

Name : RAMYA R

Company: CODTECH IT SOLUTIONS

Intern ID : CT08DS7025

Domain : SQL

Duration: August to September 2024

Mentor : Muzammil Ahmed

STUDENT DATABASE MANAGEMENT

Creating a database to manage student records is a great way to practice relational database design and SQL queries. Below is a structured approach to designing the database, including table definitions and sample SQL queries.

Database Design

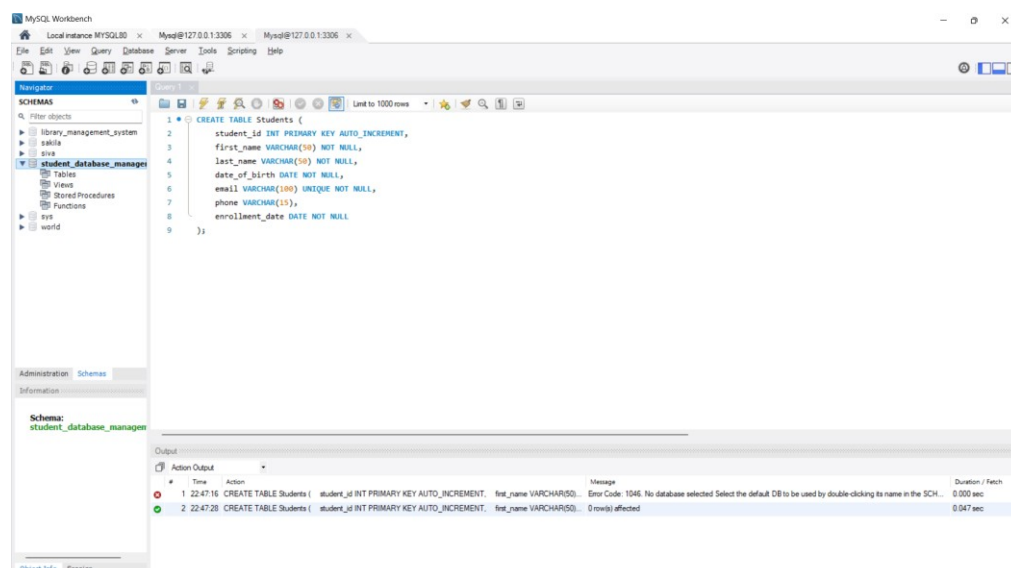
1. Tables Overview :

We'll create three main tables:

- **Students:** To store personal details of the students.
- **Courses:** To store information about the courses offered.
- **Enrollments:** To manage the relationship between students and courses, including grades.

2. Table Definitions

Students Table :



[illegible]

The screenshot displays the MySQL Workbench interface. The top toolbar contains icons for File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The Navigator pane on the left shows the 'student_database_managem' schema selected. The central Query Editor contains the following SQL code:

```
1 CREATE TABLE Enrollments (
2     enrollment_id INT PRIMARY KEY AUTO_INCREMENT,
3     student_id INT,
4     course_id INT,
5     grade CHAR(2),
6     enrollment_date DATE NOT NULL,
7     FOREIGN KEY (student_id) REFERENCES Students(student_id),
8     FOREIGN KEY (course_id) REFERENCES Courses(course_id)
9 );
```

The Output pane at the bottom shows the execution results of the SQL statements. The first statement, 'CREATE TABLE Enrollments', failed with Error Code 1046: No database selected. The subsequent statements, 'CREATE TABLE Students', 'CREATE TABLE Courses', and 'CREATE TABLE Enrollments', were executed successfully.

#	Action	Message	Duration / Fetch
1	22:47:16 CREATE TABLE Enrollments (student_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50)...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
2	22:47:28 CREATE TABLE Students (student_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50)...	0 row(s) affected	0.047 sec
3	22:48:29 CREATE TABLE Courses (course_id INT PRIMARY KEY AUTO_INCREMENT, course_name VARCHAR(1...	0 row(s) affected	0.016 sec
4	22:49:11 CREATE TABLE Enrollments (enrollment_id INT PRIMARY KEY AUTO_INCREMENT, student_id INT, c...	0 row(s) affected	0.031 sec

```
select * from Enrollments;
```

MySQL Workbench

Local instance MYSQL80 x MySQL@127.0.0.1:3306 x MySQL@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- library_management_system
- sakila
- siva
- student_database_managem...
- sys
- world

student_database_managem...

Tables

Views

Stored Procedures

Functions

sys

world

Administration Schemas

Information

Schema: student_database_managem...

