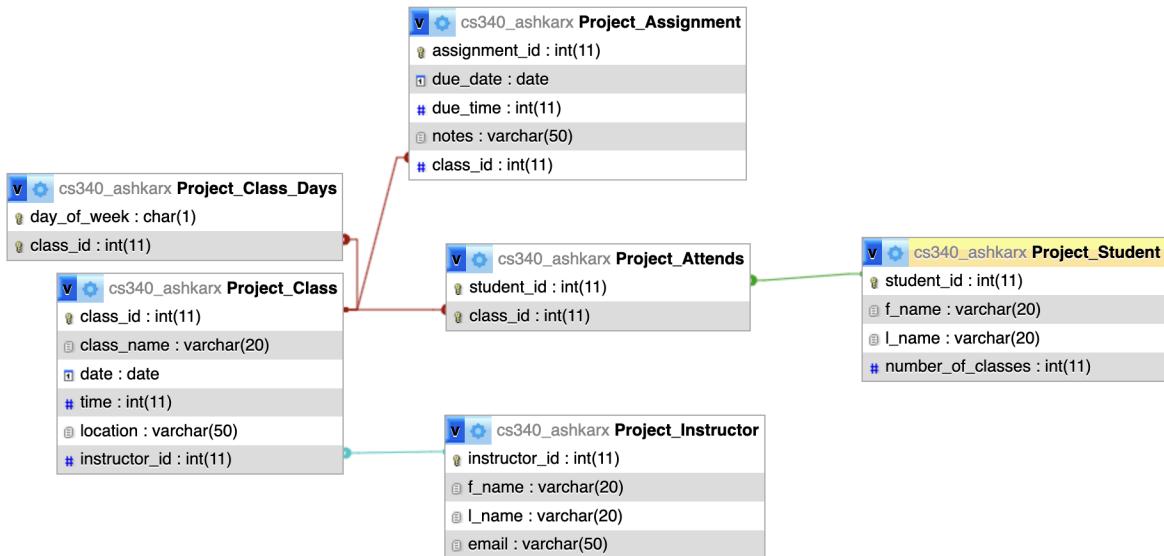


## Schema in phpMyAdmin:



## Delete and Update Constraint Properties for Foreign Keys:

**Project\_Class\_Days**

Action	Column	Foreign key constraint (INNODB)	Database	Table	Column
Drop	Project_Class_Days_ibfk_1		cs340_ashkarx	Project_Class	class_id
ON DELETE	CASCADE	ON UPDATE	CASCADE		

**Project\_Class**

Action	Column	Foreign key constraint (INNODB)	Database	Table	Column
Drop	Project_Class_ibk_1		cs340_ashkarx	Project_Attends	class_id
ON DELETE	RESTRICT	ON UPDATE	CASCADE		

**Project\_Attends**

Action	Column	Foreign key constraint (INNODB)	Database	Table	Column
Drop	Project_Attends_ibk_1		cs340_ashkarx	Project_Student	student_id
ON DELETE	CASCADE	ON UPDATE	CASCADE		
Drop	Project_Attends_ibk_2		cs340_ashkarx	Project_Instructor	instructor_id
ON DELETE	CASCADE	ON UPDATE	CASCADE		

**Project\_Assignment**

Action	Column	Foreign key constraint (INNODB)	Database	Table	Column
Drop	Project_Assignment_ibk_1		cs340_ashkarx	Project_Class	class_id
ON DELETE	CASCADE	ON UPDATE	CASCADE		

## Populated Tables:

Project_Class						
	class_id	class_name	date	time	location	instructor_id
<input type="checkbox"/>	30001	Intro to CS	2025-09-01	900	Kidder Hall Room 101	2001
<input type="checkbox"/>	30002	Data Structures	2025-09-02	1100	Kelley Engineering Room 102	2001
<input type="checkbox"/>	30003	Algorithms	2025-09-01	1300	Kelley Engineering Room 103	2001
<input type="checkbox"/>	30004	Databases	2025-09-03	1500	Milne Computer Lab Room 104	2002
<input type="checkbox"/>	30005	Networks	2025-09-01	1000	Milne Computer Lab Room 105	2002
<input type="checkbox"/>	30006	Operating Systems	2025-09-02	1200	Bearborn Hall Room 106	2003
<input type="checkbox"/>	30007	Web Development	2025-09-03	1400	Benton Hall Room 107	2004
<input type="checkbox"/>	30008	Computer Graphics	2025-09-01	800	Benton Hall Room 108	2004
<input type="checkbox"/>	30009	AI Fundamentals	2025-09-04	1600	Kearney Hall Room 109	2005
<input type="checkbox"/>	30010	Ethics in Tech	2025-09-05	1000	Kearney Hall Room 110	2006

