

1. create the database universitydb

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under SCHEMAS, there is a database named 'universitydb'. In the Query 1 pane, the following SQL code is run:

```
1 • CREATE DATABASE UNIVERSITYDB;
2
3 • USE UniversityDB;
4
5 • CREATE TABLE Students (
6     StudentID INT PRIMARY KEY,
7     Name VARCHAR(50),
8     Age INT,
9     Major VARCHAR(50)
10 );
11
12 • CREATE TABLE Courses (
13     CourseID INT PRIMARY KEY,
14     CourseName VARCHAR(50),
15     Credits INT
16 );
```

In the Output pane, the results show:

Action	Time	Message	Duration / Fetch
CREATE DATABASE UNIVERSITYDB	1 21:57:24	1 row(s) affected	0.187 sec
USE UniversityDB	2 21:58:58	0 row(s) affected	0.000 sec

2. use universtydb

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under SCHEMAS, there is a database named 'universitydb'. In the Query 1 pane, the following SQL code is run:

```
1 • CREATE DATABASE UNIVERSITYDB;
2
3 • USE UniversityDB;
4
5 • CREATE TABLE Students (
6     StudentID INT PRIMARY KEY,
7     Name VARCHAR(50),
8     Age INT,
9     Major VARCHAR(50)
10 );
11
12 • CREATE TABLE Courses (
13     CourseID INT PRIMARY KEY,
14     CourseName VARCHAR(50),
15     Credits INT
16 );
```

In the Output pane, the results show:

Action	Time	Message	Duration / Fetch
CREATE DATABASE UNIVERSITYDB	1 21:57:24	1 row(s) affected	0.187 sec
USE UniversityDB	2 21:58:58	0 row(s) affected	0.000 sec

3.CREATE TABLE Students (

StudentID INT PRIMARY KEY,

Name VARCHAR(50),

Age INT,

Major VARCHAR(50)

');

The screenshot shows the MySQL Workbench interface. In the left sidebar, under 'SCHEMAS', the 'yashdb' schema is selected. The 'Tables' section contains 'Students' and 'Courses'. The 'Query 1' tab displays the SQL code for creating the database and tables:

```
1 • CREATE DATABASE UNIVERSITYDB;
2
3 • USE UniversityDB;
4
5 • CREATE TABLE Students (
6     StudentID INT PRIMARY KEY,
7     Name VARCHAR(50),
8     Age INT,
9     Major VARCHAR(50)
10 );
11
12 • CREATE TABLE Courses (
13     CourseID INT PRIMARY KEY,
14     CourseName VARCHAR(50),
15     Credits INT
16 );
```

The 'Output' tab shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	21:57:24	CREATE DATABASE UNIVERSITYDB	1 row(s) affected	0.187 sec
2	21:58:58	USE UniversityDB	0 row(s) affected	0.000 sec
3	21:59:17	CREATE TABLE Students (StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VARCHAR(50));	0 row(s) affected	0.031 sec

4.CREATE TABLE Courses (

CourseID INT PRIMARY KEY,

CourseName VARCHAR(50),

Credits INT

');

The screenshot shows the MySQL Workbench interface with three tabs open: Local instance MySQL80 (yashdb), Local instance MySQL80 (uni...), and Local instance MySQL80 (uni...). The central pane displays a SQL query window titled 'Query 1' containing the following code:

```

4
5 • CREATE TABLE Students (
6     StudentID INT PRIMARY KEY,
7     Name VARCHAR(50),
8     Age INT,
9     Major VARCHAR(50)
10    );
11
12 • CREATE TABLE Courses (
13     CourseID INT PRIMARY KEY,
14     CourseName VARCHAR(50),
15     Credits INT
16    );
17
18 • CREATE TABLE Enrollments (
19     EnrollmentID INT PRIMARY KEY,
20     ...
21     ...
22     ...
23 );

```

The 'Output' pane below shows the execution log:

Action	Time	Action	Message	Duration / Fetch
1	21:57:24	CREATE DATABASE UNIVERSITYDB	1 row(s) affected	0.187 sec
2	21:58:58	USE UniversityDB	0 row(s) affected	0.000 sec
3	21:59:17	CREATE TABLE Students (StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VARCHAR(50));	0 row(s) affected	0.031 sec
4	22:01:12	CREATE TABLE Courses (CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT);	0 row(s) affected	0.031 sec

5.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)

);

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
Filter objects
sys
yashdb
Tables
Views
Stored Procedures
Functions
Administration Schemas
Information
No object selected
Object Info Session
Type here to search
Query 1
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)
)
Output
Action Output
# Time Action Message Duration / Fetch
1 21:57:24 CREATE DATABASE UNIVERSITYDB 1 row(s) affected 0.187 sec
2 21:58:58 USE UniversityDB 0 row(s) affected 0.000 sec
3 21:59:17 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VARCHAR(50), Grade CHAR(2), FOREIGN KEY (StudentID) REFERENCES Students(StudentID), FOREIGN KEY (CourseID) REFERENCES Courses(CourseID) )
4 22:01:12 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT )
5 22:01:56 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
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets

```

6.INSERT INTO Students (StudentID, Name, Age, Major) VALUES

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
Filter objects
sys
yashdb
Tables
Views
Stored Procedures
Functions
Administration Schemas
Information
No object selected
Object Info Session
Type here to search
Query 1
Grade CHAR(2),
FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
);
INSERT INTO Students (StudentID, Name, Age, Major) VALUES
(1, 'Aarav', 20, 'Computer Science'),
(2, 'Isha', 22, 'Data Science'),
(3, 'Rohan', 21, 'Computer Science'),
(4, 'Sneha', 23, 'Data Science'),
(5, 'Kunal', 19, 'Mathematics'),
(6, 'Priya', 20, 'Computer Science'),
(7, 'Arjun', 24, 'Data Science'),
(8, 'Neha', 22, 'Computer Science'),
(9, 'Vikram', 21, 'Data Science'),
(10, 'Ananya', 23, 'Computer Science')
Output
Action Output
# Time Action Message Duration / Fetch
1 21:57:24 CREATE DATABASE UNIVERSITYDB 1 row(s) affected 0.187 sec
2 21:58:58 USE UniversityDB 0 row(s) affected 0.000 sec
3 21:59:17 CREATE TABLE Students ( StudentID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Major VARCHAR(50), Grade CHAR(2), FOREIGN KEY (StudentID) REFERENCES Students(StudentID), FOREIGN KEY (CourseID) REFERENCES Courses(CourseID) )
4 22:01:12 CREATE TABLE Courses ( CourseID INT PRIMARY KEY, CourseName VARCHAR(50), Credits INT )
5 22:01:56 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) )
6 22:02:38 INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarav', 20, 'Computer Science'), (2, 'Isha', 22, 'Data Science'), (3, 'Rohan', 21, 'Computer Science'), (4, 'Sneha', 23, 'Data Science'), (5, 'Kunal', 19, 'Mathematics'), (6, 'Priya', 20, 'Computer Science'), (7, 'Arjun', 24, 'Data Science'), (8, 'Neha', 22, 'Computer Science'), (9, 'Vikram', 21, 'Data Science'), (10, 'Ananya', 23, 'Computer Science') 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.000 sec
SQLAdditions
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets

```

7.SELECT * FROM Students;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
sys yashdb
Tables Views Stored Procedures Functions

Administration Schemas Information

No object selected

Query 1

```
35 (8, 'Neha', 22, 'Computer Science'),
36 (9, 'Vikram', 21, 'Data Science'),
37 (10, 'Ananya', 23, 'Computer Science');

38
39 • SELECT * FROM Students;
```

Result Grid | Filter Rows: Limit to 1000 rows | SQLAdditions

StudentID	Name	Age	Major
1	Aarav	20	Computer Science
2	Isha	22	Data Science
3	Rohan	21	Computer Science
4	Sneha	23	Data Science
5	Kunal	19	Mathematics
6	Priya	20	Computer Science
7	Arjun	24	Data Science
8	Neha	22	Computer Science
9	Vikram	21	Data Science
10	Ananya	23	Computer Science

Form Editor | Field Types | Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
2	21:58:58	USE UniversityDB	0 row(s) affected	0.000 sec
3	21:59:17	CREATE TABLE Students (StudentID INT PRIMARY KEY, Name VA...)	0 row(s) affected	0.031 sec
4	22:01:12	CREATE TABLE Courses (CourseID INT PRIMARY KEY, CourseName...)	0 row(s) affected	0.031 sec
5	22:01:56	CREATE TABLE Enrollments (EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, EnrollmentDate DATE, Grade...)	0 row(s) affected	0.047 sec
6	22:02:38	INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarav', 20, 'Computer Science')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
7	22:03:52	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

Type here to search ENG 22:05 US 07-09-2025

8.INSERT INTO Courses (CourseID, CourseName, Credits) VALUES

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
sys yashdb
Tables Views Stored Procedures Functions

Administration Schemas Information

No object selected

Query 1

```
32 (5, 'Kunal', 19, 'Mathematics'),
33 (6, 'Priya', 20, 'Computer Science'),
34 (7, 'Arjun', 24, 'Data Science'),
35 (8, 'Neha', 22, 'Computer Science'),
36 (9, 'Vikram', 21, 'Data Science'),
37 (10, 'Ananya', 23, 'Computer Science');

38
39 • SELECT * FROM Students;
40
41 • INSERT INTO Courses (CourseID, CourseName, Credits) VALUES
42 (101, 'Database Systems', 4),
43 (102, 'Algorithms', 4),
44 (103, 'Machine Learning', 3),
45 (104, 'Web Development', 3);
46
47
48 -
```

SQLAdditions | Action Output

#	Time	Action	Message	Duration / Fetch
4	22:01:12	CREATE TABLE Courses (CourseID INT PRIMARY KEY, CourseName...)	0 row(s) affected	0.031 sec
5	22:01:56	CREATE TABLE Enrollments (EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, EnrollmentDate DATE, Grade...)	0 row(s) affected	0.047 sec
6	22:02:38	INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarav', 20, 'Computer Science')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
7	22:03:52	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
8	22:05:37	INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4)	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.016 sec

Type here to search ENG 22:05 US 07-09-2025

9.SELECT * FROM Courses;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
sys yashdb
Tables Views Stored Procedures Functions

Query 1

```

40
41 • INSERT INTO Courses (CourseID, CourseName, Credits) VALUES
42     (101, 'Database Systems', 4),
43     (102, 'Algorithms', 4),
44     (103, 'Machine Learning', 3),
45     (104, 'Web Development', 3);
46
47 • SELECT * FROM Courses;
48

```

Result Grid | Filter Rows | Edit: | Export/Import: | Wrap Cell Content: | Result Grid

CourseID	CourseName	Credits
101	Database Systems	4
102	Algorithms	4
103	Machine Learning	3
104	Web Development	3
NULL	NULL	NULL

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.

Administration Schemas Information

No object selected

Object Info Session

Type here to search

Action Output

#	Time	Action	Message	Duration / Fetch
5	22:01:56	CREATE TABLE Enrollments (EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, Grade CHAR(1))	0 row(s) affected	0.047 sec
6	22:02:38	INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarav', 20, 'Computer Science')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
7	22:03:52	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
8	22:05:37	INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorithms', 4), (103, 'Machine Learning', 3), (104, 'Web Development', 3);	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.016 sec
9	22:06:09	SELECT * FROM Courses LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

22:06 ENG 22:06 US 07-09-2025

10.INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
sys yashdb
Tables Views Stored Procedures Functions

Query 1

```

44
45     (103, 'Machine Learning', 3),
46     (104, 'Web Development', 3);
47
48 • SELECT * FROM Courses;
49
50 • INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES
51     (1, 1, 101, 'A'),
52     (2, 2, 103, 'B'),
53     (3, 3, 102, 'A'),
54     (4, 4, 103, 'C'),
55     (5, 1, 102, 'B'),
56     (6, 6, 101, 'A-'),
57     (7, 7, 103, 'B+'),
58     (8, 8, 102, 'C+'),
59     (9, 9, 103, 'A'),
60     (10, 2, 101, 'B');

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
6	22:02:38	INSERT INTO Students (StudentID, Name, Age, Major) VALUES (1, 'Aarav', 20, 'Computer Science')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
7	22:03:52	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
8	22:05:37	INSERT INTO Courses (CourseID, CourseName, Credits) VALUES (101, 'Database Systems', 4), (102, 'Algorithms', 4), (103, 'Machine Learning', 3), (104, 'Web Development', 3);	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.016 sec
9	22:06:09	SELECT * FROM Courses LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
10	22:07:25	INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES (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');	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec

22:07 ENG 22:07 US 07-09-2025

11.SELECT * FROM Enrollments;

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are two tabs: "Local instance MySQL80 (yashdb)" and "Local instance MySQL80 (uni...)".

The left sidebar has a "SCHEMAS" section with "sys" and "yashdb" expanded, showing "Tables", "Views", "Stored Procedures", and "Functions".

The main area contains a "Query 1" tab with the following SQL query:

```

60
61
62 • SELECT * FROM Enrollments;
63
64

```

The results are displayed in a "Result Grid" table:

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

The status bar at the bottom shows "Type here to search" and the system tray indicates "ENG US 22:08 07-09-2025".

12.-- Question: How do you define a temporary table for departments?

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are two tabs: "Local instance MySQL80 (yashdb)" and "Local instance MySQL80 (uni...)".

The left sidebar has a "SCHEMAS" section with "sys" and "yashdb" expanded, showing "Tables", "Views", "Stored Procedures", and "Functions".

The main area contains a "Query 1" tab with the following SQL queries:

```

56 (7, 103, 'B+');
57 (8, 8, 102, 'C+');
58 (9, 9, 103, 'A');
59 (10, 2, 101, 'B');
60
61 • SELECT * FROM Enrollments;
62
63 -- =====
64 -- Executing All Provided Queries
65 -- =====
66
67 -- Question: How do you define a temporary table for departments?
68 • CREATE TABLE Departments (
69     DeptID INT PRIMARY KEY,
70     DeptName VARCHAR(50)
71 );

```

The results are displayed in a "Result Grid" table:

EnrollmentID	StudentID	CourseID	Grade
7	103	7	B+
8	8	8	C+
9	9	9	A
10	2	101	B

The status bar at the bottom shows "Type here to search" and the system tray indicates "ENG US 22:11 07-09-2025".

13.-- Question: How can you add a new 'Email' column to the Students table?

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
sys yashdb
Tables Views Stored Procedures Functions

Administration Schemas Information

No object selected

Query 1

```

62
63  -- =====
64  -- Executing All Provided Queries
65  --
66
67  -- Question: How do you define a temporary table for departments?
68 • CREATE TABLE Departments (
69     DeptID INT PRIMARY KEY,
70     DeptName VARCHAR(50)
71 );
72
73  -- Question: How can you add a new 'Email' column to the Students table?
74 • ALTER TABLE Students ADD Email VARCHAR(100);
75
76 • SELECT * FROM Students;
77
78  -- Question: How do you completely remove the Departments table?

```

Action Output

#	Time	Action	Message	Duration / Fetch
9	22:06:09	SELECT * FROM Courses LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
10	22:07:25	INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
11	22:08:10	SELECT * FROM Enrollments LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
12	22:09:24	CREATE TABLE Departments (DeptID INT PRIMARY KEY, DeptName VARCHAR(50)	0 row(s) affected	0.047 sec
13	22:11:40	ALTER TABLE Students ADD Email VARCHAR(100)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

Output

Object Info Session

Type here to search ENG 22:12 US 07-09-2025

14.SELECT * FROM Students;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
sys yashdb
Tables Views Stored Procedures Functions

Administration Schemas Information

No object selected

Query 1

```

74 • ALTER TABLE Students ADD Email VARCHAR(100);
75
76 • SELECT * FROM Students;
77
78

```

Result Grid

StudentID	Name	Age	Major	Email
1	Aarav	20	Computer Science	NULL
2	Isha	22	Data Science	NULL
3	Rohan	21	Computer Science	NULL
4	Sneha	23	Data Science	NULL
5	Kunal	19	Mathematics	NULL
6	Priya	20	Computer Science	NULL
7	Arjun	24	Data Science	NULL
8	Neha	22	Computer Science	NULL
9	Vikram	21	Data Science	NULL
10	Ananya	23	Computer Science	NULL

Action Output

#	Time	Action	Message	Duration / Fetch
10	22:07:25	INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade) VALUES	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.031 sec
11	22:08:10	SELECT * FROM Enrollments LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
12	22:09:24	CREATE TABLE Departments (DeptID INT PRIMARY KEY, DeptName VARCHAR(50)	0 row(s) affected	0.047 sec
13	22:11:40	ALTER TABLE Students ADD Email VARCHAR(100)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
14	22:14:45	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

Output

Object Info Session

Type here to search ENG 22:15 US 07-09-2025

15.Question: How do you completely remove the Departments table?

DROP TABLE Departments;

MySQL Workbench Schemas: sys, yashdb (Tables, Views, Stored Procedures, Functions)

Query 1:

```

74 • ALTER TABLE Students ADD Email VARCHAR(100);
75
76 • SELECT * FROM Students;
77
78 -- Question: How do you completely remove the Departments table?
79 • DROP TABLE Departments;_
80
81 -- Question: How do you add a rule to ensure students are at least 17?
82 • ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);
83
84
85 -- Question: How do you remove the age-checking rule?
86 • ALTER TABLE Students DROP CONSTRAINT AgeCheck;
87
88 -- Question: How do you update the major for student with ID 1 to 'Data Science'?
89 • UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;
90

```

Output:

#	Time	Action	Message	Duration / Fetch
11	22:08:10	SELECT * FROM Enrollments LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
12	22:09:24	CREATE TABLE Departments (DeptID INT PRIMARY KEY, DeptName VARCHAR(255))	0 row(s) affected	0.047 sec
13	22:11:40	ALTER TABLE Students ADD Email VARCHAR(100)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
14	22:14:45	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
15	22:16:06	DROP TABLE Departments	0 row(s) affected	0.031 sec

16.-- Question: How do you add a rule to ensure students are at least 17?

ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);

MySQL Workbench Schemas: sys, yashdb (Tables, Views, Stored Procedures, Functions)

Query 1:

```

74 • ALTER TABLE Students ADD Email VARCHAR(100);
75
76 • SELECT * FROM Students;
77
78 -- Question: How do you completely remove the Departments table?
79 • DROP TABLE Departments;_
80
81 -- Question: How do you add a rule to ensure students are at least 17?
82 • ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);_
83
84
85 -- Question: How do you remove the age-checking rule?
86 • ALTER TABLE Students DROP CONSTRAINT AgeCheck;
87
88 -- Question: How do you update the major for student with ID 1 to 'Data Science'?
89 • UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;
90

```

Output:

#	Time	Action	Message	Duration / Fetch
12	22:09:24	CREATE TABLE Departments (DeptID INT PRIMARY KEY, DeptName VARCHAR(255))	0 row(s) affected	0.047 sec
13	22:11:40	ALTER TABLE Students ADD Email VARCHAR(100)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
14	22:14:45	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
15	22:16:06	DROP TABLE Departments	0 row(s) affected	0.031 sec
16	22:17:33	ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.078 sec

17.-- Question: How do you remove the age-checking rule?

ALTER TABLE Students DROP CONSTRAINT AgeCheck;

MySQL Workbench interface showing the SQL Editor with the following queries:

```

74 • ALTER TABLE Students ADD Email VARCHAR(100);
75
76 • SELECT * FROM Students;
77
78 -- Question: How do you completely remove the Departments table?
79 • DROP TABLE Departments;
80
81 -- Question: How do you add a rule to ensure students are at least 17?
82 • ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);
83
84
85 -- Question: How do you remove the age-checking rule?
86 • ALTER TABLE Students DROP CONSTRAINT AgeCheck;
87
88 -- Question: How do you update the major for student with ID 1 to 'Data Science'?
89 • UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;
90

```

The Output pane shows the following execution results:

#	Time	Action	Message	Duration / Fetch
13	22:11:40	ALTER TABLE Students ADD Email VARCHAR(100)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
14	22:14:45	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
15	22:16:06	DROP TABLE Departments	0 row(s) affected	0.031 sec
16	22:17:33	ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.078 sec
17	22:18:14	ALTER TABLE Students DROP CONSTRAINT AgeCheck	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec

18.-- Question: How do you update the major for student with ID 1 to 'Data Science'?

UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;

MySQL Workbench interface showing the SQL Editor with the following queries:

```

75
76 • SELECT * FROM Students;
77
78 -- Question: How do you completely remove the Departments table?
79 • DROP TABLE Departments;
80
81 -- Question: How do you add a rule to ensure students are at least 17?
82 • ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17);
83
84
85 -- Question: How do you remove the age-checking rule?
86 • ALTER TABLE Students DROP CONSTRAINT AgeCheck;
87
88 -- Question: How do you update the major for student with ID 1 to 'Data Science'?
89 • UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;
90
91 • SELECT * FROM Students;

```

The Output pane shows the following execution results:

#	Time	Action	Message	Duration / Fetch
14	22:14:45	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
15	22:16:06	DROP TABLE Departments	0 row(s) affected	0.031 sec
16	22:17:33	ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.078 sec
17	22:18:14	ALTER TABLE Students DROP CONSTRAINT AgeCheck	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
18	22:18:59	UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec

19.SELECT * FROM Students;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
sys yashdb
Tables Views Stored Procedures Functions

Query 1: -- 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;

StudentID	Name	Age	Major	Email
1	Aarav	20	Data Science	NULL
2	Isha	22	Data Science	NULL
3	Rohan	21	Computer Science	NULL
4	Sheha	23	Data Science	NULL
5	Kunal	19	Mathematics	NULL
6	Priya	20	Computer Science	NULL
7	Arjun	24	Data Science	NULL
8	Neha	22	Computer Science	NULL
9	Vikram	21	Data Science	NULL
10	Ananya	23	Computer Science	NULL

Result Grid | Filter Rows | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Context Help Snippets

Action Output:

#	Time	Action	Message	Duration / Fetch
15	22:16:06	DROP TABLE Departments	0 row(s) affected	0.031 sec
16	22:17:33	ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.078 sec
17	22:18:14	ALTER TABLE Students DROP CONSTRAINT AgeCheck	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
18	22:18:59	UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
19	22:19:27	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

No object selected

Object Info Session

Type here to search ENG 22:21 US 07-09-2025

20.-- Question: How do you remove all student records for students younger than 18?

DELETE FROM Students WHERE Age < 18;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS
Filter objects
sys yashdb
Tables Views Stored Procedures Functions

Query 1: -- 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 Age < 18;

-- 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;

Action Output:

#	Time	Action	Message	Duration / Fetch
15	22:17:33	ALTER TABLE Students ADD CONSTRAINT AgeCheck CHECK (Age >= 17)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.078 sec
17	22:18:14	ALTER TABLE Students DROP CONSTRAINT AgeCheck	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
18	22:18:59	UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
19	22:19:27	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
20	22:21:50	DELETE FROM Students WHERE Age < 18	0 row(s) affected	0.000 sec

No object selected

Object Info Session

Type here to search ENG 22:21 US 07-09-2025

21.SELECT * FROM Students;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Local instance MySQL80 (yashdb) x Local instance MySQL80 (uni... x

Query 1

```

88 -- Question: How do you update the major for student with ID 1 to 'Data Science'?
89 • UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1;
90
91 • SELECT * FROM Students;
92

```

Result Grid

StudentID	Name	Age	Major	Email
1	Aarav	20	Data Science	NULL
2	Isha	22	Data Science	NULL
3	Rohan	21	Computer Science	NULL
4	Sneha	23	Data Science	NULL
5	Kunal	19	Mathematics	NULL
6	Priya	20	Computer Science	NULL
7	Arjun	24	Data Science	NULL
8	Neha	22	Computer Science	NULL
9	Vikram	21	Data Science	NULL
10	Ananya	23	Computer Science	NULL

Students 6 x

Action Output

#	Time	Action	Message	Duration / Fetch
17	22:18:14	ALTER TABLE Students DROP CONSTRAINT AgeCheck	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.015 sec
18	22:18:59	UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
19	22:19:27	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
20	22:21:50	DELETE FROM Students WHERE Age < 18	0 row(s) affected	0.000 sec
21	22:22:30	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

No object selected

Object Info Session

22.-- Question: Show the name and major of all students who are older than 19.

SELECT Name, Major FROM Students WHERE Age > 19;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Local instance MySQL80 (yashdb) x Local instance MySQL80 (uni... x

Query 1

```

97
98 -- Question: Show the name and major of all students who are older than 19.
99 • SELECT Name, Major FROM Students WHERE Age > 19;
100
101 -- Question: What is the average age of all students?

```

Result Grid

Name	Major
Aarav	Data Science
Isha	Data Science
Rohan	Computer Science
Sneha	Data Science
Priya	Computer Science
Arjun	Data Science
Neha	Computer Science
Vikram	Data Science
Ananya	Computer Science

Students 7 x

Action Output

#	Time	Action	Message	Duration / Fetch
18	22:18:59	UPDATE Students SET Major = 'Data Science' WHERE StudentID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec
19	22:19:27	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
20	22:21:50	DELETE FROM Students WHERE Age < 18	0 row(s) affected	0.000 sec
21	22:22:30	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
22	22:24:22	SELECT Name, Major FROM Students WHERE Age > 19 LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec

No object selected

Object Info Session

23.-- Question: What is the average age of all students?

SELECT AVG(Age) AS AvgAge FROM Students;

MySQL Workbench interface showing a query editor with the following SQL code:

```

98 -- Question: Show the name and major of all students who are older than 19.
99 • SELECT Name, Major FROM Students WHERE Age > 19;
100
101 -- Question: What is the average age of all students?
102 • SELECT AVG(Age) AS AvgAge FROM Students;
103
104 -- Question: Which majors have more than 5 students, and what is the count for each?
105 • SELECT Major, COUNT(*) AS StudentCount
106 FROM Students
107 GROUP BY Major

```

The results grid shows a single row with AvgAge = 21.5000.

24.-- Question: Which majors have more than 5 students, and what is the count for each?

SELECT Major, COUNT(*) AS StudentCount

FROM Students

GROUP BY Major

HAVING COUNT(*) > 5;

MySQL Workbench interface showing a query editor with the completed SQL code:

```

101 -- Question: What is the average age of all students?
102 • SELECT AVG(Age) AS AvgAge FROM Students;
103
104 -- Question: Which majors have more than 5 students, and what is the count for each?
105 • SELECT Major, COUNT(*) AS StudentCount
106 FROM Students
107 GROUP BY Major
108 HAVING COUNT(*) > 5;
109

```

The results grid shows the following data:

Major	StudentCount
Computer Science	10

The action output log shows the following entries:

#	Time	Action	Message	Duration / Fetch
19	22:19:27	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
20	22:21:50	DELETE FROM Students WHERE Age < 18	0 row(s) affected	0.000 sec
21	22:22:30	SELECT * FROM Students LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
22	22:24:22	SELECT Name, Major FROM Students WHERE Age > 19 LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
23	22:25:27	SELECT AVG(Age) AS AvgAge FROM Students LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
24	22:26:02	SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY M...	0 row(s) returned	0.016 sec / 0.000 sec

25.-- Question: List all information for students older than 20 and in 'Computer Science'.

`SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science';`

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab contains the following SQL code:

```

104 -- Question: Which majors have more than 5 students, and what is the count for each?
105 • SELECT Major, COUNT(*) AS StudentCount
106   FROM Students
107   GROUP BY Major
108   HAVING COUNT(*) > 5;
109
110 -- Question: List all information for students older than 20 and in 'Computer Science'
111 • SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science';
112
113 Question: How can you rank students based on their grades?

```

The 'Result Grid' shows the following data:

StudentID	Name	Age	Major	Email
3	Rohan	21	Computer Science	rohan@example.com
8	Neha	22	Computer Science	neha@example.com
10	Ananya	23	Computer Science	ananya@example.com
•	ROLLBACK	ROLLBACK	ROLLBACK	ROLLBACK

The 'Output' pane shows the following log entries:

Action	Time	Message	Duration / Fetch
SELECT * FROM Students LIMIT 0, 1000	21 22:23:30	10 row(s) returned	0.000 sec / 0.000 sec
SELECT Name, Major FROM Students WHERE Age > 19 LIMIT 0, 1000	22 22:24:22	9 row(s) returned	0.000 sec / 0.000 sec
SELECT AVG(Age) AS AvgAge FROM Students LIMIT 0, 1000	23 22:25:27	1 row(s) returned	0.016 sec / 0.000 sec
SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY M...	24 22:26:02	0 row(s) returned	0.016 sec / 0.000 sec
SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Sci...	25 22:26:40	3 row(s) returned	0.000 sec / 0.000 sec
SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Science'	26 22:27:23	10 row(s) returned	0.031 sec / 0.000 sec

26.-- Question: How can you rank students based on their grades?

`SELECT s.Name, e.Grade,`

`RANK() OVER (ORDER BY Grade ASC) AS RankInClass`

`FROM Enrollments e JOIN Students s ON e.StudentID = s.StudentID;`

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab contains the following SQL code:

```

113 -- Question: How can you rank students based on their grades?
114 • SELECT s.Name, e.Grade,
115   RANK() OVER (ORDER BY Grade ASC) AS RankInClass
116   FROM Enrollments e JOIN Students s ON e.StudentID = s.StudentID;
117

```

The 'Result Grid' shows the following data:

Name	Grade	RankInClass
Aarav	A	1
Rohan	A	1
Vikram	A	1
Priya	A-	4
Aarav	B	5
Isha	B	5
Isha	B	5
Arjun	B+	8
Sneha	C	9
Neha	C+	10

The 'Output' pane shows the following log entries:

Action	Time	Message	Duration / Fetch
SELECT Name, Major FROM Students WHERE Age > 19 LIMIT 0, 1000	22 22:24:22	9 row(s) returned	0.000 sec / 0.000 sec
SELECT AVG(Age) AS AvgAge FROM Students LIMIT 0, 1000	23 22:25:27	1 row(s) returned	0.016 sec / 0.000 sec
SELECT Major, COUNT(*) AS StudentCount FROM Students GROUP BY M...	24 22:26:02	0 row(s) returned	0.016 sec / 0.000 sec
SELECT * FROM Students WHERE Age > 20 AND Major = 'Computer Sci...	25 22:26:40	3 row(s) returned	0.000 sec / 0.000 sec
SELECT s.Name, e.Grade, RANK() OVER (ORDER BY Grade ASC) AS Ra...	26 22:27:23	10 row(s) returned	0.031 sec / 0.000 sec

27.-- Question: List student names and the courses they are enrolled in?

```

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;

```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to 'Local instance MySQL80 (yashdb)'. The main area displays a query window with the following SQL code:

```

-- Question: Show all students and their courses, including students not enrolled in a 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;

```

The results grid shows the following data:

Name	CourseName
Aarav	Database Systems
Aarav	Algorithms
Isha	Machine Learning
Isha	Database Systems
Rohan	Algorithms
Sneha	Machine Learning
Kunal	DBMS
Priya	Database Systems
Arjun	Machine Learning
Neha	Algorithms
Vikram	Machine Learning
Ananya	DBMS

On the right side of the interface, there is a 'Context Help' message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

28. Generate every possible combination of a student paired with a course.

```
SELECT s.Name, c.CourseName
```

```
FROM Students s CROSS JOIN Courses c;
```

MySQL Workbench Screenshot:

Query 1:

```

130
131 -- Question: Generate every possible combination of a student paired with a course.
132 • SELECT s.Name, c.CourseName
133 FROM Students s CROSS JOIN Courses c
134

```

Result Grid:

Name	CourseName
Aarav	Web Development
Aarav	Machine Learning
Aarav	Algorithms
Aarav	Database Systems
Isha	Web Development
Isha	Machine Learning
Isha	Algorithms
Isha	Database Systems
Rohan	Web Development
Rohan	Machine Learning
Rohan	Algorithms
Rohan	Database Systems
Sneha	Web Development
Sneha	Machine Learning
Sneha	Algorithms

Output:

#	Time	Action	Message	Duration / Fetch
27	22:28:50	SELECT s.Name, c.CourseName FROM Students s LEFT JOIN Enrollments ...	12 row(s) returned	0.000 sec / 0.000 sec
28	22:29:44	SELECT s.Name, c.CourseName FROM Students s CROSS JOIN Courses c	40 row(s) returned	0.000 sec / 0.000 sec

29.-- 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;

MySQL Workbench Screenshot:

Query 1:

```

135
136 -- Question: Find pairs of different students who are in the same major.
137 • SELECT s1.Name AS Student1, s2.Name AS Student2
138 FROM Students s1
139 JOIN Students s2 ON s1.Major = s2.Major AND s1.StudentID <> s2.StudentID;

```

Result Grid:

Student1	Student2
Vikram	Aarav
Arjun	Aarav
Sneha	Aarav
Isha	Aarav
Vikram	Isha
Arjun	Isha
Sneha	Isha
Aarav	Isha
Ananya	Rohan
Neha	Rohan
Priya	Rohan
Vikram	Sneha
Arjun	Sneha
Isha	Sneha
Aarav	Sneha

Output:

#	Time	Action	Message	Duration / Fetch
28	22:29:44	SELECT s.Name, c.CourseName FROM Students s CROSS JOIN Courses c	40 row(s) returned	0.000 sec / 0.000 sec
29	22:32:11	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;	32 row(s) returned	0.000 sec / 0.000 sec

30.-- 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
LIMIT 0, 1000;