Query 1

```

1 select * from Students;
2 select * from Courses;
3 select * from Enrollments;

```

Result Grid

enrollment_id	student_id	course_id	grade	enrollment_date
1	1	1	1	2008-09-01

Students 5 Courses 6 Enrollments 7 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
6	22:50:51	select * from Students LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
7	22:50:51	select * from Courses LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
8	22:50:51	select * from Enrollments LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
9	22:51:11	select * from Students LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
10	22:51:11	select * from Courses LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
11	22:51:11	select * from Enrollments LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Data Insertion :

Inserting Sample Data :

MySQL Workbench

Local instance MYSQL80 x MySQL@127.0.0.1:3306 x MySQL@127.0.0.1:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- library_management_system
- sakila
- siva
- student_database_managem...
- sys
- world

student_database_managem...

Tables

Views

Stored Procedures

Functions

sys

world

Administration Schemas

Information

Schema: student_database_managem...

Query 1

```

1 INSERT INTO Students (first_name, last_name, date_of_birth, email, phone, enrollment_date) VALUES
2 ('John', 'Doe', '2000-01-15', 'john.doe@example.com', '123-456-7890', '2020-09-01'),
3 ('Jane', 'Smith', '1999-05-22', 'jane.smith@example.com', '098-765-4321', '2020-09-01'),
4 ('Alice', 'Johnson', '2001-03-10', 'alice.johnson@example.com', '234-567-8901', '2021-01-15'),
5 ('Bob', 'Brown', '2000-11-30', 'bob.brown@example.com', '345-678-9012', '2020-09-01'),
6 ('Charlie', 'Davis', '1998-07-25', 'charlie.davis@example.com', '456-789-0123', '2019-09-01'),
7 ('Diana', 'Wilson', '1999-12-05', 'diana.wilson@example.com', '567-890-1234', '2020-09-01'),
8 ('Ethan', 'Martinez', '2002-02-18', 'ethan.martinez@example.com', '678-901-2345', '2021-01-15'),
9 ('Fiona', 'Garcia', '2000-04-12', 'fiona.garcia@example.com', '789-012-3456', '2020-09-01'),
10 ('George', 'Hernandez', '1999-08-20', 'george.hernandez@example.com', '890-123-4567', '2020-09-01'),
11 ('Hannah', 'Lopez', '2001-06-15', 'hannah.lopez@example.com', '901-234-5678', '2021-01-15');

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
22	22:58:05	create database STUDENT_DATABASE_MANAGEMENT	1 row(s) affected	0.000 sec
23	22:58:23	use STUDENT_DATABASE_MANAGEMENT	0 row(s) affected	0.000 sec