Project_Instructor						
	instructor_id	f_name	l_name	email		
<input type="checkbox"/>	2001	John	Adams	john.adams@school.edu		
<input type="checkbox"/>	2002	Mary	White	mary.white@school.edu		
<input type="checkbox"/>	2003	Steven	Green	steven.green@school.edu		
<input type="checkbox"/>	2004	Laura	Black	laura.black@school.edu		
<input type="checkbox"/>	2005	Kevin	Moor	kevin.moor@school.edu		
<input type="checkbox"/>	2006	Rachel	Hill	rachel.hill@school.edu		
<input type="checkbox"/>	2007	Daniel	Young	daniel.young@school.edu		
<input type="checkbox"/>	2008	Anna	Scott	anna.scott@school.edu		
<input type="checkbox"/>	2009	Brian	King	brian.king@school.edu		
<input type="checkbox"/>	2010	Olivia	Baker	olivia.baker@school.edu		

Project_Class_Days						
	day_of_week	class_id				
<input type="checkbox"/>	F	30001				
<input type="checkbox"/>	F	30007				
<input type="checkbox"/>	F	30010				
<input type="checkbox"/>	M	30001				
<input type="checkbox"/>	M	30003				
<input type="checkbox"/>	M	30005				
<input type="checkbox"/>	M	30007				
<input type="checkbox"/>	M	30009				
<input type="checkbox"/>	R	30002				
<input type="checkbox"/>	R	30004				
<input type="checkbox"/>	R	30006				
<input type="checkbox"/>	R	30008				
<input type="checkbox"/>	T	30002				
<input type="checkbox"/>	T	30004				
<input type="checkbox"/>	T	30006				

Project_Attends						
	student_id	class_id				
<input type="checkbox"/>	100000001	30001				
<input type="checkbox"/>	100000001	30002				
<input type="checkbox"/>	100000001	30003				
<input type="checkbox"/>	100000002	30001				
<input type="checkbox"/>	100000002	30005				
<input type="checkbox"/>	100000003	30003				
<input type="checkbox"/>	100000003	30004				
<input type="checkbox"/>	100000003	30009				
<input type="checkbox"/>	100000004	30002				
<input type="checkbox"/>	100000004	30006				
<input type="checkbox"/>	100000005	30001				
<input type="checkbox"/>	100000005	30003				
<input type="checkbox"/>	100000005	30007				
<input type="checkbox"/>	100000006	30004				
<input type="checkbox"/>	100000006	30008				

Project_Assignment						
	assignment_id	due_date	due_time	notes	class_id	
<input type="checkbox"/>	4001	2025-09-10	2359	Homework 1: Basics	30001	
<input type="checkbox"/>	4002	2025-09-17	2359	Homework 2: Loops	30001	
<input type="checkbox"/>	4003	2025-09-24	2359	Homework 3: Functions	30001	
<input type="checkbox"/>	4004	2025-09-11	1700	Homework 1: Arrays	30002	
<input type="checkbox"/>	4005	2025-09-18	1700	Homework 2: Pointers	30002	
<input type="checkbox"/>	4006	2025-09-12	1200	Homework 1: Sorting	30003	
<input type="checkbox"/>	4007	2025-09-19	1200	Homework 2: Searching	30003	
<input type="checkbox"/>	4008	2025-09-26	1200	Homework 3: Recursion	30003	
<input type="checkbox"/>	4009	2025-09-13	1500	Homework 1: Schema Design	30004	
<input type="checkbox"/>	4010	2025-09-14	1600	Homework 1: IP Addressing	30005	
<input type="checkbox"/>	4011	2025-09-21	1600	Homework 2: Subnetting	30005	
<input type="checkbox"/>	4012	2025-09-15	1800	Homework 1: File Systems	30006	
<input type="checkbox"/>	4013	2025-09-16	1400	Homework 1: HTML	30007	

