

**Name** : M Sivananthan

**Company:** CODTECH IT SOLUTIONS

**Intern ID** : CT08DS6401

**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

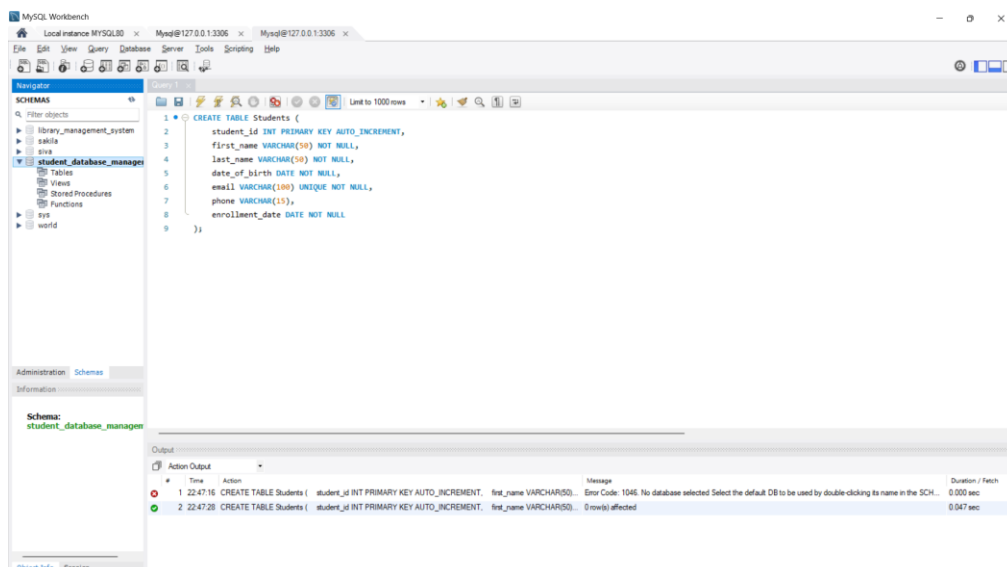
#### 1. Tables Overview :

We'll create three main tables:

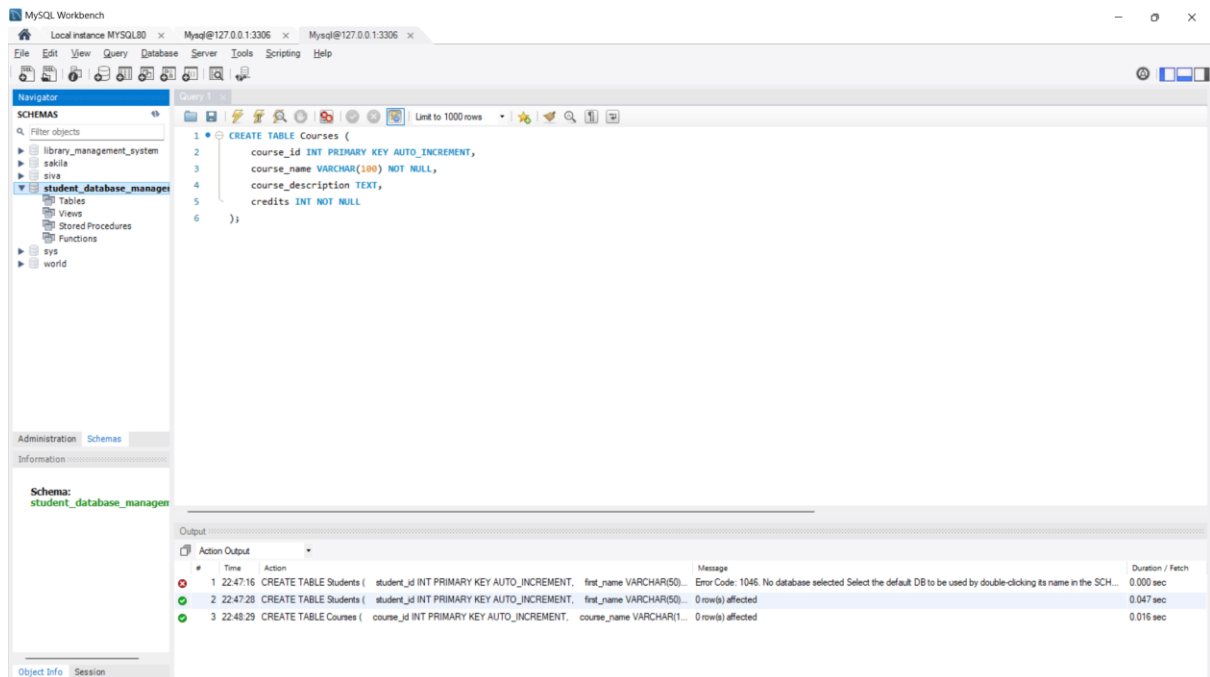
- **Students:** To store personal details of the students.
- **Courses:** To store information about the courses offered.
- **Enrollments:** To manage the relationship between students and courses, including grades.

