TUT

# Tutorial 1
# ER Modeling

## Classroom Exercise

1. Construct an ER diagram for a hospital with the following user requirements:

- The hospital has wards identified by a ward number. A ward is located in a particular block and on a particular floor.
- Each ward consists of different rooms. All rooms have room numbers unique only to the ward but not unique between different wards. Each ward has at least one room.
- Rooms may be of different sizes, with different number of beds and possibly with attached bathrooms. Rooms with the same number of beds need not be of the same size nor have similar bath facilities.
- When a patient registers, information about his IC#, first name, surname, address, date of birth, registration date and gender are recorded. Upon discharge, the discharge date is also recorded.
- If a patient is allocated to a room, no bed allocation details are recorded.
- There are three categories of wards: cardiology, paediatrics and surgery. Each paediatric ward has a number of baby nurseries, while each surgical ward has a number of intensive care units (ICU).
- During his/her hospitalisation, a patient may undergo surgical treatment, whereby multiple operations are possible. Complex procedures might require multiple surgeons.
- An operation is characterised by the type of operation and the normal duration. The date / time of a particular operation has to be recorded.
- Every surgeon has been involved in at least one operation but not every operation type has been carried out in this hospital.
- Information about nurses includes staff#, first name, surname, and date of joining.
- Every nurse is assigned to a ward. A ward is staffed by many nurses. However, a ward is managed by only one nurse. Not every nurse manages a ward.
- Nurses are designated as student nurses or registered nurses. The particular year of study for student nurses and the year of graduation and rank for registered nurses are also stored.

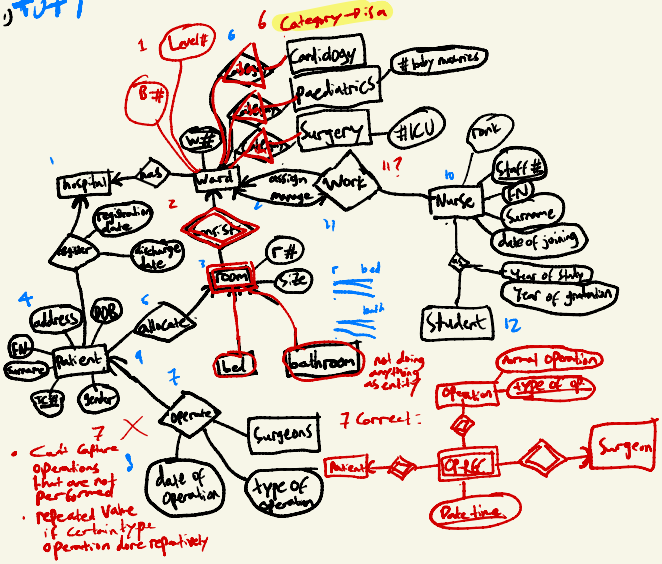
# Critical Thinking Exercise

2. Design an ER model for a university library's database with the following requirements.

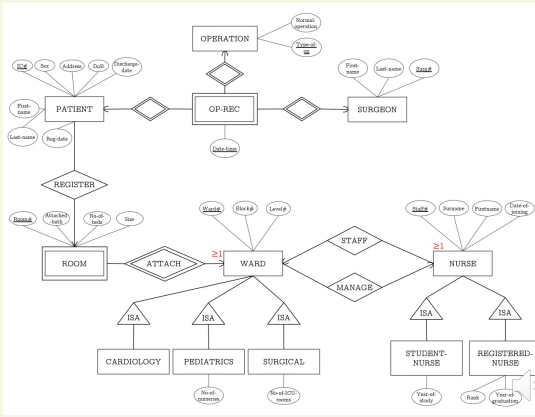
- The library keeps books with book identifiers, titles, authors and years of publication.
- Each book belongs to a certain topical category which is identified by the category name.
- Each book category may have a parent category and / or multiple sub-categories.
- Books can be borrowed by library members. Library members are identified by their identification numbers. The other member information including name, address and telephone numbers are also stored.
- There are differently named types of memberships. Each membership type has a different cap on the number of books that can be borrowed and the loan duration. It is assumed that a member can borrow a book several times on different dates. Each book loan also comes with a return date determined by the loan duration of the membership type.
- The library restricts its textbooks, reference books and student reports to be used within the library; i.e. these types of books cannot be loaned. For textbooks, reference books and student reports, the course name, subject topic and school are required, respectively.
- Finally, the topical categories of books that can be borrowed are restricted by the membership type.

State all assumptions clearly.

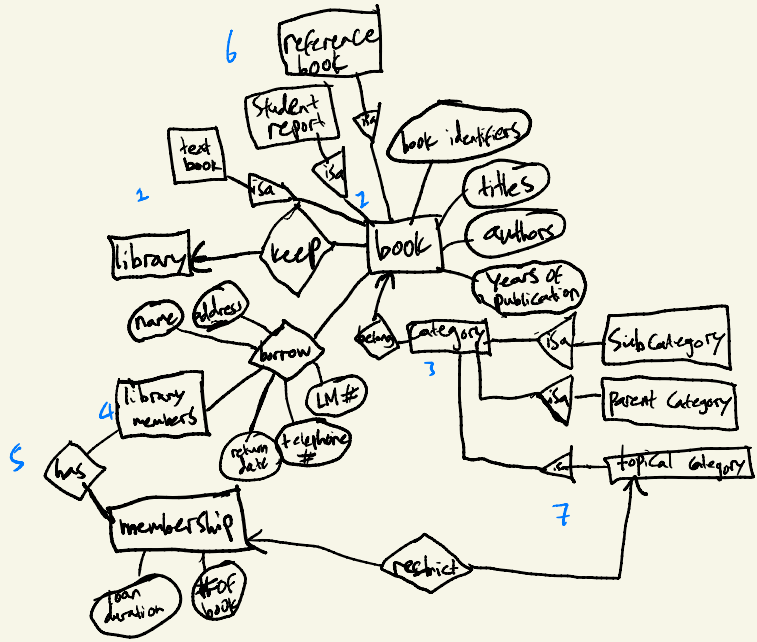
1)

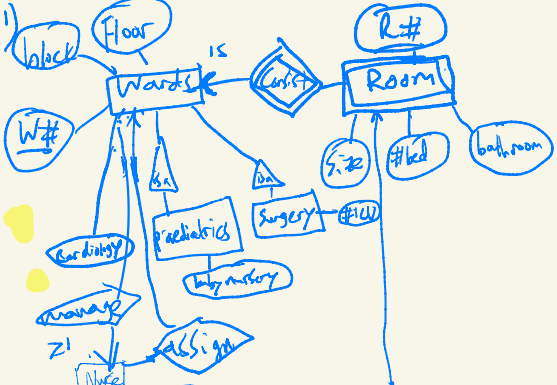


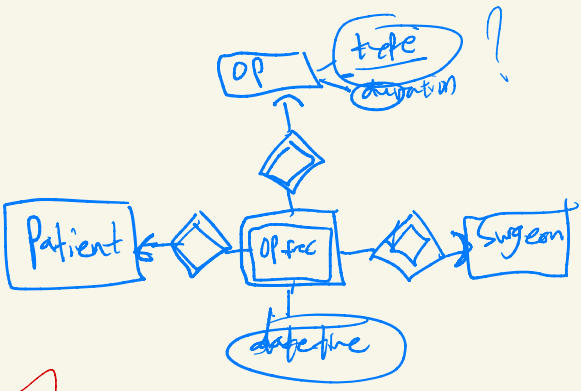

Suggested Solution

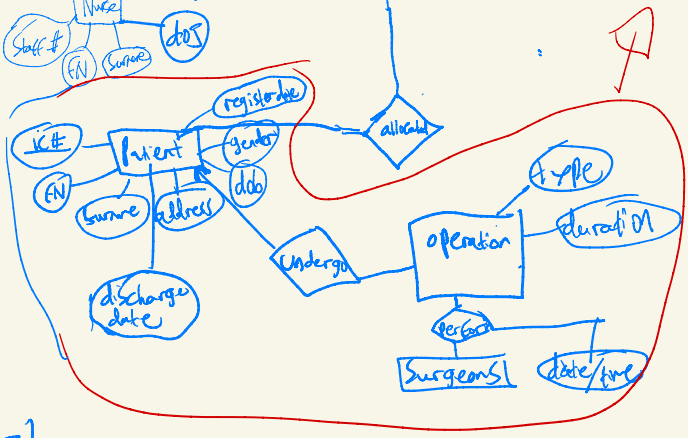


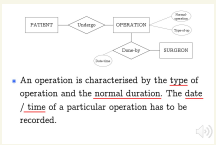

2)



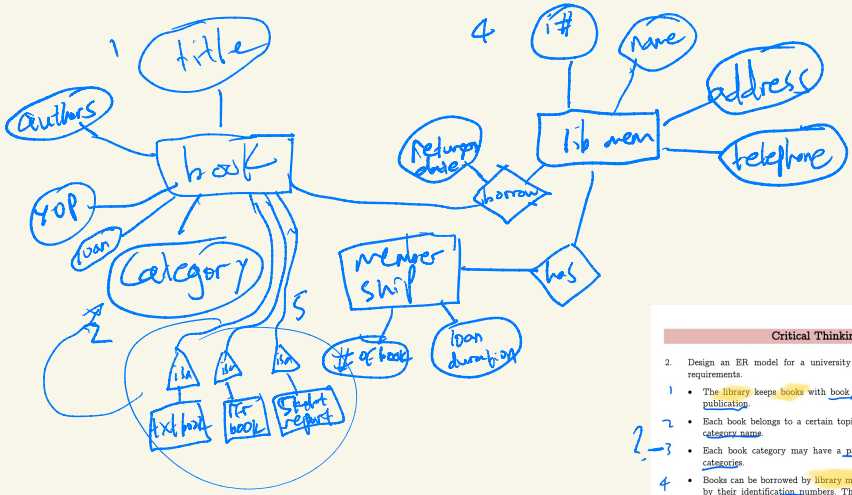



- An operation is characterised by the type of operation and the normal duration. The date / time of a particular operation has to be recorded.




## Critical Thinking Exercise

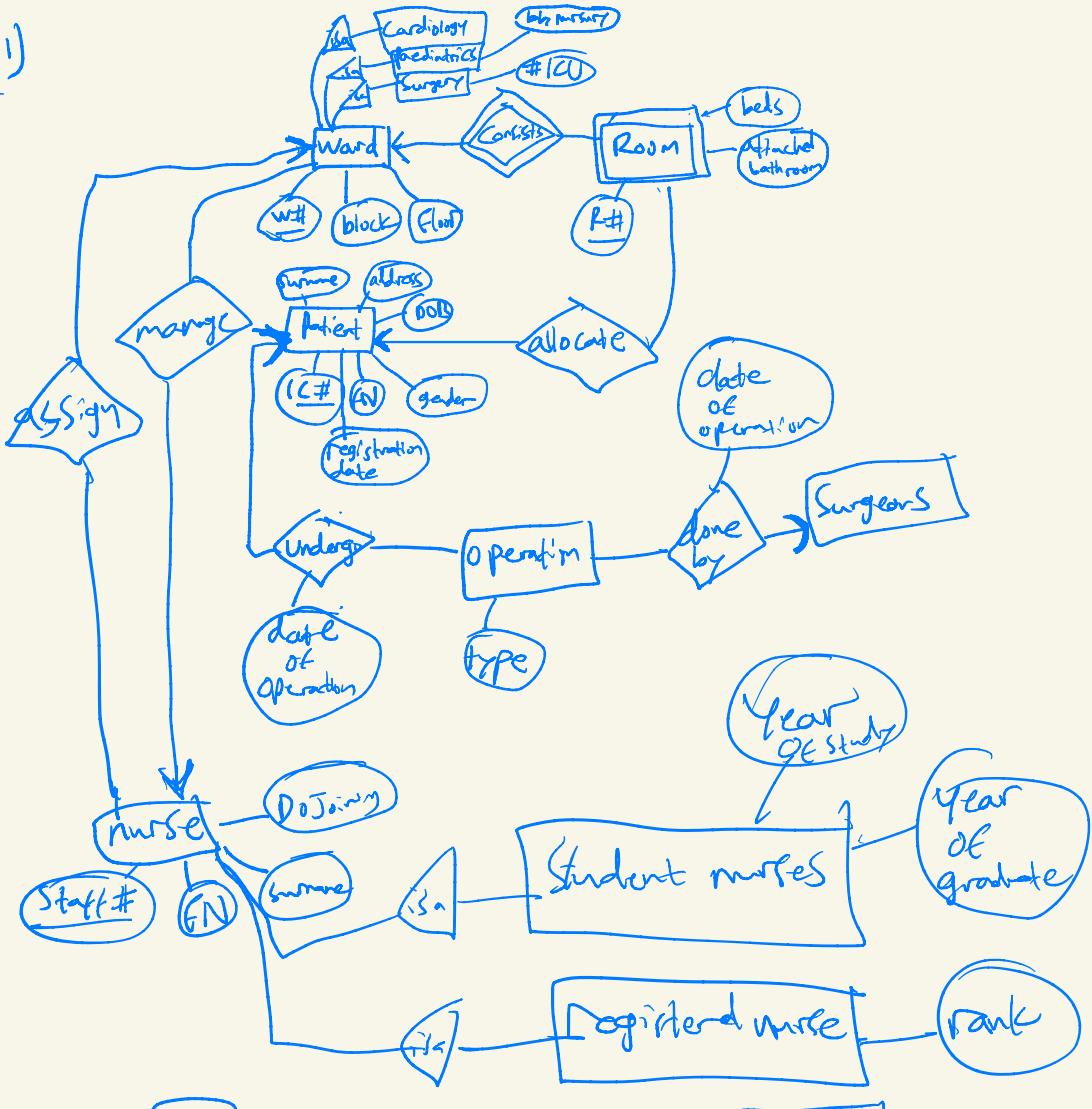
2. Design an ER model for a university library's database with the following requirements.

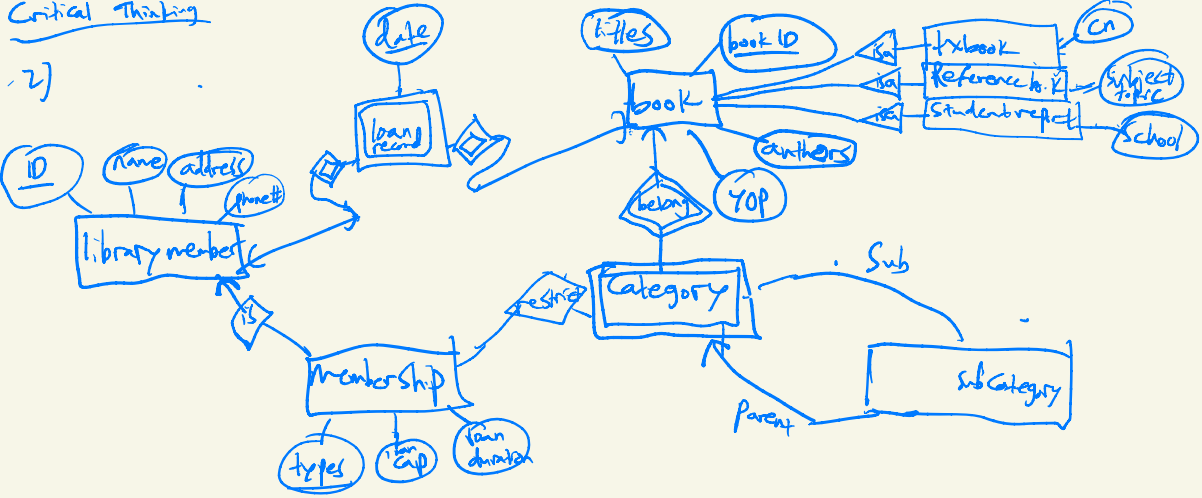
- The library keeps books with book identifiers, titles, authors and years of publication.
- Each book belongs to a certain topical category which is identified by the category name.
- Each book category may have a parent category and / or multiple sub categories.
- Books can be borrowed by library members. Library members are identified by their identification numbers. The other member information including name, address and telephone numbers are also stored.
- There are differently named types of memberships. Each membership type has a different cap on the number of books that can be borrowed and the loan duration. It is assumed that a member can borrow a book several times on different dates. Each book loan also comes with a return date determined by the loan duration of the membership type.
- The library restricts its textbooks, reference books and student reports to be used within the library; i.e. these types of books cannot be loaned. For textbooks, reference books and student reports, the course name, subject topic and school are required, respectively.
- Finally, the topical categories of books that can be borrowed are restricted by the membership type.

State all assumptions clearly.

TUT 1

1)


Cardiology
Paediatrics
Surgery
Maternity
#ICU
Ward
consists
Room
beds
Attached bathroom
W#
block
Floor
R#
manage
Patient
allocate
assign
Surname
address
DOB
IC#
FN
gender
registration date
date of operation
Undergo
Operation
done by
Surgeons
date of operation
type
Year of study
nurse
DOJoining
Staff#
FN
Surname
is a
Student nurses
Year of graduate
is a
Registered nurse
rank
2)
book ID
title
authors
years of publication
book
belong to
Category
Parent Category
Sub Category
borrow
has
membership
lib member
tel num
ID#
name
address



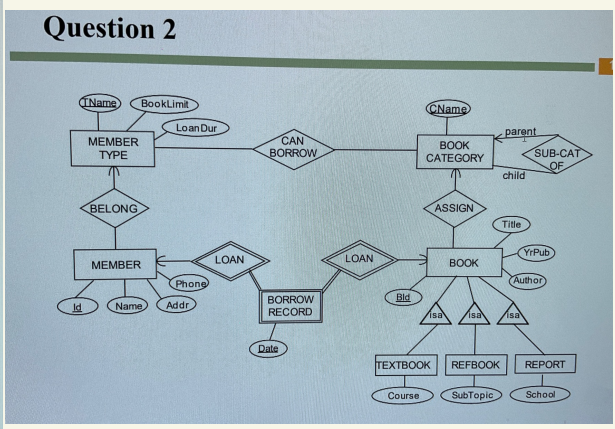

CZ2007

## Critical Thinking Exercise

2. Design an ER model for a university library's database with the following requirements.

- The library keeps books with book identifiers, titles, authors and years of publication.
- Each book belongs to a certain topical category which is identified by the category name.
- Each book category may have a parent category and / or multiple sub-categories.
- Books can be borrowed by library members. Library members are identified by their identification numbers. The other member information including name, address and telephone numbers are also stored.
- There are differently named types of memberships. Each membership type has a different cap on the number of books that can be borrowed and the loan duration. It is assumed that a member can borrow a book several times on different dates. Each book loan also comes with a return date determined by the loan duration of the membership type.
- The library restricts its textbooks, reference books and student reports to be used within the library; i.e. these types of books cannot be loaned. For textbooks, reference books and student reports, the course name, subject topic and school are required, respectively.
- Finally, the topical categories of books that can be borrowed are restricted by the membership type.

State all assumptions clearly.



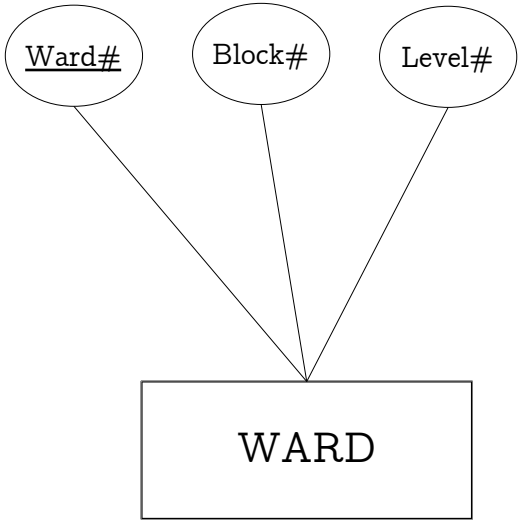
# CZ2007 Tutorial 1: ER Modeling

Week 3

# Question 1

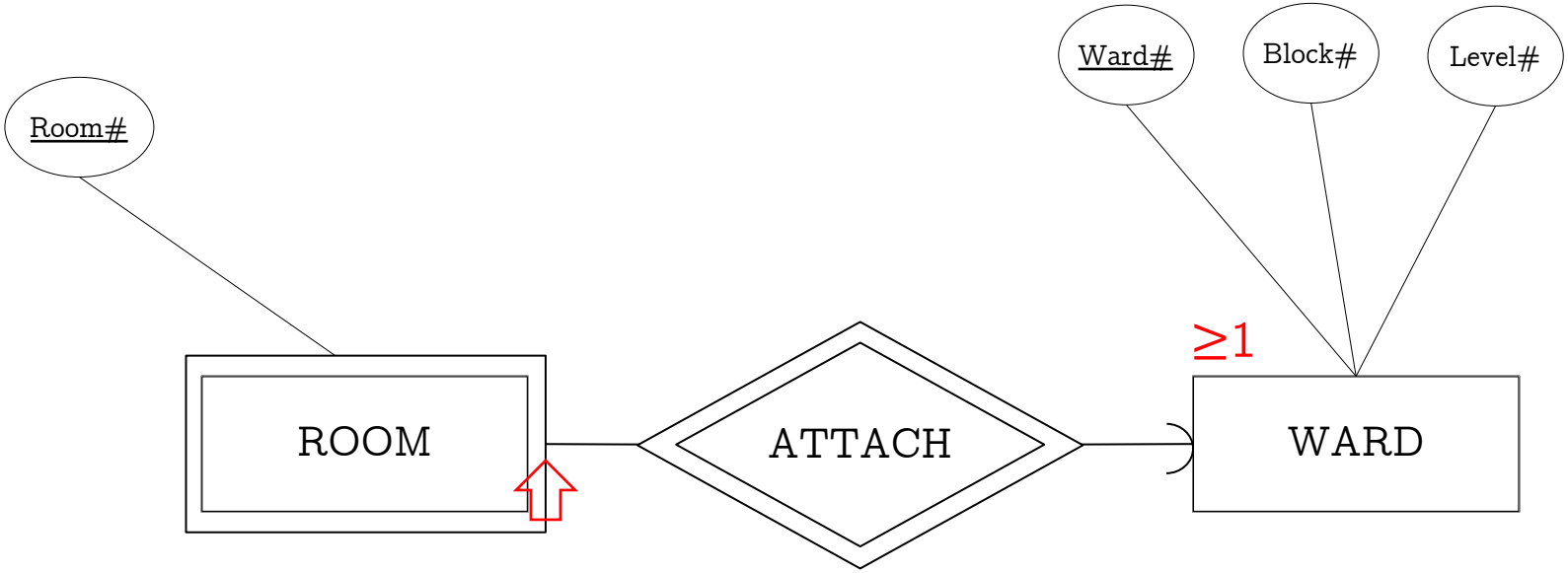
- The hospital has wards identified by a ward number. A ward is located in a particular block and on a particular floor.
- Each ward consists of different rooms. All rooms have room numbers unique only to the ward but not unique between different wards. Each ward has at least one room.
- Rooms may be of different sizes, with different number of beds and possibly with attached bathrooms. Rooms with the same number of beds need not be of the same size nor have similar bath facilities.

# Question 1



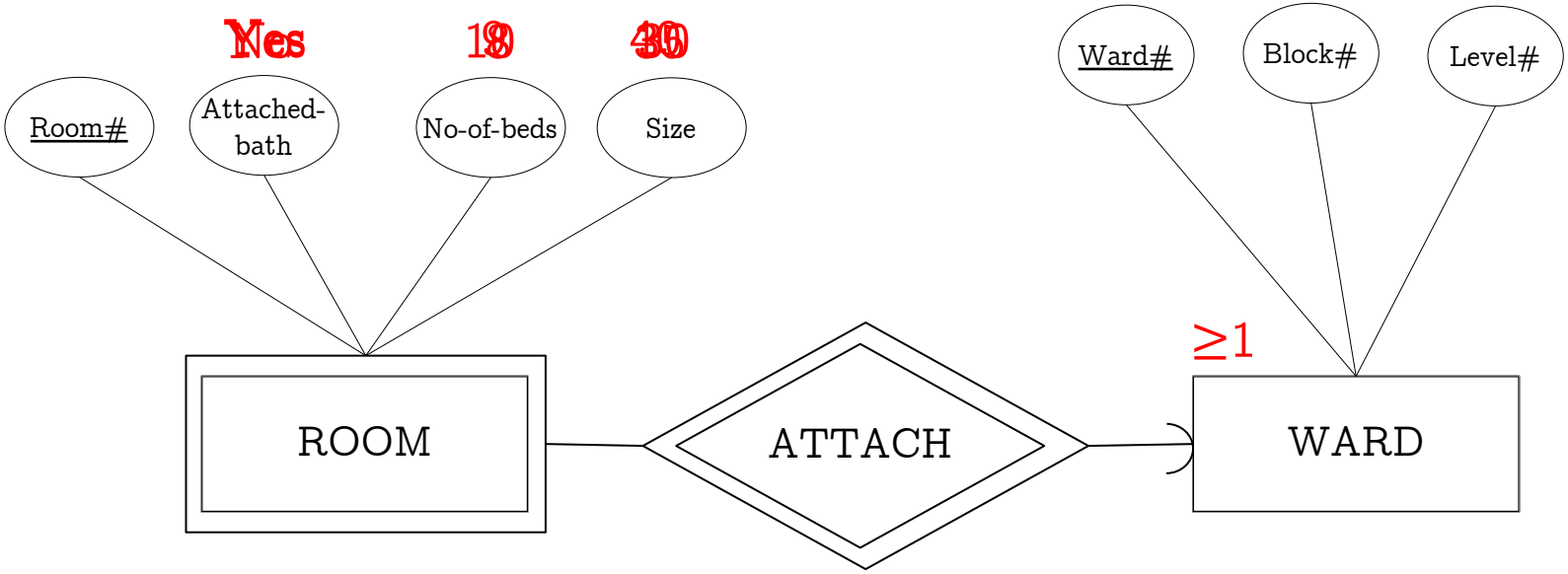

- The hospital has wards identified by a ward number. A ward is located in a particular block and on a particular floor.

# Question 1




- Each ward consists of different rooms. All rooms have room numbers unique only to the ward but not unique between different wards. Each ward has at least one room.

# Question 1

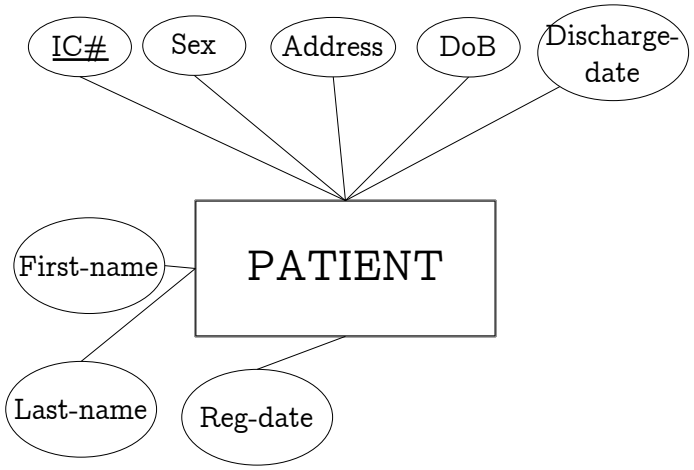



- Rooms may be of different sizes, with different number of beds and possibly with attached bathrooms. Rooms with the same number of beds need not be of the same size nor have similar bath facilities.

# Question 1

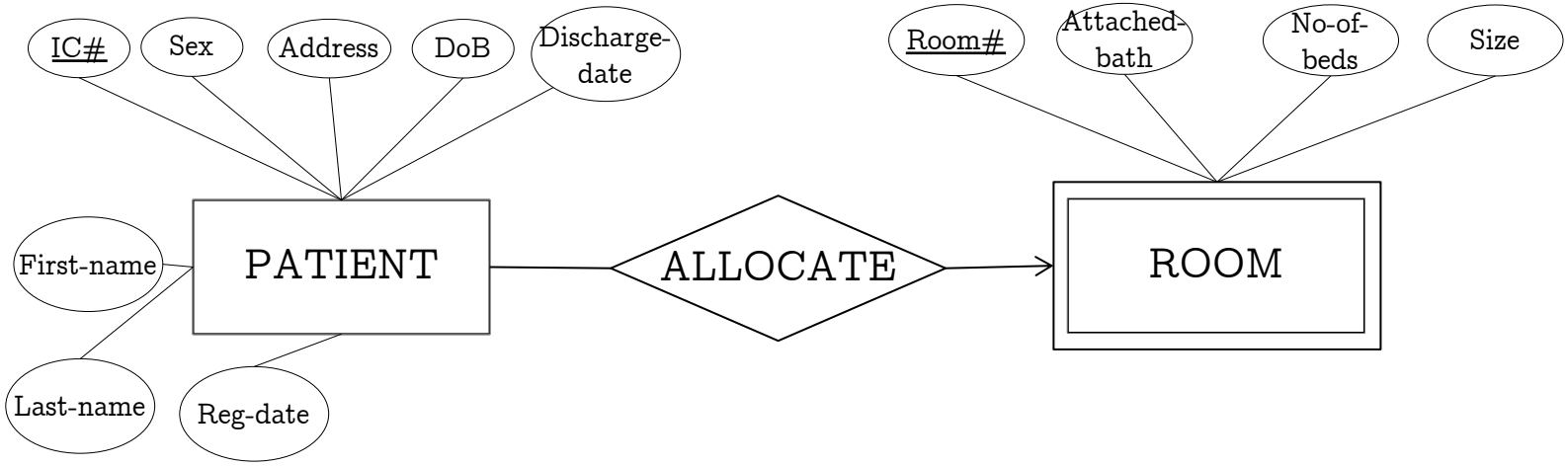
- When a patient registers, information about his IC#, first name, surname, address, date of birth, registration date and gender are recorded. Upon discharge, the discharge date is also recorded.
- If a patient is allocated to a room, no bed allocation details are recorded.
- There are three categories of wards: cardiology, paediatrics and surgery. Each paediatric ward has a number of baby nurseries, while each surgical ward has a number of intensive care units (ICU).

# Question 1



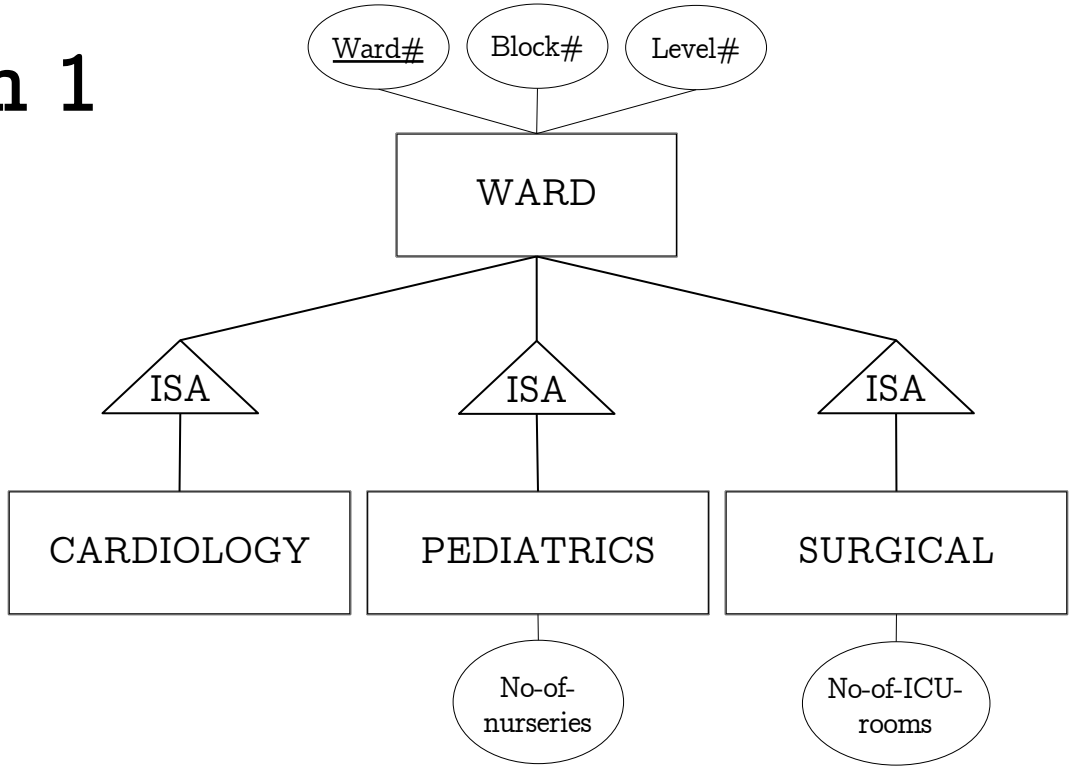

- When a patient registers, information about his IC#, first name, surname, address, date of birth, registration date and gender are recorded. Upon discharge, the discharge date is also recorded.

# Question 1




- If a patient is allocated to a room, no bed allocation details are recorded.

