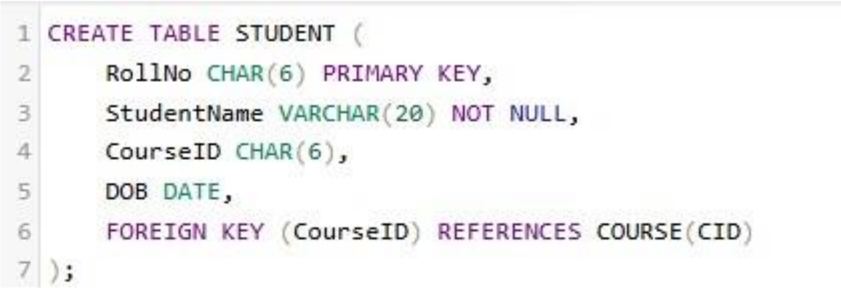
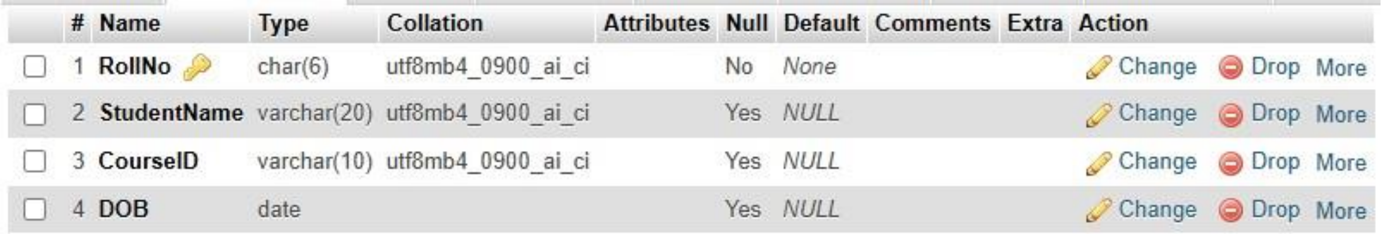
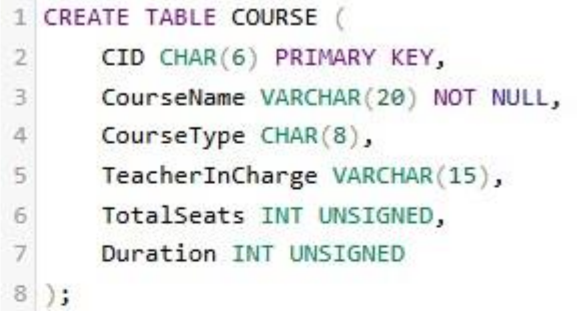
Creating Database Tables Like Student,Course,Admission.



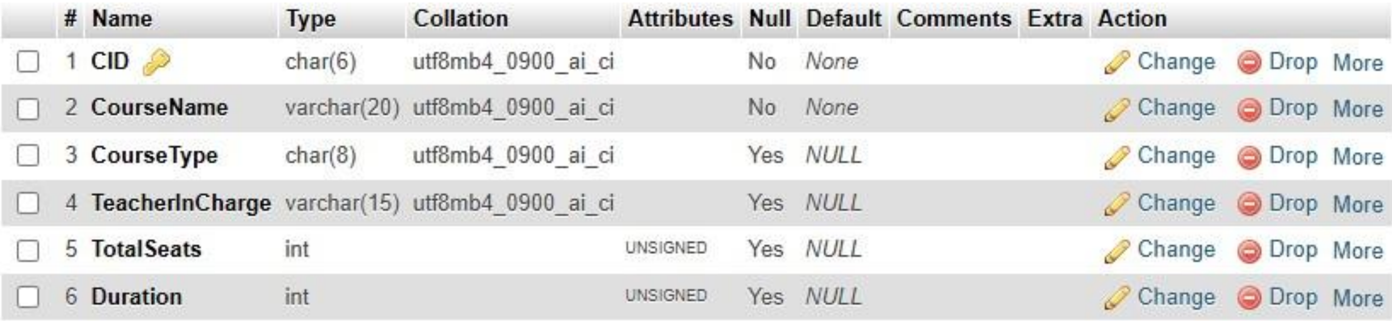
```
CREATE TABLE STUDENT (
    RollNo CHAR(6) PRIMARY KEY,
    StudentName VARCHAR(20) NOT NULL,
    CourseID CHAR(6),
    DOB DATE,
    FOREIGN KEY (CourseID) REFERENCES COURSE(CID)
);
```



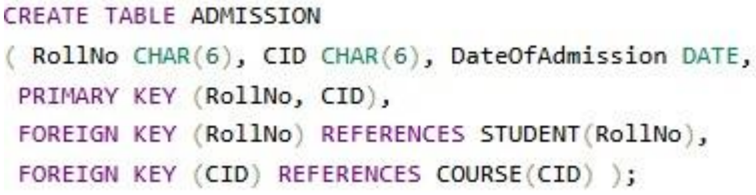
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**RollNo** 🔑	char(6)	utf8mb4_0900_ai_ci		No	*None*			Change Drop More
□	2	**StudentName**	varchar(20)	utf8mb4_0900_ai_ci		Yes	*NULL*			Change Drop More
□	3	**CourseID**	varchar(10)	utf8mb4_0900_ai_ci		Yes	*NULL*			Change Drop More
□	4	**DOB**	date			Yes	*NULL*			Change Drop More



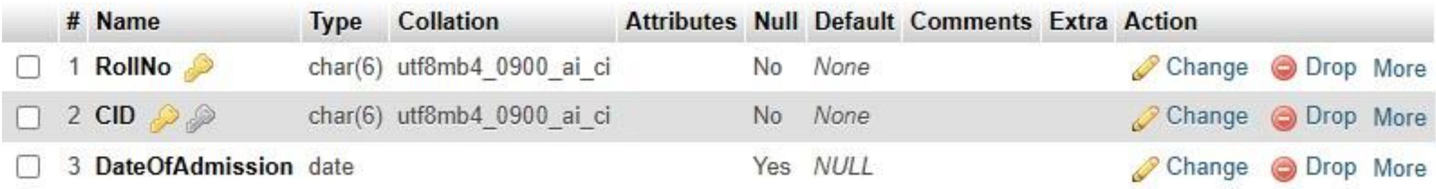
```
CREATE TABLE COURSE (
    CID CHAR(6) PRIMARY KEY,
    CourseName VARCHAR(20) NOT NULL,
    CourseType CHAR(8),
    TeacherInCharge VARCHAR(15),
    TotalSeats INT UNSIGNED,
    Duration INT UNSIGNED
);
```



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**CID** 🔑	char(6)	utf8mb4_0900_ai_ci		No	*None*			Change Drop More
□	2	**CourseName**	varchar(20)	utf8mb4_0900_ai_ci		No	*None*			Change Drop More
□	3	**CourseType**	char(8)	utf8mb4_0900_ai_ci		Yes	*NULL*			Change Drop More
□	4	**TeacherInCharge**	varchar(15)	utf8mb4_0900_ai_ci		Yes	*NULL*			Change Drop More
□	5	**TotalSeats**	int		UNSIGNED	Yes	*NULL*			Change Drop More
□	6	**Duration**	int		UNSIGNED	Yes	*NULL*			Change Drop More

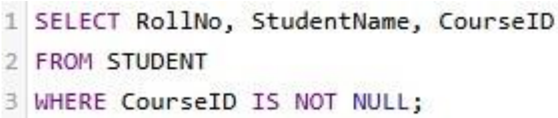


```
CREATE TABLE ADMISSION
( RollNo CHAR(6), CID CHAR(6), DateOfAdmission DATE,
 PRIMARY KEY (RollNo, CID),
 FOREIGN KEY (RollNo) REFERENCES STUDENT(RollNo),
 FOREIGN KEY (CID) REFERENCES COURSE(CID) );
```

