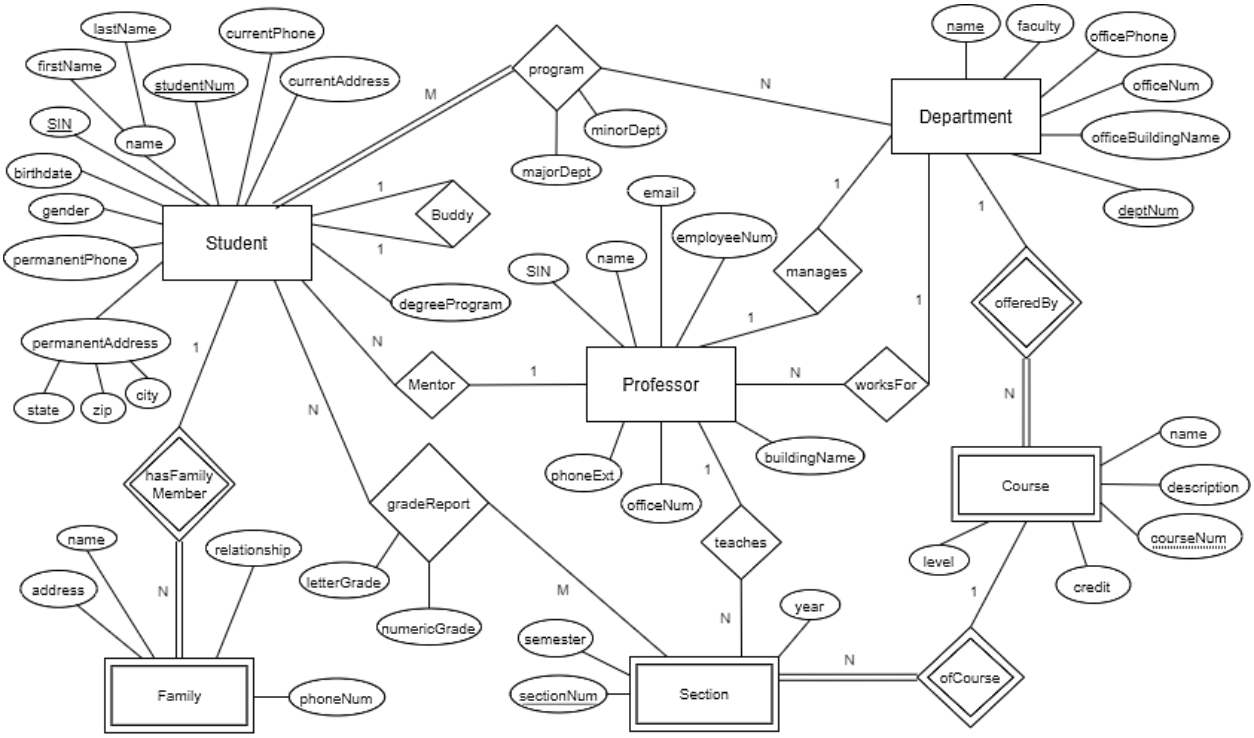
Question 1

Part A

Assumption(s)

Assuming a professor can mentor multiple students, but a student can be mentored by one professor.

