Ralph Arvin De Castro

CIS*3530

F2018

Assignment 2

Question 1a)

Assumptions

STUDENT:

- Students must always have a department from their major.
- Departments must always have students.

DEPARTMENT:

- Department must always have a director.
- Professor must always a department to work with.

COURSE:

- Course number is a weak primary key since it is only unique for each department.

SECTION:

- Section number is a weak primary key since it is only unique within courses.
- Section must be and can be taken by many students.
- Students must take courses. Students can take more than one courses.

GRADE

- Numeric grade is derived from letter grade.

PROFESSOR

- Professors can work in a building different from their department's building.

MENTOR

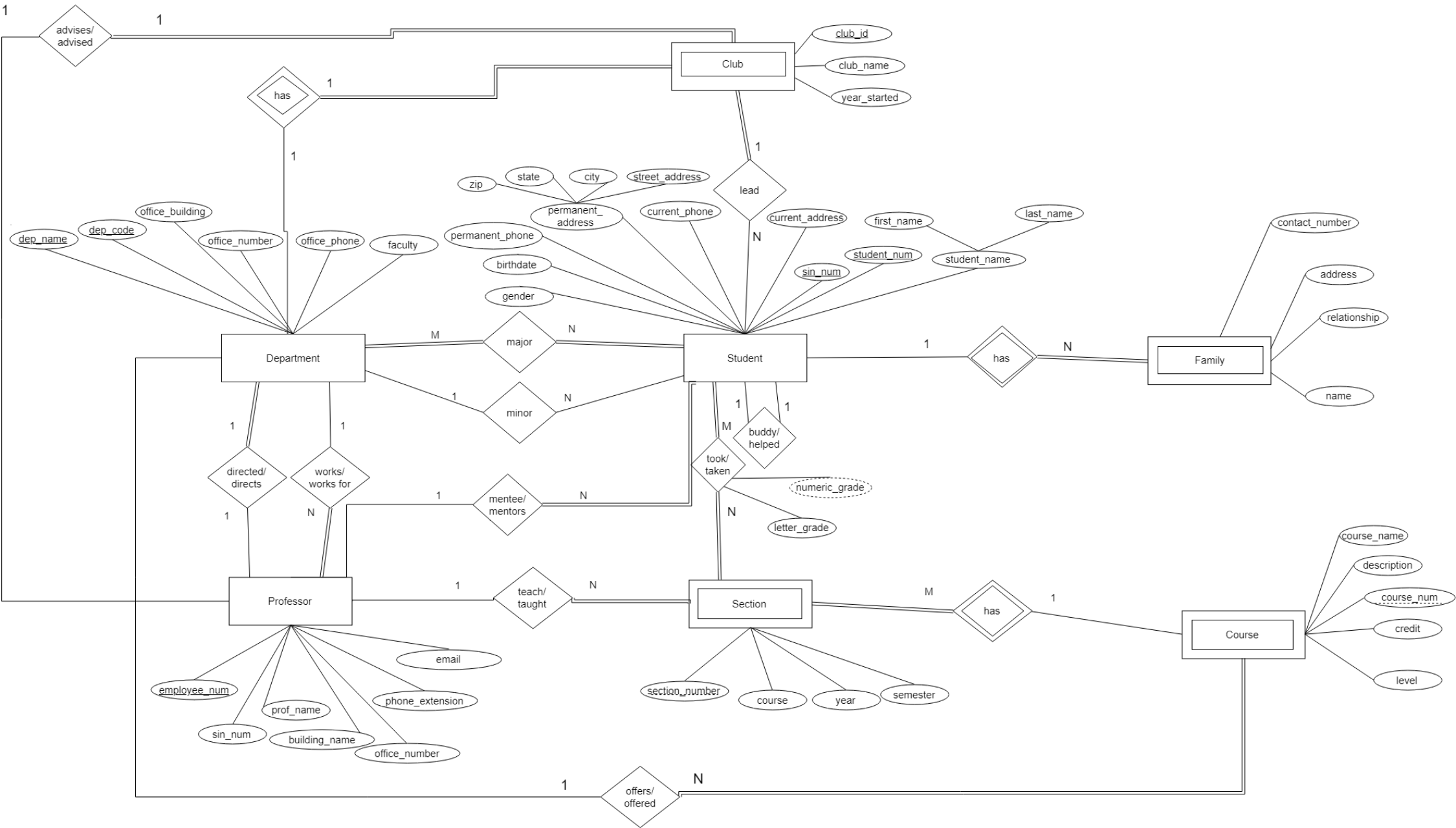
- Students must have a mentor.
- Professor does not require to have a mentee.