# Question 1

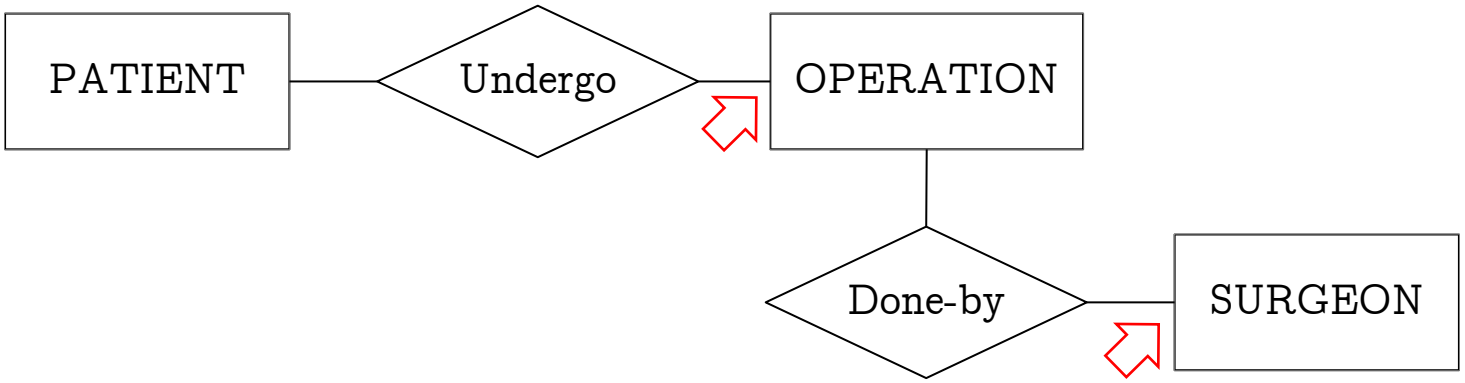



- There are three categories of wards: cardiology, paediatrics and surgery. Each paediatric ward has a number of baby nurseries, while each surgical ward has a number of intensive care units (ICU).

# Question 1

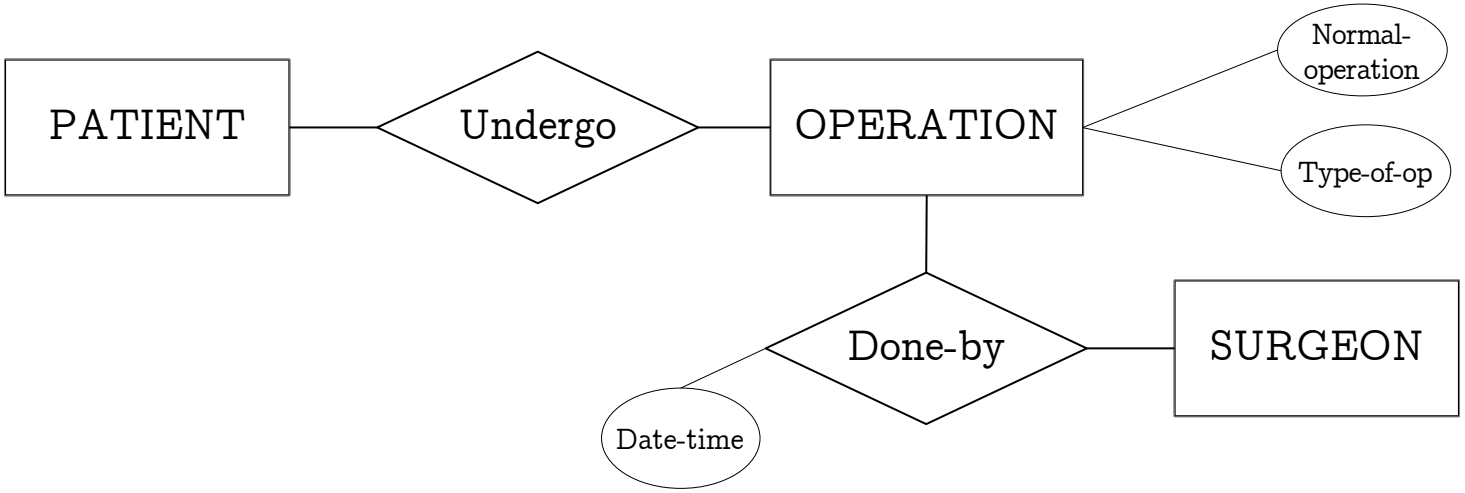
- During his/her hospitalisation, a patient may undergo surgical treatment, whereby multiple operations are possible. Complex procedures might require multiple surgeons.
- An operation is characterised by the type of operation and the normal duration. The date / time of a particular operation has to be recorded.
- Every surgeon has been involved in at least one operation but not every operation type has been carried out in this hospital.

# Question 1



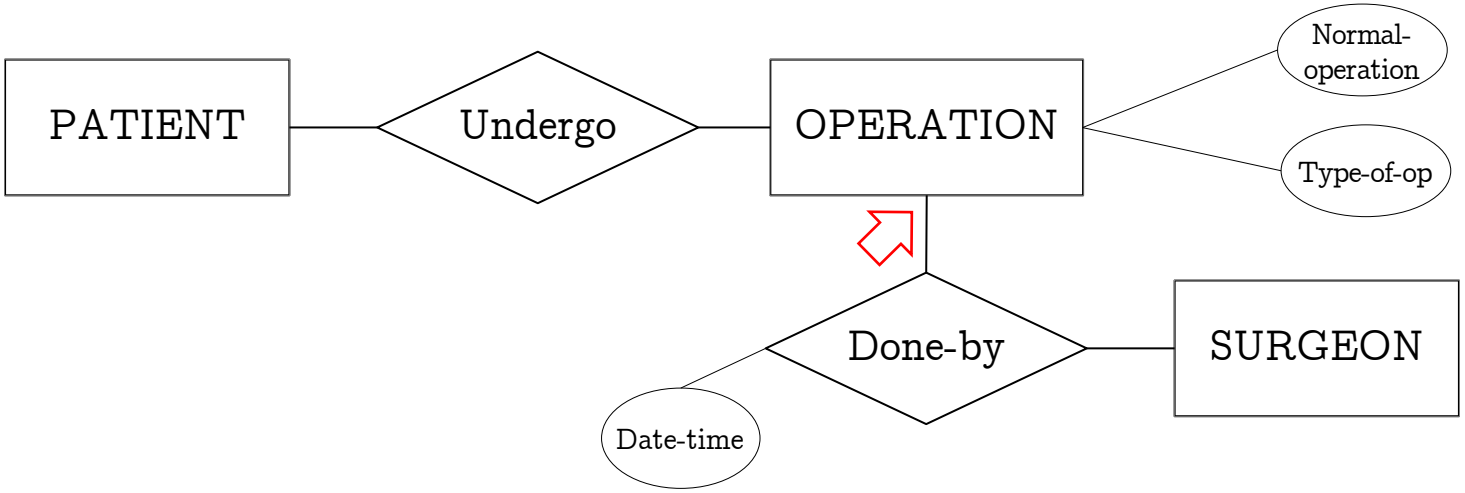

- During his/her hospitalisation, a patient may undergo surgical treatment, whereby multiple operations are possible. Complex procedures might require multiple surgeons.

# Question 1



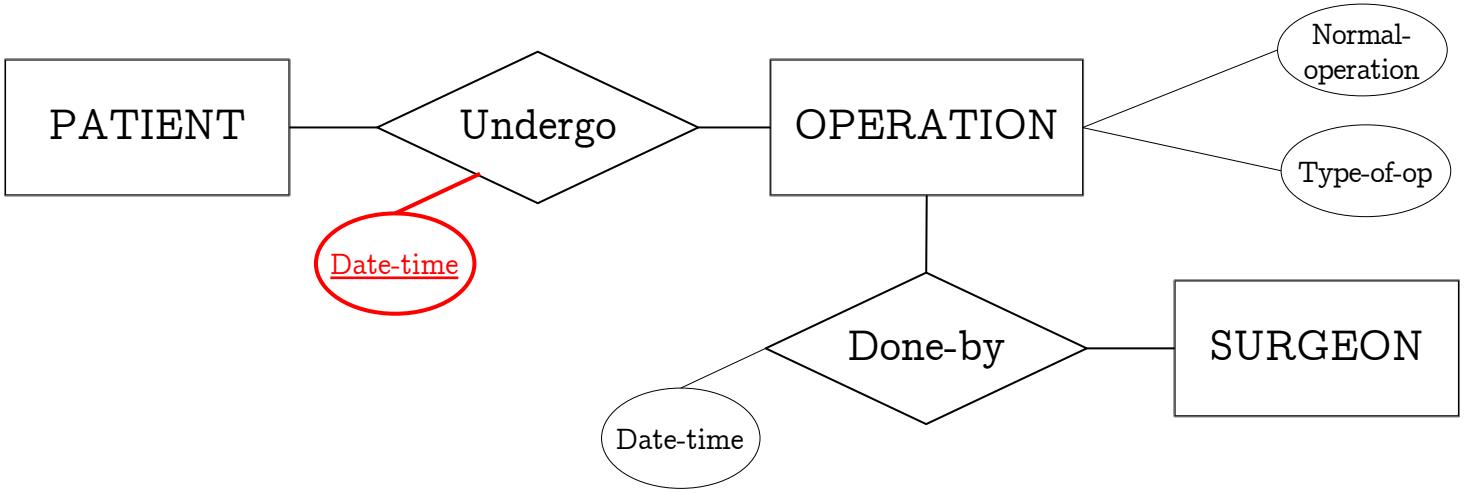

- An operation is characterised by the type of operation and the normal duration. The date / time of a particular operation has to be recorded.

# Question 1



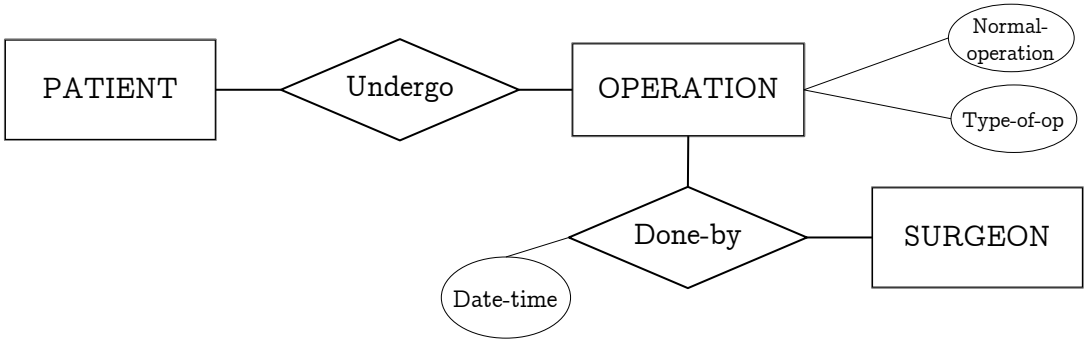

- Every surgeon has been involved in at least one operation but not every operation type has been carried out in this hospital.

# Question 1




- Problem 1:
- When a patient undergoes an operation, there is date/time when it is performed.
- If this is included as attribute of Undergo, then there are two Date-time's.
- Repetition of information in ER diagram is not good.

# Question 1




**Undergo**

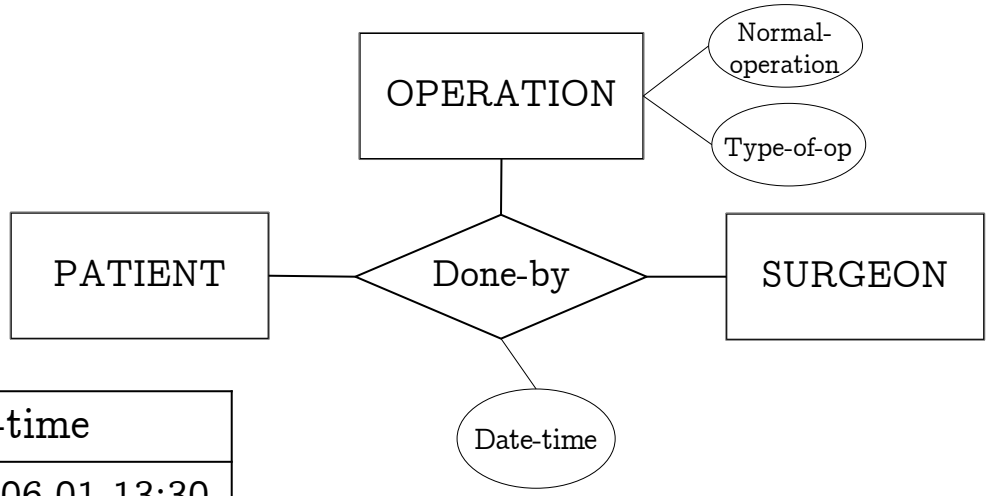
| Patient | Operation |
|---|---|
| John | Heart-op |
| Mary | Heart-op |
| Steve | Heart-op |

**Done-by**

| Operation | Surgeon | Date-time |
|---|---|---|
| Heart-op | Wang | 2020.06.01 13:30 |
| Heart-op | Chan | 2020.06.01 13:30 |
| Heart-op | Lee | 2020.06.01 13:30 |

- Problem 2:
- Consider the Undergo and Done-by tables of records.
- How do we know which surgeon operate on which patients?

# Question 1

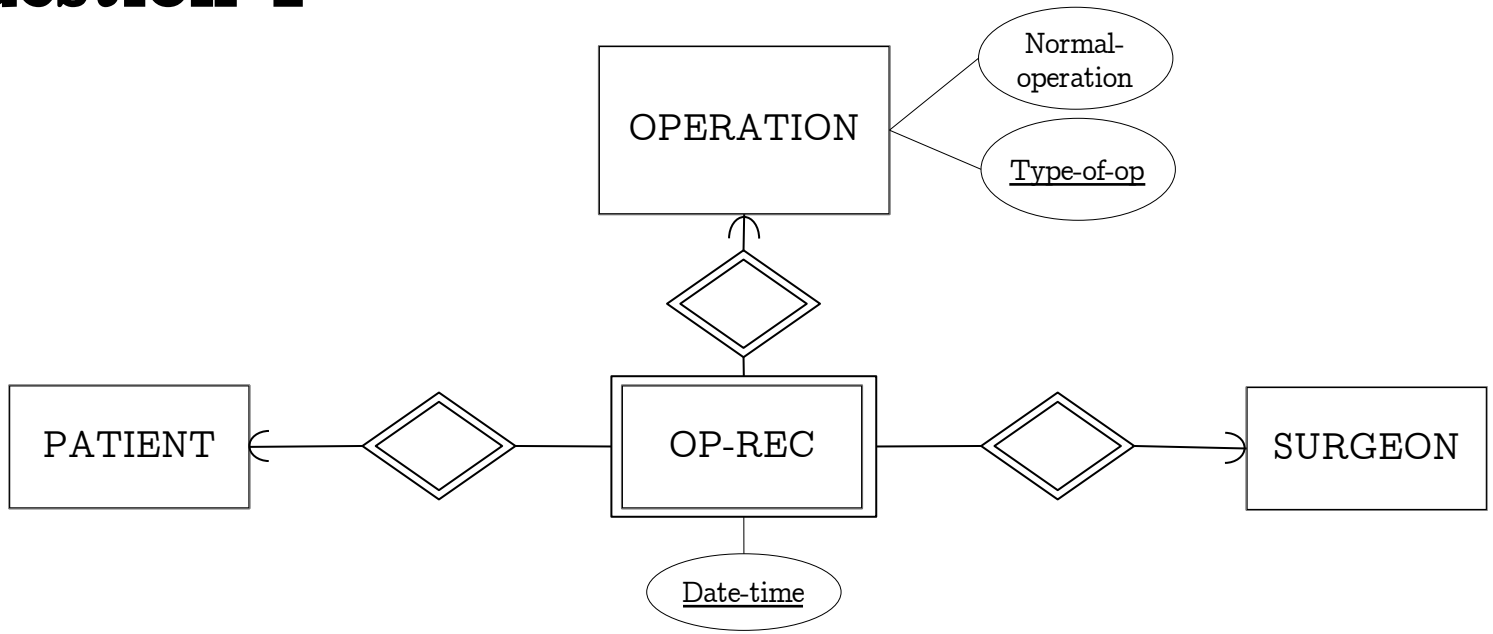



**Done-by**

| Patient | Operation | Surgeon | Date-time |
|---|---|---|---|
| John | Heart-op | Wang | 2020.06.01 13:30 |
| John | Heart-op | Chan | 2020.06.01 13:30 |
| Mary | Heart-op | Lee | 2020.06.01 13:30 |

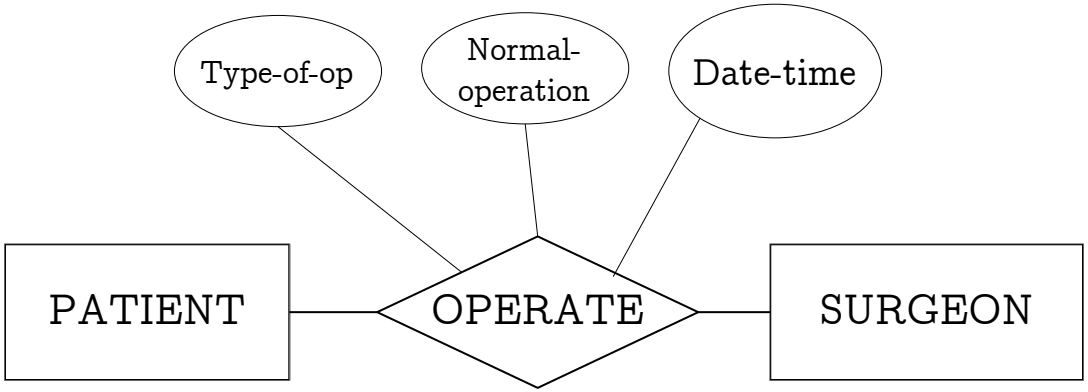
- Problem 2:
- To resolve Problem 2, we can use a 3-way relationship.

# Question 1



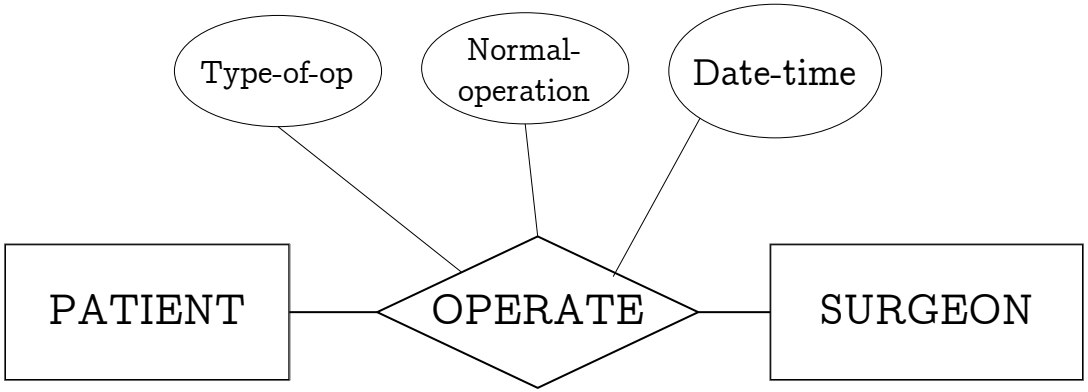

- Always convert 3-way relationship to 3 binary relationships
- These relationships are weak relationships

# Question 1




- Problem 3:
- Can't capture operations that are not performed (as given in text).

# Question 1



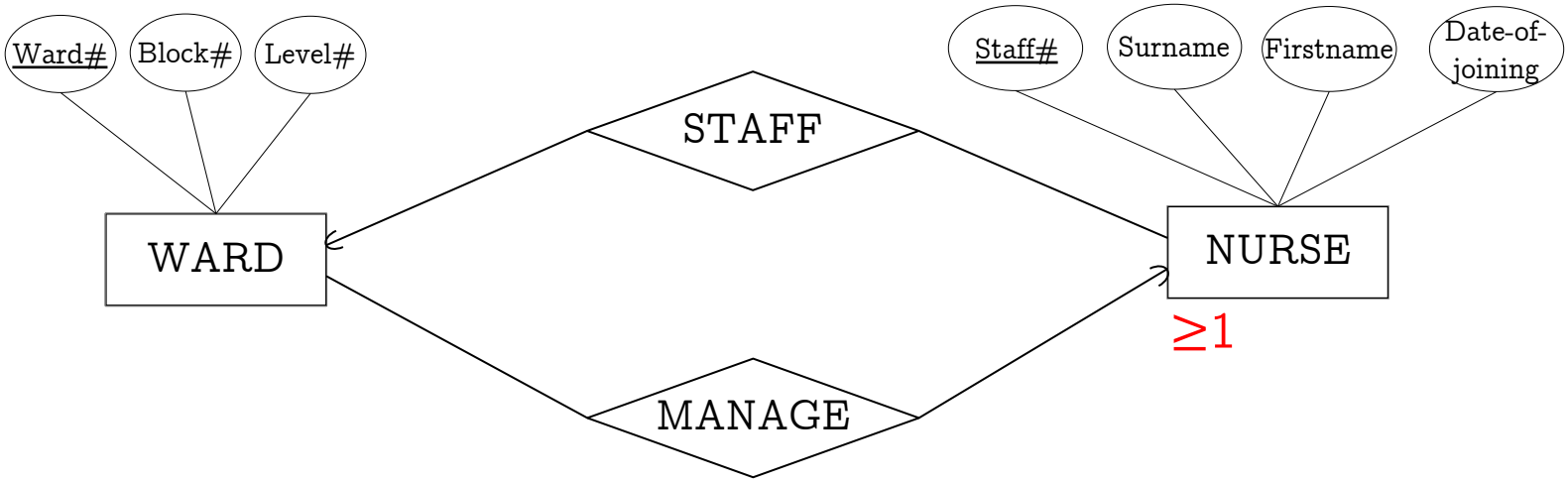

- Problem 4:

- If “heart surgery” is performed many times, “type” and “duration” will have the same values and repeated many times.

# Question 1

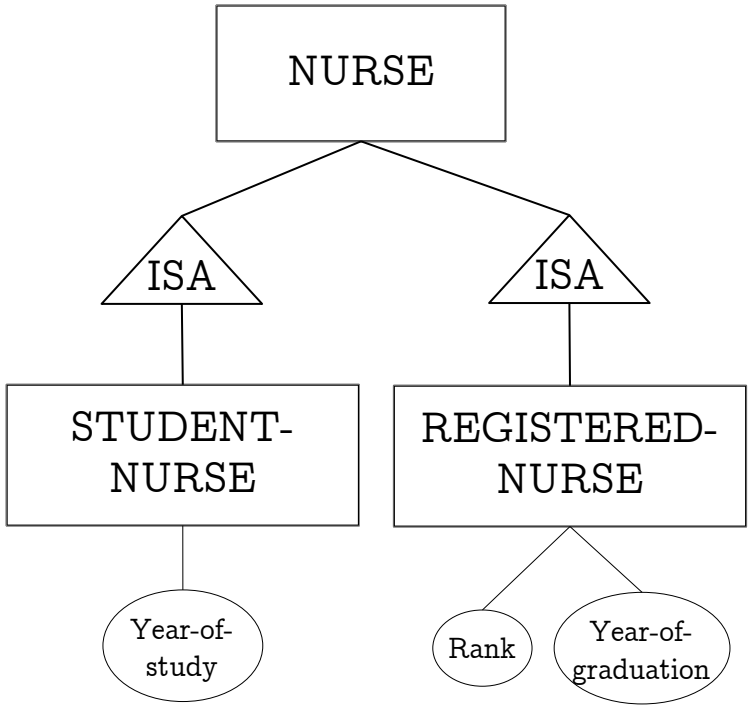
- Information about nurses includes staff#, first name, surname, and date of joining.
- Every nurse is assigned to a ward. A ward is staffed by many nurses. However, a ward is managed by only one nurse. Not every nurse manages a ward.
- Nurses are designated as student nurses or registered nurses. The particular year of study for student nurses and the year of graduation and rank for registered nurses are also stored.

# Question 1

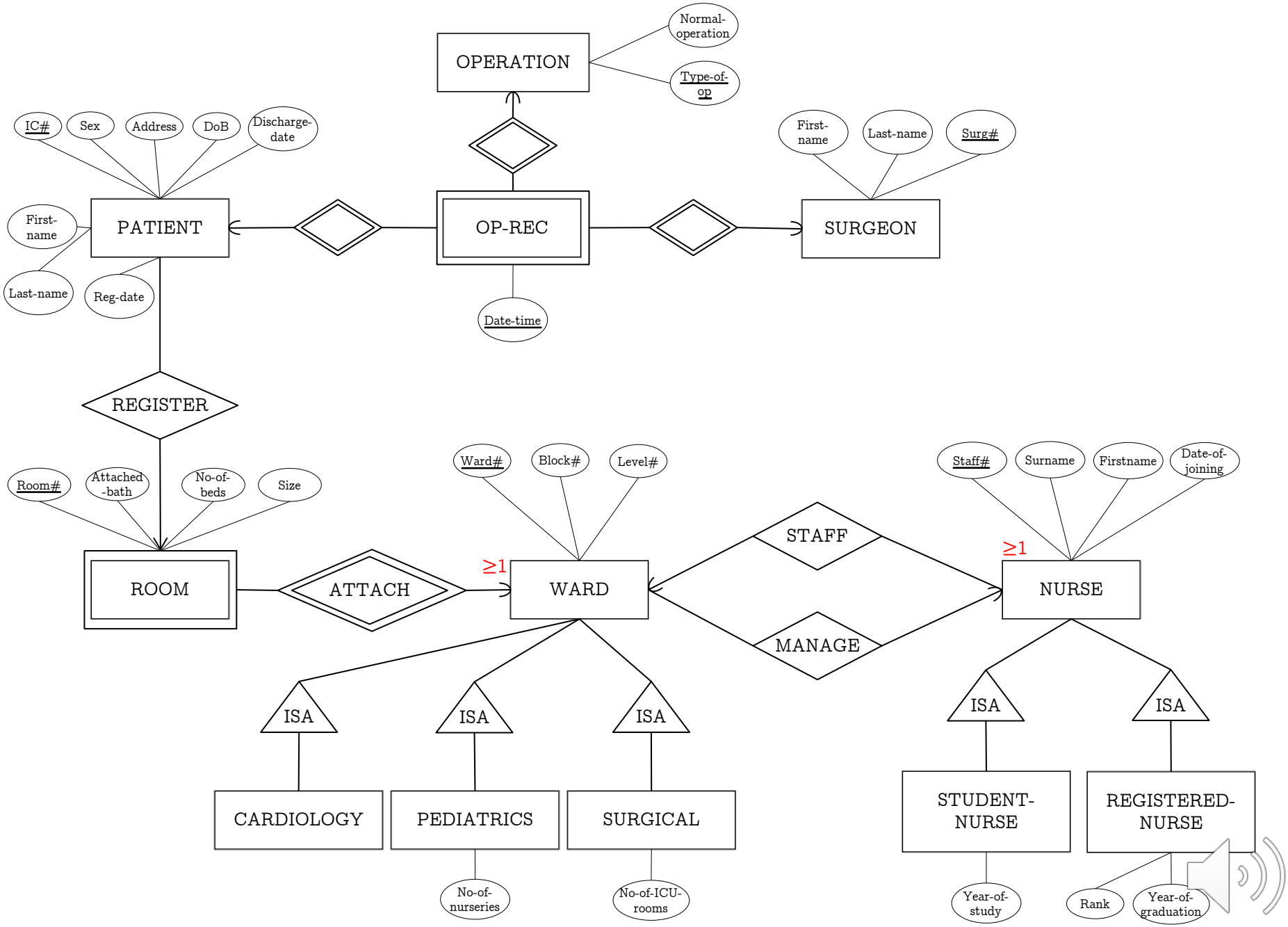



- Information about nurses includes staff#, first name, surname, and date of joining.
- Every nurse is assigned to a ward. A ward is staffed by many nurses. However, a ward is managed by only one nurse. Not every nurse manages a ward.

# Question 1




- Nurses are designated as student nurses or registered nurses. The particular year of study for student nurses and the year of graduation and rank for registered nurses are also stored.


OPERATION
Normal-operation
Type-of-op
IC#
Sex
Address
DoB
Discharge-date
First-name
Last-name
Surg#
PATIENT
OP-REC
SURGEON
First-name
Last-name
Reg-date
Date-time
REGISTER
Room#
Attached-bath
No-of-beds
Size
Ward#
Block#
Level#
Staff#
Surname
Firstname
Date-of-joining
ROOM
ATTACH
≥1
WARD
STAFF
MANAGE
≥1
NURSE
ISA
ISA
ISA
ISA
ISA
CARDIOLOGY
PEDIATRICS
SURGICAL
STUDENT-NURSE
REGISTERED-NURSE
No-of-nurseries
No-of-ICU-rooms
Year-of-study
Rank
Year-of-graduation


Do the best
you can until you know better.
Then when you know better, do better.
~ Maya Angelou
ALL THE BEST

# Tutorial 2
# Functional Dependencies

## Classroom Exercise

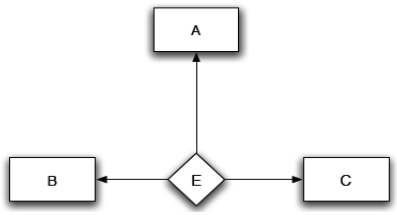
1. Translate the ER Diagram of Q1 in Tutorial 1 into a set of relations.

2. Consider the following relational schema:

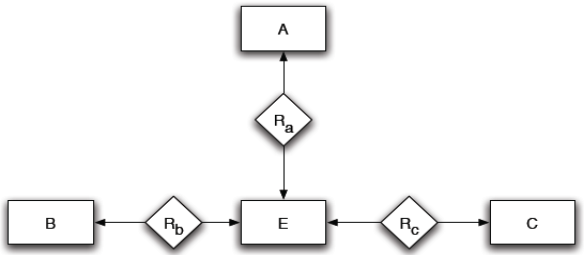
   USER(uid, name)
   SINGER(id, name)
   ALBUM(id, title, singerid)
   SONG(id, title, albumid)
   SING(singerid, songid)
   FOLLOW_USER(followeruid, followeeuid)
   RATE_SONG(uid, songid, rating)
   RATE_SINGER(uid, singerid, rating)
   PLAY_SONG(uid, songid, datetime)

   Construct an ER diagram that leads to the above schema.

3. The figure shows an attempt to represent a ternary relationship between three entities using 3 binary relationships (and one made-up entity). Show through an example why the 3 binary relationship representation is more general than the one ternary relationship representation.

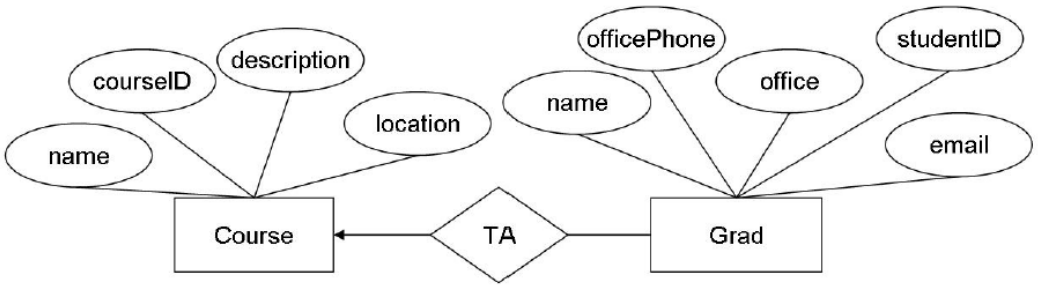



A ternary relationship




An attempt to represent it using 3 binary relationships

4. Consider the following ER diagram thath describes graduate students (Grad) and courses (Course) they serve as Teaching Assistants (TA).




(a) For each of the following statements, write a functional dependency (FD) that best captures the statement.

- The studentID of each graduate student uniquely identifies the student.
- No two offices have the same phone number (officePhone).
- No two courses have the same courseID.
- If two courses have the same course name, their course descriptions are the same.

(b) From the ER diagram and the set of FDs you listed above, can you derive new FDs? If no, explain why not. If yes, derive two non-trivial FDs.

(a) For each of the following statements, write a functional dependency (FD) that best captures the statement.

- The studentID of each graduate student uniquely identifies the student.
- No two offices have the same phone number (officePhone).
- No two courses have the same courseID.
- If two courses have the same course name, their course descriptions are the same.

(b) From the ER diagram and the set of FDs you listed above, can you derive new FDs? If no, explain why not. If yes, derive two non-trivial FDs.