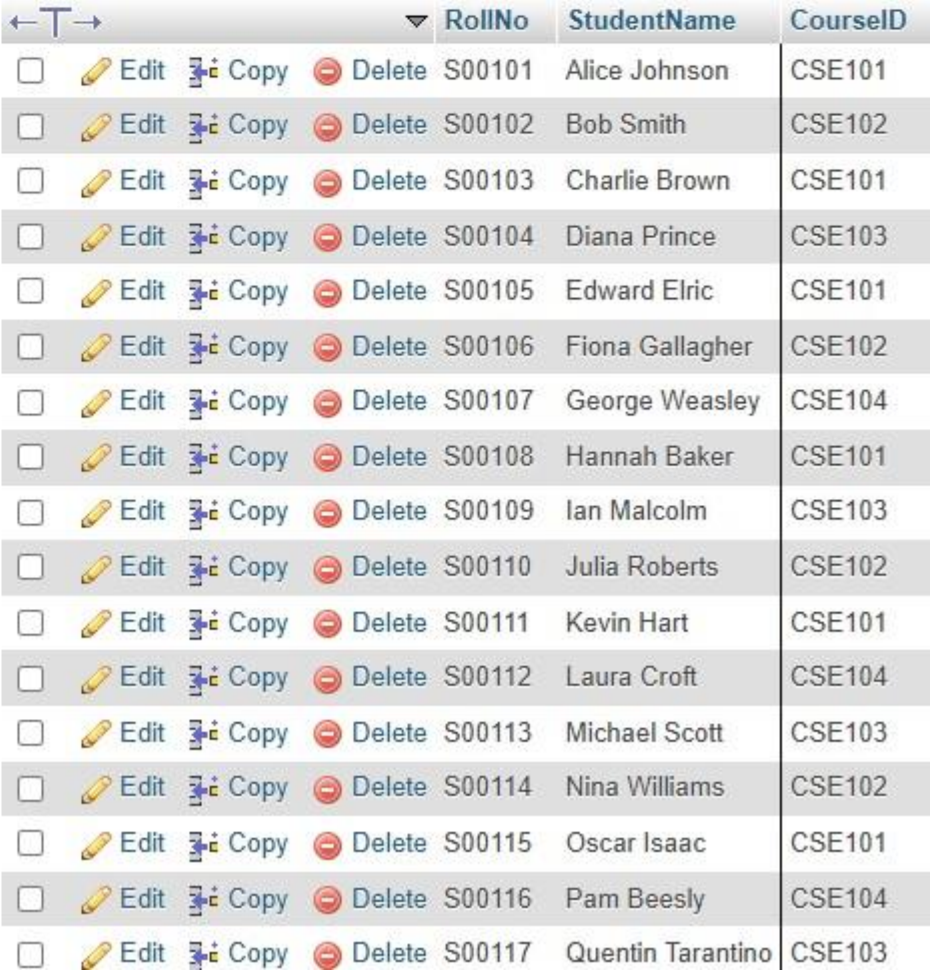


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**RollNo** 🔑	char(6)	utf8mb4_0900_ai_ci		No	*None*			Change Drop More
□	2	**CID** 🔑🔑	char(6)	utf8mb4_0900_ai_ci		No	*None*			Change Drop More
□	3	**DateOfAdmission**	date			Yes	*NULL*			Change Drop More

Q1. Retrieve names of students enrolled in any course.

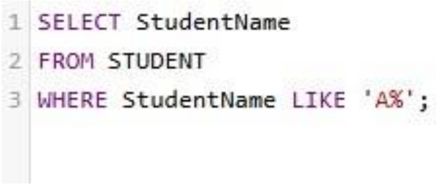


```sql
SELECT RollNo, StudentName, CourseID
FROM STUDENT
WHERE CourseID IS NOT NULL;
```

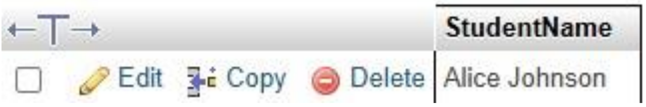


←T→			▼	RollNo	StudentName	CourseID
□	Edit	Copy	Delete	S00101	Alice Johnson	CSE101
□	Edit	Copy	Delete	S00102	Bob Smith	CSE102
□	Edit	Copy	Delete	S00103	Charlie Brown	CSE101
□	Edit	Copy	Delete	S00104	Diana Prince	CSE103
□	Edit	Copy	Delete	S00105	Edward Elric	CSE101
□	Edit	Copy	Delete	S00106	Fiona Gallagher	CSE102
□	Edit	Copy	Delete	S00107	George Weasley	CSE104
□	Edit	Copy	Delete	S00108	Hannah Baker	CSE101
□	Edit	Copy	Delete	S00109	Ian Malcolm	CSE103
□	Edit	Copy	Delete	S00110	Julia Roberts	CSE102
□	Edit	Copy	Delete	S00111	Kevin Hart	CSE101
□	Edit	Copy	Delete	S00112	Laura Croft	CSE104
□	Edit	Copy	Delete	S00113	Michael Scott	CSE103
□	Edit	Copy	Delete	S00114	Nina Williams	CSE102
□	Edit	Copy	Delete	S00115	Oscar Isaac	CSE101
□	Edit	Copy	Delete	S00116	Pam Beesly	CSE104
□	Edit	Copy	Delete	S00117	Quentin Tarantino	CSE103

Q2. Retrieve students' names starting with letter 'A'.

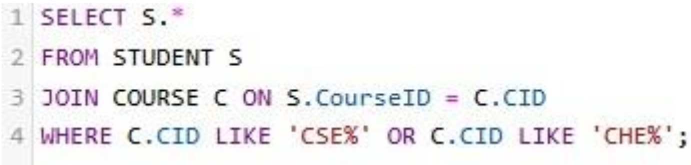


```sql
SELECT StudentName
FROM STUDENT
WHERE StudentName LIKE 'A%';
```

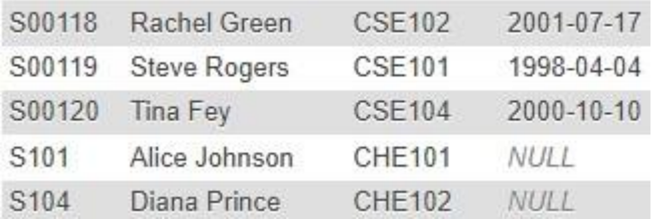


←T→				StudentName
□	Edit	Copy	Delete	Alice Johnson

Q3. Retrieve students' details studying in courses 'computer science' or 'chemistry'.

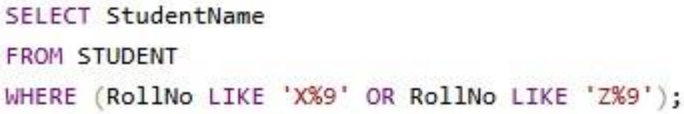


```sql
SELECT S.*
FROM STUDENT S
JOIN COURSE C ON S.CourseID = C.CID
WHERE C.CID LIKE 'CSE%' OR C.CID LIKE 'CHE%';
```

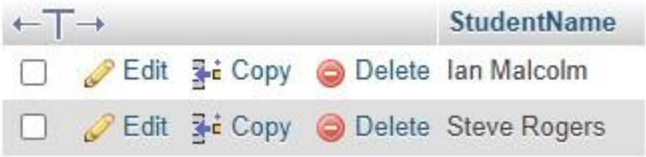


S00118	Rachel Green	CSE102	2001-07-17
S00119	Steve Rogers	CSE101	1998-04-04
S00120	Tina Fey	CSE104	2000-10-10
S101	Alice Johnson	CHE101	NULL
S104	Diana Prince	CHE102	NULL

Q4 . Retrieve students’ names whose roll no either starts with ‘X’ or ‘Z’ and ends with ‘9’

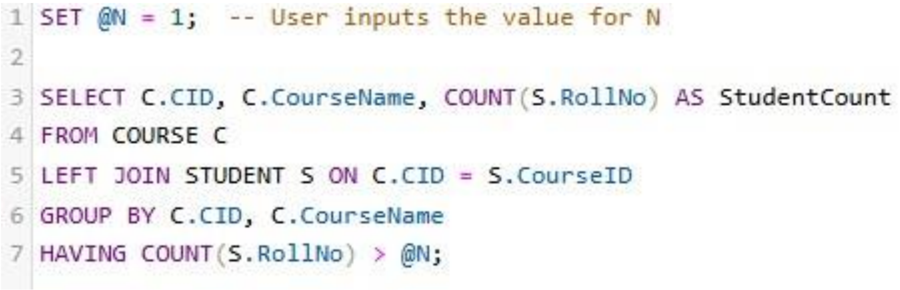


```sql
SELECT StudentName
FROM STUDENT
WHERE (RollNo LIKE 'X%9' OR RollNo LIKE 'Z%9');
```



