UNIT – I

Chapter

ER Diagrams

Introduction:

- Definition of ER Diagram
- Entity Types
- Entity Sets
- Attributes and Keys
- Relationships, Relationship types
- Roles and Structural Constraints
- Weak Entity Types
- Notations
- Naming Conventions and Design Issues
- An Example Database Application;

ER MODEL

- It is a popular high level conceptual data model.
- It is used to relate one database with another.
- Here a design is created to understand the connection between two or more databases.

- Example : Student database
- College database
- Relation is :Students studies in college.

ER MODEL

- Example : Employee database
- Company database
- Relation is :Employee works in college.

- Example : Patient database
- Hospital database
- Relation is :Patient is admitted in hospital.

ER MODEL

- Example : Doctor database
- Hospital database
- Relation is :??????.

- Example : Subject database
- Staff database
- Relation is :???????.

ER MODEL

- Example : Doctor database
- Hospital database
- Relation is :Doctor works in hospital.

- Example : Subject database
- Staff database
- Relation is :Subject is taught by Staff.

Using High-Level Conceptual Data Models for Database Design

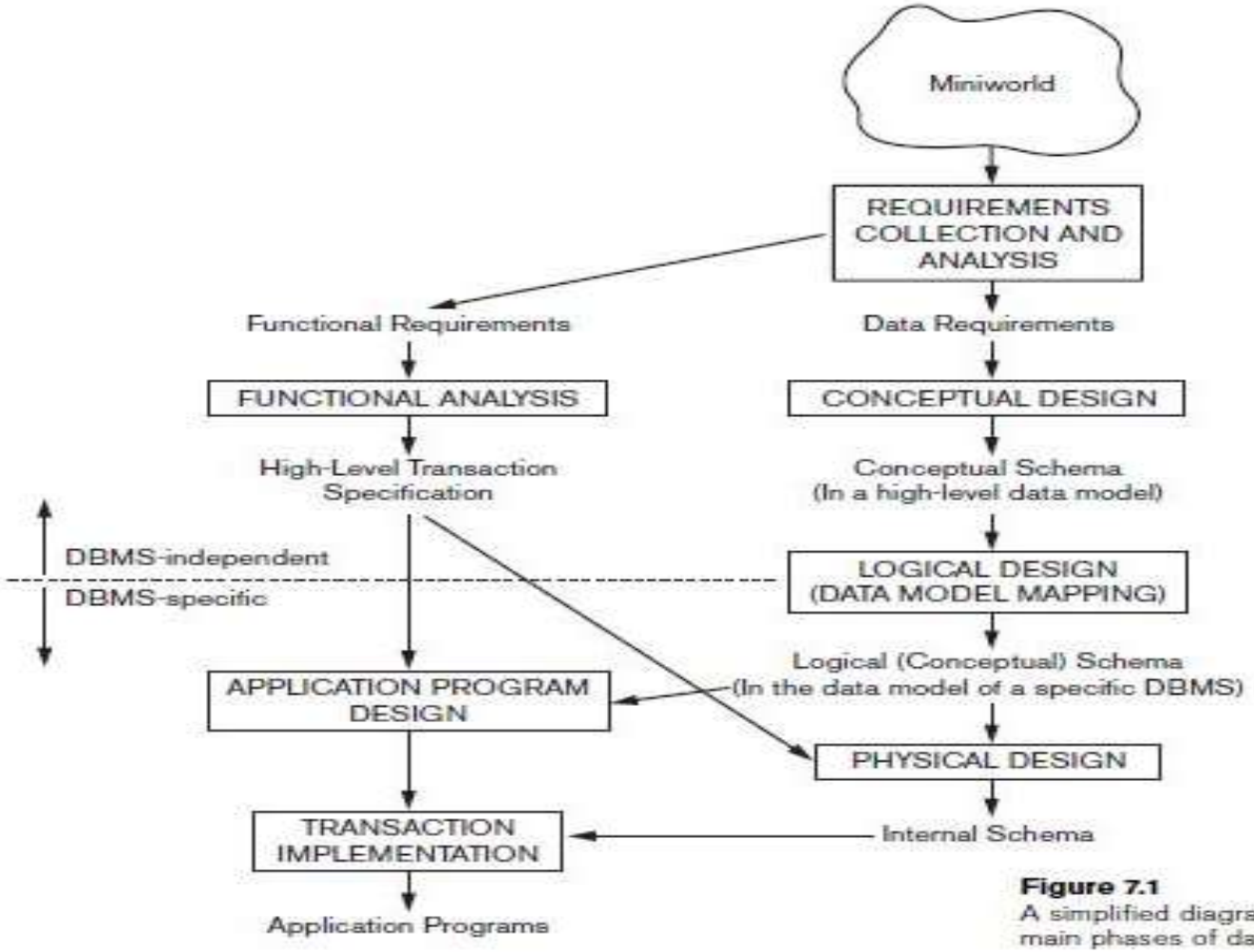


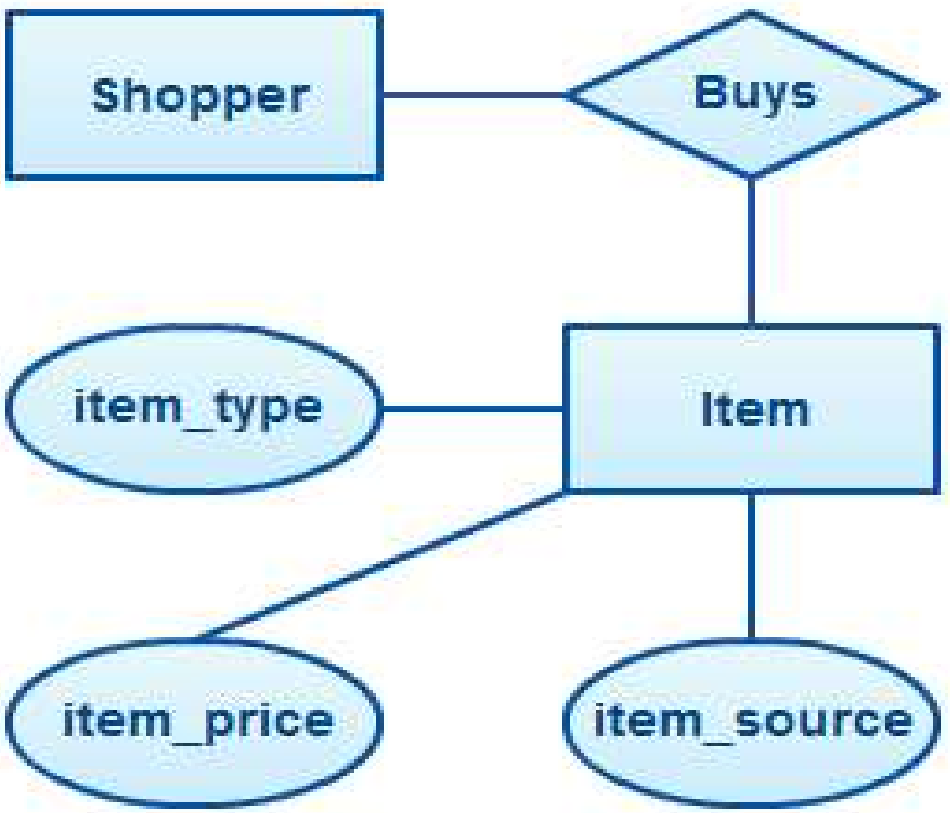
Figure 7.1
A simplified diagram to illustrate the main phases of database design.

ER diagram for small shop

- Shopper
- Shop
- Items in Shop

ER diagram for Shop

- Shopper
- Shop
- Items in Shop



- **Entity**

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

- **Attributes**

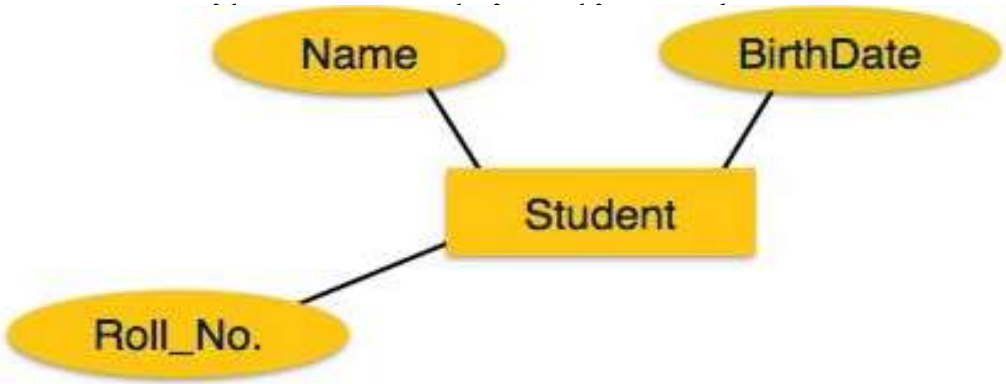
Entities are represented by means of their properties, called **attributes**. All attributes have values. For example, a student entity may have name, class, and age as attributes.

- **Entity**

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

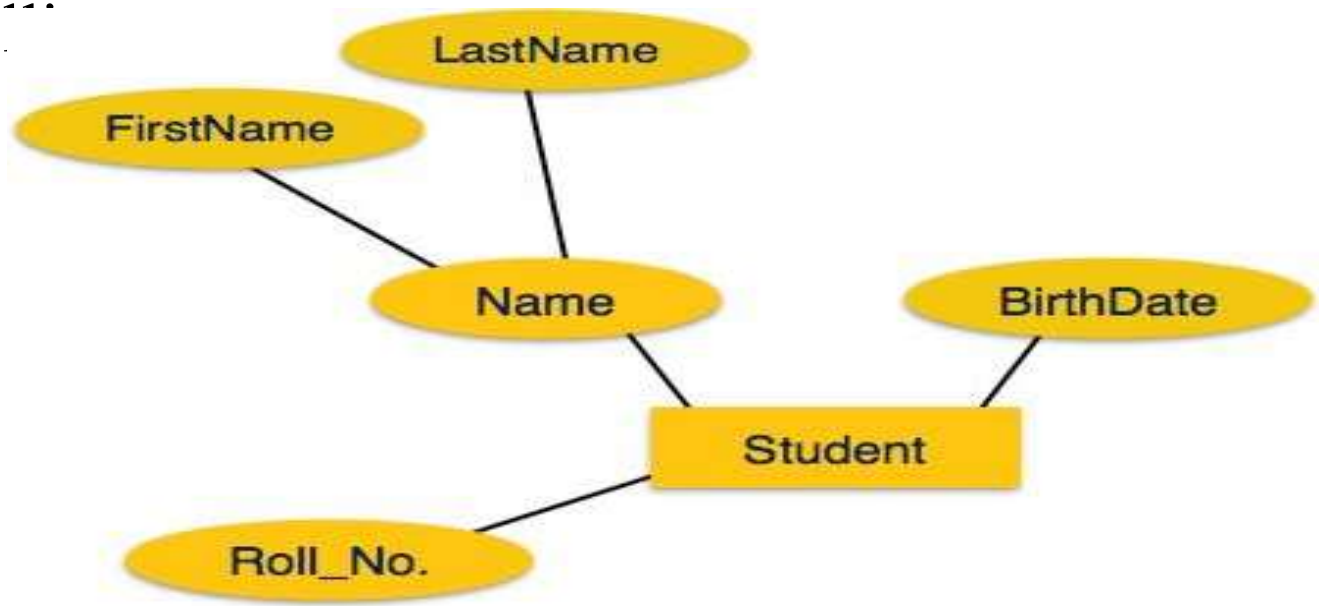
- **Attributes**

Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents ed to its entity (rect

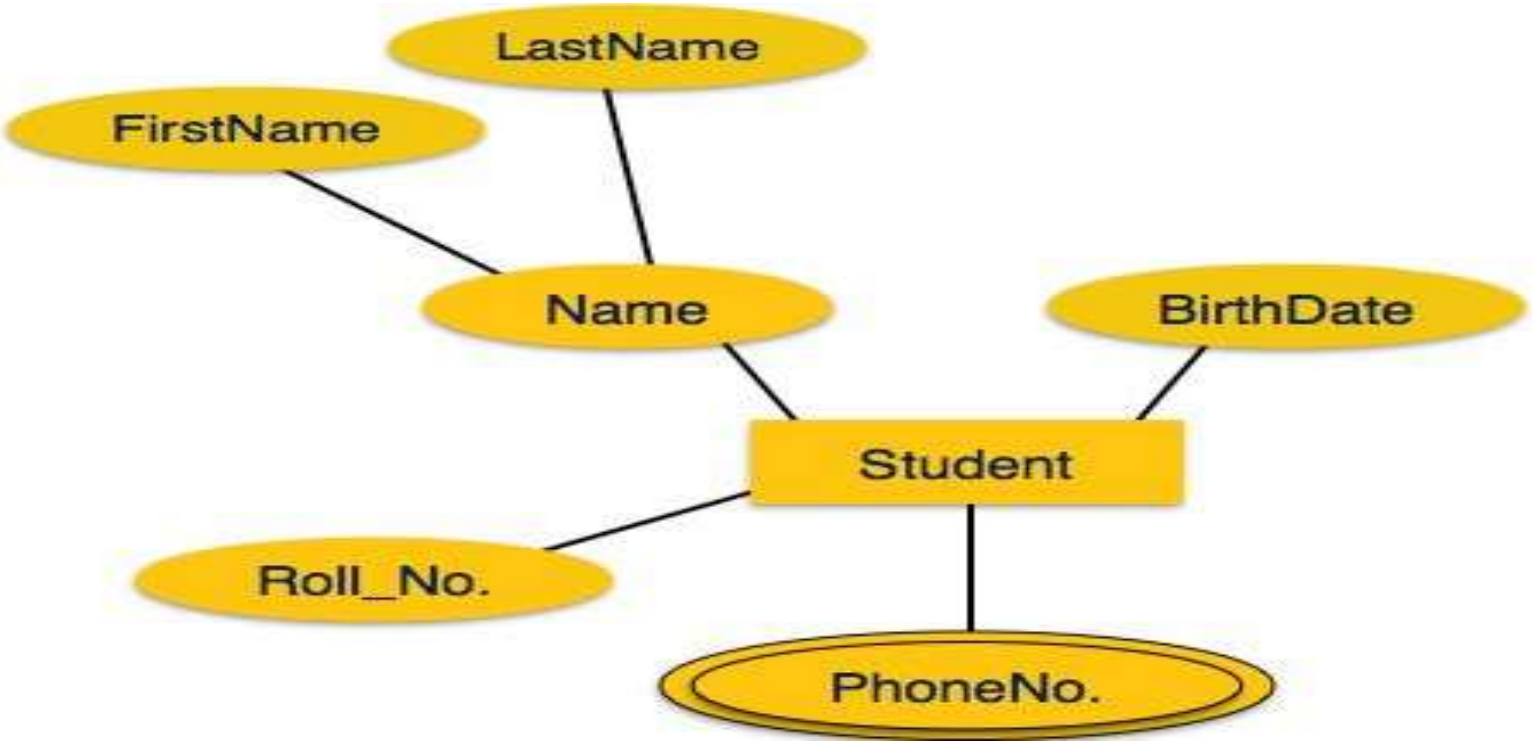


- Types of Attributes
- **Simple attribute** − Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- **Composite attribute** − Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.
- **Derived attribute** − Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For another example, age can be derived from data_of_birth.
- **Single-value attribute** − Single-value attributes

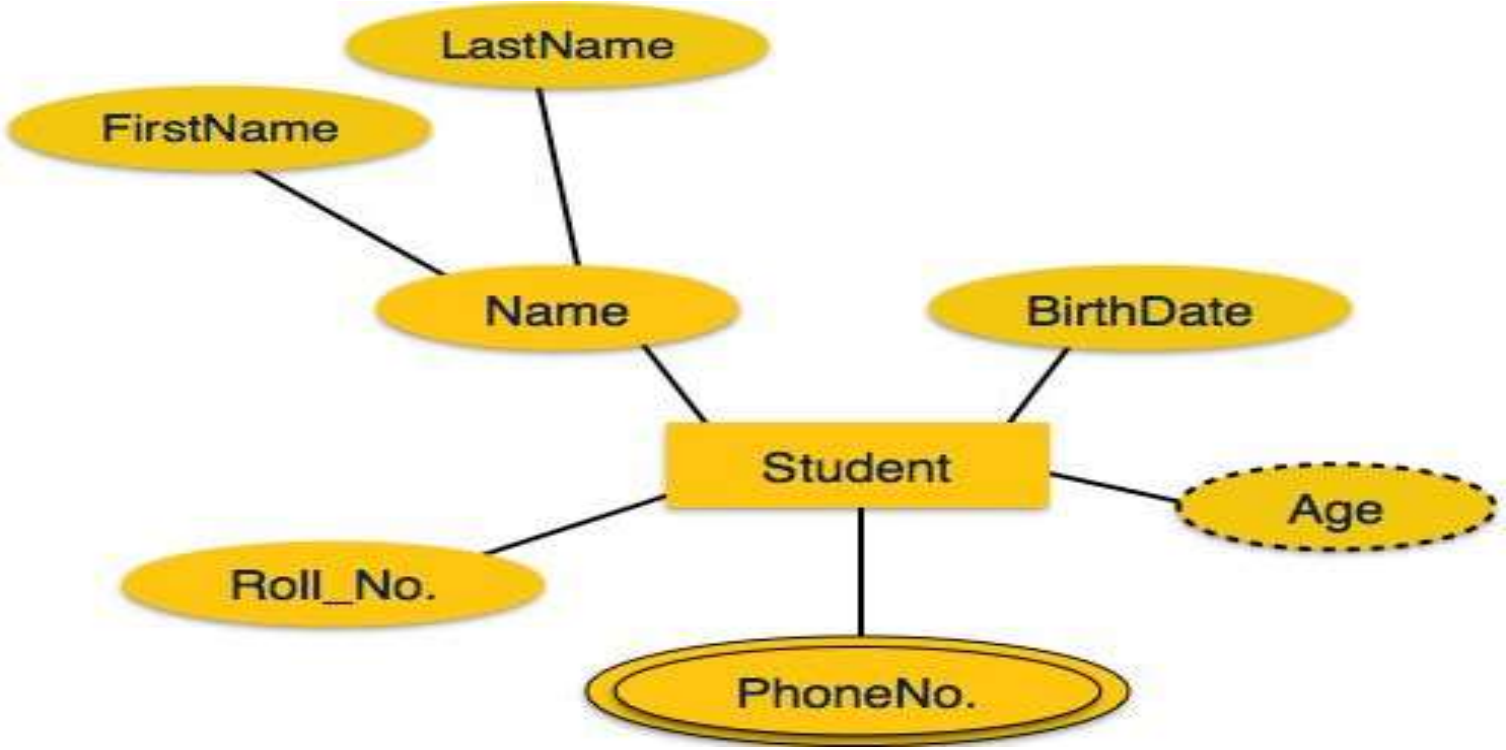
- If the attributes are **composite**, they are further divided in a tree like structure. Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an e



- **Multivalued** attributes are depicted by double ellipse.



- **Derived** attributes are depicted by dashed ellipse

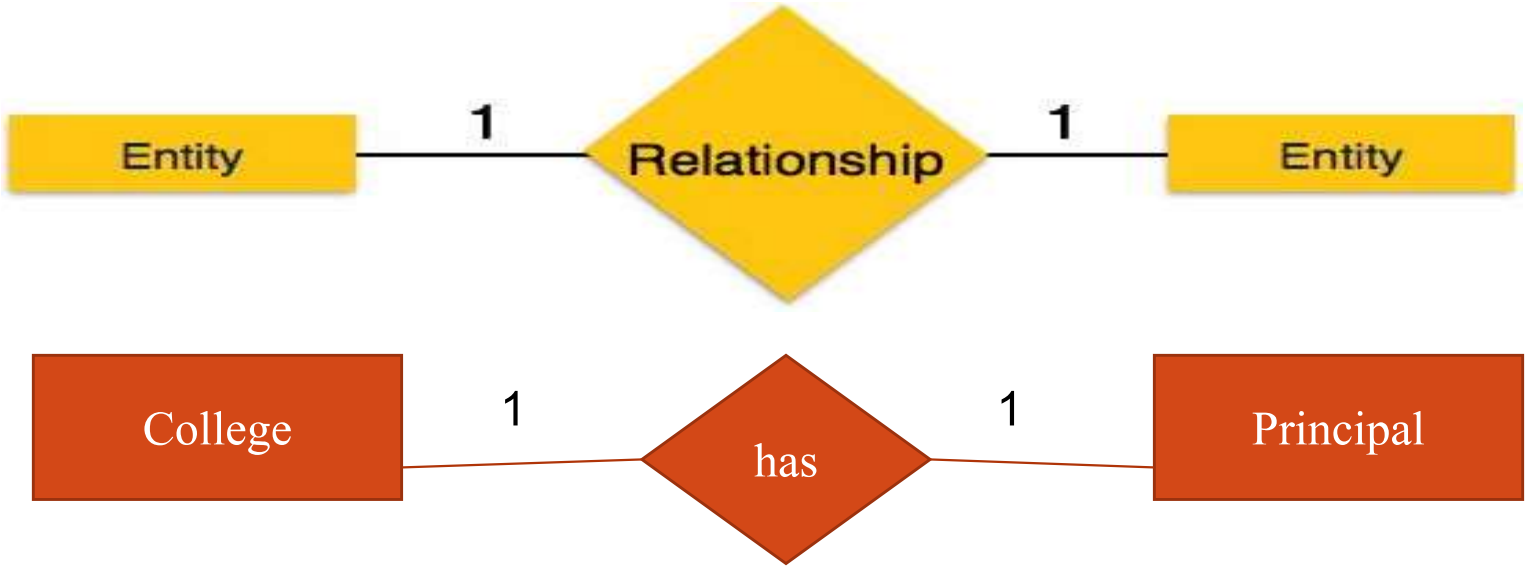


Relationship

- Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.
- Different types of relationship:
- **One-to-one** –
- **One-to-many** –
- **Many-to-one** –
- **Many-to-many** –

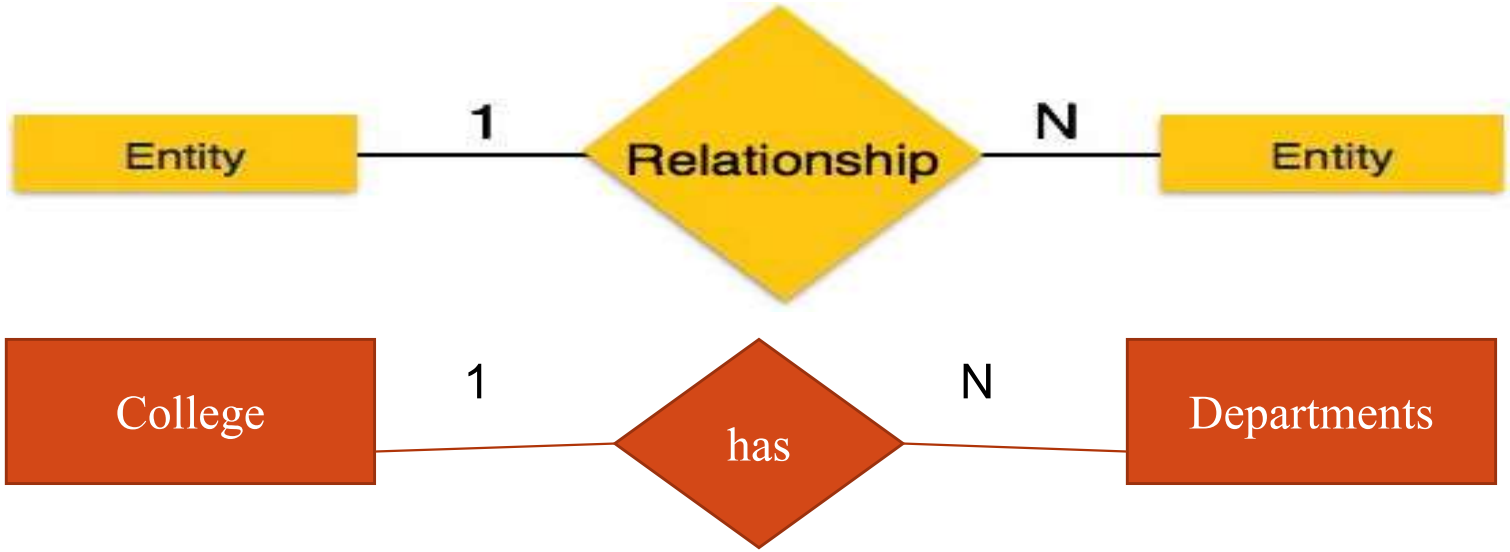
- **One-to-one** − When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.