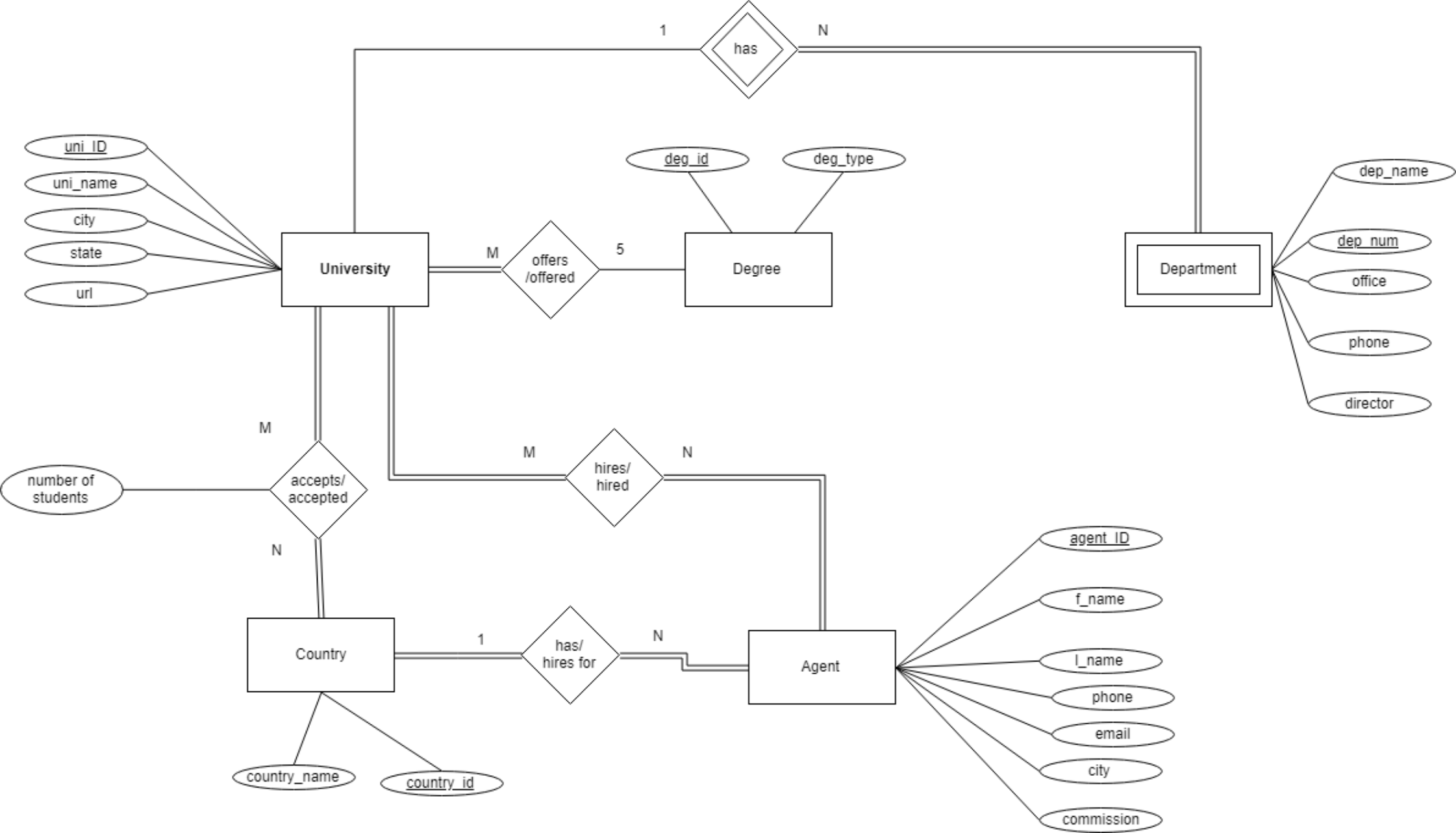
BUDDY

- Students can only have one buddy.
- A buddy can only have one student.

Question 1b)

Club: Departments can have a club that must be lead by students. This club must be advised by a professor. It is described by a club number and club name.


advises/
advised
1
1
Club
club_id
club_name
year_started
has
1
1
1
lead
N
zip
state
city
street_address
permanent_
address
current_phone
current_address
first_name
last_name
student_name
student_num
sin_num
permanent_phone
birthdate
gender
office_building
dep_code
dep_name
office_number
office_phone
faculty
Department
M
major
N
Student
1
has
N
Family
contact_number
address
relationship
name
1
minor
N
1
1
1
buddy/
helped
1
M
took/
taken
numeric_grade
letter_grade
N
1
directed/
directs
works/
works for
1
N
1
mentee/
mentors
N
Professor
1
teach/
taught
N
Section
M
has
1
Course
course_name
description
course_num
credit
level
email
employee_num
prof_name
phone_extension
sin_num
building_name
office_number
section_number
course
year
semester
1
offers/
offered
N


1
has
N
uni_ID
uni_name
city
state
url
University
M
offers
/offered
5
deg_id
deg_type
Degree
Department
dep_name
dep_num
office
phone
director
M
number of
students
accepts/
accepted
N
M
hires/
hired
N
Country
1
has/
hires for
N
Agent
agent_ID
f_name
l_name
phone
email
city
commission
country_name
country_id

Ralph Arvin De Castro

CIS*3530

F2018

Assignment 2

Question 1a)

Assumptions

STUDENT:

- Students must always have a department from their major.
- Departments must always have students.

DEPARTMENT:

- Department must always have a director.
- Professor must always a department to work with.

COURSE:

- Course number is a weak primary key since it is only unique for each department.

SECTION:

- Section number is a weak primary key since it is only unique within courses.
- Section must be and can be taken by many students.
- Students must take courses. Students can take more than one courses.

GRADE

- Numeric grade is derived from letter grade.

PROFESSOR

- Professors can work in a building different from their department's building.

MENTOR

- Students must have a mentor.
- Professor does not require to have a mentee.

BUDDY

- Students can only have one buddy.
- A buddy can only have one student.

Question 1b)

Club: Departments can have a club that must be lead by students. This club must be advised by a professor. It is described by a club number and club name.

Question 2a)

Assumptions

UNIVERSITY:

- All universities in this database must accept students at least in one country.
- Universities must offer at least one degree in the database.
- All universities must have at least one agent in this database.

DEGREE:

- There can be a degree that is not offered in all universities.

AGENTS:

- All agents in this database must be hired by at least one university.

COUNTRIES:

- All countries in this database send students in university.

Question 2b)

Recruit(uni_id, country_id, num_students)

Uni_Agents(uni_id, agent_id)

Country_Agents(country_id, agent_id)

Degree(deg_id, deg_type)