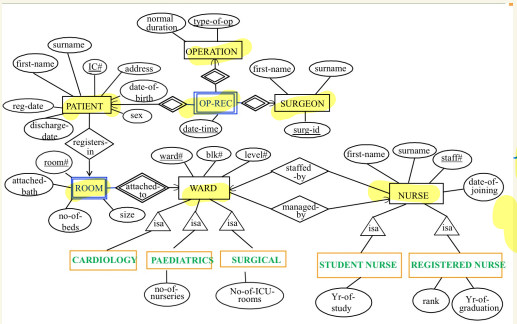
## Critical Thinking Exercise

5. Translate the ER Diagram of Q2 in Tutorial 1 into a set of relations.

TUT2

We = weak entity
M2O = many 2 one

1)




## Question 1: Relational Tables

- Surgeon (surg-id, first-name, surname)
- Operation (type-of-op, normal-duration)
- Patient (IC#, first-name, surname, address, date-of-birth, sex, reg-date, discharge-date, room#, ward#) – m2o
- Ward (ward#, blk#, level#, no-of-nurseries, no-of-ICU-rooms, manager) – m2o
- Nurse (staff#, first-name, surname, date-of-joining, wardStaff#) – m2o
- Op-Rec (IC#, surg-id, type-of-op, date-time) – w.e./m2o
- Room (room#, ward#, attached-bath, no-of-beds) – w.e./m2o

Surgeon (surg-id, FN, surname)
Operation (type-of-op, N duration)
Patient (

- Subclass relationships:
- Student_Nurse(staff#, Yr-of-study)
- Registered_Nurse(staff#, rank, yr-of-graduation)
- Surgical_Ward(ward#, No_of_ICU_rooms)
- Pediatrics_Ward(ward#, No_of_nurseries)
- Cardiology_Ward(ward#, no_of_heart_equipment)

- Ward (ward#, blk#, level#, manager) – m2o
- Nurse (staff#, first-name, surname, date-of-joining, ward#) – m2o

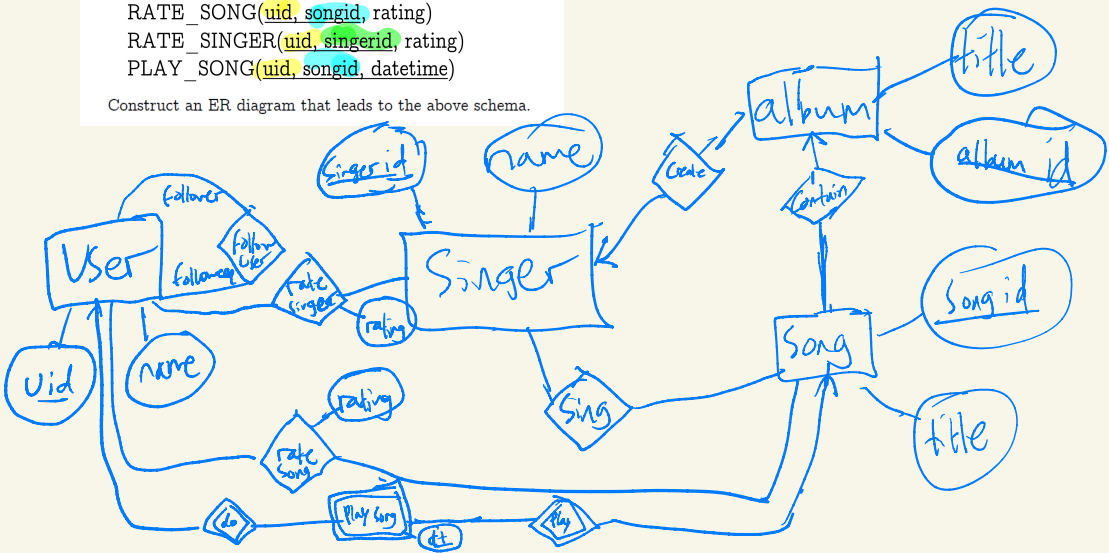
2)

## Question 2

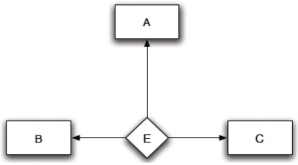
2. Consider the following relational schema:

USER(uid, name)
SINGER(id, name)
ALBUM(id, title, singerid)
SONG(id, title, albumid)
SING(singerid, songid)
FOLLOW_USER(followeruid, followeeuid)
RATE_SONG(uid, songid, rating)
RATE_SINGER(uid, singerid, rating)
PLAY_SONG(uid, songid, datetime)

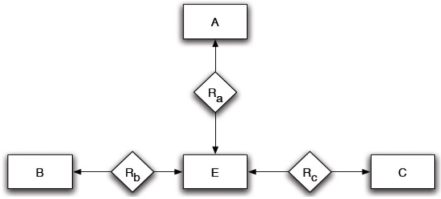
Construct an ER diagram that leads to the above schema.



3. The figure shows an attempt to represent a ternary relationship between three entities using 3 binary relationships (and one made-up entity). Show through an example why the 3 binary relationship representation is more general than the one ternary relationship representation.




A ternary relationship




An attempt to represent it using 3 binary relationships

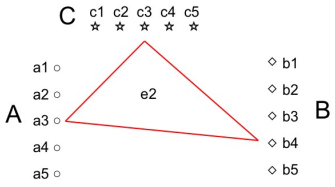
Why is a **3 binary relationship relationship representation** more general than **one ternary relationship representation**? (https://stackoverflow.com/questions/39254141/ternary-relationship-or-3-binary-relationship)

Because with 3 binary relationships each involved entity is related **separately** with each one of the others two.

Assumed that ternary relationship in essential only when you have a many-to-many-to-many relationship (otherwise you can rewrite it adding a relational entity linked with binary relationships to the others three), let's take an example:

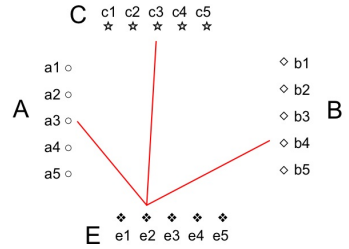
- Suppose that you have the entities A, B and C, and three relationships that link the entities "like a triangle". Now, suppose that set of tuples a3 is related to the sets b4 and c3.
- Using 3 binary relationships, it's not requested that b4 and c3 are related. for example, b4 can be related to c2, that can partially overlap c3 or be completely disjoined.
- With a ternary relationship, inste
- So you can see that with 3 binar combinations.

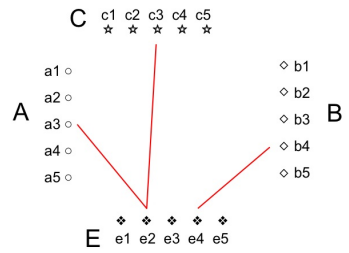
Ternary relationship



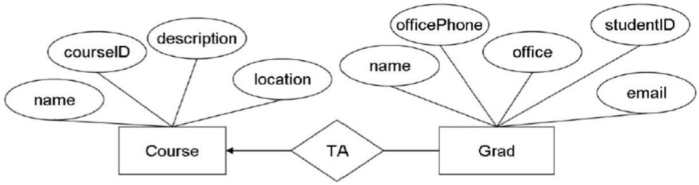

?

Three 2-binary relationship



4. Consider the following ER diagram thath describes graduate students (Grad) and courses (Course) they serve as Teaching Assistants (TA).




(a) For each of the following statements, write a functional dependency (FD) that best captures the statement.

- The studentID of each graduate student uniquely identifies the student.
- No two offices have the same phone number (officePhone).
- No two courses have the same courseID.
- If two courses have the same course name, their course descriptions are the same.

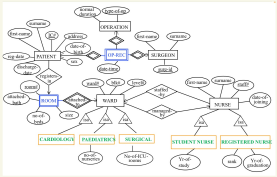
(b) From the ER diagram and the set of FDs you listed above, can you derive new FDs? If no, explain why not. If yes, derive two non-trivial FDs.

a) StudentID → Grad(name, officePhone, office, email)
officePhone → office
courseID → Courses(name, desc, instr)
name(Course) → course description

b) StudentID → courseID (from m2o TA)
StudentID → Courses(name, desc, instr)

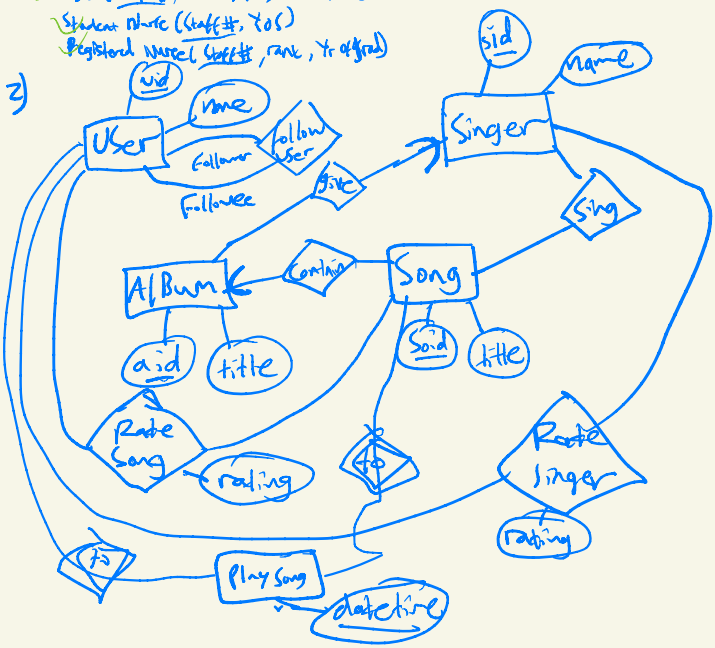
TUT 2

1)



operation ( type-of-op, Normal duration )
surgeon ( staff-id, FN, Surname)
Patient ( IC#, Surname, FN, reg-date, DischargeDate, Sex, DoB, address, room#, ward #/ # )
OpTec ( tec-no, surg-id, type of op, IC#)
Room ( room#, attached bath, no Of beds, type, ward-id # )
Ward ( ward#, Wkrs, NP, N staff)
Cardiology ( ward # )
Psychiatric ( no-of-matrons, ward# )
Surgical ( no-of ICU , ward #)
Nurse ( staff #, ward#, FN, Surname, DOJ)
Student Nurse ( staff #, YOS )
Registered Nurse ( staff #, rank , Yr of grad)

2)




4) b)

StudentID → name, description, location

student ID → courseID

a) StudentID → Grad

OfficePhone → Offices

courseID → Course

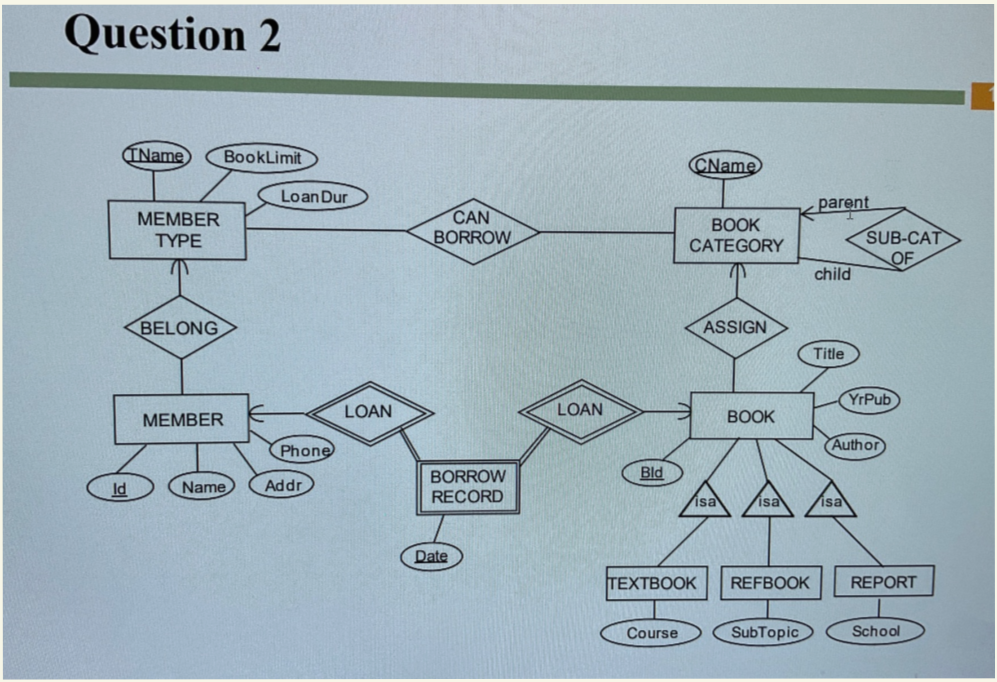
courseName → Course description

new FDs? If no, explain why not. If yes, derive two non-trivial FDs.

## Critical Thinking Exercise

5. Translate the ER Diagram of Q2 in Tutorial 1 into a set of relations.

## Question 2



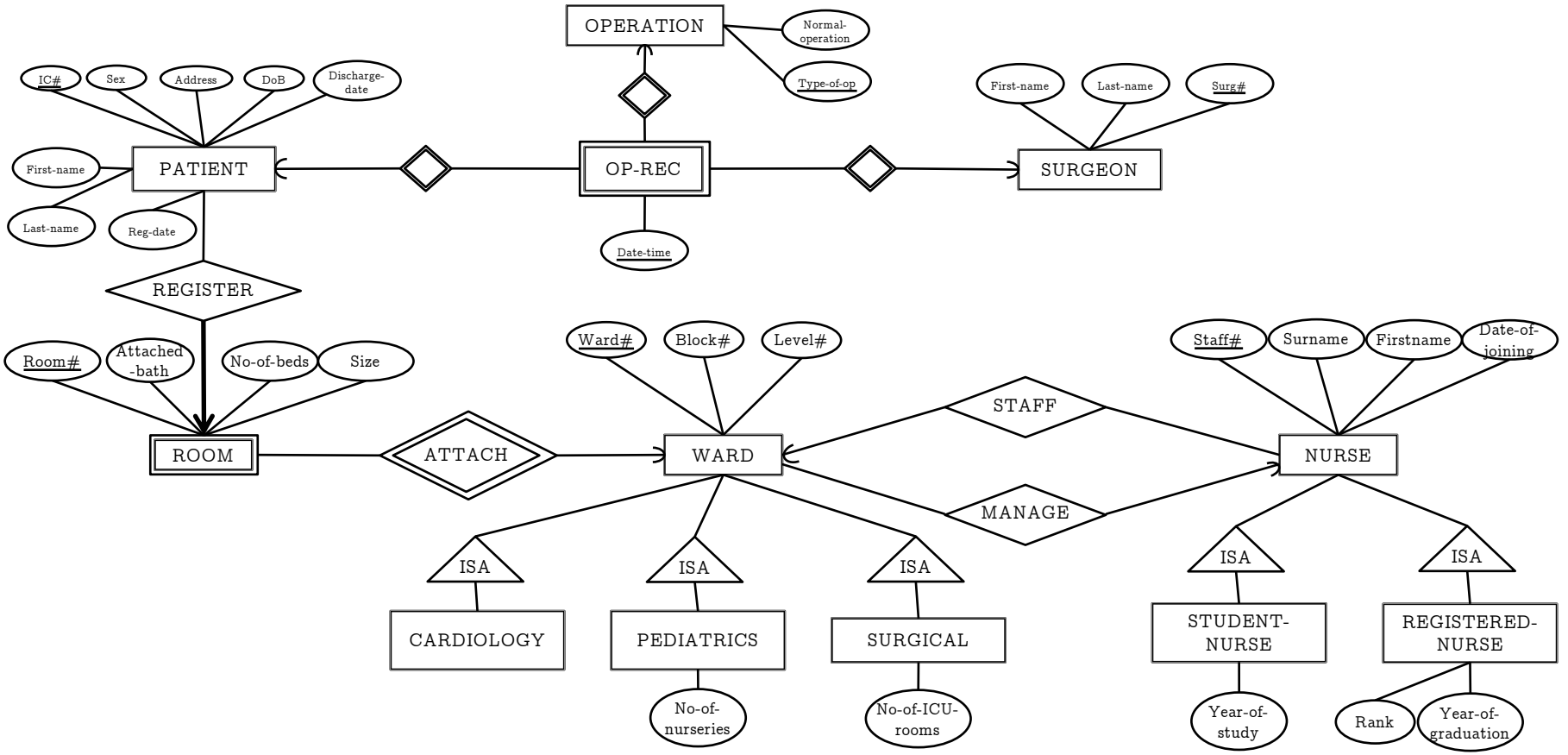

- memberType (TName, BookLimit, LoanDur)
- CanBorrow (TName, CName)
- BookCategory (CName)
- Sub-Cat of (Child-CName)
- Member (id, Name, Addr, phone, TName)
- BORRow Record (Date, id, Bid)
- Book (Bid, Title, YrPub, Author, CName)
- textbook (Course, Bid)
- Ref book (subtopic, Bid)
- Report (School, Bid)

# CZ2007 Tutorial 2: ER Diagram + FDs

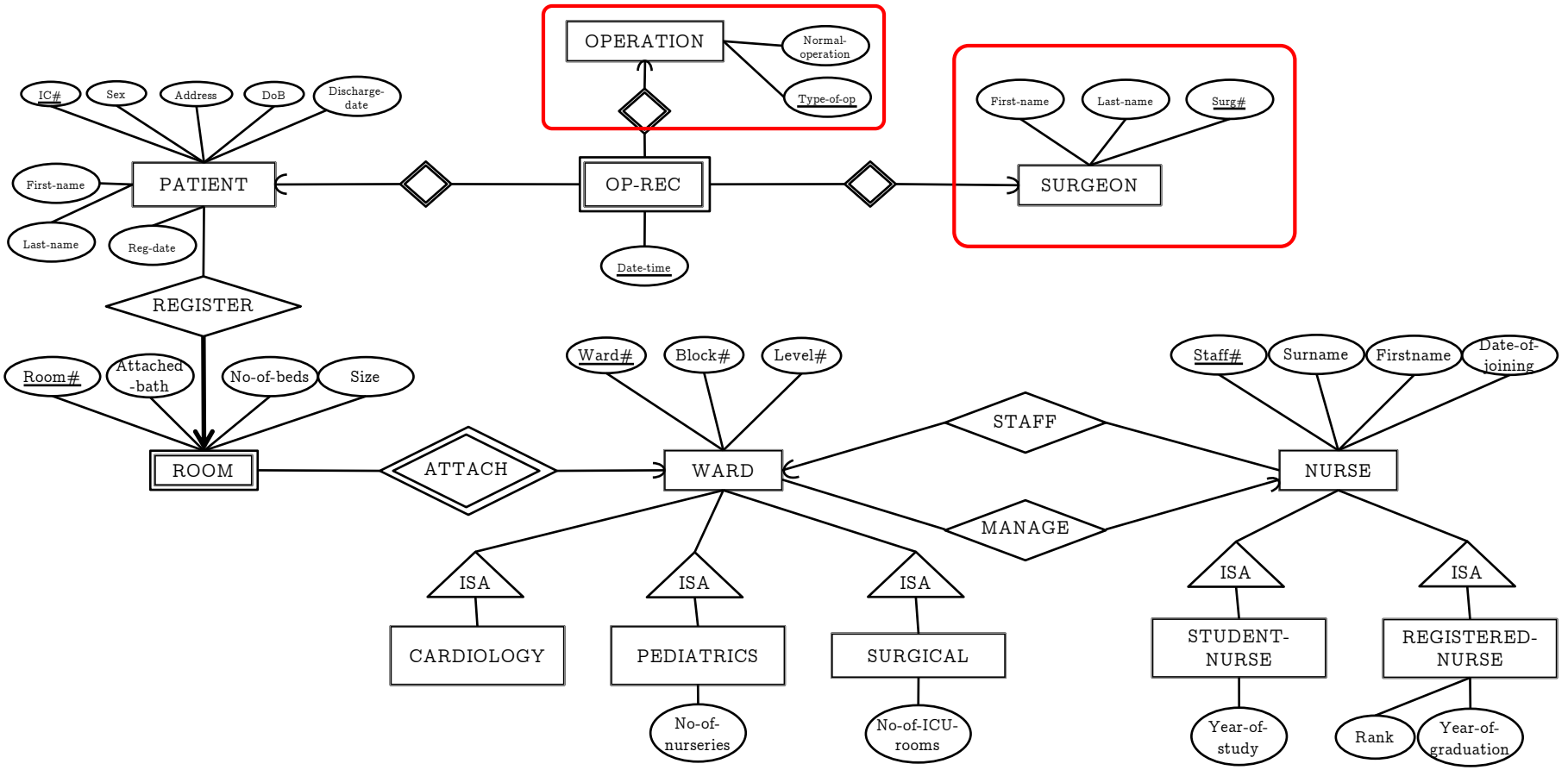
Week 4

# Question 1

1. Translate the ER Diagram of Q1 in Tutorial 1 into a set of relations.

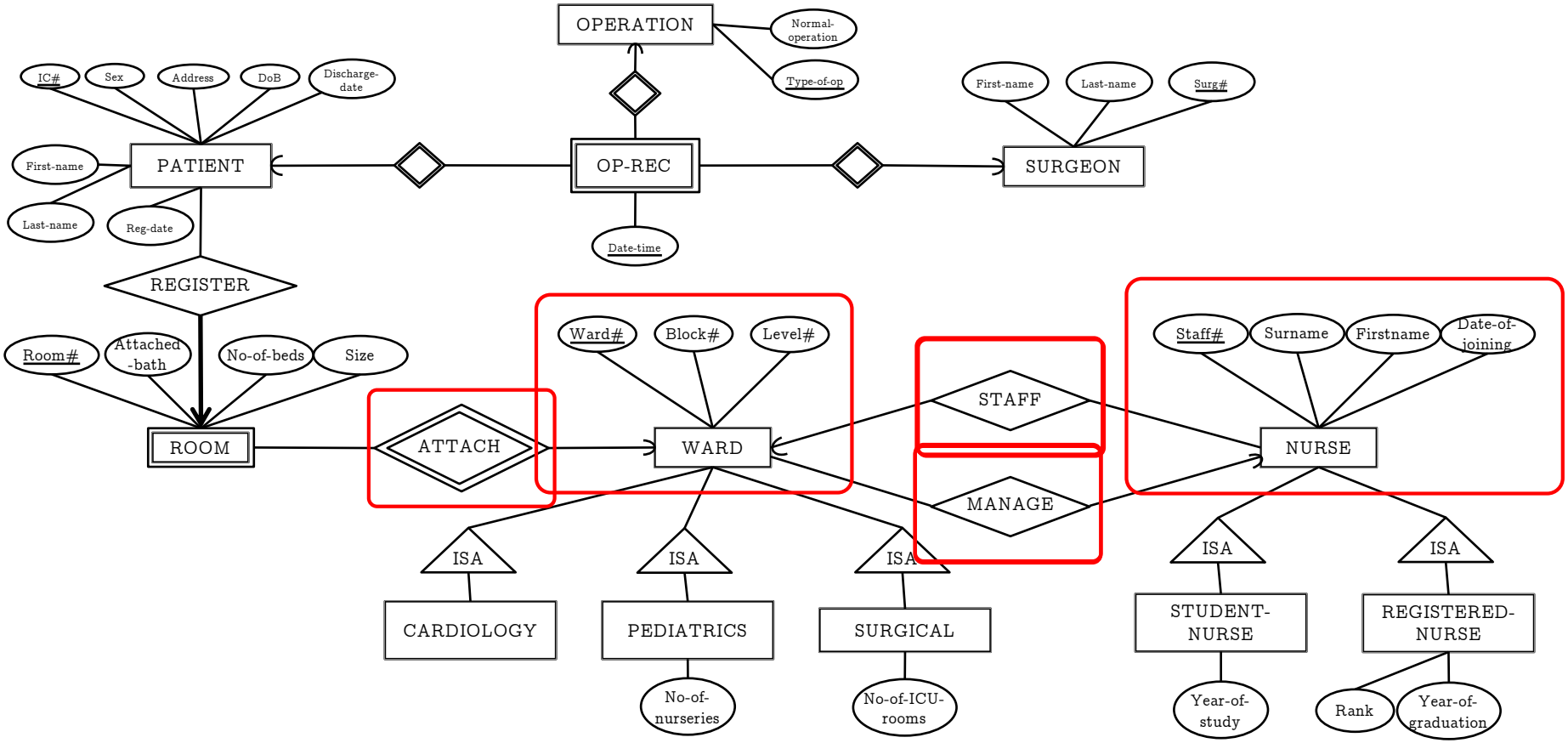


# Question 1



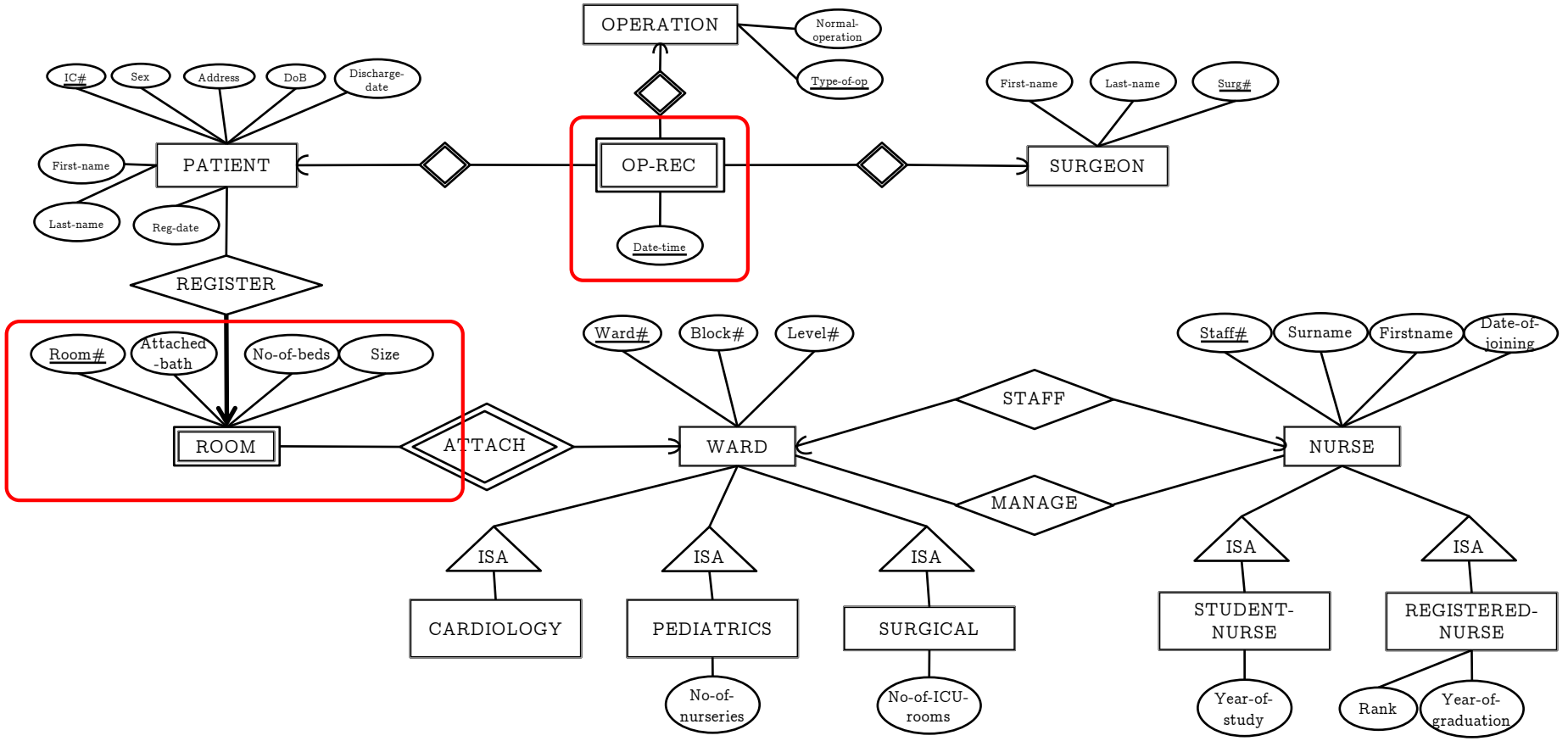

- Surgeon (surg#, first-name, surname)
- Operation (type-of-op, normal-duration)

# Question 1



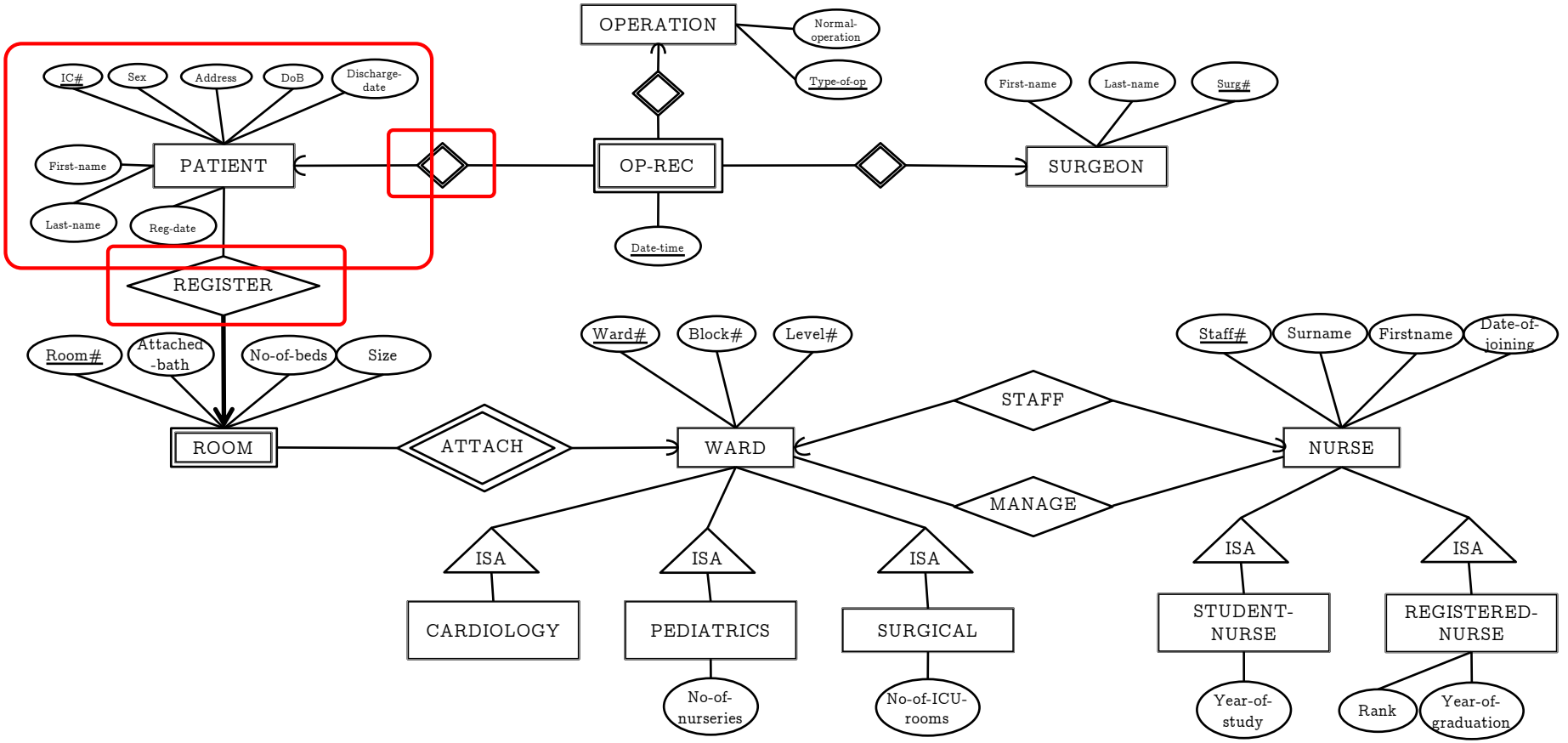

- Ward (<u>ward#</u>, blk#, level#, manager) – m2o
- Nurse (<u>staff#</u>, first-name, surname, date-of-joining, ward#) – m2o

# Question 1



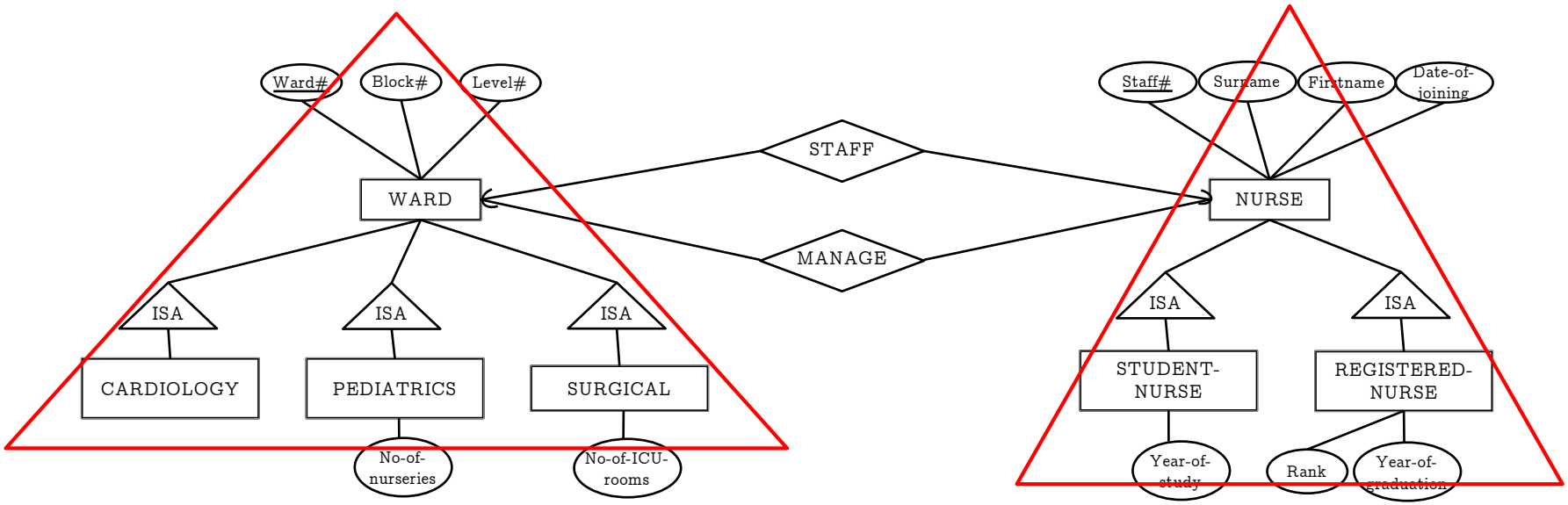

- Operate (<u>IC#, surg-id, type-of-op, date-time</u>) – 3 w.e./m2o
- Room (<u>room#, ward#</u>, attached-bath, no-of-beds) – w.e./m2o

# Question 1




- Patient (IC#, first-name, surname, address, date-of-birth, sex, reg-date, discharge-date, room#, ward#) – m2o

# Question 1



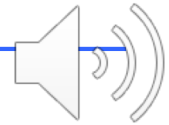

- Student_Nurse(staff#, year-of-study)
- Registered_Nurse(staff#, rank, year-of-graduation)
- Surgical_Ward(ward#, no-of-ICU-rooms)
- Pediatrics_Ward(ward#, no-of-nurseries)
- Cardiology_Ward(ward#, no-of-heart-equipment)

# Question 2

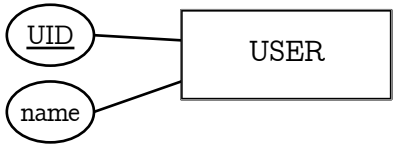
2. Consider the following relational schema:

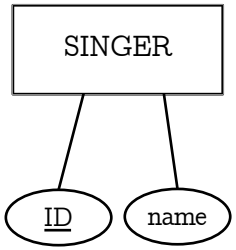
USER(<u>uid</u>, name)
SINGER(<u>id</u>, name)
ALBUM(<u>id</u>, title, singerid)
SONG(<u>id</u>, title, albumid)
SING(<u>singerid, songid</u>)
FOLLOW_USER(<u>followeruid, followeeuid</u>)
RATE_SONG(<u>uid, songid</u>, rating)
RATE_SINGER(<u>uid, singerid</u>, rating)
PLAY_SONG(<u>uid, songid, datetime</u>)

Construct an ER diagram that leads to the above schema.