- **One-to-one** − When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.

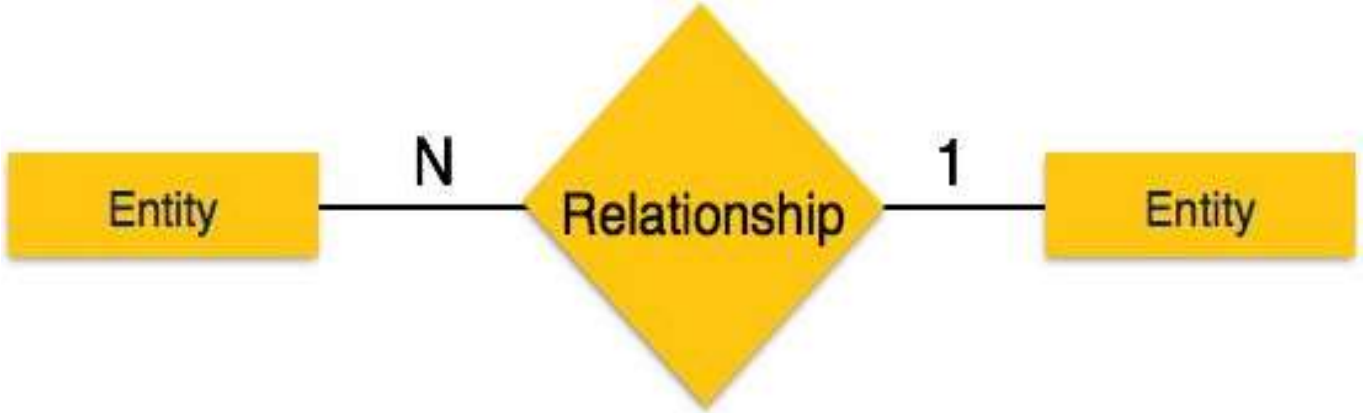


- **One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.

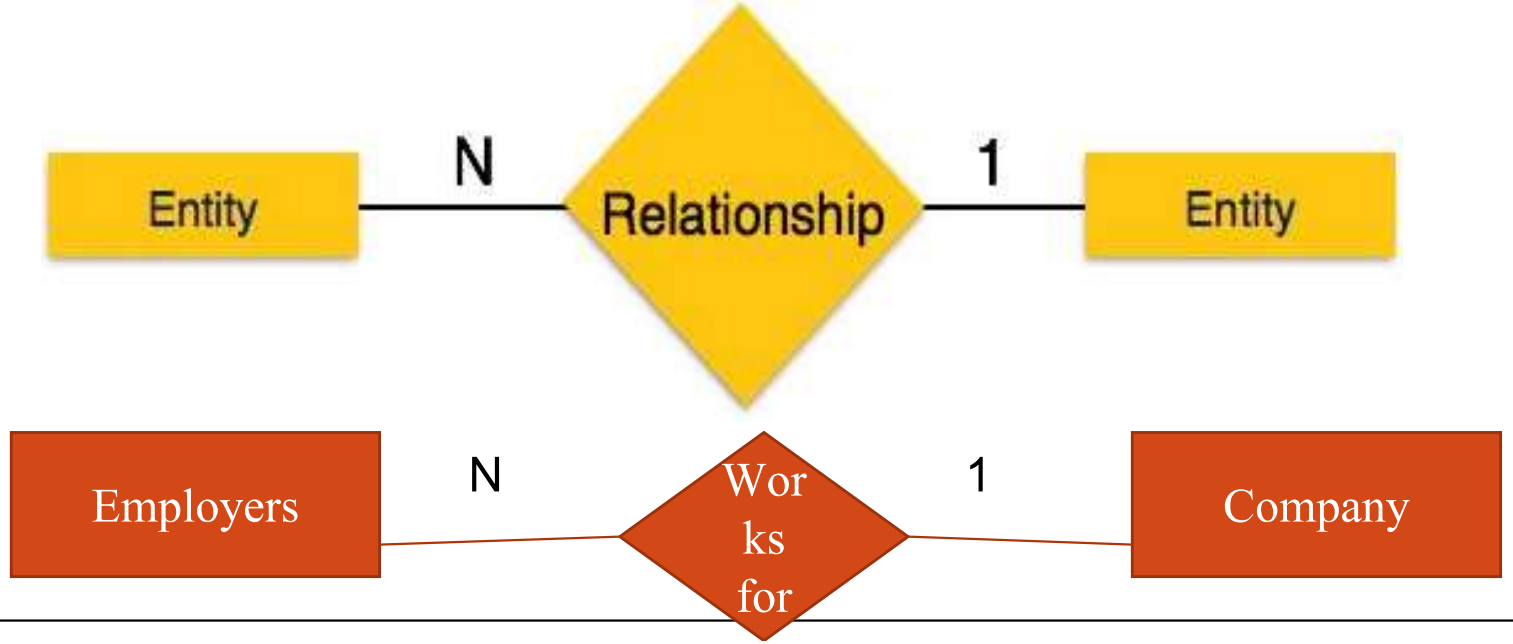
- **One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.



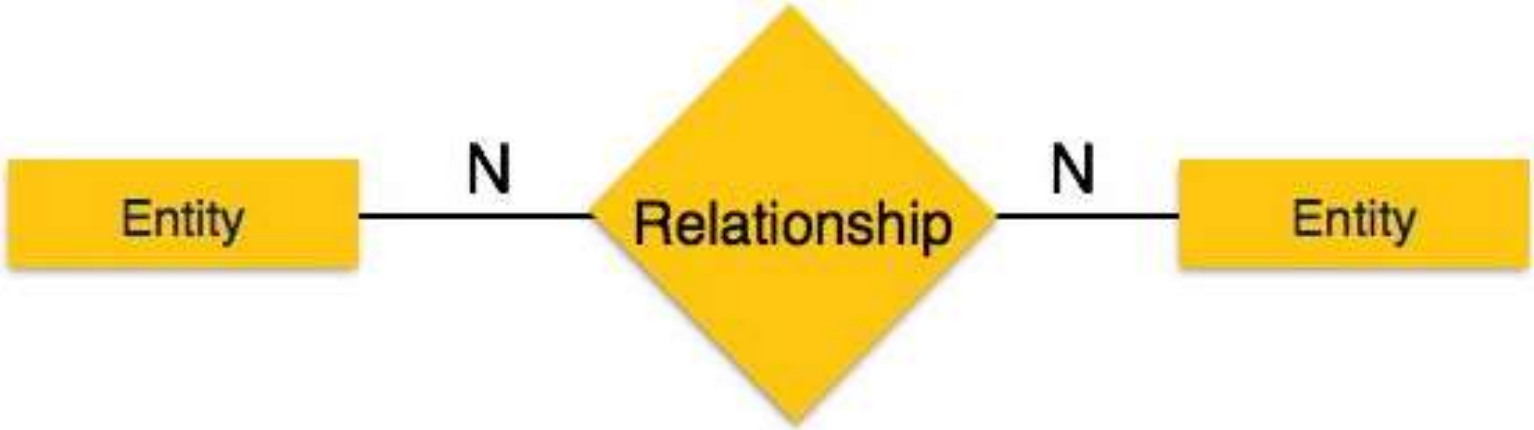
- **Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.



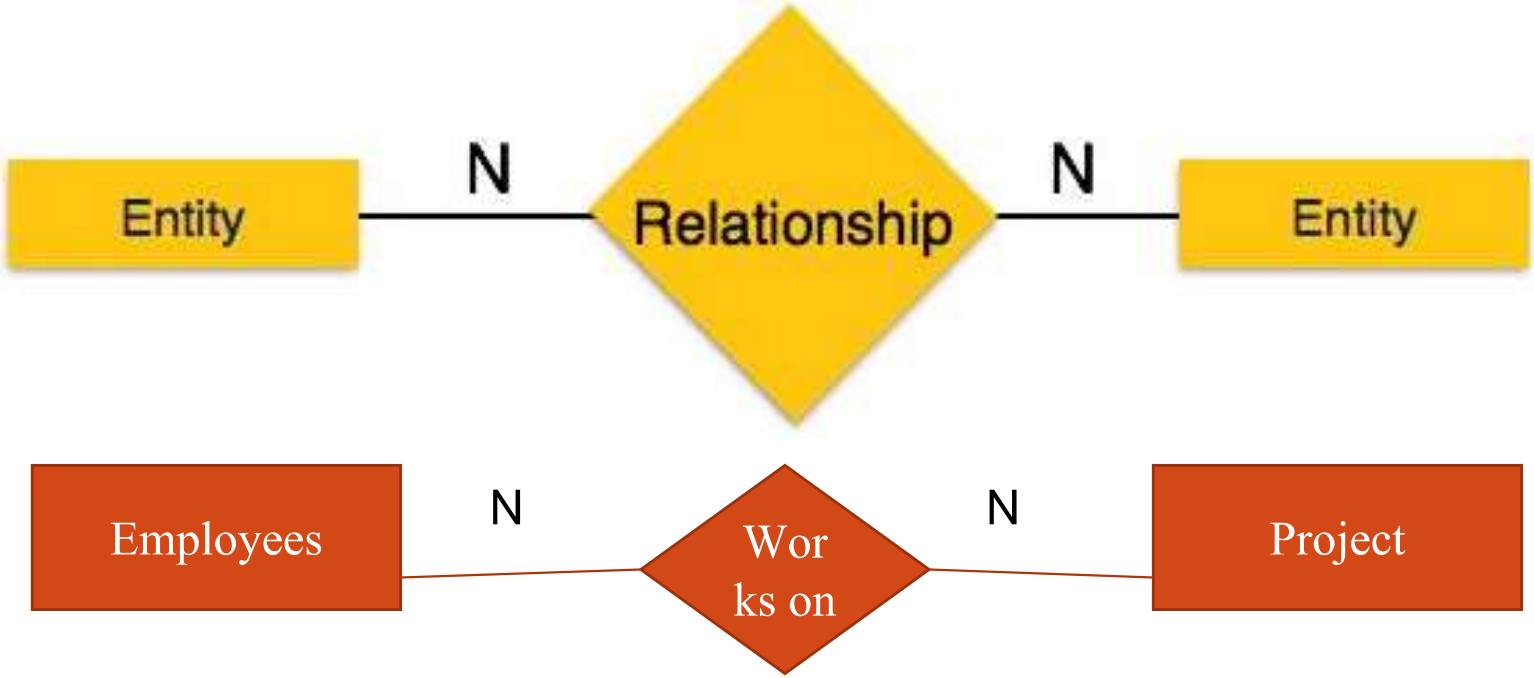
- **Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.



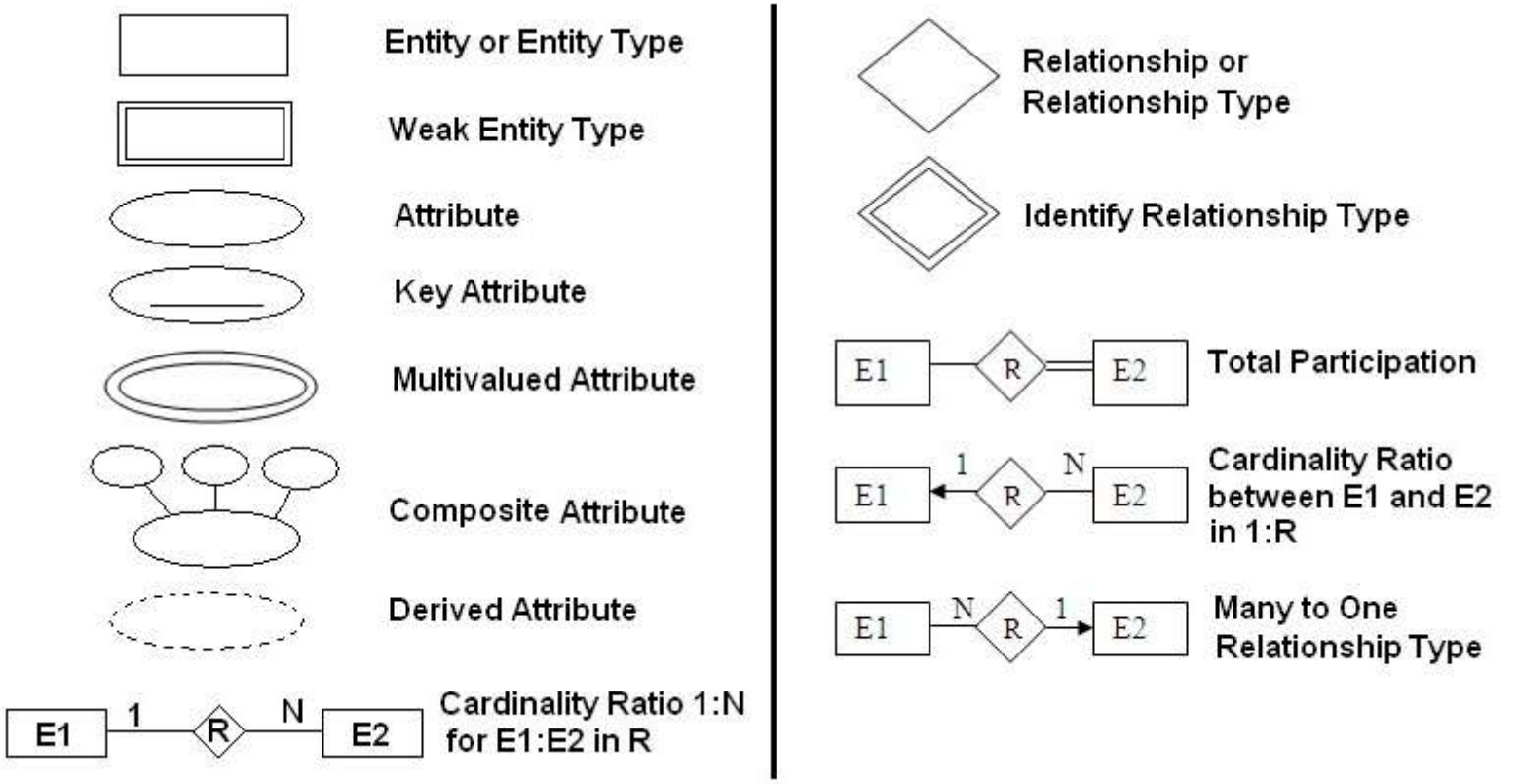
- **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts



- **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts



Symbols and Notations



Lets Design a Company Database

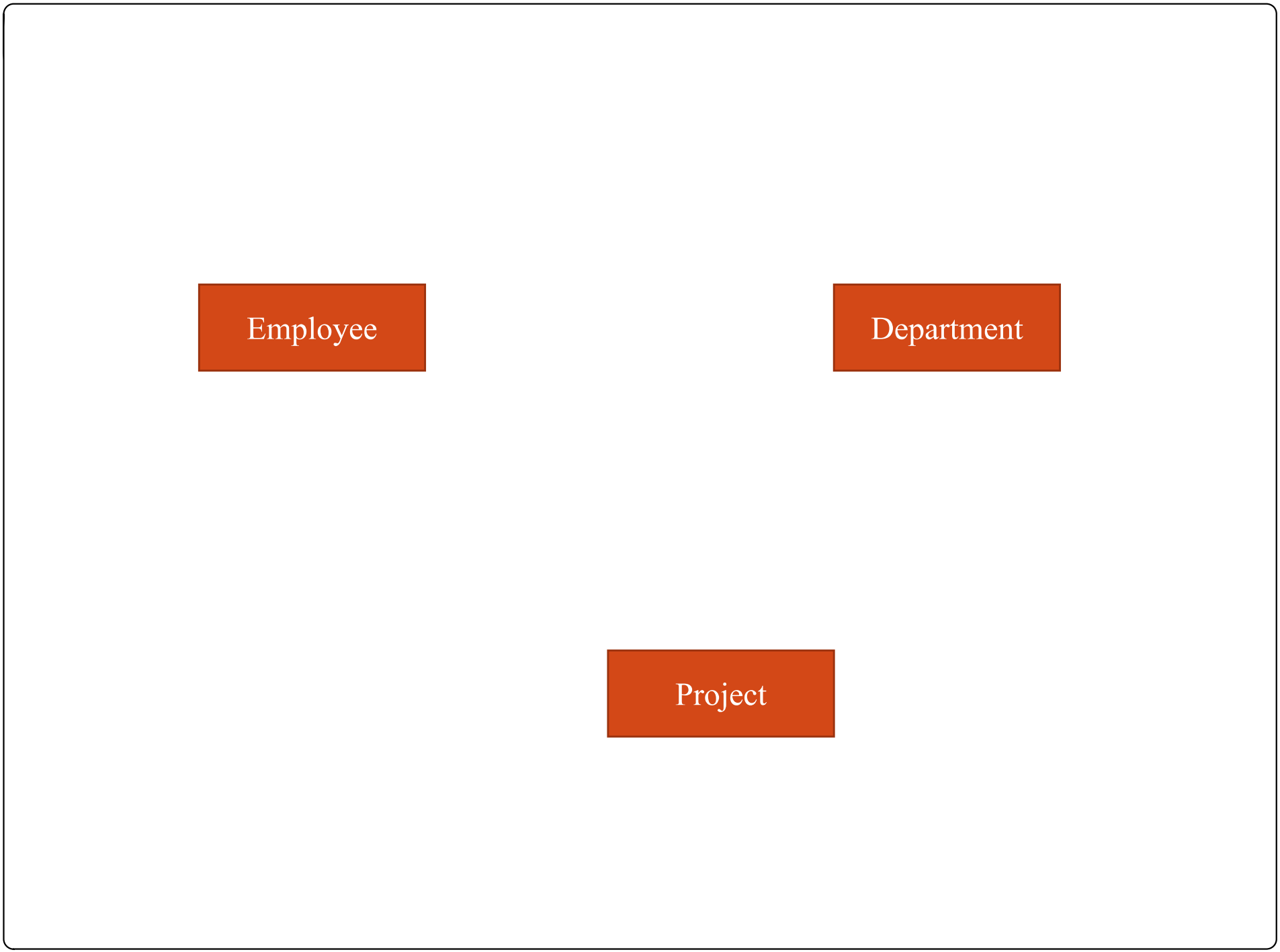
- First task is to identify the entities…