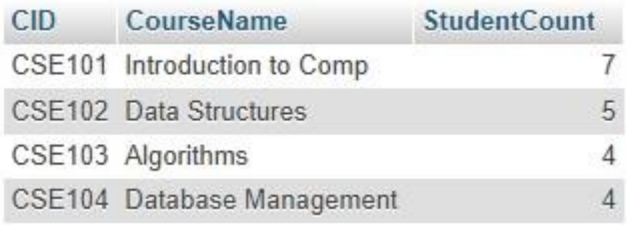
←T→				StudentName
□	Edit	Copy	Delete	Ian Malcolm
□	Edit	Copy	Delete	Steve Rogers

Q5. Find course details with more than N students enrolled where n is from user



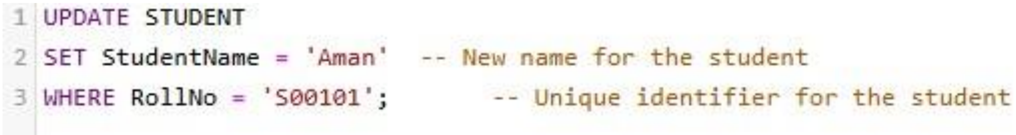
```
SET @N = 1;  -- User inputs the value for N

SELECT C.CID, C.CourseName, COUNT(S.RollNo) AS StudentCount
FROM COURSE C
LEFT JOIN STUDENT S ON C.CID = S.CourseID
GROUP BY C.CID, C.CourseName
HAVING COUNT(S.RollNo) > @N;
```

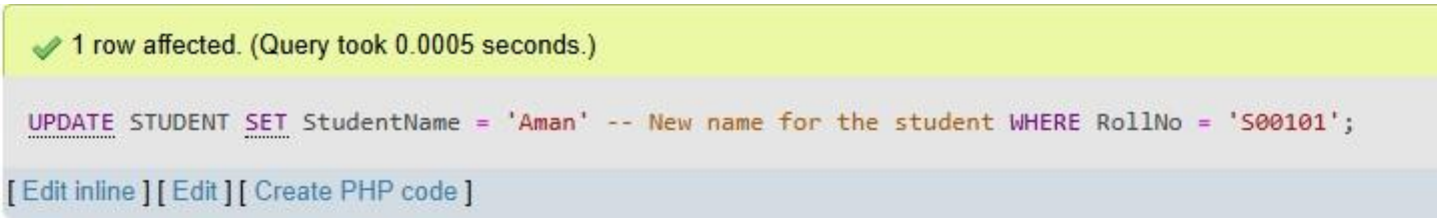


CID	CourseName	StudentCount
CSE101	Introduction to Comp	7
CSE102	Data Structures	5
CSE103	Algorithms	4
CSE104	Database Management	4

Q6. Update student table for modifying a student name.



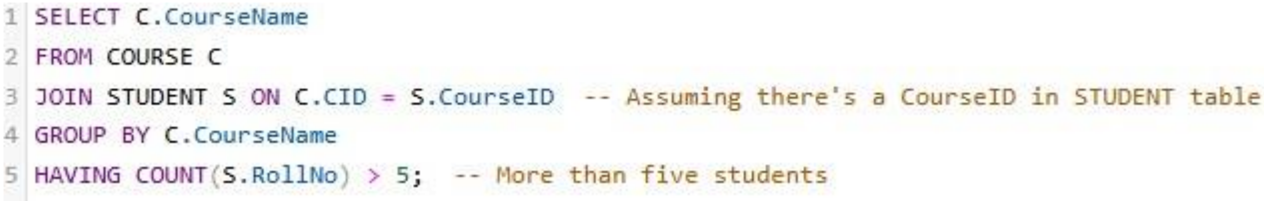
```
UPDATE STUDENT
SET StudentName = 'Aman'   -- New name for the student
WHERE RollNo = 'S00101';        -- Unique identifier for the student
```



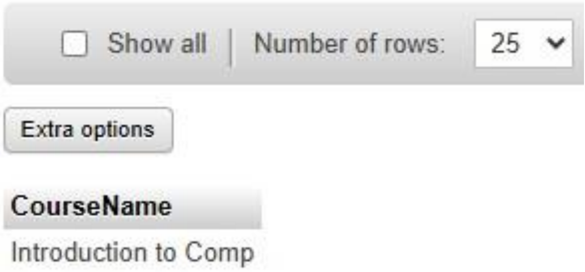
✓ 1 row affected. (Query took 0.0005 seconds.)

```
UPDATE STUDENT SET StudentName = 'Aman' -- New name for the student WHERE RollNo = 'S00101';
```

8. Find course names in which more than five students have enrolled



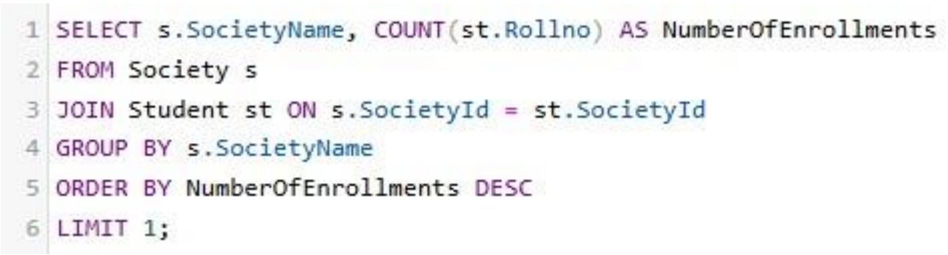
```
SELECT C.CourseName
FROM COURSE C
JOIN STUDENT S ON C.CID = S.CourseID   -- Assuming there's a CourseID in STUDENT table
GROUP BY C.CourseName
HAVING COUNT(S.RollNo) > 5;   -- More than five students
```



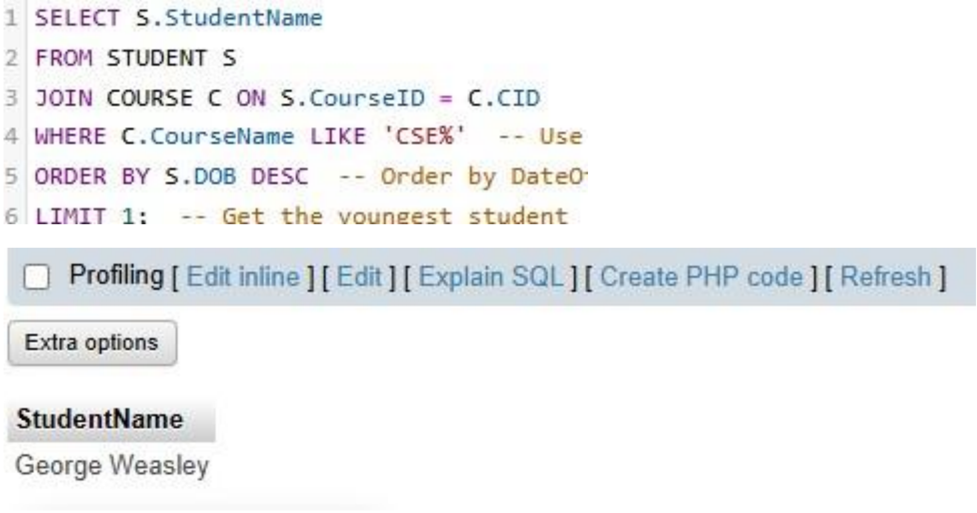
☐ Show all | Number of rows: 25

CourseName
Introduction to Comp

9. Find the name of youngest student enrolled in course ‘CSE or BSC Cs(P)’.



```sql
SELECT s.SocietyName, COUNT(st.Rollno) AS NumberOfEnrollments
FROM Society s
JOIN Student st ON s.SocietyId = st.SocietyId
GROUP BY s.SocietyName
ORDER BY NumberOfEnrollments DESC
LIMIT 1;
```

