

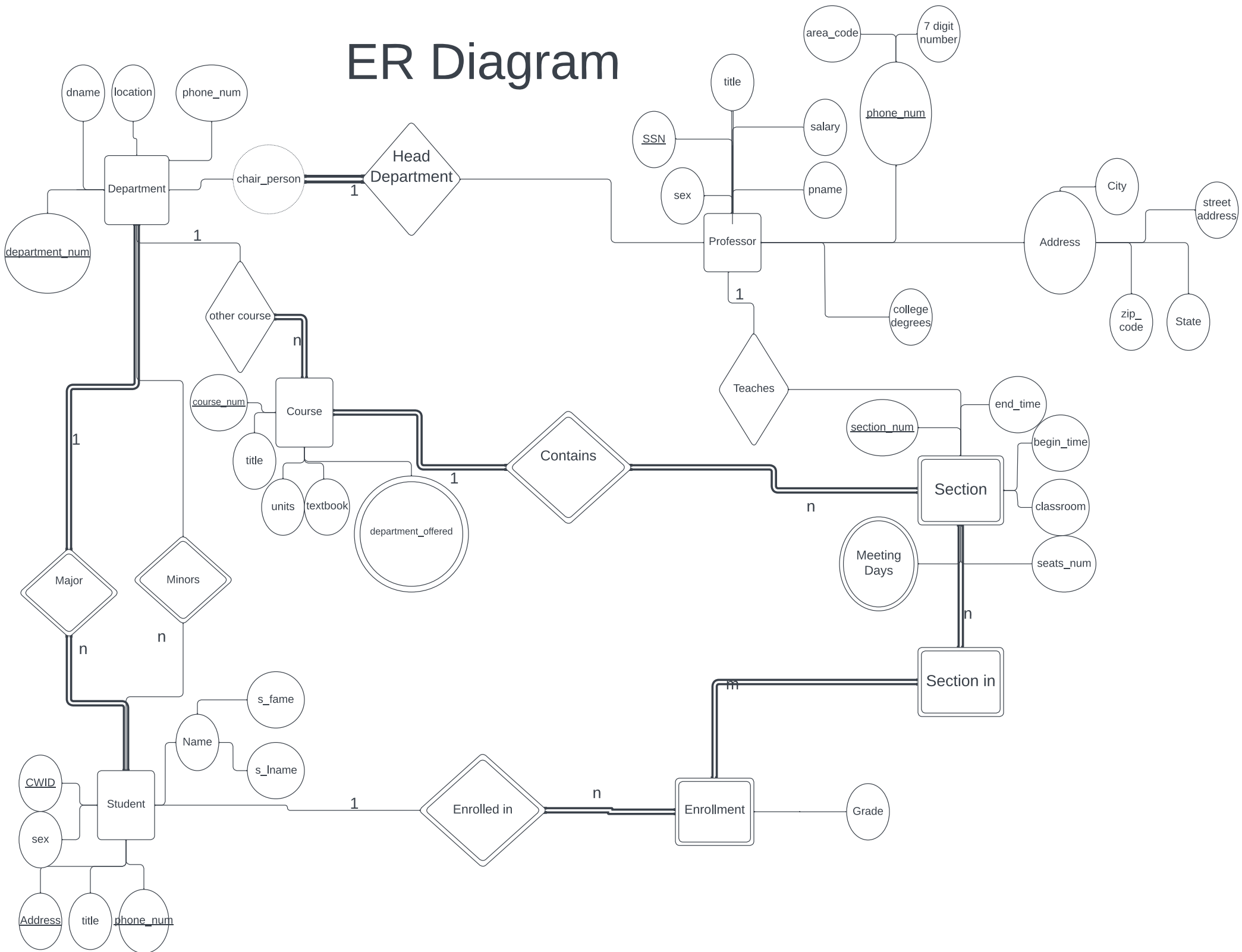
CPSC332 File Structures & Databases Systems

Brandon Nguyen (8942)

Tung Nguyen (5034)

