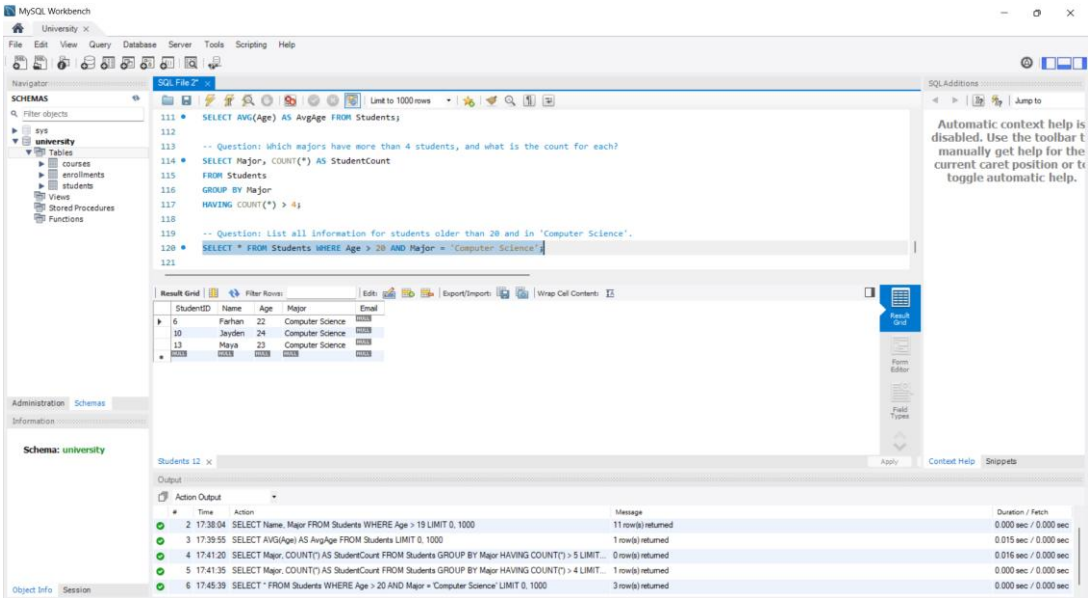

MySQL Workbench
ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);
-- Question: How do you remove the age-checking rule?
ALTER TABLE Students DROP CONSTRAINT AgeCheck;
-- Question: How do you update the major for student with ID 1 to 'Data Science'?
UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;
SELECT * FROM Students;
-- Question: How do you remove all student records for students younger than 18?
DELETE FROM Students
WHERE StudentID IN (
SELECT s.StudentID
FROM (
SELECT StudentID FROM Students WHERE Age < 18
) AS s
);
SELECT * FROM Students;
Schema: university
1 17:31:29 DELETE FROM Students WHERE Age < 18
Error Code: 1175. You are using safe update mode and you tried to update a table without a WHERE that uses a ...
2 17:34:32 DELETE FROM Students WHERE StudentID IN ( SELECT s.StudentID FROM ( SELECT StudentID F...
1 row(s) affected

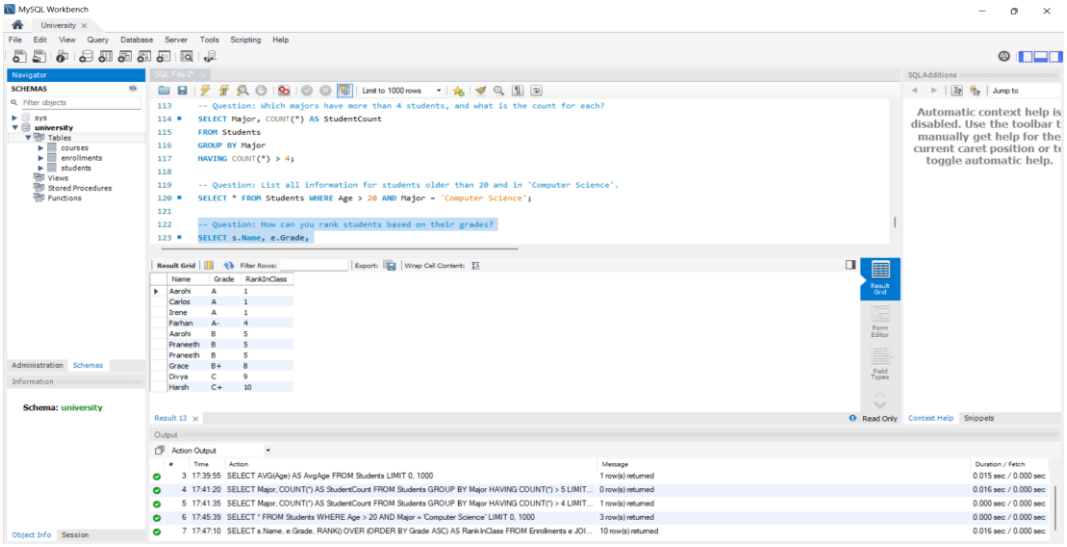

MySQL Workbench
SELECT StudentID FROM Students WHERE Age < 18
) AS s
);
SELECT * FROM Students;
StudentID Name Age Major Email
1 Aarohi 19 Data Science
2 Praneeth 21 Data Science
3 Carlos 20 Computer Science
4 Divya 24 Mathematics
5 Elena 18 Physics
6 Farhan 22 Computer Science
7 Grace 23 Undeclared
8 Harsh 20 Computer Science
9 Irene 21 Mathematics
10 Jayden 24 Computer Science
11 Kiara 22 Physics
12 Shannu 19 Data Science
13 Maya 23 Computer Science
15 Omar 20 Mathematics
Schema: university
1 17:36:45 SELECT * FROM Students LIMIT 0, 1000
14 row(s) returned

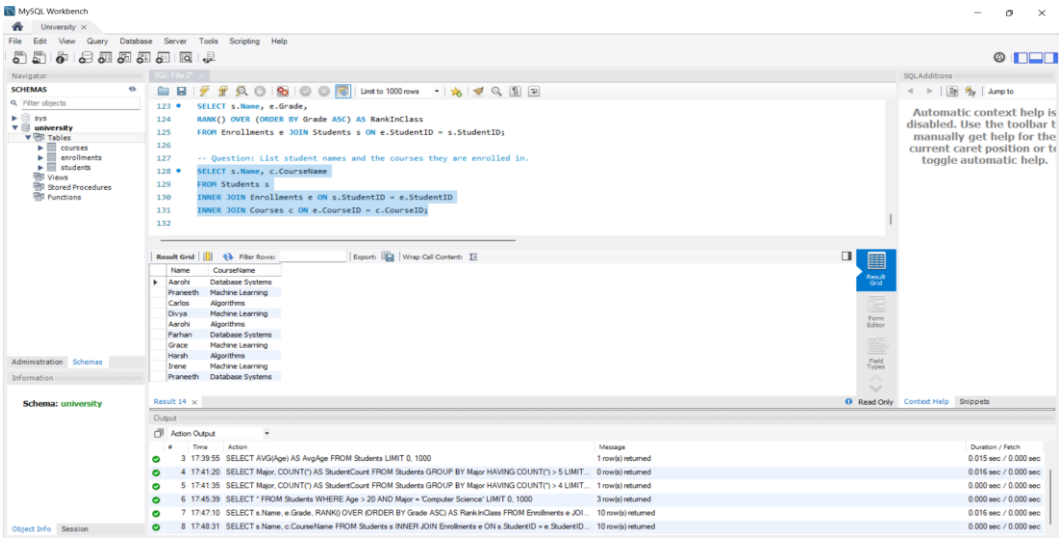

MySQL Workbench
-- Question: Show the name and major of all students who are older than 19.
SELECT Name, Major FROM Students WHERE Age > 19;
Name Major
Praneeth Data Science
Carlos Computer Science
Divya Mathematics
Farhan Computer Science
Grace Undeclared
Harsh Computer Science
Irene Mathematics
Jayden Computer Science
Kiara Physics
Maya Computer Science
Omar Mathematics
Schema: university
1 17:36:45 SELECT * FROM Students LIMIT 0, 1000
14 row(s) returned
2 17:38:04 SELECT Name, Major FROM Students WHERE Age > 19 LIMIT 0, 1000
11 row(s) returned