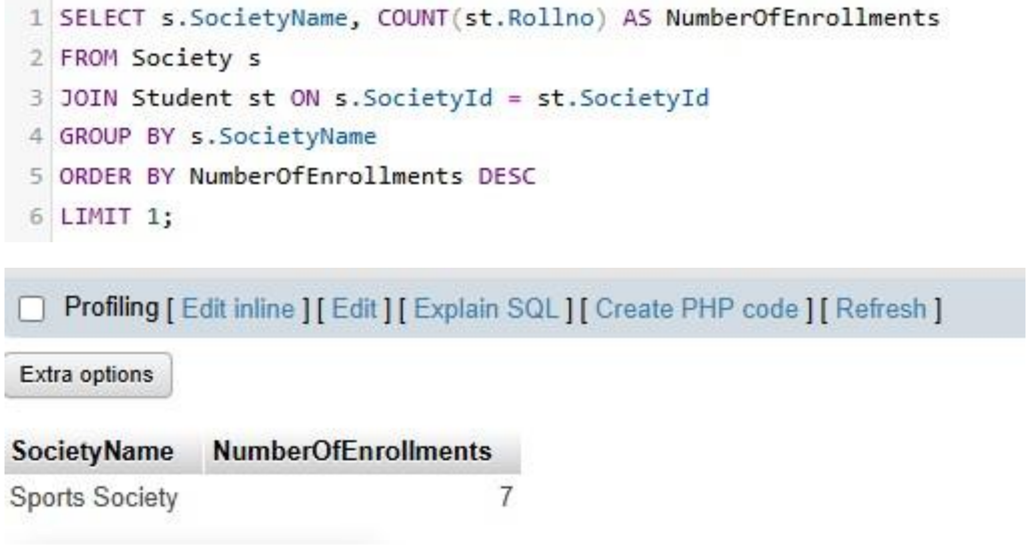


```sql
SELECT S.StudentName
FROM STUDENT S
JOIN COURSE C ON S.CourseID = C.CID
WHERE C.CourseName LIKE 'CSE%'  -- Use
ORDER BY S.DOB DESC  -- Order by DateO
LIMIT 1;  -- Get the youngest student
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options

StudentName
George Weasley

10. Find the name of most popular society (on the basis of enrolled students)

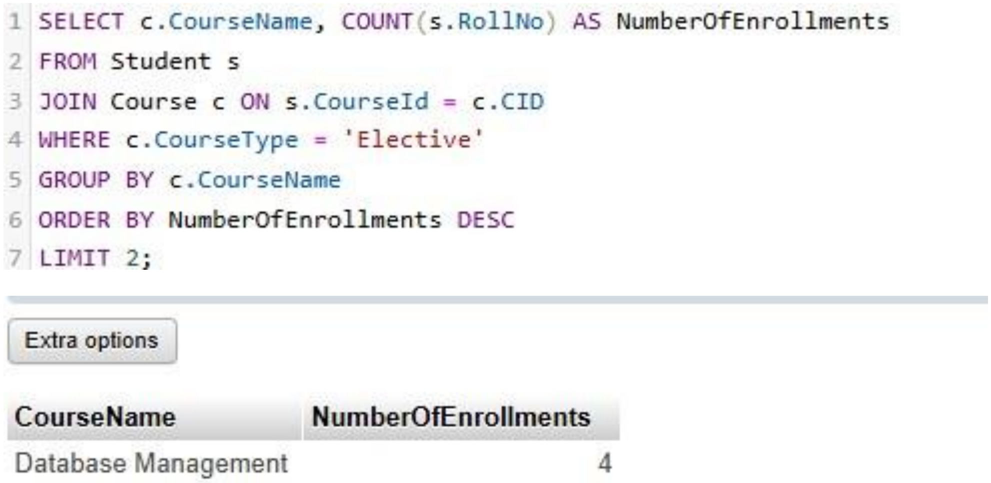


```sql
SELECT s.SocietyName, COUNT(st.Rollno) AS NumberOfEnrollments
FROM Society s
JOIN Student st ON s.SocietyId = st.SocietyId
GROUP BY s.SocietyName
ORDER BY NumberOfEnrollments DESC
LIMIT 1;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options

SocietyName	NumberOfEnrollments
Sports Society	7

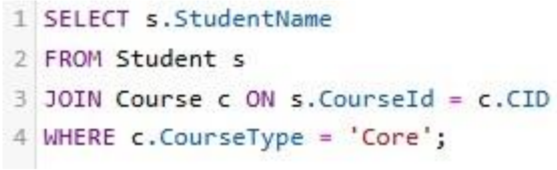
11. Find the name of two popular part time courses (on the basis of enrolled students)



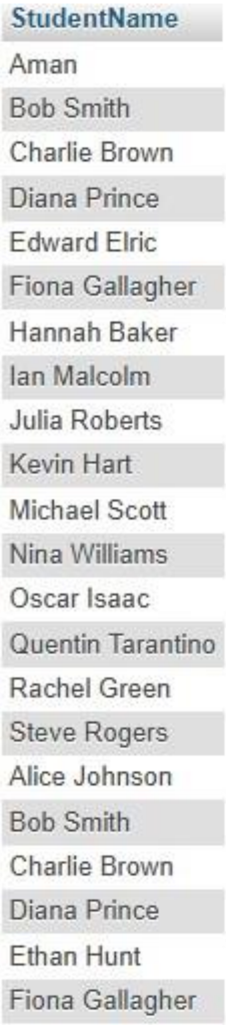
```sql
SELECT c.CourseName, COUNT(s.RollNo) AS NumberOfEnrollments
FROM Student s
JOIN Course c ON s.CourseId = c.CID
WHERE c.CourseType = 'Elective'
GROUP BY c.CourseName
ORDER BY NumberOfEnrollments DESC
LIMIT 2;
```

CourseName	NumberOfEnrollments
Database Management	4

12. Find the student names who are admitted to full time courses only.

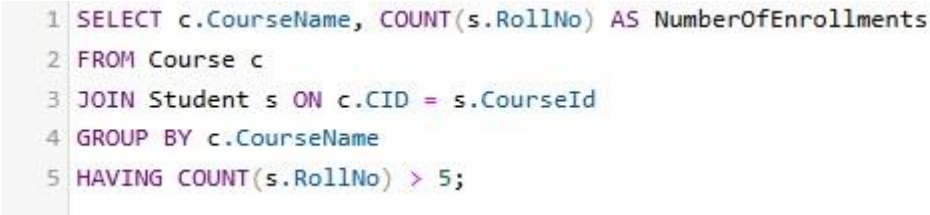


```sql
SELECT s.StudentName
FROM Student s
JOIN Course c ON s.CourseId = c.CID
WHERE c.CourseType = 'Core';
```

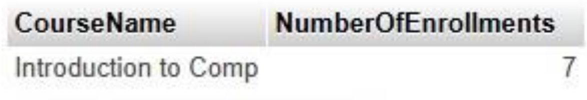


StudentName
Aman
Bob Smith
Charlie Brown
Diana Prince
Edward Elric
Fiona Gallagher
Hannah Baker
Ian Malcolm
Julia Roberts
Kevin Hart
Michael Scott
Nina Williams
Oscar Isaac
Quentin Tarantino
Rachel Green
Steve Rogers
Alice Johnson
Bob Smith
Charlie Brown
Diana Prince
Ethan Hunt
Fiona Gallagher

13. Find course names in which more than 30 students took admission

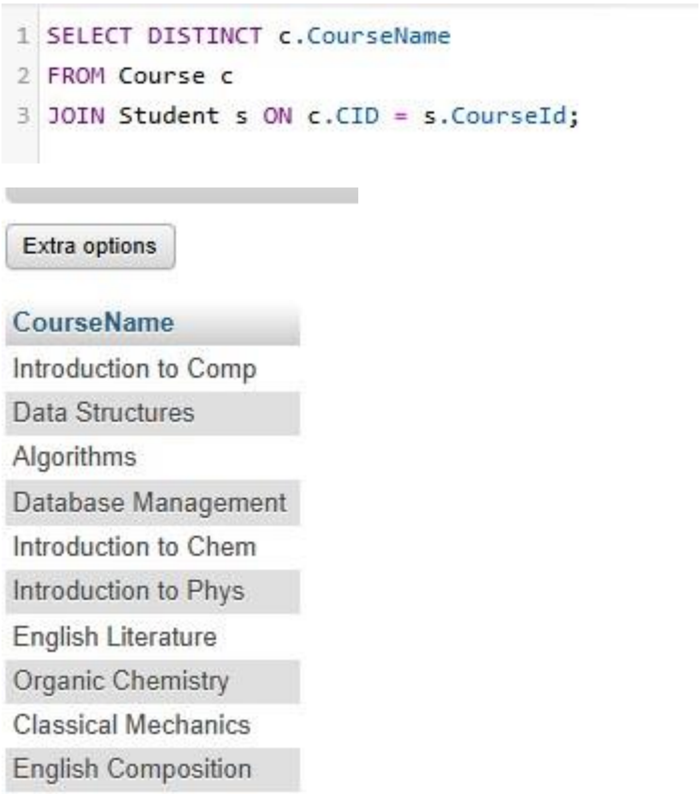


```sql
SELECT c.CourseName, COUNT(s.RollNo) AS NumberOfEnrollments
FROM Course c
JOIN Student s ON c.CID = s.CourseId
GROUP BY c.CourseName
HAVING COUNT(s.RollNo) > 5;
```