Demo the project here:
<http://ecs.fullerton.edu/~cs332g20/>

ER Diagram



Relational Model

Professor

<u>SSN(PK)</u>	pname	sex	Area_code	zip_code	state	city	Street_address	salary	phone_num	college_degrees	title
----------------	-------	-----	-----------	----------	-------	------	----------------	--------	-----------	-----------------	-------

Department

<u>department_num(PK)</u>	dname	phone_num	location	<u>Chair_person(FK)</u>
---------------------------	-------	-----------	----------	-------------------------

Course

<u>course_num(PK)</u>	units	Title	textbook	<u>department_offered(FK)</u>
-----------------------	-------	-------	----------	-------------------------------

Section

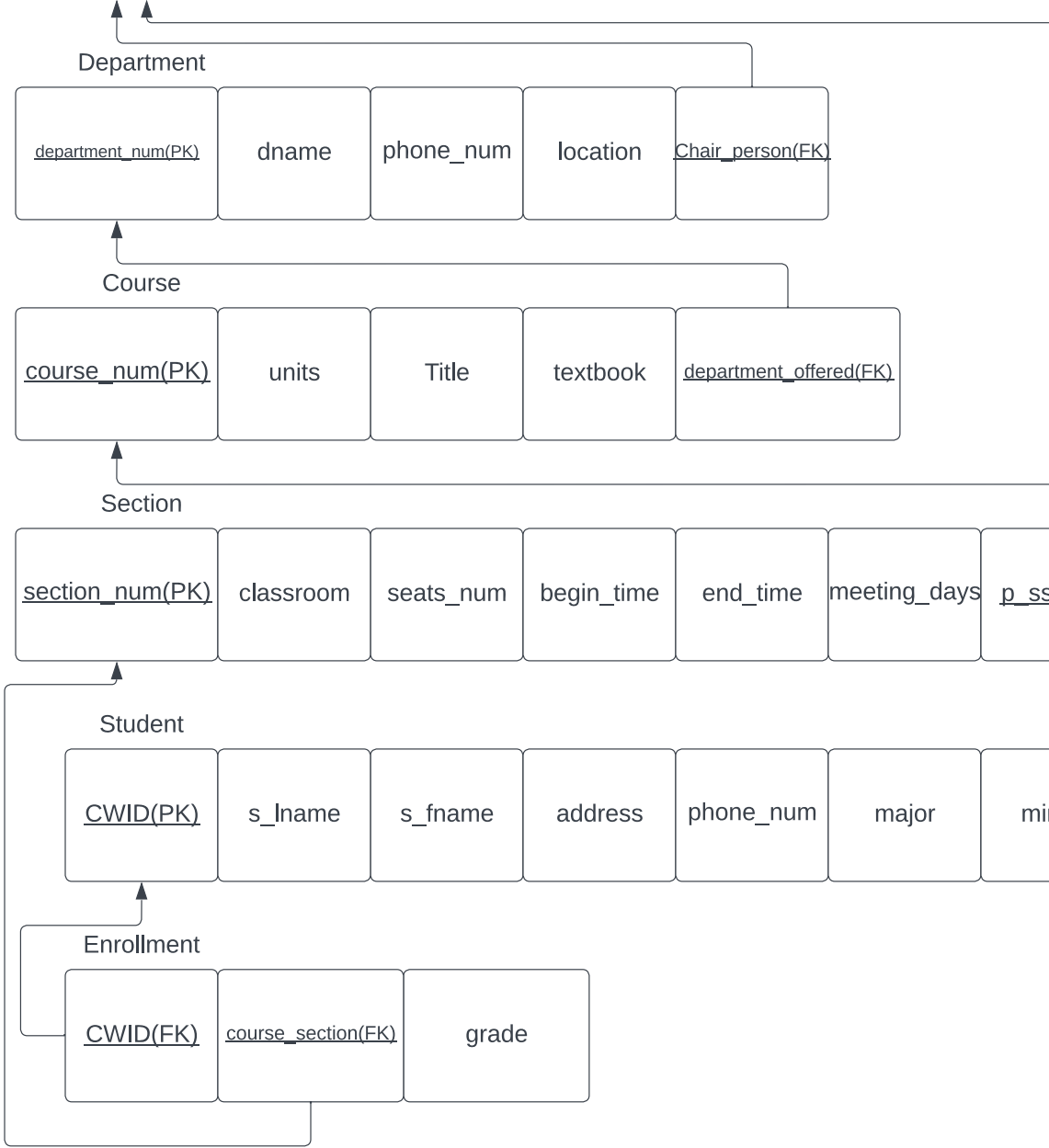
<u>section_num(PK)</u>	classroom	seats_num	begin_time	end_time	meeting_days	<u>p_ssn(FK)</u>	<u>course(FK)</u>
------------------------	-----------	-----------	------------	----------	--------------	------------------	-------------------

