

# Integrity Constraints

## ID Domain

- any table which uses and ID as a primary key will use the ID Domain type
- a value of type ID must be a 4 digit integer
- will be added to every use of this domain as MyPHP doesn't support domain creation

```
-- ID Attribute will be a 4 digit Integer
CHECK (student_id >= 1000 AND student_id <= 9999)
```

## Table Creation

### Department

```
create table department (
    dept_name varchar(20),
    building varchar(15),
    budget numeric(10,2),
    Primary key (dept_name));
```

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0063 seconds.)

```
-- department column create table department ( dept_name varchar(20), building
varchar(15), budget numeric(10,2), Primary key (dept_name));
```

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

### course

```
create table course (
    course_id varchar(8),
    title varchar(50),
    dept_name varchar(20),
    credits numeric(2,0),
    primary key (course_id),
    foreign key (dept_name) references department (dept_name));
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0135 seconds.)

```
create table course ( course_id varchar(8), title varchar(50), dept_name varchar(20),  
credits numeric(2,0), primary key (course_id), foreign key (dept_name) references  
department (dept_name));
```

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

## instructor

```
/* instructor table  
    - ID (primary)  
    - name  
    - dept_name (foreign),  
    - salary  
  
*/  
  
CREATE TABLE instructor(  
    instructor_id varchar(8),  
    name VARCHAR(20),  
    dept_name VARCHAR(20),  
    salary NUMERIC(5,0),  
  
    PRIMARY KEY (instructor_id),  
    CHECK (instructor_id >= 1000 AND instructor_id <= 9999),  
    FOREIGN KEY (dept_name) REFERENCES department (dept_name));
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0163 seconds.)

```
CREATE TABLE instructor( instructor_id varchar(8), name VARCHAR(20), dept_name VARCHAR(20), salary NUMERIC(5,0), PRIMARY  
KEY (instructor_id), CHECK (instructor_id >= 1000 AND instructor_id <= 9999), FOREIGN KEY (dept_name) REFERENCES  
department (dept_name));
```

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

## Section

### Indexing building from department allow foreign key relationships

```
ALTER TABLE department ADD INDEX(building);
```

### Creating the table

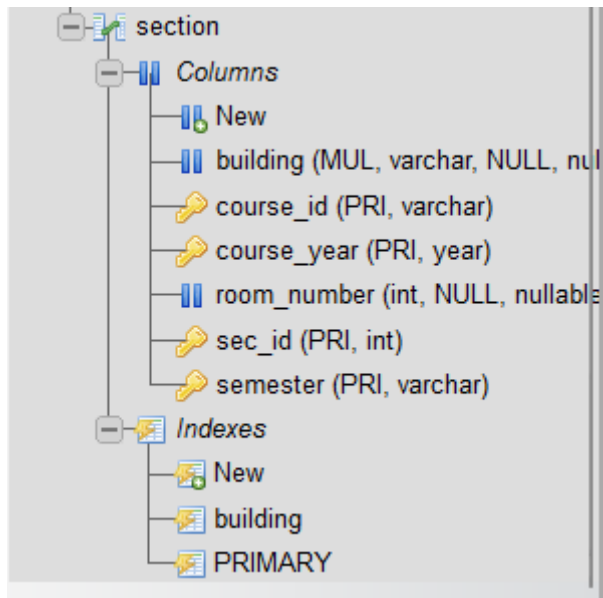
```

/*section table
    - (course_id, sec_id, semester, year) (primary key)
    - building (foreign key from department)
    - room number

constraints:
    - course_id (foreign key integrity)
    - building (foreign key integrity)
*/
CREATE TABLE section (
    course_id VARCHAR(8),
    sec_id varchar(8),
    semester VARCHAR(10),
    course_year YEAR,
    building VARCHAR(15),
    room_number INT,

PRIMARY KEY (course_id, sec_id, semester, course_year),
FOREIGN KEY (course_id) REFERENCES course (course_id),
FOREIGN KEY (building) REFERENCES department (building));

```



✓ Table section has been altered successfully.

```
ALTER TABLE `section` CHANGE `sec_id` `sec_id` INT(8) NOT NULL;
```