CourseName	NumberOfEnrollments
Introduction to Comp	7

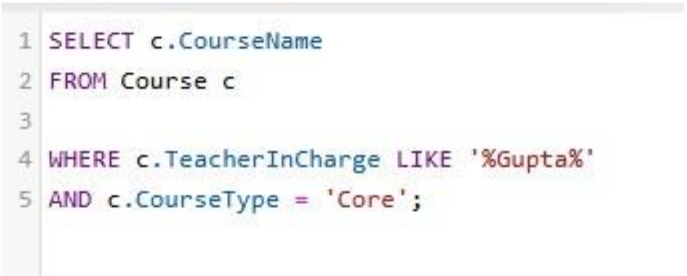
14. Find names of all students who took admission to any course and course names in which at least one student has enrolled



```
SELECT DISTINCT c.CourseName
FROM Course c
JOIN Student s ON c.CID = s.CourseId;
```

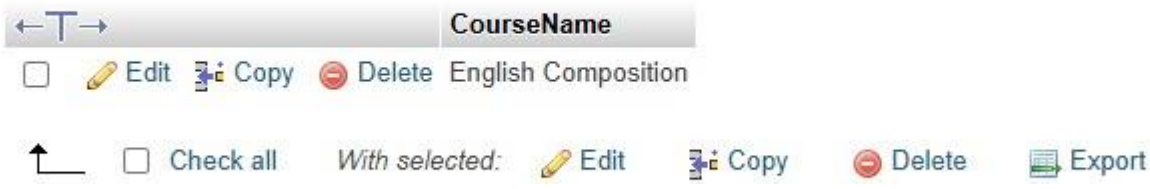
CourseName
Introduction to Comp
Data Structures
Algorithms
Database Management
Introduction to Chem
Introduction to Phys
English Literature
Organic Chemistry
Classical Mechanics
English Composition

15. Find course names such that its teacher-in-charge has a name with ‘Gupta’ in it and the course is full time.



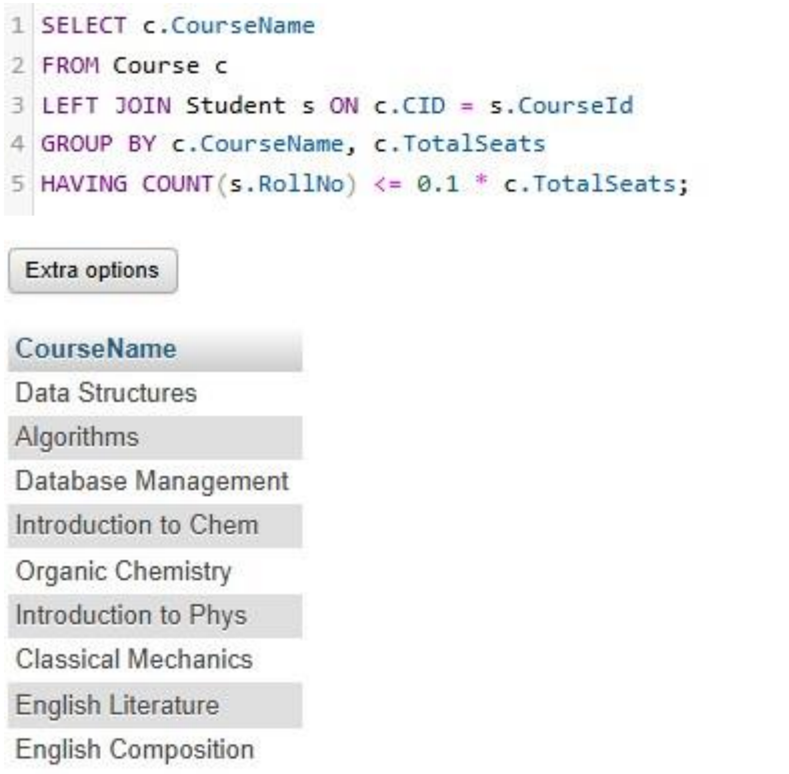
```
SELECT c.CourseName
FROM Course c

WHERE c.TeacherInCharge LIKE '%Gupta%'
AND c.CourseType = 'Core';
```



CourseName
English Composition

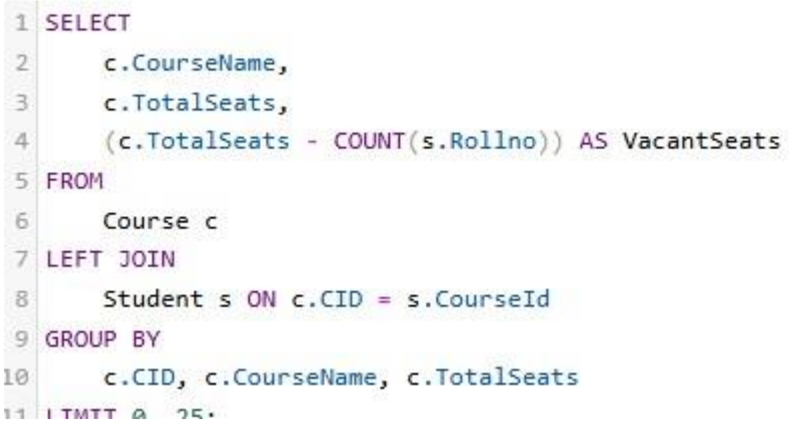
16. Find the course names in which the number of enrolled students is only 10% of its total seats.



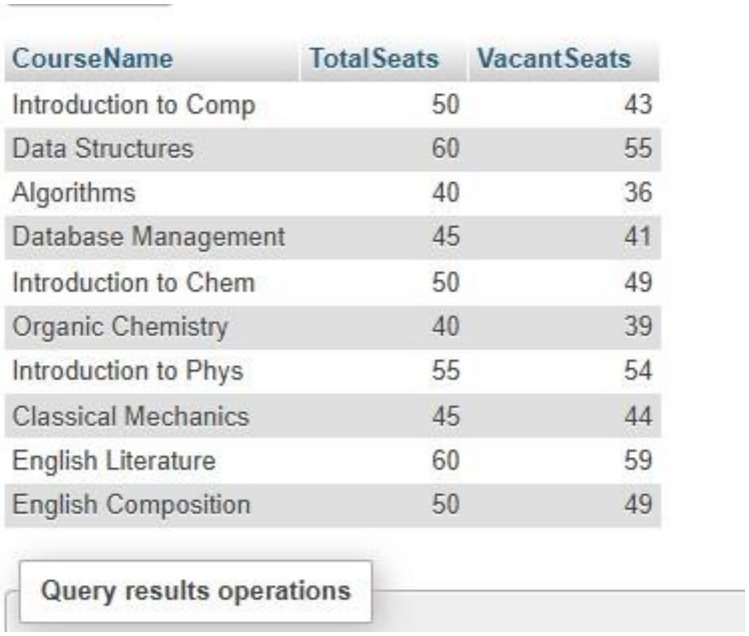
```sql
SELECT c.CourseName
FROM Course c
LEFT JOIN Student s ON c.CID = s.CourseId
GROUP BY c.CourseName, c.TotalSeats
HAVING COUNT(s.RollNo) <= 0.1 * c.TotalSeats;
```

CourseName
Data Structures
Algorithms
Database Management
Introduction to Chem
Organic Chemistry
Introduction to Phys
Classical Mechanics
English Literature
English Composition