Lets Design a Company Database

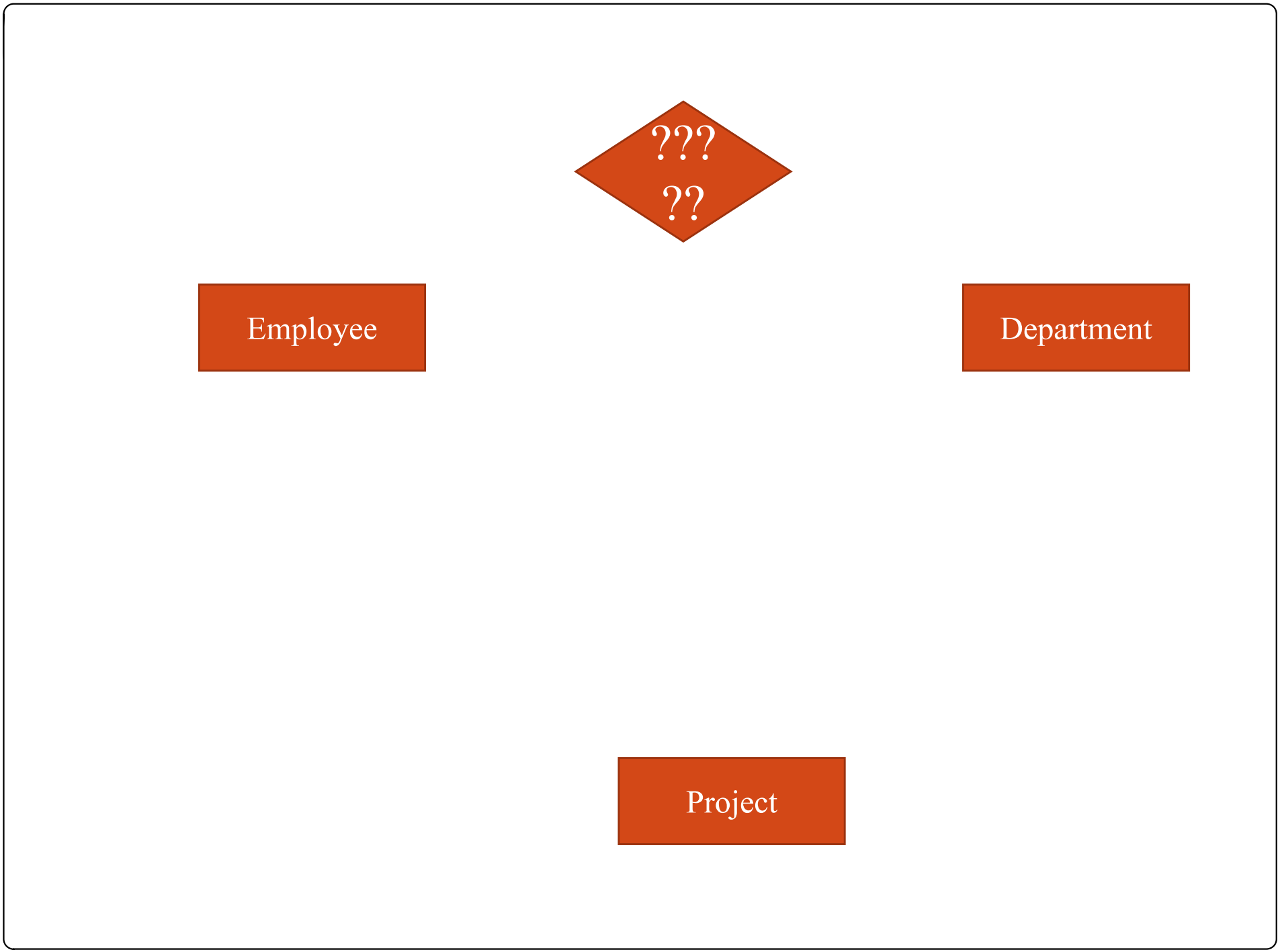
- First task is to identify the entities…
- Example..
  - Company has Employee
  - Company has department
  - Company has projects

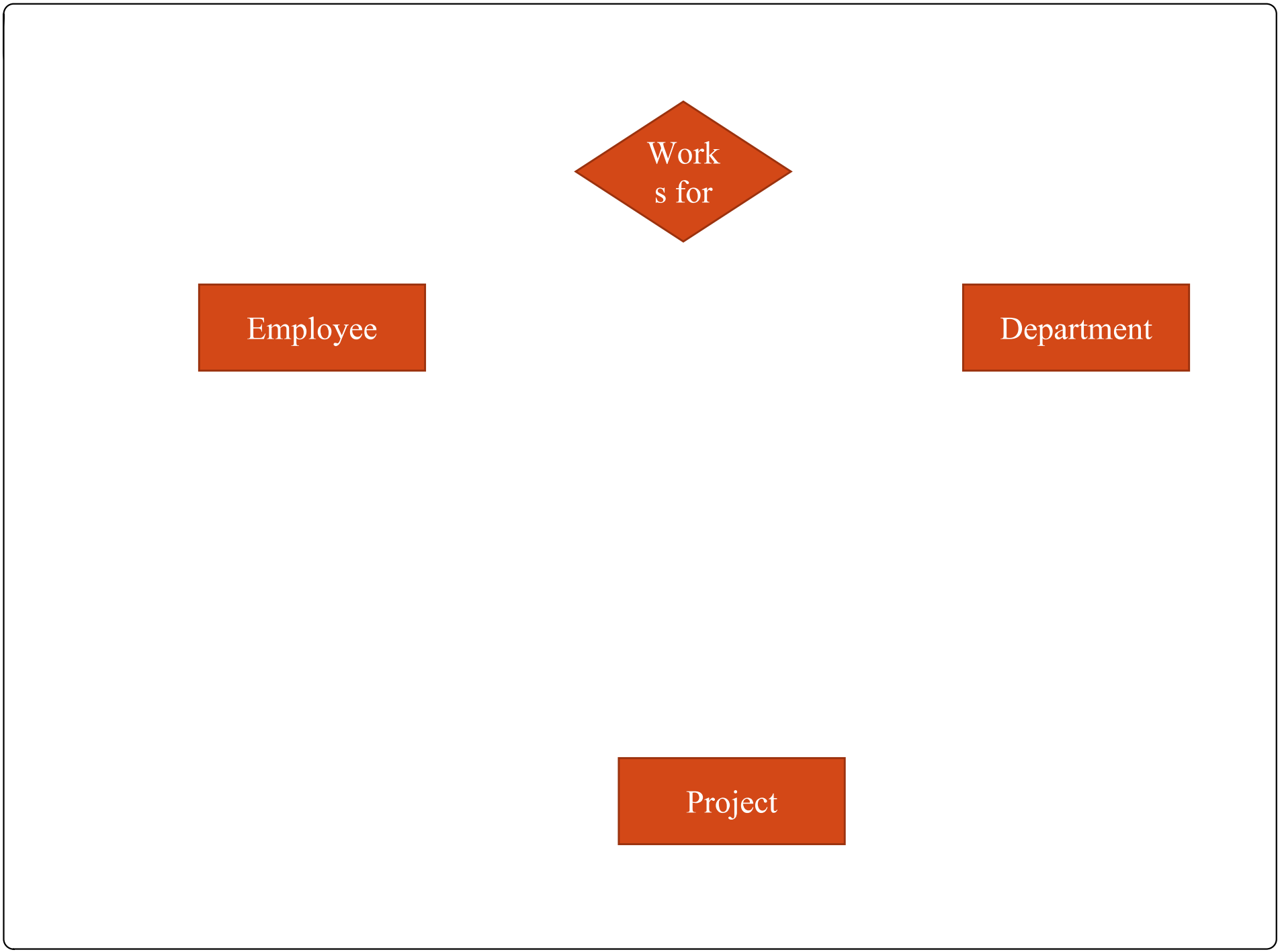
First step….

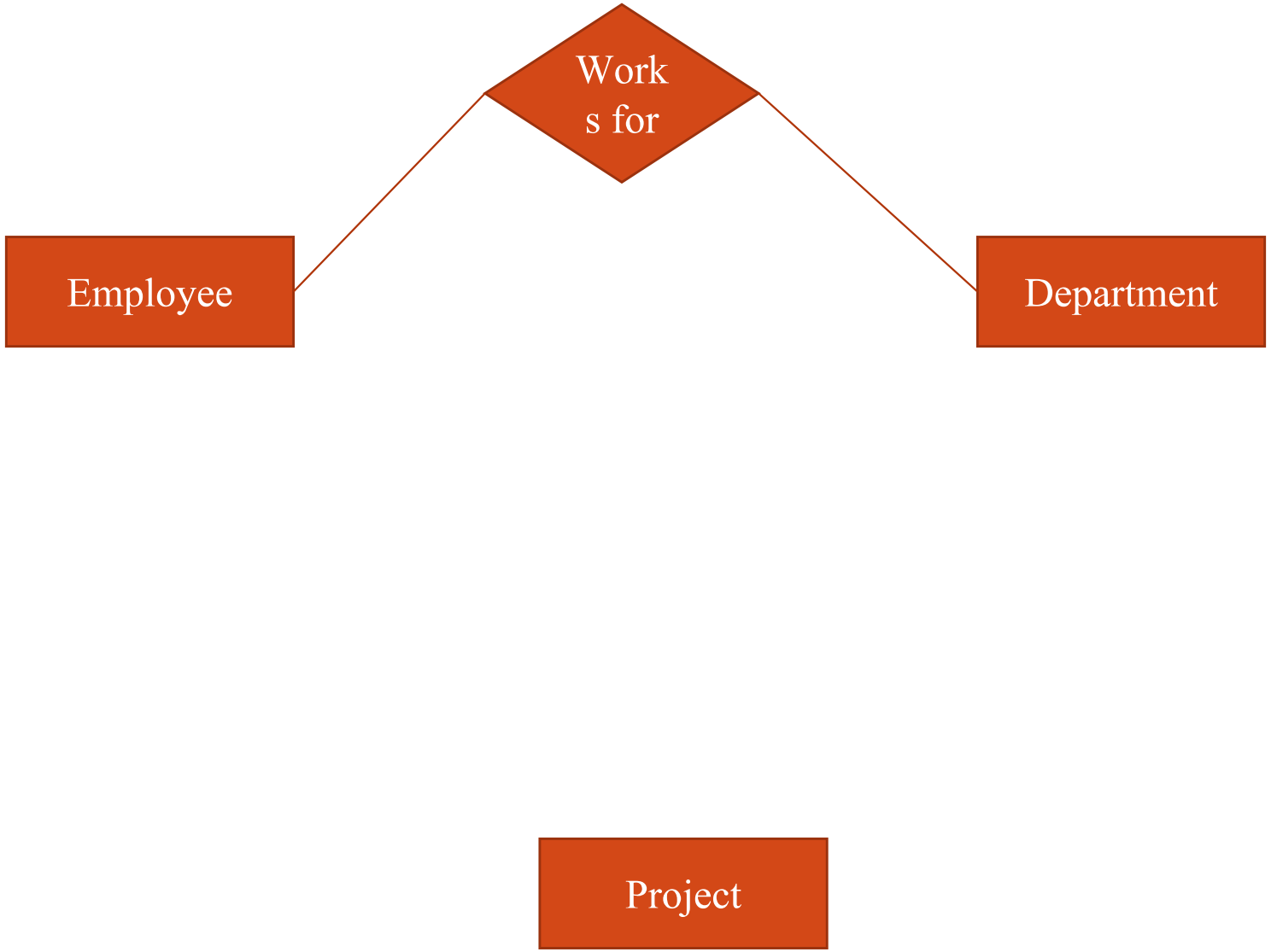
- Place the entities properly

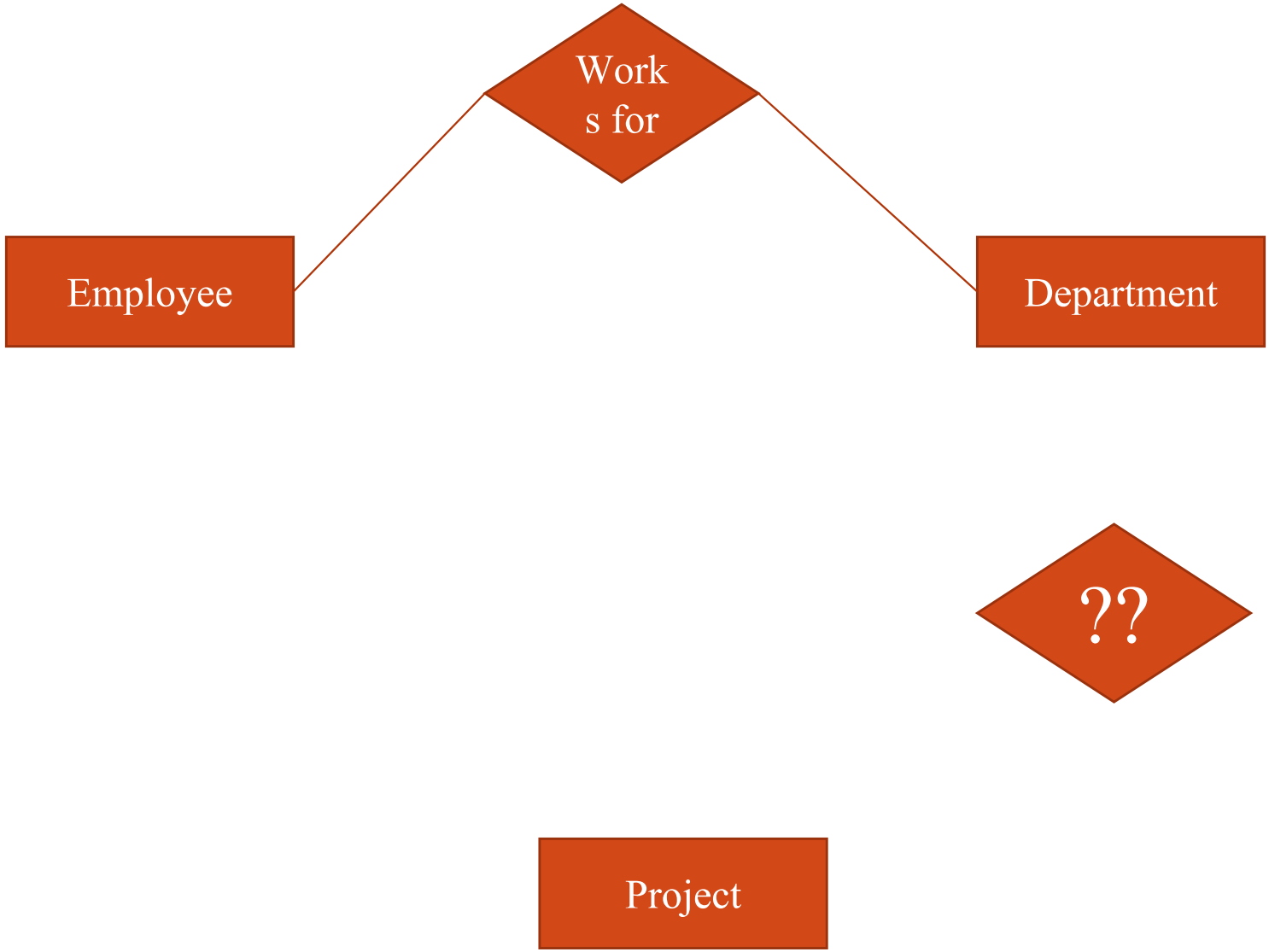

Employee
Department
Project

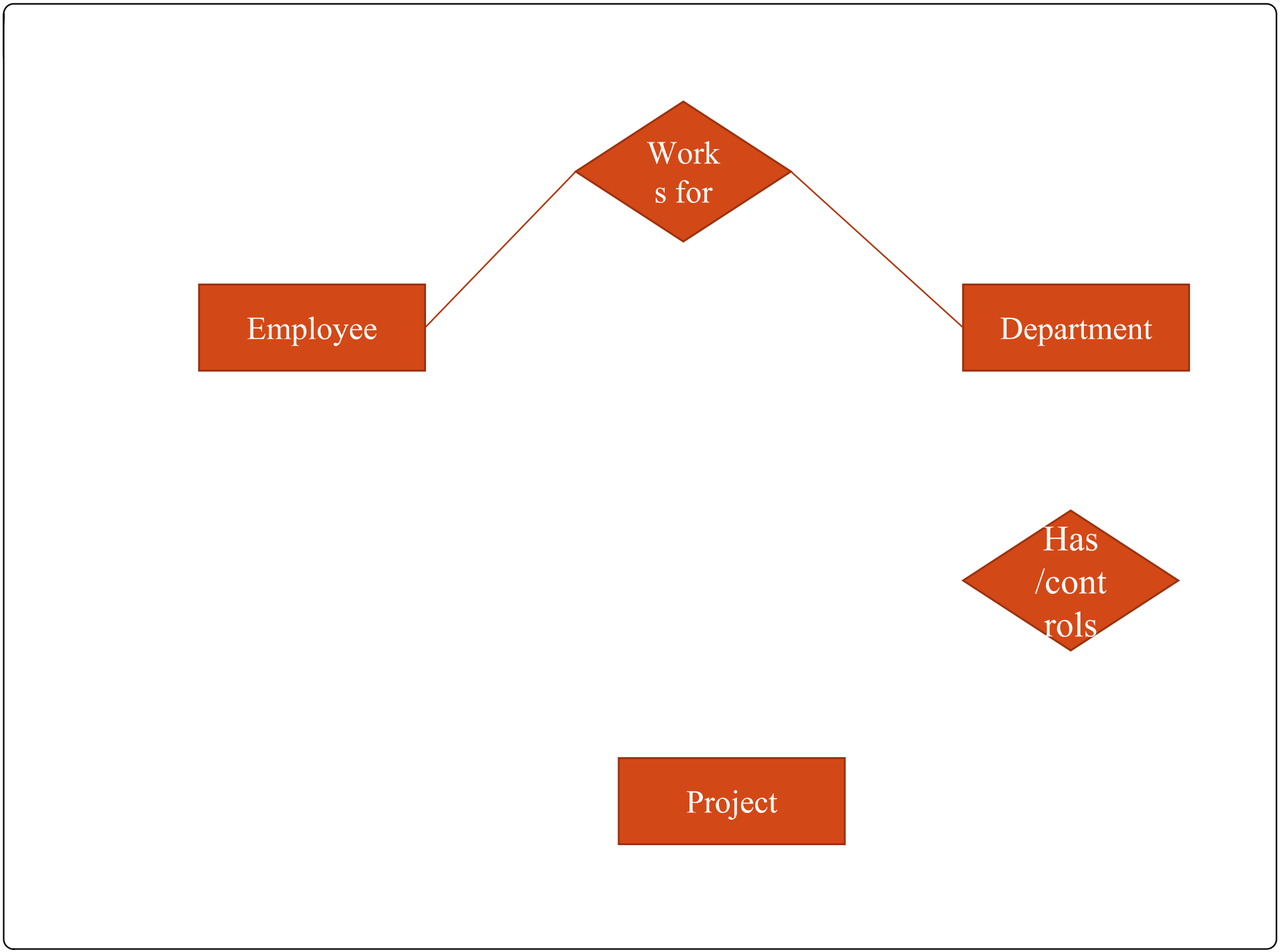
- Try building relationship among entities