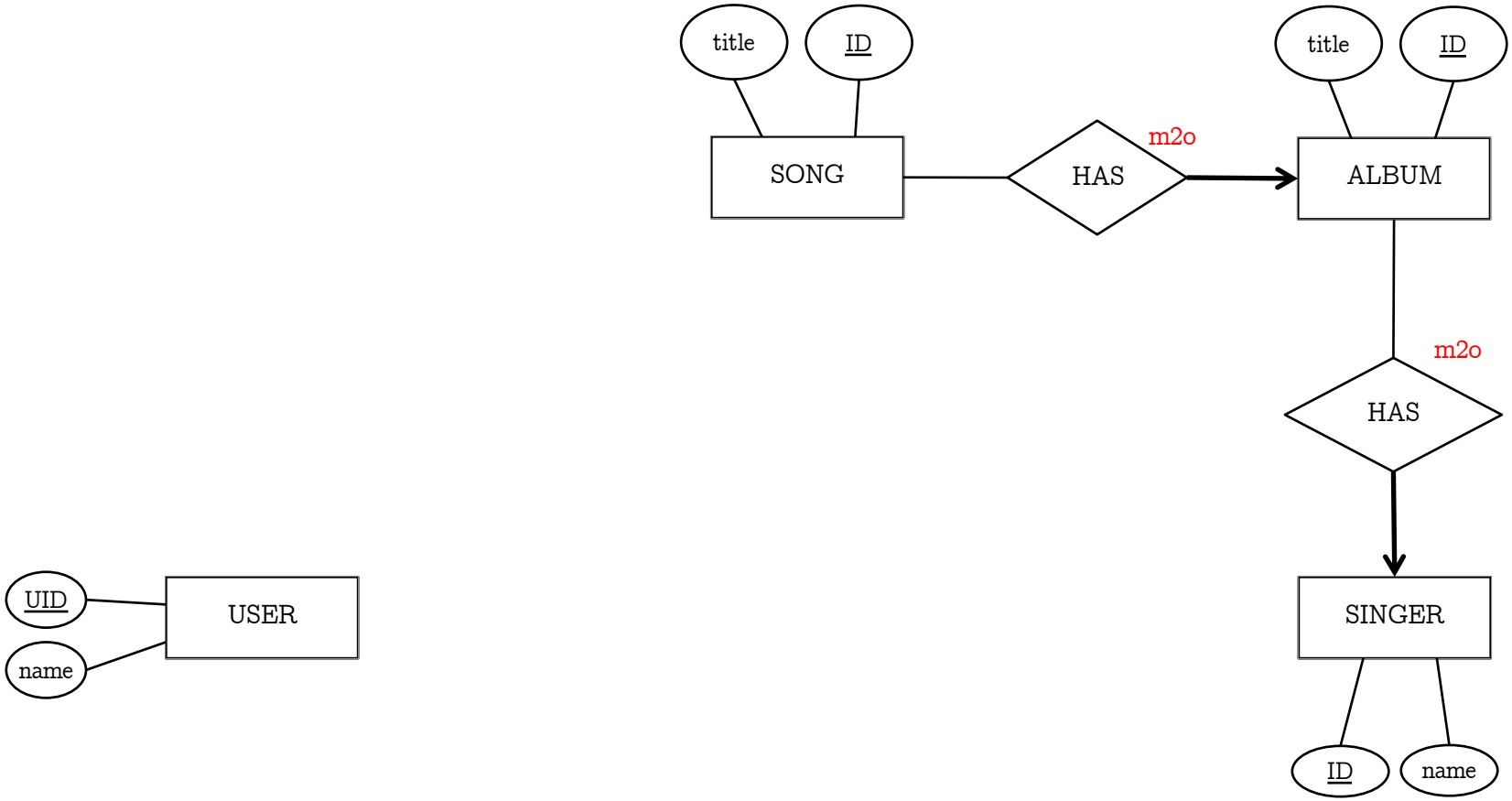
# Question 2



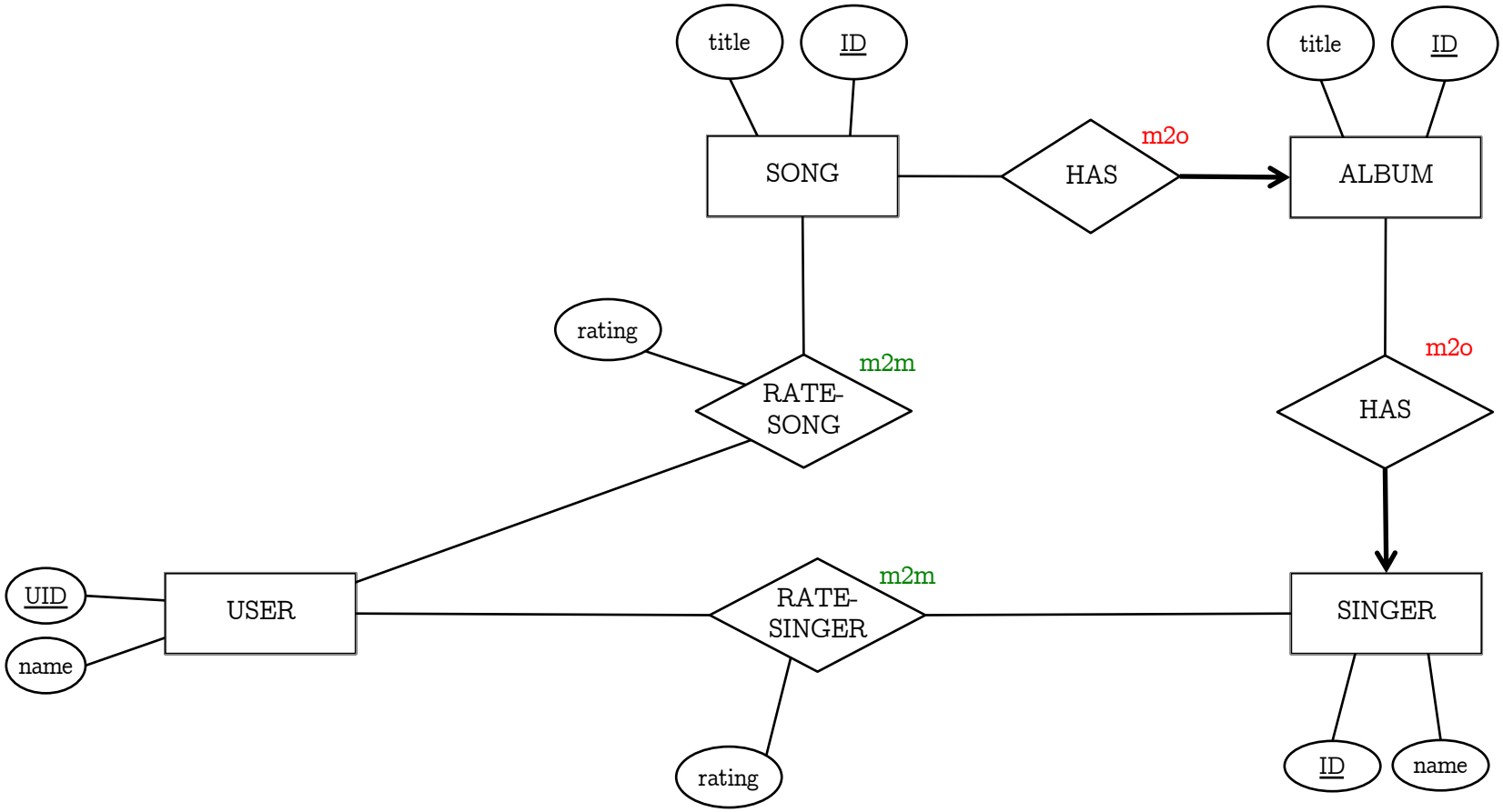

- USER(<u>uid</u>, name)
- SINGER(<u>id</u>, name)

# Question 2



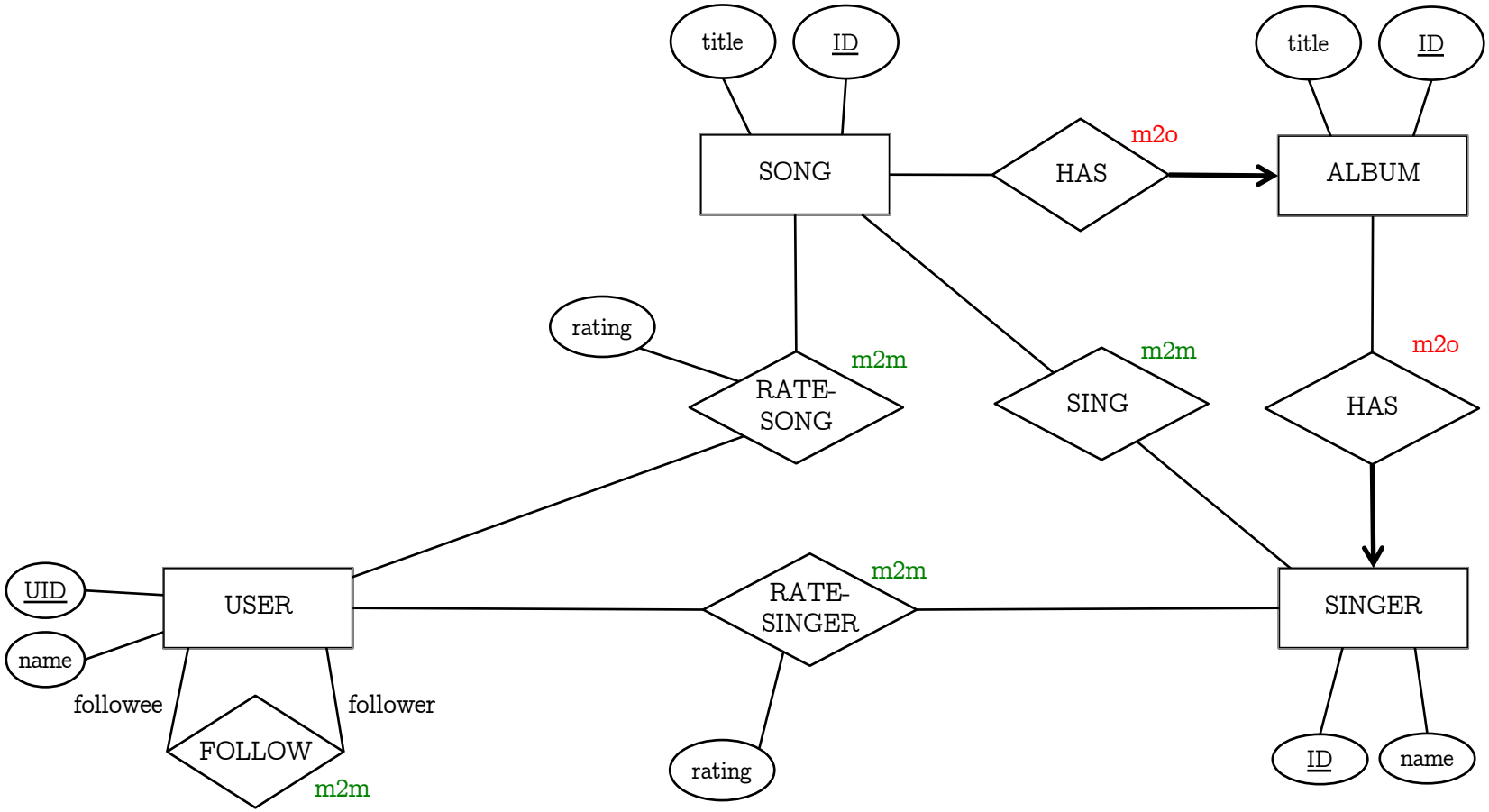

- ALBUM(<u>id</u>, title, singerid)
- SONG(<u>id</u>, title, albumid)

# Question 2



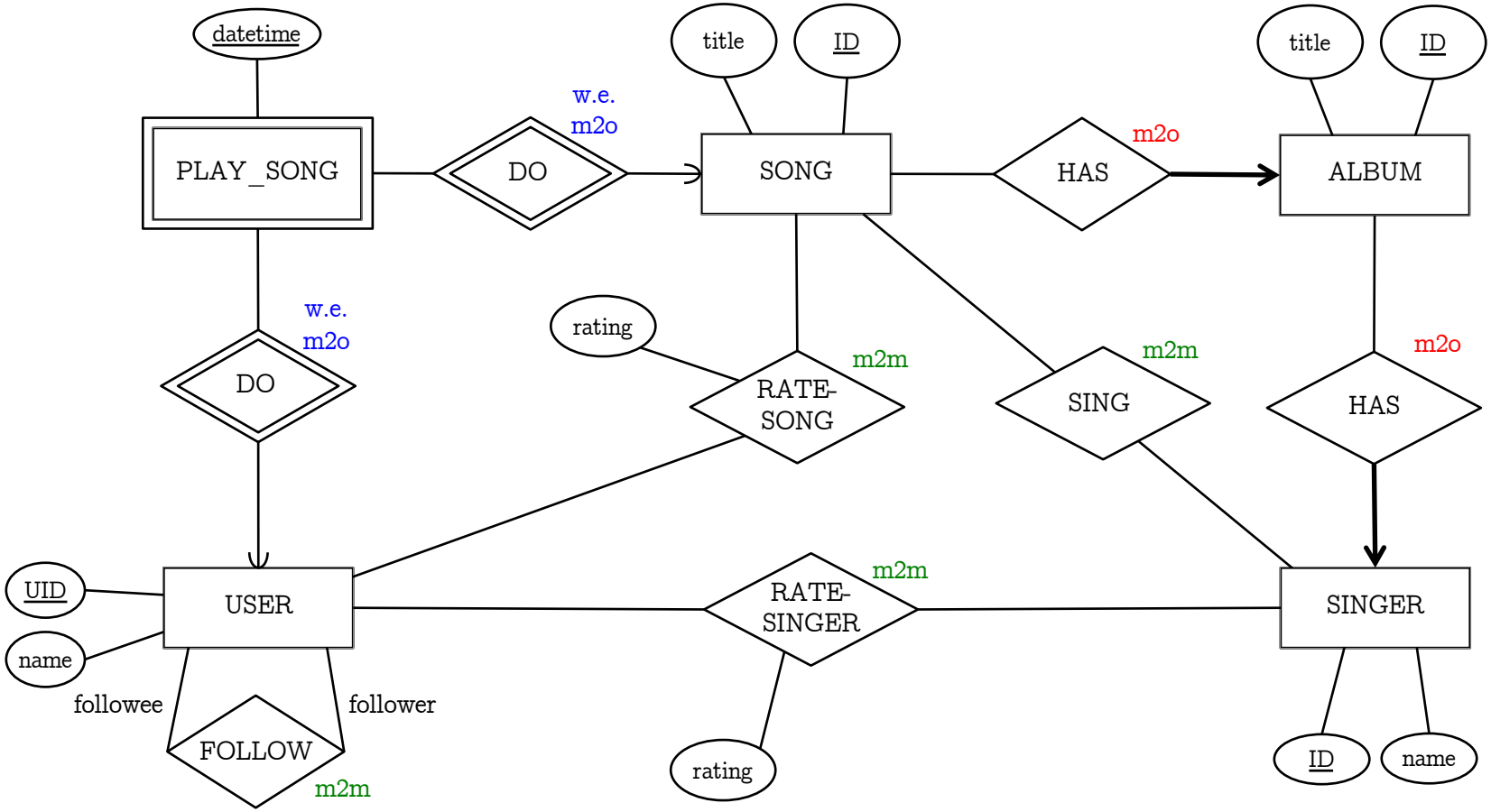

- RATE_SONG(uid, songid, rating)
- RATE_SINGER(uid, singerid, rating)

# Question 2

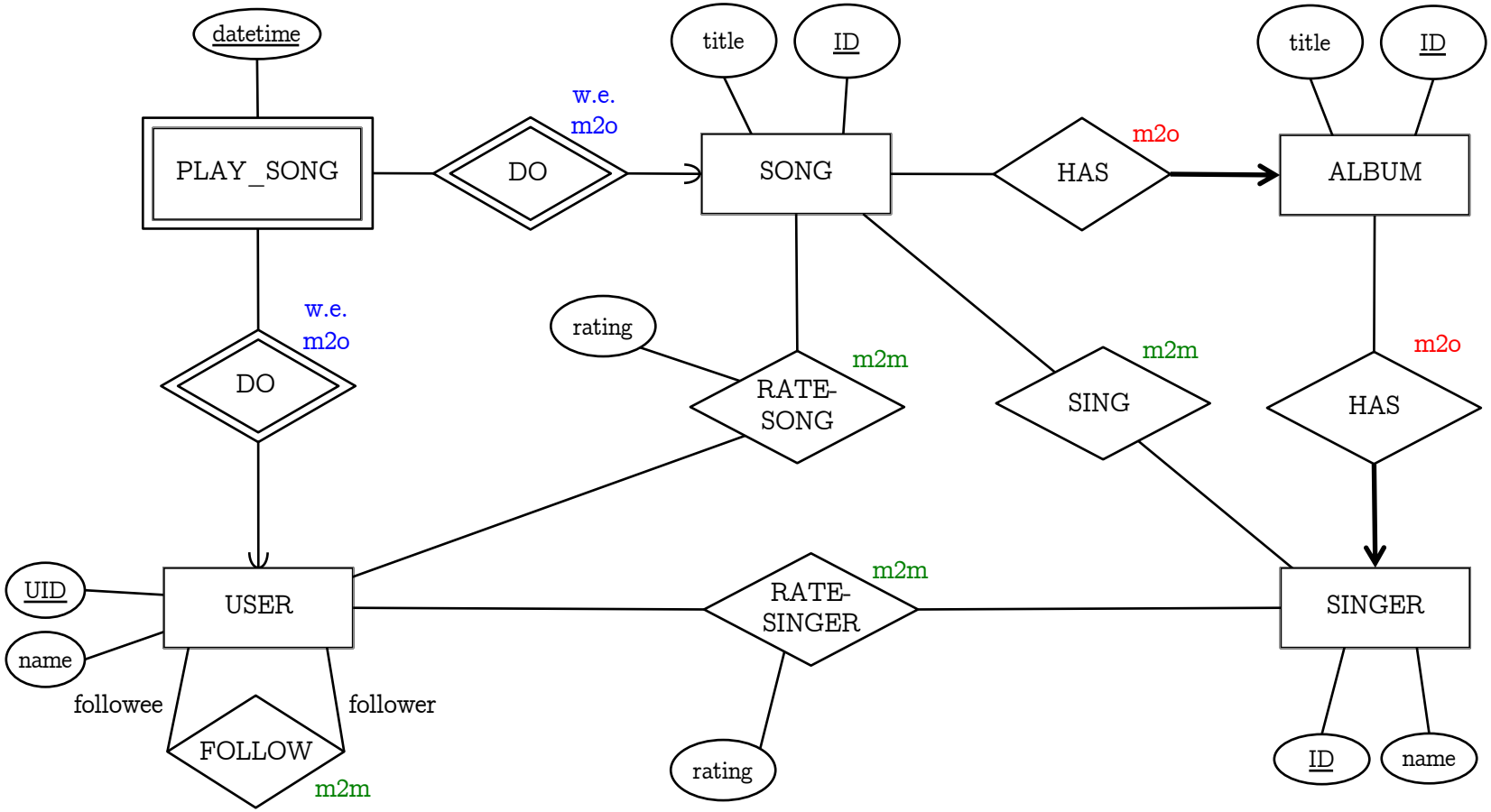



- SING(<u>singerid, songid</u>)
- FOLLOW(<u>followeruid, followeeuid</u>)

# Question 2

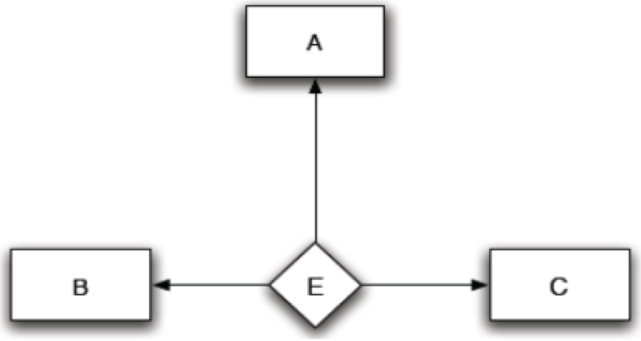



- PLAY_SONG(uid, songid, datetime)

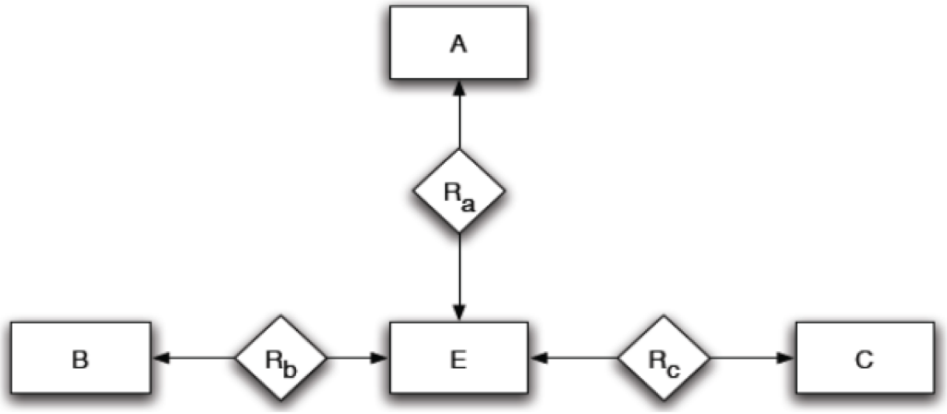

datetime
PLAY_SONG
w.e.
m2o
DO
title
ID
SONG
HAS
m2o
title
ID
ALBUM
w.e.
m2o
DO
rating
RATE-
SONG
m2m
SING
m2m
m2o
HAS
UID
USER
name
followee
follower
FOLLOW
m2m
RATE-
SINGER
m2m
rating
SINGER
ID
name

# Question 3

3. The figure shows an attempt to represent a ternary relationship between three entities using 3 binary relationships (and one made-up entity). Show through an example why the 3 binary relationship representation is more general than the one ternary relationship representation.