24	22:58:41	CREATE TABLE Students (student_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(45), last_name VARCHAR(45), date_of_birth DATE, email VARCHAR(90), phone VARCHAR(20), enrollment_date DATE)	0 row(s) affected	0.031 sec
25	22:58:55	CREATE TABLE Courses (course_id INT PRIMARY KEY AUTO_INCREMENT, course_name VARCHAR(45), credits INT)	0 row(s) affected	0.015 sec
26	22:59:08	CREATE TABLE Enrollments (enrollment_id INT PRIMARY KEY AUTO_INCREMENT, student_id INT, course_id INT, grade VARCHAR(2), enrollment_date DATE)	0 row(s) affected	0.031 sec
27	22:59:44	INSERT INTO Students first_name, last_name, date_of_birth, email, phone, enrollment_date) VALUES (John, Doe, 2000-01-15, john.doe@example.com, 123-456-7890, 2020-09-01), (Jane, Smith, 1999-05-22, jane.smith@example.com, 098-765-4321, 2020-09-01), (Alice, Johnson, 2001-03-10, alice.johnson@example.com, 234-567-8901, 2021-01-15), (Bob, Brown, 2000-11-30, bob.brown@example.com, 345-678-9012, 2020-09-01), (Charlie, Davis, 1998-07-25, charlie.davis@example.com, 456-789-0123, 2019-09-01), (Diana, Wilson, 1999-12-05, diana.wilson@example.com, 567-890-1234, 2020-09-01), (Ethan, Martinez, 2002-02-18, ethan.martinez@example.com, 678-901-2345, 2021-01-15), (Fiona, Garcia, 2000-04-12, fiona.garcia@example.com, 789-012-3456, 2020-09-01), (George, Hernandez, 1999-08-20, george.hernandez@example.com, 890-123-4567, 2020-09-01), (Hannah, Lopez, 2001-06-15, hannah.lopez@example.com, 901-234-5678, 2021-01-15);	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec

Object Info Session

✓ Students Table has been created.

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'student_database_managen' schema. The 'Query' pane shows a query: `select * from Students`. The 'Result Grid' displays 10 rows of student data. The 'Output' pane shows the 'Action Output' log, which includes the following entries:

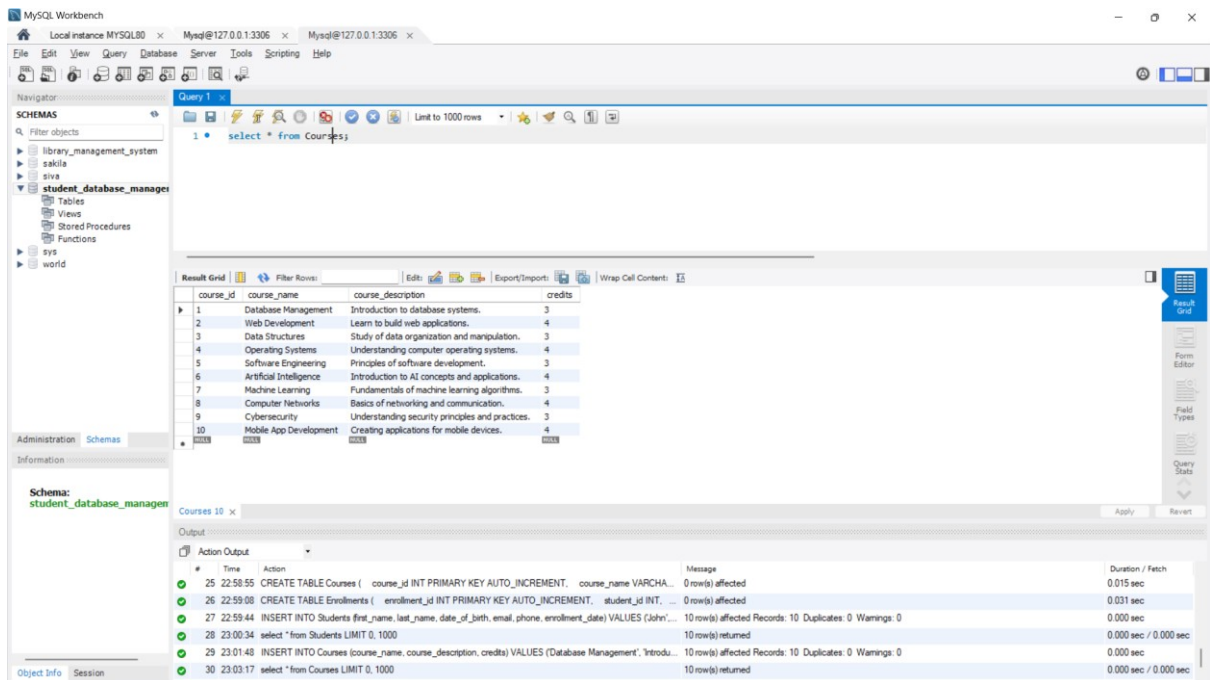
#	Time	Action	Message	Duration / Fetch
23	22:58:23	use STUDENT_DATABASE_MANAGEMENT	0 row(s) affected	0.000 sec
24	22:58:41	CREATE TABLE Students (student_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50), date_of_birth DATE, email VARCHAR(100), phone VARCHAR(20), enrollment_date DATE)	0 row(s) affected	0.031 sec
25	22:58:55	CREATE TABLE Courses (course_id INT PRIMARY KEY AUTO_INCREMENT, course_name VARCHAR(100), course_description VARCHAR(255), credits INT)	0 row(s) affected	0.015 sec
26	22:59:08	CREATE TABLE Enrollments (enrollment_id INT PRIMARY KEY AUTO_INCREMENT, student_id INT, course_id INT)	0 row(s) affected	0.031 sec