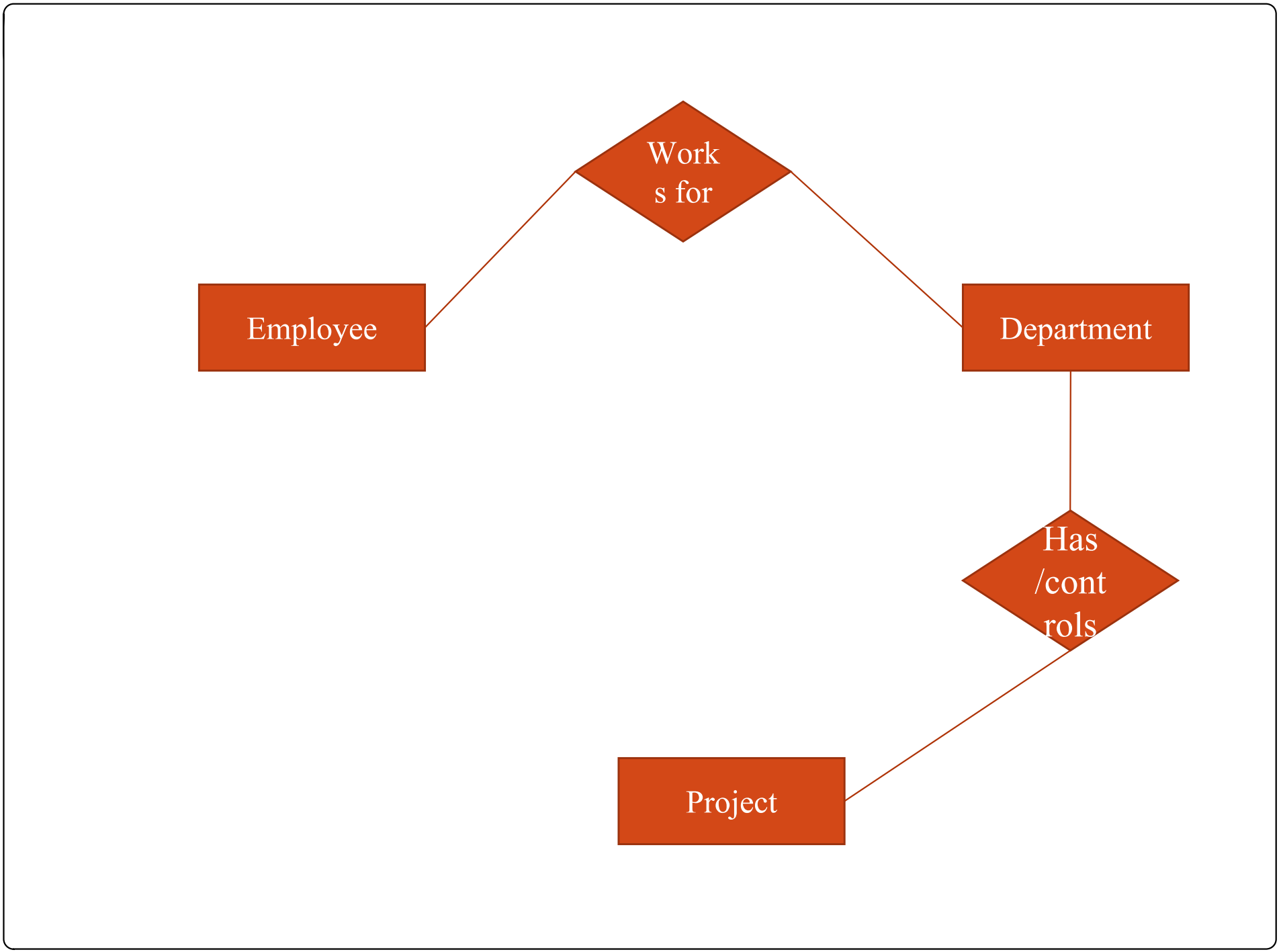

???
??
Employee
Department
Project

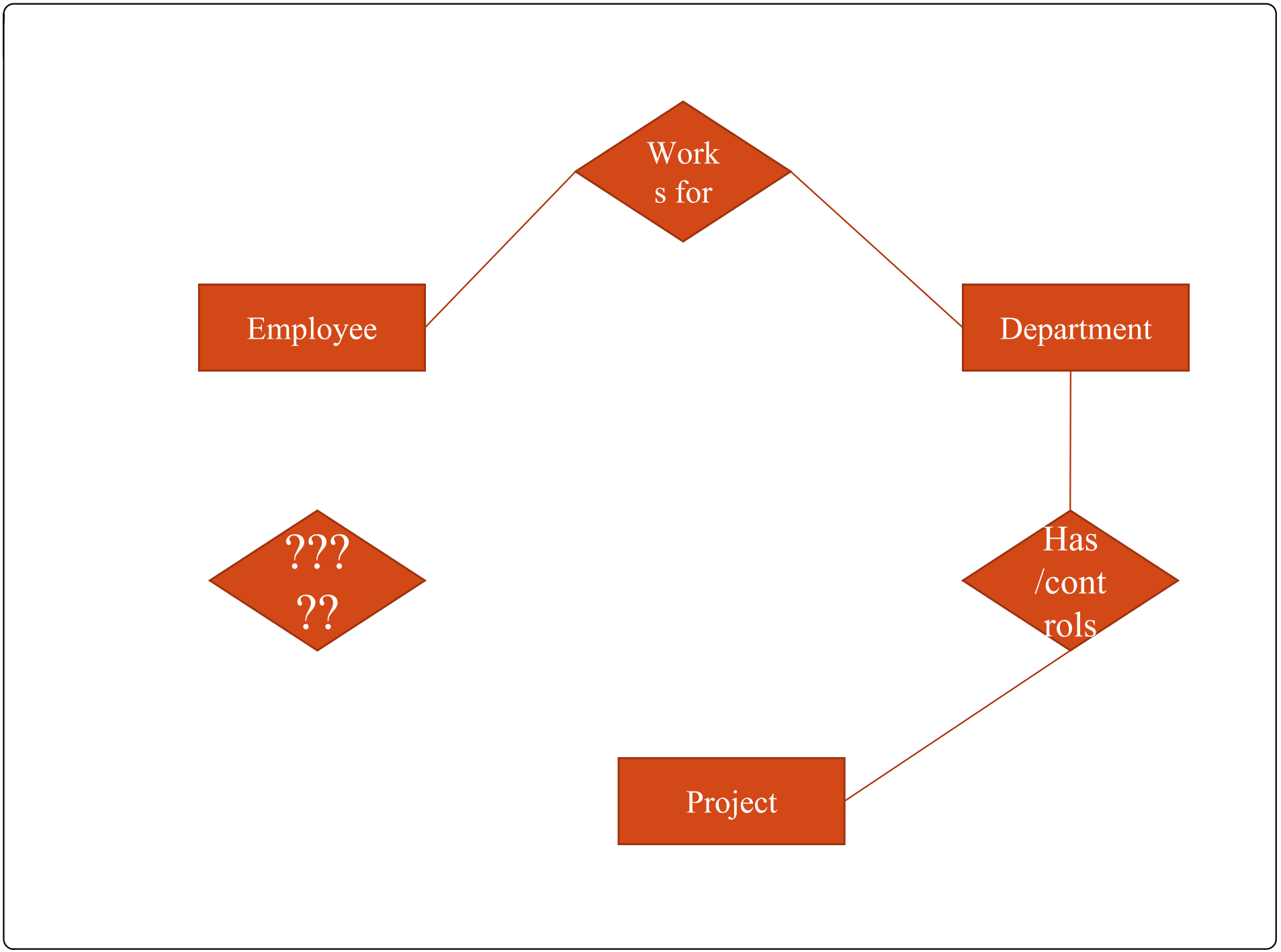

Work
s for
Employee
Department
Project

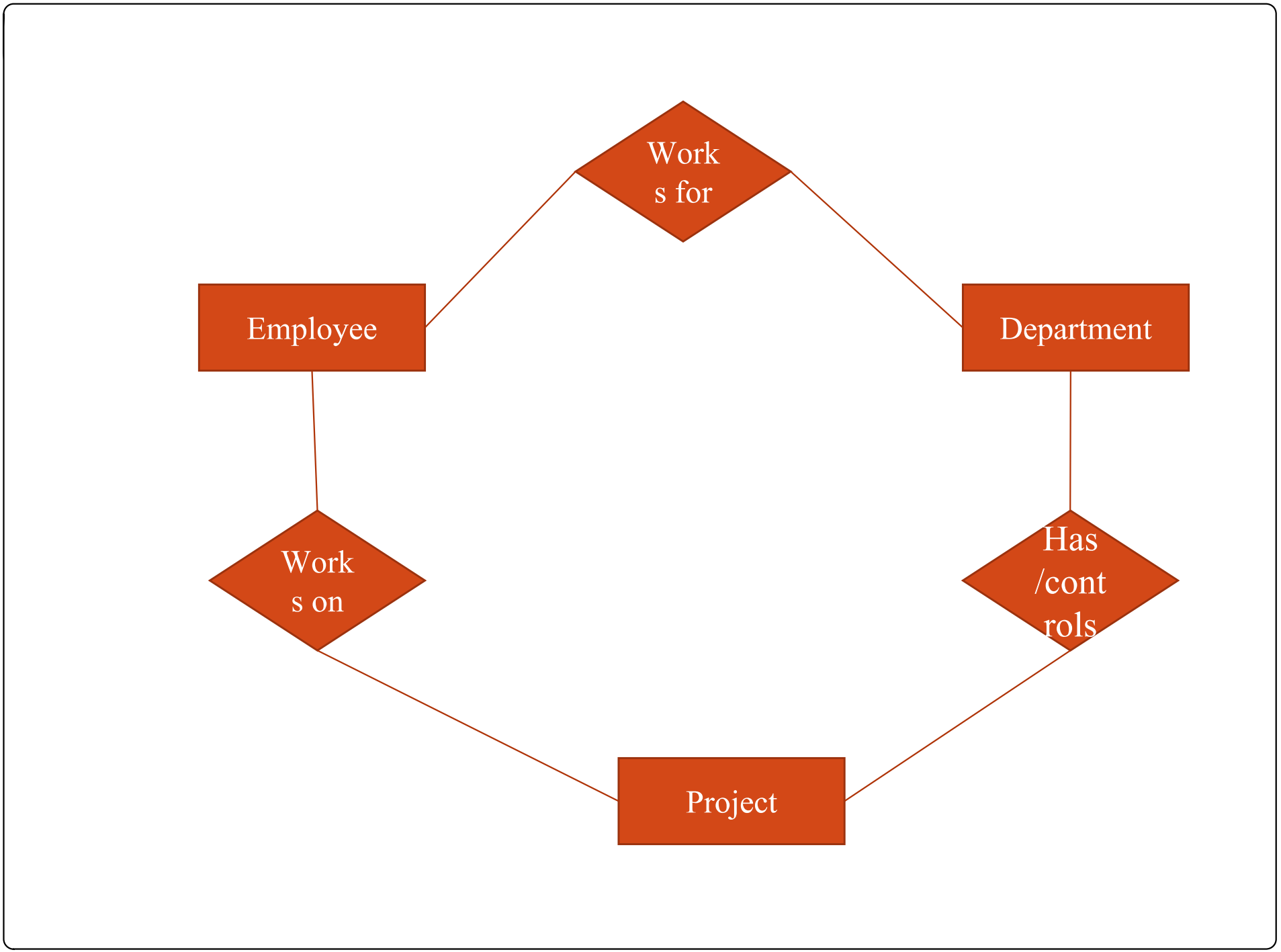

Work
s for
Employee
Department
Project


Work
s for
Employee
Department
??
Project

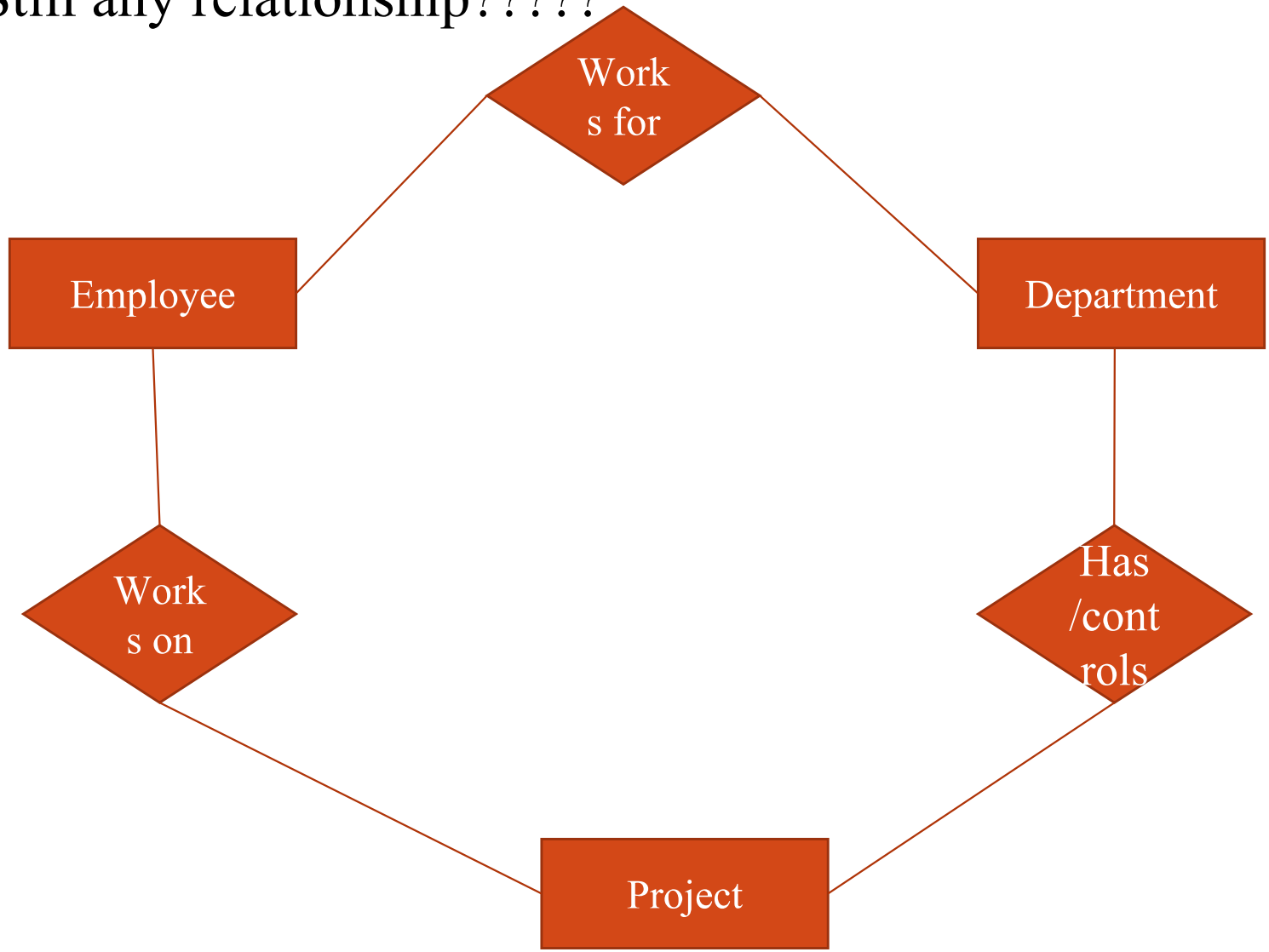

Work
s for
Employee
Department
Has
/cont
rols
Project


Work
s for
Employee
Department
Has
/cont
rols
Project

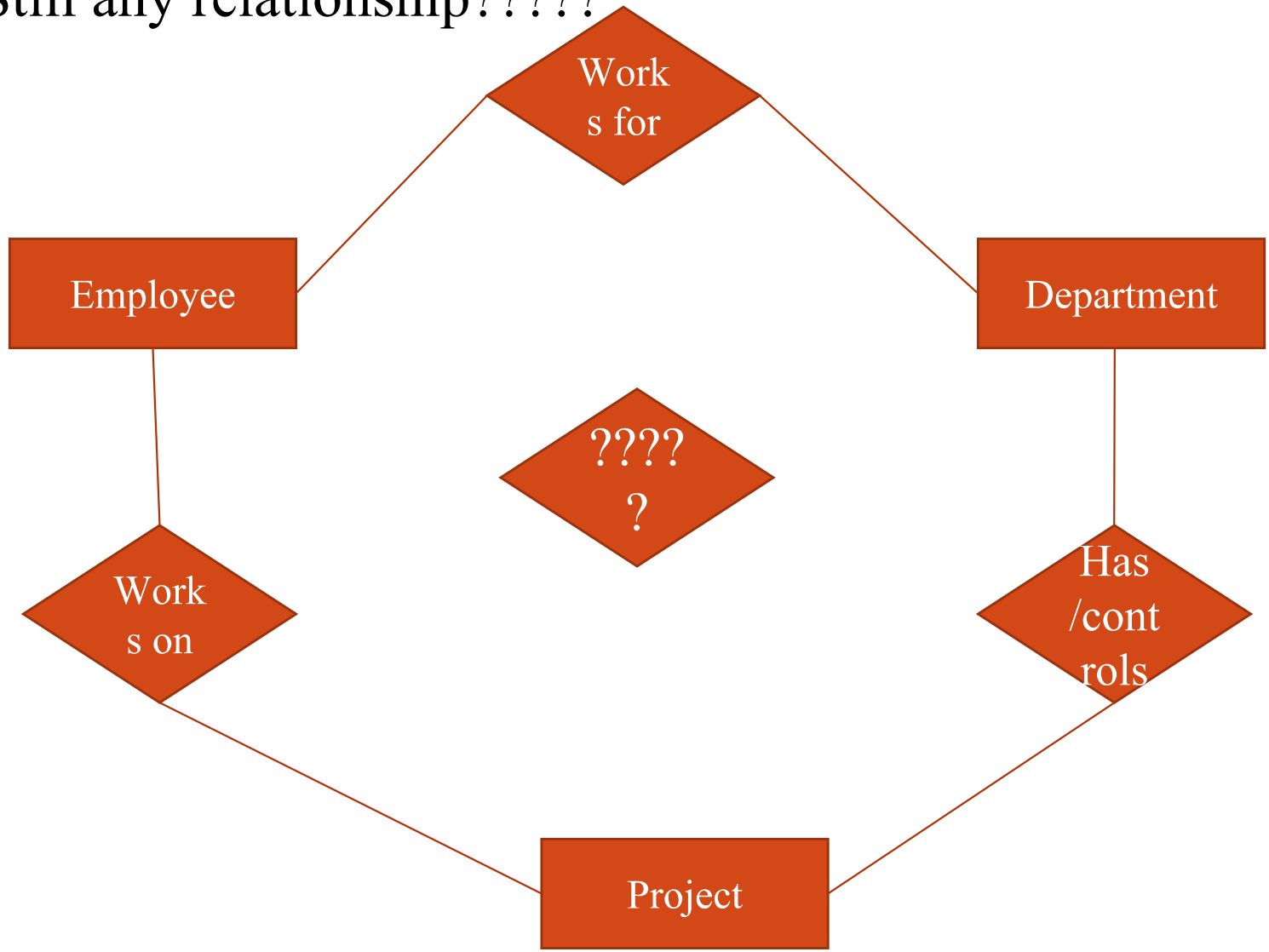

Work
s for
Employee
Department
???
??
Has
/cont
rols
Project

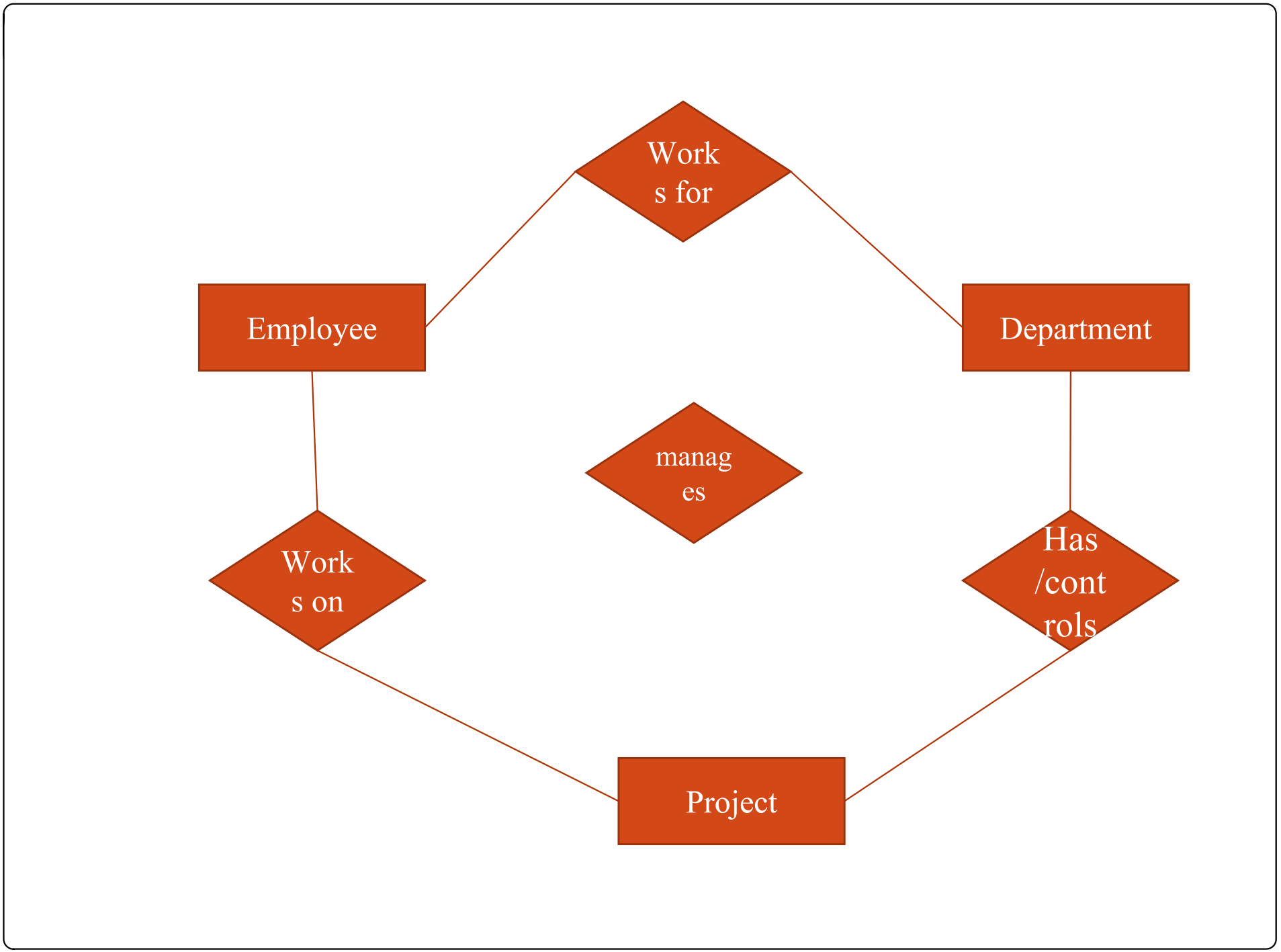

Work
s for
Employee
Department
Work
s on
Has
/cont
rols
Project

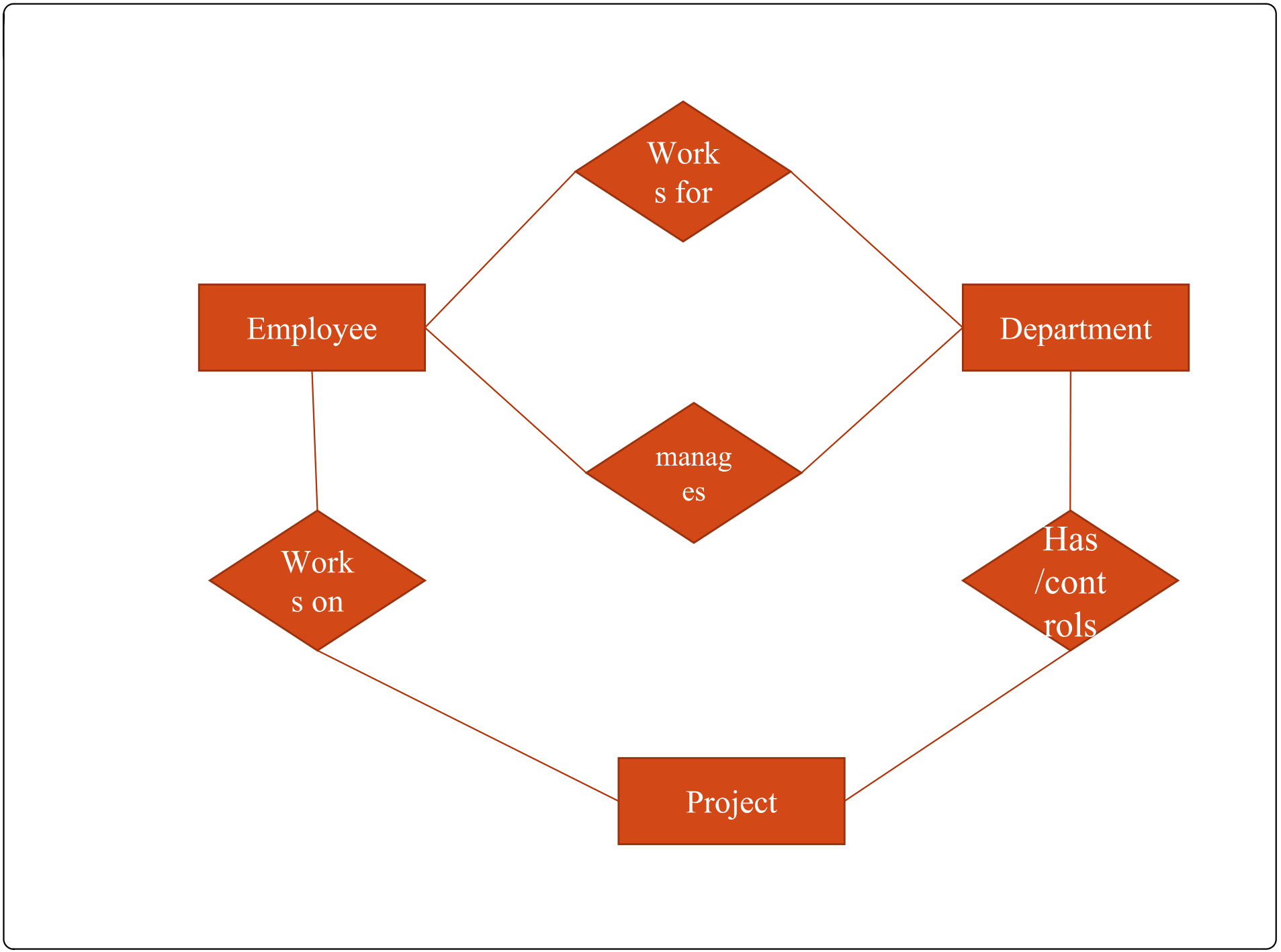
- Still any relationship?????