17. Display the vacant seats for each course

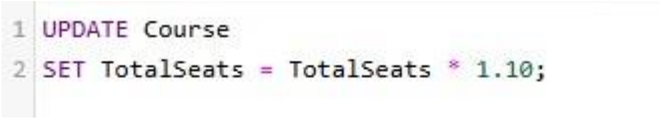


```sql
SELECT
    c.CourseName,
    c.TotalSeats,
    (c.TotalSeats - COUNT(s.Rollno)) AS VacantSeats
FROM
    Course c
LEFT JOIN
    Student s ON c.CID = s.CourseId
GROUP BY
    c.CID, c.CourseName, c.TotalSeats
LIMIT 0, 25;
```

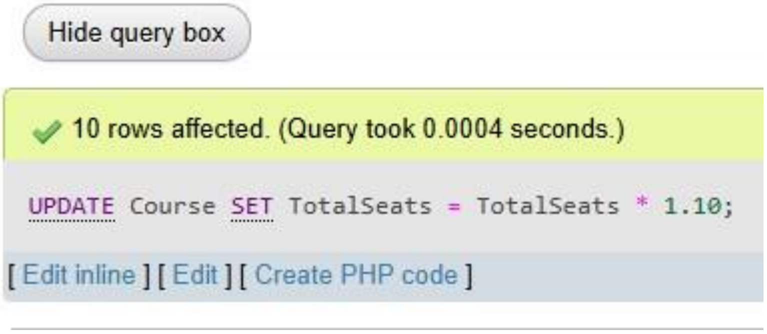


CourseName	TotalSeats	VacantSeats
Introduction to Comp	50	43
Data Structures	60	55
Algorithms	40	36
Database Management	45	41
Introduction to Chem	50	49
Organic Chemistry	40	39
Introduction to Phys	55	54
Classical Mechanics	45	44
English Literature	60	59
English Composition	50	49

18. Increment Total Seats of each course by 10%



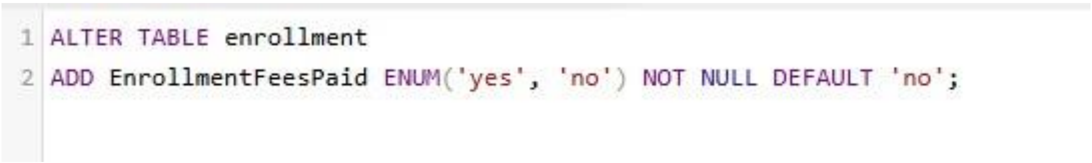
```
UPDATE Course
SET TotalSeats = TotalSeats * 1.10;
```



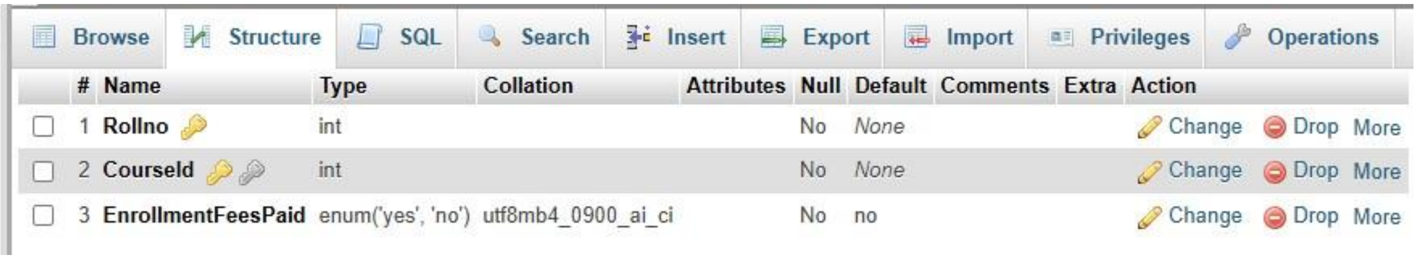
✓ 10 rows affected. (Query took 0.0004 seconds.)

```
UPDATE Course SET TotalSeats = TotalSeats * 1.10;
```

19. Add enrollment fees paid (‘yes’/’No’) field in the enrollment table. 20. Update date of admission of all the courses by 1 year.

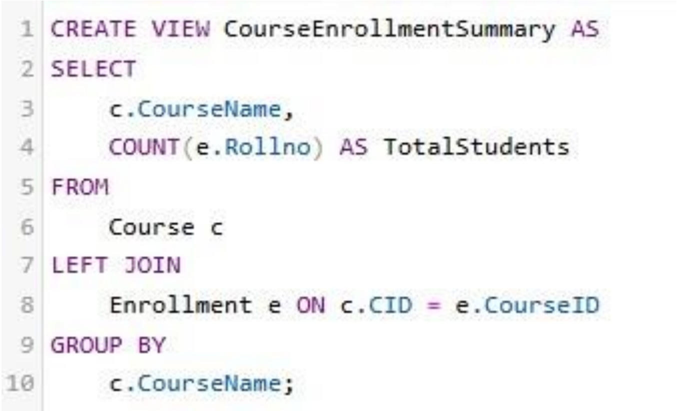


```
ALTER TABLE enrollment
ADD EnrollmentFeesPaid ENUM('yes', 'no') NOT NULL DEFAULT 'no';
```



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	Rollno	int			No	None			Change Drop More
□	2	CourseId	int			No	None			Change Drop More
□	3	EnrollmentFeesPaid	enum('yes', 'no')	utf8mb4_0900_ai_ci		No	no			Change Drop More

21. Create a view to keep track of course names with the total number of students enrolled in it.



```
CREATE VIEW CourseEnrollmentSummary AS
SELECT
    c.CourseName,
    COUNT(e.Rollno) AS TotalStudents
FROM
    Course c
LEFT JOIN
    Enrollment e ON c.CID = e.CourseID
GROUP BY
    c.CourseName;
```

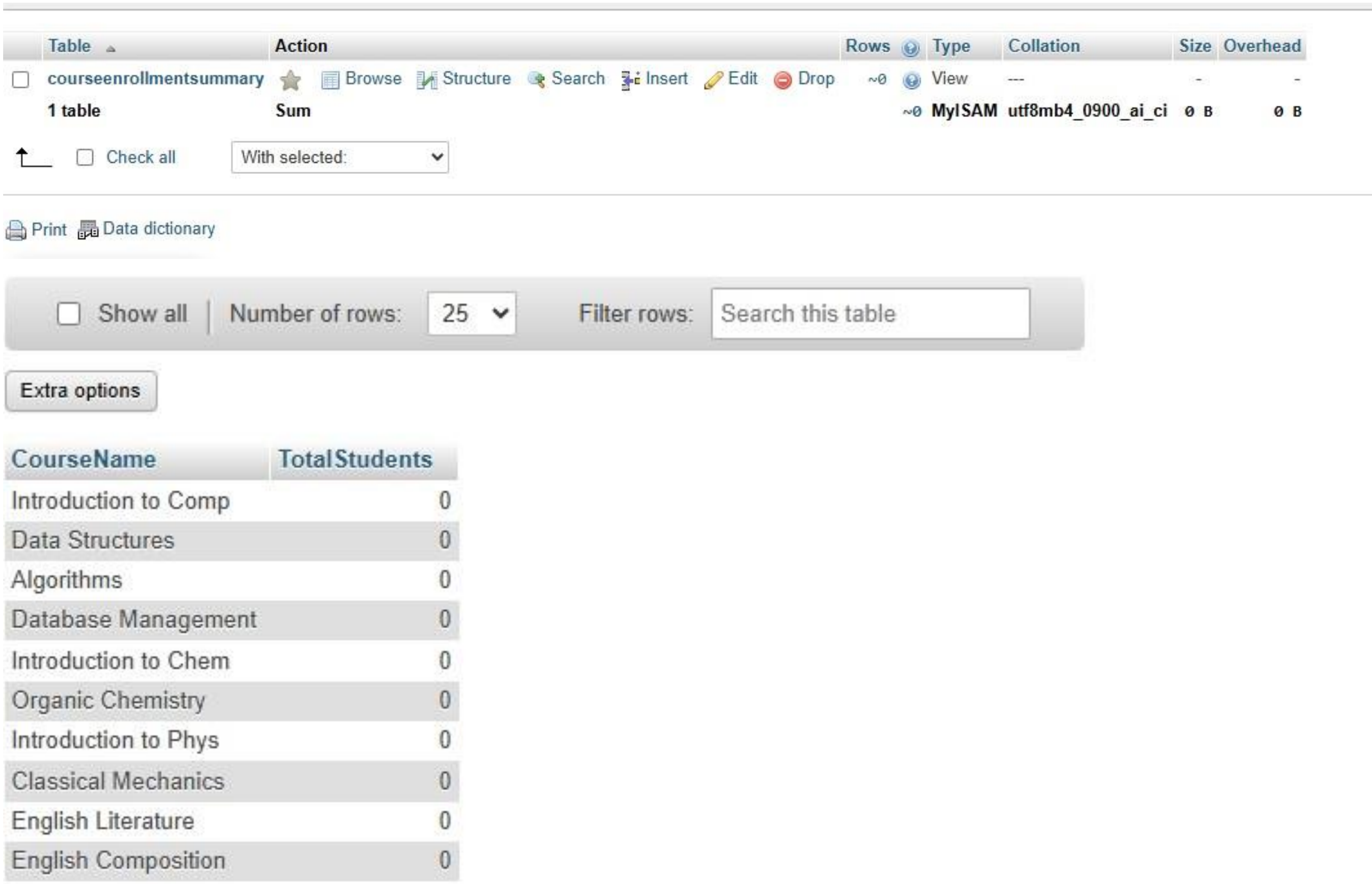
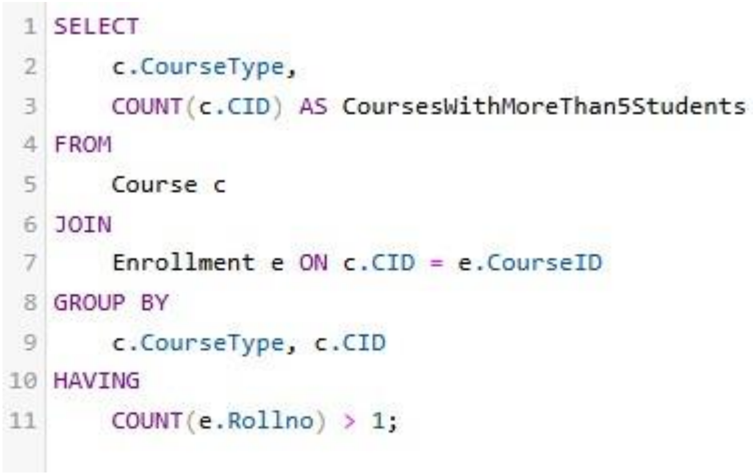


