1) CREATE DATABASE Registration;

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)
CREATE DATABASE Registration
[Edit inline] [Edit] [Create PHP code]

Please refer to 'DDL-Student Registration.docx' for additional SQL queries.

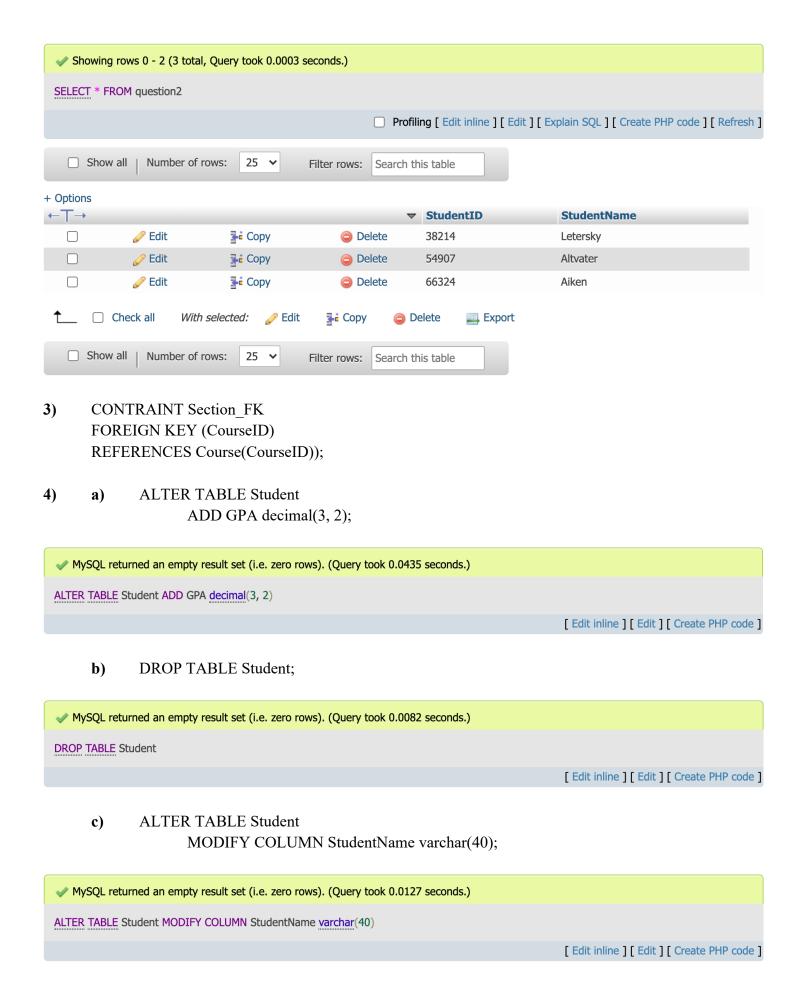


2) CREATE VIEW Question2 AS

SELECT s.StudentID, s.StudentName FROM Student AS s
WHERE s.StudentID != 70542;

SELECT *
FROM question2;

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0112 seconds.)
CREATE VIEW Question2 AS SELECT s.StudentID, s.StudentName FROM Student AS s WHERE s.StudentID != 70542
[Edit inline] [Edit] [Create PHP code]



5) a) INSERT INTO Student (StudentID, StudentName)
VALUES
(12345, 'Lapin');

INSERT INTO Student VALUES (12345, 'Lapin');

√ 1 row inserted. (Query took 0.0056 seconds.)

INSERT INTO Student (StudentID, StudentName) VALUES (12345, 'Lapin')

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

√ 1 row inserted. (Query took 0.0056 seconds.)

INSERT INTO Student VALUES (12345, 'Lapin')

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

b) DELETE FROM Student WHERE StudentID = 12345;

√ 1 row affected. (Query took 0.0056 seconds.)

DELETE FROM Student WHERE StudentID = 12345

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

c) ALTER TABLE Course

MODIFY COLUMN CourseName varchar(40);

UPDATE Course

SET CourseName = 'Introduction to Relational Databases'

WHERE CourseID = 'ISM 4212';

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0176 seconds.)