Project_Student						
	student_id	f_name	l_name	number_of_classes	email	
<input type="checkbox"/>	100000001	Bob	Smith	0	bobsmith@gmail.com	
<input type="checkbox"/>	100000002	Alice	Johnson	0	alicejohnson@gmail.com	
<input type="checkbox"/>	100000003	David	Lee	0	davidlee@gmail.com	
<input type="checkbox"/>	100000004	Emily	Cook	0	emilycook@gmail.com	
<input type="checkbox"/>	100000005	Michael	Brown	0	michaelbrown@gmail.com	
<input type="checkbox"/>	100000006	Sarah	Davis	0	sarahdavis@gmail.com	
<input type="checkbox"/>	100000007	James	Taylor	0	jamestaylor@gmail.com	
<input type="checkbox"/>	100000008	Linda	Harris	0	lindaharris@gmail.com	
<input type="checkbox"/>	100000009	Robert	Martinez	0	robertmartinez@gmail.com	
<input type="checkbox"/>	100000010	Jessica	Anderson	0	jessicanderson@gmail.com	

## 1. Select Queries

### a. Select Using JOIN

List the first name and last name of students who attend the class Databases.

```
SELECT s.f_name, s.l_name  
FROM Project_Student AS s  
JOIN Project_Attends AS a ON s.student_id = a.student_id  
JOIN Project_Class AS c ON a.class_id = c.class_id  
WHERE c.class_name = 'Databases';
```

The screenshot shows a MySQL query results interface. At the top, a green bar indicates "Showing rows 0 - 2 (3 total, Query took 0.0000 seconds.)". Below it is the SQL query. A toolbar below the query includes options like "Profiling", "Edit inline", "Explain SQL", "Create PHP code", and "Refresh". Further down are filters for "Show all" (checkbox), "Number of rows" (dropdown set to 25), "Filter rows" (text input), "Search this table" (text input), and "Sort by key" (dropdown set to "None"). An "Extra options" button is also present. The result table has columns "f\_name" and "l\_name". The data rows are:

f_name	l_name
David	Lee
Sarah	Davis
Jessica	Anderson

**Explanation:** This query can be used when we want to display the list of students enrolled in a specific class such as a student roster page or an instructor dashboard.

This query retrieves the first and last names of students enrolled in the "Databases" class by joining the student, attendance, and class tables and filtering for that specific class name..

### b. Select Using a Nested Query

List the student ids who are attending at least one class that has more than 3 students enrolled.

```
SELECT s.student_id  
FROM Project_Student s  
WHERE s.student_id IN (  
    SELECT a.student_id  
    FROM Project_Attends a  
    JOIN Project_Class c ON a.class_id = c.class_id  
    WHERE c.number_of_students > 3  
);
```

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

```
SELECT s.student_id FROM Project_Student s WHERE s.student_id IN ( SELECT a.student_id FROM Project_Attends a JOIN Project_Class c ON a.class_id = c.class_id WHERE c.number_of_students > 3 );
```

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

	student_id
<input type="checkbox"/>	100000001
<input type="checkbox"/>	100000003
<input type="checkbox"/>	100000010
<input type="checkbox"/>	100000005

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

**Explanation:** This query can be used when retrieving the IDs of students who are enrolled in classes that have more than three students. This can be used for filtering or eligibility checks. The inner query finds all student\_ids who attend a class where number\_of\_students > 3 by using JOIN and WHERE. The outer query selects the names of those students from the student table.

### c. Select Using an Aggregate Function

Find the total number of students attending the class 'Intro to CS'

```
SELECT COUNT(a.student_id) AS total_students
FROM Project_Attends a
JOIN Project_Class c ON a.class_id = c.class_id
WHERE c.class_name = 'Intro to CS';
```

Your SQL query has been executed successfully.

```
SELECT COUNT(a.student_id) AS total_students FROM Project_Attends a JOIN Project_Class c ON a.class_id = c.class_id WHERE c.class_name = 'Intro to CS';
```

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

total_students
3

Extra options

**Explanation:** This can be used to find out how many students are enrolled in a specific class. This will be used to manage enrollment or displaying class statistics. I select the COUNT(a.student\_id) to retrieve the number of students who are attending the specific class and rename the column as total\_students. The JOIN ensures we only include students from the class named 'Intro to CS'.