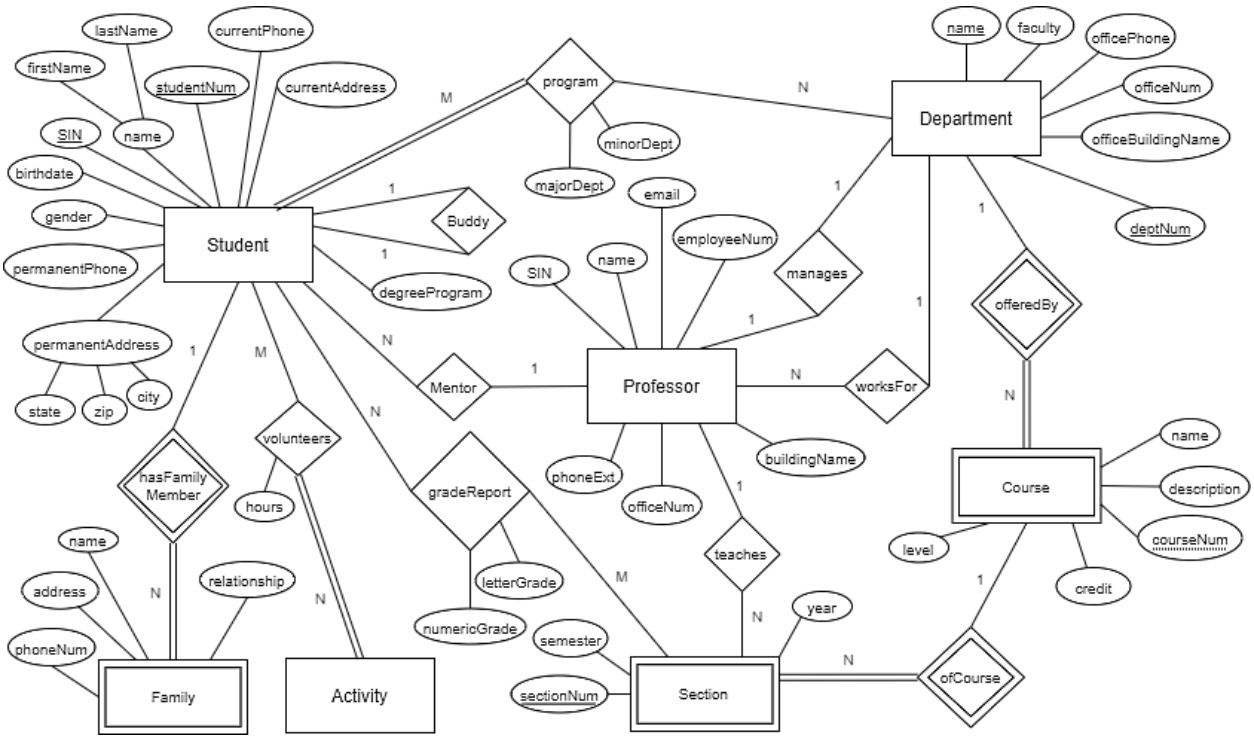
Assuming a student can have a grade report for multiple sections (one section for each of the courses they are enrolled in), and each section can have multiple grade reports for students enrolled in that section.