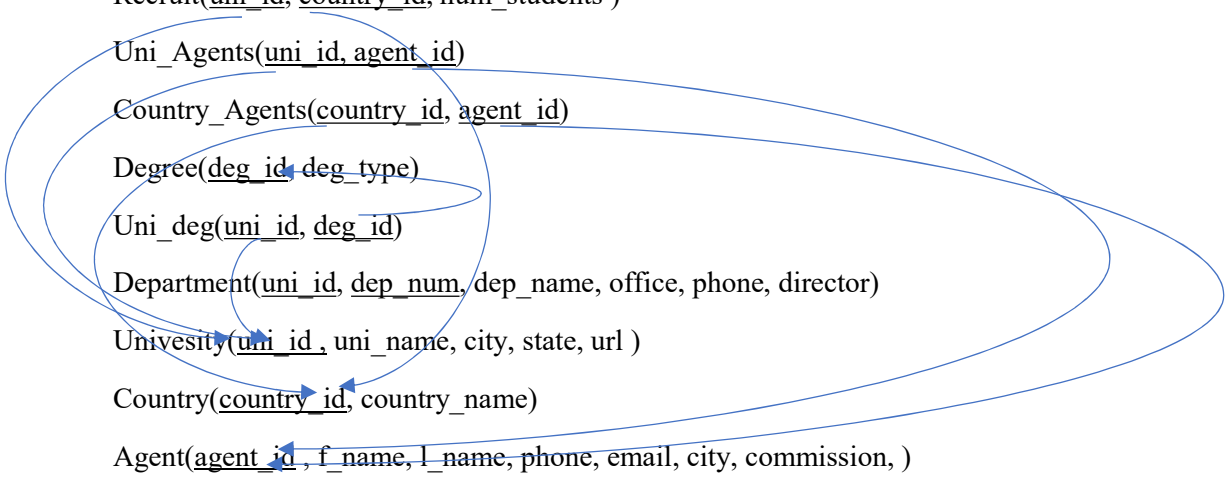
Uni_deg(uni_id, deg_id)

Department(uni_id, dep_num, dep_name, office, phone, director)

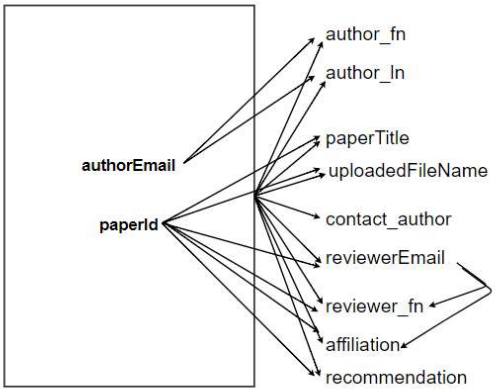
Univesity(uni_id , uni_name, city, state, url)

Country(country_id, country_name)

Agent(agent_id , f_name, l_name, phone, email, city, commission,)



Question 3a)



b) **2NF – take out partial dependencies (non-prime attributes depend on a prime attribute)**

In this case, every attribute that has partially dependencies on authorEmail and paperId has created new relations:

conference_review (authorEmail, paperId, contactAuthor)

author (authorEmail, author_fn, author_ln)

paper (paperId, paperTitle, uploadedFileName, reviewerEmail, reviewer_fn, affiliation, recommendation)

NON LOSS DECOMPOSITION because

- conference_review INTERSECT author = authorEmail (key of both relations)
- conference_review INTERSECT paper = paperId (key of both relations)

3NF – must be in 2NF and no transitive dependencies

In this case, every attribute that has transitive dependencies on reviewerEmail has created new relations:

conference_review (authorEmail, paperId, contactAuthor)

author (authorEmail, author_fn, author_ln)

paper (paperId, paperTitle, uploadedFileName, reviewerEmail, recommendation)

reviewer(reviewerEmail, reviewer_fn , affliation)

NON LOSS DECOMPOSITION because

- reviewer INTERSECT paper = reviewerEmail (key of reviewer)

c)

A relation is in BCNF if and only if every determinant is a Candidate key.