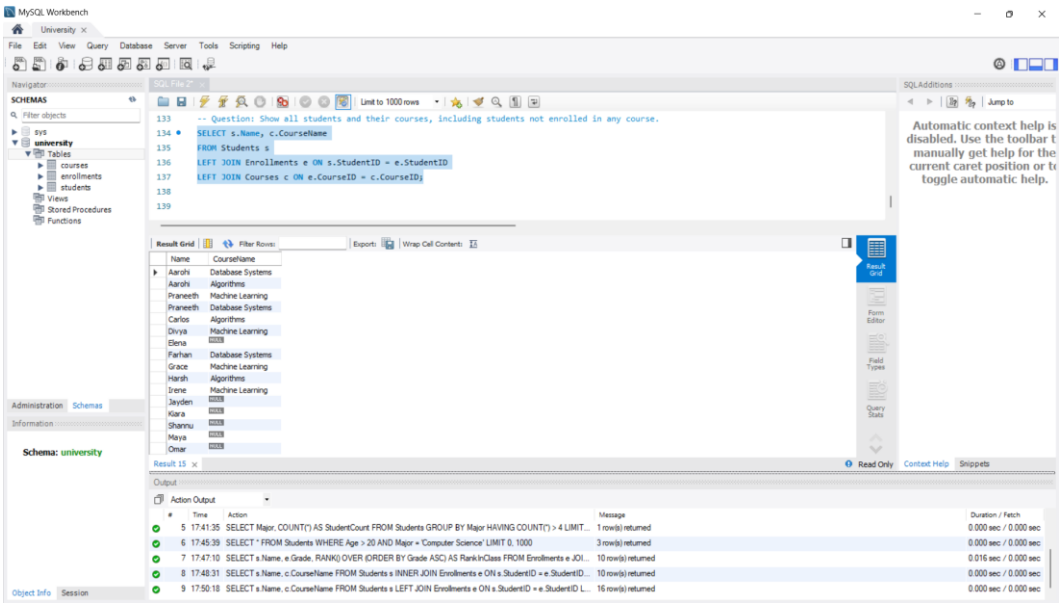

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
SQL File 2*
-- Question: Show the name and major of all students who are older than 19.
SELECT Name, Major FROM Students WHERE Age > 19;
-- Question: What is the average age of all students?
SELECT AVG(Age) AS AvgAge FROM Students;
AvgAge
21.1429
Schema: university
Action Output
1 17:36:45 SELECT * FROM Students LIMIT 0, 1000 14 row(s) returned 0.000 sec / 0.000 sec
2 17:38:04 SELECT Name, Major FROM Students WHERE Age > 19 LIMIT 0, 1000 11 row(s) returned 0.000 sec / 0.000 sec
3 17:39:55 SELECT AVG(Age) AS AvgAge FROM Students LIMIT 0, 1000 1 row(s) returned 0.015 sec / 0.000 sec

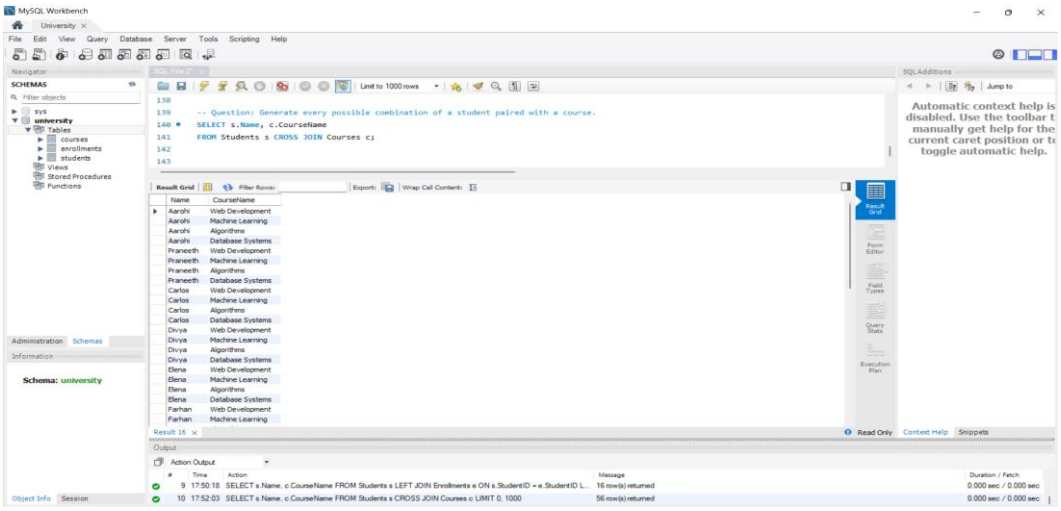

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
SQL File 2*
SELECT AVG(Age) AS AvgAge FROM Students;
-- Question: Which majors have more than 4 students, and what is the count for each?
SELECT Major, COUNT(*) AS StudentCount
FROM Students
GROUP BY Major
HAVING COUNT(*) > 4;
Major
StudentCount
Computer Science
5
Schema: university
Action Output
1 17:36:45 SELECT * FROM Students LIMIT 0, 1000 14 row(s) returned 0.000 sec / 0.000 sec
2 17:38:04 SELECT Name, Major FROM Students WHERE Age > 19 LIMIT 0, 1000 11 row(s) returned 0.000 sec / 0.000 sec
3 17:39:55 SELECT AVG(Age) AS AvgAge FROM Students LIMIT 0, 1000 1 row(s) returned 0.015 sec / 0.000 sec
4 17:41:20 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 5 LIMIT 0... 0 row(s) returned 0.016 sec / 0.000 sec
5 17:41:35 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 4 LIMIT 0... 1 row(s) returned 0.000 sec / 0.000 sec

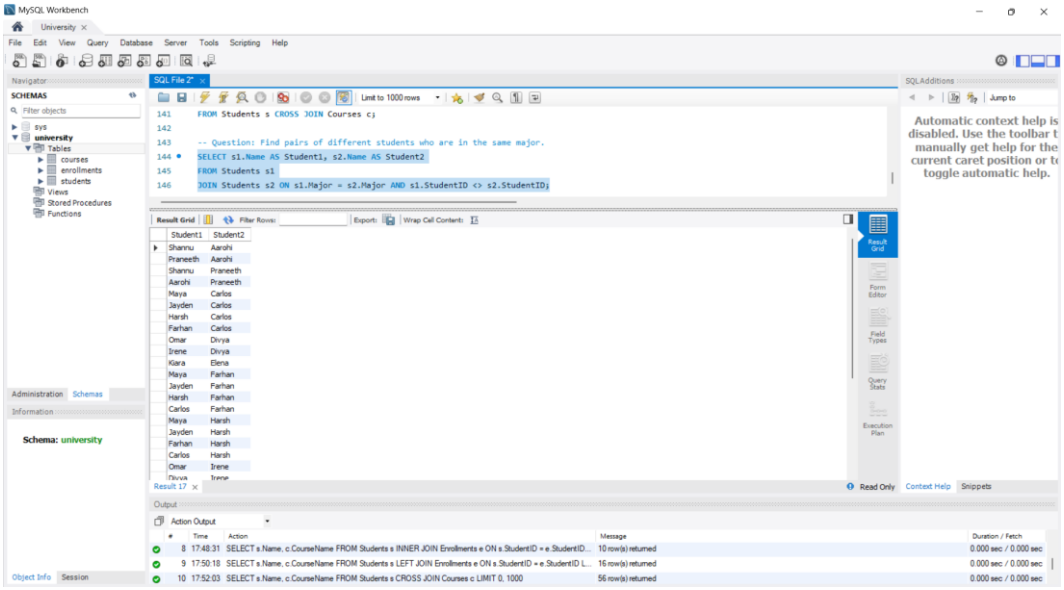

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
SQL File 2*
SELECT AVG(Age) AS AvgAge FROM Students;
-- Question: Which majors have more than 4 students, and what is the count for each?
SELECT Major, COUNT(*) AS StudentCount
FROM Students
GROUP BY Major
HAVING COUNT(*) > 4;
-- Question: List all information for students older than 20 and in 'Computer Science'.
SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science';
StudentID Name Age Major Email
6 Farhan 22 Computer Science NULL
10 Jayden 24 Computer Science NULL
13 Maya 23 Computer Science NULL
Schema: university
Action Output
2 17:38:04 SELECT Name, Major FROM Students WHERE Age > 19 LIMIT 0, 1000 11 row(s) returned 0.000 sec / 0.000 sec
3 17:39:55 SELECT AVG(Age) AS AvgAge FROM Students LIMIT 0, 1000 1 row(s) returned 0.015 sec / 0.000 sec
4 17:41:20 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 5 LIMIT... 0 row(s) returned 0.016 sec / 0.000 sec
5 17:41:35 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 4 LIMIT... 1 row(s) returned 0.000 sec / 0.000 sec
6 17:45:39 SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science' LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec

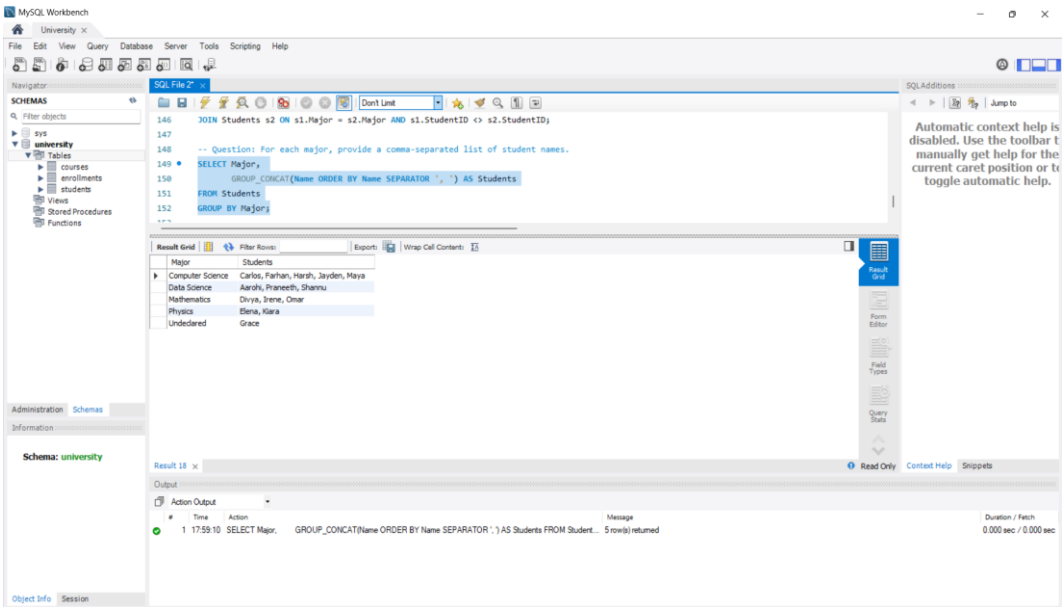

MySQL Workbench
-- Question: Which majors have more than 4 students, and what is the count for each?
SELECT Major, COUNT(*) AS StudentCount
FROM Students
GROUP BY Major
HAVING COUNT(*) > 4;
-- Question: List all information for students older than 20 and in 'Computer Science'.
SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science';
-- Question: How can you rank students based on their grades?
SELECT s.Name, e.Grade,
Name Grade RankInClass
Aarohi A 1
Carlos A 1
Irene A 1
Farhan A- 4
Aarohi B 5
Praneeth B 5
Praneeth B 5
Grace B+ 8
Divya C 9
Harsh C+ 10
Schema: university
3 17:39:55 SELECT AVG(Age) AS AvgAge FROM Students LIMIT 0, 1000 1 row(s) returned 0.015 sec / 0.000 sec
4 17:41:20 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 5 LIMIT... 0 row(s) returned 0.016 sec / 0.000 sec
5 17:41:35 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 4 LIMIT... 1 row(s) returned 0.000 sec / 0.000 sec
6 17:45:39 SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science' LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
7 17:47:10 SELECT s.Name, e.Grade, RANK() OVER (ORDER BY Grade ASC) AS RankInClass FROM Enrollments e JOI... 10 row(s) returned 0.016 sec / 0.000 sec

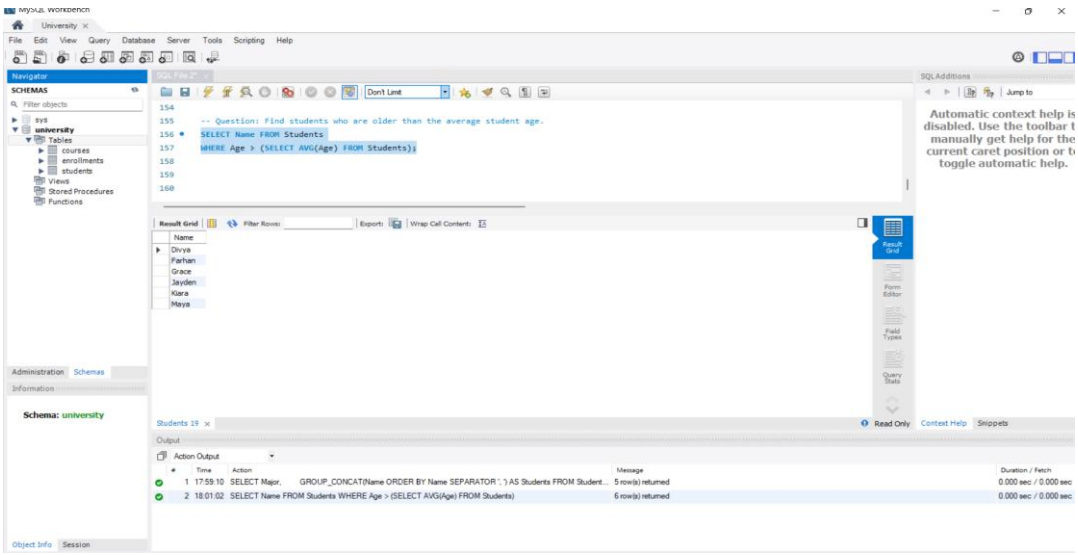

MySQL Workbench
SELECT s.Name, e.Grade,
RANK() OVER (ORDER BY Grade ASC) AS RankInClass
FROM Enrollments e JOIN Students s ON e.StudentID = s.StudentID;
-- Question: List student names and the courses they are enrolled in.
SELECT s.Name, c.CourseName
FROM Students s
INNER JOIN Enrollments e ON s.StudentID = e.StudentID
INNER JOIN Courses c ON e.CourseID = c.CourseID;
Name CourseName
Aarohi Database Systems
Praneeth Machine Learning
Carlos Algorithms
Divya Machine Learning
Aarohi Algorithms
Farhan Database Systems
Grace Machine Learning
Harsh Algorithms
Irene Machine Learning
Praneeth Database Systems
Schema: university
3 17:39:55 SELECT AVG(Age) AS AvgAge FROM Students LIMIT 0, 1000 1 row(s) returned 0.015 sec / 0.000 sec
4 17:41:20 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 5 LIMIT... 0 row(s) returned 0.016 sec / 0.000 sec
5 17:41:35 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 4 LIMIT... 1 row(s) returned 0.000 sec / 0.000 sec
6 17:45:39 SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science' LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
7 17:47:10 SELECT s.Name, e.Grade, RANK() OVER (ORDER BY Grade ASC) AS RankInClass FROM Enrollments e JOI... 10 row(s) returned 0.016 sec / 0.000 sec
8 17:48:31 SELECT s.Name, c.CourseName FROM Students s INNER JOIN Enrollments e ON s.StudentID = e.StudentID... 10 row(s) returned 0.000 sec / 0.000 sec

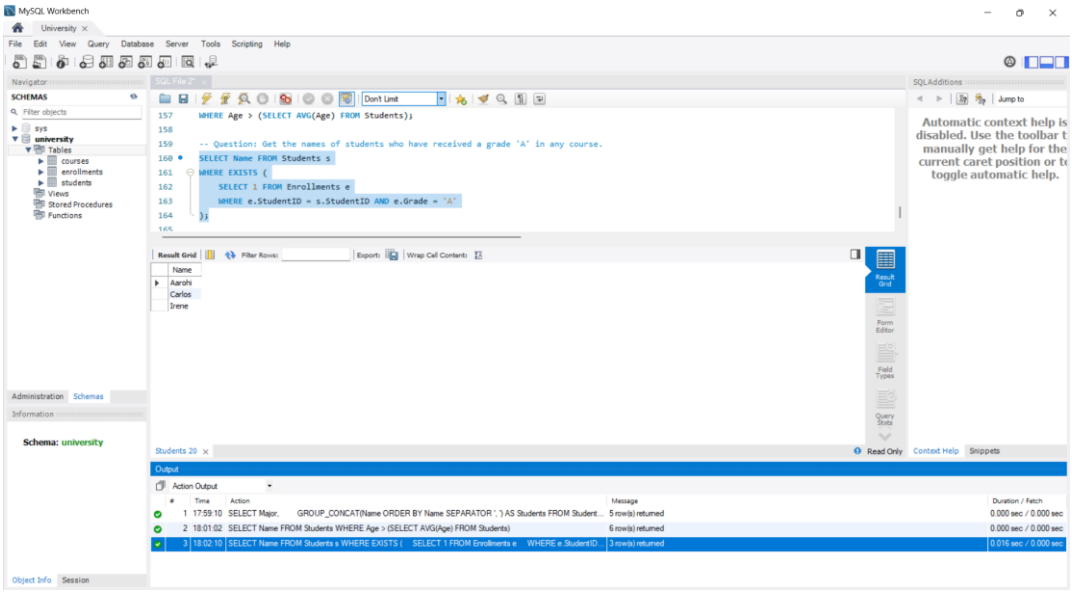

MySQL Workbench
-- Question: Show all students and their courses, including students not enrolled in any course.
SELECT s.Name, c.CourseName
FROM Students s
LEFT JOIN Enrollments e ON s.StudentID = e.StudentID
LEFT JOIN Courses c ON e.CourseID = c.CourseID;
Name CourseName
Aarohi Database Systems
Aarohi Algorithms
Praneeth Machine Learning
Praneeth Database Systems
Carlos Algorithms
Divya Machine Learning
Elena NULL
Farhan Database Systems
Grace Machine Learning
Harsh Algorithms
Irene Machine Learning
Jayden NULL
Kiara NULL
Shannu NULL
Maya NULL
Omar NULL
Schema: university
5 17:41:35 SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY Major HAVING COUNT(*) > 4 LIMIT... 1 row(s) returned 0.000 sec / 0.000 sec
6 17:45:39 SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science' LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
7 17:47:10 SELECT s.Name, e.Grade, RANK() OVER (ORDER BY Grade ASC) AS RankInClass FROM Enrollments e JOI... 10 row(s) returned 0.016 sec / 0.000 sec
8 17:48:31 SELECT s.Name, c.CourseName FROM Students s INNER JOIN Enrollments e ON s.StudentID = e.StudentID... 10 row(s) returned 0.000 sec / 0.000 sec
9 17:50:18 SELECT s.Name, c.CourseName FROM Students s LEFT JOIN Enrollments e ON s.StudentID = e.StudentID L... 16 row(s) returned 0.000 sec / 0.000 sec