Student

<u>CWID(PK)</u>	s_lname	s_fname	address	phone_num	major	minors
-----------------	---------	---------	---------	-----------	-------	--------

Enrollment

<u>CWID(FK)</u>	<u>course_section(FK)</u>	grade
-----------------	---------------------------	-------



Source Code (SQL)

table.sql

The default database is named: cs332g20

```
CREATE TABLE professors(  
    ssn INT NOT NULL,  
    pname VARCHAR(255),  
    street_address VARCHAR(255),  
    city VARCHAR(255),  
    state VARCHAR(255),  
    zip_code INT,  
    area_code INT,  
    phone_num VARCHAR(255),  
    sex VARCHAR(255),  
    title VARCHAR(255),  
    salary INT,  
    college_degrees VARCHAR(255),  
    PRIMARY KEY(ssn)  
);  
  
CREATE TABLE departments(  
    department_num INT NOT NULL,  
    dname VARCHAR(255),  
    phone_num VARCHAR(255),  
    location VARCHAR(255),  
    chair_person INT,  
    FOREIGN KEY(chair_person) REFERENCES professors(ssn),  
    PRIMARY KEY(department_num)  
);  
  
CREATE TABLE courses(  
    course_num INT NOT NULL,  
    title VARCHAR(255),  
    textbook VARCHAR(255),  
    units INT,  
    department_offered INT,
```

```
FOREIGN KEY(department_offered) REFERENCES departments(department_num),  
PRIMARY KEY(course_num)  
);
```

```
CREATE TABLE sections(  
    section_num INT NOT NULL,  
    classroom VARCHAR(255),  
    seats_num INT,  
    meeting_days VARCHAR(255),  
    begin_time VARCHAR(255),  
    end_time VARCHAR(255),  
    course INT,  
    p_ssn INT,  
  
    PRIMARY KEY(section_num),  
    FOREIGN KEY (p_ssn) REFERENCES professors(ssn),  
    FOREIGN KEY (course) REFERENCES courses(course_num)  
);
```

```
CREATE TABLE students(  
    cwid INT NOT NULL,  
    s_fname VARCHAR(255),  
    s_lname VARCHAR(255),  
    address VARCHAR(255),  
    phone_num VARCHAR(255),  
    major VARCHAR(255),  
    minors VARCHAR(255),  
  
    PRIMARY KEY(cwid)  
);
```

```
CREATE TABLE enrollments(  
    student_id INT NOT NULL,  
    course_section INT NOT NULL,  
    grade VARCHAR(255),  
  
    FOREIGN KEY(student_id) REFERENCES students(cwid),  
    FOREIGN KEY(course_section) REFERENCES sections(section_num)  
);
```

Source Code (SQL) continued...

drop.sql

This script will drop all tables associated with the default database: c332g20

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
SET @tables = NULL;
```

```
SET GROUP_CONCAT_MAX_LEN=32767;
```

```
SELECT GROUP_CONCAT("'", TABLE_SCHEMA, '`.`', TABLE_NAME, "'") INTO @tables
```

```
FROM information_schema.TABLES
```

```
WHERE TABLE_SCHEMA = (SELECT DATABASE());
```

```
SELECT IFNULL(@tables, "") INTO @tables;
```

```
SET @tables = CONCAT('DROP TABLE IF EXISTS ', @tables);
```

```
PREPARE stmt FROM @tables;
```

```
EXECUTE stmt;
```

```
DEALLOCATE PREPARE stmt;
```

```
SET FOREIGN_KEY_CHECKS = 1;
```

```
SET @tables = NULL;
```

Source Code (SQL) continued...

data.sql

INSERT some data that includes at least 8 students, 2 departments, 3 professors, 4 courses, 6 sections, and 20 enrollments.

-- Path: homepage\sql\data.sql