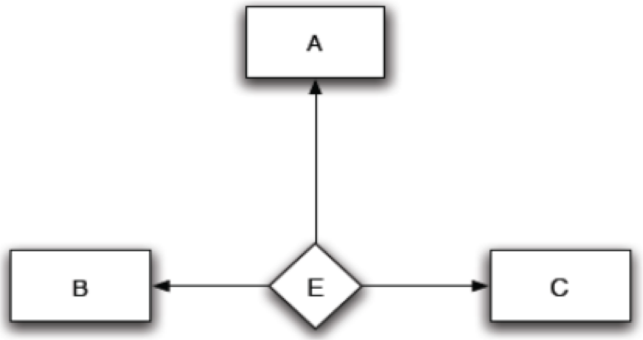



A ternary relationship

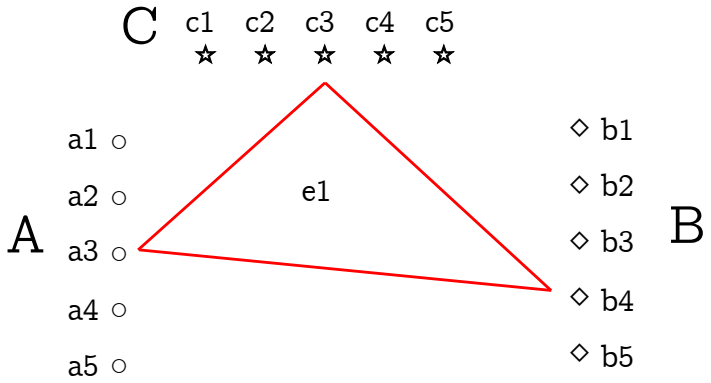



An attempt to represent it using 3 binary relationships

# Question 3



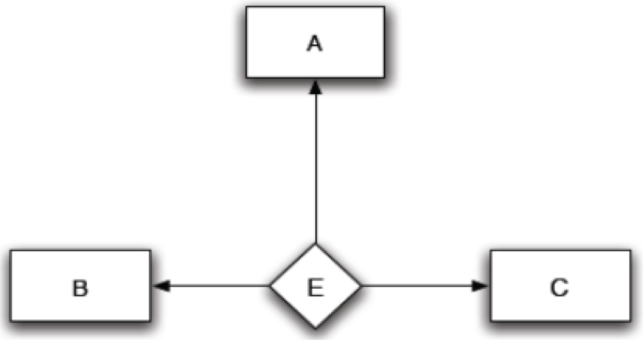

A ternary relationship



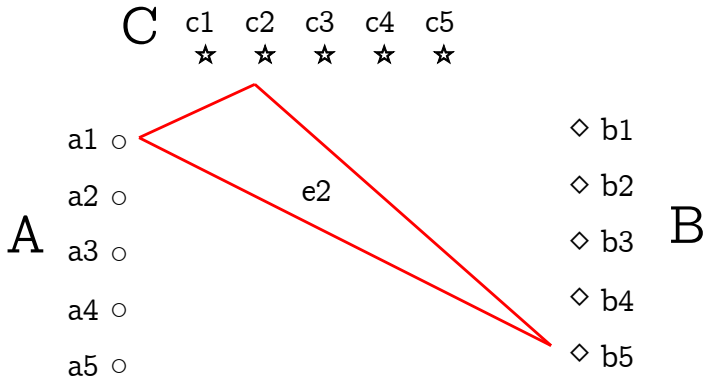

e1 is relationship instance

involving a3, b4, c3

# Question 3



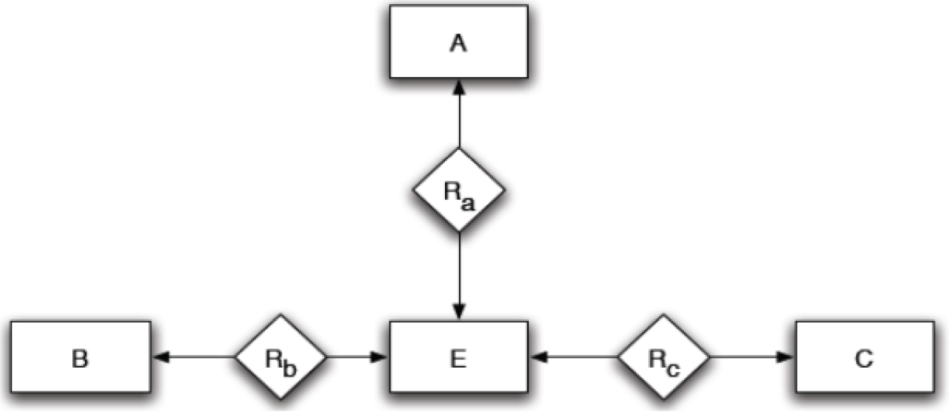

A ternary relationship

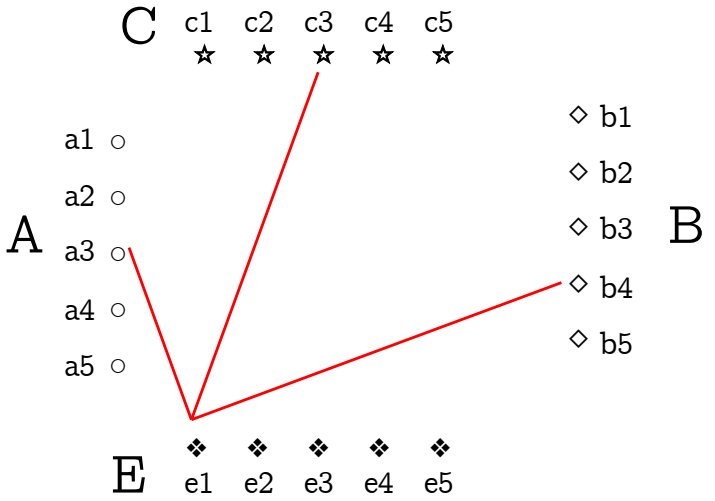



e2 is relationship instance

involving a1, b5, c2

# Question 3




e1 related to a3,

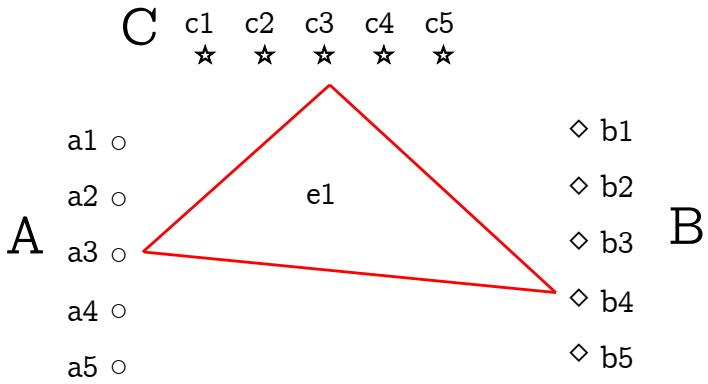
e1 related to b4,

e1 related to c3

# Question 3

e1 is relationship instance

involving a3, b4, c3

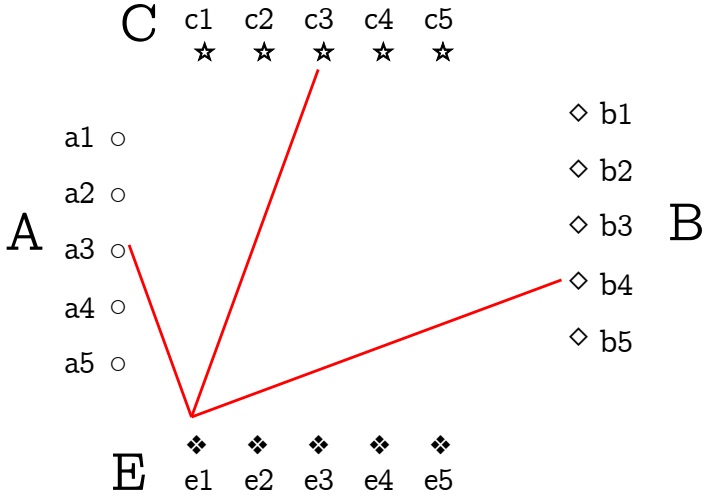



Single ternary relationship

e1 related to a3,

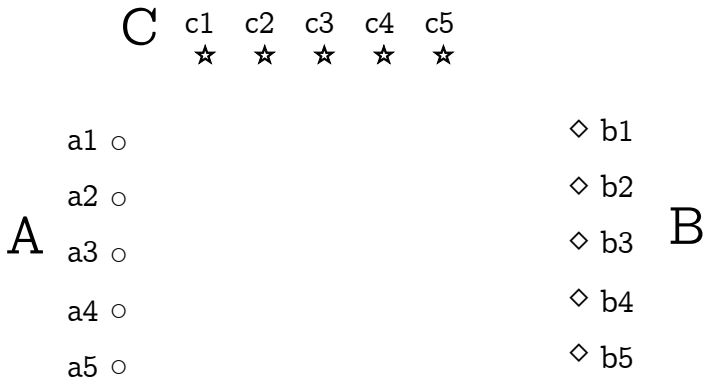
e1 related to b4,

e1 related to c3




3 binary relationship + 1 entity set

# Question 3

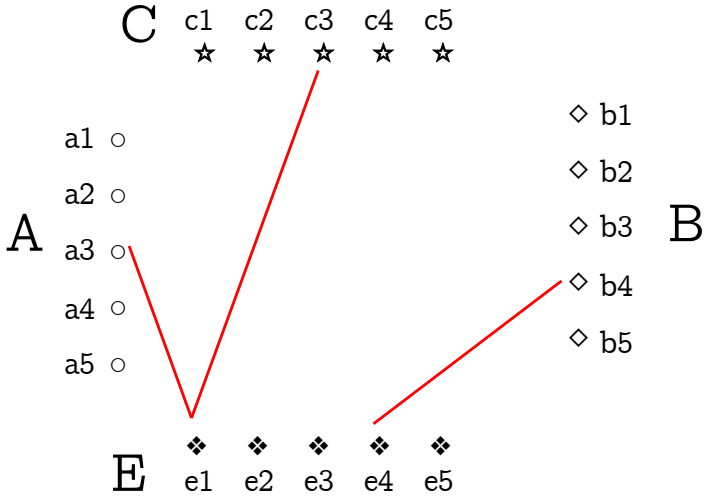



Single ternary relationship

e1 related to a3,

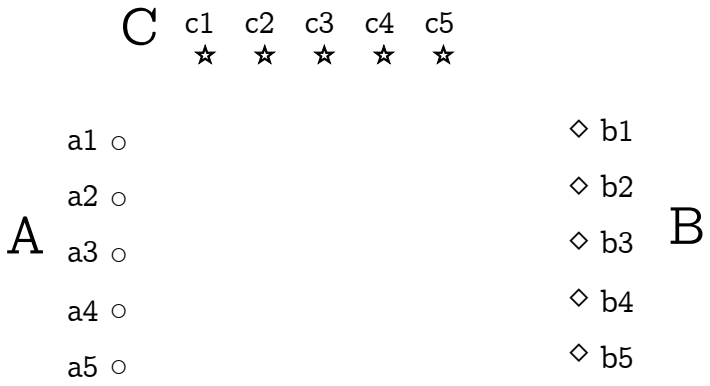
e4 related to b4,

e1 related to c3




3 binary relationship + 1 entity set

# Question 3

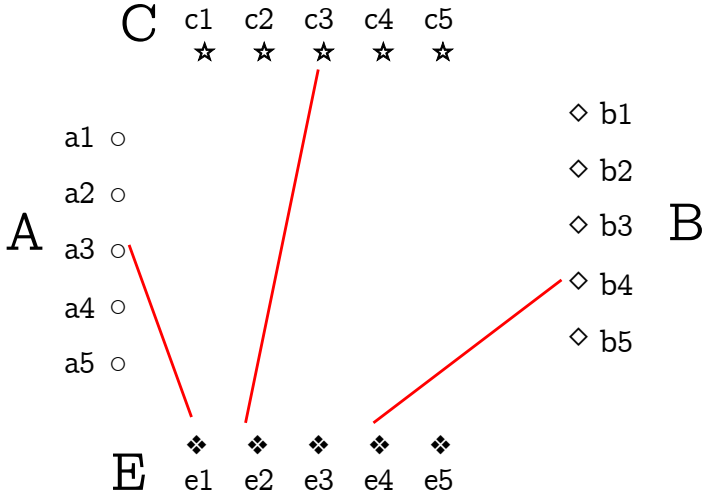



Single ternary relationship

e1 related to a3,

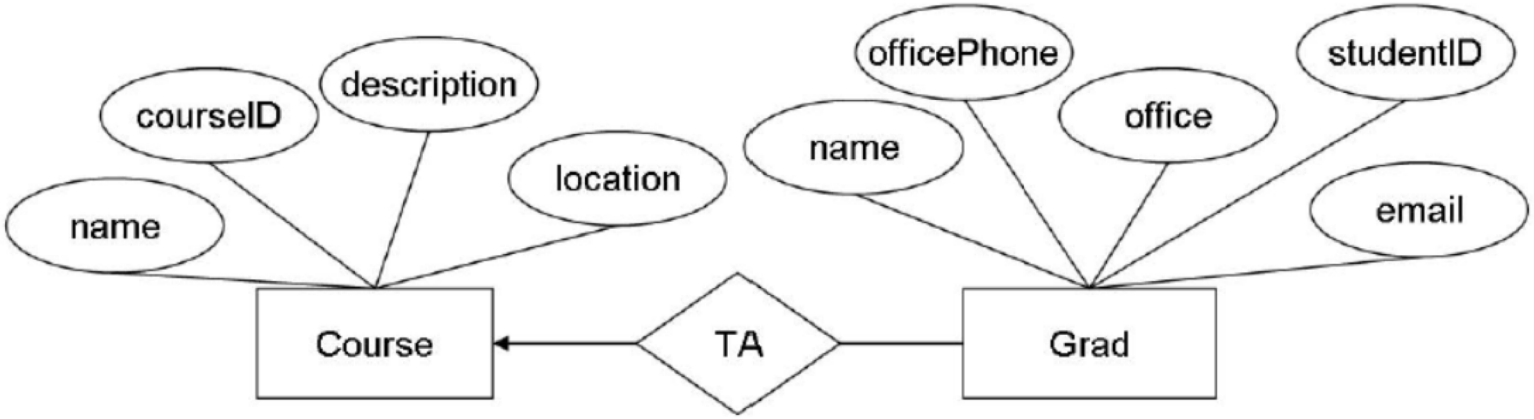
e4 related to b4,

e2 related to c3




3 binary relationship + 1 entity set

4. Consider the following ER diagram thath describes graduate students (Grad) and courses (Course) they serve as Teaching Assistants (TA).




(a) For each of the following statements, write a functional dependency (FD) that best captures the statement.

- The studentID of each graduate student uniquely identifies the student.
- No two offices have the same phone number (officePhone).
- No two courses have the same courseID.
- If two courses have the same course name, their course descriptions are the same.

(b) From the ER diagram and the set of FDs you listed above, can you derive new FDs? If no, explain why not. If yes, derive two non-trivial FDs.

# Question 4(a)

(a) For each of the following statements, write a functional dependency (FD) that best captures the statement.

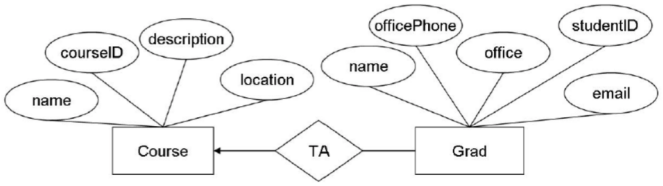
- The studentID of each graduate student uniquely identifies the student.
- No two offices have the same phone number (officePhone).
- No two courses have the same courseID.
- If two courses have the same course name, their course descriptions are the same.

- F1: studentID $\rightarrow$ officePhone, office, email, name
- F2: officePhone $\rightarrow$ Office
- F3: courseID $\rightarrow$ name, description, location
- F4: name $\rightarrow$ description

# Question 4(b)

(b) From the ER diagram and the set of FDs you listed above, can you derive new FDs? If no, explain why not. If yes, derive two non-trivial FDs.

- F1: studentID $\rightarrow$ officePhone, office, email, name
- F2: officePhone $\rightarrow$ Office
- F3: courseID $\rightarrow$ name, description, location
- F4: name $\rightarrow$ description




- F5: studentID $\rightarrow$ courseID. This is derived from m2o TA relationship.
- F6: studentID $\rightarrow$ name, description, location. This is derived from F3 and F5.


Winners are not people who never fail
but people who never quit.
Best of Luck

# Tutorial 2

# Functional Dependencies

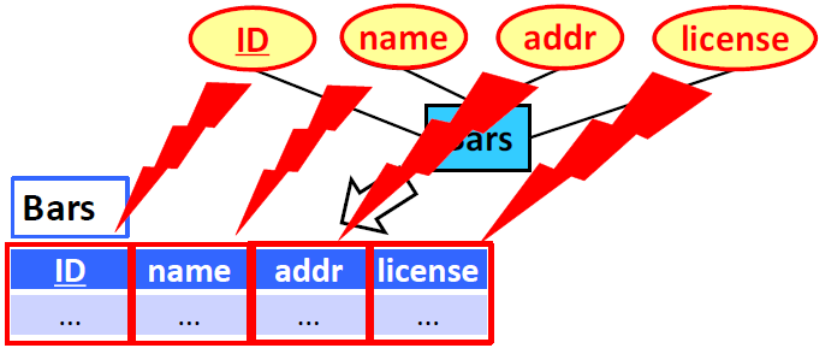
## CZ2007
## Introduction to Databases

# Tutorial 2

## ER Diagram → Relational Schema

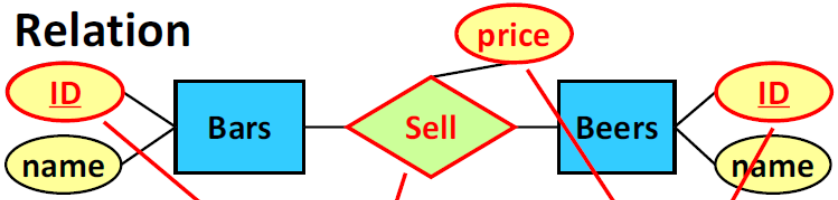
- General rules:
  - Each entity set  becomes a relation 
  - Each many-to-many relationship  becomes a relation 
- Special treatment needed for:
  - Weak entity sets 
  - Subclasses 
  - Many-to-one and one-to-one relationships 

## Entity Set → Relation




- Each entity set is converted into a relation that contains all its attributes
- Key of the relation = key of the entity set

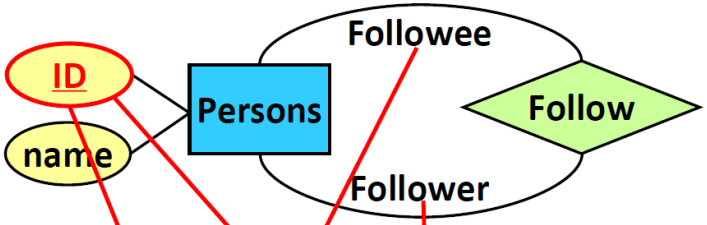
## Many-to-Many Relationship → Relation




- Converted into a relation that contains
  - all keys of the participating entity sets, and
  - the attributes of the relationship (if any)
- Key of relation = Keys of the participating entity sets

| Sell | Bars-ID | Beers-ID | price |
|---|---|---|---|
| | ... | ... | ... |

## Many-to-Many Relationship → Relation

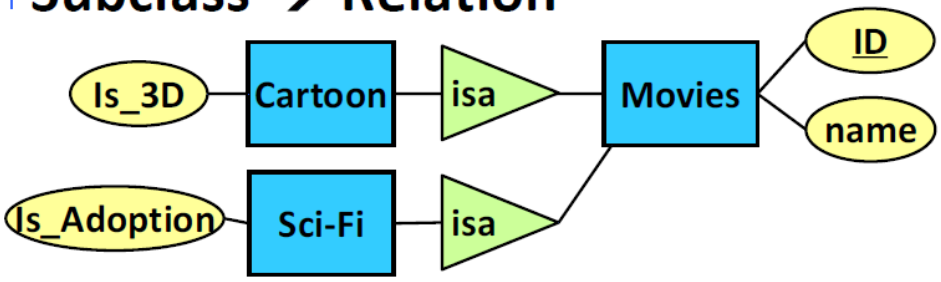



- If an entity is involved multiple times in a relationship
  - Its key will appear in the corresponding relation multiple times
  - The key is re-named according to the corresponding role

| Follow | Followee-ID | Follower-ID |
|---|---|---|
| | ... | ... |

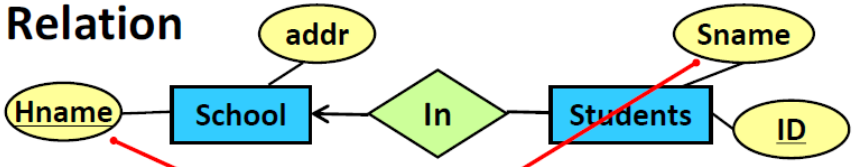
# Tutorial 2

## Subclass → Relation




- One relation for each entity set
  - Movies( ID, name )
  - Cartoon( ID, Is_3D )
  - Sci-Fi( ID, Is_Adoption )

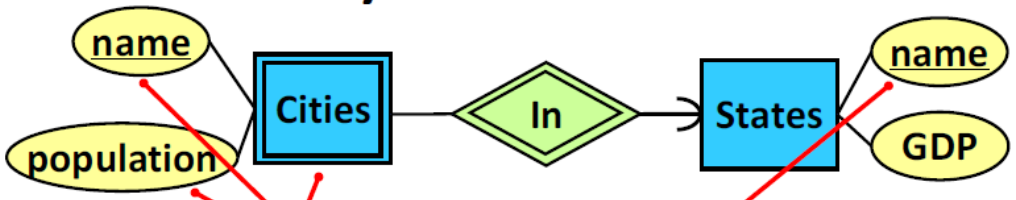
## Many-to-One Relationship → Relation




- Translation:
  - School( Hname, addr )
  - Students( ID, Sname, Hname )

- Only need to put the key of the "one" side into the relation of the "many" side

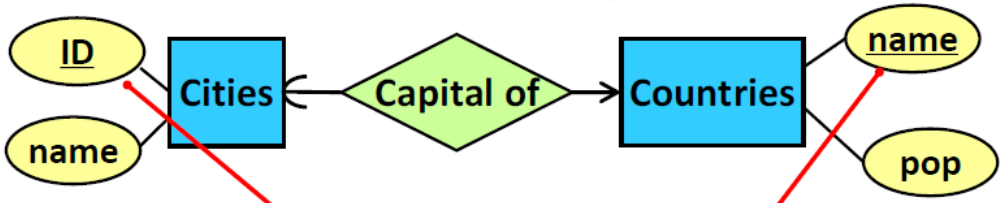
## Weak Entity Set → Relation




- Each weak entity set is converted to a relation that contains
  - all of its attributes, and
  - the key of the supporting entity set
  - attribute (if any) of supporting relationship
- The supporting relationship is ignored

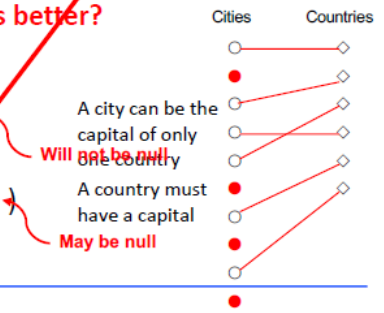
| Cities | state-name | city-name | population |
|---|---|---|---|
| | ... | ... | ... |

## One-to-One Relationship → Relation




- No need to create a relation for a one-to-one relationship
- Only need to put the key of one side into the relation of the other
- Solution 1
  - Cities( CityID, Cityname )
  - Countries( Countryname, pop, CityID )
- Solution 2
  - Cities( CityID, Cityname, Countryname )
  - Countries( Countryname, pop )

Sol 1 vs Sol 2: Which one is better?



# Tutorial 2

## Functional Dependencies:

- In general, how do we know whether a combination of attributes is bad?
- We need to check the correlations among those attributes

## Formal Definition of FD

- Attributes $A_1, A_2, \ldots, A_m, B_1, B_2, \ldots, B_n$
- $A_1A_2\ldots A_m \rightarrow B_1B_2\ldots B_n$
- Meaning: There do not exist two objects that
  - Have the same values on $A_1, A_2, \ldots, A_m$
  - but different values on $B_1, B_2, \ldots, B_n$

## Reasoning with FDs

- Given A→B, BC→D
- Can you prove that AC → D?
- Proof
  - Given A→B, we have AC → BC (Augmentation)
  - Given AC→BC and BC → D, we have AC → D (Transitivity)

## Reasoning with FDs

- Armstrong's Axioms
  - Three axioms for FD reasoning
  - Easy to understand, but not easy to apply
- Axiom of Reflexivity
  - A set of attributes → A subset of the attributes
- Axiom of Augmentation
  - Given A → B
  - We always have AC → BC, for any C
- Axiom of Transitivity
  - Given A → B and B → C
  - We always have A → C

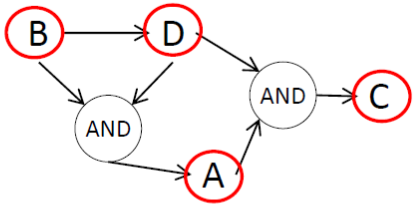
# Tutorial 2

## Reasoning with FDs

- Given A→B, BC→D
- Can you prove that AC → D?
- Proof
  - Given A→B, we have AC → BC (Augmentation)
  - Given AC→BC and BC → D, we have AC → D (Transitivity)

## Steps of the Intuitive Solution

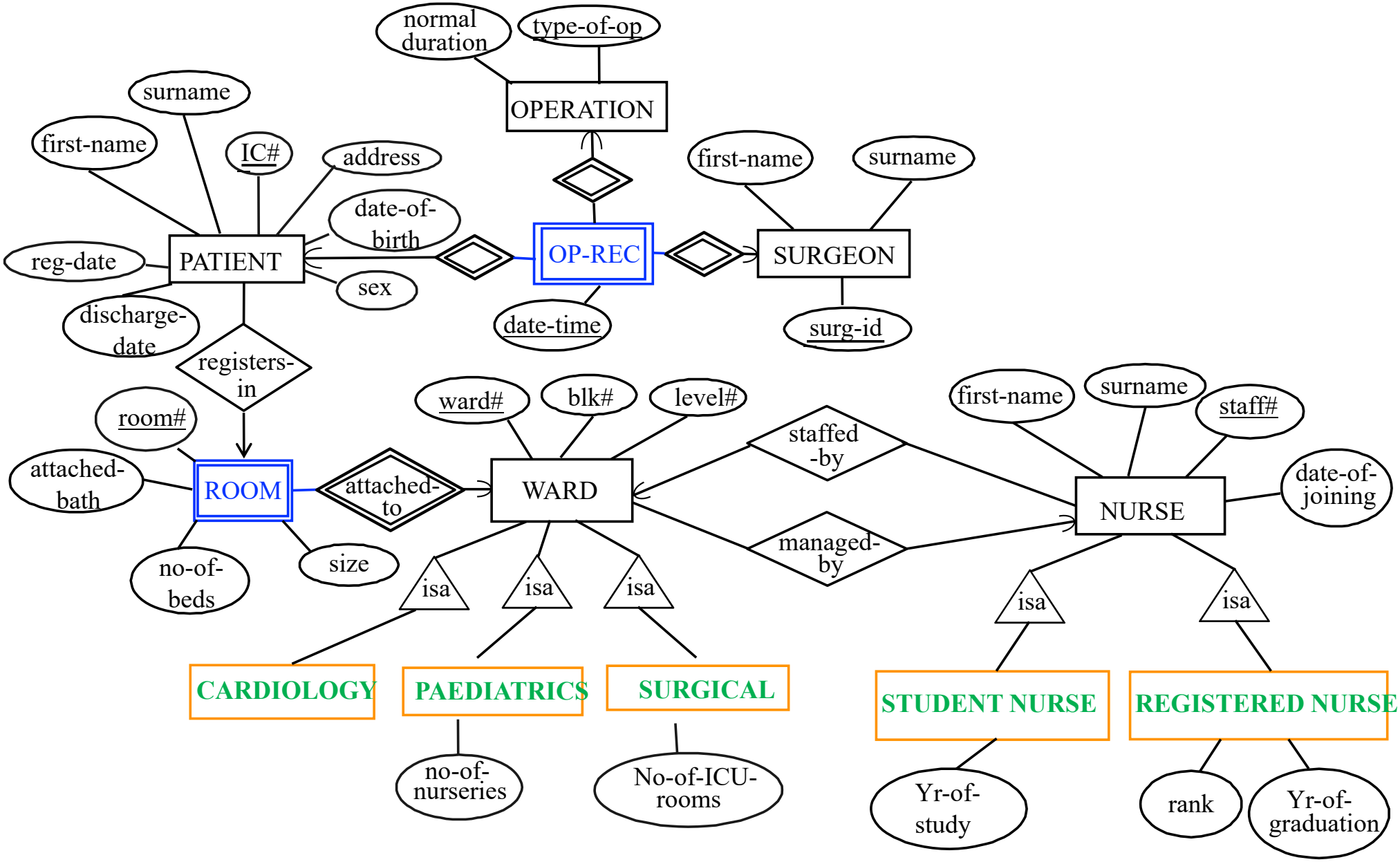
- Four attributes: A, B, C, D
- Given: B→D, DB→A, AD→C
- Can you prove B→C?




- First, activate B
  - Activated set = { B }
- Second, activate whatever B can activate
  - Activated set = { B, D }, since B→D
- Third, use all activated elements to activate more
  - Activated set = { B, D, A }, since DB→A
- Repeat the third step, until no more activation is possible
  - Activated set = { B, D, A, C }, since AD→C; done

# Question 1

1. Translate the ER Diagram of Q1 in Tutorial 1 into a set of relations.



# Question 1: Relational Tables

- Surgeon (surg-id, first-name, surname)
- Operation (type-of-op, normal-duration)
- Patient (IC#, first-name, surname, address, date-of-birth, sex, reg-date, discharge-date, room#, ward#) – m2o
- Ward (ward#, blk#, level#, no-of-nurseries, no-of-ICU-rooms, manager) – m2o
- Nurse (staff#, first-name, surname, date-of-joining, wardStaff#) – m2o
- Op-Rec (IC#, surg-id, type-of-op, date-time) – w.e./m2o
- Room (room#, ward#, attached-bath, no-of-beds) – w.e./m2o

# Question 1: Relational Tables

- Subclass relationships:
- Student_Nurse(staff#, Yr-of-study)
- Registered_Nurse(staff#, rank, yr-of-graduation)
- Surgical_Ward(ward#, No_of_ICU_rooms)
- Pediatrics_Ward(ward#, No_of_nurseries)
- Cardiology_Ward(ward#, no_of_heart_equipment)

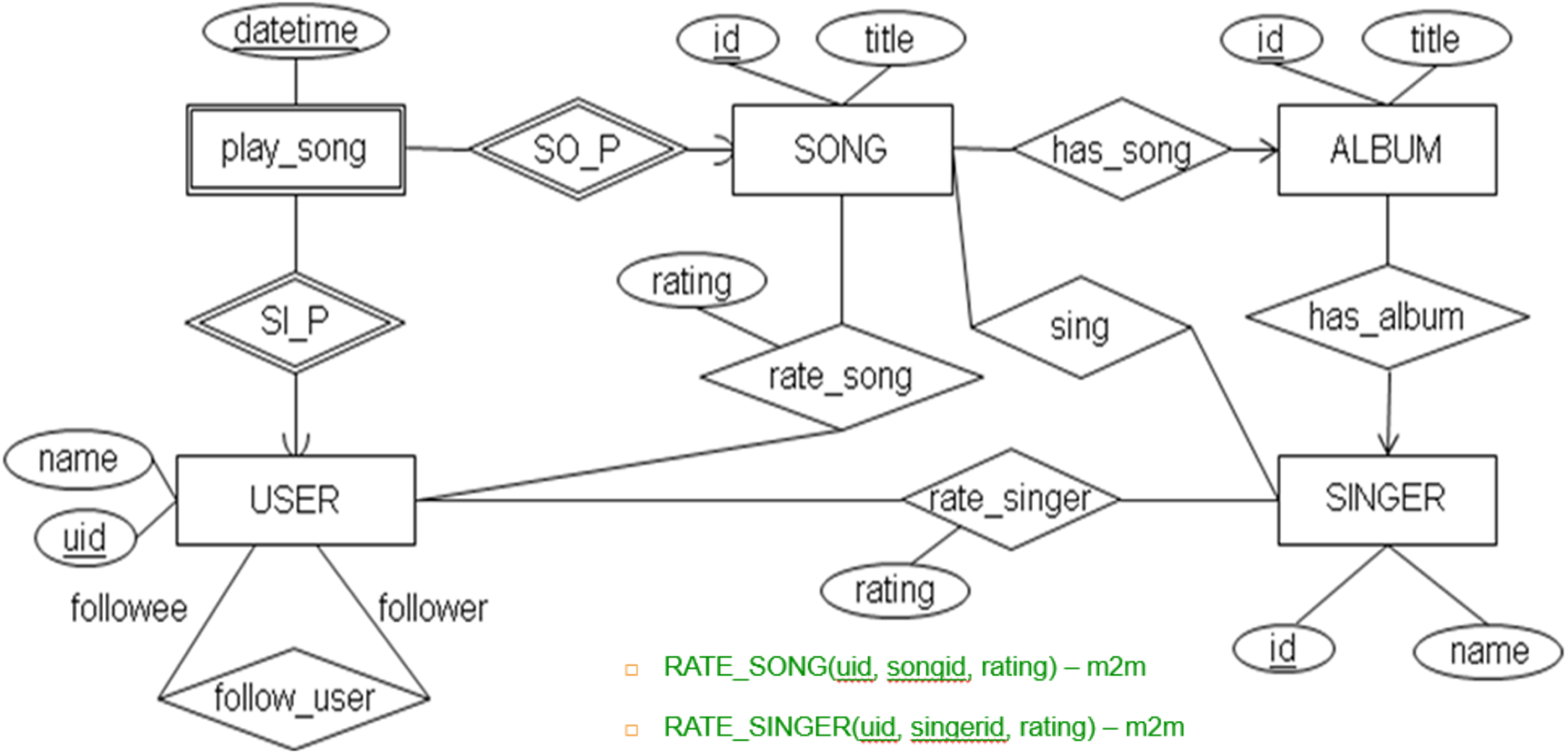
# Question 2

2. Consider the following relational schema:

- **USER**(uid, name)
- **SINGER**(id, name)
- **ALBUM**(id, title, singerid) – m2o
- **SONG**(id, title, albumid) – m2o
- SING(singerid, songid) – **m2m**
- FOLLOW_USER(followeruid, followeeuid) – **m2m**
- RATE_SONG(uid, songid, rating) – **m2m**
- RATE_SINGER(uid, singerid, rating) – **m2m**
- PLAY_SONG(uid, songid, datetime) – weak entity m2o

Construct an ER diagram that leads to the above schema.

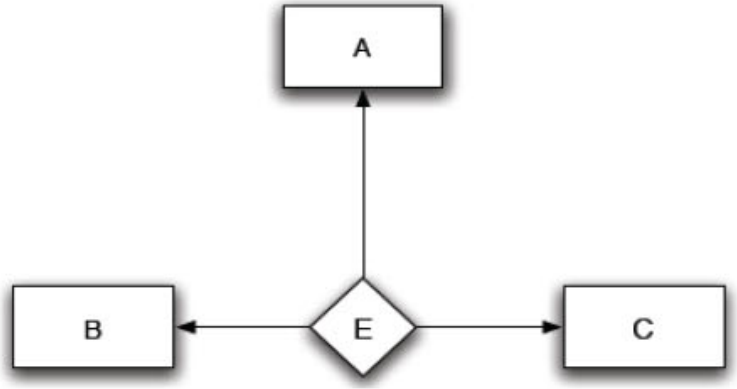
# Question 2



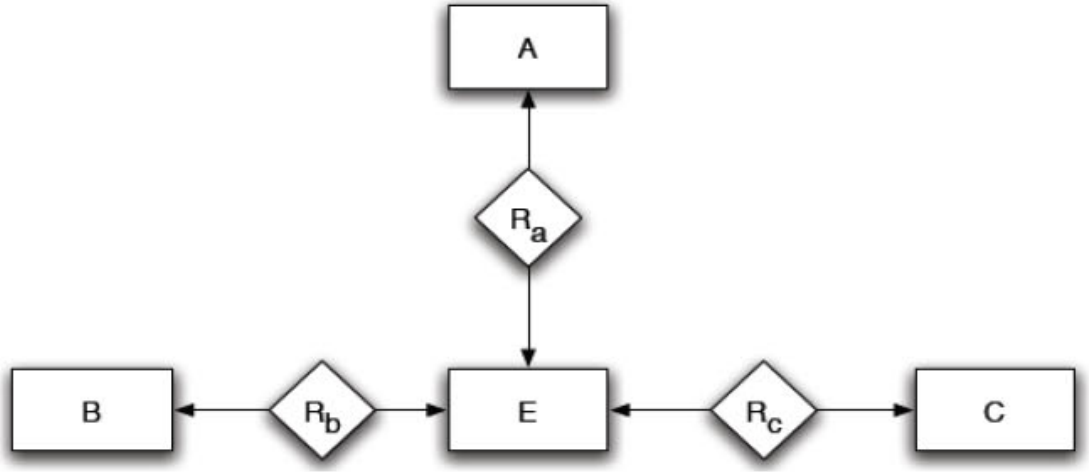

- RATE_SONG(uid, songid, rating) – m2m
- RATE_SINGER(uid, singerid, rating) – m2m
- PLAY_SONG(uid, songid, datetime) – weak entity m2o
- SING(singerid, songid) – m2m
- FOLLOW_USER(followeruid, followeeuid) – m2m
- SINGER(id, name)
- USER(uid, name)
- SONG(id, title, albumid) – m2o
- ALBUM(id, title, singerid) – m2o

# Question 3

3. The figure shows an attempt to represent a ternary relationship between three entities using 3 binary relationships (and one made-up entity). Show through an example why the 3 binary relationship representation is more general than the one ternary relationship representation.




A ternary relationship




An attempt to represent it using 3 binary relationships

# Question 3

Why is a **3 binary relationship relationship representation** more general than **one ternary relationship representation**? (https://stackoverflow.com/questions/39254141/ternary-relationship-or-3-binary-relationship)

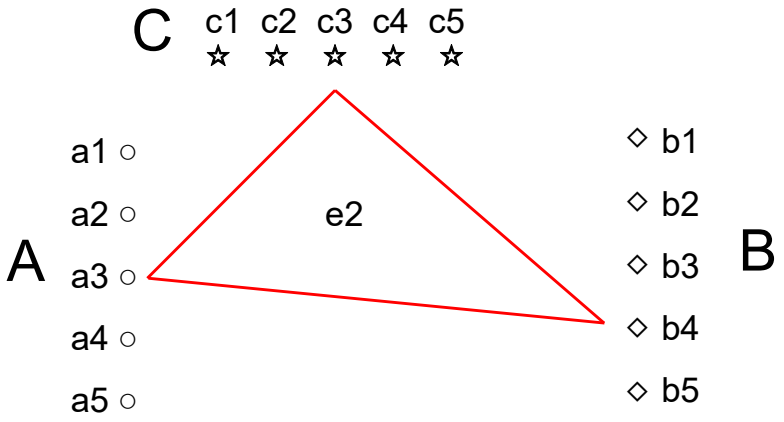
Because with 3 binary relationships each involved entity is related **separately** with each one of the others two.

Assumed that ternary relationship in essential only when you have a many-to-many-to-many relationship (otherwise you can rewrite it adding a relational entity linked with binary relationships to the others three), let's take an example:

- Suppose that you have the entities A, B and C, and three relationships that link the entities "like a triangle". Now, suppose that set of tuples a3 is related to the sets b4 and c3.
- Using 3 binary relationships, it's not requested that b4 and c3 are related. for example, b4 can be related to c2, that can partially overlap c3 or be completely disjoined.
- With a ternary relationship, instead, b4 must be related to c3.
- So you can see that with 3 binary relationships you can have much more combinations.

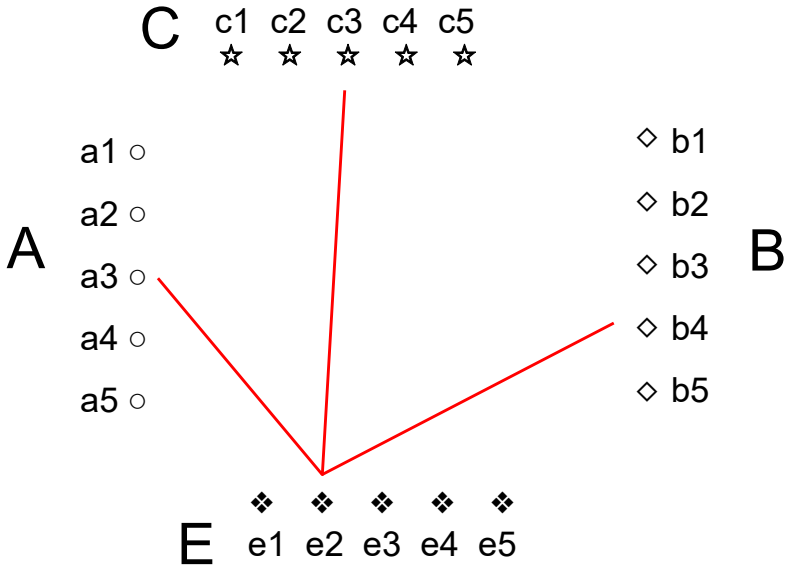
# Question 3

## Ternary relationship

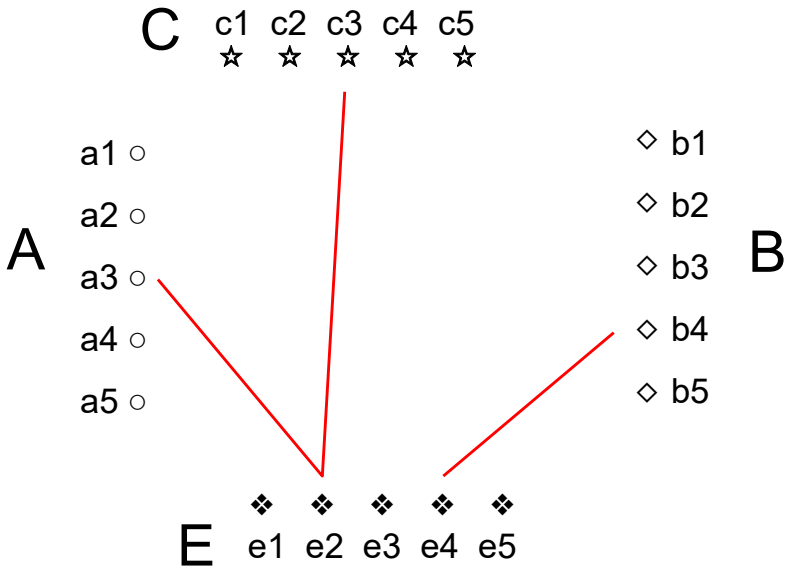



=

## Three 2-binary relationship

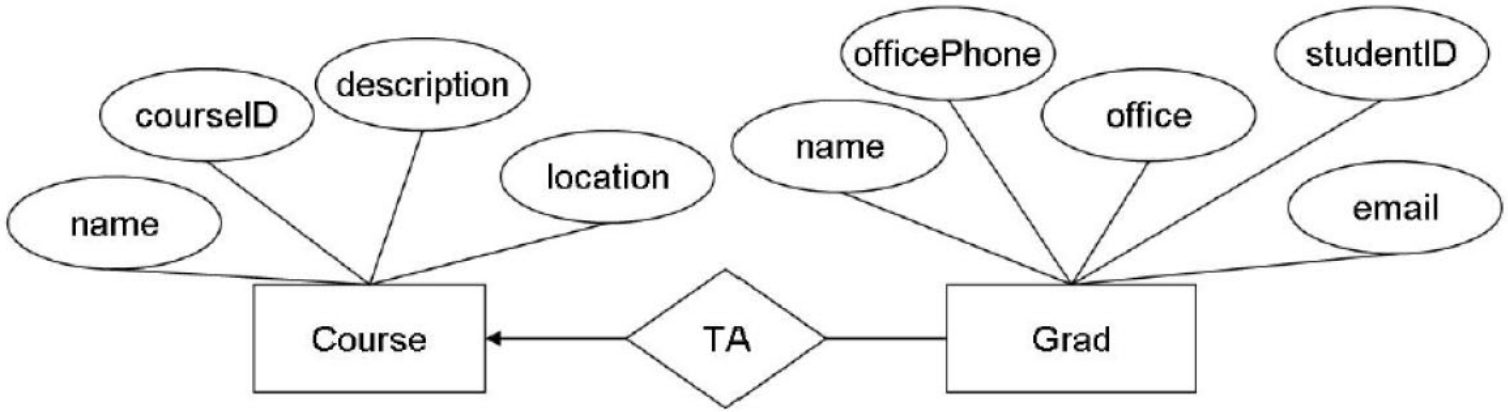



?