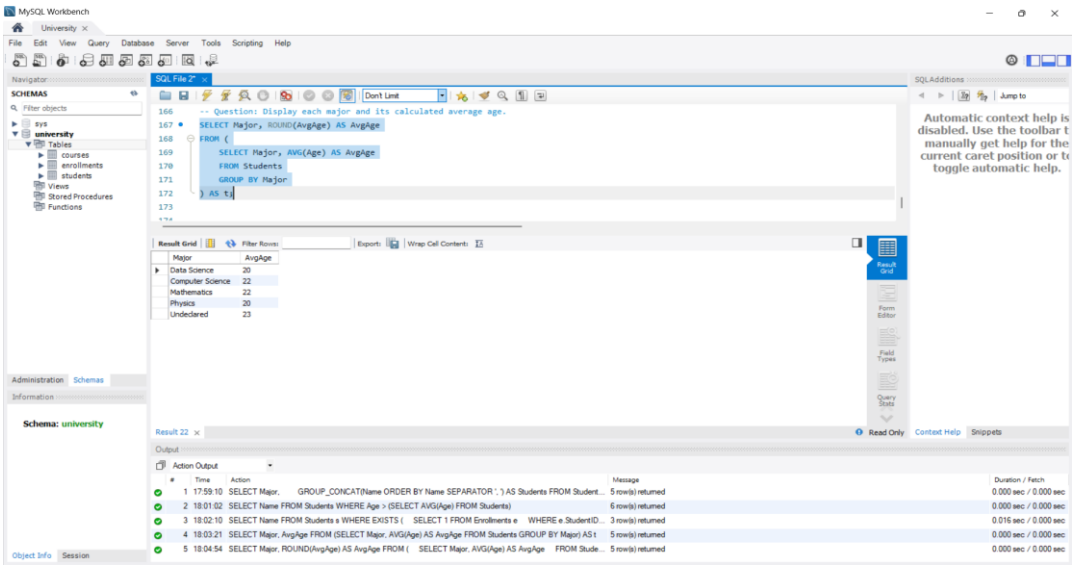

MySQL Workbench
-- Question: Generate every possible combination of a student paired with a course.
SELECT s.Name, c.CourseName
FROM Students s CROSS JOIN Courses c;
Name CourseName
Aarohi Web Development
Aarohi Machine Learning
Aarohi Algorithms
Aarohi Database Systems
Praneeth Web Development
Praneeth Machine Learning
Praneeth Algorithms
Praneeth Database Systems
Carlos Web Development
Carlos Machine Learning
Carlos Algorithms
Carlos Database Systems
Divya Web Development
Divya Machine Learning
Divya Algorithms
Divya Database Systems
Elena Web Development
Elena Machine Learning
Elena Algorithms
Elena Database Systems
Farhan Web Development
Farhan Machine Learning
Schema: university

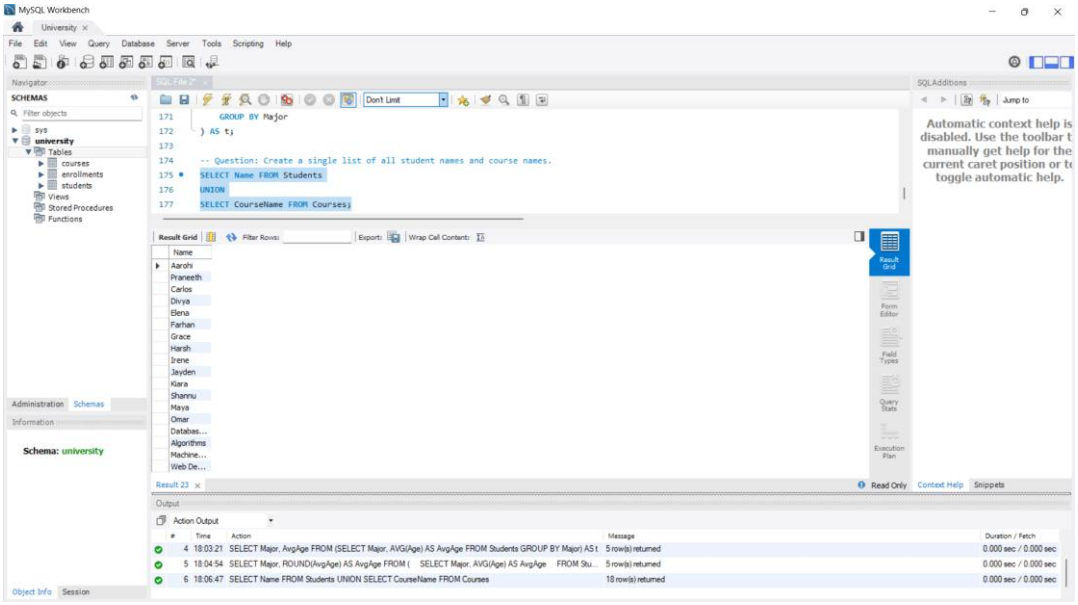

MySQL Workbench
FROM Students s CROSS JOIN Courses c;
-- Question: Find pairs of different students who are in the same major.
SELECT s1.Name AS Student1, s2.Name AS Student2
FROM Students s1
JOIN Students s2 ON s1.Major = s2.Major AND s1.StudentID <> s2.StudentID;
Student1 Student2
Shannu Aarohi
Praneeth Aarohi
Shannu Praneeth
Aarohi Praneeth
Maya Carlos
Jayden Carlos
Harsh Carlos
Farhan Carlos
Omar Divya
Irene Divya
Kiara Elena
Maya Farhan
Jayden Farhan
Harsh Farhan
Carlos Farhan
Maya Harsh
Jayden Harsh
Farhan Harsh
Carlos Harsh
Omar Irene
Schema: university

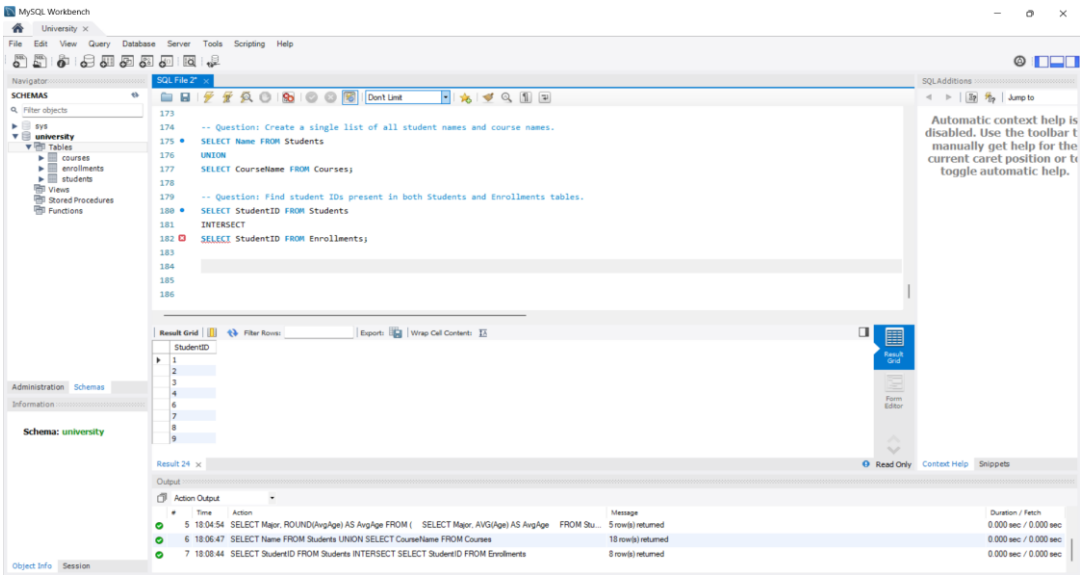

MySQL Workbench
JOIN Students s2 ON s1.Major = s2.Major AND s1.StudentID <> s2.StudentID;
-- Question: For each major, provide a comma-separated list of student names.
SELECT Major,
GROUP_CONCAT(Name ORDER BY Name SEPARATOR ', ') AS Students
FROM Students
GROUP BY Major;
Major Students
Computer Science Carlos, Farhan, Harsh, Jayden, Maya
Data Science Aarohi, Praneeth, Shannu
Mathematics Divya, Irene, Omar
Physics Elena, Kiara
Undeclared Grace
Schema: university