```
INSERT INTO professors(ssn, pname, street_address, city, state, zip_code, area_code,
phone_num, sex, title, salary, college_degrees) VALUES
(123456789, 'John Smith', '1234 Main St', 'Fullerton', 'California', 12345, 123, '4567890', 'M',
'Professor', 100000, 'PhD'),
(234567890, 'Jane Doe', '2345 Main St', 'Westminster', 'California', 23456, 234, '5678901', 'F',
'Professor', 120000, 'PhD'),
(345678901, 'John Doe', '3456 Main St', 'Santa Ana', 'California', 34567, 345, '6789012', 'M',
'Professor', 210000, 'PhD');
```

```
INSERT INTO departments(department_num, dname, phone_num, location, chair_person)
VALUES
(1, 'Computer Science', '1234567890', '1234 Main St', 123456789),
(2, 'Mathematics', '2345678901', '2345 Main St', 234567890);
```

```
INSERT INTO courses(course_num, title, textbook, units, department_offered) VALUES
(1, 'Intro to Computer Science', 'Intro to Computer Science', 4, 1),
(2, 'Advanced Computer Science', 'Advanced Computer Science', 4, 1),
(3, 'Intro to Mathematics', 'Intro to Mathematics', 4, 2),
(4, 'Advanced Mathematics', 'Advanced Mathematics', 4, 2);
```

```
INSERT INTO sections(section_num, classroom, seats_num, meeting_days, begin_time,
end_time, course, p_ssn) VALUES
(1, '167B', 35, 'MWF', '9:00am', '10:00am', 1, 123456789),
(2, '367A', 42, 'MW', '10:00am', '11:00am', 1, 123456789),
(3, '376C', 36, 'TTh', '11:00am', '12:00pm', 2, 234567890),
(4, '326A', 22, 'Fri', '12:00pm', '1:00pm', 2, 234567890),
(5, '124D', 34, 'Sat', '1:00pm', '2:00pm', 3, 345678901),
(6, '531F', 37, 'MWF', '2:00pm', '3:00pm', 4, 345678901);
```

```
INSERT INTO students(cwid, s_fname, s_lname, address, phone_num, major, minors) VALUES
```

(1, 'John', 'Smith', '1234 Main St', '1234567890', 'Mathematics', 'Computer Science'),
(2, 'Jane', 'Doe', '2345 Main St', '2345678901', 'Computer Science', ''),
(3, 'John', 'Doe', '3456 Main St', '3456789012', 'Mathematics', ''),
(4, 'Jane', 'Smith', '4567 Main St', '4567890123', 'Computer Science', 'Physics'),
(5, 'John', 'Doe', '5678 Main St', '5678901234', 'Mathematics', ''),
(6, 'Jane', 'Doe', '6789 Main St', '6789012345', 'Computer Science', 'Electrical Engineering,
Biology'),
(7, 'John', 'Smith', '7890 Main St', '7890123456', 'Mathematics', ''),
(8, 'Jane', 'Doe', '8901 Main St', '8901234567', 'Computer Science', 'Electrical Engineering');

INSERT INTO enrollments(student_id, course_section, grade) VALUES

(1, 3, 'A+'),
(1, 6, 'B-'),
(1, 5, 'C'),
(2, 1, 'D-'),
(2, 3, 'F'),
(2, 5, 'A'),
(3, 2, 'B'),
(3, 5, 'C+'),
(3, 6, 'D+'),
(4, 4, 'F'),
(4, 3, 'A'),
(4, 1, 'B-'),
(5, 5, 'C'),
(5, 3, 'D-'),
(5, 6, 'F'),
(6, 2, 'A'),
(6, 3, 'B+'),
(6, 5, 'C'),
(7, 6, 'D-'),
(7, 3, 'F'),
(7, 2, 'A+'),
(8, 4, 'B'),
(8, 5, 'C-'),
(8, 6, 'F');

Source Code (DB)