ALTER TABLE Course MODIFY COLUMN CourseName varchar(40)

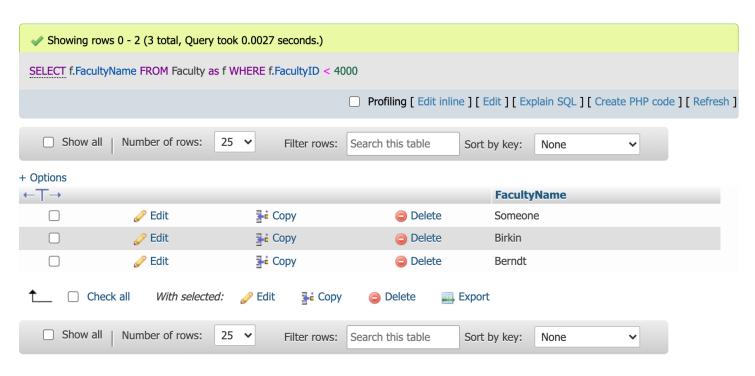
[Edit inline] [Edit] [Create PHP code]

1 row affected. (Query took 0.0006 seconds.)

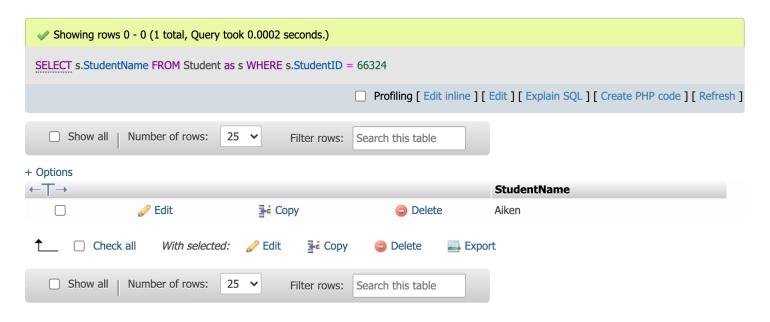
UPDATE Course SET CourseName = 'Introduction to Relational Databases' WHERE CourseID = 'ISM 4212'

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

- d) An error was thrown that the "Data too long for column 'CourseName' at row 1," in which I addressed it by modifying the column to have a longer maximum varchar length (please see above).
- 6) a) SELECT f.FacultyName FROM Faculty as f WHERE f.FacultyID < 4000;



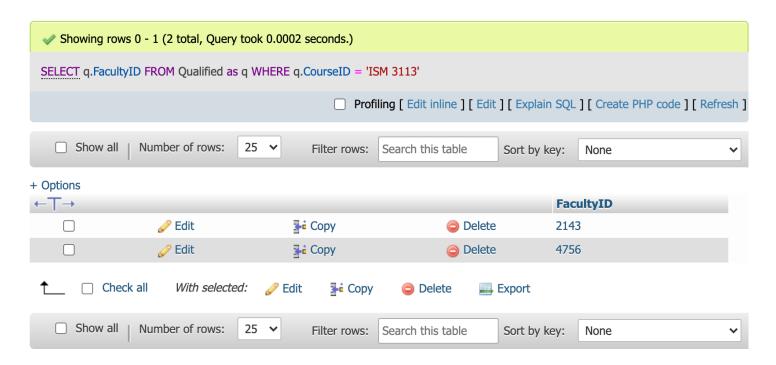
b) SELECT s.StudentName FROM Student as s WHERE s.StudentID = 66324;



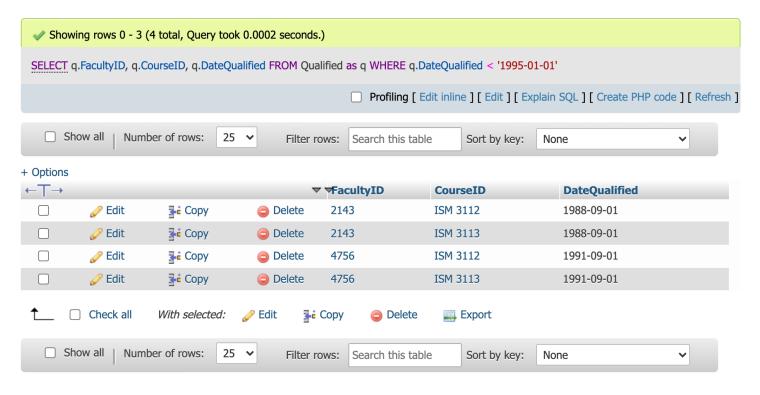
c) SELECT MAX(s.SectionNo) FROM Section as s WHERE s.Semester = 'I-2008';



b) SELECT q.FacultyID FROM Qualified as q WHERE q.CourseID = 'ISM 3113';



c) SELECT q.FacultyID, q.CourseID, q.DateQualified FROM Qualified as q WHERE q.DateQualified < '1995-01-01';