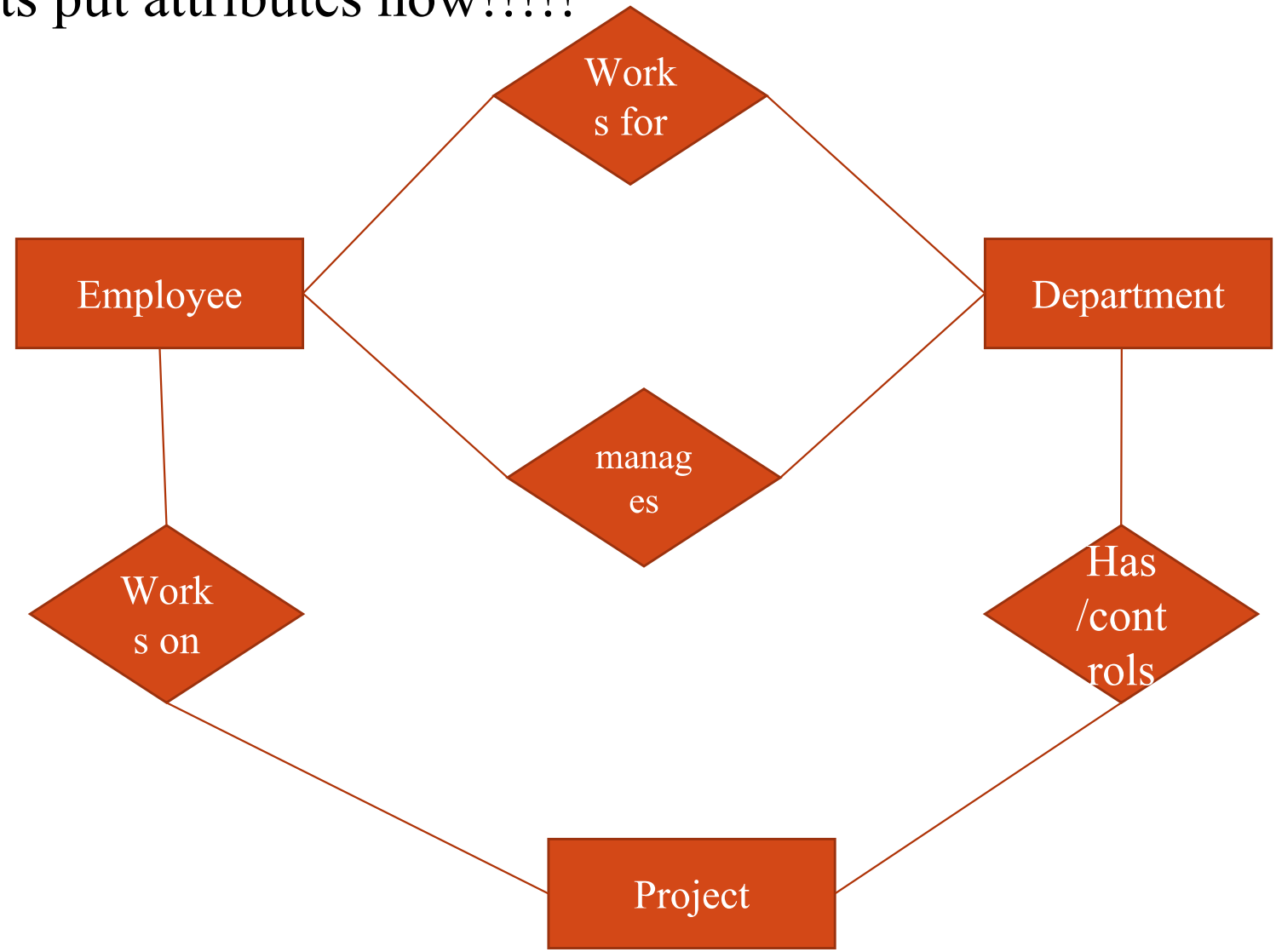
- Still any relationship?????

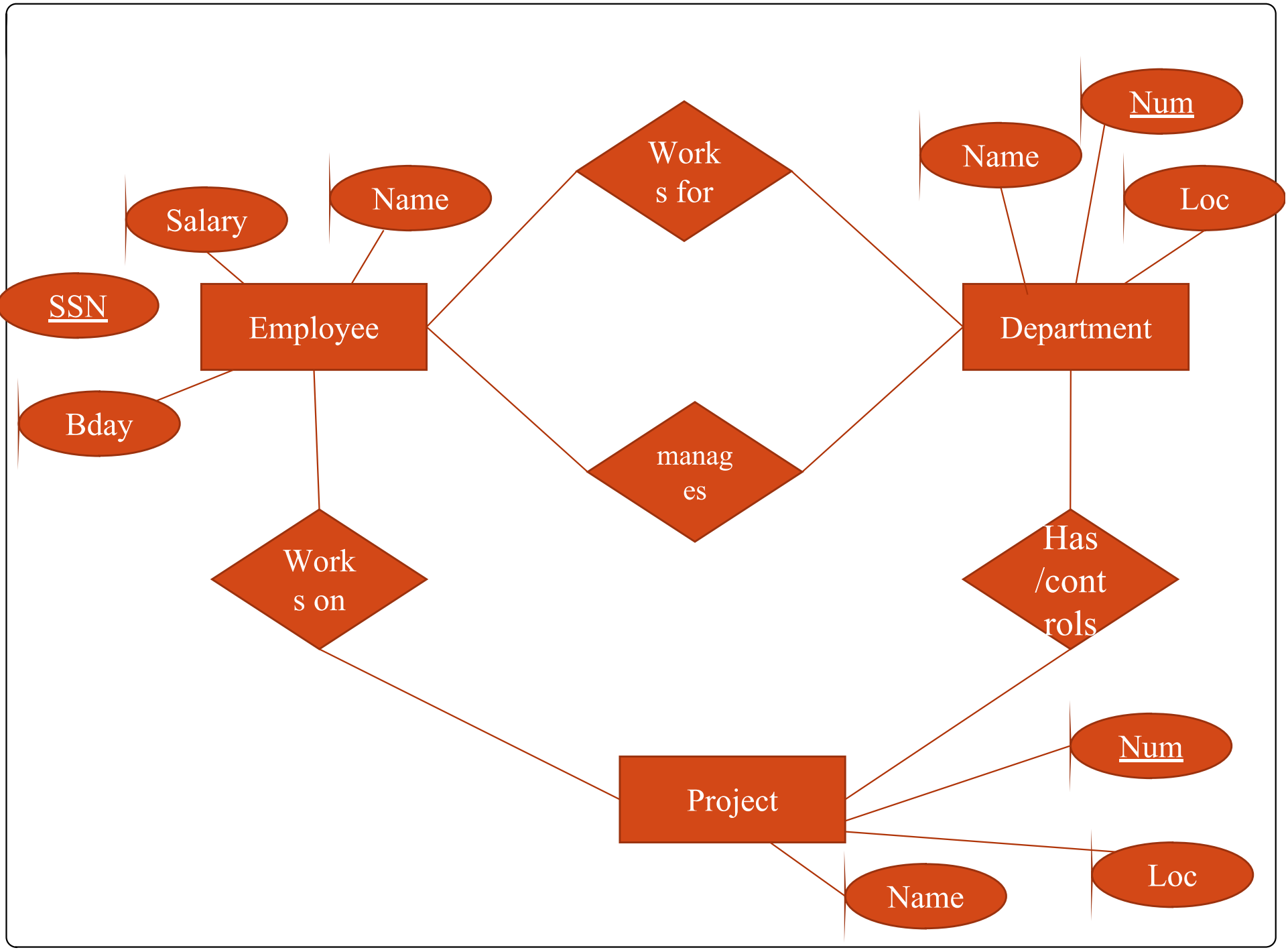


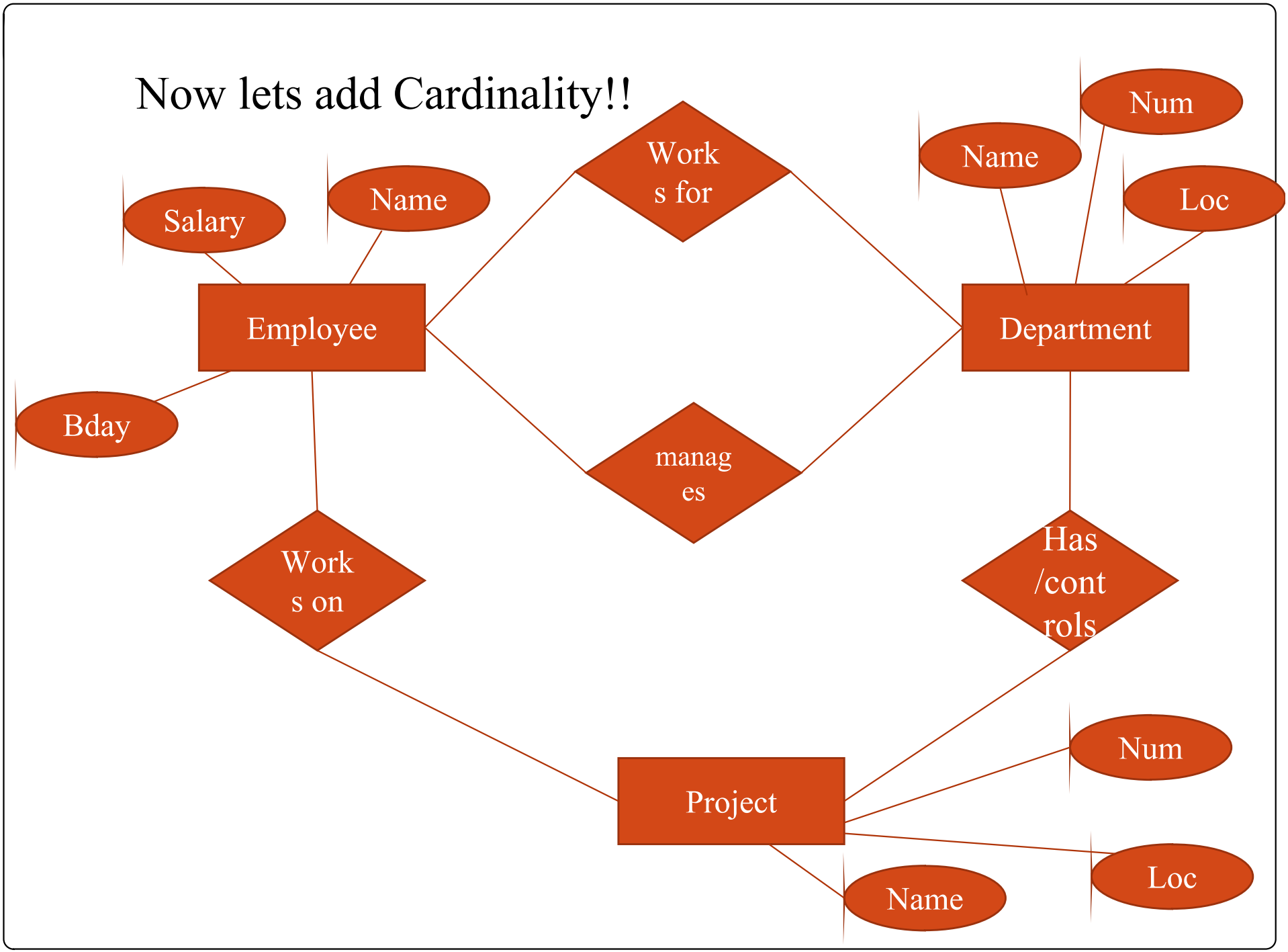

Work
s for
Employee
Department
manag
es
Work
s on
Has
/cont
rols
Project

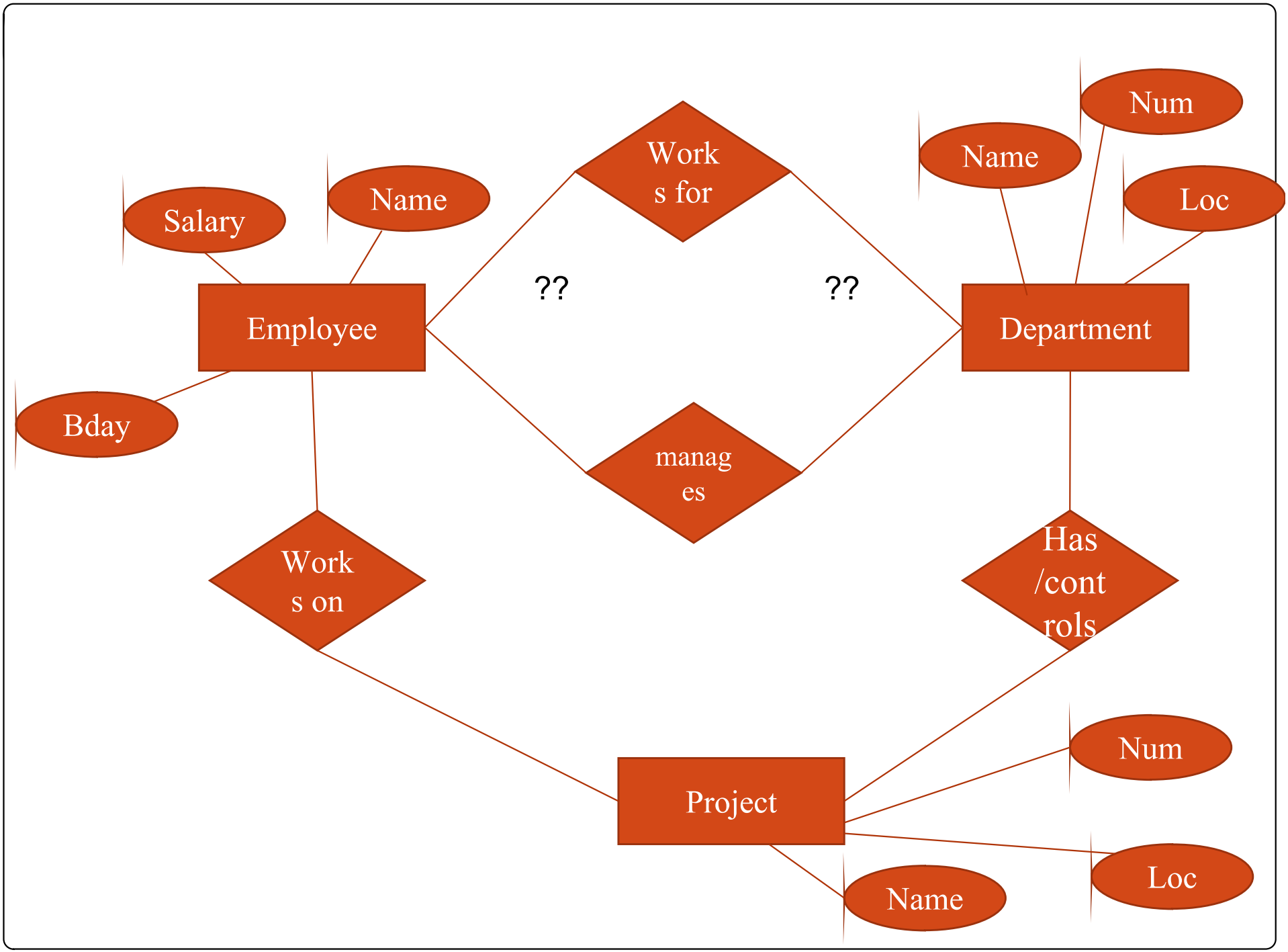

Work
s for
Employee
Department
manag
es
Work
s on
Has
/cont
rols
Project

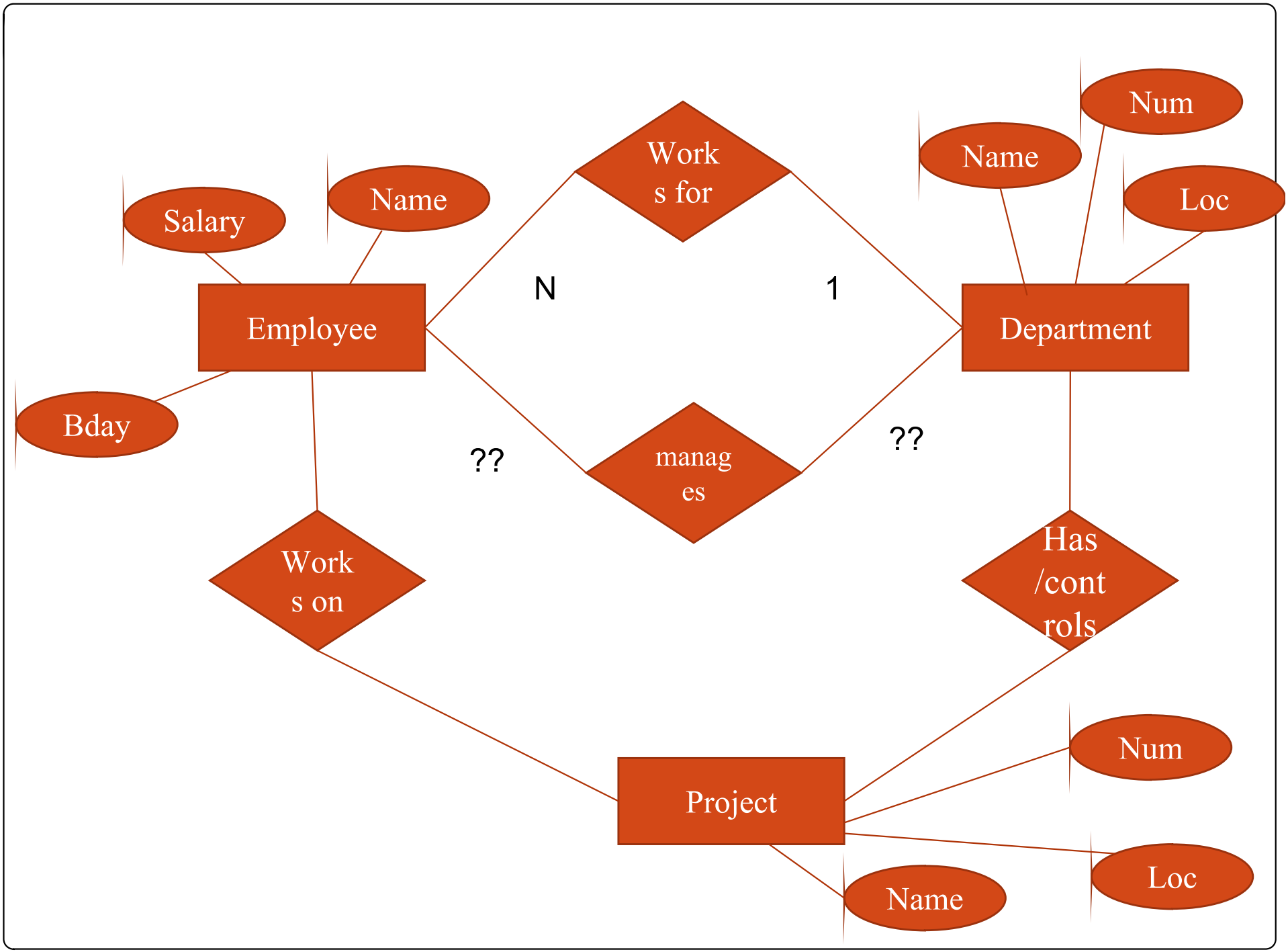
Lets put attributes now!!!!!