Part B

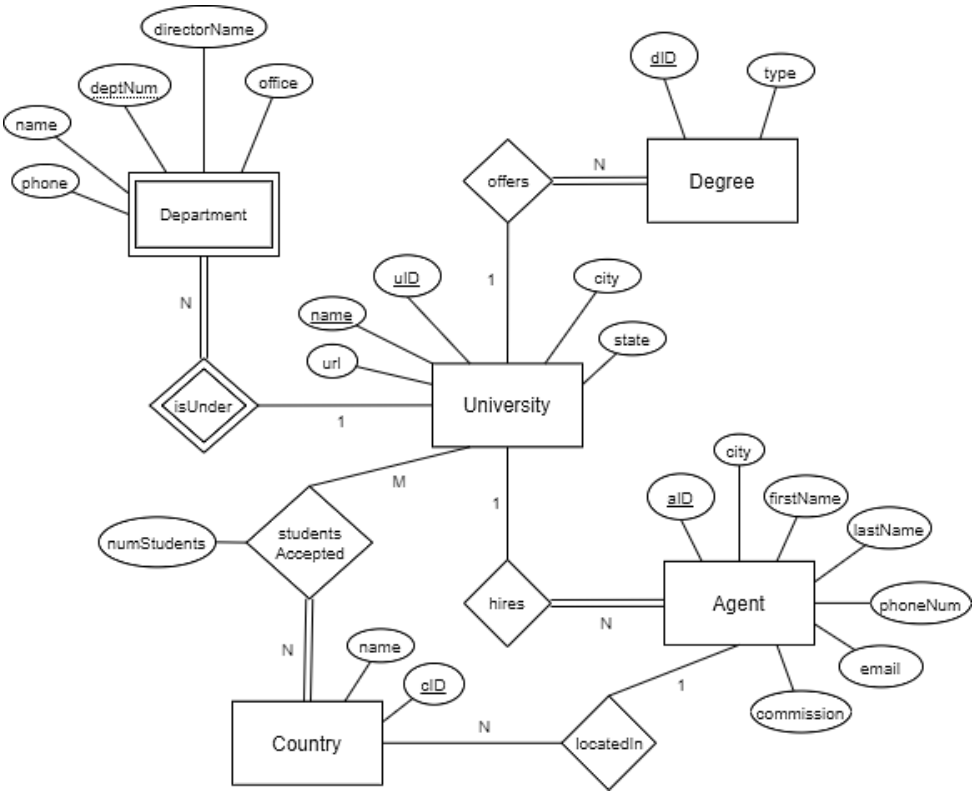
Additional Requirement Specification

Activity: Students may be volunteers for activities at the University if they desire. An activity must have at least one student volunteer in it. The University keeps track of the number of hours per week that a student spends volunteering for each activity.



Question 2

Part A



Part B

University (uID, name, url, city, state)

Degree (dID, type, uID)

Department (deptNum, uID, name, office, phone, directorName)

Agent (aID, city, firstName, lastName, phoneNum, email, commission, cID, uID)

Country (cID, name)

StudentsAccepted (numStudents, cID, uID)

Part C

- Is University in 1NF?
  - Yes, because it is a relation; are no repeating tuples, and every value in a cell is atomic
- Is University in 2NF?
  - Yes, because (1) it is in 1NF and (2) all the non-prime attributes are fully functionally dependent on the primary key since it is a simple attribute.
- Is University in 3NF?
  - Yes, because it is in 2NF and there are no transitive dependencies.

- Is Degree in 1NF?
  - Yes, because it is a relation; are no repeating tuples, and every value in a cell is atomic
- Is Degree in 2NF?
  - Yes, because (1) it is in 1NF and (2) all the non-prime attributes are fully functionally dependent on the primary key since it is a simple attribute.
- Is Degree in 3NF?
  - Yes, because it is in 2NF and there are no transitive dependencies.

- Is Department in 1NF?
  - Yes, because it is a relation; are no repeating tuples, and every value in a cell is atomic
- Is Department in 2NF?
  - Yes, because (1) it is in 1NF and (2) all the non-prime attributes are fully functionally dependent on the primary key since it is a simple attribute.
- Is Department in 3NF?
  - Yes, because it is in 2NF and there are no transitive dependencies.

- Is Agent in 1NF?
  - Yes, because it is a relation; are no repeating tuples, and every value in a cell is atomic
- Is Agent in 2NF?
  - Yes, because (1) it is in 1NF and (2) all the non-prime attributes are fully functionally dependent on the primary key since it is a simple attribute.
- Is Agent in 3NF?
  - Yes, because it is in 2NF and there are no transitive dependencies.

- Is Country in 1NF?
  - Yes, because it is a relation; are no repeating tuples, and every value in a cell is atomic
- Is Country in 2NF?
  - Yes, because (1) it is in 1NF and (2) all the non-prime attributes are fully functionally dependent on the primary key since it is a simple attribute.
- Is Country in 3NF?
  - Yes, because it is in 2NF and there are no transitive dependencies.

Question 3

Note that there is a composite primary key: (examNo, questionNo, userId)

Part A

- (examNo, questionNo, userId) → examTitle
- (examNo, questionNo, userId) → examTimeAllowed
- (examNo, questionNo, userId) → questionText
- (examNo, questionNo, userId) → correctAnswer
- (examNo, questionNo, userId) → answerOptions
- (examNo, questionNo, userId) → optionText
- (examNo, questionNo, userId) → userEmail
- (examNo, questionNo, userId) → userPassword
- (examNo, questionNo, userId) → userFname
- (examNo, questionNo, userId) → userLname
- (examNo, questionNo, userId) → userAddress
- (examNo, questionNo, userId) → userCity
- (examNo, questionNo, userId) → userState
- (examNo, questionNo, userId) → userZip
- (examNo, questionNo, userId) → startTime
- (examNo, questionNo, userId) → finishTime
- (examNo, questionNo, userId) → userResponse
- (userId) → userFname
- (userId) → userLname
- (userId) → userAddress
- (userId) → userCity
- (userId) → userState
- (userId) → userZip
- (userId) → userEmail
- (userId) → userPassword
- (userId) → startTime
- (userId) → finishTime
- (userId, userEmail) → userPassword
- (userEmail) → userPassword
- (userEmail) → userFname
- (userEmail) → userLname
- (userZip) → userState
- (userZip) → userCity
- (examNo) → examTitle
- (examNo) → examTimeAllowed
- (examNo, questionNo) → questionText
- (examNo, questionNo) → answerOptions
- (examNo, questionNo, answerOptions) → correctAnswer
- (examNo, questionNo, answerOptions) → optionText

Part B

Step 1: Determine whether the relation is in 3NF.

