

# Student Information System

## 1. Create necessary database and tables

### Database:

```
CREATE DATABASE Studentinfromationsystem;
```

### Students Table:

```
CREATE TABLE students (  
    student_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255),  
    age INT,  
    gender VARCHAR(10),  
    email VARCHAR(255),  
    phone VARCHAR(20)  
);
```

### Courses Table:

```
CREATE TABLE courses(  
    courses_id INT AUTO_INCREMENT PRIMARY key,  
    Course_name VARCHAR(255),  
    instructor VARCHAR(255),  
    credits INT  
);
```

## Grades Table:

```
CREATE TABLE Grades(  
    grade_id INT AUTO_INCREMENT Primary key,  
    students_id INT,  
    courses_id INT,  
    grade VARCHAR(2),  
    FOREIGN KEY(students_id) REFERENCES students(students_id),  
    FOREIGN KEY(courses_id) REFERENCES courses(courses_id)  
);
```

## Attendance Table:

```
CREATE TABLE attendance (  
    attendance_id INT AUTO_INCREMENT PRIMARY KEY,  
    student_id INT,  
    courses_id INT,  
    date DATE,  
    status VARCHAR(10),  
    FOREIGN KEY (student_id) REFERENCES students(student_id),  
    FOREIGN KEY (courses_id) REFERENCES courses(courses_id)  
);
```

## 2. Add sample data into the database

### Sample data for Students table:

```
INSERT INTO students( name, age, gender, email, phone)  
VALUES  
    ('Ram', 20, 'male', 'ram@gmail.com', '9876543212'),  
    ('Jenny', 22, 'Female', 'jenny@gmail.com', '9876543210'),  
    ('Alex', 22, 'male', 'Alex@gmail.com', '9876543456');
```

Server: 127.0.0.1 » Database: studentinformationsystem » Table: students

UPDATE students SET Phone\_number = '9876543456' WHERE students.student\_id = 3;

[ Edit inline ] [ Edit ] [ Create PHP code ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

		student_id	student_name	age	gender	email	Phone_number		
<input type="checkbox"/>	Edit	Copy	Delete	1	Ram	20	male	ram@gmail.com	9876543212
<input type="checkbox"/>	Edit	Copy	Delete	2	Jenny	22	Female	jenny@gmail.com	9876543210
<input type="checkbox"/>	Edit	Copy	Delete	3	Alex	22	male	Alex@gmail.com	9876543456

### Sample data for Courses Table:

```
INSERT INTO courses(Course_name, instructor, credits)
VALUES ('Mathematics', 'Prof.Smith', 4),
       ('English', 'Dr.Johnson', 3),
       ('Computer Science', 'Prof.Hari', 5);
```

Server: 127.0.0.1 » Database: studentinformationsystem » Table: courses

Showing rows 0 - 2 (3 total, Query took 0.0011 seconds.)

SELECT \* FROM `courses`

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

			courses_id	Course_name	instructor	credits	
<input type="checkbox"/>	Edit	Copy	Delete	1	Mathematics	Prof.Smith	4
<input type="checkbox"/>	Edit	Copy	Delete	2	English	Dr.Johnson	3
<input type="checkbox"/>	Edit	Copy	Delete	3	Computer Science	Prof.Hari	5

☐ Check all | With selected: Edit Copy Delete Export

## Sample data for Grades Table:

```
INSERT INTO grades (student_id, course_id, grade)
VALUES
    (1, 1, 'A'),
    (1, 2, 'B'),
    (2, 1, 'C');
```

Server: 127.0.0.1 » Database: studentinformationsystem » Table: grades

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

✓ Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

`SELECT * FROM `grades``

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

			grade_id	student_id	courses_id	grade
<input type="checkbox"/>	Edit	Copy	Delete	1	1	A
<input type="checkbox"/>	Edit	Copy	Delete	2	1	B
<input type="checkbox"/>	Edit	Copy	Delete	3	2	C

↑ ☐ Check all With selected: Edit Copy Delete Export

## Sample data for Attendance Table:

```
INSERT INTO attendance (student_id, course_id, date, status)
VALUES (1, 1, '2024-04-01', 'Present'),
    (1, 2, '2024-04-01', 'Present'),
    (2, 1, '2024-04-01', 'Absent');
```

Server: 127.0.0.1 » Database: studentinformationsystem » Table: attendance

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 2 (3 total, Query took 0.0008 seconds.)

```
SELECT * FROM `attendance`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	attendance_id	student_id	courses_id	date	status
<input type="checkbox"/> Edit Copy Delete	1	1	1	2024-04-01	Present
<input type="checkbox"/> Edit Copy Delete	2	1	2	2024-04-01	Present
<input type="checkbox"/> Edit Copy Delete	3	2	1	2024-04-01	Absent

Check all | With selected: Edit Copy Delete Export

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

### 3. Retrieve student information