```

LIMIT 0, 1000;

```

Query 1 x
141 • SELECT Major,
142     GROUP_CONCAT(Name ORDER BY Name SEPARATOR ',') AS Students
143 FROM Students
144 GROUP BY Major
145 LIMIT 0, 1000;

Result Grid | Filter Rows | Export: | Wrap Cell Content: |
Major       Students
Computer Science Ananya, Neha, Priya, Rohan
Data Science   Aarav, Arjun, Isha, Sneha, Vikram
Mathematics   Kunal

Result 15 x
Output
Action Output | Action | Message | Duration / Fetch
30 22:33:12 SELECT Major, STRING_AGG(Name, ',') AS Students FROM Students GR... Error Code: 1305. FUNCTION universitydb.STRING_AGG does not exist 0.016 sec
31 22:45:08 SELECT Major, GROUP_CONCAT(Name ORDER BY Name SEPARA... 3 row(s) returned 0.000 sec / 0.000 sec

Object Info Session
Type here to search ENG 22:45 07-09-2025

```

31.Find students who are older than the average student age.?

SELECT Name FROM Students

WHERE Age > (SELECT AVG(Age) FROM Students);

The screenshot shows the MySQL Workbench interface. In the top-left, there are two tabs: "Local instance MySQL80 (yashdb)" and "Local instance MySQL80 (uni...)".

The main area contains a "Query 1" tab with the following SQL code:

```
148 -- Question: Find students who are older than the average student age.  
149 • SELECT Name FROM Students  
150 WHERE Age > (SELECT AVG(Age) FROM Students);  
151  
152 -- Question: Get the names of students who have received a grade 'A' in any course.
```

Below the code is a "Result Grid" showing the following data:

Name
Isha
Sheha
Arjun
Neha
Ananya

On the right side of the interface, there is a "SQLAdditions" panel with the message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

At the bottom, the Windows taskbar is visible with icons for File Explorer, Edge, Task View, and others. The system tray shows the date and time as "07-09-2025".

32. 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'

);

MySQL Workbench Screenshot:

```

154 WHERE EXISTS (
155     SELECT 1 FROM Enrollments e
156     WHERE e.StudentID = s.StudentID AND e.Grade = 'A'
157 )
158
Result Grid | Filter Rows: [ ] Export: [ ] Wrap Cell Content: [ ]
Name
▶ Aarav
▶ Rohan
▶ Vikram

```

Output window:

Action	Time	Message	Duration / Fetch
32	22:46:26	SELECT Name FROM Students WHERE Age > (SELECT AVG(Age) FROM Students)	0.000 sec / 0.000 sec
33	22:48:11	SELECT Name FROM Students s WHERE EXISTS (SELECT 1 FROM Enrollments e WHERE e.StudentID = s.StudentID AND e.Grade = 'A')	0.000 sec / 0.000 sec

33.-- Question: Display each major and its calculated average age.

`SELECT Major, AvgAge`

`FROM (SELECT Major, AVG(Age) AS AvgAge FROM Students GROUP BY Major) AS t;`

MySQL Workbench Screenshot:

```

159 -- Question: Display each major and its calculated average age.
160 • SELECT Major, AvgAge
161   FROM (SELECT Major, AVG(Age) AS AvgAge FROM Students GROUP BY Major) AS t;
162
163 -- Question: Create a single list of all student names and course names.

```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []
Major AvgAge
▶ Data Science 22.0000
▶ Computer Science 21.5000
▶ Mathematics 19.0000

Output window:

Action	Time	Message	Duration / Fetch
33	22:48:11	SELECT Name FROM Students s WHERE EXISTS (SELECT 1 FROM Enrollments e WHERE e.StudentID = s.StudentID AND e.Grade = 'A')	0.000 sec / 0.000 sec
34	22:49:03	SELECT Major, AvgAge FROM (SELECT Major, AVG(Age) AS AvgAge FR...	0.000 sec / 0.000 sec

34. Create a single list of all student names and course names.

`SELECT Name FROM Students`

`UNION`

SELECT CourseName FROM Courses;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with tables like Students, Enrollments, and Courses.
- Query Editor (Query 1):** Contains the SQL query:

```
164 •  SELECT Name FROM Students
165 UNION
166 SELECT CourseName FROM Courses;
167
168 -- Question: Find student IDs present in both Students and Enrollments tables.
```
- Result Grid:** Displays the results of the query, showing course names: Aarav, Isha, Rohan, Sneha, Kunal, Priya, Arjun, Neha, Vikram, Ananya, Datab.., Algorit.., Machin.., Web D...
- Output:** Shows the action log with two entries:

#	Time	Action	Message	Duration / Fetch
34	22:49:03	SELECT Major, AvgAge FROM (SELECT Major, AVG(Age) AS AvgAge FR...	3 row(s) returned	0.000 sec / 0.000 sec
35	22:49:49	SELECT Name FROM Students UNION SELECT CourseName FROM Cour...	14 row(s) returned	0.000 sec / 0.000 sec

35.-- Question: Find student IDs present in both Students and Enrollments tables.

SELECT StudentID

FROM Students

WHERE StudentID IN (SELECT StudentID FROM Enrollments);

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with tables like Students, Enrollments, and Courses.
- Query Editor (Query 1):** Contains the SQL query:

```
167
168 -- Question: Find student IDs present in both Students and Enrollments tables.
169 •  SELECT StudentID
170   FROM Students
171 WHERE StudentID IN (SELECT StudentID FROM Enrollments);
172
```
- Result Grid:** Displays the results of the query, showing student IDs: 1, 2, 3, 4, 6, 7, 8, 9, NULL.
- Output:** Shows the action log with two entries:

#	Time	Action	Message	Duration / Fetch
35	22:49:49	SELECT Name FROM Students UNION SELECT CourseName FROM Cour...	14 row(s) returned	0.000 sec / 0.000 sec
36	22:53:10	SELECT StudentID FROM Students WHERE StudentID IN (SELECT Stud...	8 row(s) returned	0.000 sec / 0.000 sec

36.-- Question: Find students registered but not enrolled in any course.

```

SELECT Name
FROM Students
WHERE StudentID NOT IN (SELECT StudentID FROM Enrollments);

```

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is run:

```

173 -- Question: Find students registered but not enrolled in any course.
174
175 • SELECT Name
176 FROM Students
177 WHERE StudentID NOT IN (SELECT StudentID FROM Enrollments);
178

```

The Result Grid displays the names of two students: Kunal and Ananya.