# Question 4

4. Consider the following ER diagram thath describes graduate students (Grad) and courses (Course) they serve as Teaching Assistants (TA).



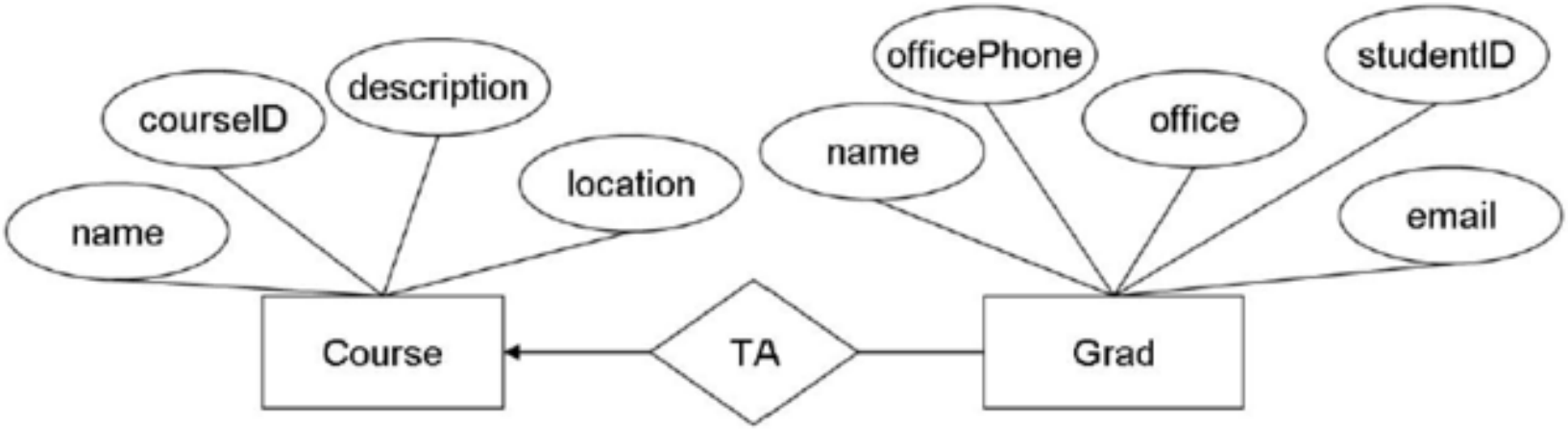

(a) For each of the following statements, write a functional dependency (FD) that best captures the statement.

- The studentID of each graduate student uniquely identifies the student.
- No two offices have the same phone number (officePhone).
- No two courses have the same courseID.
- If two courses have the same course name, their course descriptions are the same.

(b) From the ER diagram and the set of FDs you listed above, can you derive new FDs? If no, explain why not. If yes, derive two non-trivial FDs.

# Question 4(a)

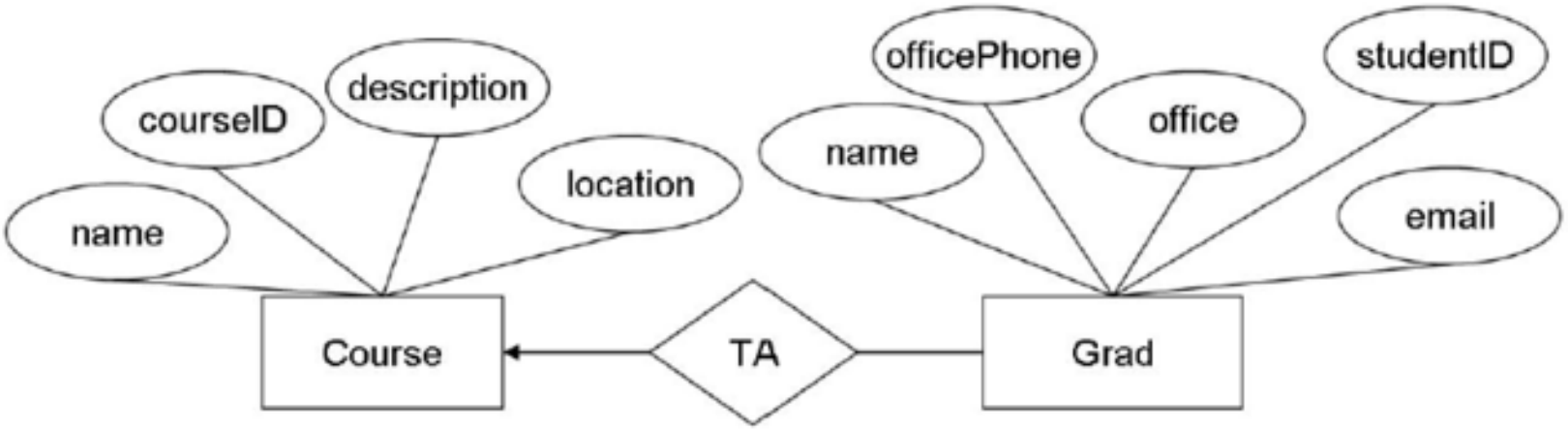
(a) For each of the following statements, write a functional dependency (FD) that best captures the statement.




- The studentID of each graduate student uniquely identifies the student.
  - F1: studentID $\rightarrow$ officePhone, office, email, name
- No two offices have the same phone number (officePhone).
  - F2: officePhone $\rightarrow$ Office

# Question 4(a)

(a) For each of the following statements, write a functional dependency (FD) that best captures the statement.




- No two courses have the same courseID.

- F3: courseID $\rightarrow$ name, description, location

- If two courses have the same course name, their course descriptions are the same.

- F4: name $\rightarrow$ description

# Question 4(b)

(b) From the ER diagram and the set of FDs you listed above, can you derive new FDs? If no, explain why not. If yes, derive two non-trivial FDs.

- F5: studentID $\rightarrow$ courseID.

  This is derived from the TA relationship, F1, and F3.

- F6: studentID $\rightarrow$ name, description, location.

  This is derived from F3 and F5.

# Tutorial 3
# Functional Dependencies & Normalisation

## Classroom Exercise

1. A medical clinic database schema contains the following:

   APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

   Show, with suitable examples, the insertion anomalies that the schema is liable to encounter.

2. Consider the relation ADDRESS having attributes Street, City, State and Zip. Assume that for any given zipcode, there is just one city and state. Also, for any given street, city, and state, there is just one zipcode.

   (a) Infer all possible functional dependencies (FDs) for this relation.
   (b) Which are possible minimal keys?

3. Prove the following properties using Armstrong's axioms or reject it by counterexample relations.

   (a) $A \rightarrow B \Rightarrow AC \rightarrow B$
   (b) $A \rightarrow C$ and $AB \rightarrow C \Rightarrow B \rightarrow C$

4. Consider a relation R(A, B, C, D) with the following FDs: $B \rightarrow C$, $D \rightarrow B$

   (a) Find the key(s) of R.
   (b) Is this relation in BCNF? Why or why not? If it is not, decompose R into a collection of relations that are in BCNF.

5. Prove that every two-attribute relationis in BCNF.

## Critical Thinking Exercise

6. Consider a relation R(A,B,C,D,E) and FD's $AB \rightarrow C$, $DE \rightarrow C$, and $B \rightarrow D$. Is this relation in BCNF? Why or why not? If it's not, decompose the relation into collections of relations that are in BCNF.

7. Consider a relation R(A,B,C,D,E) and FD's $A \rightarrow B$, $A \rightarrow C$, $BC \rightarrow A$, and $D \rightarrow E$. Is this relation in BCNF or 3NF or neither? If it's not in BCNF, decompose the relation into collections of relations that are in BCNF. Show each step of the decomposition process.

1. A medical clinic database schema contains the following:

   APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

   Show, with suitable examples, the insertion anomalies that the schema is liable to encounter.

key = patient-id

patient-id → Appointment (...)

(same patient-id being input with different doctor-id if anomalies

APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

| patient-id | patient-name | doctor-id | doctor-name | appt-date | appt-time | room-no |
|---|---|---|---|---|---|---|
| p001 | john | d100 | james | 20.01.01 | 0930 | r01 |
| p002 | steve | d100 | james | 20.01.02 | 1030 | r09 |
| p003 | william | d100 | james | 20.01.03 | 0930 | r08 |
| p004 | charles | d100 | james | 20.01.04 | 1030 | r07 |
| p005 | bob | null | null | null | null | null |
| null | null | d102 | lee | null | null | null |
| p003 | william | d100 | james | 20.01.10 | 1030 | r10 |
| p003 | william | d110 | michael | 20.01.11 | 1130 | r10 |

Clearly doctor-id → doctor-name is an FD and LHS is not superkey

1. We must ensure that the doctor attributes are consistent with other patients under this doctor; potential update anomalies
2. In order to insert a new patient, all the attributes of the doctor who is treating the patient must be also included; if no doctor, then use NULL, which is not good.
3. If a new doctor joins and has not seen any patients, we cannot enter the doctor's details. We cannot fill patient information with NULLs since this violates the primary key condition.
4. If the same patient has more than one appointment, its ID will be repeated, which violates the unique property of a key.

2. Consider the relation ADDRESS having attributes Street, City, State and Zip. Assume that for any given zipcode, there is just one city and state. Also, for any given street, city, and state, there is just one zipcode.

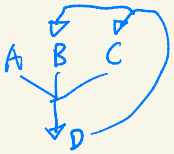
   (a) Infer all possible functional dependencies (FDs) for this relation.
   (b) Which are possible minimal keys?

A B C D

- FD's:
  - D → BC because D+={DBC}
  - AD → BC because AD⁺={ABCD}
  - BD → C because BD⁺={BDC}
  - CD → B because CD⁺={CDB}
  - ABC → D because ABC⁺={ABCD}
  - ABD → C because ABD⁺={ABCD}
  - ACD → B because ACD⁺={ABCD}
- Minimal keys: ABC, AD

a) ABC → D
D → BC

A A
B B
C C
D BCD




key = AD

b) ABC, AD

(1) make sure right only 1 attr
ABC → D
D → B
D → C

(2) make sure there is no redundant
ABC → D ?
D → B ?
D → C ?

{AB C}⁺ = {ABC} no D not redundant
{D}⁺ = {D} no B not redundant
{D}⁺ = {D,B} no C not redundant

(3) check if we can remove any left hand side FD

BC ABC → D ?

A → D ? A⁺ = {A}
B → D ? B⁺ = {B}
C → D ? C⁺ = {C}
minimal basis:

ABC → D
D → B
D → C

3. Prove the following properties using Armstrong's axioms or reject it by counterexample relations.

(a) $A \to B \Rightarrow AC \to B$

(b) $A \to C$ and $AB \to C \Rightarrow B \to C$

a) $A \to B$

$AC \to BC$ Augmentation

$BC \to B$ Reflexivity

$AC \to B$ Transitivity

closure

b) $A \to C$

$AB \to C$

then $B \to C$

$A^+ = \{A, C\}$

$B^+ = \{B\}$ C is not in B

$C^+ = \{C\}$

given $(a_1, b_1, c_1)$

$(a_2, b_1, c_2)$

4. Consider a relation R(A, B, C, D) with the following FDs: $B \to C$, $D \to B$

(a) Find the key(s) of R.

(b) Is this relation in BCNF? Why or why not? If it is not, decompose R into a collection of relations that are in BCNF.

a) $B \to C$

D A

$A^+ = \{A\}$

$B^+ = \{B, C\}$

$C^+ = \{C\}$

$D^+ = \{D, B, C\}$

$AD = \{A, D, B, C\}$

key = AD

b) No, because non-trivial

$B \to C$ LHS has no AD attrs

$D \to B$

R(A, B, C, D)

$B \to C$ $B^+ = \{B, C\}$

$R_1 = \{B, C\}$ BCNF ✓

key AD $R_2 = \{A, B, D\}$ BCNF

$D^+ = \{D, B, C\}$

key D $R_1 = D, B, C$ $B \to C$ vio

$R_2 = AD$

$B^+ = \{B, C\}$

$R_1 = B, C$

$R_2 = B, D$

5. Prove that every two-attribute relationis in BCNF.

- Case 1: A→B holds, but B→A does not. A is the key. The only nontrivial FD is A→B; no BCNF violation.
- Case 2: B→A holds, but A→B does not. B is the key. The only nontrivial FD is B→A; no BCNF violation.
- Case 3: Both A→B and B→A hold. A and B are both keys; no BCNF violation.
- Case 4: AB is key; so we have AB→AB, which trivial.

TUT3

2) $D \to BC$

$ABC \to D$

$D \to B$

$D \to C$

b) AD, ABC

3) a) $A \to B \Rightarrow AC \to B$

$Ac \to BC$ $\quad Ac \to B$

$BC \to B$

b) $A \to C$ $\quad B \to C$

$AB \to C$

given

| A | B | C |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |

4) a) AD

b) Not BCNF as FD LHS doesn't contain key

R(A,B,C,D) $B \to C$ $D \to B$

Violate $D \to B$ $\{D\}^+ = \{D, B, C\}$

$R_1(B,C,D)$ $R_2(A,D)$ ✓

↳ $B \to C$ $\alpha\{B\}^+ = \{B, C\}$

$R_3(B,C)$ ✓ $R_4(B,D)$ ✓

## Critical Thinking Exercise

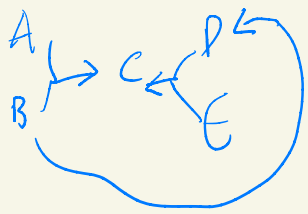
6. Consider a relation R(A,B,C,D,E) and FD's AB → C, DE → C, and B → D. Is this relation in BCNF? Why or why not? If it's not, decompose the relation into collections of relations that are in BCNF.

7. Consider a relation R(A,B,C,D,E) and FD's A → B, A → C, BC → A, and D → E. Is this relation in BCNF or 3NF or neither? If it's not in BCNF, decompose the relation into collections of relations that are in BCNF. Show each step of the decomposition process.

6) key : ABE

AB → C violate

$\{AB\}^+ = \{ABCD\}$




R1 (A,B,C,D) key AB Not BCNF B → D

R2 (A,B,E) key ABE BCNF ✓

R1 (A,B,C,D)

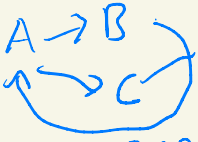
B → D $\{B^+\} = \{B,D\}$ ✓

R3 (B,D) BCNF

R4 (A,B,C) key AB BCNF ✓

7) R(A,B,C,D,E)

A → B




D → E

key AD, BCD

A → B $\{A,B,C\}$

R1 (ABC) BCNF

R2 (ADE)

# CZ2007 Tutorial 3: FDs + BCNF

Week 5

# Question 1

- A medical clinic database schema contains the following:
- APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)
- Show, with suitable examples, the anomalies that the schema is liable to encounter.
- Key is patient-id.
- So FD is patient-id $\rightarrow$ patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no

# Question 1

APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

| patient-id | patient-name | doctor-id | doctor-name | appt-date | appt-time | room-no |
|---|---|---|---|---|---|---|
| p001 | john | d100 | james | 20.01.01 | 0930 | r01 |
| p002 | steve | d100 | james | 20.01.02 | 1030 | r09 |
| p003 | william | d100 | james | 20.01.03 | 0930 | r08 |
| p004 | charles | d100 | james | 20.01.04 | 1030 | r07 |
| | | | | | | |
| | | | | | | |

1

# Question 1

APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

| patient-id | patient-name | doctor-id | doctor-name | appt-date | appt-time | room-no |
|---|---|---|---|---|---|---|
| p001 | john | d100 | james | 20.01.01 | 0930 | r01 |
| p002 | steve | d100 | james | 20.01.02 | 1030 | r09 |
| p003 | william | d100 | james | 20.01.03 | 0930 | r08 |
| p004 | charles | d100 | james | 20.01.04 | 1030 | r07 |
| p005 | bob | null | null | null | null | null |
| | | | | | | |

1

2

# Question 1

APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

| patient-id | patient-name | doctor-id | doctor-name | appt-date | appt-time | room-no |
|---|---|---|---|---|---|---|
| p001 | john | d100 | james | 20.01.01 | 0930 | r01 |
| p002 | steve | d100 | james | 20.01.02 | 1030 | r09 |
| p003 | william | d100 | james | 20.01.03 | 0930 | r08 |
| p004 | charles | d100 | james | 20.01.04 | 1030 | r07 |
| p005 | bob | null | null | null | null | null |
| null | null | d102 | lee | null | null | null |

1

2

3

# Question 1

APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

| patient-id | patient-name | doctor-id | doctor-name | appt-date | appt-time | room-no |
|---|---|---|---|---|---|---|
| p001 | john | d100 | james | 20.01.01 | 0930 | r01 |
| p002 | steve | d100 | james | 20.01.02 | 1030 | r09 |
| p003 | william | d100 | james | 20.01.03 | 0930 | r08 |
| p004 | charles | d100 | james | 20.01.04 | 1030 | r07 |
| p005 | bob | null | null | null | null | null |
| null | null | d102 | lee | null | null | null |
| p003 | william | d100 | james | 20.01.10 | 1030 | r10 |
| p003 | william | d110 | michael | 20.01.11 | 1130 | r10 |

1

2

3

4

# Question 1

Clearly doctor-id → doctor-name is an FD and LHS is not superkey

1. We must ensure that the doctor attributes are consistent with other patients under this doctor; potential update anomalies

2. In order to insert a new patient, all the attributes of the doctor who is treating the patient must be also included; if no doctor, then use NULL, which is not good.

3. If a new doctor joins and has not seen any patients, we cannot enter the doctor's details. We cannot fill patient information with NULLs since this violates the primary key condition.

4. If the same patient has more than one appointment, its ID will be repeated, which violates the unique property of a key.

# Question 1

- APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)
- **Can we infer all FDs in this relation?**
- patient-id → patient-name
- doctor-id → doctor-name
- patient-id, doctor-id → appointment-date, appointment-time, clinic-room-no ?
- appointment-date, appointment-time, clinic-room-no → patient-id, doctor-id ?

# Question 2

- Consider the relation ADDRESS having attributes Street, City, State and Zip.
- Assume that for any given zipcode, there is just one city and state. Also, for any given street, city, and state, there is just one zipcode.
- (a) Infer all possible functional dependencies (FDs) for this relation.
- (b) Which are possible minimal keys?

# Question 2

- Let us denote attributes STREET, CITY, STATE, ZIP as A, B, C and D respectively. Then we have D $\rightarrow$ BC and ABC $\rightarrow$ D.
- Use closure to find FDs:
  - $A^+$={A}; $B^+$={B}; $C^+$={C}; $D^+$={DBC}
  - $AB^+$={AB}; $AC^+$={AC}; $AD^+$={ABCD}; $BC^+$={BC}; $BD^+$={BDC}; $CD^+$={CDB};
  - $ABC^+$={ABCD}; $ABD^+$={ABCD}; $ACD^+$={ABCD}; $BCD^+$={BCD}

# Question 2

- FD's:
  - D $\rightarrow$ BC because D+={DBC}
  - AD $\rightarrow$ BC because $AD^{+}$={ABCD}
  - BD $\rightarrow$ C because $BD^{+}$={BDC}
  - CD $\rightarrow$ B because $CD^{+}$={CDB}
  - ABC $\rightarrow$ D because $ABC^{+}$={ABCD}
  - ABD $\rightarrow$ C because $ABD^{+}$={ABCD}
  - ACD $\rightarrow$ B because $ACD^{+}$={ABCD}
- Minimal keys: ABC, AD

# Question 3

- Prove the following properties using Armstrong's axioms or reject it by counterexample relations.
- (a) $A \rightarrow B \Rightarrow AC \rightarrow B$
- (b) $A \rightarrow C$ and $AB \rightarrow C \Rightarrow B \rightarrow C$

## Question 3a

- (a) $A \rightarrow B \Rightarrow AC \rightarrow B$
- $A \rightarrow B \Rightarrow AC \rightarrow BC$ (Augmentation Rule)
- $BC \rightarrow B$ (Reflexivity Rule)
- $AC \rightarrow BC$ and $BC \rightarrow B \Rightarrow AC \rightarrow B$ (Transitivity Rule)

# Question 3b

- (b) $A \rightarrow C$ and $AB \rightarrow C \Rightarrow B \rightarrow C$
- Consider the records:
- (a1, b1, c1)
- (a2, b1, c2)
- Both $A \rightarrow C$ and $AB \rightarrow C$ are true but $B \rightarrow C$ does not hold

# Question 4

4. Consider a relation R(A, B, C, D) with the following FDs: $B \rightarrow C$, $D \rightarrow B$

   (a) Find the key(s) of R.
   (b) Is this relation in BCNF? Why or why not? If it is not, decompose R

   into a collection of relations that are in BCNF.

# Question 4

- Use closure to find FDs:
  - $A^+$={A}; $B^+$={BC}; $C^+$={C}; $D^+$={DBC}
  - $AB^+$={ABC}; $AC^+$={AC}; $AD^+$={ABCD}; $BC^+$={BC}; $BD^+$={BCD}; $CD^+$={BCD};
  - $ABC^+$={ABC}; $ABD^+$={ABCD}; $ACD^+$={ABCD}; $BCD^+$={BCD}
  - Key is AD
- Also we note that A and D do not appear on RHS of FDs; so key must contain AD. In fact, key is AD.

# Question 4

- Key is AD.
- Both FDs B $\rightarrow$ C, D $\rightarrow$ B violate BCNF. Can decompose using either one first.
- By first decomposing on B $\rightarrow$ C , we get R1(B, C ) and R2(A, B, D).
- R2 is not in BCNF due to violating FD D $\rightarrow$ B, so we decompose further: R3(A,D) and R4(B, D).
- Both R3 and R4 are in BCNF (see Q5)

# Question 4

- By first decomposing on D → B, we get R1(B, C, D) and R2(A, D).
- R1 is not in BCNF due to violating FD B → C, so we must decompose further.
- In both cases, we end up with three relations, Ra(A, D), Rb(B, C) and Rc(B, D).

# Question 5

- Prove that every two-attribute relationis in BCNF.

- There are 4 cases in a two-attribute relation.

# Question 5

- Case 1: A→B holds, but B→A does not. A is the key.

  The only nontrivial FD is A→B; no BCNF violation.

- Case 2: B→A holds, but A→B does not. B is the key.

  The only nontrivial FD is B→A; no BCNF violation.

- Case 3: Both A→B and B→A hold. A and B are both keys; no BCNF violation.

- Case 4: AB is key; so we have AB→AB, which trivial.


Do the best
you can until you know better.
Then when you know better, do better.
~ Maya Angelou
ALL THE BEST

# Tutorial 4
# Normalisation

## Classroom Exercise

1. A medical clinic database schema contains the following:

   APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

   Identify the functional dependencies in the schema, stating any assumptions made. Using these functional dependencies, normalise the schema to Third Normal Form.

2. Consider the relation Courses(C, T, H, R, S, G) whose attributes may be thought informally as course, teacher, hour, room, student, and grade. Let the set of FD's of Courses be:

   C → T, HR → C, HT → R, HS → R, and CS → G.

   (a) What are all the keys for Courses?

   (b) Verify that the given FDs are their own minimal basis.

   (c) Use the 3NF decomposition algorithm to find a lossless-join, dependency-preserving decomposition.

3. Consider a relation R(W, X, Y, Z) which satisfies the following set of FDs G = {Z→W, Y→X, Y→Z, XW→Y}, where G is a minimal basis.

   (a) Decompose R into a set of relations in 3NF.

   (b) Is the decomposition also in BCNF? Explain your answer.

4. Consider a relation R(A,B,C,D,E) and FD's A → BC, CD → E, E → A, and B → D.

   (a) Is the decomposition R1(A,B,C) and R2(A,D,E) of R lossless or lossy? Justify your answer. Is this decomposition dependency preserving? If your answer is NO, then what is not preserved?

   (b) Is the decomposition R3(A,B,C,D) and R4(C,D,E) of R lossless or lossy? Justify your answer. Is this decomposition dependency preserving? If your answer is NO, then what is not preserved?

## Critical Thinking Exercise

5. We perform decomposition to normalize an original schema to be of certain normal forms. For such a decomposition to be "equivalent" to the original schema, it is desirable to be lossless. To study this concept, let's consider an original schema R(A, B, C). Suppose we decompose R into R1(A, B) and R2(A, C).

   (a) Is this decomposition always lossless? Answer *yes* or *no* and briefly explain why.
   (b) Give an example instance of *R* (*i.e.*, an example table with several tuples) and demonstrate its decomposition, to support your answer in (a).

1. A medical clinic database schema contains the following:

   APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)

   Identify the functional dependencies in the schema, stating any assumptions made. Using these functional dependencies, normalise the schema to Third Normal Form.

FD:

patient-id → patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no

doctor-id → doctor-name

key: patient-id

a → bcdefg

c → d

efg → a, c

minimum basis =

① keep right 1 attr

| | |
|---|---|
| a→b | c→d |
| a→c | efg→a |
| a→d | efg→c |
| a→e | |
| a→f | |
| a→g | |

② remove redundant

| | | |
|---|---|---|
| a→b | $a^+$ = acdefg | nr |
| a→c | $a^+$ = abdefg | nr |
| a→d | $a^+$ = abcefg | r |
| a→e | $a^+$ = abcdfg | nr |
| a→f | $a^+$ = abcdg | nr |
| a→g | $a^+$ = abcdf | nr |
| c→d | c = c | nr |
| efg→a | | nr |
| efg→c | efg→c | r |

③ see if left can be 1

a→b & efg → x

a→c c→d

a→e

a→f e efg→g

a→g f⁺=f x

efg→a ,

→ minimal basis

- Suppose other than A→BCDEFG, we derive other FDs using common sense:
  - doctor-id → doctor-name; i.e., C→D
  - appointment-date, appointment-time, clinic-room-no → patient-id, doctor-id; i.e., EFG→AC
- So altogether we have: A→BCDEFG, C→D, EFG→AC
- Let's check MB conditions: Condition 1 says RHS must be single attribute, so we have: A→B, A→C, A→D, A→E, A→F, A→G, C→D, EFG→A, EFG→C
- Condition 2 says no redundant FDs. EFG→C and A→D are redundant and removed. Condition 3 says no redundant LHS attributes. None of LHS attribute of EFG→A is redundant.
- So MB={A→B, A→C, A→E, A→F, A→G, C→D, EFG→A} and we form relations AB, AC, AE, AF, AG, CD, AEFG.
- Do these relations make sense? Any other way to do this?

a→b ① step1: RHS contain 1 attr: a→b, c→d, efg→a, efg→c

efg→ac step2: remove redundant FDs: a→b not redundant; c→d not redundant; efg→a nr; efg→c nr

step3: remove any LHS attr that is redundant: none

3NF: ① step1 a→b c→d efg→ac; a→b; c→d; efg→a; efg→c; R1 ab, R2 cd, R3 acefg

2. Consider the relation Courses(C, T, H, R, S, G) whose attributes may be thought informally as course, teacher, hour, room, student, and grade. Let the set of FD's of Courses be:

   C → T, HR → C, HT → R, HS → R, and CS → G.

   (a) What are all the keys for Courses? SH

   (b) Verify that the given FDs are their own minimal basis.

   (c) Use the 3NF decomposition algorithm to find a lossless-join, dependency-preserving decomposition.

- Suppose we have the following FDs using common sense:
  - patient-id → patient-name; i.e., A→B
  - doctor-id → doctor-name; i.e., C→D
  - appointment-date, appointment-time, clinic-room-no → patient-id, doctor-id; i.e., EFG→AC
- So altogether we have: A→B, C→D, EFG→AC
- Let's check MB conditions: Condition 1 says RHS must be single attribute, so we have: A→B, C→D, EFG→A, EFG→C
- Condition 2 says no redundant FDs. There are none. Condition 3 says no redundant LHS attributes. There are none.
- So MB={A→B, C→D, EFG→A, EFG→C} and we form relations AB, CD, ACEFG.
- Do these relations make sense? Yes. This last approach seems the best.

c) CT, CHR, THR, SHR, CSG → CHRTS

a)


$\{SH\}^+ = \{S, H, R, C, T, G\}$

b) ① make right of FD only 1 attr

C → T

HR → C

HT → R

HS → R

CS → G

② remove redundant FDs

| | |
|---|---|
| C→T | nr |
| HR→C | nr |
| HT→R | nr |
| HS→R | nr |
| CS→G | nr |

③ LHS attr redundant

3. Consider a relation R(W, X, Y, Z) which satisfies the following set of FDs G = {Z→W, Y→X, Y→Z, XW→Y}, where G is a minimal basis.

   (a) Decompose R into a set of relations in 3NF.

   (b) Is the decomposition also in BCNF? Explain your answer.

Z→W
Y→X
Y→Z
XW→Y

Z→W, Y→W, Y→X, Y→Z, XW→Y × key Y, XW, XZ

a)

$R_{1,2,3}$
ZW
YXZ
XWY

Z→W
Y→XZ
XW→Y

b) non-trivial FD must contain key

key = Y

Z→W ✓
Y→XZ Y ✓
XW→Y

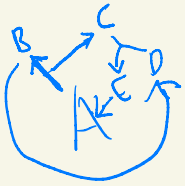
4. Consider a relation R(A,B,C,D,E) and FD's A → BC, CD → E, E → A, and B → D.

   (a) Is the decomposition R1(A,B,C) and R2(A,D,E) of R lossless or lossy? Justify your answer. Is this decomposition dependency preserving? If your answer is NO, then what is not preserved?

   (b) Is the decomposition R3(A,B,C,D) and R4(C,D,E) of R lossless or lossy? Justify your answer. Is this decomposition dependency preserving? If your answer is NO, then what is not preserved?

A→BC
CD→E
E→A
B→D

a) lossless as key A can visit all other key (superkey)




key = A

dependency preserving

CD→E

B→D

$R_1(A,B,C)$
A→BC Bc→A ✓

$R_2(A,B,E)$
E→A A→E ✓

b) $R_3(A,B,C,D)$
$R_4(C,D,E)$
CD

$R_3(A,B,C,D)$
A→Bc Bc→A
B→D D→B

$R_4(C,D,E)$
CD→G, E→D

E→A break

5. We perform decomposition to normalize an original schema to be of certain normal forms. For such a decomposition to be "equivalent" to the original schema, it is desirable to be lossless. To study this concept, let's consider an original schema R(A, B, C). Suppose we decompose R into R1(A, B) and R2(A, C).