27	22:59:44	INSERT INTO Students first_name, last_name, date_of_birth, email, phone, enrollment_date VALUES (John, Doe, 2000-01-15, john.doe@example.com, 123-456-7890, 2020-09-01)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
28	23:00:34	select * from Students LIMIT 0.1000	10 row(s) returned	0.000 sec / 0.000 sec

Inserting Sample Courses:

The screenshot shows the MySQL Workbench interface. The 'Query' pane shows a query: `INSERT INTO Courses (course_name, course_description, credits) VALUES ('Database Management', 'Introduction to database systems.', 3), ('Web Development', 'Learn to build web applications.', 4), ('Data Structures', 'Study of data organization and manipulation.', 3), ('Operating Systems', 'Understanding computer operating systems.', 4), ('Software Engineering', 'Principles of software development.', 3), ('Artificial Intelligence', 'Introduction to AI concepts and applications.', 4), ('Machine Learning', 'Fundamentals of machine learning algorithms.', 3), ('Computer Networks', 'Basics of networking and communication.', 4), ('Cybersecurity', 'Understanding security principles and practices.', 3), ('Mobile App Development', 'Creating applications for mobile devices.', 4)`. The 'Output' pane shows the 'Action Output' log, which includes the following entries:

#	Time	Action	Message	Duration / Fetch
24	22:58:41	CREATE TABLE Students (student_id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50), date_of_birth DATE, email VARCHAR(100), phone VARCHAR(20), enrollment_date DATE)	0 row(s) affected	0.031 sec
25	22:58:55	CREATE TABLE Courses (course_id INT PRIMARY KEY AUTO_INCREMENT, course_name VARCHAR(100), course_description VARCHAR(255), credits INT)	0 row(s) affected	0.015 sec
26	22:59:08	CREATE TABLE Enrollments (enrollment_id INT PRIMARY KEY AUTO_INCREMENT, student_id INT, course_id INT)	0 row(s) affected	0.031 sec
27	22:59:44	INSERT INTO Students first_name, last_name, date_of_birth, email, phone, enrollment_date VALUES (John, Doe, 2000-01-15, john.doe@example.com, 123-456-7890, 2020-09-01)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
28	23:00:34	select * from Students LIMIT 0.1000	10 row(s) returned	0.000 sec / 0.000 sec