In the Output pane, the log shows two actions:

Action	Time	Message	Duration / Fetch
36	22:53:10	SELECT StudentID FROM Students WHERE StudentID NOT IN (SELECT StudentID FROM Enrollments); 8 row(s) returned	0.000 sec / 0.000 sec
37	22:56:29	SELECT Name FROM Students WHERE StudentID NOT IN (SELECT StudentID FROM Enrollments); 2 row(s) returned	0.031 sec / 0.000 sec

37.-- Question: How to perform several changes as a single atomic operation?

START TRANSACTION;

INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade)

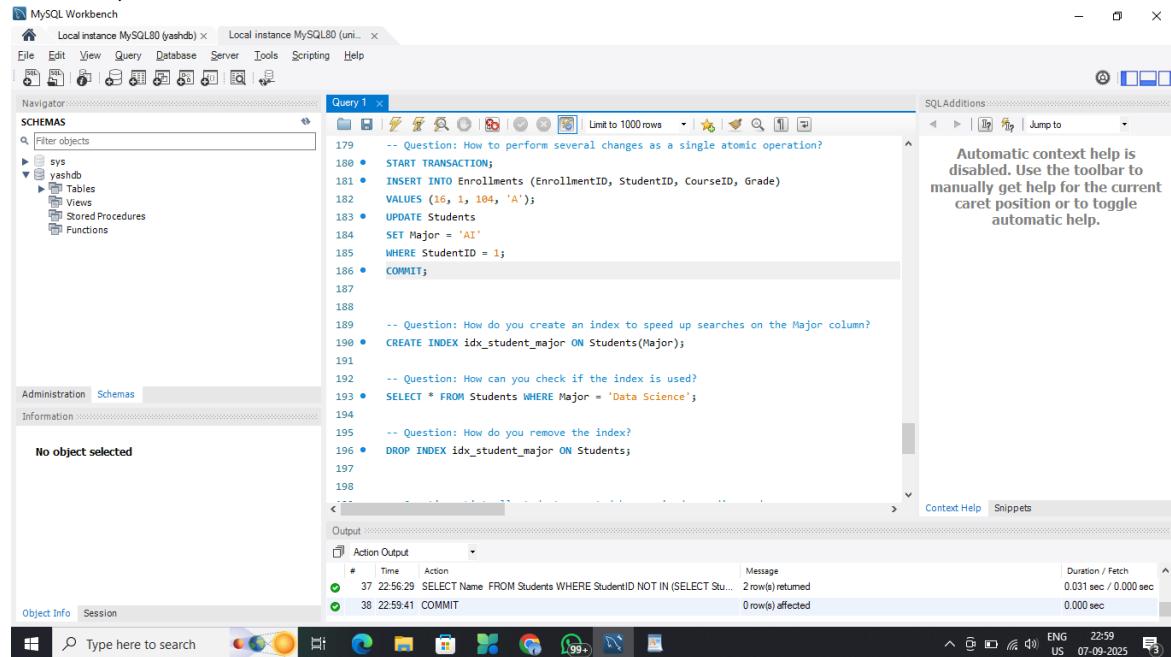
VALUES (16, 1, 104, 'A');

UPDATE Students

SET Major = 'AI'

WHERE StudentID = 1;

COMMIT;



The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is visible:

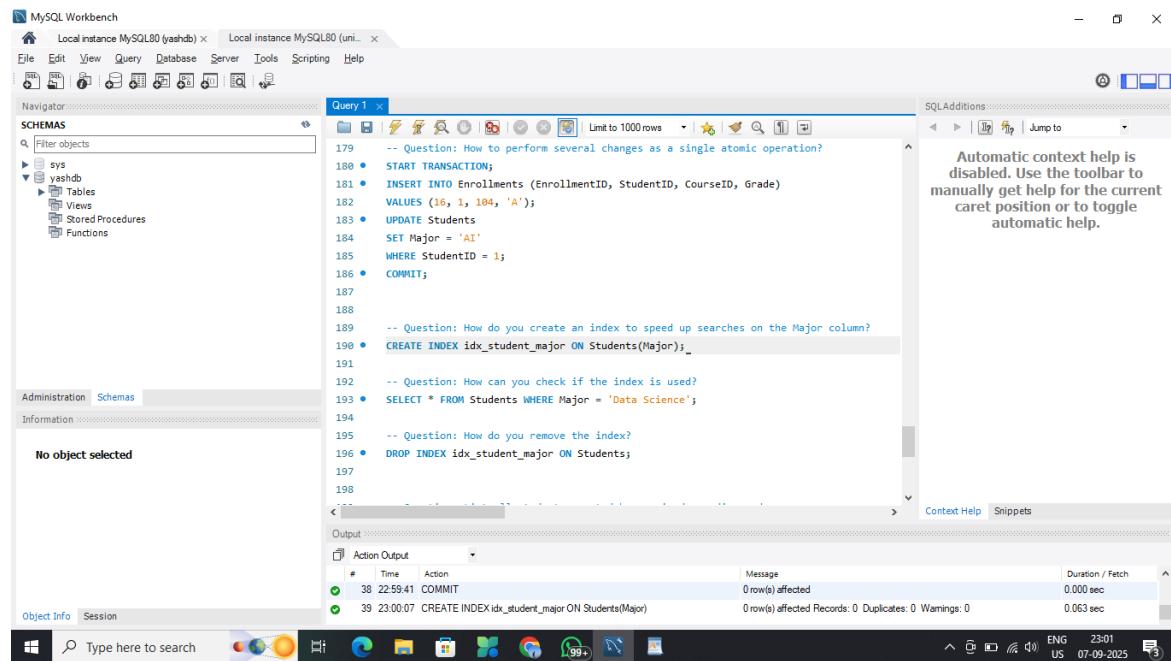
```
-- Question: How to perform several changes as a single atomic operation?
179 • START TRANSACTION;
180 • INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade)
181     VALUES (16, 1, 104, 'A');
182 • UPDATE Students
183     SET Major = 'AI'
184     WHERE StudentID = 1;
185 • COMMIT;
186
187
188
189 -- Question: How do you create an index to speed up searches on the Major column?
190 • CREATE INDEX idx_student_major ON Students(Major);
191
192 -- Question: How can you check if the index is used?
193 • SELECT * FROM Students WHERE Major = 'Data Science';
194
195 -- Question: How do you remove the index?
196 • DROP INDEX idx_student_major ON Students;
197
198
```

In the Output pane, the results of the last two statements are shown:

#	Time	Action	Message	Duration / Fetch
37	22:56:29	SELECT Name FROM Students WHERE StudentID NOT IN (SELECT Stu...	2 row(s) returned	0.031 sec / 0.000 sec
38	22:59:41	COMMIT	0 row(s) affected	0.000 sec

38. Question: How do you create an index to speed up searches on the Major column?

CREATE INDEX idx_student_major ON Students(Major);



The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is visible:

```
-- Question: How to perform several changes as a single atomic operation?
179 • START TRANSACTION;
180 • INSERT INTO Enrollments (EnrollmentID, StudentID, CourseID, Grade)
181     VALUES (16, 1, 104, 'A');
182 • UPDATE Students
183     SET Major = 'AI'
184     WHERE StudentID = 1;
185 • COMMIT;
186
187
188
189 -- Question: How do you create an index to speed up searches on the Major column?
190 • CREATE INDEX idx_student_major ON Students(Major);
191
192 -- Question: How can you check if the index is used?
193 • SELECT * FROM Students WHERE Major = 'Data Science';
194
195 -- Question: How do you remove the index?
196 • DROP INDEX idx_student_major ON Students;
197
198
```

In the Output pane, the results of the last two statements are shown:

#	Time	Action	Message	Duration / Fetch
38	22:59:41	COMMIT	0 row(s) affected	0.000 sec
39	23:00:07	CREATE INDEX idx_student_major ON Students(Major)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec

39.Question: How can you check if the index is used?

SELECT * FROM Students WHERE Major = 'Data Science';

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Local instance MySQL80 (yashdb) x Local instance MySQL80 (uni... x

Query 1

```

190 • CREATE INDEX idx_student_major ON Students(Major);
191
192 -- Question: How can you check if the index is used?
193 • SELECT * FROM Students WHERE Major = 'Data Science';
194
195 -- Question: How do you remove the index?

```

Result Grid | Filter Rows | Edit: | Export/Import: | Wrap Cell Content:

StudentID	Name	Age	Major	Email
1	Aarav	20	Data Science	NULL
2	Isha	22	Data Science	NULL
4	Sneha	23	Data Science	NULL
7	Arjun	24	Data Science	NULL
9	Vikram	21	Data Science	NULL
NULL	NULL	NULL	NULL	NULL

Administration Schemas

No object selected

Students 22 x

Action Output

#	Time	Action	Message	Duration / Fetch
39	23:00:07	CREATE INDEX idx_student_major ON Students(Major)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
40	23:02:04	SELECT * FROM Students WHERE Major = 'Data Science' LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

40.- Question: How do you remove the index?

DROP INDEX idx_student_major ON Students;

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Local instance MySQL80 (yashdb) x Local instance MySQL80 (uni... x

Query 1

```

190 • CREATE INDEX idx_student_major ON Students(Major);
191
192 -- Question: How can you check if the index is used?
193 • SELECT * FROM Students WHERE Major = 'Data Science';
194
195 -- Question: How do you remove the index?
196 • DROP INDEX idx_student_major ON Students;
197
198
199 -- Question: List all students, sorted by age in descending order.
200 • SELECT Name, Age
201   FROM Students
202   ORDER BY Age DESC;
203
204 -- Question: List students, sorted first by age (descending), then by name (ascending)
205 • SELECT Name, Age
206   FROM Students
207   ORDER BY Age DESC, Name ASC;
208
209 -- Question: Show the top 5 oldest students.

```

Output

#	Time	Action	Message	Duration / Fetch
40	23:02:04	SELECT * FROM Students WHERE Major = 'Data Science' LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
41	23:03:20	DROP INDEX idx_student_major ON Students	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

Object Info Session

Type here to search

41.Question: List all students, sorted by age in descending order.

SELECT Name, Age

FROM Students

ORDER BY Age DESC;

MySQL Workbench Screenshot:

Query 1 (SQL):

```

191
192 -- Question: How can you check if the index is used?
193 •   SELECT * FROM Students WHERE Major = 'Data Science';
194
195 -- Question: How do you remove the index?
196 •   DROP INDEX idx_student_major ON Students;
197
198
199 -- Question: List all students, sorted by age in descending order.
200 •   SELECT Name, Age
201     FROM Students
202     ORDER BY Age DESC;
203

```

Result Grid (Output):

Name	Age
Arjun	24
Sneha	23
Ananya	23
Isha	22
Neha	22

Action Output (Output):

Action	Time	Message	Duration / Fetch
DROP INDEXidx_student_major ON Students	41 23:03:20	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
SELECT Name, Age FROM Students ORDER By Age DESC LIMIT 0, 1000	42 23:03:41	10 row(s) returned	0.000 sec / 0.000 sec

42. Question: List students, sorted first by age (descending), then by name (ascending).

SELECT Name, Age

FROM Students

ORDER BY Age DESC, Name ASC;

MySQL Workbench Screenshot:

Query 1 (SQL):

```

197
198
199 -- Question: List all students, sorted by age in descending order.
200 •   SELECT Name, Age
201     FROM Students
202     ORDER BY Age DESC;
203
204 -- Question: List students, sorted first by age (descending), then by name (ascending)
205 •   SELECT Name, Age
206     FROM Students
207     ORDER BY Age DESC, Name ASC;
208
209 -- Question: Show the top 5 oldest students.

```

Result Grid (Output):

Name	Age
Ananya	23
Sneha	23
Isha	22
Neha	22
Arjun	24

Action Output (Output):

Action	Time	Message	Duration / Fetch
SELECT Name, Age FROM Students ORDER By Age DESC LIMIT 0, 1000	42 23:03:41	10 row(s) returned	0.000 sec / 0.000 sec
SELECT Name, Age FROM Students ORDER By Age DESC, Name ASC LI...	43 23:04:02	10 row(s) returned	0.015 sec / 0.000 sec

43.- Question: Show the top 5 oldest students.

SELECT Name, Age

FROM Students

ORDER BY Age DESC

LIMIT 5;

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab contains the following SQL code:

```
203 -- Question: List students, sorted first by age (descending), then by name (ascending)
204 • SELECT Name, Age
205   FROM Students
206   ORDER BY Age DESC, Name ASC;
207
208
209 -- Question: Show the top 5 oldest students.
210 • SELECT Name, Age
211   FROM Students
212   ORDER BY Age DESC
213   LIMIT 5;
214
215 -- Question: Using a CTE, find students who are older than the average age.
```

The results grid shows the following data:

Name	Age
Arjun	24
Ananya	23
Sneha	23
Isha	22
Neha	22

The status bar at the bottom right indicates the session is 'Read Only'.

44. Question: Using a CTE, find students who are older than the average age.

WITH AvgAgeCTE AS (

SELECT AVG(Age) AS AgeValue FROM Students

)

SELECT Name, Age

FROM Students, AvgAgeCTE

WHERE Students.Age > AvgAgeCTE.AgeValue;