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

## Teaches

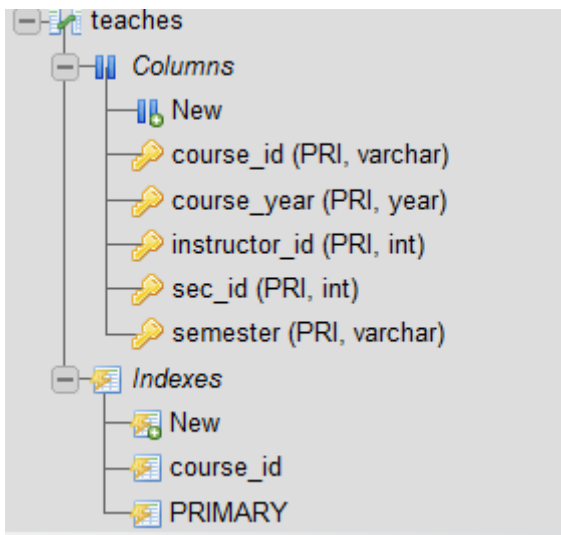
- foreign key constraint is composite as we must check for this specific combination of attributes in their order to ensure that a section exists

```

/*Teaches table
    - (instructor_id, course_id, sec_id, semester, year) (primary key)
    Constraints:
    - instructor_id (foreign key instructor)
    - course_id (foreign key course) - sec_id (foreign key section)
    - semester (foreign key section) - year (foreign key to section) */
CREATE TABLE teaches(
    instructor_id varchar(8),
    course_id VARCHAR(8),
    sec_id varchar(8),
    semester VARCHAR(10),
    course_year YEAR,

    PRIMARY KEY (instructor_id, course_id, sec_id, semester, course_year),
    FOREIGN KEY (instructor_id) REFERENCES instructor (instructor_id),
    FOREIGN KEY (course_id) REFERENCES course (course_id),
    FOREIGN KEY (course_id, sec_id, semester, course_year) REFERENCES section (course_id,
    sec_id, semester, course_year));

```



## Student

```

/* student table
    - student_id (primary key)
    - name
    - dept_name (foreign key department)
    - tot_cred
*/

```

```

CREATE TABLE student(
    student_id VARCHAR(8),
    name VARCHAR(20),
    dept_name VARCHAR(20),
    tot_cred INT,

```

```
PRIMARY KEY (student_id),  
FOREIGN KEY (dept_name) REFERENCES department (dept_name));
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0134 seconds.)

```
CREATE TABLE student( student_id VARCHAR(8), name VARCHAR(20), dept_name  
VARCHAR(20), tot_cred INT, PRIMARY KEY (student_id), FOREIGN KEY (dept_name)  
REFERENCES department (dept_name));
```

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

## Insertion Operations

### 1. Department

```
-- department insertions  
INSERT INTO department (dept_name, building, budget)  
VALUES  
( 'Computer Science', 'Woodland', 10000),  
( 'Statistics', 'Sutherland', 7500),  
( 'Chemistry', 'Woodland', 9000)  
( 'Law', 'Cloverly', 10000);
```

✓ 3 rows inserted. (Query took 0.0009 seconds.)

```
/* inserting 3 rows into each table */ -- course insertions  
INSERT INTO department (dept_name,  
building, budget) VALUES ( 'Computer Science', 'Woodland', 10000), ( 'Statistics', 'Sutherland', 7500),  
( 'Chemistry', 'Woodland', 9000);
```

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

✓ 1 row inserted. (Query took 0.0013 seconds.)

```
INSERT INTO department (dept_name, building, budget) VALUES ( 'Law',  
'Cloverly', 10000);
```

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

### 2. Course

```
-- course insertions  
INSERT INTO course (course_id, title, dept_name, credits)  
VALUES  
( 'CS101', 'Intro to Computer Science', 'Computer Science', 3),  
( 'STAT101', 'Intro to Statistics', 'Statistics', 3),  
( 'CHEM101', 'General Chemistry', 'Chemistry', 4);
```

✓ 3 rows inserted. (Query took 0.0008 seconds.)

```
-- course insertions INSERT INTO course (course_id, title, dept_name, credits) VALUES ('CS101', 'Intro
to Computer Science', 'Computer Science', 3), ('STAT101', 'Intro to Statistics', 'Statistics', 3),
('CHEM101', 'General Chemistry', 'Chemistry', 4);
```

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

### 3. Instructor

```
INSERT INTO instructor (instructor_id, name, dept_name, salary)
VALUES
('1234', 'Ishti Hussain', 'Computer Science', 80000),
('5678', 'Wei Gao', 'Statistics', 85000),
('9012', 'Prof Heisenberg', 'Chemistry', 90000);
```

✓ 3 rows inserted. (Query took 0.0012 seconds.)

