**USE CASES/FUNCTIONALITIES WITH THEIR QUERIES**

**Non-Logged in:**

* **Login Student**
  + To check if Student login info exists for login authorization and sessions
    - (**Search for Student)**

SELECT Student\_ID, Student\_Password FROM Student WHERE Student\_ID = %s and Student\_Password = %s

* **Login Instructor**
  + To check if Instructor login info exists for login authorization and sessions
    - (**Search for Instructor)**

SELECT Instructor\_ID, Instructor\_Password FROM Instructor WHERE Instructor\_ID = %s and Instructor\_Password = %s

* **Login Admin**
  + To check if Admin login info exists for login authorization and sessions
    - (**Search for Admin)**

SELECT Admin\_ID, Admin\_Password FROM Instructor WHERE Admin\_ID = %s and Admin\_Password = %s

* **Student Registration**
  + To insert a student information into the system and create a profile/account
    - (**Insert Student Into Database)**

INSERT INTO Student (Student\_ID, Student\_Password, Student\_Fname, Student\_Lname, Major, Credits\_Taken) VALUES (%s, %s,%s, %s, %s, %s)

* **Non-logged course search by course name**
  + To fetch and display available courses and sections searched by course name
    - (**Fetch Courses with specific name)**

SELECT \* FROM Course WHERE Course\_Name=%s

* + - **(Fetch Sections of that course)**

SELECT \* FROM Section WHERE Course\_Name=%s

* **Non-logged course search by department name**
  + To fetch and display available courses and sections under a specific department
    - (**Fetch Courses in department)**

SELECT \* FROM Course WHERE Department\_Name=%s

* + - **(Fetch Sections of those courses)**

Select \* FROM Section Natural Join Course WHERE Department\_Name=%s

* **Non-logged course search by instructor first name**
  + To fetch and display available courses and sections under a specific instructor
    - (**Fetch Courses based on instructor)**

SELECT \* FROM Course Natural Join Section Natural Join Teaches Natural Join Instructor WHERE Instructor\_Fname=%s

* + - **(Fetch Sections of those courses)**

Select \* FROM Teaches Natural Join Section Natural Join Instructor WHERE Instructor\_Fname=%s

**Student Logged in:**

* **Homepage student information**
  + To fetch and display all information of the logged in student
    - (**Fetch student info)**

Select \* from Student where Student\_ID=%s

* **Student Views his own courses**
  + To fetch and display all courses a student is taking
    - (**Fetch student courses)**

SELECT \* FROM Takes Natural Join Course WHERE Student\_ID=%s

* **Student provides ratings for their own courses**
  + To rate one of their courses
    - (**Update Course Rating)**

UPDATE Takes SET Rating = %s WHERE (Student\_ID = %s) and (Section\_ID = %s) and (Course\_ID = %s)

* **Student course search by course name**
  + To fetch and display available sections searched by course name
    - (**Fetch Sections and info with specific name and available capacity)**

SELECT \* FROM Section Natural Join Instructor Natural Join Teaches Natural Join Course Natural Join coursesectioncapacitydiff where (Course\_Name = %s) and (remaining\_spots > 0)

* **Student course search by department name**
  + To fetch and display available sections under a specific department
    - (**Fetch Sections and info in department and available capacity)**

SELECT \* FROM Section Natural Join Course Natural Join Instructor Natural Join Teaches Natural Join coursesectioncapacitydiff where (Department\_Name = %s) and (remaining\_spots > 0)

* **Student course search by instructor first name**
  + To fetch and display available sections under a specific instructor
    - (**Fetch Sections and info based on instructor and available capacity)**

SELECT \* FROM Section Natural Join Teaches Natural Join Instructor Natural Join coursesectioncapacitydiff where (Instructor\_Fname = %s) and (remaining\_spots > 0)

* **Student selects and registers for a course**
  + To insert a student into a section (Student is now taking the course)
    - (**Insert Student into section)**

INSERT INTO Takes (Course\_ID, Student\_ID, Section\_ID) VALUES (%s, %s,%s)

* + To fetch the number of credits in that course and then update the student’s credits taken
    - (**Fetch Credits of course)**

Select Course\_Credits from Course Where Course\_ID = %s"

* + - **(Update student’s taken credits)**

Update Student SET Credits\_Taken = Credits\_Taken + %s

**Instructor Logged in:**

* **Homepage Instructor information**
  + To fetch and display all information of the logged in Instructor
    - **(Fetch Instructor information)**

Select \* from Instructor where Instructor\_ID=%s

* **Instructor Views courses they are teaching**
  + To fetch and display all courses an instructor is teaching
    - (**Fetch instructor courses)**

SELECT \* FROM Teaches WHERE Instructor\_ID=%s

* **Instructor Views students they are teaching in a selected course**
  + To fetch and display all students in a specific course
    - (**Fetch students in the course)**

Select \* from (Takes Natural Join Student Natural Join Teaches) where (Course\_ID = %s) and (Section\_ID = %s) and (Instructor\_ID = %s)