Table	Action	Rows	Type	Collation	Size	Overhead
□ courseenrollmentsummary	☆ Browse Structure Search Insert Edit Drop	~0	View	---	-	-
1 table	Sum	~0	MyISAM	utf8mb4_0900_ai_ci	0 B	0 B

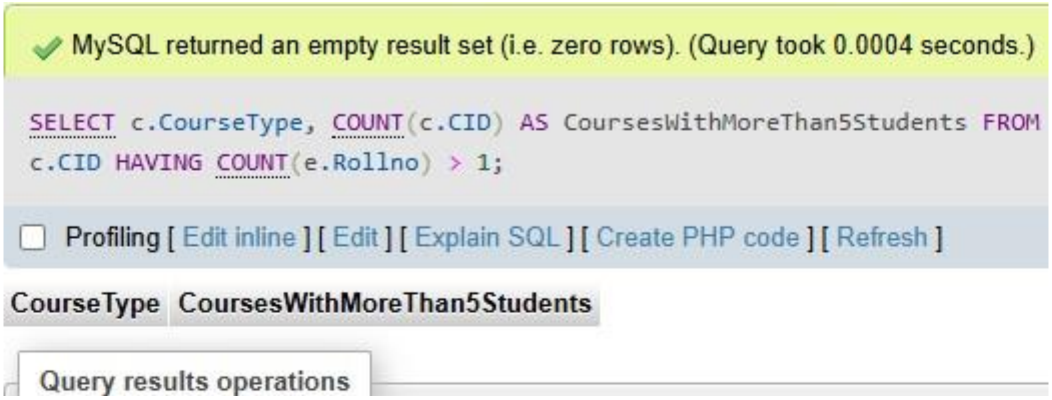
□ Show all | Number of rows: 25 | Filter rows: Search this table

CourseName	TotalStudents
Introduction to Comp	0
Data Structures	0
Algorithms	0
Database Management	0
Introduction to Chem	0
Organic Chemistry	0
Introduction to Phys	0
Classical Mechanics	0
English Literature	0
English Composition	0

22. Count the number of courses with more than 5 students enrolled for each type of course.



```
SELECT
    c.CourseType,
    COUNT(c.CID) AS CoursesWithMoreThan5Students
FROM
    Course c
JOIN
    Enrollment e ON c.CID = e.CourseID
GROUP BY
    c.CourseType, c.CID
HAVING
    COUNT(e.Rollno) > 1;
```



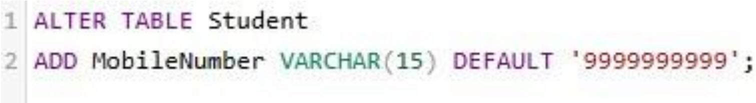
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
SELECT c.CourseType, COUNT(c.CID) AS CoursesWithMoreThan5Students FROM
c.CID HAVING COUNT(e.Rollno) > 1;
```

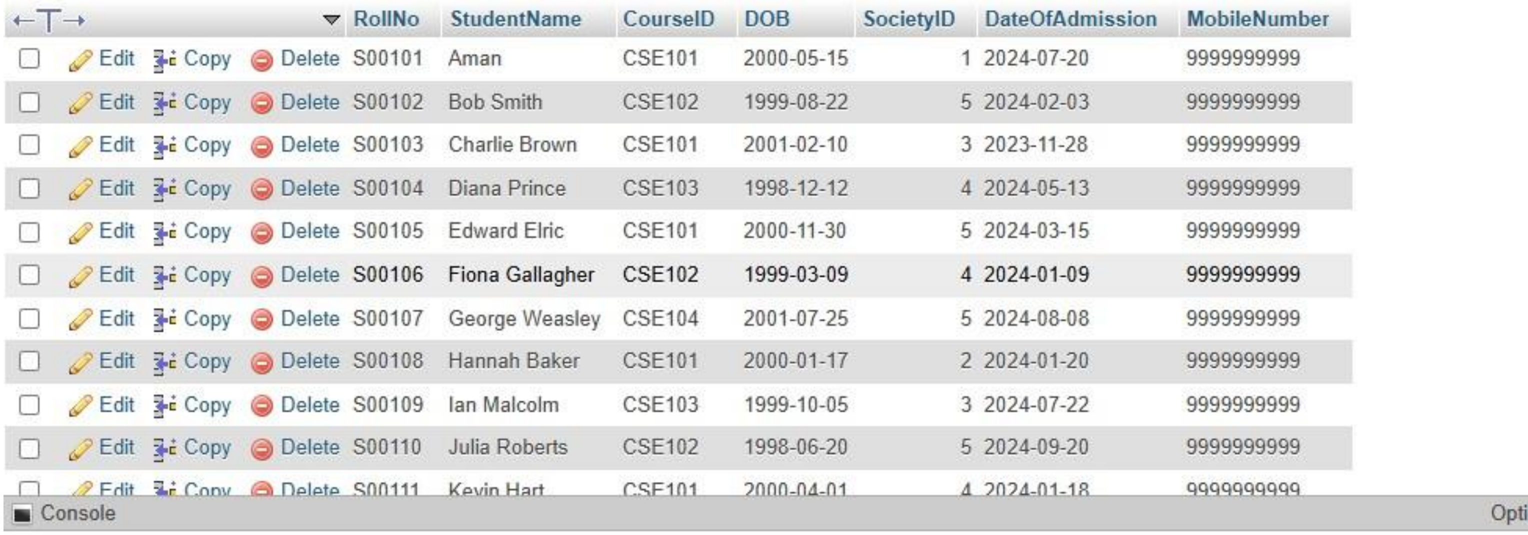
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

CourseType	CoursesWithMoreThan5Students

23. Add column Mobile number in student table with default value ‘9999999999’

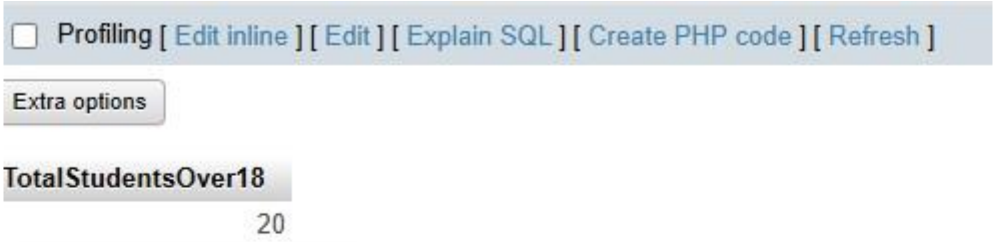