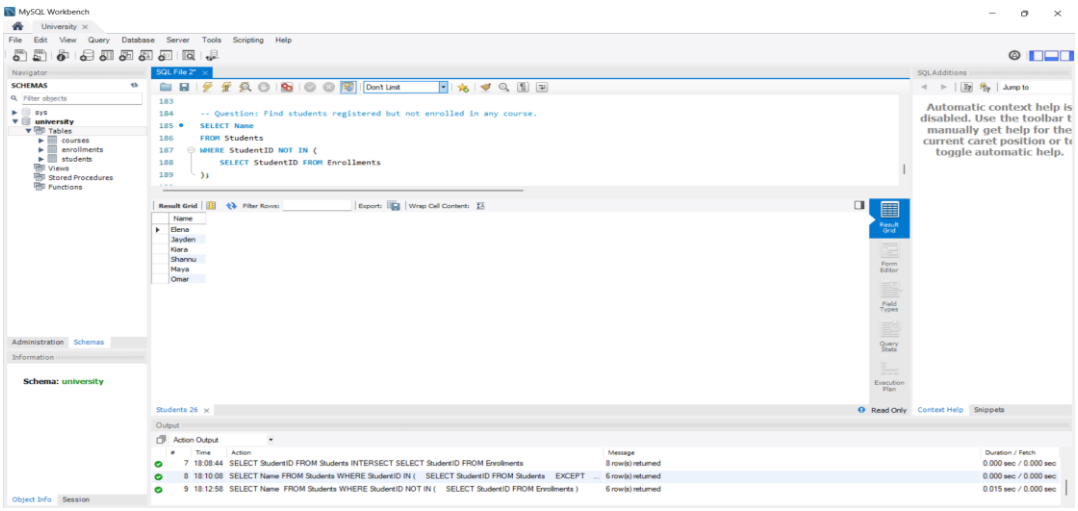

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Don't Limit
-- Question: Find students who are older than the average student age.
SELECT Name FROM Students
WHERE Age > (SELECT AVG(Age) FROM Students);
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Name
Divya
Farhan
Grace
Jayden
Kiara
Maya
Students 19
Read Only
Context Help
Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 17:59:10 SELECT Major, GROUP_CONCAT(Name ORDER BY Name SEPARATOR ', ') AS Students FROM Student... 5 row(s) returned 0.000 sec / 0.000 sec
2 18:01:02 SELECT Name FROM Students WHERE Age > (SELECT AVG(Age) FROM Students) 6 row(s) returned 0.000 sec / 0.000 sec
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

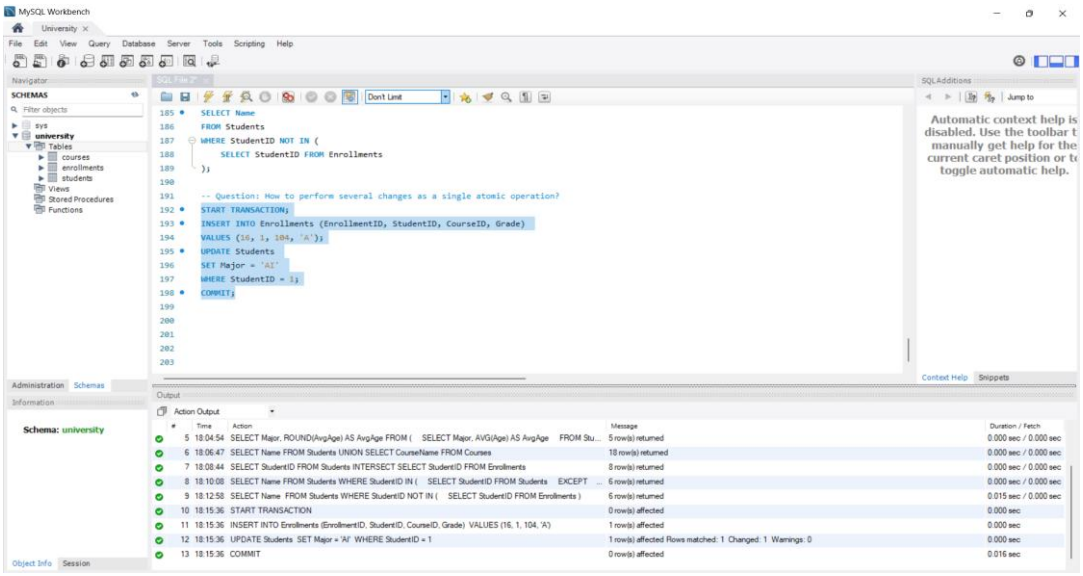

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Don't Limit
WHERE Age > (SELECT AVG(Age) FROM Students);
-- Question: Get the names of students who have received a grade 'A' in any course.
SELECT Name FROM Students s
WHERE EXISTS (
SELECT 1 FROM Enrollments e
WHERE e.StudentID = s.StudentID AND e.Grade = 'A'
);
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Name
Aarohi
Carlos
Irene
Students 20
Read Only
Context Help
Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 17:59:10 SELECT Major, GROUP_CONCAT(Name ORDER BY Name SEPARATOR ', ') AS Students FROM Student... 5 row(s) returned 0.000 sec / 0.000 sec
2 18:01:02 SELECT Name FROM Students WHERE Age > (SELECT AVG(Age) FROM Students) 6 row(s) returned 0.000 sec / 0.000 sec
3 18:02:10 SELECT Name FROM Students s WHERE EXISTS ( SELECT 1 FROM Enrollments e WHERE e.StudentID... 3 row(s) returned 0.016 sec / 0.000 sec
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

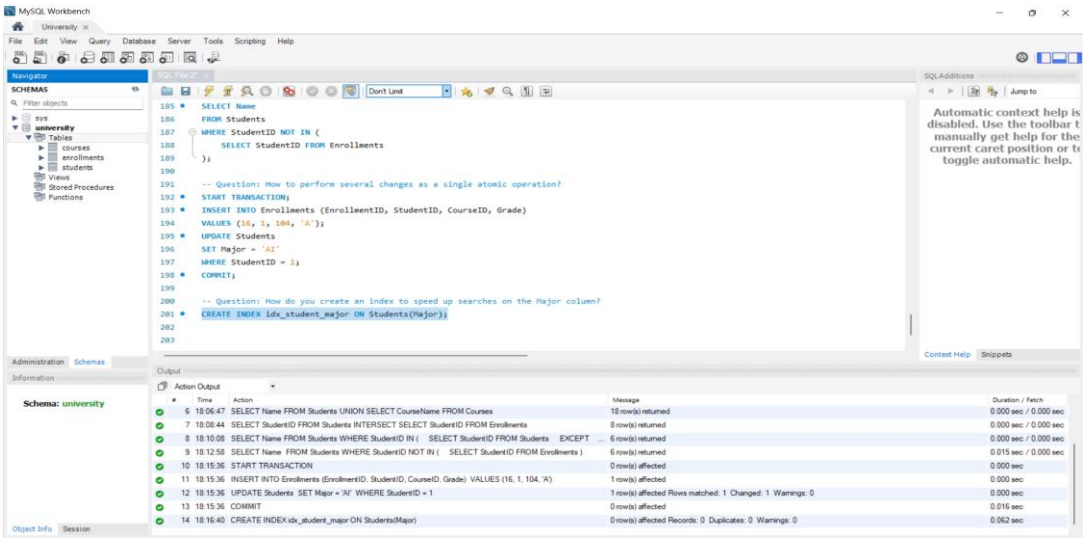

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Don't Limit
-- Question: Display each major and its calculated average age.
SELECT Major, ROUND(AvgAge) AS AvgAge
FROM (
SELECT Major, AVG(Age) AS AvgAge
FROM Students
GROUP BY Major
) AS t;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Major AvgAge
Data Science 20
Computer Science 22
Mathematics 22
Physics 20
Undeclared 23
Result 22
Read Only
Context Help
Snippets
Output
Action Output
# Time Action Message Duration / Fetch
1 17:59:10 SELECT Major, GROUP_CONCAT(Name ORDER BY Name SEPARATOR ', ') AS Students FROM Student... 5 row(s) returned 0.000 sec / 0.000 sec
2 18:01:02 SELECT Name FROM Students WHERE Age > (SELECT AVG(Age) FROM Students) 6 row(s) returned 0.000 sec / 0.000 sec
3 18:02:10 SELECT Name FROM Students s WHERE EXISTS ( SELECT 1 FROM Enrollments e WHERE e.StudentID... 3 row(s) returned 0.016 sec / 0.000 sec
4 18:03:21 SELECT Major, AvgAge FROM (SELECT Major, AVG(Age) AS AvgAge FROM Students GROUP BY Major) AS t 5 row(s) returned 0.000 sec / 0.000 sec
5 18:04:54 SELECT Major, ROUND(AvgAge) AS AvgAge FROM ( SELECT Major, AVG(Age) AS AvgAge FROM Stude... 5 row(s) returned 0.000 sec / 0.000 sec
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