```
-- Retrieve student information including their ID, student_name, age,gender and phone_number
SELECT student_id, student_name, age, gender , phone_number
FROM students;
```

Server: 127.0.0.1 » Database: studentinformationsystem » Table: students

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Show query box

Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
-- Retrieve student information including their ID, student_name, age,gender and phone_number
SELECT student_id, student_name, age, gender ,
phone_number FROM students;
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

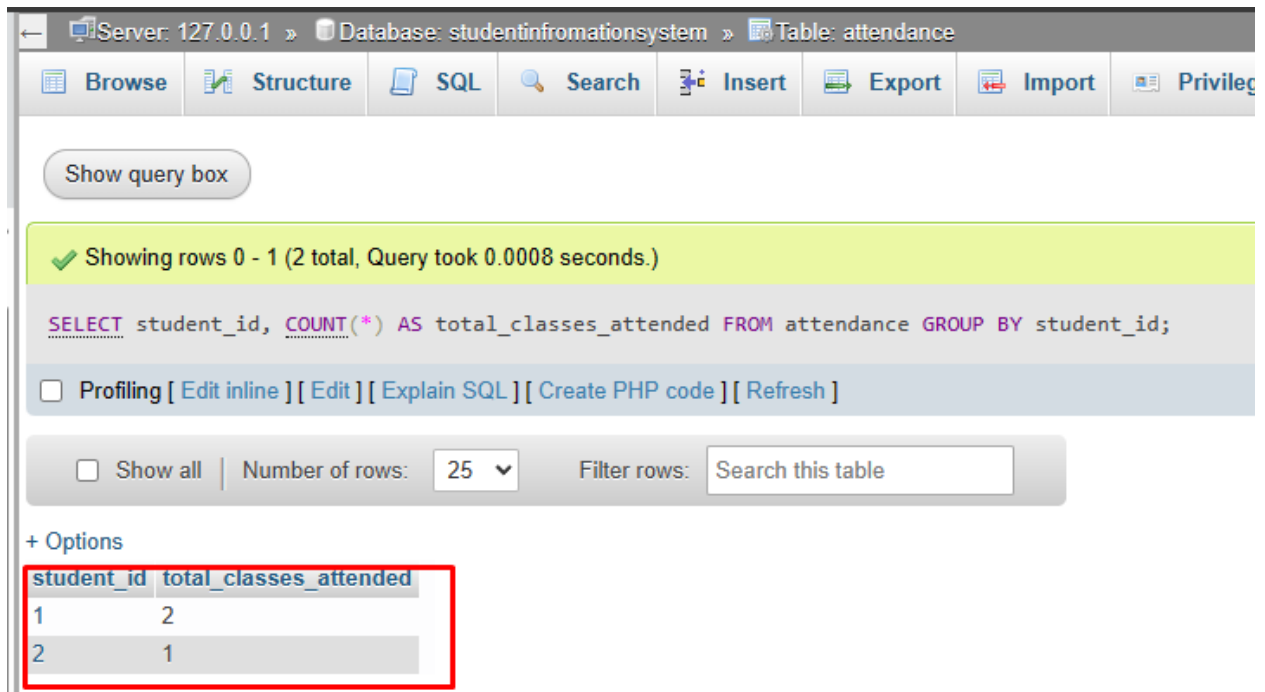
+ Options

	student_id	student_name	age	gender	phone_number
<input type="checkbox"/> Edit Copy Delete	1	Ram	20	male	9543211234
<input type="checkbox"/> Edit Copy Delete	2	Jenny	22	Female	9821367890
<input type="checkbox"/> Edit Copy Delete	3	Alex	22	male	9832145678

Check all | With selected: Edit Copy Delete Export

#### 4. Calculate the total number of classes attended by each student:

```
SELECT student_id, COUNT(*) AS total_classes_attended
FROM attendance
GROUP BY student_id;
```



The screenshot shows a database management tool interface. At the top, the breadcrumb navigation indicates the path: Server: 127.0.0.1 » Database: studentinformationsystem » Table: attendance. Below this is a toolbar with buttons for Browse, Structure, SQL, Search, Insert, Export, Import, and Privileges. A 'Show query box' button is located below the toolbar. The main area displays a green status bar indicating 'Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)'. Below this, the SQL query is shown: `SELECT student_id, COUNT(*) AS total_classes_attended FROM attendance GROUP BY student_id;`. There are links for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. Below the query, there are controls for 'Show all', 'Number of rows' (set to 25), and a 'Filter rows' search box. The results are shown in a table with two columns: student\_id and total\_classes\_attended. The table contains two rows: (1, 2) and (2, 1). The table is highlighted with a red border.

student_id	total_classes_attended
1	2
2	1

#### 5. Retrieve student information along with their grades:

```
-- Retrieve student ID, student name, and grade by joining the students and grades tables
SELECT s.student_id, s.student_name, g.grade
FROM students s
INNER JOIN grades g ON s.student_id = g.student_id;
```

Server: 127.0.0.1 » Database: studentinformationsystem

Structure SQL Search Query Export Import Operations Privileges Routines Events More

Show query box

Showing rows 0 - 2 (3 total, Query took 0.0009 seconds.)

-- Retrieve student ID, student name, and grade by joining the students and grades tables `SELECT s.student_id, s.student_name, g.grade FROM students s INNER JOIN grades g ON s.student_id = g.student_id;`

[ Edit inline ] [ Edit ] [ Create PHP code ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

student_id	student_name	grade
1	Ram	A
1	Ram	B
2	Jenny	C

6. Count the number of attendance records (e.g., present, absent) for a specific course:

```
SELECT status, COUNT(*) AS count
FROM attendance
WHERE courses_id IN (1, 2)
GROUP BY status;
```

Server: 127.0.0.1 » Database: studentinformationsystem » Table: attendance

Browse Structure SQL Search Insert Export Import Privileges

Show query box

Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)

`SELECT status, COUNT(*) AS count FROM attendance WHERE courses_id IN (1, 2) GROUP BY status;`

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

status	count
Absent	1
Present	2

## 7. Update the instructor for a specific course:

```
UPDATE courses SET instructor = 'Prof.devin' WHERE courses_id = 1;
```

Server: 127.0.0.1 » Database: studentinformationsystem » Table: courses

Browse Structure SQL Search Insert Export Import Privileges Operations

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT * FROM `courses`
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

				courses_id	Course name	instructor	credits
<input type="checkbox"/>	Edit	Copy	Delete	1	Mathematics	Prof.devin	4
<input type="checkbox"/>	Edit	Copy	Delete	2	English	Dr.Johnson	3
<input type="checkbox"/>	Edit	Copy	Delete	3	Computer Science	Prof.Hari	5

## 8. Find students with perfect attendance (no absences):

```
SELECT student_id -- Select the student_id column
FROM attendance -- From the attendance table
GROUP BY student_id -- Group the records by student_id
HAVING COUNT(CASE WHEN status = 'Absent' THEN 1 END) = 0; -- Filter groups to include only those with zero absences
```

Server: 127.0.0.1 » Database: studentinformationsystem » Table: attendance

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Show query box

Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT student_id -- Select the student_id column FROM attendance -- From the attendance table GROUP BY student_id -- Group the records by student_id HAVING COUNT(CASE WHEN status = 'Absent' THEN 1 END) = 0;
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

				student_id
<input type="checkbox"/>	Edit	Copy	Delete	1

## 9. Find Students who Have Taken All Courses:



```

SELECT
    student_id -- Select student_id
FROM (
    SELECT
        student_id, -- Select student_id
        COUNT(DISTINCT courses_id) AS num_courses_taken -- Count the number of distinct courses taken by each student
    FROM grades -- From the grades table
    GROUP BY student_id -- Group the records by student_id
) AS subquery -- Create a subquery
WHERE num_courses_taken = (SELECT COUNT(*) FROM courses); -- Filter to include only students who have taken all courses available in the courses table

```

Server: 127.0.0.1 » Database: studentinformationsystem

Structure SQL Search Query Export Import Operations Privileges Routines Events More

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```

SELECT student_id -- Select student_id FROM ( SELECT student_id, -- Select student_id COUNT(DISTINCT courses_id) AS num_courses_taken -- Count the number of distinct courses taken by each student FROM grades -- From the grades table GROUP BY student_id -- Group the records by student_id ) AS subquery -- Create a subquery WHERE num_courses_taken = (SELECT COUNT(*) FROM courses);

```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

student\_id

## 10. Retrieve Courses with No Attendees:

```

SELECT
    c.* -- Select all columns from the courses table
FROM courses c -- From the courses table
LEFT JOIN attendance a ON c.courses_id = a.courses_id -- Left join with the attendance table on courses_id
WHERE a.attendance_id IS NULL; -- Filter to include only records where there are no corresponding attendance records

```

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```

SELECT c.* -- Select all columns from the courses table FROM courses c -- From the courses table LEFT JOIN attendance a ON c.courses_id = a.courses_id -- Left join with the attendance table on courses_id WHERE a.attendance_id IS NULL;

```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

courses_id	Course_name	instructor	credits
3	Computer Science	Prof.Hari	5

## 11. Delete a student from the table:

```

DELETE FROM students WHERE student_id = 3; -- Delete the student record with student_id equal to 3

```

Show query box

✓ 1 row affected. (Query took 0.0036 seconds.)

```
DELETE FROM students WHERE student_id = 3;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)

```
-- Delete the student record with student_id equal to 3;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

## 12. Count the Number of Absences for Each Student:

```
1 SELECT
2     student_id, -- Select the student_id column
3     COUNT(*) AS num_absences -- Count the number of absences for each student and alias it as num_absences
4 FROM attendance -- From the attendance table
5 WHERE status = 'Absent' -- Filter to include only records where the status is 'Absent'
6 GROUP BY student_id; -- Group the results by student_id
7
```

Your SQL query has been executed successfully.

```
SELECT student_id, -- Select the student_id column COUNT(*) AS num_absences -- Count the number of absences for each student and alias it as
num_absences FROM attendance -- From the attendance table WHERE status = 'Absent' -- Filter to include only records where the status is
'Absent' GROUP BY student_id;
```

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

+ Options

student_id	num_absences
2	1