```
# This script will drop old databases/tables and re-create them in the appropriate order
# This script requires an option file to be set (see README), for provision of password (not over
command line)
```

```
# Helper Functions -----
```

```
# For logging:
```

```
timestamp() {
    date +"%T"
}
```

```
# Constants-----
```

```
# SQL script dir
```

```
location="/home/titan0/cs332g/cs332g20/homepage/sql"
```

```
# Ordered Array:
```

```
scripts=("$location/drop.sql"
```

```
"$location/tables.sql"
```

```
"$location/data.sql")
```

```
# Default database:
```

```
schema="cs332g20"
```

```
echo "Now running SQL scripts"
```

```
for val in ${scripts[@]}; do
```

```
    timestamp >> log.txt
```

```
    echo "    Running $val" >> log.txt
```

```
    sshpass -f './password' ssh cs332g20@shell.ecs.fullerton.edu "mysql -h mariadb $schema <
$val" >> log.txt
```

```
done
```

```
timestamp >> log.txt
```

```
echo -e "    Sequence completed! \n----- \n" >> log.txt
```

```
echo "" >> log.txt
```

```
echo "Sequence completed! Check log.txt for details."
```

Source Code (PHP & HTML)

41 lines (39 sloc) | 1.55 KB

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <!-- Add styles css to html -->
4      <head>
5          <meta charset="UTF-8">
6          <meta name="viewport" content="width=device-width, initial-scale=1.0">
7          <title>CPSC 332 Project</title>
8          <link rel="stylesheet" href="css/styles.css">
9      </head>
10     <body>
11         <!-- Inputs needed for Professors -->
12         <h2 text="#ffffff">
13             Professor Portal
14         </h2>
15         <form action="php_files/ssn_prof_request.php">
16             <input type="text" id="ssn" name="ssn" placeholder="Social Security" required><br>
17             <input type="submit" value="Query">
18         </form>
19         <br>
20         <form action="php_files/cn_sn_prof_request.php">
21             <input type="text" id="cnum" name="cnum" placeholder="Course Number" required><br>
22             <input type="text" id="snum" name="snum" placeholder="Section Number" required><br>
23             <input type="submit" value="Query">
24         </form>
25
26         <!-- Inputs needed for Students -->
27         <h2 id="student-portal" text="#ffffff">
28             Student Portal
29         </h2>
30         <form action="php_files/cnum_student_request.php">
31             <input type="text" id="cnum" name="cnum" placeholder="Course Number" required><br>
32             <input type="submit" value="Query">
33         </form>
34         <br>
35         <form action="php_files/cwid_student_request.php">
36             <input type="text" id="cwid" name="cwid" placeholder="CWID" required><br>
37             <input type="submit" value="Query">
38         </form>
39
40     </body>
41 </html>
```

47 lines (47 sloc) 1.84 KB

Raw

```
1 <html>
2 <body>
3 Your requested information for Course Number and Section Number: <?php echo $_GET["cnum"]; ?>, <?php echo $_GET["snum"]; ?>
4 <br>
5 <br>
6 <?php
7 $link = mysqli_connect('mariadb', 'cs332g20', '8CTKTqTV', 'cs332g20');
8 if (!$link) {
9     echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
10    exit();
11 }
12 // Given a course number and a section number, count how many students get each distinct grade, 'A+', 'B' for each course.
13 $query = "SELECT grade, COUNT(grade) AS grade_count FROM enrollments WHERE course_section = " . $_GET["snum"] . " GROUP BY grade;";
14 $result = mysqli_query($link, $query);
15 if (!$result) {
16     echo "Failed to run query: " . $mysqli -> error;
17     exit();
18 }
19 echo "Course Number: " . $_GET["cnum"] . "<br>";
20 echo "Section Number: " . $_GET["snum"] . "<br>";
21 echo "Grade Count for each grade: <br>";
22 echo "-----<br>";
23 $grades = array(
24     'A+' => 0,
25     'A' => 0,
26     'A-' => 0,
27     'B+' => 0,
28     'B' => 0,
29     'B-' => 0,
30     'C+' => 0,
31     'C' => 0,
32     'C-' => 0,
33     'D+' => 0,
34     'D' => 0,
35     'D-' => 0,
36     'F' => 0,
37 );
38 // Display every possible letter grade and the number of students who got that grade
39 while ($row = mysqli_fetch_assoc($result)) {
40     $grades[$row["grade"]] = $row["grade_count"];
41 }
42 foreach ($grades as $key => $value) {
43     echo $key . ": " . $value . "<br>";
44 }
45 ?>
46 </body>
47 </html>
```