8) a) SELECT r.StudentID, COUNT(*)
FROM registration as r
WHERE r.CourseID = 'ISM 4212'
OR r.CourseID = 'ISM 4930'
GROUP BY r.StudentID
HAVING COUNT(r.SectionNo) = 2;

```
Your SQL query has been executed successfully.
 SELECT r.StudentID, COUNT(*) FROM registration as r WHERE r.CourseID = 'ISM 4212' OR r.CourseID = 'ISM 4930' GROUP BY r.StudentID
 HAVING COUNT(r.SectionNo) = 2
                                                        Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
+ Options
StudentID
                                                           COUNT(*)
54907
        b)
                SELECT q.FacultyID, COUNT(*)
                FROM Qualified as q
                WHERE q.CourseID = 'ISM 3112'
                        OR q.CourseID = 'ISM 3113'
                GROUP BY q.FacultyID
                HAVING COUNT(q.CourseID) = 1;

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

  SELECT q.FacultyID, COUNT(*) FROM Qualified as q WHERE q.CourseID = 'ISM 3112' OR q.CourseID = 'ISM 3113' GROUP BY
  q.FacultyID HAVING COUNT(q.CourseID) = 1
                                                    Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]
FacultyID
                                                          COUNT(*)
                INSERT INTO Faculty
                        VALUES
                                (1111, 'Someone');
                INSERT INTO Qualified
                        VALUES
                                (1111, 'ISM 3113', '1996-09-01');
  1 row inserted. (Query took 0.0004 seconds.)
 INSERT INTO Faculty VALUES (1111, 'Someone')
                                                                                       [ Edit inline ] [ Edit ] [ Create PHP code ]
  1 row inserted. (Query took 0.0006 seconds.)
  INSERT INTO Qualified VALUES (1111, 'ISM 3113', '1996-09-01')
                                                                                       [ Edit inline ] [ Edit ] [ Create PHP code ]
```

FROM Qualified as q
WHERE q.CourseID = 'ISM 3112'
OR q.CourseID = 'ISM 3113'
GROUP BY q.FacultyID
HAVING COUNT(q.CourseID) = 1;

Your SQL query has been executed successfully.

SELECT q.FacultyID, COUNT(*) FROM Qualified as q WHERE q.CourseID = 'ISM 3112' OR q.CourseID = 'ISM 3113' GROUP BY q.FacultyID HAVING COUNT(q.CourseID) = 1