- Is this relation in 1NF?
  - Yes, because there are no repeating tuples; every value in a cell is atomic.
- Is this relation in 2NF?
  - No, because although the relation is in 1NF, there are non-full functional dependencies on the primary key.
    (userId) → userFname
    (userId) → userLname
    (userId) → userAddress
    (userId) → userCity
    (userId) → userState
    (userId) → userZip
    (userId) → userEmail
    (userId) → userPassword
    (userId, userEmail) → userPassword
    (examNo) → examTitle
    (examNo) → examTimeAllowed
    (examNo, questionNo) → questionText
    (examNo, questionNo) → answerOptions
    (examNo, questionNo, answerOptions) → optionText
    (examNo, questionNo, answerOptions) → correctAnswer
- Since the relation is not in 2NF, it is not in 3NF.

Step 2: Decompose to remove the partial dependencies.

Original Relation:

> OnlineExam (examNo, questionNo, userId, examTitle, examTimeAllowed, questionText, answerOptions, optionText, correctAnswer, userEmail, userPassword, userFname, userLname, userAddress, userCity, userState, userZip, userResponse, startTime, finishTime)

First Decomposition: Decomposing OnlineExam relation

> Note that many of the non-prime attributes in the original relation depend on *userId* and *examNo* (i.e., proper subsets of the primary key).

- ExamInfo (examNo, examTitle, examTimeAllowed)
- UserAndExamInfo (examNo, questionNo, userId, questionText, answerOptions, optionText, correctAnswer, userEmail, userPassword, userFname, userLname, userAddress, userCity, userState, userZip, userResponse, startTime, finishTime)

➢ Is the decomposition non-loss?
  - Since the common attribute of both relations (*examNo*) is the primary key of the ExamInfo relation, a natural join would restore the original OnlineExam relation. Therefore, the decomposition is non-loss.

Second Decomposition: Decomposing UserExamInfo relation

Note that some of the non-prime attributes in the UserExamInfo relation depend on (*examNo*, *questionNo*) and *userId* (i.e., proper subsets of the primary key).

- ExamQuestions (examNo, questionNo, questionText, answerOptions, optionText, correctAnswer)
- UserAndExamInfo2 (examNo, questionNo, userId, userEmail, userPassword, userFname, userLname, userAddress, userCity, userState, userZip, userResponse, startTime, finishTime)

➢ Is the decomposition non-loss?
  - Since the common attributes of both relations (*examNo*, *questionNo*) make the primary key of the ExamQuestions relation, a natural join would restore the original UserExamInfo relation. Therefore, the decomposition is non-loss.

Third Decomposition: Decomposing UserExamInfo2 relation

Note that some of the non-prime attributes in the UserExamInfo2 relation depend on *userId* (i.e., a proper subset of the primary key).

- UserInfo (userId, userFname, userLname, userAddress, userCity, userState, userZip, userEmail, userPassword)
- BasicUserExamInfo (examNo, questionNo, userId, userResponse, startTime, finishTime)

➢ Is the decomposition non-loss?
  - Since the common attribute of both relations (*userId*) is the primary key of the UserInfo relation, a natural join would restore the original UserExamInfo2 relation. Therefore, the decomposition is non-loss.

Step 3: Determine whether the decomposed relations are in 3NF.

Check 2NF

- Is ExamInfo in 2NF? Yes
- Is ExamQuestions in 2NF? Yes
- Is UserInfo in 2NF? Yes
- Is BasicUserExamInfo in 2NF? Yes

Check 3NF

- Is ExamInfo in 3NF?
  - Yes, because it is in 2NF, and there are no transitive dependencies.
- Is ExamQuestions in 3NF?
  - No, because although it is in 2NF, there are transitive dependencies. Attributes *optionText* and *correctAnswer* depend on the non-prime attribute *answerOptions*.
- Is UserInfo in 3NF?
  - No, because although it is in 2NF, there are transitive dependencies. Attributes *userState* and *userCity* depend on the non-prime attribute *userZip*.
- Is BasicUserExamInfo in 3NF?
  - Yes, because it is in 2NF, and there are no transitive dependencies.