   (a) Is this decomposition always lossless? Answer *yes* or *no* and briefly explain why.
   (b) Give an example instance of $R$ (*i.e.*, an example table with several tuples) and demonstrate its decomposition, to support your answer in (a).

a)

TUT 4

1) $A \to G$

$a \to b$

$C \to d$

$e,f,g \to a,c$

① derive minimal basis

- ensure RHS of FD has only 1 attr

$a \to b$

$C \to d$

$e,f,g \to a$

$e,f,g \to c$

- Check redundancy

if $a \to b$ redundant: $\{a\}^+ = \{a\}$ no

if $c \to d$: $\{c\}^+ = \{c\}$ no

if $efg \to a, c$: $\{efg\}^+ = \{efg\}$ no

- Check if LHS redundant

no

2) G, C, T, H, S, R

a) keys: HS

b) ① ensure RHS of FD has only 1 attr

$C \to T$

$HR \to C$

$HT \to R$

$HS \to R$

$CS \to G$

② Check redundancy of all FDs

$C \to T$ $\{C\}^+ = \{C\}$

$HR \to C$ $\{HR\}^+ = \{H,R\}$

$HT \to R$ $\{HT\}^+ = \{H,T\}$

$HS \to R$ $\{HS\}^+ = \{H,S\}$

$CS \to G$ $\{C,S\}^+ = \{C,S\}$

-1 minimal basis

-2 combine FD that has the same LHS ✓

-3 CT, HRC HTR HSR CSG

3) ensure LHS multi attr is not redundant

$HR \to C$ $H \to C$ ✗ $R \to C$ ✗

$HT \to R$ $H \to R$ ✓ $T \to R$ ✗

$HS \to R$ $H \to R$ ✓ $S \to R$ ✗

$CS \to G$ $C \to G$ ✗ $S \to G$ ✗

$C \to T$

$HR \to C$

$HT \to R$

$HS \to R$

$CS \to G$

1) minimal basis

2) combine FD with same LHS

$Z \to W$ $XZ$

$Y \to XZ$ $Y$

$XW \to Y$ $XW$

3) table

ZW XYZ WXY

↓ ↓ ↓

BCNF Z BCNF Y BCNF XW

4) a) $A \to B$, $A \to C$, $B \to D$, $CD \to E$, $E \to C$

key: A, E, CD, BC

lossless

$R(A,B,C)$ $A \to BC$ ✓

$R_2(A,D,E)$ $A \to E$ ✓ $E \to A$ ✓

$CD \to E$ $CD^+ = \{CD\}^+$

$B \to D$ $B^+ = \{B\}^+$

b) lossless

$R_1(A,B,C)$ $A \to BC$ ✓ $B \to D$ ✓

$R_2(C,D,E)$ $CD \to E$ ✓

$E \to A$ $E$ ✓

# CZ2007 Tutorial 4: BCNF + 3NF

Week 6

# Question 1

- A medical clinic database schema contains the following:
- APPOINTMENT (patient-id, patient-name, doctor-id, doctor-name, appointment-date, appointment-time, clinic-room-no)
- Identify the functional dependencies in the schema, stating any assumptions made.
  - There could be different sets of FDs depending on how you interpret them.
- Using these functional dependencies, normalise the schema to Third Normal Form.
  - We may get different sets of normalized relations.

# Question 1

- Let's map the attribute names to simpler letters:
  - patient-id to A
  - patient-name to B
  - doctor-id to C
  - doctor-name to D
  - appointment-date to E
  - appointment-time to F
  - clinic-room-no to G
- Since A is a key, we have default functional dependency:

  A→BCDEFG, assuming no other FDs.
- Does this FD form a minimal basis (MB)?

# Question 1

- Since A is a key, we have default functional dependency: A→BCDEFG, assuming no other FDs.
- Does this FD form a minimal basis (MB)?
- Condition 1 of MB: RHS is single attribute. So we break the FD into: A→B, A→C, A→D, A→E, A→F, A→G
- Conditions 2 and 3 are satisfied since there are no other FDs to reason.
- So MB = {A→B, A→C, A→D, A→E, A→F, A→G} and we can form relations AB, AC, AD, AE, AF, AG from it.
- Do these relations make sense? Any other way to do this?

# Question 1

- Suppose other than A→BCDEFG, we derive other FDs using common sense:
  - doctor-id → doctor-name; i.e., C→D
  - appointment-date, appointment-time, clinic-room-no → patient-id, doctor-id; i.e., EFG→AC
- So altogether we have: A→BCDEFG, C→D, EFG→AC
- Let's check MB conditions: Condition 1 says RHS must be single attribute, so we have: A→B, A→C, A→D, A→E, A→F, A→G, C→D, **EFG→A, EFG→C**
- Condition 2 says no redundant FDs. **EFG→C** and A→D are redundant and removed. Condition 3 says no redundant LHS attributes. None of LHS attribute of EFG→A is redundant.
- So MB={A→B, A→C, A→E, A→F, A→G, C→D, EFG→A} and we form relations AB, AC, AE, AF, AG, CD, AEFG.
- Do these relations make sense? Any other way to do this?

# Question 1

- Suppose we have the following FDs using common sense:
  - patient-id → patient-name; i.e., A→B
  - doctor-id → doctor-name; i.e., C→D
  - appointment-date, appointment-time, clinic-room-no → patient-id, doctor-id; i.e., EFG→AC
- So altogether we have: A→B, C→D, EFG→AC
- Let's check MB conditions: Condition 1 says RHS must be single attribute, so we have: A→B, C→D, EFG→A, EFG→C
- Condition 2 says no redundant FDs. There are none. Condition 3 says no redundant LHS attributes. There are none.
- So MB={A→B, C→D, EFG→A, EFG→C} and we form relations AB, CD, ACEFG.
- Do these relations make sense? Yes. This last approach seems the best.

# Question 1

- Using common sense, we derive functional dependencies:
  - FD1: patient-id → patient-name
  - FD2: doctor-id → doctor-name
  - FD3: appointment-date, appointment-time, clinic-room-no → patient-id, doctor-id
- Using 3NF normalization, we have decomposed relations:
  - PATIENT( patient-id, patient-name )
  - DOCTOR( doctor-id, doctor-name )
  - APPOINTMENT( appointment-date, appointment-time, clinic-room-no , patient-id, doctor-id )

# Question 2

Consider the relation Courses(C, T, H, R, S, G) whose attributes may be thought informally as course, teacher, hour, room, student, and grade. Let the set of FD's of Courses be:

$C \rightarrow T$, $HR \rightarrow C$, $HT \rightarrow R$, $HS \rightarrow R$, and $CS \rightarrow G$.

(a) What are all the keys for Courses?

(b) Verify that the given FDs are their own minimal basis.

(c) Use the 3NF decomposition algorithm to find a lossless-join, dependency-preserving decomposition.

# Question 2(a)

- The usual procedure to find keys is to take the closure of all 63 nonempty subsets.
- However, we notice that none of the right sides of the FDs contains attributes H and S; we may conclude that **H** and **S** must be part of any key.
- Given FDs C→T, HR→C, HT→R, HS→R, CS→G, let's start with HS.
  - HS→R ⇒ HS→HR and HR→C ⇒ HS→C; $HS^+$={CHRS}
  - CS→G and C→T ⇒ **$HS^+$={CGHRST}**
- Using the closure method, we eventually find out that **HS** is the only key in the Courses relation.

# Question 2(b)

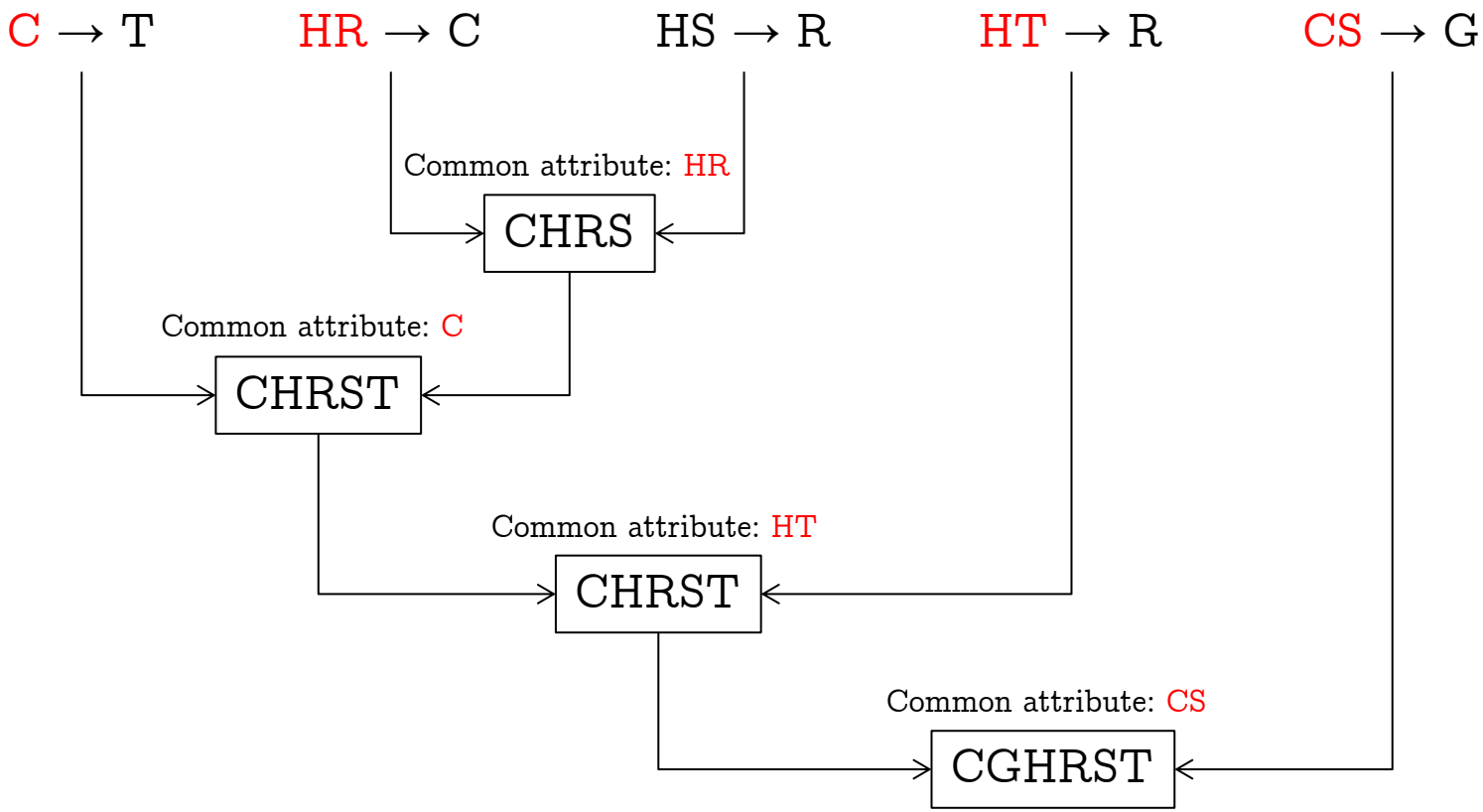
- Given FDs: C→T, HR→C, HT→R, HS→R, CS→G
- Check if any of the FDs is redundant.
  - None. If we remove any one of the five FDs, the remaining four FDs do not imply the removed FD.
- Check if any of the LHS attribute of an FD can be removed without losing the dependencies.
  - None. The attributes on the left side of the four FDs are not redundant.
- Thus, the given set of FDs is a minimal basis.

# Question 2(c)

- Since the only key is HS, the given set of FDs has some dependencies that violate 3NF.
  - Violating FDs: C→T, HR→C, HT→R, CS→G
- We also know that the given set of FDs is a minimal basis. Thus, the decomposed relations are (CT), (HRC), (HTR), (HSR) and (CSG).
- Since the relation HSR contains a key, we do not need to add an additional relation. The final set of decomposed relations is (CT), (HRC), (HTR), (HSR) and (CSG).
- Since each decomposed relation came from a FD, the decomposition is FD preserving.

# Question 2(c)

The following sequence of joins shows that the decomposition is lossless:



# Question 3

Consider a relation R(W, X, Y, Z) which satisfies the following set of FDs G = {Z→W, Y→X, Y→Z, XW→Y}, where G is a minimal basis.

(a) Decompose R into a set of relations in 3NF.

(b) Is the decomposition also in BCNF? Explain your answer.

# Question 3(a)

- Given G ={Z→W, Y→X, Y→Z, XW→Y}, where G is a minimal basis
  - Decomposed relations: R1(Z, W), R2(X, Y, Z), R3(X, Y, W)
- To determine whether BCNF satisfied, find keys using reasoning:
  - Y→X, Y→Z, Z→W yield Y→WXYZ; so Y is a key
  - XW→Y and Y is a key means XW is a key.
  - XW is a key and Z→W yields XZ→XW; so XZ is a key.
  - Keys: Y, WX, XZ

# Question 3(b)

- Keys: Y, WX, XZ; check FDs in R1(Z,W), R2(X,Y,Z), R3(X,Y,W):
  - FDs in R1: 2-attribute relation is in BCNF
  - FDs in R2: Y→X, Y→Z; LHS are keys
  - FDs in R3: Y→X, XW→Y; LHS are keys
- Since all LHS of FDs are keys; relations are in BCNF

# Question 4

Consider a relation R(A,B,C,D,E) and FD's A → BC, CD → E, E → A, and B → D.

(a) Is the decomposition R1(A,B,C) and R2(A,D,E) of R lossless or lossy? Justify your answer. Is this decomposition dependency preserving? If your answer is NO, then what is not preserved?

(b) Is the decomposition R3(A,B,C,D) and R4(C,D,E) of R lossless or lossy? Justify your answer. Is this decomposition dependency preserving? If your answer is NO, then what is not preserved?

# Question 4

- FDs: A → BC, E → A, CD → E, B → D
- (a) Decomposition: R1(A,B,C) and R2(A,D,E)
- (b) Decomposition: R3(A,B,C,D) and R4(C,D,E)
- Is decomposition lossless (can be joined back)?
- Is decomposition dependency preserving (no FDs lost)?
- Keys: A, E, CD, BC

# Question 4(a)

- Decomposition: R1(A,B,C) and R2(A,D,E); FDs: A → BC, E → A, CD → E, B → D; Keys: A, E, CD, BC
- Decomposition R1(A,B,C) and R2(A,D,E) is lossless because
  - R1 and R2 have a common attribute A, and
  - A is a superkey for R1(A,B,C)
- FDs that hold on R1(A,B,C)
  - A → BC, BC → A since R1 contains A, B, and C
- FDs that hold on R2(A,D,E)
  - E → A, A → E since R2 contains A and E
- Two other FDs need to be checked: CD → E, B → D
  - From A → BC, BC → A, E → A, A → E, we have:
  - {B}+ = {B}, so B → D is not preserved
  - {CD}+ = {CD}, so CD → E is not preserved
- Decomposition is NOT dependency-preserving

# Question 4(b)

- Decomposition: R3(A,B,C,D) and R4(C,D,E); FDs: $A \rightarrow BC$, $E \rightarrow A$, $CD \rightarrow E$, $B \rightarrow D$; Keys: A, E, CD, BC
- Decomposition R3(A,B,C,D) and R4(C,D,E) is lossless because
  - R3 and R4 have common attributes CD, and
  - CD is a superkey for R4(C,D,E)
- FDs that hold on R3(A,B,C,D)
  - $A \rightarrow BC$, $BC \rightarrow A$, ~~$CD \rightarrow E$~~, $CD \rightarrow A$, and $B \rightarrow D$ since R1 contains A, B, C, and D
- FDs that hold on R4(C,D,E)
  - $CD \rightarrow E$, $E \rightarrow CD$
- One other FD needs to be checked: $E \rightarrow A$
  - From $A \rightarrow BC$, $BC \rightarrow A$, $CD \rightarrow A$, $B \rightarrow D$, $CD \rightarrow E$, $E \rightarrow CD$ and, we have:
  - $\{E\}+ = \{A, B, C, D, E\}$, $E \rightarrow CD$ holds in R4, $CD \rightarrow A$ holds in R3, $E \rightarrow A$ is preserved
- Decomposition is dependency-preserving


Do the best
you can until you know better.
Then when you know better, do better.
~ Maya Angelou
ALL THE BEST

# Tutorial 5
# Relational Algebra

## Classroom Exercise

Consider a database with three tables as follows:

Shopper(**shopperName**, street, ageGroup)
Mall(**mallName**, street)
ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

The three tables record information about shoppers, shopping malls, as well as "which shoppers shop at which malls". Primary Keys are in **bold**.

Write the following queries in relational algebra. When answering relational algebra queries, breakdown your answers into intermediate steps. When answering relational algebra queries, each answer should be in the form of one single table containing only relevant output attributes. If you think a question's solution cannot be expressed in relational algebra, explain why.

1. Find those shopper(s) who shopped at all the malls on "Nanyang Ave" every Thursday between 10am to 5pm, and find the streets that these shoppers live in.

2. Find the age groups of those shoppers(s) who only shop at malls that are located on the street where he/she lives.

3. Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers who have shopped there more times than anyone else does. Also find out these shoppers' age groups.

4. Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers in the 20s-30s age group who have never shopped at Jurong Point Mall on Friday evenings between 7pm to 10pm. Also find out which streets these shoppers live in.

5. Find shopping malls that have never been visited by shoppers in the 40s-50s age group on Wednesday mornings between 9am to 11am. Also find out which streets these malls are located.

6. For each shopper, find how many other shoppers shopped at the same malls as him/her on the same date.

7. Find the mall(s) that is/are shopped by the largest number of repeat shoppers in the 20s-30s age group. Repeat shoppers of a mall are shoppers who have shopped at least once in the mall.

## Additional Exercises (Optional)

1. A library database schema contains the following tables:

   LIB-MEMBER(ID, name, age)
   BOOK(serial#, title, author, year-of-publication)
   LOAN(ID, serial#, date-due)

   State what each of the following relational algebra queries is looking for:

   (a) $\pi_{name}((\sigma_{year\text{-}of\text{-}publication<1960}\ BOOK \bowtie LOAN) \bowtie LIB\text{-}MEMBER)$

   (b) $\pi_{ID}(\sigma_{age<21}\ LIB\text{-}MEMBER) - \pi_{ID}(\sigma_{author=\text{“J.K.Rowling”}}\ BOOK \bowtie LOAN)$

   (c) $\pi_{name}((\pi_{ID,serial\#}\ LOAN \div \pi_{serial\#}(\sigma_{title\ like\ \text{‘C Progamming’}}\ BOOK)) \bowtie LIB\text{-}MEMBER)$

2. The schema of a database containing university-type data is given below. Primary key is underlined for each relation.

   STUDENT(Sid, Sname, Sex, Age, Year, GPA)
   DEPT(Dname, NumPhds)
   PROF(Pname, Dname)
   MAJOR(Dname, Sid)
   COURSE(Dname, C#, Cname)
   SECTION(Dname, C#, Sect#, Pname)
   ENROLL(Sid, Dname, C#, Sect#, Grade)

   Write the following queries in relational algebra.

   (a) Find the names of professors who work in departments that have fewer than 50 PhD students.

   (b) Find the name(s) of student(s) with the lowest GPA.

   (c) Find the names and majors of students who have taken the ‘Database System’ course.

   (d) Find the ids, names, and GPAs of the students who have taken all courses from the ‘Civil Engineering’ department.

3. Consider the following relational schema (primary keys are underlined). eventtype can take values: SWI (swimming), ATH (athletics), GYM (gymnastics), etc. medal can take values: gold, silver, bronze. You may assume player names are unique.

   PLAYERS(player-id, name, countryname, age)
   EVENTS(event-id, name, eventtype)
   RESULTS(player-id, event-id, medal)

   Write relational algebra expression for the following queries.

   (a) Find the names of the players who won at least one gold and one silver.

(b) Find the players who did not win a medal.

TUT 5

Shopper(**shopperName**, street, ageGroup)
Mall(**mallName**, street)
ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

1. Find those shopper(s) who shopped at all the malls on "Nanyang Ave" every Thursday between 10am to 5pm, and find the streets that these shoppers live in.

2. Find the age groups of those shoppers(s) who only shop at malls that are located on the street where he/she lives.

3. Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers who have shopped there more times than anyone else does. Also find out these shoppers' age groups.

4. Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers in the 20s-30s age group who have never shopped at Jurong Point Mall on Friday evenings between 7pm to 10pm. Also find out which streets these shoppers live in.

5. Find shopping malls that have never been visited by shoppers in the 40s-50s age group on Wednesday mornings between 9am to 11am. Also find out which streets these malls are located.

6. For each shopper, find how many other shoppers shopped at the same malls as him/her on the same date.

7. Find the mall(s) that is/are shopped by the largest number of repeat shoppers in the 20s-30s age group. Repeat shoppers of a mall are shoppers who have shopped at least once in the mall.

1) $\pi_{shopper}((\sigma_{street='Nanyang\ Ave'}(Mall)) \bowtie (\sigma_{time \geq 10 \text{ AND } time \leq 5}(ShopAT)))$

2) $\pi_{ageGroup}(Shopper \bowtie_{Shopper.street = Mall.street\ Mall})$

3) $\pi_{shopper, ageGroup}(Shopper \bowtie ($

4) $\sigma_{mallname='Jurong\ point\ mall'}$ $\gamma_{shoppername, Count(date)}$

5) $\pi_{mallna}$

Shopper(**shopperName**, street, ageGroup)

Mall(**mallName**, street)

ShopAt(**shopperName, mallName, date, time**, dayOfWeek)

1. Find those shopper(s) who shopped at all the malls on "Nanyang Ave" every Thursday between 10am to 5pm, and find the streets that these shoppers live in.
2. Find the age groups of those shoppers(s) who only shop at malls that are located on the street where he/she lives.
3. Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers who have shopped there more times than anyone else does. Also find out these shoppers' age groups.
4. Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers in the 20s-30s age group who have never shopped at Jurong Point Mall on Friday evenings between 7pm to 10pm. Also find out which streets these shoppers live in.
5. Find shopping malls that have never been visited by shoppers in the 40s-50s age group on Wednesday mornings between 9am to 11am. Also find out which streets these malls are located.
6. For each shopper, find how many other shoppers shopped at the same malls as him/her on the same date.
7. Find the mall(s) that is/are shopped by the largest number of repeat shoppers in the 20s-30s age group. Repeat shoppers of a mall are shoppers who have shopped at least once in the mall.

1) $R_1 := \sigma_{street = \text{'Nanyang Ave'}}\ Mall$

$R_2 := R_1 \bowtie_{R1.mallName = ShopAt.mallN}$

$R_3 := \sigma_{dayOfWeek = \text{'Thursday'}\ AND\ (time}$

$R_4 := \Pi_{shopperName,\ street}\ (Shopper$

2) $R_0 := \rho_{mall1(mallName,\ mallStreet)}\ Mall$

$R_1 := ShopAt \bowtie R_0$

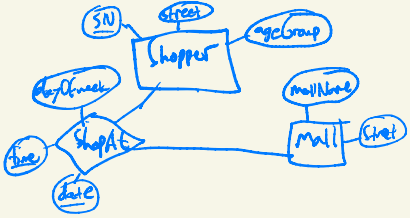
$R_2 := Shopper \bowtie R_1$

$R_3 := \sigma_{R2.street = R2.mallStreet}\ R_2$

3) $R_1 :=$

Tut 6

2) $R_1 := \sigma_{mallName = \text{'Hougang Ave'}} Mall$
$R_2 := \rho_{R2(mallName,\ street\_mall)} R_1$
$R_3 := ShopAt \bowtie R_2$
$R_4 := \sigma_{dayOfWeek = \text{'Thursday'} \text{ AND } (time \geq 10 \text{ AND } time \leq 17)} R_3$
$R_5 := R_4 \bowtie R_1$
$R_6 := \pi_{shopperName,\ Street}(R_5)$




2) $R_1 := \rho_{R1(mallName,\ street\_mall)} Mall$
$R_2 := ShopAt \bowtie R1$
$R_3 := Shopper \bowtie_{shopper.street = R_2.street\_mall \text{ and } Shopper.ShopperName = R_2.ShopperName} R_2$
$R_4 := \pi_{shopper,\ ageGroup}(R_3)$

3) $R_1 := \sigma_{mallName = \text{'Jurong Point Mall'}} ShopAt$
$R_2 := \gamma_{shopperName,\ Count(date) \rightarrow Cdate} R_1$
$R_3 := R_2 \bowtie Shopper$
$R_4 := \pi_{shopperName,\ ageGroup,\ Cdate} R_3$
$R_5 := \gamma_{Max(Cdate) \rightarrow CMax} R_4$
$R_4 := R_4 \bowtie_{R_4.cdate = cmax} R_5$

4) $R_1 := (\sigma_{ageGroup >= 20 \text{ AND } ageGroup <= 30} Shopper) \bowtie ShopAt$
$R_2 := \sigma_{mallName = \text{'Jurong Point Mall'} \text{ AND } dayOfWeek = \text{'friday'} \text{ AND } time >= 19 \text{ AND } time <= 22} R_1$
$R_3 := R_1 - R_2$
$R_4 := \pi_{ShopperName,\ Street} R_3$

5) $R_1 := (\sigma_{agegroup = \text{'40-50'}} Shopper) \bowtie ShopAt$
$R_2 := \sigma_{dayOfweek = \text{'wednesday'} \text{ AND } time >= 9am \text{ AND } time <= 11am}$
$R_3 := R_1 - R_2$
$R_4 := \pi_{mallName} R_3$
$R_5 := R_4 \bowtie Mall$

6) $R_1 := \pi_{ShopperName,\ mallName,\ date} ShopAt$
$\rho_{R2(S,\ mallName,\ date)} R_1$
$R_3 := R_1 \bowtie_{R_1.mallName = R_2.mallName \text{ AND } R_1.date = R_2.date \text{ AND } R_1.Shopper <> R_2.S}$
$R_4 := \gamma_{ShopperName,\ count(ShopperName) \rightarrow CS}(\delta(R_3))$

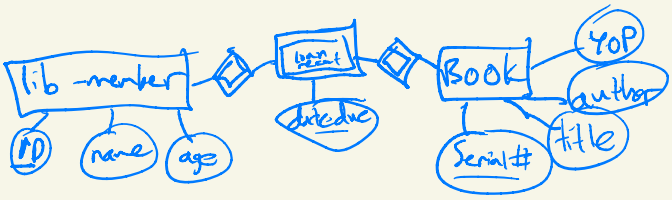
7) $R_1 := \sigma_{ageGroup = \text{'19-30'}} Shopper$
$R_2 := R_1 \bowtie ShopAt$
$R_3 := \gamma_{shopperName,\ mallName,\ Count(date) \rightarrow cdate}$
$R_4 := \gamma_{mallName,\ Count(ShopperName) \rightarrow cShopper}$
$R_5 := \gamma_{max(CShopper) \rightarrow Max}$

Additional

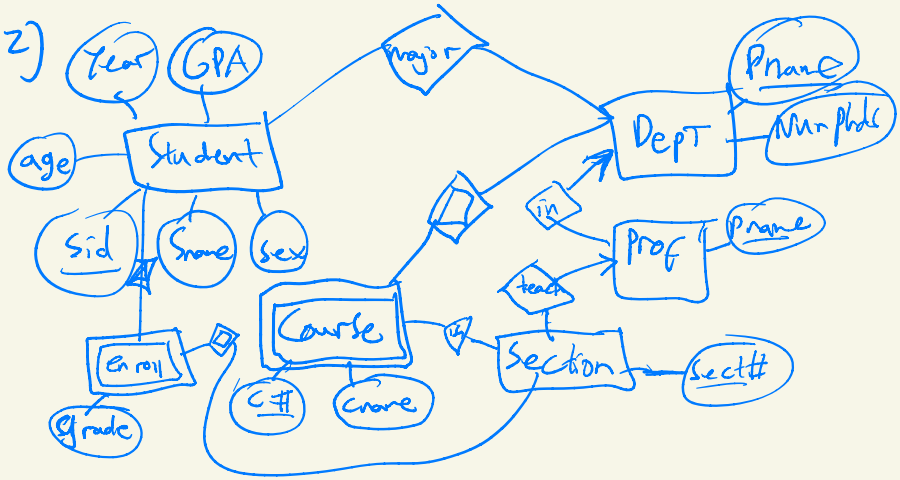
1) a) Find Library member who loan book publish before 19 60

b) Find ID of Lib member who is less than 21 and has never borrow book from author name JK. Rowling

c) Find name of lib member who loan all book with title containing c programming




2)




a) $R1 := \sigma_{Num phds < 50}\ DEPT$

$R2 := \pi_{Pname}(R1 \bowtie PROF)$

b) $R1 := \gamma_{Sname, Min(GPA) \rightarrow lowest}$

c) $R1 := \sigma_{Cname = 'Database System'} COURSE$

$R2 := (R1 \bowtie MAJOR) \bowtie STUDENT$

$R3 := \pi_{Sname, Dname}(R2)$

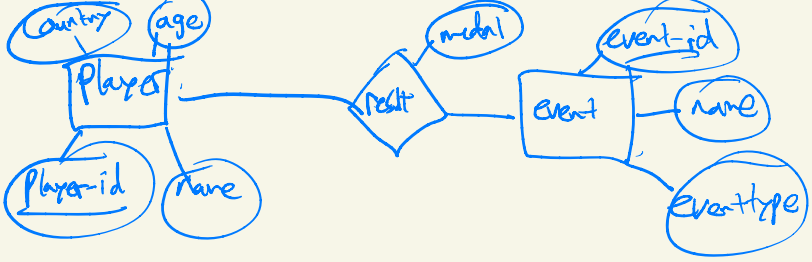
d) $R1 := \pi_{Cname}(Course \bowtie (\sigma_{Dname = 'Civil Engineering'} DEPT))$

$R2 := \pi_{Sname, Cname}((Student \bowtie MAJOR) \bowtie Course)$

$R3 := R1 \div R2$

$R4 := STUDENT \bowtie_{student.Sname = R3.Sname} R3$

3)




a) $R_1 := \pi_{name, medal}(PLAYER \bowtie RESULTS)$

$R_2 := \gamma_{name, medal, count(medal) \rightarrow Tmedal} R_1$

$R_3 := \pi_{name}(\sigma_{(medal = 'gold' \text{ AND } Tmedal > 1)} R_2)$

$R_4 := \pi_{name}(\sigma_{(medal = 'Silver' \text{ AND } Tmedal > 1)} R_2)$

$R_5 := R_3 \bowtie R_4$

b) $R_1 := \pi_{Player\_id}(PLAYERS \bowtie RESULTS)$

$R_2 := \pi_{Player\_id}(PLAYERS)$

$R_3 := R_1 - R_3$

$R_4 := \pi_{name}(R_3 \bowtie Players)$

# CZ2007 Tutorial 5: Relational Algebra

Week 7

# Question 1

Shopper(shopperName, street, ageGroup)

Mall(mallName, street)

ShopAt(shopperName, mallName, date, time, dayOfWeek)

# Question 1

Find those **shopper**(s) who shopped at all the **malls** on "Nanyang Ave" every **Thursday** between **10am to 5pm**, and find the **streets** that these shoppers live in.

Select **malls** on Nanyang Ave

$R1 := \Pi_{mallName} (\sigma_{street = \text{'Nanyang Ave'}} Mall)$

Select **shoppers** who shopped on Nanyang Ave malls on Thursday between 10am to 5pm

$R2 := \sigma_{street = \text{'Nanyang Ave'} \text{ and } dayOfWeek = \text{'Thursday'} \text{ and } time >= 10am \text{ and } time <= 5pm} (ShopAt \bowtie Mall)$

Select **shoppers** and **malls**

$R3 := \Pi_{shopperName,\ mallName} (R2)$

Find **shoppers** who shopped at ALL **malls** on Nanyang Ave

$R4 := R3 \div R1$

Find **shoppers'** streets

$R5 := R4 \bowtie Shopper$

Answer: $\Pi_{shopperName,\ street} (R5)$

# Question 2

Find the **age** groups of those **shoppers**(s) who **only** shop at malls that are located on the **street** where he/she lives.

Find **shoppers** who shopped, take note of the **streets** the shoppers lived

$R1 := \Pi_{shopperName,\ mallName,\ street}\ (Shopper \bowtie_{Shopper.shopperName=ShopAt.shopperName} ShopAt)$

Find **malls** shopped by shoppers, take note of the **streets** where malls are located

$R2 := \Pi_{shopperName,\ mallName,\ street}\ (Mall \bowtie_{Mall.mallName=ShopAt.mallName} ShopAt)$

Do relation renaming for easier manipulation later

$\rho_{R3(sName,\ mName,\ sStreet}\ (R1)$

$\rho_{R4(sName,\ mName,\ mStreet}\ (R2)$

# Question 2

Find **shoppers** who ever before shopped at **malls** on the same **street** they live

$R5 := \Pi_{shopperName} (R3 \bowtie_{R1.sName=R2.sName \text{ and } R1.mName=R2.mName \text{ and } sStreet<>mStreet} R4)$

Find **shoppers** who only shopped at **malls** on the same **streets** they live

$R6: \Pi_{shopperName} (Shopper) - R5$

Find **shoppers**' age group

$R7 := R6 \bowtie Shopper$

**Answer:** $\Pi_{shopperName,\ ageGroup} (R7)$

# Question 3

Consider **Jurong Point Mall**, the shopping mall that is 3.5km south of NTU. Find those shoppers who have shopped there **more times** than anyone else does. Also find out these shoppers' **age** groups.