29	23:01:48	INSERT INTO Courses (course_name, course_description, credits) VALUES ('Database Management', 'Introduction to database systems.', 3), ('Web Development', 'Learn to build web applications.', 4), ('Data Structures', 'Study of data organization and manipulation.', 3), ('Operating Systems', 'Understanding computer operating systems.', 4), ('Software Engineering', 'Principles of software development.', 3), ('Artificial Intelligence', 'Introduction to AI concepts and applications.', 4), ('Machine Learning', 'Fundamentals of machine learning algorithms.', 3), ('Computer Networks', 'Basics of networking and communication.', 4), ('Cybersecurity', 'Understanding security principles and practices.', 3), ('Mobile App Development', 'Creating applications for mobile devices.', 4)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec

✓ Courses Table has been created:

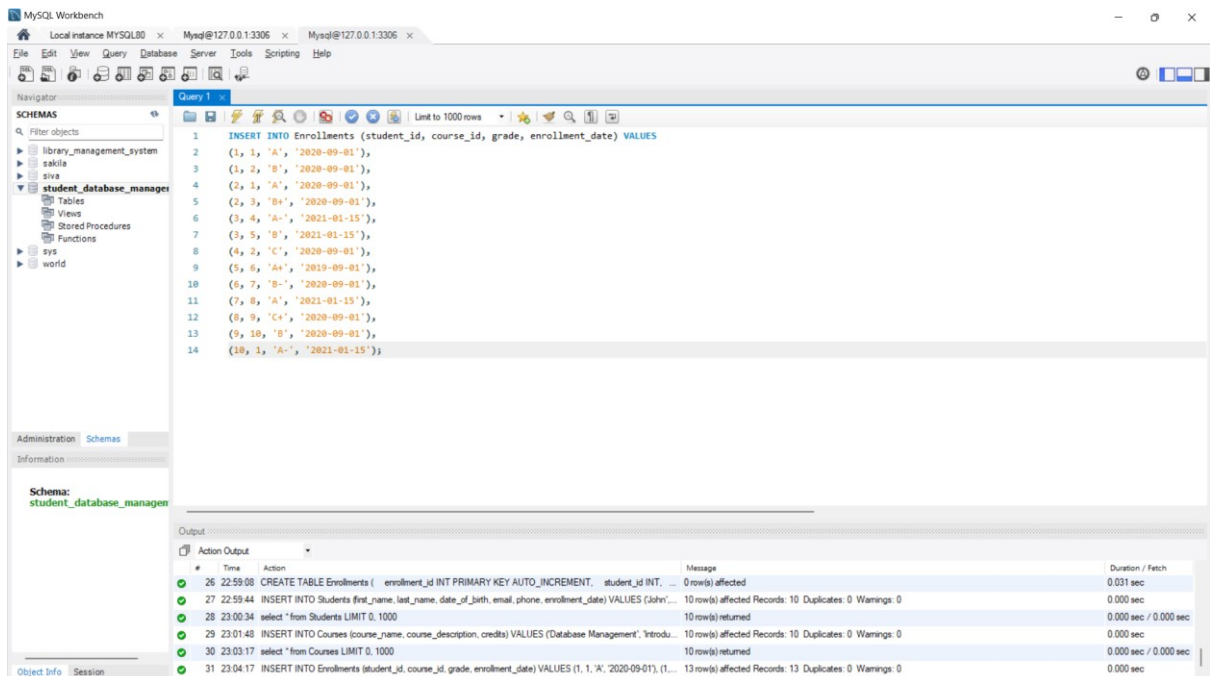


MySQL Workbench interface showing the 'Courses' table in the 'student_database_managen' schema. The table structure is as follows:

course_id	course_name	course_description	credits
1	Database Management	Introduction to database systems.	3
2	Web Development	Learn to build web applications.	4
3	Data Structures	Study of data organization and manipulation.	3
4	Operating Systems	Understanding computer operating systems.	4
5	Software Engineering	Principles of software development.	3
6	Artificial Intelligence	Introduction to AI concepts and applications.	4
7	Machine Learning	Fundamentals of machine learning algorithms.	3
8	Computer Networks	Basics of networking and communication.	4
9	Cybersecurity	Understanding security principles and practices.	3
10	Mobile App Development	Creating applications for mobile devices.	4

The 'Courses' table is selected in the 'student_database_managen' schema. The 'course_id' column is the primary key.

Inserting Sample Enrollments :

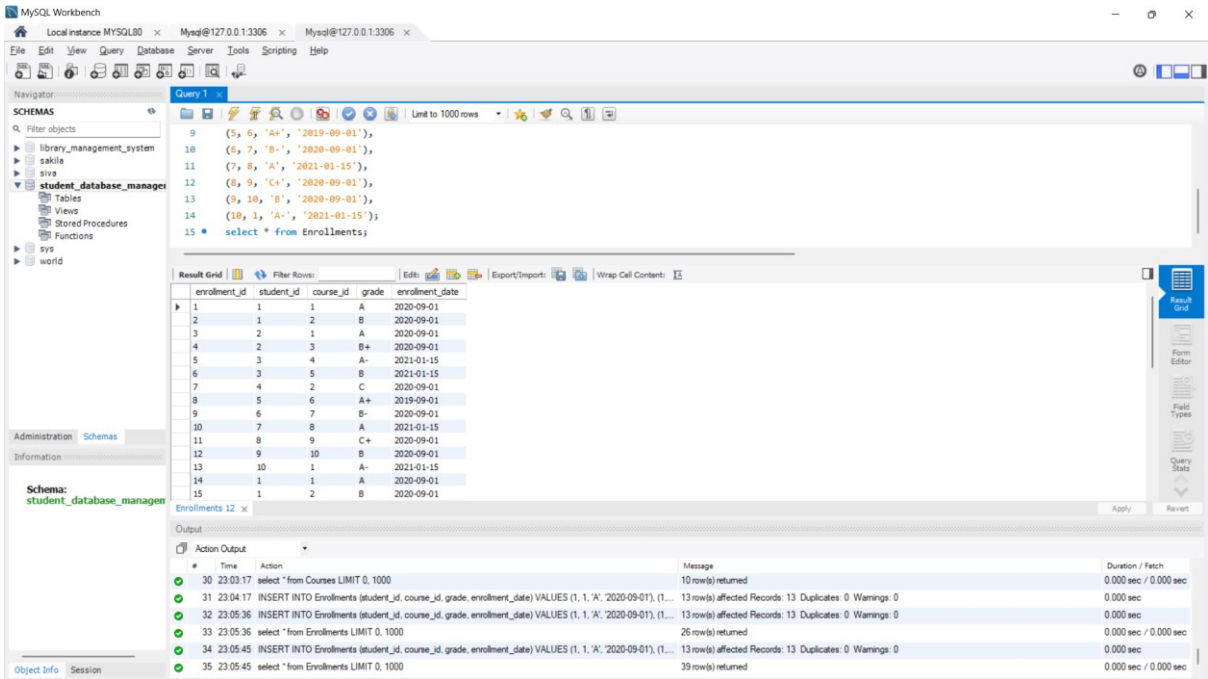


MySQL Workbench interface showing the 'Enrollments' table in the 'student_database_managen' schema. The table structure is as follows:

enrollment_id	student_id	course_id	grade	enrollment_date
1	1	1	A	2020-09-01
2	1	2	B	2020-09-01
3	2	1	A	2020-09-01
4	2	3	B+	2020-09-01
5	3	4	A-	2021-01-15
6	3	5	B-	2021-01-15
7	4	2	C	2020-09-01
8	5	6	A+	2019-09-01
9	6	7	B-	2020-09-01
10	7	8	A	2021-01-15
11	8	9	C+	2020-09-01
12	9	10	B	2020-09-01
13	10	1	A-	2021-01-15
14	10	1	A-	2021-01-15

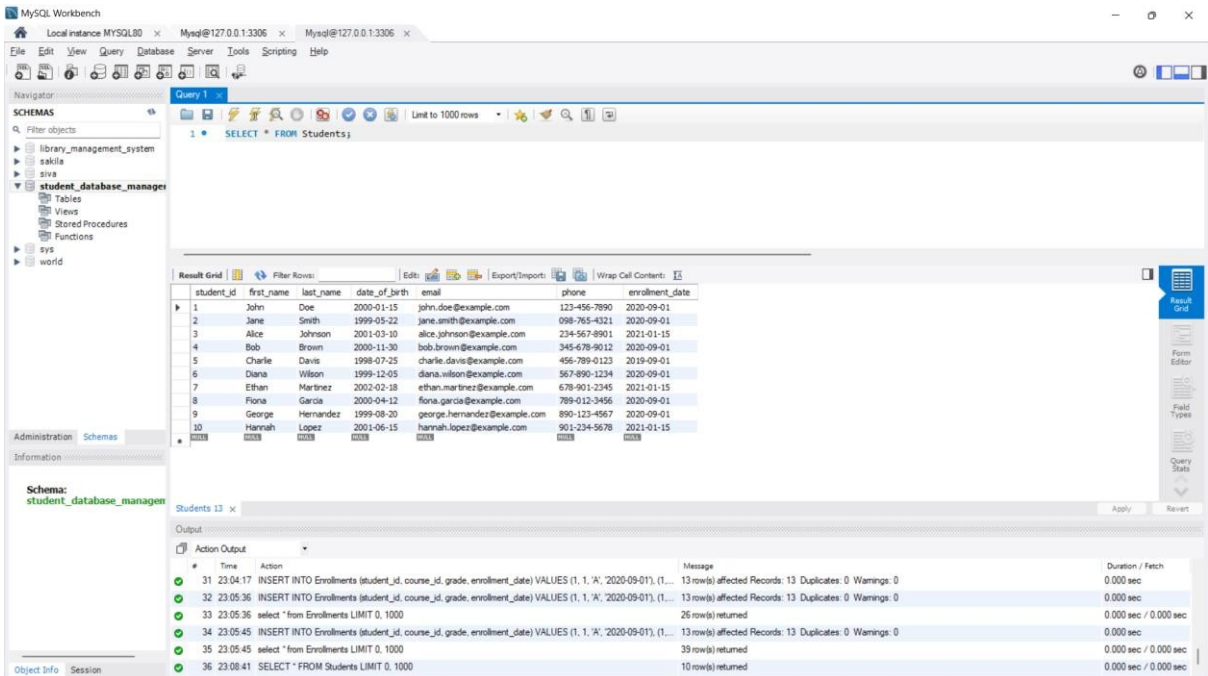
The 'Enrollments' table is selected in the 'student_database_managen' schema. The 'enrollment_id' column is the primary key.

✓ **Enrollments Table has been created .**



SQL Queries

Retrieve All Students :



Retrieve All Courses :

The screenshot shows the MySQL Workbench interface. The 'Query Editor' contains the following SQL query:

```
1 SELECT * FROM Courses;
```

The 'Result Grid' displays the results of the query, showing 10 rows of course data:

course_id	course_name	course_description	credits
1	Database Management	Introduction to database systems.	3
2	Web Development	Learn to build web applications.	4
3	Data Structures	Study of data organization and manipulation.	3
4	Operating Systems	Understanding computer operating systems.	4
5	Software Engineering	Principles of software development.	3
6	Artificial Intelligence	Introduction to AI concepts and applications.	4
7	Machine Learning	Fundamentals of machine learning algorithms.	3
8	Computer Networks	Basics of networking and communication.	4
9	Cybersecurity	Understanding security principles and practices.	3
10	Mobile App Development	Creating applications for mobile devices.	4

The 'Output' pane shows the execution of the query, indicating that 13 rows were affected and 13 duplicates were found.

Retrieve Enrollments with Student and Course Details :

The screenshot shows the MySQL Workbench interface. The 'Query Editor' contains the following SQL query:

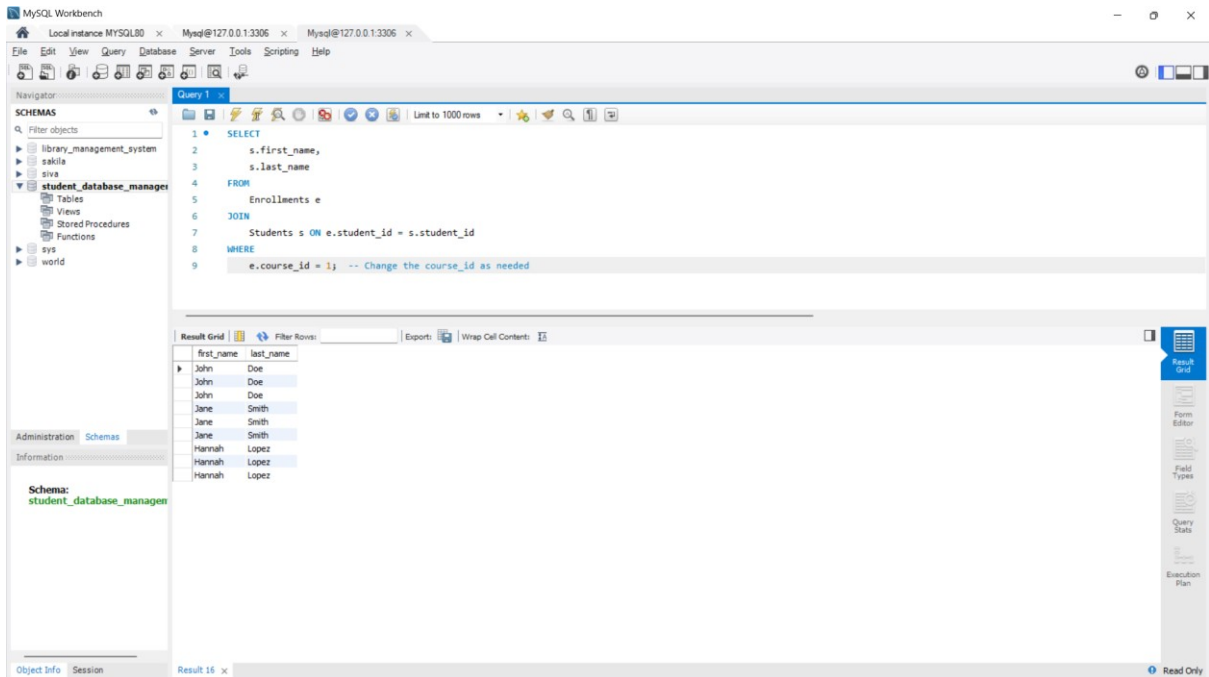
```
1 SELECT  
2 s.first_name,  
3 s.last_name,  
4 c.course_name,  
5 e.grade  
6 FROM  
7 Enrollments e  
8 JOIN  
9 Students s ON e.student_id = s.student_id  
10 JOIN  
11 Courses c ON e.course_id = c.course_id;
```

The 'Result Grid' displays the results of the query, showing 15 rows of enrollment data:

first_name	last_name	course_name	grade
John	Doe	Database Management	A
John	Doe	Web Development	B
John	Doe	Database Management	A
John	Doe	Web Development	B
John	Doe	Database Management	A
John	Doe	Web Development	B
Jane	Smith	Database Management	A
Jane	Smith	Data Structures	B+
Jane	Smith	Database Management	A
Jane	Smith	Data Structures	B+
Jane	Smith	Database Management	A
Jane	Smith	Data Structures	B+
Alice	Johnson	Operating Systems	A-
Alice	Johnson	Software Engineering	B
Alice	Johnson	Operating Systems	A-
Alice	Johnson	Software Engineering	B
Alice	Johnson	Operating Systems	A-
Alice	Johnson	Software Engineering	B
Bob	Brown	Web Development	C
Bob	Brown	Web Development	C
Bob	Brown	Web Development	C
Charlie	Davis	Artificial Intelligence	A+

The 'Output' pane shows the execution of the query, indicating that 13 rows were affected and 13 duplicates were found.

Find Students Enrolled in a Specific Course :



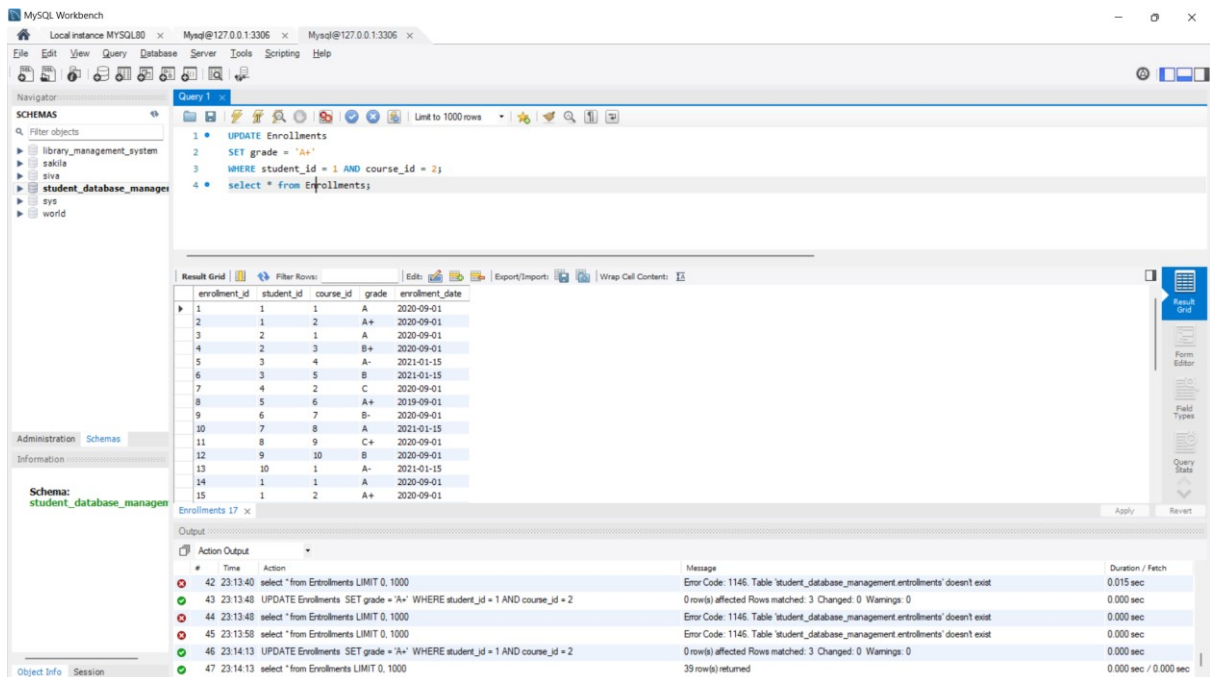
The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' list with 'student_database_manager' selected. The main query editor contains the following SQL code:

```
1 • SELECT
2   s.first_name,
3   s.last_name
4 FROM
5   Enrollments e
6 JOIN
7   Students s ON e.student_id = s.student_id
8 WHERE
9   e.course_id = 1; -- Change the course_id as needed
```

The 'Result Grid' shows the following data:

first_name	last_name
John	Doe
John	Doe
John	Doe
Jane	Smith
Jane	Smith
Hannah	Lopez
Hannah	Lopez

Update a Student's Grade :



The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' list with 'student_database_manager' selected. The main query editor contains the following SQL code:

```
1 • UPDATE Enrollments
2 SET grade = 'A+'
3 WHERE student_id = 1 AND course_id = 2;
4 select * from Enrollments;
```

The 'Result Grid' shows the following data:

enrollment_id	student_id	course_id	grade	enrollment_date
1	1	1	A	2020-09-01
2	1	2	A+	2020-09-01
3	2	1	A	2020-09-01
4	2	3	B+	2020-09-01
5	3	4	A-	2021-01-15
6	3	5	B	2021-01-15
7	4	2	C	2020-09-01
8	5	6	A+	2019-09-01
9	6	7	B-	2020-09-01
10	7	8	A	2021-01-15
11	8	9	C+	2020-09-01
12	9	10	B	2020-09-01
13	10	1	A-	2021-01-15
14	1	1	A	2020-09-01
15	1	2	A+	2020-09-01

The 'Output' pane shows the following messages:

#	Time	Action	Message	Duration / Fetch
42	23:13:40	select * from Enrollments LIMIT 0, 1000	Error Code: 1146. Table 'student_database_management.enrollments' doesn't exist	0.015 sec
43	23:13:48	UPDATE Enrollments SET grade = 'A+' WHERE student_id = 1 AND course_id = 2	0 row(s) affected Rows matched: 3 Changed: 0 Warnings: 0	0.000 sec
44	23:13:48	select * from Enrollments LIMIT 0, 1000	Error Code: 1146. Table 'student_database_management.enrollments' doesn't exist	0.000 sec
45	23:13:58	select * from Enrollments LIMIT 0, 1000	Error Code: 1146. Table 'student_database_management.enrollments' doesn't exist	0.000 sec
46	23:14:13	UPDATE Enrollments SET grade = 'A+' WHERE student_id = 1 AND course_id = 2	0 row(s) affected Rows matched: 3 Changed: 0 Warnings: 0	0.000 sec
47	23:14:13	select * from Enrollments LIMIT 0, 1000	39 row(s) returned	0.000 sec / 0.000 sec

Delete a Student Record :

The screenshot shows the MySQL Workbench interface. The 'Query' tab is active, displaying the following SQL query:

```
1 DELETE FROM Enrollments WHERE student_id = 2;
2 DELETE FROM Students WHERE student_id = 2;
3 select * from Students;
```

The 'Result Grid' shows the output of the query, displaying a table with 10 rows of student data. The columns are: student_id, first_name, last_name, date_of_birth, email, phone, and enrollment_date.

The 'Output' tab shows the execution results, including the error message:

```
50 23:15:18 DELETE FROM Students WHERE student_id = 2 Error Code: 1451: Cannot delete or update a parent row: a foreign key constraint fails ('student_database_man... Duration: / Fetch
51 23:16:10 DELETE FROM Enrollments WHERE student_id = 2 6 row(s) affected 0.000 sec
52 23:16:10 DELETE FROM Students WHERE student_id = 2 1 row(s) affected 0.000 sec
53 23:16:43 DELETE FROM Enrollments WHERE student_id = 2 0 row(s) affected 0.000 sec
54 23:16:43 DELETE FROM Students WHERE student_id = 2 0 row(s) affected 0.000 sec
55 23:16:43 select * from Students LIMIT 0, 1000 9 row(s) returned 0.000 sec / 0.000 sec
```

Conclusion :

Student Database Management System (SDBMS) is essential for educational institutions to efficiently manage student records and streamline administrative processes. It centralizes data, improves accuracy, and enhances communication among students, faculty, and staff.