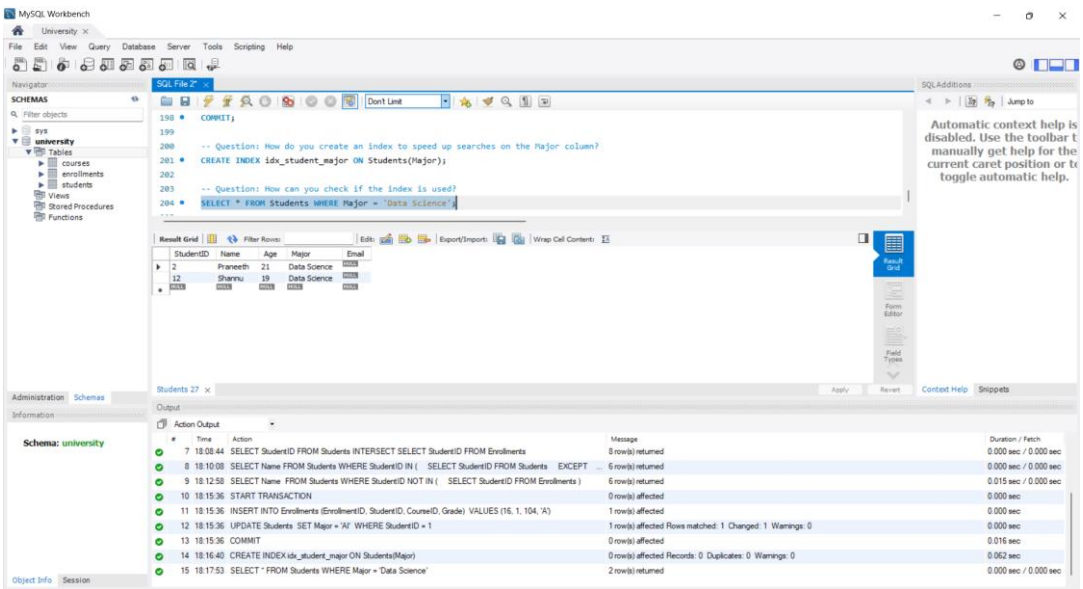

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
SQL File 2*
GROUP BY Major
) AS t;
-- Question: Create a single list of all student names and course names.
SELECT Name FROM Students
UNION
SELECT CourseName FROM Courses;
Result Grid
Name
Aarohi
Praneeth
Carlos
Divya
Elena
Farhan
Grace
Harsh
Irene
Jayden
Kiara
Shannu
Maya
Omar
Databas...
Algorithms
Machine...
Web De...
Result 23
Output
Action Output
4 18:03:21 SELECT Major, AvgAge FROM (SELECT Major, AVG(Age) AS AvgAge FROM Students GROUP BY Major) AS t 5 row(s) returned 0.000 sec / 0.000 sec
5 18:04:54 SELECT Major, ROUND(AvgAge) AS AvgAge FROM ( SELECT Major, AVG(Age) AS AvgAge FROM Stu... 5 row(s) returned 0.000 sec / 0.000 sec
6 18:06:47 SELECT Name FROM Students UNION SELECT CourseName FROM Courses 18 row(s) returned 0.000 sec / 0.000 sec
Schema: university

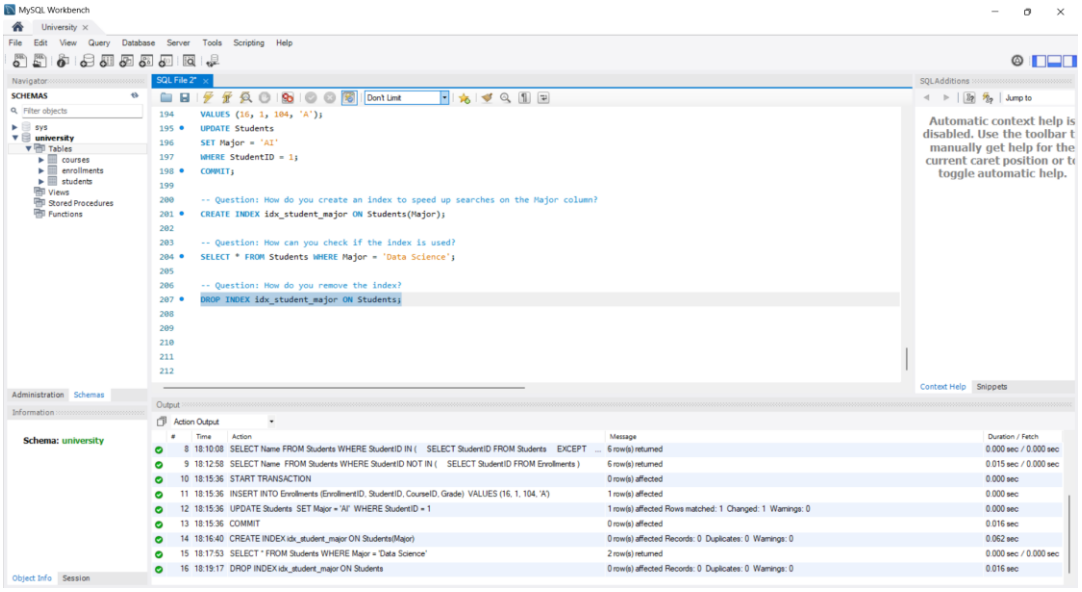

MySQL Workbench
University
SQL File 2*
-- Question: Create a single list of all student names and course names.
SELECT Name FROM Students
UNION
SELECT CourseName FROM Courses;
-- Question: Find student IDs present in both Students and Enrollments tables.
SELECT StudentID FROM Students
INTERSECT
SELECT StudentID FROM Enrollments;
Result Grid
StudentID
1
2
3
4
6
7
8
9
Result 24
Output
Action Output
5 18:04:54 SELECT Major, ROUND(AvgAge) AS AvgAge FROM ( SELECT Major, AVG(Age) AS AvgAge FROM Stu... 5 row(s) returned 0.000 sec / 0.000 sec
6 18:06:47 SELECT Name FROM Students UNION SELECT CourseName FROM Courses 18 row(s) returned 0.000 sec / 0.000 sec
7 18:08:44 SELECT StudentID FROM Students INTERSECT SELECT StudentID FROM Enrollments 8 row(s) returned 0.000 sec / 0.000 sec
Schema: university

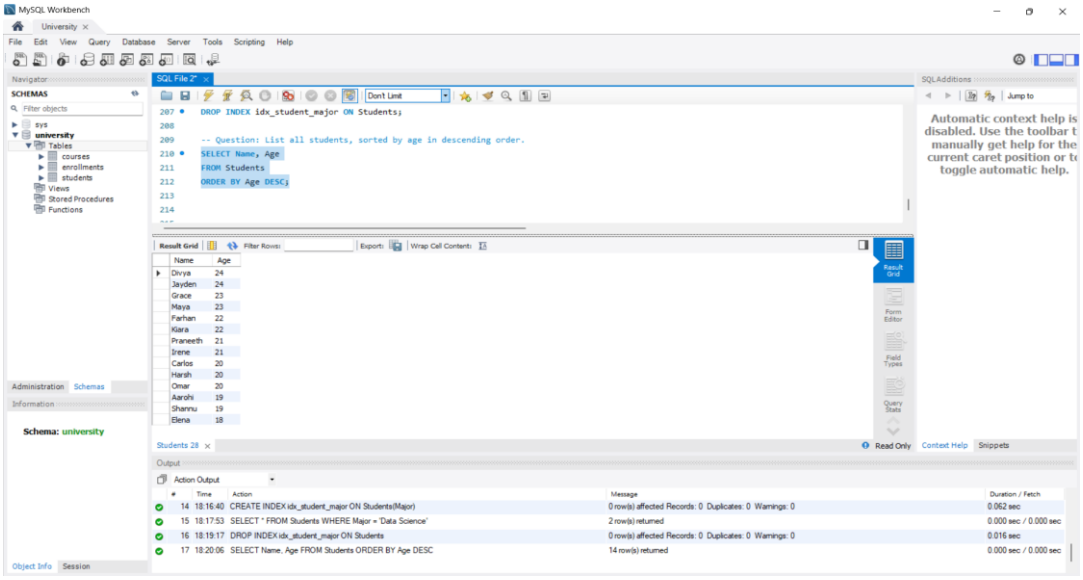

MySQL Workbench
University
SQL File 2*
-- Question: Find students registered but not enrolled in any course.
SELECT Name
FROM Students
WHERE StudentID NOT IN (
SELECT StudentID FROM Enrollments
);
Result Grid
Name
Elena
Jayden
Kiara
Shannu
Maya
Omar
Students 26
Output
Action Output
7 18:08:44 SELECT StudentID FROM Students INTERSECT SELECT StudentID FROM Enrollments 8 row(s) returned 0.000 sec / 0.000 sec
8 18:10:08 SELECT Name FROM Students WHERE StudentID IN ( SELECT StudentID FROM Students EXCEPT ... 6 row(s) returned 0.000 sec / 0.000 sec
9 18:12:58 SELECT Name FROM Students WHERE StudentID NOT IN ( SELECT StudentID FROM Enrollments ) 6 row(s) returned 0.015 sec / 0.000 sec
Schema: university