#### 2. Table Definitions

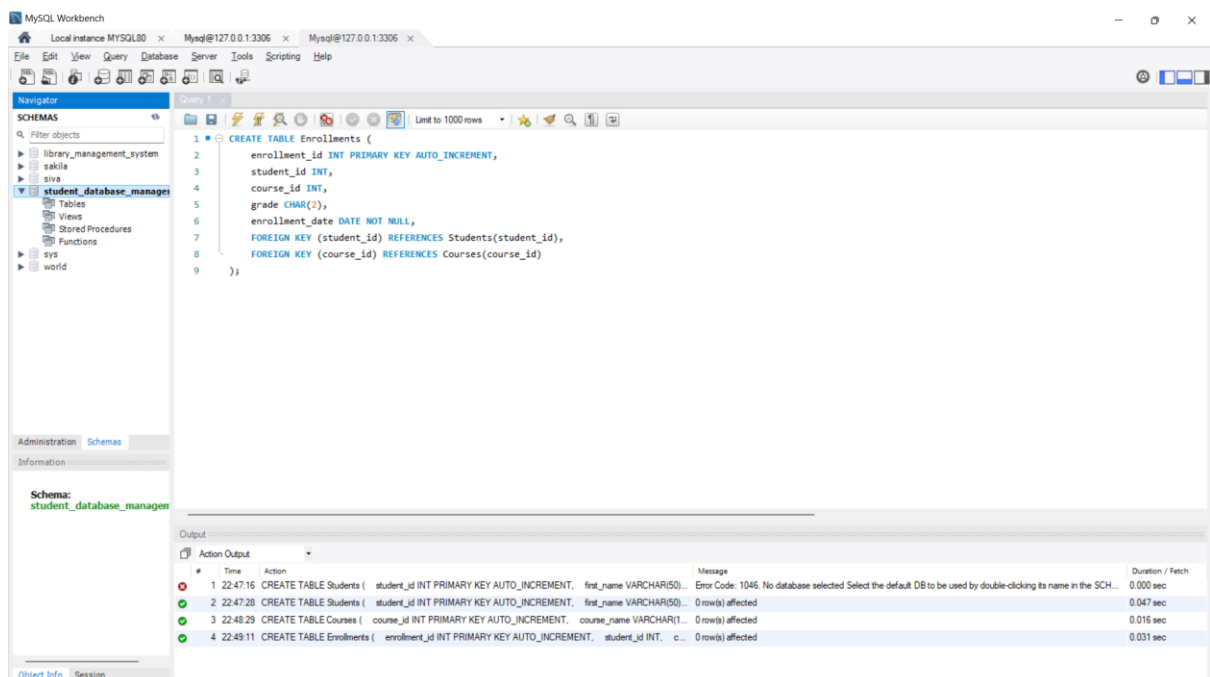
##### Students Table :



## Courses Table :



## Enrollments Table :



## To view the Table :

**select \* from Students;**

**select \* from Courses;**

**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\_manager
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys
- world

Administration Schemas

Information

Schema: student\_database\_manager

Query 1: x

```

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

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

## 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\_manager
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys
- world

Administration Schemas

Information

Schema: student\_database\_manager

Query 1: x

```

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

#	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(255), 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 INT, 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

## ✓ Students Table has been created.

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left lists the 'student\_database\_manager' 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), 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), (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
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), 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), (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
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:

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' panel on the left displays the 'student\_database\_manager' schema. The 'Query' panel shows a query: `select * from Courses;`. The 'Result Grid' displays the data for the 'Courses' table, which has 10 rows. The 'Output' panel shows the execution log, including the creation of the 'Courses' table and the execution of the query.