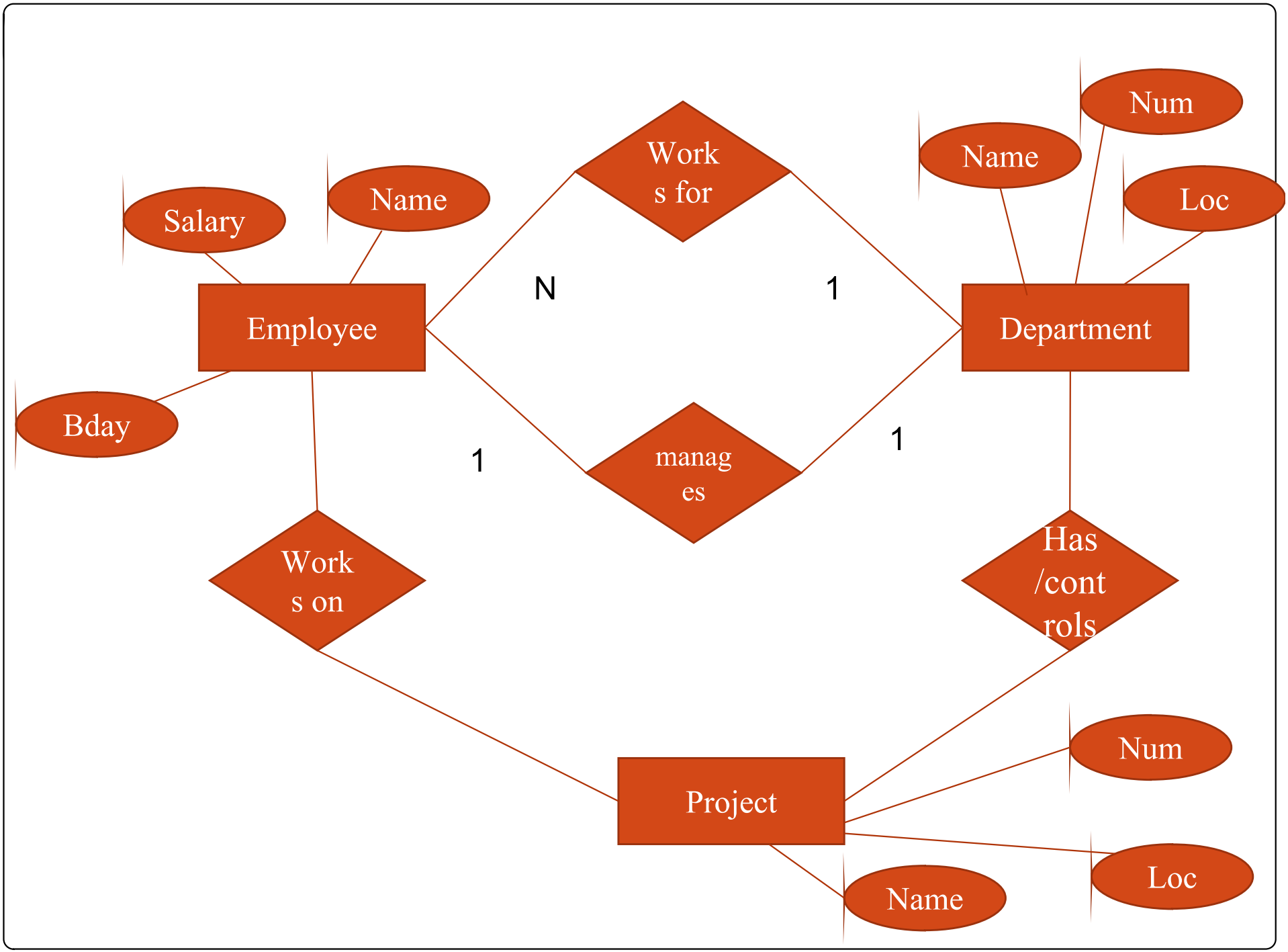


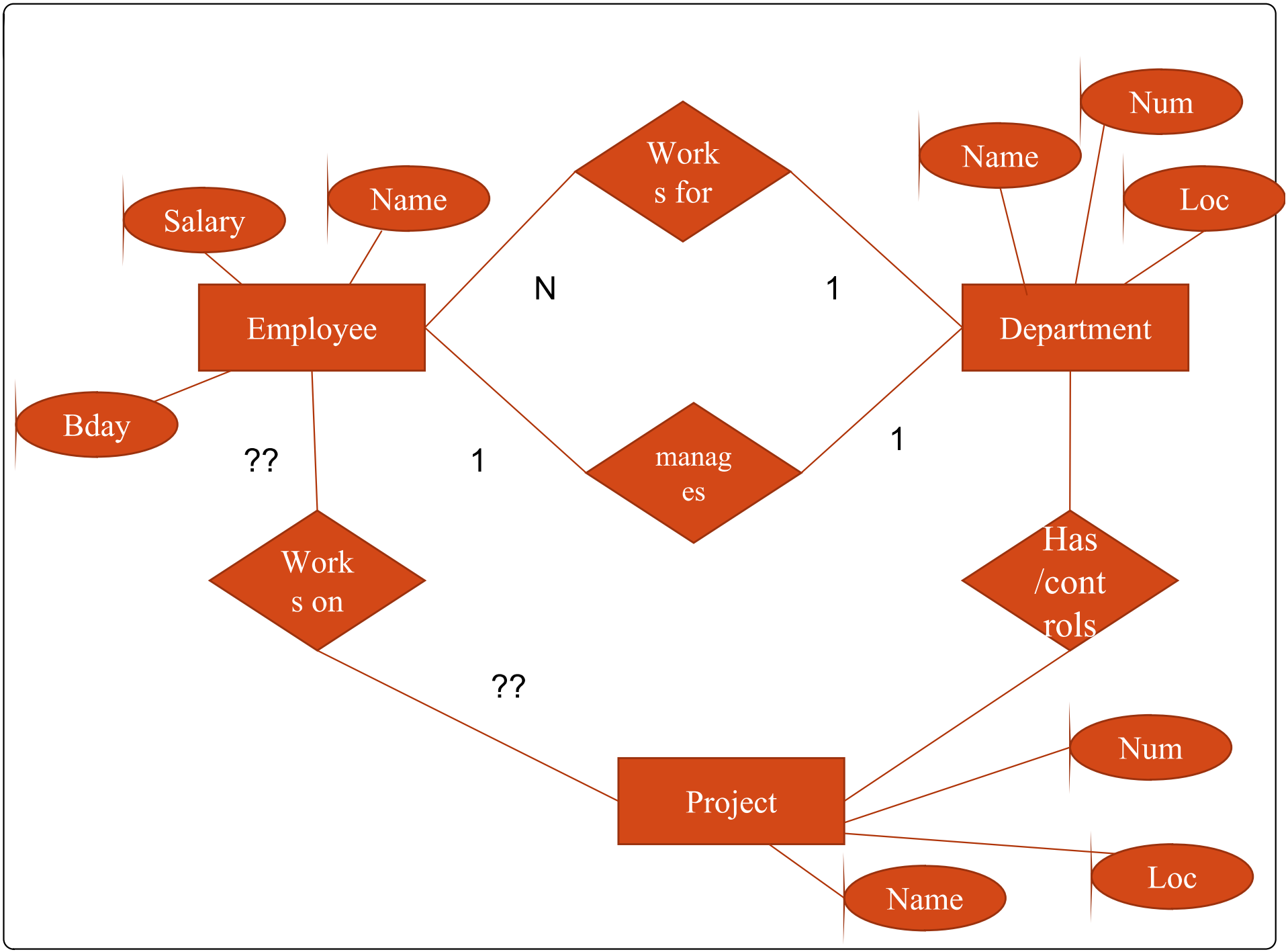

Num
Name
Loc
Work
s for
Name
Salary
SSN
Employee
Department
Bday
manag
es
Work
s on
Has
/cont
rols
Num
Project
Loc
Name

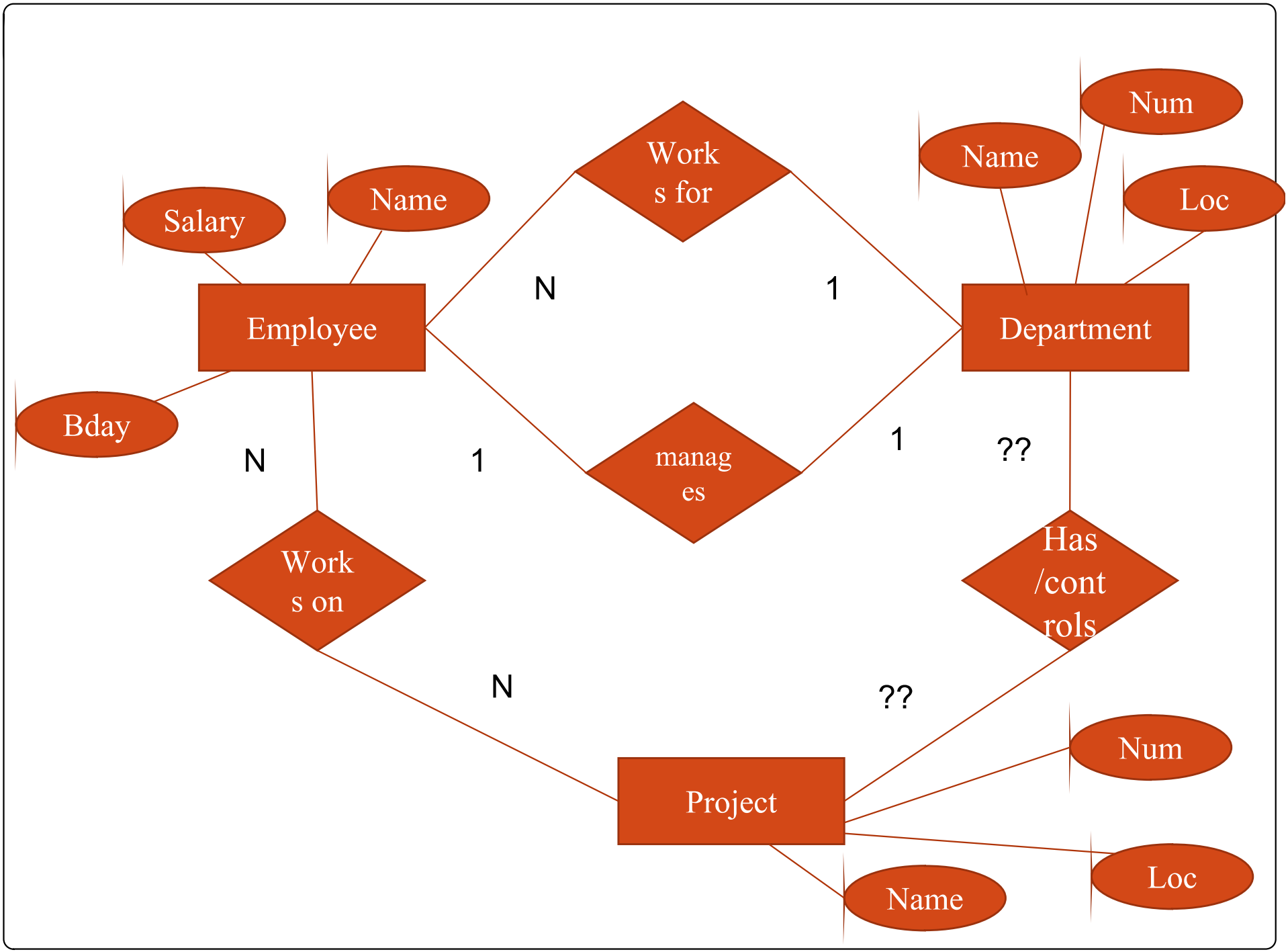

Now lets add Cardinality!!
Work s for
Num
Name
Loc
Salary
Name
Employee
Department
Bday
manag es
Work s on
Has /cont rols
Num
Project
Loc
Name

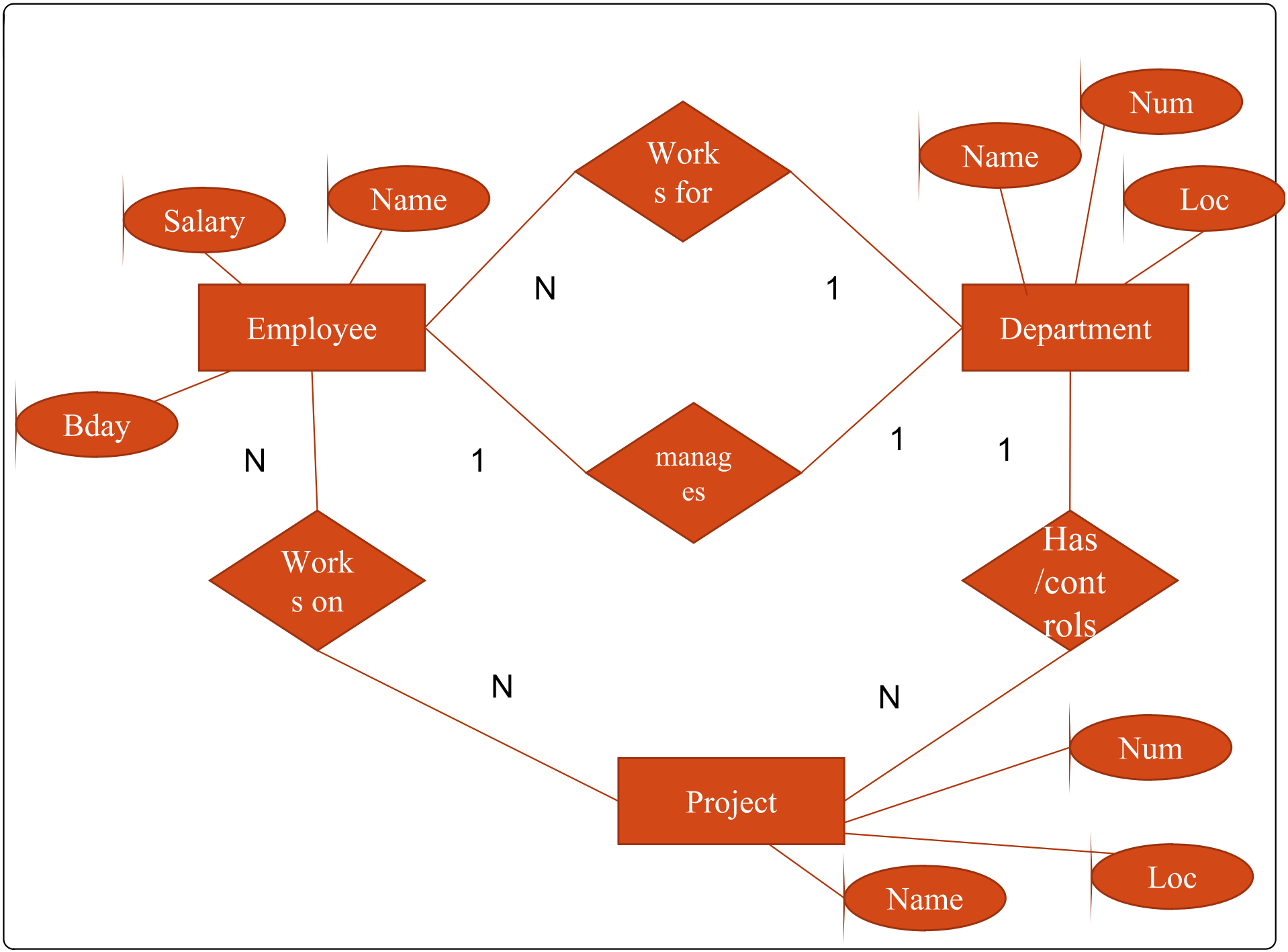

Num
Name
Work
s for
Loc
Salary
Name
??
??
Employee
Department
Bday
manag
es
Work
s on
Has
/cont
rols
Num
Project
Loc
Name


Num
Name
Work
s for
Loc
Salary
Name
N
1
Employee
Department
Bday
??
??
manag
es
Work
s on
Has
/cont
rols
Num
Project
Loc
Name


Num
Name
Work
s for
Loc
Salary
Name
N
1
Employee
Department
Bday
1
1
manag
es
Work
s on
Has
/cont
rols
Num
Project
Loc
Name


Work
s for
Num
Name
Loc
Salary
Name
N
1
Employee
Department
Bday
??
1
manag
es
1
Work
s on
Has
/cont
rols
??
Num
Project
Loc
Name


Num
Name
Work
s for
Loc
Salary
Name
N
1
Employee
Department
Bday
N
1
manag
es
1
??
Has
/cont
rols
Work
s on
N
??
Num
Project
Loc
Name


Num
Name
Work
s for
Loc
Salary
Name
N
1
Employee
Department
Bday
N
1
manag
es
1
1
Work
s on
Has
/cont
rols
N
N
Num
Project
Loc
Name

Understood just the basics at least ???????

Lets try one more now!!!

Lets try one more now!!!

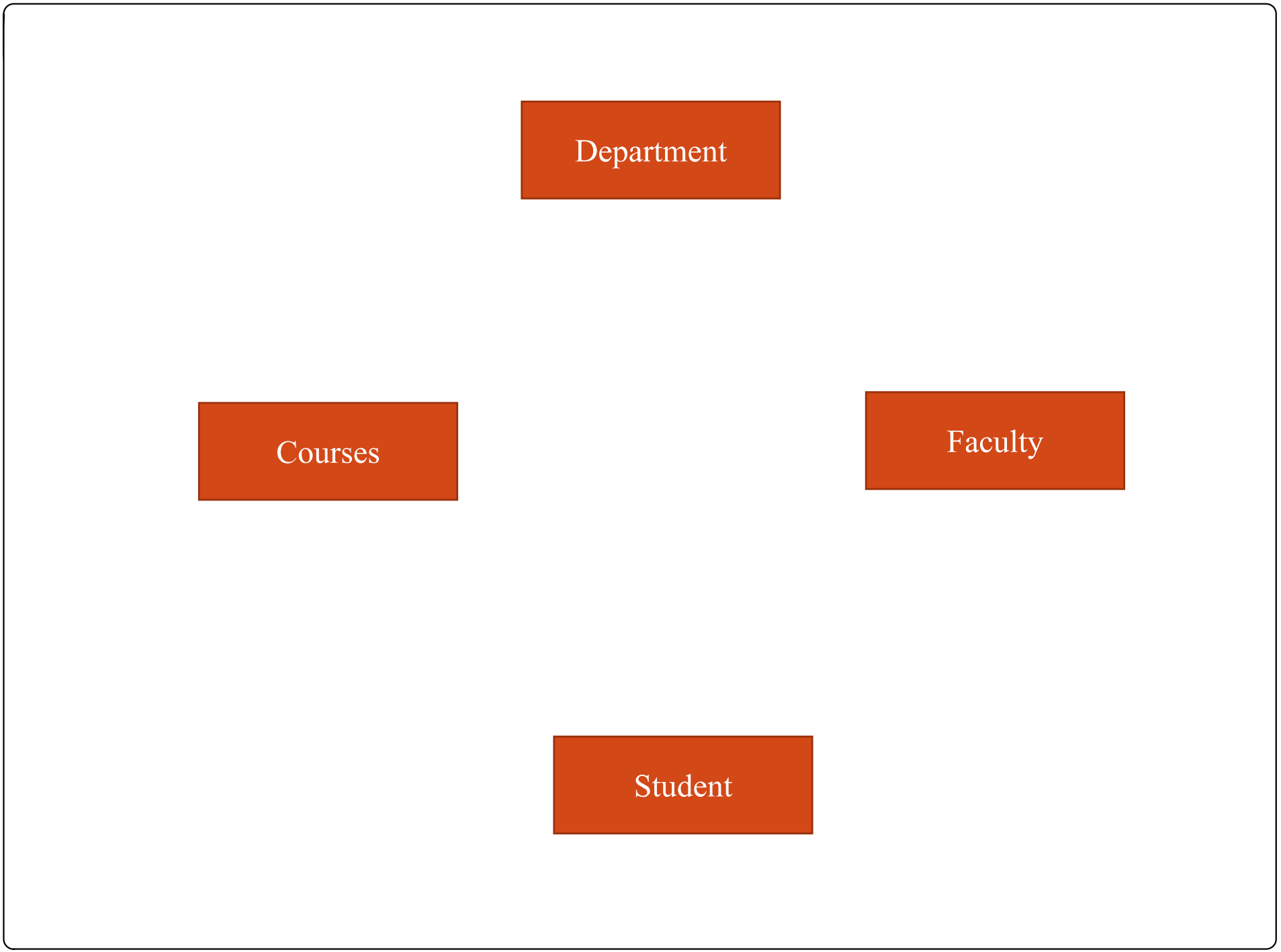
- Draw an ER Diagram for College Database!!!!!

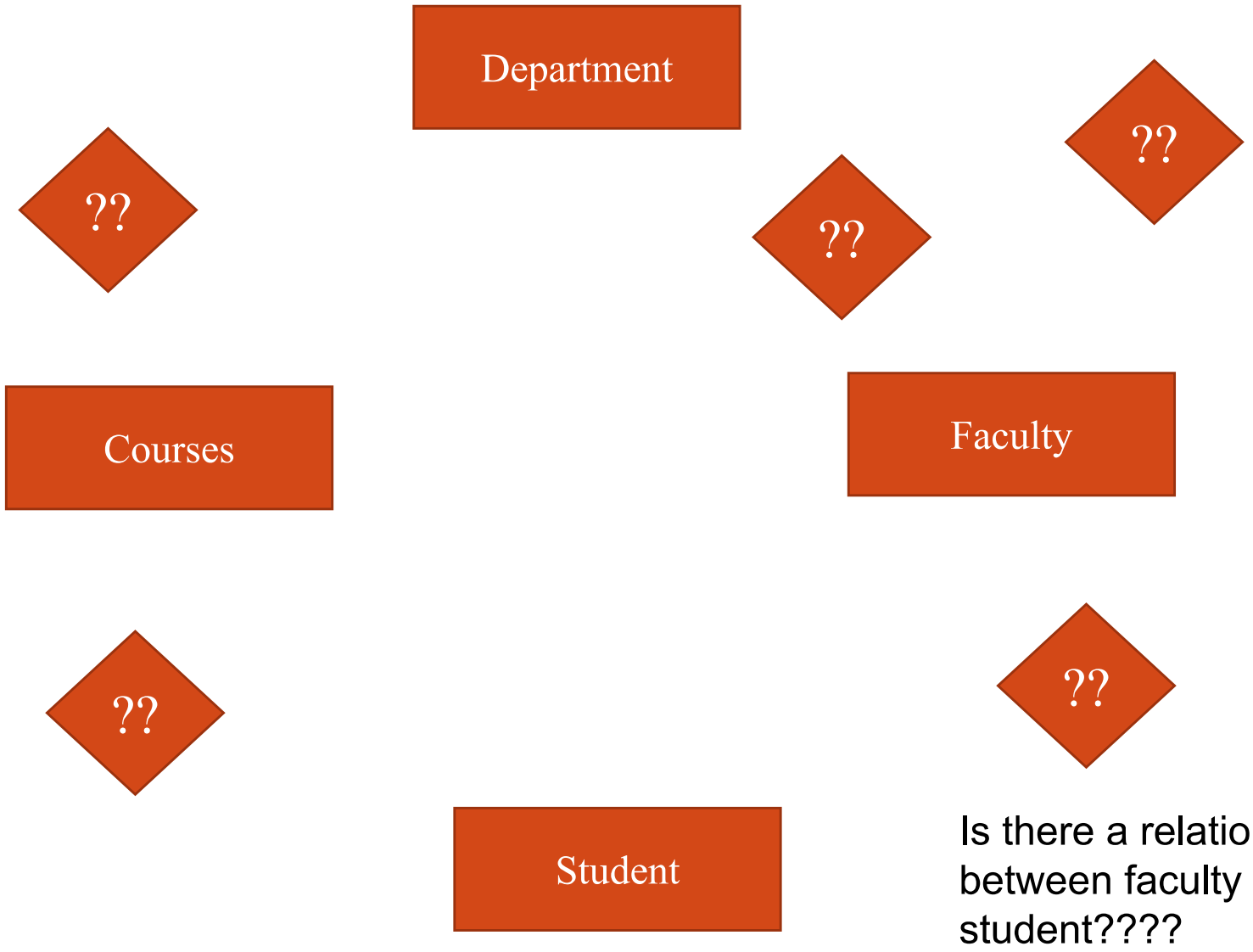
Lets try one more now!!!