## 2. INSERT query

```
INSERT INTO Project_Student (student_id, f_name, l_name, number_of_classes, email)
VALUES (100000011, 'Nina', 'Wells', 0, ninawells@example.com);
```

✓ 1 row inserted. (Query took 0.0002 seconds.)

```
INSERT INTO Project_Student (student_id, f_name, l_name, number_of_classes, email) VALUES (100000011, 'Nina', 'Wells', 0, 'ninawells@example.com');
```

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

**Explanation:** This query would be used when we are adding a new student, such as 'Nina Wells', to the database.

### 3. UPDATE query

```
UPDATE Project_Class
```

```
SET instructor_id= 2007
```

```
WHERE class_id = 30010;
```

✓ 1 row affected. (Query took 0.0004 seconds.)

```
UPDATE Project_Class SET instructor_id= 2007 WHERE class_id = 30010;
```

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

**Explanation:** This query would be used when updating a class with a new or different instructor.

### 4. DELETE query

```
DELETE FROM Project_Student
```

```
WHERE student_id = 100000011;
```

✓ 1 row deleted. (Query took 0.0002 seconds.)

```
DELETE FROM Project_Student WHERE student_id = 100000011;
```

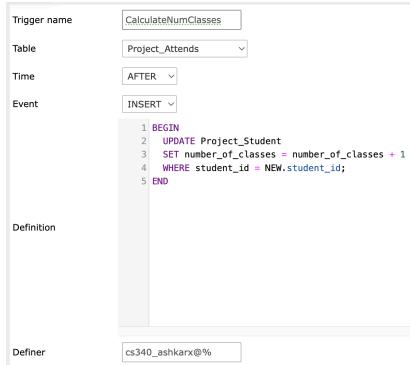
[ Edit inline ] [ Edit ] [ Create PHP code ]

**Explanation:** This query will be used when deleting a student, such as Nina Wells, which was previously added to the database. It uses the student\_id.

## 5. Triggers

### A. Trigger 1 - CalculateNumClasses

```
CREATE TRIGGER `CalculateNumClasses`  
AFTER INSERT ON `Project_Attends`  
FOR EACH ROW BEGIN  
    UPDATE Project_Student  
    SET number_of_classes = number_of_classes + 1  
    WHERE student_id = New.student_id;  
END;
```



**Explanation:** Every time we assign a new class for a student, it increases the number of classes a student is taking. It just increments the number of classes by 1 for each class.

### B. Trigger 2 - MaxNumberofStudents

```

CREATE TRIGGER `MaxNumberofStudents`
BEFORE INSERT ON `Project_Attends`
FOR EACH ROW BEGIN
    DECLARE student_count INT;

    SELECT COUNT(*) INTO student_count
        FROM Project_Attends
        WHERE class_id = NEW.class_id;

    IF (student_count >= 50) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'A class can't have more than 50 students.';
    END IF;
END;

```

	Name	Table	Time	Event			
<input type="checkbox"/>	CalculateNumClasses	Project_Attends	AFTER	INSERT	Edit	Export	Drop
<input type="checkbox"/>	Max40Hours	WORKS_ON_wk4	BEFORE	INSERT	Edit	Export	Drop
<input type="checkbox"/>	MaxNumberofStudents	Project_Attends	BEFORE	INSERT	Edit	Export	Drop

**Explanation:** Automatically checks if the class has reached the maximum number of students whenever we add a student to a class. It will give an error whenever it reaches 50 students in a class.

### C. Function 3 - getAssignments

```

CREATE FUNCTION getAssignments (p_student_id INT)
RETURNS VARCHAR(1000)
READS SQL DATA
BEGIN
    DECLARE assignment_list VARCHAR(1000);
    SELECT GROUP_CONCAT(
        CONCAT('Class ', c.class_name, ': Due ', a.due_date)
        ORDER BY a.due_date SEPARATOR '|'
    )
    INTO assignment_list
    FROM Project_Assignment a
    JOIN Project_Class c ON a.class_id = c.class_id
    JOIN Project_Attends attends ON attends.class_id = c.class_id
    WHERE attends.student_id = p_student_id;

    RETURN IFNULL(assignment_list, 'No assignments found.');
END;

```

	Name	Type	Returns				
<input type="checkbox"/>	getAssignments	FUNCTION	varchar(1000)	 Edit	 Execute	 Export	 Drop

**Explanation:** This function returns a list of all the assignments' names and due dates for a student in a database and sorts them by due date. This could be used at the beginning of the web page, so the students can immediately see what assignments are due.