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

## Inserting Sample Enrollments :

The screenshot shows the MySQL Workbench interface. The 'Query' panel shows a query: `INSERT INTO Enrollments (student_id, course_id, grade, enrollment_date) VALUES (1, 1, 'A', '2020-09-01'), (1, 2, 'B', '2020-09-01'), (2, 1, 'A', '2020-09-01'), (2, 3, 'B+', '2020-09-01'), (3, 4, 'A-', '2021-01-15'), (3, 5, 'B', '2021-01-15'), (4, 2, 'C', '2020-09-01'), (5, 6, 'A+', '2019-09-01'), (6, 7, 'B-', '2020-09-01'), (7, 8, 'A', '2021-01-15'), (8, 9, 'C+', '2020-09-01'), (9, 10, 'B', '2020-09-01'), (10, 1, 'A-', '2021-01-15');`. The 'Output' panel shows the execution log, including the creation of the 'Enrollments' table and the execution of the query.

#	Time	Action	Message	Duration / Fetch
26	22:59:08	CREATE TABLE Enrollments ( enrollment_id INT PRIMARY KEY AUTO_INCREMENT, student_id INT, ...	0 row(s) affected	0.031 sec
27	22:59:44	INSERT INTO Enrollments ( enrollment_id INT PRIMARY KEY AUTO_INCREMENT, student_id INT, ...	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', 'Intro...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
30	23:03:17	select * from Courses LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
31	23:04:17	INSERT INTO Enrollments (student_id, course_id, grade, enrollment_date) VALUES (1, 1, 'A', '2020-09-01'), (1...	13 row(s) affected Records: 13 Duplicates: 0 Warnings: 0	0.000 sec

## ✓ Enrollments Table has been created .

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with the 'student\_database\_manager' schema selected. The main query window shows a query that inserts data into the 'Enrollments' table. The 'Result Grid' displays the following data:

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	1	1	A	2020-09-01
15	1	2	B	2020-09-01

The 'Output' panel shows the execution of the query, indicating that 15 rows were returned and the table was successfully created.

## SQL Queries

### Retrieve All Students :

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with the 'student\_database\_manager' schema selected. The main query window shows a query that selects all data from the 'Students' table. The 'Result Grid' displays the following data:

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

The 'Output' panel shows the execution of the query, indicating that 10 rows were returned.

## Retrieve All Courses :

The screenshot displays the MySQL Workbench environment. At the top, the title bar shows 'MySQL Workbench' and several open tabs for 'Local instance MYSQL80' and 'mysql@127.0.0.1:3306'. The main interface is divided into several panes:

- Navigator (Left):** Shows a tree view of the database schema. The 'student\_database\_manager' schema is selected, showing tables, views, stored procedures, and functions.
- Query Editor (Top Center):** Contains a single query: `SELECT * FROM Courses;`
- Result Grid (Center):** Displays the results of the query in a table format. The table has four columns: 'course\_id', 'course\_name', 'course\_description', and 'credits'. It lists 10 courses, including 'Database Management', 'Web Development', 'Data Structures', 'Operating Systems', 'Software Engineering', 'Artificial Intelligence', 'Machine Learning', 'Computer Networks', 'Cybersecurity', and 'Mobile App Development'.
- Administration (Bottom Left):** Includes tabs for 'Schemas' and 'Information'.
- Schema: student\_database\_manager (Bottom Left):** A detailed view of the selected schema.
- Action Output (Bottom Center):** Shows the execution log of the query, including the time taken and the number of rows affected.
- Message (Bottom Right):** Displays a message indicating that 13 rows were affected, with 0 duplicates and 0 warnings.

The bottom of the screen shows the Windows taskbar with various icons and the system clock indicating 23:10 on 09-08-2024.

### Retrieve Enrollments with Student and Course Details :

The screenshot displays the MySQL Workbench environment. At the top, the title bar shows 'MySQL Workbench' and several open tabs: 'Local instance MySQL80', 'mysql@127.0.0.1:3306', and 'mysql@127.0.0.1:3306'. The main menu bar includes 'File', 'Edit', 'View', 'Query', 'Database', 'Server', 'Tools', 'Scripting', and 'Help'. Below the menu is a toolbar with various icons for file operations and database management.