- Draw an ER Diagram for College Database!!!!!
- Identify the entities!!!!!!!

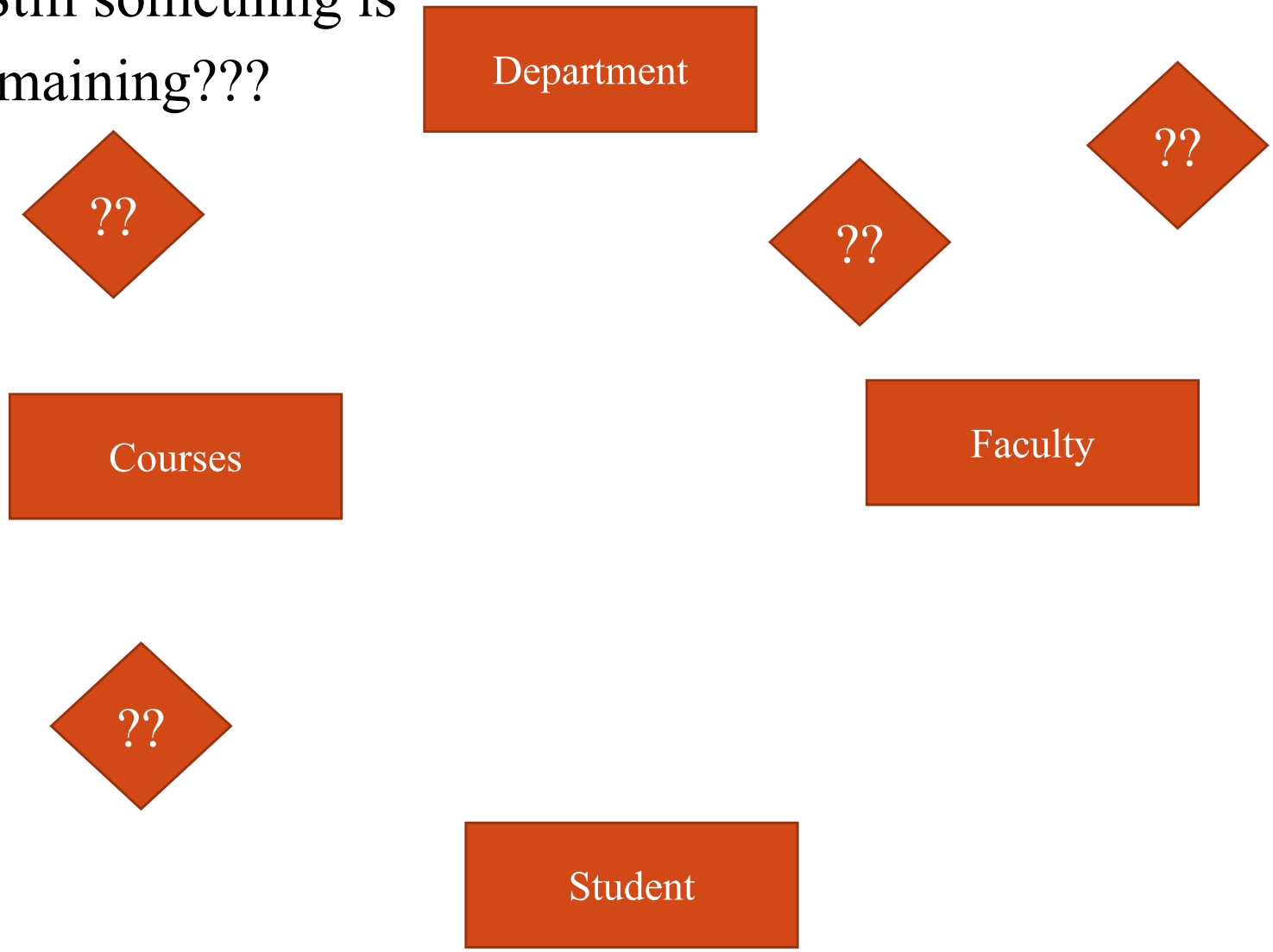
Lets try one more now!!!

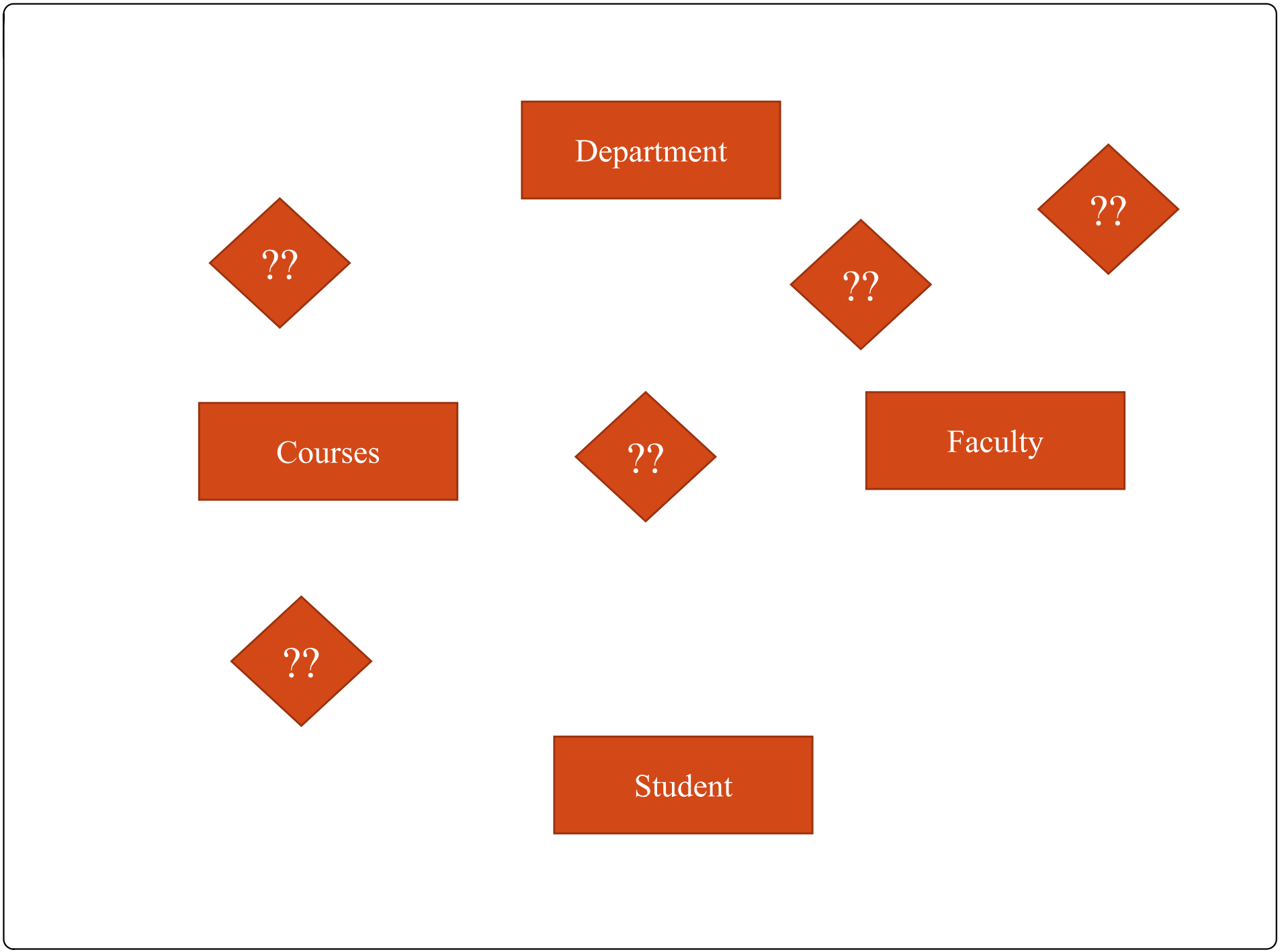
- Draw an ER Diagram for College Database!!!!!
- Identify the entities!!!!!!!
  - Department
  - Courses
  - Faculty
  - Students

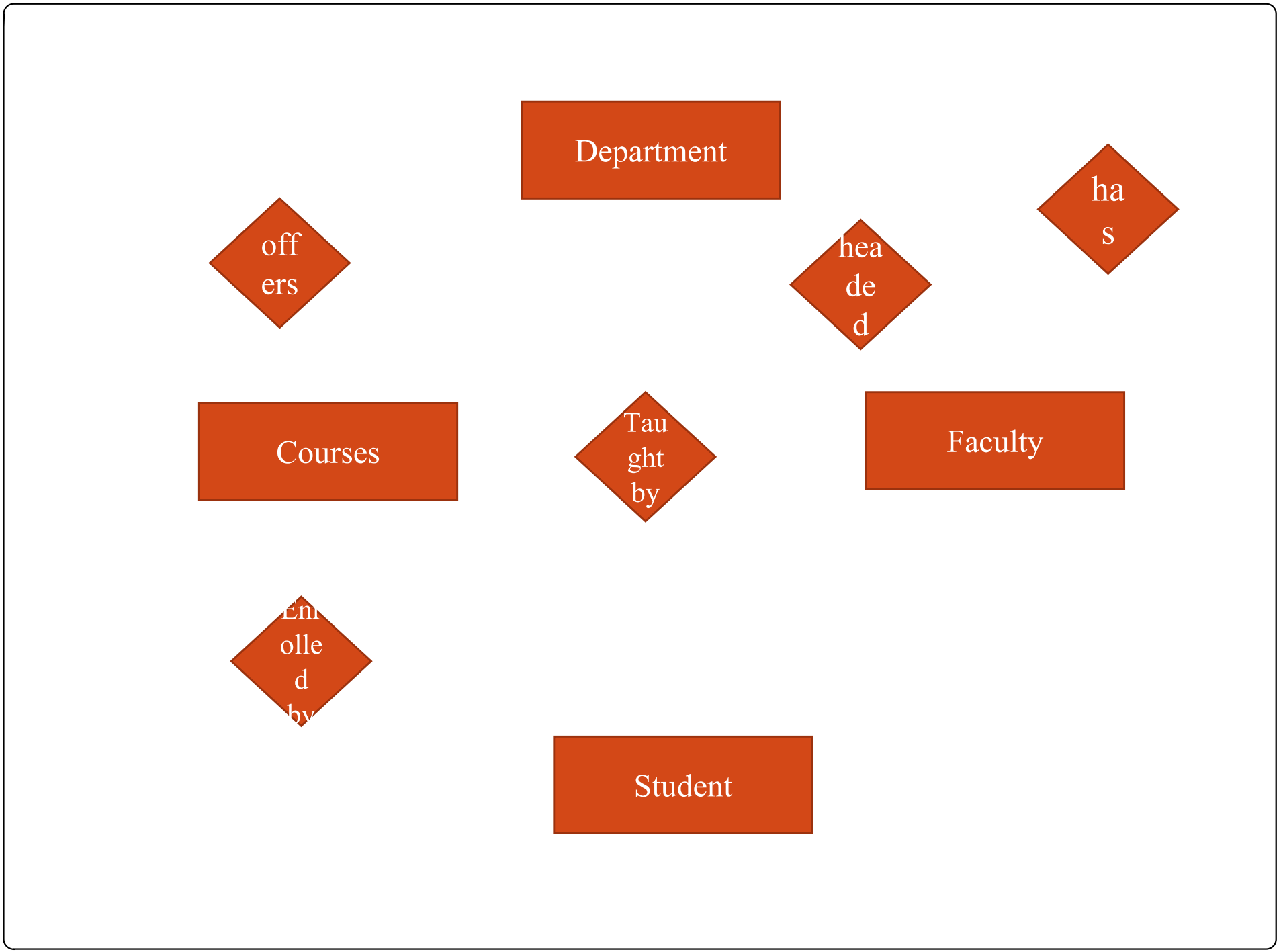

Department
Courses
Faculty
Student


Department
??
??
??
Courses
Faculty
??
Student
Is there a relation between faculty & student????

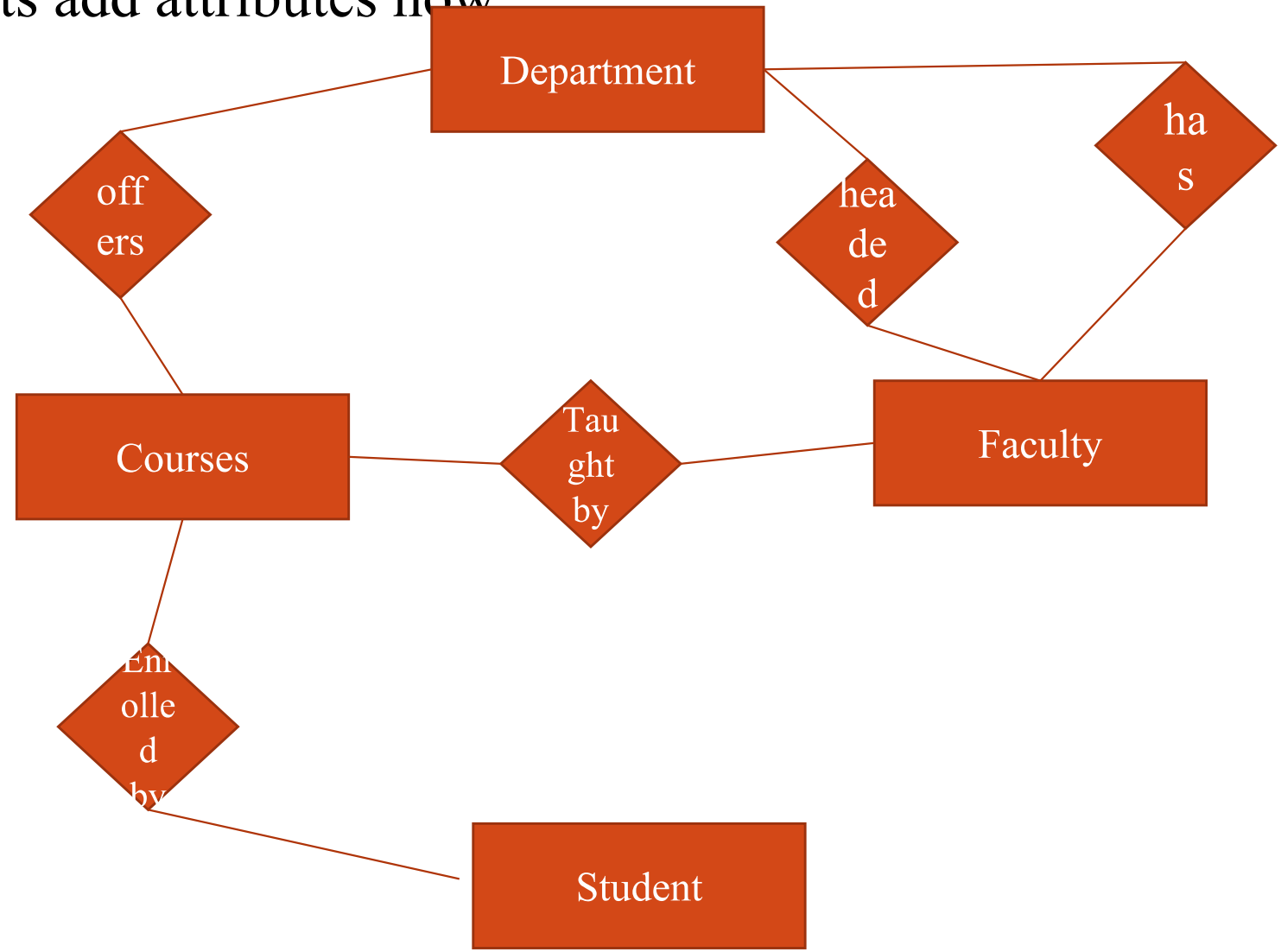
- Still something is Remaining???