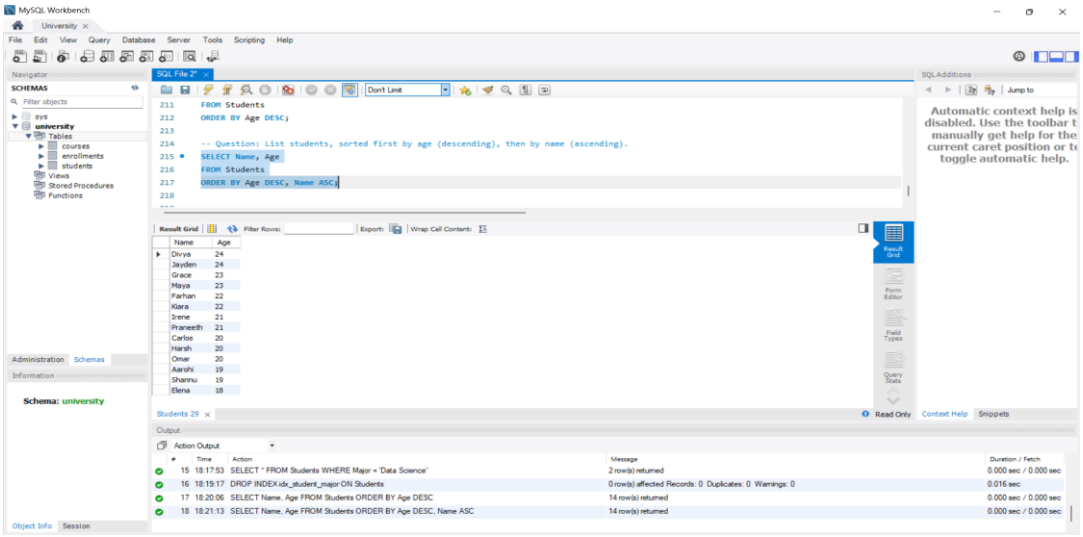

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Don't Limit
185 SELECT Name
186 FROM Students
187 WHERE StudentID NOT IN (
188 SELECT StudentID FROM Enrollments
189 );
190
191 -- Question: How to perform several changes as a single atomic operation?
192 START TRANSACTION;
193 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade)
194 VALUES (16, 1, 104, 'A');
195 UPDATE Students
196 SET Major = 'AI'
197 WHERE StudentID = 1;
198 COMMIT;
199
200
201
202
203
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
5 18:04:54 SELECT Major, ROUND(AvgAge) AS AvgAge FROM ( SELECT Major, AVG(Age) AS AvgAge FROM Stu... 5 row(s) returned 0.000 sec / 0.000 sec
6 18:06:47 SELECT Name FROM Students UNION SELECT CourseName FROM Courses 18 row(s) returned 0.000 sec / 0.000 sec
7 18:08:44 SELECT StudentID FROM Students INTERSECT SELECT StudentID FROM Enrollments 8 row(s) returned 0.000 sec / 0.000 sec
8 18:10:08 SELECT Name FROM Students WHERE StudentID IN ( SELECT StudentID FROM Students EXCEPT ... 6 row(s) returned 0.000 sec / 0.000 sec
9 18:12:58 SELECT Name FROM Students WHERE StudentID NOT IN ( SELECT StudentID FROM Enrollments ) 6 row(s) returned 0.015 sec / 0.000 sec
10 18:15:36 START TRANSACTION 0 row(s) affected 0.000 sec
11 18:15:36 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (16, 1, 104, 'A') 1 row(s) affected 0.000 sec
12 18:15:36 UPDATE Students SET Major = 'AI' WHERE StudentID = 1 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec
13 18:15:36 COMMIT 0 row(s) affected 0.016 sec

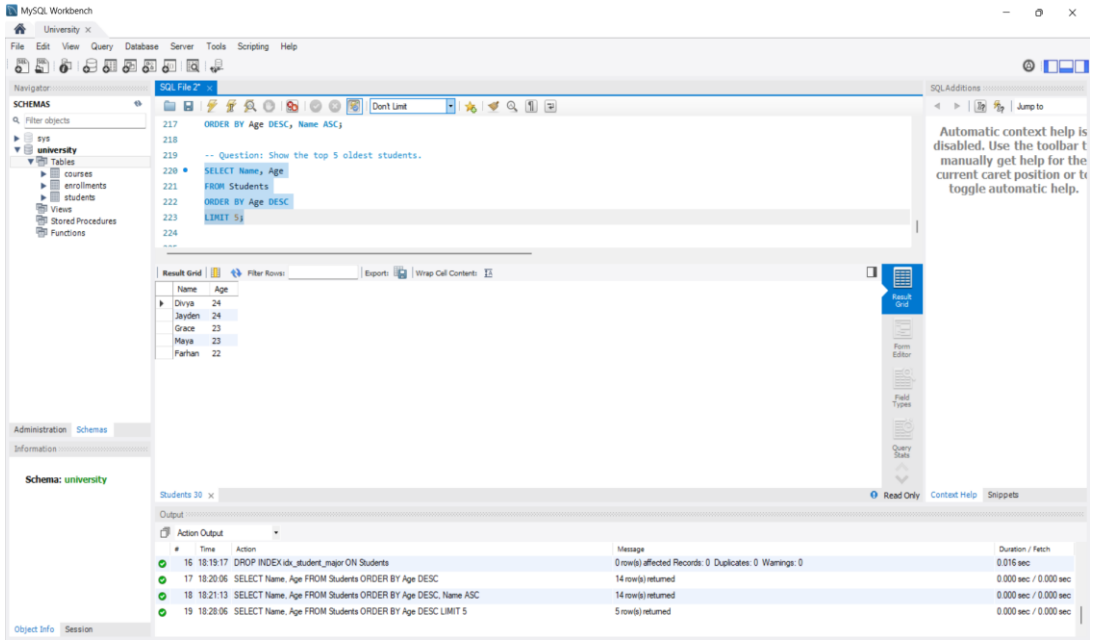

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Don't Limit
185 SELECT Name
186 FROM Students
187 WHERE StudentID NOT IN (
188 SELECT StudentID FROM Enrollments
189 );
190
191 -- Question: How to perform several changes as a single atomic operation?
192 START TRANSACTION;
193 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade)
194 VALUES (16, 1, 104, 'A');
195 UPDATE Students
196 SET Major = 'AI'
197 WHERE StudentID = 1;
198 COMMIT;
199
200 -- Question: How do you create an index to speed up searches on the Major column?
201 CREATE INDEX idx_student_major ON Students(Major);
202
203
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
6 18:06:47 SELECT Name FROM Students UNION SELECT CourseName FROM Courses 18 row(s) returned 0.000 sec / 0.000 sec
7 18:08:44 SELECT StudentID FROM Students INTERSECT SELECT StudentID FROM Enrollments 8 row(s) returned 0.000 sec / 0.000 sec
8 18:10:08 SELECT Name FROM Students WHERE StudentID IN ( SELECT StudentID FROM Students EXCEPT ... 6 row(s) returned 0.000 sec / 0.000 sec
9 18:12:58 SELECT Name FROM Students WHERE StudentID NOT IN ( SELECT StudentID FROM Enrollments ) 6 row(s) returned 0.015 sec / 0.000 sec
10 18:15:36 START TRANSACTION 0 row(s) affected 0.000 sec
11 18:15:36 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (16, 1, 104, 'A') 1 row(s) affected 0.000 sec
12 18:15:36 UPDATE Students SET Major = 'AI' WHERE StudentID = 1 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec
13 18:15:36 COMMIT 0 row(s) affected 0.016 sec
14 18:16:40 CREATE INDEX idx_student_major ON Students(Major) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.062 sec