The 'Navigator' pane on the left shows the 'SCHEMAS' section with a search filter 'Filter objects'. The tree view includes 'library\_management\_system', 'skilla', 'sys', 'student\_database\_manager' (selected), 'Tables', 'Views', 'Stored Procedures', 'Functions', 'sys', and 'world'.

The 'Query' editor shows a SQL query:
 

```

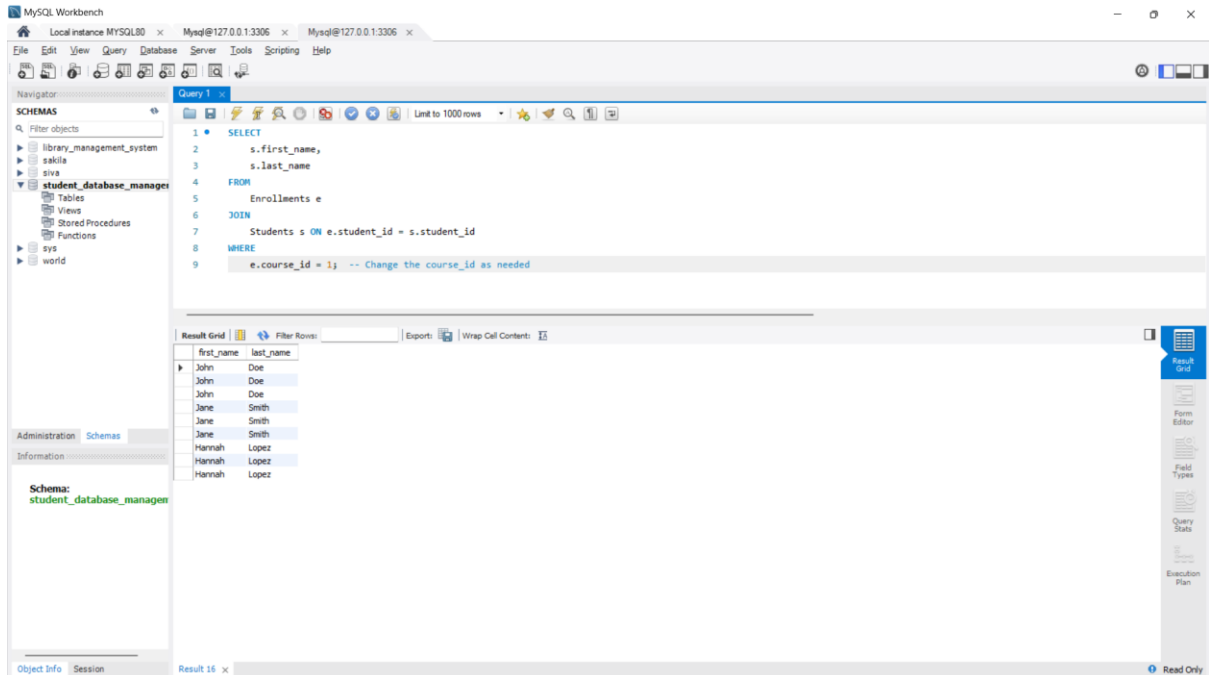
1  SELECT
2      s.first_name,
3      s.last_name,
4      c.course_name,
5      e.grade
6  FROM
7      Enrollments e
8  JOIN
9      Students s ON e.student_id = s.student_id
10 JOIN
11      Courses c ON e.course_id = c.course_id;
    
```

The 'Result Grid' pane at the bottom displays the query results in a table with columns: 'first\_name', 'last\_name', 'course\_name', and 'grade'. The results show a list of students and their enrolled courses with corresponding grades. The first few rows are:
 

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+
Charlie	Davis	Artificial Intelligence	A+

The 'Administration' pane on the left shows 'Schemas' and 'Information' sections. The 'Schema: student\_database\_manager' is selected. The 'Object Info' pane at the bottom shows 'Result: 15 x'.