After normalizing conference_review to 3NF, the determinants of each relation are

conference_review (<u>authorEmail, paperId</u>, contactAuthor)

- authorEmail,paperId -> contactAuthor
- authorEmail,paperId is a CANDIDATE KEY and no prime attributes depending on non-prime attributes.

author (<u>authorEmail</u>, author_fn, author_ln)

- authorEmail -> author_fn, author_ln
- authorEmail is a CANDIDATE KEY and no prime attributes depending on non-prime attributes.

paper (<u>paperId</u>, paperTitle, uploadedFileName, reviewerEmail, recommendation)

- paperId-> paperTitle, uploadedFileName, reviewerEmail, recommendation CANDIDATE KEY
- paperId is a CANDIDATE KEY and no prime attributes depending on non-prime attributes.

reviewer(<u>reviewerEmail</u>, reviewer_fn , affliation)

- reviewerEmail -> reviewer_fn , affliation CANDIDATE KEY
- reviewerEmail is a CANDIDATE KEY and no prime attributes depending on non-prime attributes.

Result:

conference_review (<u>authorEmail, paperId</u>, contactAuthor)

author (<u>authorEmail</u>, author_fn, author_ln)

paper (<u>paperId</u>, paperTitle, uploadedFileName, reviewerEmail, recommendation)

reviewer(<u>reviewerEmail</u>, reviewer_fn , affliation)

Therefore, BNCF in this case is the same as 3NF.

Question 4a)

patient_info (Patient No, Drug No, Start Date, OHIP_No, FName, LName, Bed_No, Ward_No, Ward_Name, Ward_Manager, Drug_Name, Description, Dosage, Method, Units_pday, Start_date, Finish_date)

Question 4b)

Patient_No, Drug_No, Start_Date -> OHIP_No, FName, LName, Bed_No, Ward_No, Ward_Name, Ward_Manager, Drug_Name, Description, Dosage, Method, Units_pday, Finish_date

Patient_No -> OHIP_No, FName, LName, Bed_No, Ward_No, Ward_Name, Ward_Manager

Drug_No -> Drug_Name, Description

Bed_No -> Ward_No, Ward_Name, Ward_Manager

Ward_No -> Ward_Name, Ward_Manager

Assumptions: All beds in the hospital are unique.

Question 4c)

2NF – take out partial dependencies (non-prime attributes depend on a prime attribute)

In this case, every attribute that has partially dependencies on Patient No and Drug_No has created new relations:

patient_info (Patient No, Drug No, Start date , Finish_date, Units_pday, Dosage, Method)

Patient_only(Patient No, OHIP_No, FName, LName, Bed_No, Ward_No, Ward_Name, Ward_Manager)

Drug(Drug No , Drug_Name, Description, Dosage, Method)

NON LOSS DECOMPOSITION because

- patient_info INTERSECT Patient_only = Patient_No (key of both relations)
- patient_info INTERSECT Drug= Drug_No (key of both relations)

3NF – must be in 2NF and no transitive dependencies

In this case, every attribute that has transitive dependencies on Ward_No and Bed_No has created new relations:

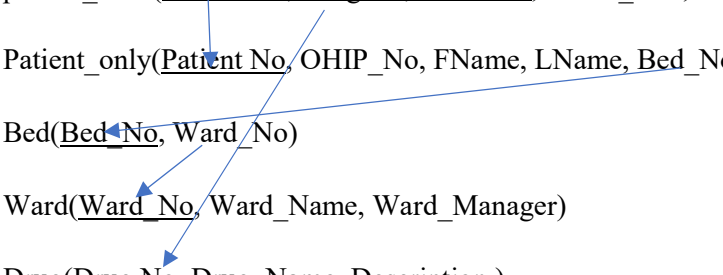
patient_info (Patient No, Drug No, Start date, Finish_date, Units_pday, Dosage, Method)

Patient_only(Patient No, OHIP_No, FName, LName, Bed_No)

Bed(Bed_No, Ward_No)

Ward(Ward_No, Ward_Name, Ward_Manager)

Drug(Drug No, Drug_Name, Description,)



NON LOSS DECOMPOSITION because

- Patient_only INTERSECT Bed = Bed_No(key of Bed)
- Ward_No INTERSECT Bed = WardNo(key of Ward)

Question 5a and 5b) on files