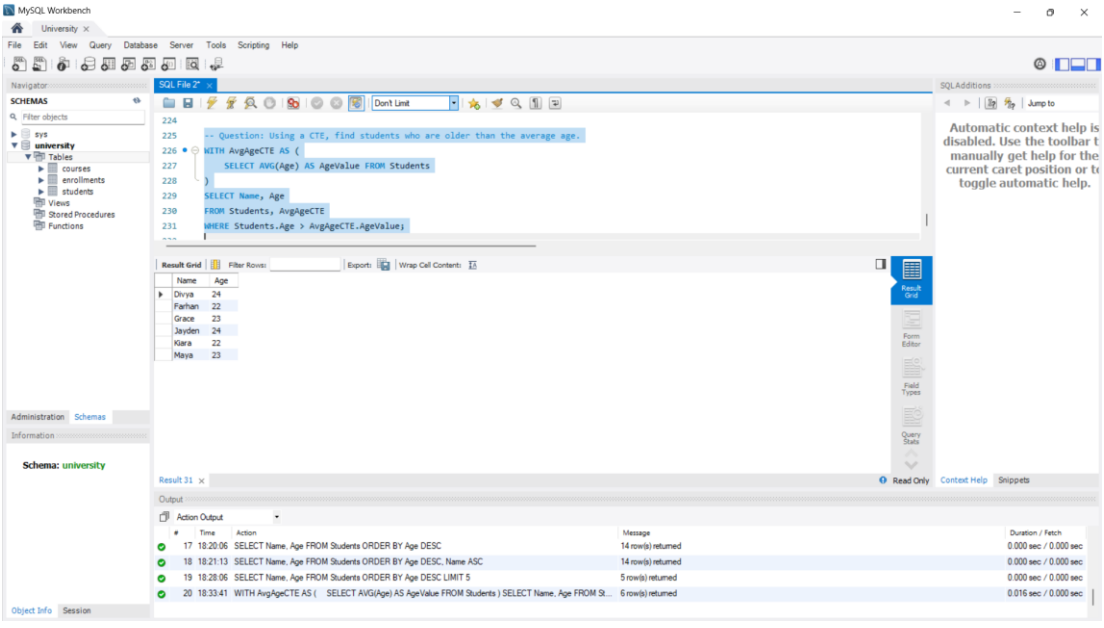

MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
Administration Schemas
Information
Schema: university
Object Info Session
SQL File 2*
Don't Limit
198 COMMIT;
199
200 -- Question: How do you create an index to speed up searches on the Major column?
201 CREATE INDEX idx_student_major ON Students(Major);
202
203 -- Question: How can you check if the index is used?
204 SELECT * FROM Students WHERE Major = 'Data Science';
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
StudentID Name Age Major Email
2 Praneeth 21 Data Science NULL
12 Shannu 19 Data Science NULL
NULL NULL NULL NULL NULL
Students 27
Apply
Revert
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
7 18:08:44 SELECT StudentID FROM Students INTERSECT SELECT StudentID FROM Enrollments 8 row(s) returned 0.000 sec / 0.000 sec
8 18:10:08 SELECT Name FROM Students WHERE StudentID IN ( SELECT StudentID FROM Students EXCEPT ... 6 row(s) returned 0.000 sec / 0.000 sec
9 18:12:58 SELECT Name FROM Students WHERE StudentID NOT IN ( SELECT StudentID FROM Enrollments ) 6 row(s) returned 0.015 sec / 0.000 sec
10 18:15:36 START TRANSACTION 0 row(s) affected 0.000 sec
11 18:15:36 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (16, 1, 104, 'A') 1 row(s) affected 0.000 sec
12 18:15:36 UPDATE Students SET Major = 'AI' WHERE StudentID = 1 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec
13 18:15:36 COMMIT 0 row(s) affected 0.016 sec
14 18:16:40 CREATE INDEX idx_student_major ON Students(Major) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.062 sec
15 18:17:53 SELECT * FROM Students WHERE Major = 'Data Science' 2 row(s) returned 0.000 sec / 0.000 sec


MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
SQL File 2*
Don't Limit
194 VALUES (16, 1, 104, 'A');
195 UPDATE Students
196 SET Major = 'AI'
197 WHERE StudentID = 1;
198 COMMIT;
199
200 -- Question: How do you create an index to speed up searches on the Major column?
201 CREATE INDEX idx_student_major ON Students(Major);
202
203 -- Question: How can you check if the index is used?
204 SELECT * FROM Students WHERE Major = 'Data Science';
205
206 -- Question: How do you remove the index?
207 DROP INDEX idx_student_major ON Students;
208
209
210
211
212
SQLAdditions
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets
Administration
Schemas
Information
Schema: university
Object Info
Session
Output
Action Output
# Time Action Message Duration / Fetch
8 18:10:08 SELECT Name FROM Students WHERE StudentID IN ( SELECT StudentID FROM Students EXCEPT ... 6 row(s) returned 0.000 sec / 0.000 sec
9 18:12:58 SELECT Name FROM Students WHERE StudentID NOT IN ( SELECT StudentID FROM Enrollments ) 6 row(s) returned 0.015 sec / 0.000 sec
10 18:15:36 START TRANSACTION 0 row(s) affected 0.000 sec
11 18:15:36 INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (16, 1, 104, 'A') 1 row(s) affected 0.000 sec
12 18:15:36 UPDATE Students SET Major = 'AI' WHERE StudentID = 1 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.000 sec
13 18:15:36 COMMIT 0 row(s) affected 0.016 sec
14 18:16:40 CREATE INDEX idx_student_major ON Students(Major) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.062 sec
15 18:17:53 SELECT * FROM Students WHERE Major = 'Data Science' 2 row(s) returned 0.000 sec / 0.000 sec
16 18:19:17 DROP INDEX idx_student_major ON Students 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.016 sec


MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
SQL File 2*
Don't Limit
207 DROP INDEX idx_student_major ON Students;
208
209 -- Question: List all students, sorted by age in descending order.
210 SELECT Name, Age
211 FROM Students
212 ORDER BY Age DESC;
213
214
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Name Age
Divya 24
Jayden 24
Grace 23
Maya 23
Farhan 22
Kiara 22
Praneeth 21
Irene 21
Carlos 20
Harsh 20
Omar 20
Aarohi 19
Shannu 19
Elena 18
Students 28
Read Only
SQLAdditions
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets
Administration
Schemas
Information
Schema: university
Object Info
Session
Output
Action Output
# Time Action Message Duration / Fetch
14 18:16:40 CREATE INDEX idx_student_major ON Students(Major) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.062 sec
15 18:17:53 SELECT * FROM Students WHERE Major = 'Data Science' 2 row(s) returned 0.000 sec / 0.000 sec
16 18:19:17 DROP INDEX idx_student_major ON Students 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.016 sec
17 18:20:06 SELECT Name, Age FROM Students ORDER BY Age DESC 14 row(s) returned 0.000 sec / 0.000 sec


MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
SQL File 2*
Don't Limit
211 FROM Students
212 ORDER BY Age DESC;
213
214 -- Question: List students, sorted first by age (descending), then by name (ascending).
215 SELECT Name, Age
216 FROM Students
217 ORDER BY Age DESC, Name ASC;
218
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Name Age
Divya 24
Jayden 24
Grace 23
Maya 23
Farhan 22
Kiara 22
Irene 21
Praneeth 21
Carlos 20
Harsh 20
Omar 20
Aarohi 19
Shannu 19
Elena 18
Students 29
Read Only
SQLAdditions
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help
Snippets
Administration
Schemas
Information
Schema: university
Object Info
Session
Output
Action Output
# Time Action Message Duration / Fetch
15 18:17:53 SELECT * FROM Students WHERE Major = 'Data Science' 2 row(s) returned 0.000 sec / 0.000 sec
16 18:19:17 DROP INDEX idx_student_major ON Students 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.016 sec
17 18:20:06 SELECT Name, Age FROM Students ORDER BY Age DESC 14 row(s) returned 0.000 sec / 0.000 sec
18 18:21:13 SELECT Name, Age FROM Students ORDER BY Age DESC, Name ASC 14 row(s) returned 0.000 sec / 0.000 sec


MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
SQL File 2*
Don't Limit
217 ORDER BY Age DESC, Name ASC;
219 -- Question: Show the top 5 oldest students.
220 SELECT Name, Age
221 FROM Students
222 ORDER BY Age DESC
223 LIMIT 5;
Result Grid
Name Age
Divya 24
Jayden 24
Grace 23
Maya 23
Parhan 22
SQL Additions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Administration Schemas
Information
Schema: university
Object Info Session
Students 30
Read Only
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
16 18:19:17 DROP INDEX idx_student_major ON Students 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.016 sec
17 18:20:06 SELECT Name, Age FROM Students ORDER BY Age DESC 14 row(s) returned 0.000 sec / 0.000 sec
18 18:21:13 SELECT Name, Age FROM Students ORDER BY Age DESC, Name ASC 14 row(s) returned 0.000 sec / 0.000 sec
19 18:28:06 SELECT Name, Age FROM Students ORDER BY Age DESC LIMIT 5 5 row(s) returned 0.000 sec / 0.000 sec


MySQL Workbench
University
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sys
university
Tables
courses
enrollments
students
Views
Stored Procedures
Functions
SQL File 2*
Don't Limit
224
225 -- Question: Using a CTE, find students who are older than the average age.
226 WITH AvgAgeCTE AS (
227 SELECT AVG(Age) AS AgeValue FROM Students
228 )
229 SELECT Name, Age
230 FROM Students, AvgAgeCTE
231 WHERE Students.Age > AvgAgeCTE.AgeValue;
Result Grid
Name Age
Divya 24
Parhan 22
Grace 23
Jayden 24
Kiara 22
Maya 23
SQL Additions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Administration Schemas
Information
Schema: university
Object Info Session
Result 31
Read Only
Context Help Snippets
Output
Action Output
# Time Action Message Duration / Fetch
17 18:20:06 SELECT Name, Age FROM Students ORDER BY Age DESC 14 row(s) returned 0.000 sec / 0.000 sec
18 18:21:13 SELECT Name, Age FROM Students ORDER BY Age DESC, Name ASC 14 row(s) returned 0.000 sec / 0.000 sec
19 18:28:06 SELECT Name, Age FROM Students ORDER BY Age DESC LIMIT 5 5 row(s) returned 0.000 sec / 0.000 sec
20 18:33:41 WITH AvgAgeCTE AS ( SELECT AVG(Age) AS AgeValue FROM Students ) SELECT Name, Age FROM St... 6 row(s) returned 0.016 sec / 0.000 sec