Step 4: Further decompose relations that are not in 3NF so they can be in 3NF.

Decompose ExamQuestions to make 3NF.

> Note that some of the non-prime attributes in the ExamQuestions relation depend on *answerOptions* (i.e., a non-prime attribute).

- ExamQuestions2 (examNo, questionNo, questionText, answerOptions)
- QuestionInfo (examNo, questionNo, answerOptions, optionText, correctAnswer)

➢ Is the decomposition non-loss?
  - Since the common attributes of both relations (*examNo, questionNo, answerOptions*) make the primary key of the QuestionInfo relation, a natural join would restore the original ExamQuestions relation. Therefore, the decomposition is non-loss.

➢ Is this a good decomposition?
  - Yes, because the original dependencies from the ExamQuestions relation are reserved.
    (examNo, questionNo) → questionText
    (examNo, questionNo) → answerOptions
    (examNo, questionNo, answerOptions) → optionText
    (examNo, questionNo, answerOptions) → correctAnswer

Decompose UserInfo to make 3NF.

> Note that some of the non-prime attributes in the UserInfo relation depend on *userZip* (i.e., a non-prime attribute).

- UserInfo2 (userId, userFname, userLname, userAddress, userEmail, userPassword. userZip)
- UserLocation (userZip, userCity, userState)

➢ Is the decomposition non-loss?
  - Since the common attribute of both relations (*userZip*) is the primary key of the UserLocation relation, a natural join would restore the original UserInfo relation. Therefore, the decomposition is non-loss.

➢ Is this a good decomposition?
  - Yes, because the original dependencies from the UserInfo relation are reserved.
    (userId) → userZip
    (userZip) → userCity
    (userZip) → userState
    And from a natural join on the two decomposed relations, we can retrieve
    (userId) → userCity
    (userId) → userState

Step 5: Decomposed Relations

- ExamInfo (examNo, examTitle, examTimeAllowed)
- BasicUserExamInfo (examNo, questionNo, userId, userResponse, startTime, finishTime)
- ExamQuestions2 (examNo, questionNo, questionText, answerOptions)
- QuestionInfo (examNo, questionNo, answerOptions, optionText, correctAnswer)
- UserInfo2 (userId, userFname, userLname, userAddress, userEmail, userPassword. userZip)
- UserLocation (userZip, userCity, userState)

Part C

- Is ExamInfo in BCNF?
  - Yes, because ExamInfo is in 3NF and every determinant is a candidate key
    (examNo) → examTitle
    (examNo) → examTimeAllowed
- Is BasicUserExamInfo in BCNF?
  - Yes, because BasicUserExamInfo is in 3NF and every determinant is a candidate key
    (examNo, questionNo, userId) → startTime
    (examNo, questionNo, userId) → finishTime
    (examNo, questionNo, userId) → userResponse
- Is ExamQuestions2 in BCNF?
  - Yes, because ExamQuestions2 is in 3NF and every determinant is a candidate key
    (examNo, questionNo) → questionText
    (examNo, questionNo) → answerOptions
- Is QuestionInfo in BCNF?
  - Yes, because QuestionInfo is in 3NF and every determinant is a candidate key
    (examNo, questionNo, answerOptions) → optionText
    (examNo, questionNo, answerOptions) → correctAnswer
- Is UserInfo2 in BCNF?
  - Yes, because UserInfo2 is in 3NF and every determinant is a candidate key
    (userId) → userFname
    (userId) → userLname
    (userId) → userAddress
    (userId) → userEmail
    (userId) → userPassword
    (userId) → userZip
- Is UserLocation in BCNF?
  - Yes, because UserLocation is in 3NF and every determinant is a candidate key
    (userZip) → userCity
    (userZip) → userState

Question 4

Part A: Answers attached in create_icru.sql

Part B: Answers attached in insert_icru.sql