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

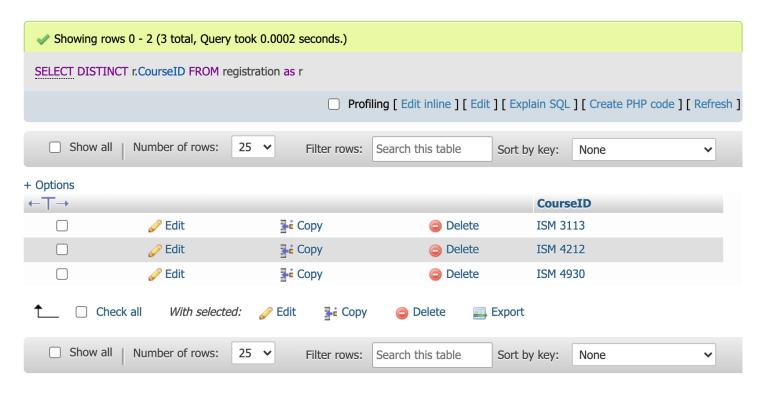
+ Options

FacultyID

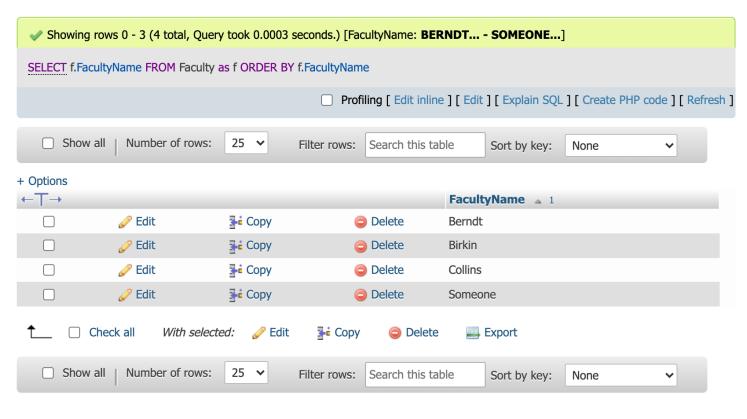
COUNT(*)

1111

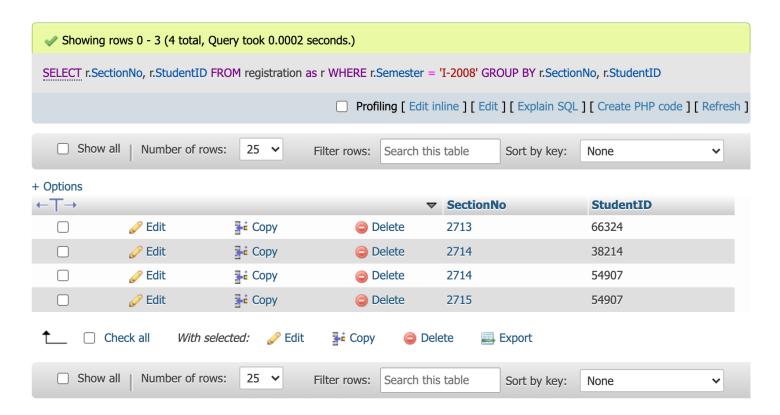
9) a) SELECT DISTINCT r.CourseID FROM registration as r;



b) SELECT f.FacultyName FROM Faculty as f ORDER BY f.FacultyName;



c) SELECT r.SectionNo, r.StudentID FROM registration as r WHERE r.Semester = 'I-2008' GROUP BY r.SectionNo, r.StudentID;



d, e) SELECT LEFT(c.CourseID, 3) AS Prefix, COUNT(*) FROM Course as c

GROUP BY Prefix;

Your SQL query has been executed successfully.	
SELECT LEFT(c.CourseID, 3) AS Prefix, COUNT(*) FROM Course as c GROUP BY Prefix	
	Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
+ Options	
Prefix	COUNT(*)
ISM	4

- 10) a) Acceptable.
 - **b)** Referential integrity constraint violated on Dnum because no corresponding primary key exists in DEPARTMENT with Dnumber = 2. A possible solution could be to insert a tuple into DEPARTMENT with Dnumber = 2, then perform the desired insertion operation.
 - c) Primary key constraint violated on Dnumber because another tuple already exists in DEPARTMENT with Dnumber = 4. A possible solution could be to choose an unused value for Dnumber, then perform the desired insertion operation.
 - **d)** Referential integrity constraint violated on Essn because no corresponding primary key exists in EMPLOYEE with Ssn = 677678989; Entity integrity constraint violated on Pno attempting to insert NULL into a composite primary key, unless an employee is allowed to work unassigned to a project. A possible solution could be to insert a tuple into EMPLOYEE with Ssn = 677678989 and refer to an existing Pnumber within PROJECT for Pno, then perform the desired insertion operation.
 - e) Acceptable.
 - f) Acceptable and deletes 4 tuples.
 - g) Referential integrity constraint violated because tuples within DEPARTMENT, WORKS_ON, and DEPENDANT reference this tuple. A possible solution could be to delete those tuples with foreign keys referencing this tuple, then perform the desired deletion operation.
 - **h)** Referential integrity constraint violated because tuples within WORKS_ON reference this tuple. A possible solution could be to delete those tuples with foreign keys referencing this tuple, then perform the desired deletion operation.
 - i) Acceptable.
 - j) Referential integrity constraint violated because tuples within WORKS_ON reference this tuple. A possible solution could be to delete those tuples with foreign keys referencing this tuple, then perform the desired deletion operation.
 - k) Acceptable.