Find shopping activities at Jurong Point Mall

$R1 := \sigma_{\text{mallName} = \text{'Jurong Point'}} \text{ ShopAt}$

Count how many times each shopper shopped at JPM

$R2 := \gamma_{\text{shopperName, COUNT(date)} \rightarrow \text{VisitCount}} \text{ R1}$

Find out maximum count

$R3 := \gamma_{\text{MAX(VisitCount)} \rightarrow \text{MaxVisitCount}} \text{ R2}$

Find out which shopper has this max count

$R4 := \Pi_{\text{shopperName}} (R2 \bowtie_{\text{VisitCount} = \text{MaxVisitCount}} R3)$

Find out shoppers' age group

$\text{Result} := \Pi_{\text{shopperName, ageGroup}} (\text{Shopper} \bowtie R4)$

# Question 4

Consider Jurong Point Mall, the shopping mall that is 3.5km south of NTU. Find those shoppers in the 20s-30s age group who have **never shopped** at Jurong Point Mall on Friday evenings between 7pm to 10pm. Also find out which streets these shoppers live in.

Find shopping activities at JPM on Friday between 7-10pm

$R1 := \sigma_{\text{mallName = 'Jurong Point' and dayOfWeek = 'Friday' and time >= 7pm and time <= 10pm}} (\text{ShopAt})$

Extract shoppers' names from these activities

$R2 := \Pi_{\text{shopperName}} (R1)$

Find shoppers in 20s-30s

$R3 := \sigma_{\text{ageGroup = '20s-30s'}} (\text{Shopper})$

Find shoppers in 20s-30s who shopped at JPM on Friday 7-10pm

$R4 := \Pi_{\text{shopperName}} (R2 \bowtie R3)$

Find all other shoppers

$R5 := (\Pi_{\text{shopperName}} R3) - R4$

Find shoppers' streets

$\text{Result} := \Pi_{\text{shopperName, street}} (\text{Shopper} \bowtie R5)$

# Question 5

Find shopping **malls** that have never been visited by **shoppers** in the 40s-50s age group on Wednesday mornings between 9am to 11am. Also find out which **streets** these malls are located.

Find **shopping activities** on Wednesday between 9 to 11am.

$R1 := \sigma_{\text{dayOfWeek = 'Wednesday' and time >= 9am and time <= 11am}} (\text{ShopAt})$

Find shoppers in the **40-50s** age group.

$R2 := \sigma_{\text{ageGroup = '40s-50s'}} (\text{Shopper})$

Find those **malls** shopped by shoppers in the 40-50s age group who shopped on Wednesday mornings between 9am to 11am.

$R3 := \Pi_{\text{mallName}} (R1 \bowtie R2)$

Find out all **other malls**.

$R4 := (\Pi_{\text{mallName}} \text{Mall}) - R3$

Find the **streets** of these **other malls**.

$\text{Result} := \Pi_{\text{mallName, street}} (\text{Mall} \bowtie R4)$

# Question 6

For each shopper, find **how many other shoppers** shopped at the **same malls** as him/her on the **same date**.

Extract relevant attributes.

$\rho_{R1(s1,\ mall,\ date)}\ (\Pi_{shopperName,\ mallName,\ date}\ ShopAt)$

$\rho_{R2(s2,\ mall,\ date)}\ (\Pi_{shopperName,\ mallName,\ date}\ ShopAt)$

Find shoppers who shopped at the **same malls** on the **same date**.

$R3 := R1 \bowtie_{s1<>s2\ and\ R1.mall=R2.mall\ and\ R1.date=R2.date}\ R2$

Extract just shoppers' **names**.

$R4 := \Pi_{s1,\ s2}\ (R3)$

For each shopper, count **how many other shoppers** shopped at the same malls on the same date.

$Result := \gamma_{s1,\ COUNT(s2)\rightarrow numS2}\ R4$

# Question 7

Find the mall(s) that is/are shopped by the **largest number** of **repeat shoppers** in the 20s-30s age group. Repeat shoppers of a mall are shoppers who have shopped **more than once** in the mall.

For each mall, find **how many times** shoppers shopped there.

$R1 := \gamma_{\text{mallName, shopperName, COUNT(date)}\rightarrow\text{numTimes}}(\text{ShopAt})$

Find those **malls** being shopped by the **same shoppers** more than once.

$R2 := \sigma_{\text{numTimes} > 1}(R1)$

These are the **repeat shoppers**.

$\rho_{\text{RepeatShoppers}(\textbf{shopperName}, \textbf{mallName}, \text{numTimes})}\ R2$

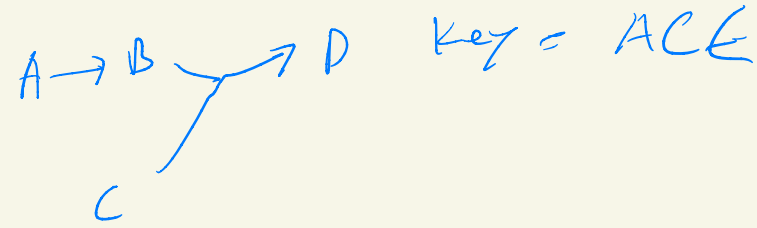
From these repeat shoppers, select those in the **20-30s** age group.

$R3 := \sigma_{\text{ageGroup} = \text{“20s-30s”}}(\text{Shopper} \bowtie \text{RepeatShopper})$

R(A, B, C, D, E)

A → B

BC → D




A → B $\{A\}^+ = \{A, B\}$

R1 (A, B)

R2 (A, C, D, E) ACE

A → B $\{AC\}^+ = \{A, C, B\}$

AC → D

R3 = ACD ✓

R4 = ACE ✓

A → BD
AB → C
C → D
BC → D

① Functional basis
1. ensure all RHS has only 1 attr
A → B
A → D
AB → C
C → D
BC → D
2. ensure no FDs are redundant
A → B ? $\{A\}^+ = \{A\}$ NO
with other FDs
A → D ? Yes A → B, AB → C, C → D $\{A\}^+ = \{A, B, C, D\}$
AB → C ? NO
C → D ? NO
BC → D Yes $\{BC\}^+ \{BCD\}$

S = {A → B
AB → C
C → D
3. Checks if we can remove multiple attr from LHS
AB → C is the only one FD
is A redundant ?
B → C $\{B\}^+ = \{B\}$
is B redundant ? ✓
A → C $\{A\}^+ = \{A, B, C, D\}$
A → B
A → C
C → D
② See if any FDs can be combine
A → BC
C → D

3) make tables out of FD
$R_1$ (A, B) $R_2$ (C, D) $R_3$ (A, C)
4) remove redundant table

TUT6

1) i) Select distinct Pname
From PROF, DEPT
Where PROF.Pname = DEPT.Dname
AND DEPT.Numphds < 50

ii) Select distinct S.Sname, M.Dname
From Student as S
JOIN Major as M on S.Sid = M.Sid
JOIN ENROLL as E on S.Sid = E.Sid
JOIN COURSE as C on C.Cno = E.Cno
and C.Cname = 'DatabaseSystem'

3) Select distinct S.Sid, S.Sname, S.GPA
From Student as S
Join
(Select distinct Sid, count(Cno)
From ENROLL as E
Where E.Dname = 'Civil Engineering'
Group by Sid
Having count(Cno) =
(Select count(Cname)
From Course
where Course.Dname = 'Civil Engineering')) as G
On G.Sid = S.Sid

**1.** Write SQL statements for the following queries.

The schema of a database containing university-type data is given below. Primary key is underlined for each relation.

```
STUDENT(Sid, Sname, Sex, Age, Year, GPA)
DEPT(Dname, Numphds)
PROF(Pname, Dname)
COURSE(Cno, Cname, Dname)
MAJOR(Dname, Sid)
SECTION(Dname, Cno, Sectno, Pname)
ENROLL(Sid, Grade, Dname, Cno, Sectno)
```

Write the following queries.
(1) Find the names of professors who work in departments that have fewer than 50 PhD students.
(2) Find the names and majors of students who have taken the 'Database System' course.
(3) Find the ids, names, and GPAs of the students who have taken all courses from the 'Civil Engineering' department.

3) i) Select Distinct A.ArticleID, I.IssueID
From Article as A
JOIN Issue AS I On A.IssueID = I.IssueID
JOIN WordAppears AsWA on WA.IssueID = I.IssueID
and WA.ArticleID = A.ArticleID
JOIN Words AS W on W.WordID = WA.WordID
Where (W.WordText Like '%Politician%' ) or
W.WordText (like '%Corruption%')

**2.** Suppose we are maintaining a database of articles published in our newspaper, the Straits Times. We have the following schema (where keys are underlined):

Article (issueID, articleID, author, title)
Citation (articleID, issueID, citedArticleID, citedIssueID)
WordAppears (wordID, issueID, articleID, position)
Words (wordID, wordText)
Issue (issueID, date, howManyDistributed)

For each of the following queries, write the query in SQL. Assume that dates can be compared using comparison operators (<, >, =). Assume that position is an index specifying where the word appears (1 = first word, 2 = second, etc.).

(i) Find the documents in which the words "politician" and "corruption" appear.

3)

**3.** For the following relational schema:

employee (employee-name, street, city)
works (employee-name, company-name, salary)
company (company-name, city)
manages (employee-name, manager-name)

Give an expression in SQL for each of the following queries:

i. Find the names of all employees in the database who live in the same cities and on the same streets as do their managers. Assume that all people work for at most one company. Each company has at most one manager, who is also an employee of the same company.

Select distinct
e1.employee-name as employee
e2.employee-name as manager
from employee as e1
Join manages as M on M.employee-name = e1.employee-name
Join employee as e2 on M.manager-name = e2.employee-name
and e1.street = e2.street and
e1.City = e2.City

# Tutorial 6
# SQL

## Classroom Exercise

## Question 1

(i)

```
SELECT      Pname
FROM        PROF, DEPT
WHERE       DEPT.Dname = PROF.Dname AND Numphds < 50;
```

(iii)

```
SELECT      Sname, Dname
FROM        COURSE C, ENROLL E, MAJOR M, STUDENT S
WHERE       C.Cname = 'Database Systems' AND
            C.Dname =   E.Dname AND C.Cno = E.Cno AND
            E.Sid = M.Sid  AND   E.Sid = S.Sid;
```

(iv)

```
SELECT      Sid, Sname, GPA
FROM        STUDENT S
WHERE       NOT EXISTS
                (SELECT     C.CID
                FROM        COURSE C
                WHERE       Dname = 'Civil Engineering'
                EXCEPT
                SELECT      E.CID
                FROM        ENROLL E
                WHERE       Dname = 'Civil Engineering'
                            AND E.Sid = S.Sid);
```

## Question 2

(i)

```
SELECT      DISTINCT wa1.issueID, wa1.articleID
FROM        WordAppears wa1, WordIs wi1, WordAppears wa2, WordIs wi2
WHERE       wa1.issueID = wa2.issueID AND wa1.articleID = wa2.articleID
            AND wa1.wordID = wi1.wordID AND wa2.wordID = wi2.wordID
            AND wi1.wordText = 'politician' AND
            wi2.wordText = 'corruption';
```

# Question 3

(b) Find the names of all employees in the database who live in the same cities and on the same streets as do their managers. Assume that all people work for at most one company. Each company has at most one manager, who is also an employee of the same company.

**Solution:**

```
SELECT p.employee-name
FROM employee p, employee r, manages m
WHERE p.employee-name = m.employee-name
AND m.manager-name = r.employee-name
AND p.street = r.street AND p.city = r.city;
```

TUT 7

1) a)
```
Select S.name, S.GPA
From STUDENT as S
Where S.GPA IN (select Min(GPA) from STUDENT);
```

b) AC → ABCD
D → AB<D

Stupid question

2) a)
```
Select CategoryName
From Category AS C
where C.[]Not IN
(Select distinct BelongsTo from Category);
```

b)
```
Select B.*
From Book AS B
Join
(Select distinct ISBN, Count(CopyNumber) AS CN
From Copy
where Copy.CopyNumber NOT IN
(Select CopyNumber from LOAN)
Group by ISBN) AS C
on B.ISBN = C.ISBN
AND C.CN > 1
```

2. Consider the following relational schema:

Reader( RDNR, Surname, Firstname, City, Birthdate )
Book( ISBN, Title, Author, NoPages, PubYear, PublisherName )
Publisher( PublisherName, PublisherCity )
Category( CategoryName, BelongsTo )
Copy( ISBN, CopyNumber, Shelf, Position )
Loan( ReaderNr, ISBN, Copy, ReturnDate )
BookCategory( ISBN, CategoryName )

BelongsTo refers to which parent categories the current category belongs to. Each book has a specific ISBN, and many copies of a book might be available under the same ISBN. A reader may borrow the same copy for multiple times, and each instance is recorded by its ReturnDate. All the[] categories that a book belongs to are stored in the table BookCategory.

Formulate the following queries in SQL.

(a) Which categories do not have any subcategories?

(b) For which of the books there is at least one copy available?

(c) Which books have more pages than twice the average of the number of pages of all books?

(d) What are the surnames of the readers from the city "New York"?

c)
```
Select B.*
From Book AS B
Where B.NoPages >= 2*
(Select AVG(NoPages)
From Book)
```

d)
```
Select distinct
Surname
From Reader AS R
Where
R.City = 'New York'
AND R.RDNR is in
(Select ReaderNr from
Loan);
```

3) a)
```
Select employee-name
From works W
Join
(Select distinct
Company-name, AVG(Salary) AS AS
from works
Group by Company-name) AS AVGS
On W.company-name = AVGS.company-name
and W.salary > AVGS.AS;
```

3. For the following relational schema:

employee (employee-name, street, city)
works (employee-name, company-name, salary)
company (company-name, city)
manages (employee-name, manager-name)

Give an expression in SQL for each of the following queries:

(a) Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company.

4) i)
```
Select citedArticleID, citedIssueID, Max(X)
From
(Select
citedArticleID, citedIssueID, Count(articleID) as X
From Citation
Groupby CitedarticleID, CitedIssueID)
group by citedArticleID, citedIssueID
```

4. Suppose we are maintaining a database of articles published in our newspaper, the Straits Times. We have the following schema (where keys are underlined):

Article (issueID, articleID, author, title)
Citation (articleID, issueID, citedArticleID, citedIssueID)
WordAppears (wordID, issueID, articleID, position)
WordIs (wordID, wordText)
Issue (issueID, date, howManyDistributed)

For each of the following queries, write the query in SQL. Assume that dates can be compared using comparison operators (<, >, =). Assume that position is an index specifying where the word appears (1 = first word, 2 = second, etc.).

(ii) Find the most-cited article(s) in the newspaper's history.

(iii) Find the number of citations per author for "senior" authors (i.e., an author who has at least one article that was published 10 or more years ago).

ii)
```
Select a.author, Count(*)
From Article as A
JOIN Citation as CA
on CA.articleID = a.articleID
and CA.issueID = a.issueID
where A.author in (Select distinct author
from Article JOIN Issue on Article.issueID = Issue.issueID
where Issue.(Year(getDate()) - Year(issueDate)
>=10 );
```

# Tutorial 7
# SQL

## Classroom Exercise

## Question 1

(a)

**(i)**

```
SELECT   Sname
FROM     STUDENT
WHERE    GPA IN
           (SELECT MIN(GPA) FROM STUDENT);
```

(b)

This query will report error. The reason is the subquery returns a scalar, and therefore cannot be compared against a single data value. You would need to add "ANY" or "ALL" before the subquery for the query to run.

## Question2

(a) Which categories do not have any subcategories?

**Solution:**

```
SELECT C1.CategoryName
FROM Category C1
WHERE NOT EXISTS
(SELECT CategoryName
FROM Category C2
WHERE C2.BelongsTo = C1.CategoryName);
```

(b) For which of the books there is at least one copy available?

**Solution:**

```
SELECT Title
FROM Book
WHERE ISBN IN
(SELECT ISBN FROM
((SELECT CopyNumber, ISBN FROM Copy)
EXCEPT
(SELECT Copy, ISBN FROM Loan)));
```

An assumption here is that loan table only records books that are on loan; once returned, the record is removed from Loan table.

(c) Which books have more pages than twice the average of the number of pages of all books?

**Solution:**

```
SELECT ISBN
FROM Book
WHERE NumberOfPages>= 2* (SELECT AVG(NumberOfPages)
                          FROM Book);
```

(d) What are the surnames of the readers from the city "New York"?

**Solution:**

```
SELECT DISTINCT Surname
FROM Reader
WHERE City = 'New York'
```

## Question 3

(a) Find the names of all employees who earn more than the average salary of all employees of their company. Assume that all people work for at most one company.

**Solution:**

```
SELECT employee-name
FROM works t
WHERE salary > (SELECT AVG(salary)
                FROM works s
                WHERE t.company-name = s.company-name);
```

---

## Question 4

```
SELECT      citedIssueID, citedArticleID
FROM        Citation
GROUP BY    citedIssueID, citedArticleID
HAVING COUNT(*) >= ALL
                    (SELECT     COUNT(*)
                    FROM        Citation
                    GROUP BY    citedArticleID, citedIssueID)
```

(iii) return author and number of references/citations

```
SELECT      a.author, COUNT(*)
FROM        Article a, Citation c
WHERE       a.issueID = c.citedIssueID AND a.articleID = c.citedArticleID
AND EXISTS
                    (SELECT     *
                    FROM        Article a2, Issue i
                    WHERE       a2.issueID = i.issueID AND
                                a2.author = a.author AND
                                Year(getDate()) – Year(i.date) >= 10)
GROUP BY a.author;
```

TUT 8

1) i) Update employs
Set street = 'Newstreet'
AND city = 'Newtown'
where person-name = 'Mark'

ii) Update works
Set salary = salary + (0.1 * salary)
where company-name = 'FaceMatch'

iii) Update works
Set salary = salary + (0.1 * salary)
where
company-name = 'FaceMatch'
AND person-name IN (Select Distinct manager-name from manages);

**Classroom Exercise**

1. The relational database schema for a Car-insurance company is given below.

person (driver-id, name, address)
car (license, year, model)
accident (report-number, location, date)
owns (driver-id, license)
participated (report-number, driver-id, license, damageamount)
employee (person-name, street, city)
works (person-name, company-name, salary)
company (company-name, city)
manages (person-name, manager-name)

(i) Modify the database so that Mark now lives in "Newstreet, Newtown" (i.e. the street changes to Newstreet and city changes to Newtown for Mark).

(ii) Give all employees of "FaceMatch" a 10 percent salary raise.

(iii) Give all managers of "FaceMatch" a 10 percent salary raise.

2) i) Yes Same size
ii) No

2. Consider the relation schema: STUDENT (name, age, majorcode, gpa).
Suppose the following view is defined on it: STUDENT-NO-GPA(name, age, majorcode).

(i) Is it possible to insert a tuple ('X', 123, 'CMSC') into this view? If yes, what should be the result of it? If not, why not?

(ii) Now consider the following view.
CREATE VIEW MAJOR-AVG AS
SELECT majorcode, avg(gpa) AS avggpa
FROM STUDENT
GROUP BY majorcode;
Is it possible now to insert a tuple ('CMSC', 3.2) into this view? If yes, what should be the result of it? If not, why not?

3) i) CREATE ASSERTION A1 CHECK
(NOT EXIST
(Select P.color
From Catalog AS C
JOIN Parts AS P on P.p.id = C.pid
where P.color IN ('red', 'green')
)

3. Consider the following schema of a product database:
Parts(pid, integer, pname: string, color: string)
Suppliers(sid, integer, sname: string, address: string)
Catalog(pid, integer, sid integer, price: real)

The Catalog records that some Supplier sid supplies Part pid at a given price. Formulate each of the following integrity constraints as an SQL assertion.

(i) No Supplier may supply red and green Parts.

(ii) For all Parts, no other Supplier has a lower price than the Supplier with "sid" = 1.

ii) CREATE ASSERTION A2 CHECK
(NOT EXIST
(Select *
From Catalog AS C
where C.price < (Select price
From Catalog
where sid = 1))
)

4. Consider the relation R (A, B, C, D, E). Write one or more CREATE TRIGGER statement(s) or SQL92 that can be used to impose an FD constraints AB→C.

5. An airline stores information about its flights and ticket sales in a relational database with the following schema (primary keys are underlined):

FLIGHT (FLIGHTNO, DAY, MONTH, YEAR, NUMSEATS, PRICE)
PASSENGERS (PASSENGERID, NAME, ADDRESS)
BOOKING (REF#, FLIGHTNO, DAY, MONTH, YEAR, PASSENGERID)

The (FLIGHTNO, DAY, MONTH, YEAR) attribute of BOOKING is a foreign key into the FLIGHT relation. Also, PASSENGER attributes of BOOKING is a foreign key into the PASSENGERS relation. You can assume the number of seats for a flight must be between 50 and 200. When details of a new flight are added to the FLIGHT relation, NUMSEATS is initialized to zero by default.

Whenever there are fewer than 10 seats remaining on a flight after a booking, then the airline's policy is to set the price of remaining seats on that flight to 8000.50. Write a trigger to implement this policy. You may assume that bookings cannot be cancelled.

4) Create TRIGGER T1
AFTER UPDATE ON R
REFERENCING NEW ROW AS NR
OLD ROW AS OR

FOR EACH ROW
WHEN ((NR.C <> OR.C) and ((NR.A = OR.A) AND (NR.B = OR.B))
DELETE FROM R
WHERE A = OR.A AND
B = OR.B

6. A database system used by a university's examinations office has the following relations:

Exams (course, examDate, examTime)
Students (studentid, name)
registeredFor (student, course, examDate)

Attribute course is a six-character course code. A course cannot have more than one exam on the same date, and that exam will either be in the morning ('AM') or in the afternoon ('PM'). Attribute examDate is a date (e.g. '2014-12-18') and examTime is either 'AM' or 'PM'.

If a clash occurs for a student, special arrangements will be made for the student to sit the exam at another time on the same date. For example, if the clash is with another 'AM' exam, then a special exam will be arranged for that student in the afternoon ('PM'), and vice versa.

We tie a trigger in SQL99 that, when a student registers for an exam that is at the same time on the same date as another exam for which they are already registered, adds a row to relation SevereExams(student, course, examDate, examTime). Here, examTime should be the time of the (clash) arranged exam ('AM' or 'PM').

A row should be added to the registeredFor relation even when the student will sit a special exam.

6) CREATE TRIGGER S
AFTER INSERT INTO registeredFor
REFERENCING
NEW ROW AS NR
FOR EACH ROW
WHEN (
Exist
(Select S.*, E.examTime
FROM registeredFor AS S
JOIN Exams as E on S.Course = E.Course
where S.Student = NR.Student
AND S.examDate = NR.examDate
AND S.Course <> NR.Course
)

Begin
INSERT INTO SpecialExams
Values (
NR.Student,
NR.Course,
NR.examDate,
(Select examTime
From Exams As e
WHERE
E.Course = NR.Course
e.examDate = NR.examDate
e.examTime NOT IN
(Select examTime
From registeredFor As r
JOIN Exams AS E
on r.Course = E.Course
WHERE r.Student = NR.Student AND
r.examDate = NR.examDate AND
r.Course <> NR.Course
)
);

5) CREATE TRIGGER T1
AFTER UPDATE ON FLIGHT
REFERENCING
FOR EACH ROW NEW ROW AS NR
WHEN NR.NUMSEAT < 20
BEGIN
UPDate FLIGHT
Set Price = 4000
WHERE FLIGHTNO = NR.FLIGHTNO
AND DAY = NR.DAY
AND Month = NR.Month
AND YEAR = NR.YEAR
END

CREATE TRIGGER T2
AFTER INSERT INTO FLIGHT
REFERENCING
NEW ROW AS NR
FOR EACH ROW
WHEN (NR.NUMSEATS < 50 OR NR.NUMSEAT > 200)
DELETE FROM FLIGHT
WHERE NR.FLIGHTNO = FLIGHTNO

CREATE TRIGGER T3
AFTER INSERT INTO FLIGHT
REFERENCING
NEW ROW AS NR
FOR EACH ROW
WHEN (NR.PRICE is NULL)
UPDATE FLIGHT
Set Price = 2000
where FLIGHTNO = NR.FLIGHT NO

TUT 9

1)

## Classroom Exercise

**1.** Consider the following relation:

Articles (ID, title, journal, issue, year, startpage, endpage, TR-ID)

It contains information on articles published in scientific journals. Each article has a unique ID, a title, and information on where to find it (name of journal, what issue, and on which pages). Also, if results of an article previously appeared in a "technical report" (TR), the ID of this technical report can be specified. We have the following information on the attributes:

- For each journal, an issue with a given number is published in a single year.
- The endpage of an article is never smaller than the startpage.
- There is never (part of) more than one article on a single page.

Consider the following six queries on Articles:

a. SELECT title FROM Articles WHERE year=2005;

b. SELECT title FROM Articles WHERE endpage=100;

c. SELECT title FROM Articles WHERE year>1995 AND year<2000;

d. SELECT title FROM Articles WHERE journal='JACM' AND issue=55;

e. SELECT title FROM Articles WHERE issue=55 AND journal='JACM';

f. SELECT title FROM Articles WHERE endpage-startpage>50;

Indicate which of the above queries would likely be faster (based on the knowledge you have from the course), if all of the following indexes were created.

- CREATE INDEX Idx1 ON Articles (year, startpage);
- CREATE INDEX Idx2 ON Articles (startpage, endpage);
- CREATE INDEX Idx3 ON Articles (journal, issue, year);

2)

**2.** Consider the following XML DTD for an employee database.

```
<!DOCTYPE emp [
<!ELEMENT emp (ename, children*, skills*)>
<!ELEMENT children (name, birthday)>
<!ELEMENT birthday (day, month, year)>
<!ELEMENT skills (type, exams+)>
<!ELEMENT exams (year, city)>
<!ELEMENT ename( #PCDATA )>
<!ELEMENT name( #PCDATA )>
<!ELEMENT day( #PCDATA )>
<!ELEMENT month( #PCDATA )>
<!ELEMENT year( #PCDATA )>
<!ELEMENT type( #PCDATA )>
<!ELEMENT city( #PCDATA )>
] >
```

Create a valid XML document that follows the rules of the above DTD.

Your document must illustrate all the rules of the DTD.

**3.** Consider the following relational database schema about students and courses (primary keys are underlined):

STUDENT(SID, NAME, EMAIL)

COURSE(CID, NAME, INSTRUCTOR, ROOM)

ENROLLS(SID, CID, GRADE)

In ENROLLS, SID and CID are foreign keys into STUDENT and COURSE, respectively. The following two constraints also hold: (a) each student is enrolled in some course, and (b) each course has at least one student enrolled.

(i) Design a DTD for exporting this data as an XML view to the chair's office. The chair wants to see the data grouped by courses and needs to have access to all the information in the database.

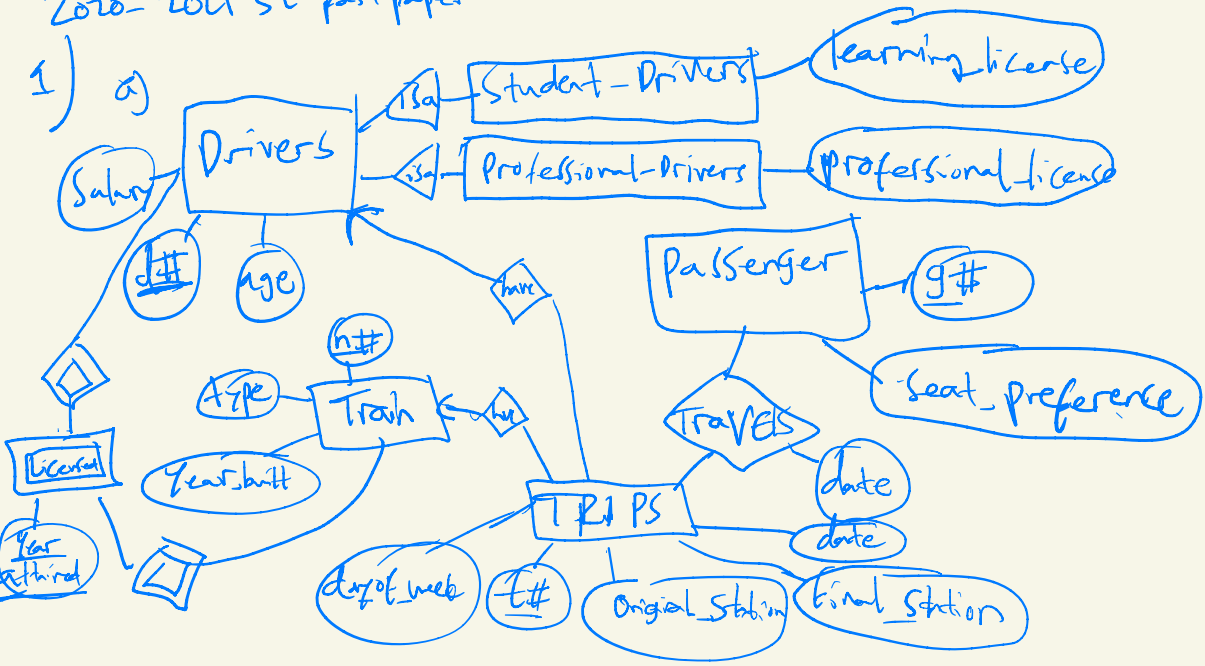
(ii) Give an example of a database instance for the relational schema and show the resulting document for the XML view according to the DTD that you have designed.

Past year

Paper

2020-2021 S2 Past Paper

1) a)




b) $R_1 := \sigma_{year\_built=2010}\ TRAIN$

$R_2 := (\sigma_{year\_attained<2018}\ LICENSED) \bowtie R_1$

$R_3 := (LICENSED \bowtie R_1) - R_2$

$R_4 := R_3 \bowtie R_1$

$R_5 := STUDENT\text{-}DRIVERS \bowtie R_4$

c) TRIPS

$R_1 := \sigma_{day\ of\ week='Monday'\ AND\ Original\_Station='Jurong\ St'}$

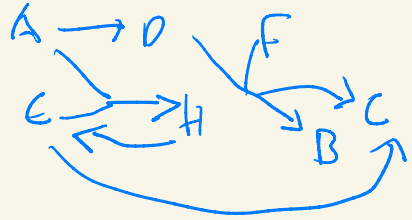
$R_2 := (\sigma_{age=50}\ DRIVERS) \bowtie R_1$

$R_3 := \pi_{d\#}(R_2)$

$R_4 := R_3$

$R_5 := R_3 \bowtie_{R_3.d\# <> R_4.d\#} R_4$

2) a)




keys : AFE, AFH

b)

i) $EC \to AD$ $\{EC\}^+ = \{E, C\}$ no

ii) $ADF \to E$ $\{ADF\}^+ = \{A, D, F, B, C\}$ no

iii) $A \to DH$ $\{A\}^+ = \{A, D\}$ no

iv) $AED \to C$ $\{AED\}^+ = \{A, E, D, H, C\}$ yes

v) $DH \to C$ $\{DH\}^+ = \{D, H, E, C\}$ yes

c) $A \to D$ ✓

$AE \to H$ ✓

$DF \to BC$ ✓

$E \to C$ ✓

$H \to E$ ✓

R3 (A, B, E, F, H) AFE or AFH

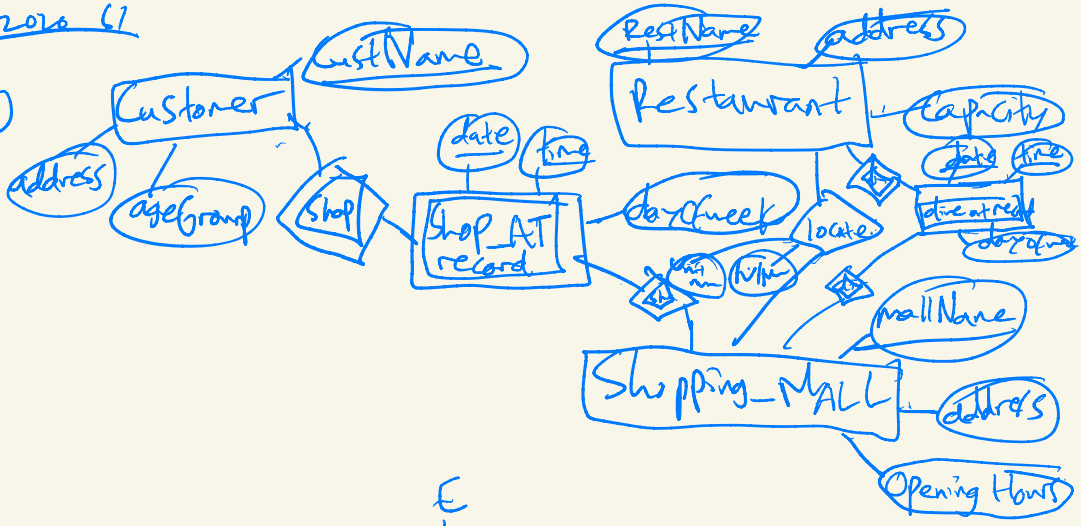
not BCNF due to ~~H~~ → E.

3NF yes

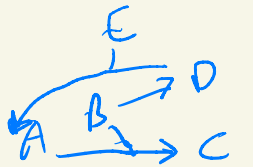
3)

2019-2020 CI

1) a)




2) $AB \to C$
$B \to D$
$DE \to A$




key BE

a) no as the RHS of FDs given is not in primary keys

b) $AB \to C$ RHS not in primary key BE

$\{AB\}^+ = \{A, B, C\}$

$R_1 \{A, B, C\}$ $R_2 \{A, B, D, E\}$

c) lossless as both table still has B which is part of BE

dependency preserving

$R_1 \{A, B, C\}$ $AB \to C$ ✓ $R_1$ yes

$B \to D$ ✓

$R_2 \{A, B, D, E\}$ $DE \to A$ ✓