

HW3 – ER Diagrams

COMP 353

Brian Nguyen

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.

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0136 seconds.)
CREATE TABLE Student (StudentID NUMERIC(10,0) NOT NULL, StudentName VARCHAR(25), CONSTRAINT PK_Primary_Key_StudentID)
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0128 seconds.)
CREATE TABLE Faculty (FacultyID NUMERIC(10,0) NOT NULL, FacultyName VARCHAR(25), CONSTRAINT PK_Primary_Key_FacultyID)
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0164 seconds.)
CREATE TABLE Course (CourseID NUMERIC(10,0) NOT NULL, CourseName VARCHAR(25), CONSTRAINT PK_Primary_Key_CourseID)
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0173 seconds.)
CREATE TABLE Section (SectionID NUMERIC(10,0) NOT NULL, Semester VARCHAR(2), CourseID NUMERIC(10,0) NOT NULL, CONSTRAINT Section2_PK_Primary_Key (SectionID, Semester, CourseID), CONSTRAINT Section_FK_Foreign_Key (CourseID) REFERENCES Course (CourseID))
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0178 seconds.)
CREATE TABLE Qualified (FacultyID NUMERIC(10,0) NOT NULL, CourseID NUMERIC(10,0) NOT NULL, DataQualified DATE, CONSTRAINT IQQualified_PK_Primary_Key (FacultyID, CourseID), CONSTRAINT Qualified_Faculty_FK_Foreign_Key (FacultyID) REFERENCES Faculty (FacultyID), CONSTRAINT Qualified_Course_FK_Foreign_Key (CourseID) REFERENCES Course (CourseID))
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0141 seconds.)
CREATE TABLE Registration (StudentID NUMERIC(10,0) NOT NULL, SectionID NUMERIC(10,0) NOT NULL, Semester VARCHAR(2), CourseID NUMERIC(10,0) NOT NULL, CONSTRAINT Registration_PK_Primary_Key (StudentID, SectionID, Semester), CONSTRAINT Registration_FK_Foreign_Key (SectionID, Semester, CourseID) REFERENCES Section (SectionID, Semester, CourseID))
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 4 rows inserted. (Query took 0.0061 seconds.)
INSERT INTO Student (StudentID, StudentName) VALUES (38214, 'Luturany', '1980-01-01'), (70542, 'Maana')
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 3 rows inserted. (Query took 0.0010 seconds.)
INSERT INTO Faculty (FacultyID, FacultyName) VALUES (1143, 'Baker', '1940-01-01'), (4756, 'Gallen')
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 4 rows inserted. (Query took 0.0009 seconds.)
INSERT INTO Course (CourseID, CourseName) VALUES (1001, 'DB 3117, Syst Analysis'), (1002, 'DB 3117, Syst Design'), (1003, 'DB 4212, Database'), (1004, 'DB 4930, Networking')
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 6 rows inserted. (Query took 0.0010 seconds.)
INSERT INTO Qualified (FacultyID, CourseID, DataQualified) VALUES (1143, 1001, '1980-01-01'), (1143, 1002, '1980-01-01'), (1143, 1003, '1980-01-01'), (1143, 1004, '1980-01-01'), (4756, 1001, '1980-01-01'), (4756, 1002, '1980-01-01')
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 4 rows inserted. (Query took 0.0010 seconds.)
INSERT INTO Section (SectionID, Semester, CourseID) VALUES (1, 1, 1001), (1, 1, 1002), (1, 1, 1003), (1, 1, 1004), (2, 1, 1001), (2, 1, 1002), (2, 1, 1003), (2, 1, 1004)
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 4 rows inserted. (Query took 0.0009 seconds.)
INSERT INTO Registration (StudentID, SectionID, Semester, CourseID) VALUES (38214, 1, 1, 1001), (38214, 1, 1, 1002), (38214, 1, 1, 1003), (38214, 1, 1, 1004), (70542, 2, 1, 1001), (70542, 2, 1, 1002), (70542, 2, 1, 1003), (70542, 2, 1, 1004)
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

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](#)]

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

`SELECT * FROM question2`

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

☐ Show all | Number of rows: Filter rows:

+ Options

				StudentID	StudentName
<input type="checkbox"/>	Edit	Copy	Delete	38214	Letersky
<input type="checkbox"/>	Edit	Copy	Delete	54907	Altwater
<input type="checkbox"/>	Edit	Copy	Delete	66324	Aiken

☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: Filter rows:

3) CONSTRAINT Section_FK
FOREIGN KEY (CourseID)
REFERENCES Course(CourseID));

4) a) ALTER TABLE Student
ADD GPA decimal(3, 2);

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0435 seconds.)