* **Instructor assigns grades to students they are teaching in a selected course**
  + To assign/update a student’s grade
    - (**Insert/Update student’s grade)**

UPDATE Takes SET Grade = %s WHERE (Student\_ID = %s) and (Section\_ID = %s) and (Course\_ID = %s)

* **Instructor creates a course and section simultaneously**
  + To create a course and a section
    - (**Insert course)**

INSERT INTO Course (Course\_ID, Course\_Name, Course\_Credits, Course\_Description, Department\_Name ) VALUES (%s, %s,%s, %s, %s)

* + - **(Insert Section)**

INSERT INTO Section (Section\_ID, Course\_ID, Course\_Name, Semester, Section\_Day, Section\_Time, Student\_Limit ) VALUES (%s, %s,%s, %s, %s, %s, %s)

* **Instructor creates a section for an existing course**
  + To check if course exists
    - **(Check if course exists)**

Select \* FROM Course Where Course\_ID = %s

* + To create a section for existing course
    - (**Insert section)**

INSERT INTO Section (Section\_ID, Course\_ID, Course\_Name, Semester, Section\_Day, Section\_Time, Student\_Limit ) VALUES (%s, %s,%s, %s, %s, %s, %s)

* **Instructor views the proposed course/section choices**
  + To display choices of sections
    - **(Fetch choice information)**

SELECT \* FROM Choice where Instructor\_ID = %s

* **Instructor chooses a section from the options provided**
  + To create the section and course for the section chosen by the instructor and assign them to that section, and then delete the choices from the database
    - (**Insert course)**

INSERT INTO Course (Course\_ID, Course\_Name, Course\_Credits, Course\_Description, Department\_Name ) VALUES (%s, %s,%s, %s, %s)

* + - **(Insert Section)**

INSERT INTO Section (Section\_ID, Course\_ID, Course\_Name, Semester, Section\_Day, Section\_Time, Student\_Limit ) VALUES (%s, %s,%s, %s, %s, %s, %s)

* + - (**Assign Instructor to chosen section)**

Insert Into Teaches (Course\_ID, Instructor\_ID, Section\_ID) VALUES (%s, %s, %s)

* + - **(Delete choice entity from database)**

DELETE FROM Choice WHERE Instructor\_ID = %s AND Course\_ID = %s AND Section\_ID = %s AND Section2\_ID = %s

**Admin Logged in:**

* **Admin creates a course and section simultaneously**
  + To create a course and a section
    - (**Insert course)**

INSERT INTO Course (Course\_ID, Course\_Name, Course\_Credits, Course\_Description, Department\_Name ) VALUES (%s, %s,%s, %s, %s)

* + - **(Insert Section)**

INSERT INTO Section (Section\_ID, Course\_ID, Course\_Name, Semester, Section\_Day, Section\_Time, Student\_Limit ) VALUES (%s, %s,%s, %s, %s, %s, %s)

* **Admin creates a section for an existing course**
  + To check if course exists
    - **(Check if course exists)**

Select \* FROM Course Where Course\_ID = %s

* + To create a section for existing course
    - (**Insert section)**

INSERT INTO Section (Section\_ID, Course\_ID, Course\_Name, Semester, Section\_Day, Section\_Time, Student\_Limit ) VALUES (%s, %s,%s, %s, %s, %s, %s)

* **Admin creates a new department**
  + To create a department
    - (**Insert department)**

INSERT INTO Department (Department\_Name, Department\_Address, Department\_Email) VALUES (%s, %s, %s)

* **Admin creates a new admin account**
  + To create an admin account
    - (**Insert admin)**

INSERT INTO Administrator (Admin\_ID, Admin\_Password) VALUES (%s, %s)

* **Admin creates a new instructor account and profile**
  + To create an instructor account/profile
    - (**Insert instructor)**

INSERT INTO Instructor (Instructor\_ID, Instructor\_Password, Instructor\_Fname, Instructor\_Lname, Instructor\_Email, Department\_Name) VALUES (%s, %s, %s, %s, %s, %s)

* **Admin assigns an instructor to an existing course/section**
  + To check if course and section exist
    - **(Check Course and check section)**

Select \* FROM Course Where Course\_ID = %s

Select \* FROM Section Where Section\_ID = %s and Course\_ID = %s

* + To assign an instructor to a section
    - (**Assign Instructor)**

INSERT INTO Teaches (Section\_ID, Course\_ID, Instructor\_ID) VALUES (%s, %s,%s)

* **Admin proposes two sections to an instructor**
  + To create a new choice option with two sections
    - **(Create Choice)**

INSERT INTO Choice (Admin\_ID, Instructor\_ID, Course\_ID, Course\_Name, Course\_Credits, Course\_Description, Department\_Name, Semester, Student\_Limit, Section\_ID, Section\_Day, Section\_Time, Section2\_ID, Section2\_Day, Section2\_Time) VALUES (%s, %s, %s, %s,%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)