51 lines (48 sloc) | 2.34 KB

```
1  <html>
2      <body>
3          Your requested information for Course Number: <?php echo $_GET["cnum"]; ?>
4          <br>
5          <br>
6          <?php
7              $link = mysqli_connect('mariadb', 'cs332g20', '8CTKTqTV', 'cs332g20');
8              if (!$link) {
9                  echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
10                 exit();
11             }
12             // Given the course number list the sections of the course, including the classrooms, the
13             $query = "SELECT DISTINCT section_num, classroom, meeting_days, begin_time, end_time
14                 FROM sections NATURAL JOIN enrollments
15                 WHERE course = " . $_GET["cnum"] . "
16                 GROUP BY section_num, classroom, begin_time, end_time, meeting_days
17                 ORDER BY section_num;";
18
19             $result = mysqli_query($link, $query);
20             if (!$result) {
21                 echo "Failed to run query: " . $mysqli -> error;
22                 exit();
23             }
24             echo "Course Number: " . $_GET["cnum"] . "<br>";
25             while ($row = mysqli_fetch_assoc($result)) {
26                 echo "-----<br>";
27                 echo "Section Number: " . $row["section_num"] . "<br>";
28
29                 // Get the number of students enrolled in the section
30                 $query2 = "SELECT COUNT(student_id) AS enrolled_students
31                     FROM enrollments
32                     WHERE course_section = " . $row["section_num"] . ";";
33                 $result2 = mysqli_query($link, $query2);
34                 if (!$result2) {
35                     echo "Failed to run query that grabs student count: " . $mysqli -> error;
36                     exit();
37                 }
38
39                 $row2 = mysqli_fetch_assoc($result2);
40                 echo "Classroom: " . $row["classroom"] . "<br>";
41                 echo "Meeting Days: " . $row["meeting_days"] . "<br>";
42                 echo "Meeting Time: " . $row["begin_time"] . " - " . $row["end_time"] . "<br>";
43                 echo "Enrolled Students: " . $row2["enrolled_students"] . "<br>";
44                 echo "<br>";
45                 mysqli_free_result($result2);
46             }
47             mysqli_free_result($result);
48             $link->close();
49         ?>
50     </body>
51 </html>
```

35 lines (34 sloc) | 1.54 KB

```
1  <html>
2      <body>
3          Your requested information for Social Security Number: <?php echo $_GET["ssn"]; ?>
4      <br>
5      <br>
6      <?php
7          $link = mysqli_connect('mariadb', 'cs332g20', '8CTKTqTV', 'cs332g20');
8          if (!$link) {
9              echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
10             exit();
11         }
12         // Given the ssn of a professor, list the titles, classrooms, meeting days and time of his/her
13         $query = "SELECT courses.title, classroom, meeting_days, begin_time, end_time
14                 FROM (professors JOIN sections ON ssn = p_ssn) JOIN courses ON course_num = course
15                 WHERE ssn = " . $_GET["ssn"] . ";";
16
17         $result = mysqli_query($link, $query);
18         if (!$result) {
19             echo "Failed to run query: " . $mysqli -> error;
20             exit();
21         }
22         echo "---Professor with SSN: " . $_GET["ssn"] . "'s Classes---<br>";
23         while ($row = mysqli_fetch_assoc($result)) {
24             echo "-----<br>";
25             echo "Course: " . $row["title"] . "<br>";
26             echo "Classroom: " . $row["classroom"] . "<br>";
27             echo "Meeting Days: " . $row["meeting_days"] . "<br>";
28             echo "Meeting Time: " . $row["begin_time"] . " - " . $row["end_time"] . "<br>";
29             echo "<br>";
30         }
31         mysqli_free_result($result);
32         $link->close();
33     ?>
34 </body>
35 </html>
```