## Find Students Enrolled in a Specific Course :



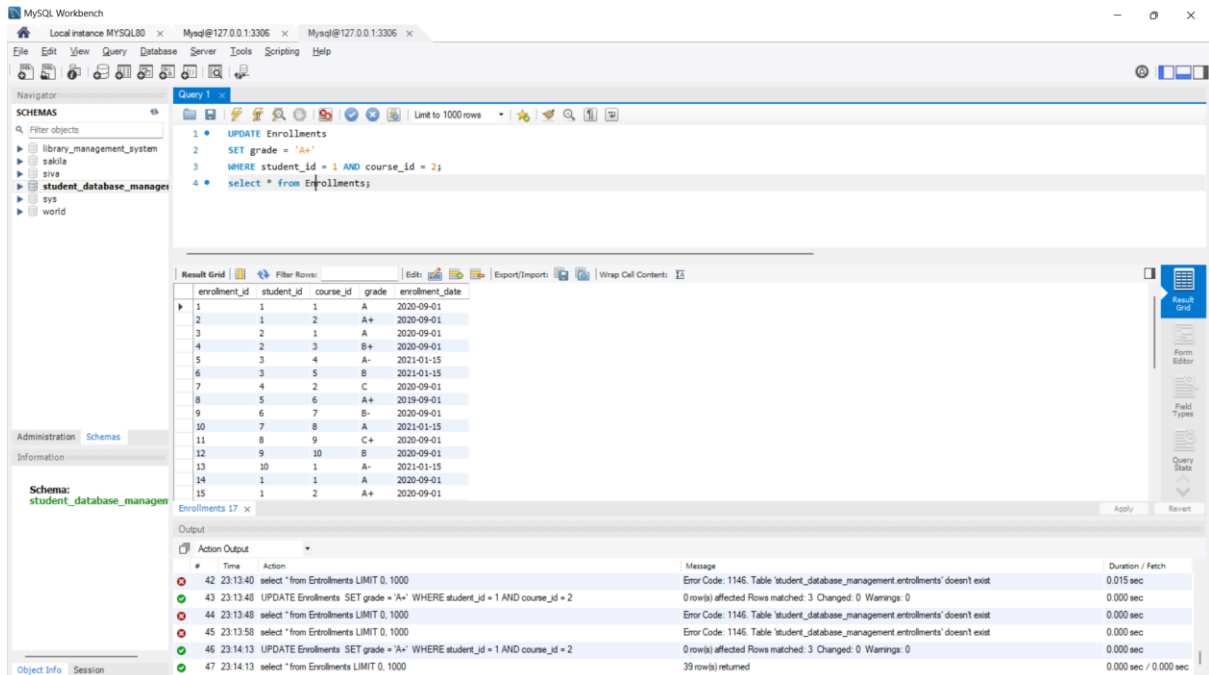
The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' list, with 'student\_database\_managen' selected. The main query editor contains the following SQL code:

```
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' at the bottom shows the results of the query:

first_name	last_name
John	Doe
John	Doe
John	Doe
Jane	Smith
Jane	Smith
Jane	Smith
Hannah	Lopez
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\_managen' 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' at the bottom shows the results of the query:

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 at the bottom shows the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
42	23:13:40	select * from Enrollments LIMIT 0, 1000	Error Code: 1146. Table 'student_database_managen.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_managen.enrollments' doesn't exist	0.000 sec
45	23:13:50	select * from Enrollments LIMIT 0, 1000	Error Code: 1146. Table 'student_database_managen.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 displays the MySQL Workbench interface. The 'Query' tab is active, showing a SQL query that deletes a student record and its associated enrollments. The 'Result Grid' shows the current state of the 'Students' table. The 'Output' tab at the bottom shows the execution log, indicating that the deletion was successful for the specified student ID.

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

student_id	first_name	last_name	date_of_birth	email	phone	enrollment_date
1	John	Doe	2000-01-15	john.doe@example.com	123-456-7890	2020-09-01
3	Alice	Johnson	2001-03-10	alice.johnson@example.com	234-567-8901	2021-01-15
4	Bob	Brown	2000-11-30	bob.brown@example.com	345-678-9012	2020-09-01
5	Charlie	Davis	1998-07-25	charlie.davis@example.com	456-789-0123	2019-09-01
6	Diana	Wilson	1999-12-05	diana.wilson@example.com	diana.wilson@example.com	2020-09-01
7	Ethan	Martinez	2002-02-18	ethan.martinez@example.com	567-890-1234	2021-01-15
8	Fiona	Garcia	2003-04-12	fiona.garcia@example.com	789-012-3456	2020-09-01
9	George	Hernandez	1999-08-20	george.hernandez@example.com	890-123-4567	2020-09-01
10	Hannah	Lopez	2001-06-15	hannah.lopez@example.com	901-234-5678	2021-01-15

Time	Action	Message	Duration / Fetch
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...	0.000 sec
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.**