```
INSERT INTO instructor (instructor_id, name, dept_name, salary) VALUES
('1234', 'Ishti Hussain', 'Computer Science', 80000), ('5678', 'Wei Gao',
'Statistics', 85000), ('9012', 'Prof Heisenberg', 'Chemistry', 90000);
```

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

### 4. Section

```
INSERT INTO section (course_id, sec_id, semester, course_year, building, room_number)
VALUES
('CHEM101', 'S001', 'Fall', 2023, 'Woodland', 101),
('CS101', 'S002', 'Spring', 2023, 'Woodland', 202),
('STAT101', 'S003', 'Fall', 2023, 'Sutherland', 103);
```

✓ 3 rows inserted. (Query took 0.0012 seconds.)

```
INSERT INTO instructor (instructor_id, name, dept_name, salary) VALUES
('1234', 'Ishti Hussain', 'Computer Science', 80000), ('5678', 'Wei Gao',
'Statistics', 85000), ('9012', 'Prof Heisenberg', 'Chemistry', 90000);
```

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

### 5. Student

```
INSERT INTO student (student_id, name, dept_name, tot_cred)
VALUES
('S1000001', 'Emily Watson', 'Computer Science', 40),
```

```
('S1000002', 'Michael Brown', 'Chemistry', 38),
('S1000003', 'Sophia Johnson', 'Statistics', 42);
```

✓ 3 rows inserted. (Query took 0.0012 seconds.)

```
INSERT INTO instructor (instructor_id, name, dept_name, salary) VALUES
('1234', 'Ishti Hussain', 'Computer Science', 80000), ('5678', 'Wei Gao',
'Statistics', 85000), ('9012', 'Prof Heisenberg', 'Chemistry', 90000);
```

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

## 6. Instructor

```
INSERT INTO teaches (instructor_id, course_id, sec_id, semester, course_year)
VALUES
('1234', 'CHEM101', 'S001', 'Fall', 2023),
('5678', 'CS101', 'S002', 'Spring', 2023),
('9012', 'STAT101', 'S003', 'Fall', 2023);
```

✓ 3 rows inserted. (Query took 0.0013 seconds.)

```
INSERT INTO teaches (instructor_id, course_id, sec_id, semester,
course_year) VALUES ('1234', 'CHEM101', 'S001', 'Fall', 2023), ('5678',
'CS101', 'S002', 'Spring', 2023), ('9012', 'STAT101', 'S003', 'Fall', 2023);
```

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

## Data Manipulation

### (a) Increase the budget of law department by 5 percent

```
UPDATE department
Set budget = 1.05 * budget
WHERE dept_name = 'Law';
```

✓ 1 row affected. (Query took 0.0009 seconds.)

```
UPDATE department Set budget = 1.05 * budget WHERE dept_name = 'Law';
```

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

dept_name	building	budget
Law	Cloverly	10500.00


**(b) Show the budget of CSC department.**

```
SELECT budget from department
WHERE dept_name = 'Computer Science'
```

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

```
SELECT budget from department WHERE dept_name = 'Computer Science';
```

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

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

### Extra options

←T→ budget  
□ Edit Copy Delete 10000.00

**(c) List name and salary of all instructors of CSC department.**

```
SELECT name, salary from instructor
WHERE dept_name = 'Computer Science';
```





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

```
SELECT name, salary from instructor WHERE dept_name = 'Computer Science';
```

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

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

### Extra options

		name	salary
	 Edit  Copy  Delete	Ishti Hussain	80000

**(d) List name and total credits of all students.**

```
SELECT name, tot_cred FROM student;
```












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

```
SELECT name, tot_cred FROM student;
```

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

☐ Show all | Number of rows:  Filter rows:

Extra options

				name	tot_cred
<input type="checkbox"/>	 Edit	 Copy	 Delete	Mark Zuk	40
<input type="checkbox"/>	 Edit	 Copy	 Delete	Michael Brown	38
<input type="checkbox"/>	 Edit	 Copy	 Delete	Sophia Johnson	42