CSS changes :

```
body {
    background: black;
    background-size: cover;
    color: white;
    text-align: center;
}
```

33 lines (33 sloc) 1.4 KB

Raw Blame

```
1  <html>
2  <body>
3      Your requested information for CWID: <?php echo $_GET["cwid"]; ?>
4      <br>
5      <br>
6      <?php
7          $link = mysqli_connect('mariadb', 'cs332g20', '8CTKTqTV', 'cs332g20');
8          if (!$link) {
9              echo "Failed to connect to MySQL: " . mysqli_connect_error();
10             exit();
11         }
12         // Given the CWID of a student, list all courses the student took and their grades for that course.
13         // Grab the course name given the section number
14         $query = "SELECT course, grade, title FROM (enrollments JOIN sections ON course_section = section_num) JOIN courses ON course = course_num
15             WHERE student_id = " . $_GET["cwid"] . " ";
16         $result = mysqli_query($link, $query);
17         if (!$result) {
18             echo "Failed to run query: " . mysqli_error();
19             exit();
20         }
21         echo "CWID: " . $_GET["cwid"] . "<br>";
22         while ($row = mysqli_fetch_assoc($result)) {
23             echo "-----<br>";
24             echo "Course Number: " . $row["course"] . "<br>";
25             echo "Course: " . $row["title"] . "<br>";
26             echo "Grade: " . $row["grade"] . "<br>";
27             echo "<br>";
28         }
29         mysqli_free_result($result);
30         $link->close();
31     ?>
32 </body>
33 </html>
```

Website Photos

A screenshot of a web browser showing the 'Professor Portal' login page. The page has a dark background with white text. At the top, it says 'Professor Portal'. Below this, there are two sections: 'Social Security' and 'Course Number'. Each section has a text input field and a 'Query' button. The 'Social Security' section is currently active, and the 'Course Number' section is below it. There is also a 'Student Portal' section further down.

← → ↻ ⚠ Not secure | ecs.fullerton.edu/~cs332g20/php_files/ssn_prof_request.php?ssn=123456789

Your requested information for Social Security Number: 123456789

---Professor with SSN: 123456789's Classes---

Course: Intro to Computer Science
Classroom: 167B
Meeting Days: MWF
Meeting Time: 9:00am - 10:00am

Course: Intro to Computer Science
Classroom: 367A
Meeting Days: MW
Meeting Time: 10:00am - 11:00am

← → ↻ ⚠ Not secure | ecs.fullerton.edu/~cs332g20/php_files/cnum_student_request.php?cnum=1

Your requested information for Course Number: 1

Course Number: 1

Section Number: 1
Classroom: 167B
Meeting Days: MWF
Meeting Time: 9:00am - 10:00am
Enrolled Students: 2

Section Number: 2
Classroom: 367A
Meeting Days: MW
Meeting Time: 10:00am - 11:00am
Enrolled Students: 3

← → ↻ ⚠ Not secure | ecs.fullerton.edu/~cs332g20/php_files/cwid_student_re

Your requested information for CWID: 1

CWID: 1

Course Number: 2
Course: Advanced Computer Science
Grade: A+

Course Number: 3
Course: Intro to Mathematics
Grade: C

Course Number: 4
Course: Advanced Mathematics
Grade: B-

A screenshot of the 'Professor Portal' login page, similar to the one above. It shows the 'Social Security' and 'Course Number' sections. The 'Course Number' section has a text input field with the value '1' and a 'Query' button. A yellow warning box with an exclamation mark is overlaid on the 'Section Number' field, which is currently empty. The warning message says 'Please fill out this field.' Below the warning box, the 'Student Portal' section is visible, with 'Course Number' and 'CWID' input fields and 'Query' buttons.