```
ALTER TABLE Student
ADD MobileNumber VARCHAR(15) DEFAULT '9999999999';
```



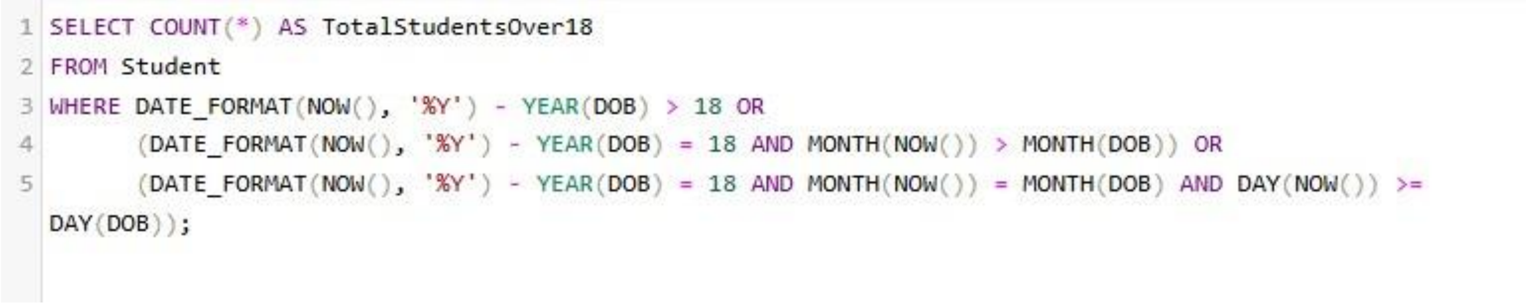
←T→			▼	RollNo	StudentName	CourseID	DOB	SocietyID	DateOfAdmission	MobileNumber
□	Edit	Copy	Delete	S00101	Aman	CSE101	2000-05-15	1	2024-07-20	9999999999
□	Edit	Copy	Delete	S00102	Bob Smith	CSE102	1999-08-22	5	2024-02-03	9999999999
□	Edit	Copy	Delete	S00103	Charlie Brown	CSE101	2001-02-10	3	2023-11-28	9999999999
□	Edit	Copy	Delete	S00104	Diana Prince	CSE103	1998-12-12	4	2024-05-13	9999999999
□	Edit	Copy	Delete	S00105	Edward Elric	CSE101	2000-11-30	5	2024-03-15	9999999999
□	Edit	Copy	Delete	S00106	Fiona Gallagher	CSE102	1999-03-09	4	2024-01-09	9999999999
□	Edit	Copy	Delete	S00107	George Weasley	CSE104	2001-07-25	5	2024-08-08	9999999999
□	Edit	Copy	Delete	S00108	Hannah Baker	CSE101	2000-01-17	2	2024-01-20	9999999999
□	Edit	Copy	Delete	S00109	Ian Malcolm	CSE103	1999-10-05	3	2024-07-22	9999999999
□	Edit	Copy	Delete	S00110	Julia Roberts	CSE102	1998-06-20	5	2024-09-20	9999999999
□	Edit	Copy	Delete	S00111	Kevin Hart	CSE101	2000-04-01	4	2024-01-18	9999999999

24. Find the total number of students whose age is > 18 years.



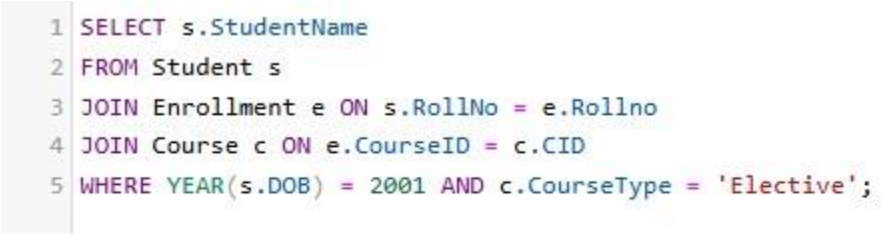
☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options

TotalStudentsOver18
20

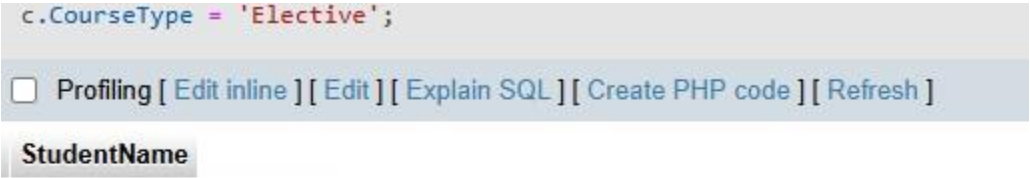


```
SELECT COUNT(*) AS TotalStudentsOver18
FROM Student
WHERE DATE_FORMAT(NOW(), '%Y') - YEAR(DOB) > 18 OR
      (DATE_FORMAT(NOW(), '%Y') - YEAR(DOB) = 18 AND MONTH(NOW()) > MONTH(DOB)) OR
      (DATE_FORMAT(NOW(), '%Y') - YEAR(DOB) = 18 AND MONTH(NOW()) = MONTH(DOB) AND DAY(NOW()) >=
DAY(DOB));
```

25.Find names of students who are born in 2001 and are admitted to at least one part time course.



```
SELECT s.StudentName
FROM Student s
JOIN Enrollment e ON s.RollNo = e.Rollno
JOIN Course c ON e.CourseID = c.CID
WHERE YEAR(s.DOB) = 2001 AND c.CourseType = 'Elective';
```

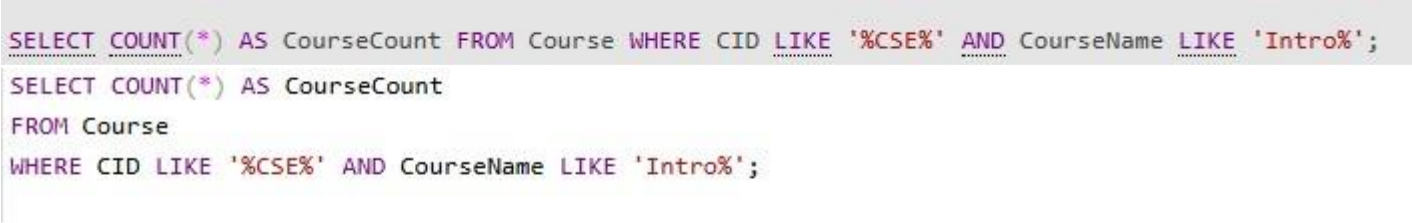


```
c.CourseType = 'Elective';
```

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

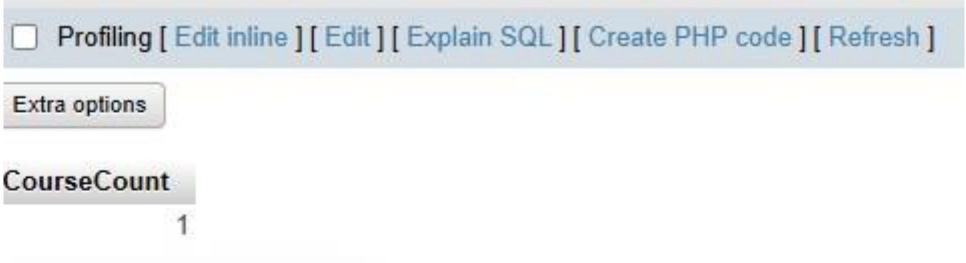
StudentName

26. Count all courses having ‘science’ in the name and starting with the word ‘BSc’.



```
SELECT COUNT(*) AS CourseCount FROM Course WHERE CID LIKE '%CSE%' AND CourseName LIKE 'Intro%';
```

```
SELECT COUNT(*) AS CourseCount
FROM Course
WHERE CID LIKE '%CSE%' AND CourseName LIKE 'Intro%';
```



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

CourseCount
1