`ALTER TABLE Student ADD GPA decimal(3, 2)`

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

b) DROP TABLE Student;

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0082 seconds.)

`DROP TABLE Student`

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

c) ALTER TABLE Student
MODIFY COLUMN StudentName varchar(40);

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0127 seconds.)

`ALTER TABLE Student MODIFY COLUMN StudentName varchar(40)`

[[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;
```

✓ Showing rows 0 - 2 (3 total, Query took 0.0027 seconds.)

```
SELECT f.FacultyName FROM Faculty as f WHERE f.FacultyID < 4000
```

☐ 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

				FacultyName
<input type="checkbox"/>	Edit	Copy	Delete	Someone
<input type="checkbox"/>	Edit	Copy	Delete	Birkin
<input type="checkbox"/>	Edit	Copy	Delete	Berndt

☐ Check all | With selected: Edit Copy Delete Export

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

b)

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

✓ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
SELECT s.StudentName FROM Student as s WHERE s.StudentID = 66324
```

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

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

+ Options

				StudentName
<input type="checkbox"/>	Edit	Copy	Delete	Aiken

☐ Check all | With selected: Edit Copy Delete Export

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

c)

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

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

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

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

☐ Show all | Number of rows: 25 ▼ Filter rows:

+ Options

MAX(s.SectionNo)

2715

☐ Show all | Number of rows: 25 ▼ Filter rows:

7) a)

```
SELECT COUNT(*)
FROM Qualified as q
WHERE q.CourseID = 'ISM 3113';
```

Your SQL query has been executed successfully.

```
SELECT COUNT(*) FROM Qualified as q WHERE q.CourseID = 'ISM 3113'
```

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

+ Options

COUNT(*)

2

b)

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

✓ Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
SELECT q.FacultyID FROM Qualified as q WHERE q.CourseID = 'ISM 3113'
```

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

☐ Show all | Number of rows: Filter rows: Sort by key:

+ Options



FacultyID

<input type="checkbox"/>	Edit	Copy	Delete	2143
<input type="checkbox"/>	Edit	Copy	Delete	4756

☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: Filter rows: Sort by key:

c)

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

✓ Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

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

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

☐ Show all | Number of rows: Filter rows: Sort by key:

+ Options



▼ FacultyID

CourseID

DateQualified

<input type="checkbox"/>	Edit	Copy	Delete	2143	ISM 3112	1988-09-01
<input type="checkbox"/>	Edit	Copy	Delete	2143	ISM 3113	1988-09-01
<input type="checkbox"/>	Edit	Copy	Delete	4756	ISM 3112	1991-09-01
<input type="checkbox"/>	Edit	Copy	Delete	4756	ISM 3113	1991-09-01

☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: Filter rows: Sort by key:

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	2

- 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](#) ]

- c)
- ```
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;

```

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	1

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

✓ Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)

```

SELECT DISTINCT r.CourseID FROM registration as r

```

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

☐ Show all | Number of rows: Filter rows: Sort by key:

+ Options

				CourseID
<input type="checkbox"/>	Edit	Copy	Delete	ISM 3113
<input type="checkbox"/>	Edit	Copy	Delete	ISM 4212
<input type="checkbox"/>	Edit	Copy	Delete	ISM 4930

☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: Filter rows: Sort by key:

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

✓ Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.) [FacultyName: **BERNDT... - SOMEONE...**]

```
SELECT f.FacultyName FROM Faculty as f ORDER BY f.FacultyName
```

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

☐ Show all | Number of rows: Filter rows: Sort by key:

+ Options



FacultyName ▲ 1

<input type="checkbox"/>	Edit	Copy	Delete	Berndt
<input type="checkbox"/>	Edit	Copy	Delete	Birkin
<input type="checkbox"/>	Edit	Copy	Delete	Collins
<input type="checkbox"/>	Edit	Copy	Delete	Someone

↑ ☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: Filter rows: Sort by key:

- c)

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

✓ Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

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

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

☐ Show all | Number of rows: Filter rows: Sort by key:

+ Options



▼ SectionNo StudentID

<input type="checkbox"/>	Edit	Copy	Delete	2713	66324
<input type="checkbox"/>	Edit	Copy	Delete	2714	38214
<input type="checkbox"/>	Edit	Copy	Delete	2714	54907
<input type="checkbox"/>	Edit	Copy	Delete	2715	54907

↑ ☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: Filter rows: Sort by key:

- 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.