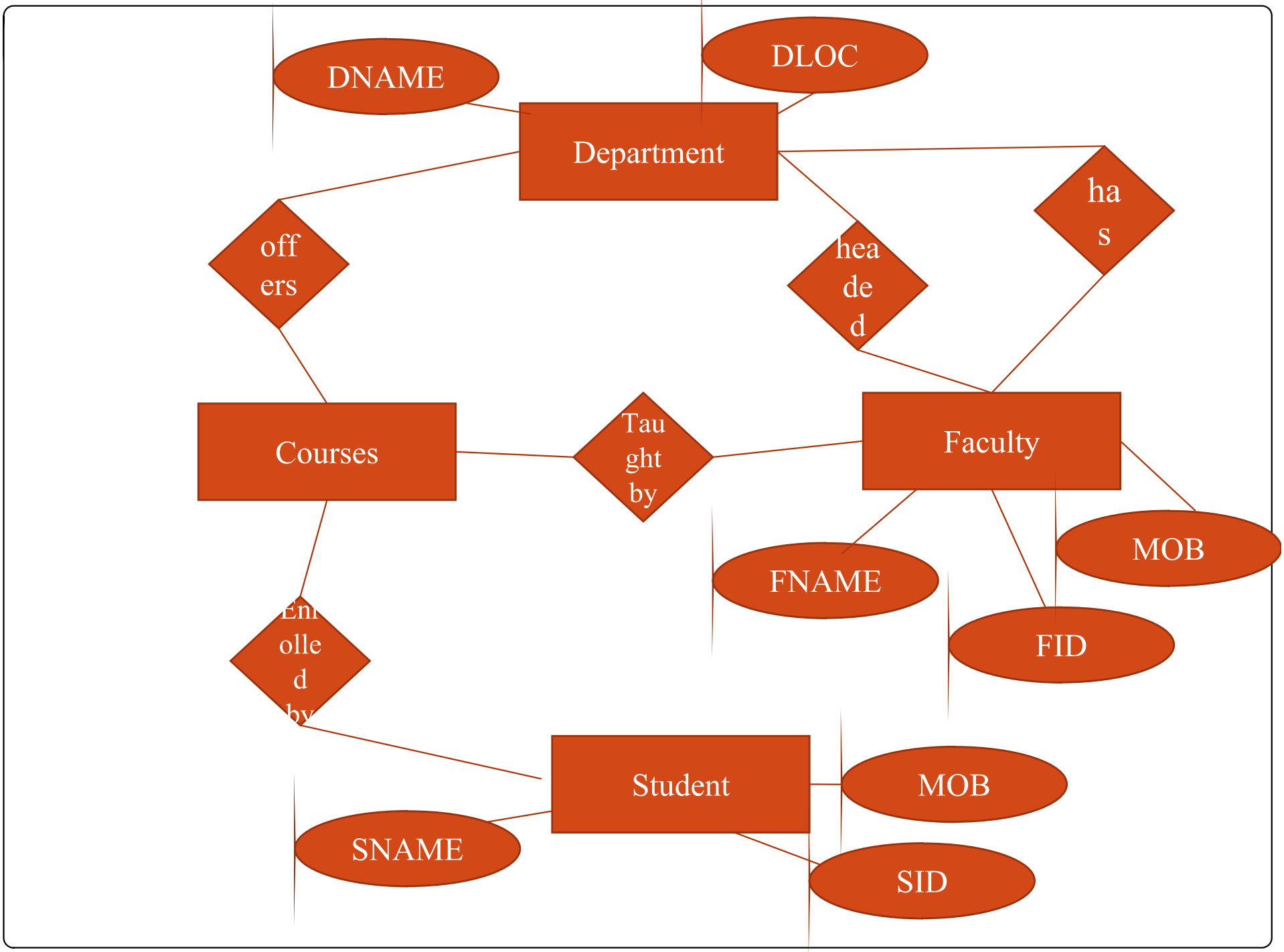



Department
??
??
??
Courses
??
Faculty
??
Student

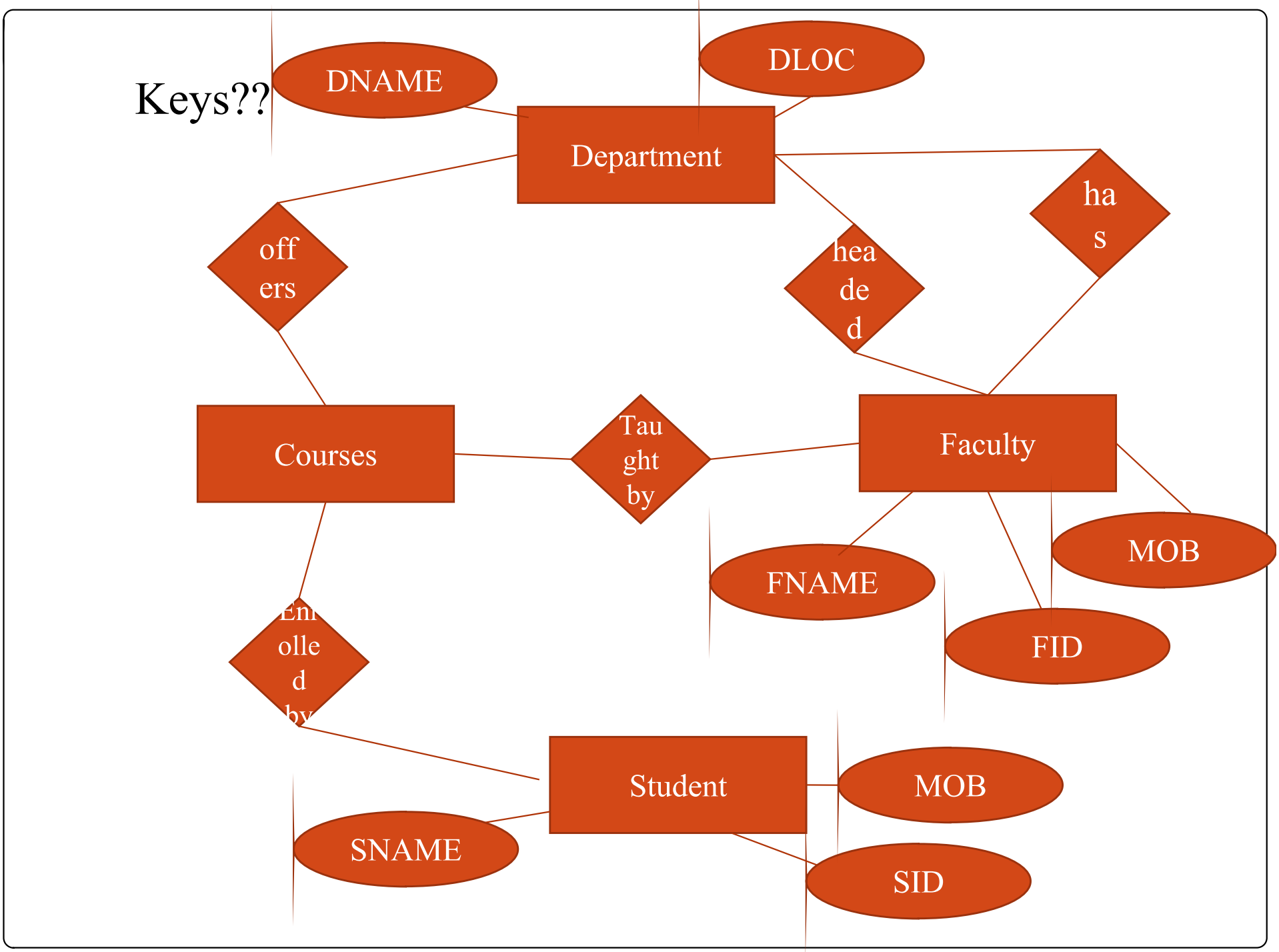

Department
offers
headed
has
Courses
Taught by
Faculty
Enrolled by
Student

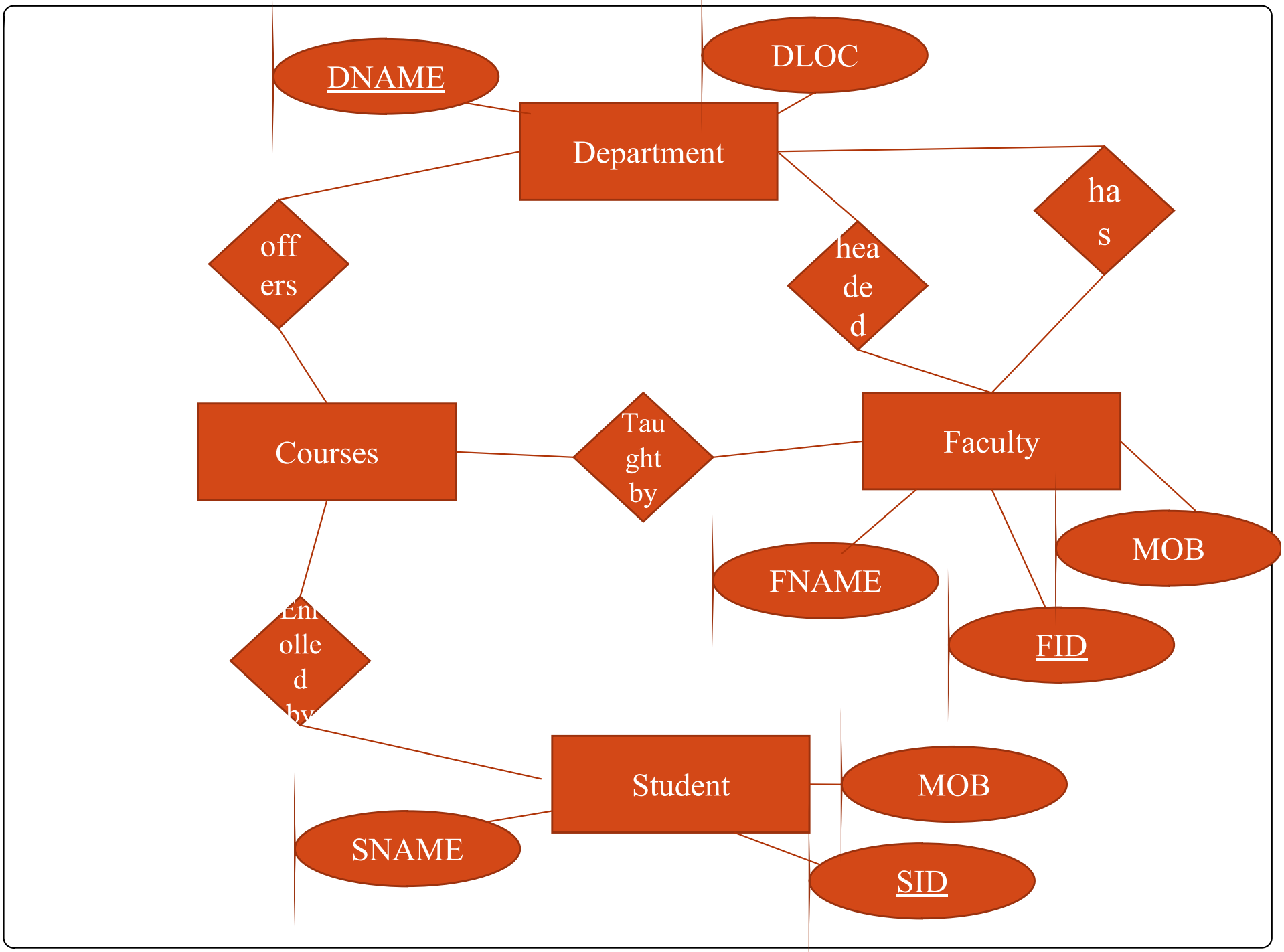
Lets add attributes now

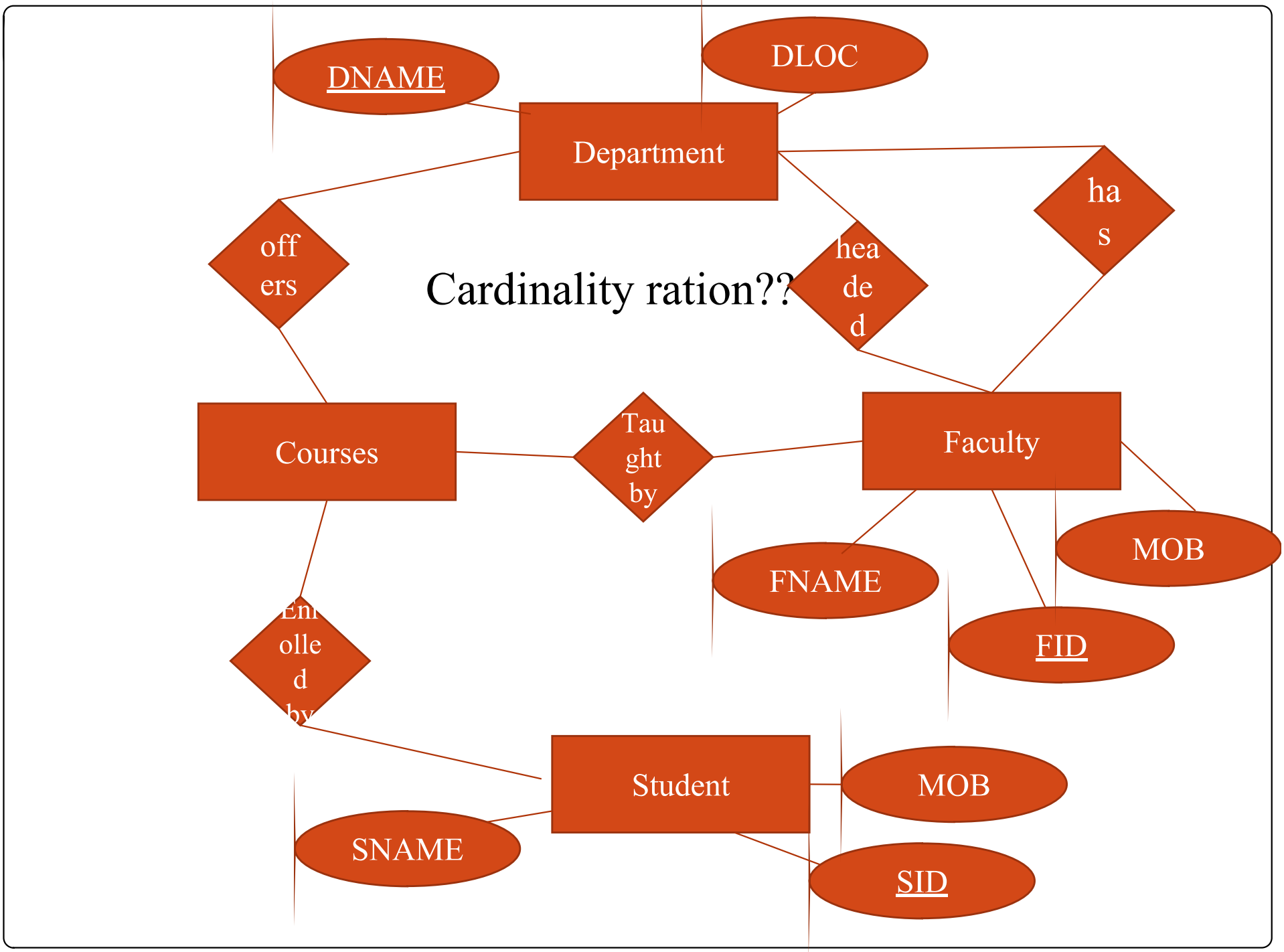


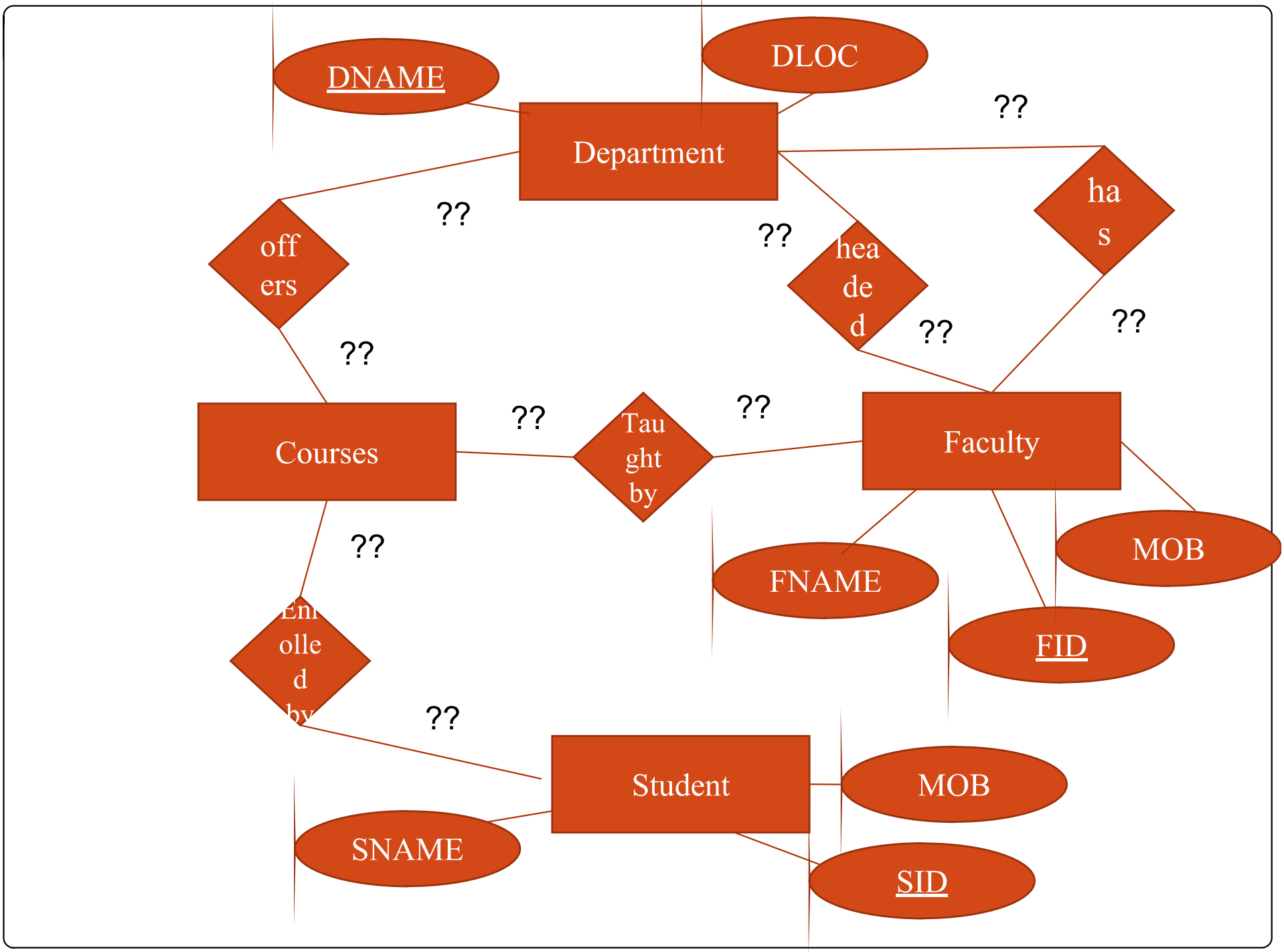

DNAME
DLOC
Department
offers
headed
has
Courses
Taught by
Faculty
FNAME
MOB
FID
Enrolled by
Student
MOB
SNAME
SID

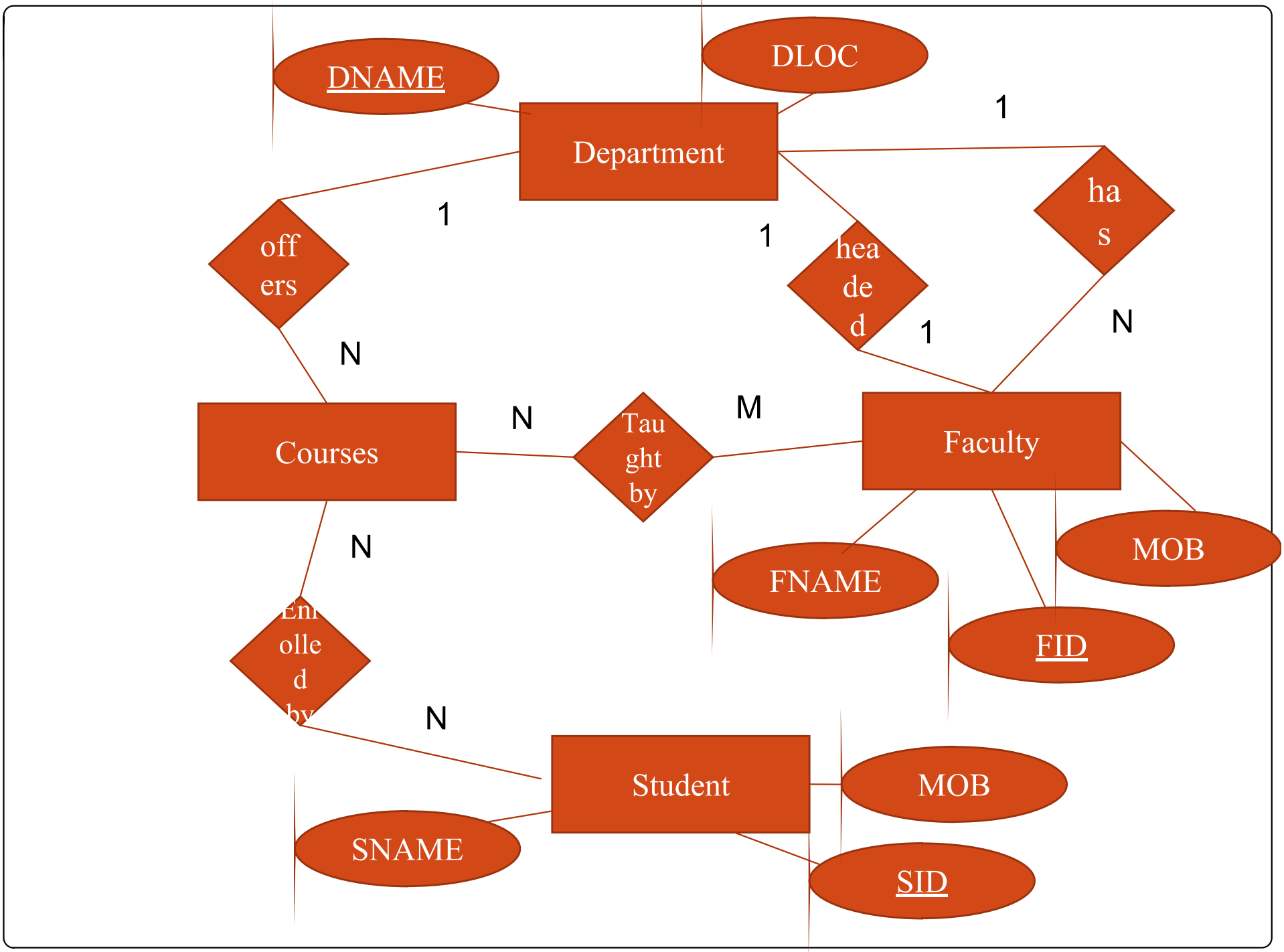
Keys??


DNAME
DLOC
Department
offers
headed
has
Courses
Taught by
Faculty
FNAME
MOB
FID
Enrolled by
Student
MOB
SNAME
SID


DNAME
DLOC
Department
offers
headed
has
Courses
Taught by
Faculty
FNAME
MOB
FID
Enrolled by
Student
MOB
SNAME
SID


DNAME
DLOC
Department
offers
Cardinality ration??
headed
has
Courses
Taught by
Faculty
FNAME
MOB
FID
Enrolled by
Student
MOB
SNAME
SID


DNAME
DLOC
Department
??
??
??
??
??
??
??
??
??
??
??
??
off
ers
hea
de
d
ha
s
Courses
Tau
ght
by
Faculty
FNAME
MOB
FID
Enr
olle
d
by
Student
MOB
SNAME
SID


DNAME
DLOC
Department
1
1
1
1
N
N
N
N
M
N
off
ers
hea
de
d
ha
s
Courses
Tau
ght
by
Faculty
FNAME
MOB
FID
olle
d
by
Student
MOB
SNAME
SID

Constructing an ER model

- Before beginning to draw the ER model, read the requirements specification carefully. Document any assumptions you need to make

1. **Identify entities -** list all potential entity types. These are the object of interest in the system. It is better to put too many entities in at this stage and them discard them later if necessary.

2. **Remove duplicate entities -** Ensure that they really separate entity types or just two names for the same thing.

    Also do not include the system as an entity type

    e.g. if modelling a library, the entity types might be books, borrowers, etc.

3. **List the attributes of each entity** (all properties to describe the entity which are relevant to the application).

- Ensure that the entity types are really needed.
- are any of them just attributes of another entity type?
- Do not have attributes of one entity as attributes of another entity!

4. **Mark the primary keys**.
   - Which attributes uniquely identify instances of that entity type?
   - This may not be possible for some weak entities.
5. **Define the relationships**
   - Examine each entity type to see its relationship to the others.
6. **Describe the cardinality** and optionality of the relationships
   - Examine the constraints between participating entities.
7. **Remove redundant relationships**
   - Examine the ER model for redundant relationships

ER Diagram for Bus Database

- Entities……

ER Diagram for Bus Database

- Entities……
- **Bus -** Company owns busses and will hold information about them.
- **Route -** Buses travel on routes and will need described.
- **Town -** Buses pass through towns and need to know about them
- **Driver -** Company employs drivers, personnel will hold their data.
- **Stage -** Routes are made up of stages
- **Garage -** Garage houses buses, and need to know where they are.

ER Diagram for Bus Database

- Entities and attributes
- **Bus –(b-no, b-name,b-colour,b-type)**
- **Route –(r-no,r-name,r-distance,r-type)**
- **Town –(t-name,t-loc)**
- **Driver –(d-no,d-name,d-age)**
- **Stage – (s-no,s-name,s-loc)**
- **Garage –(g-no,g-name,g-loc)**

ER Diagram for Bus Database

- Entities and attributes & key attributes
- **Bus –(b-no, b-name,b-colour,b-type)**
- **Route –(r-no,r-name,r-distance,r-type)**
- **Town –(t-name,t-loc)**
- **Driver –(d-no,d-name,d-age)**
- **Stage – (s-no,s-name,s-loc)**
- **Garage –(g-no,g-name,g-loc)**

ER Diagram for Bus Database

- Entities and attributes & key attributes
- **Bus –(b-no, b-name,b-colour,b-type)**
- **Route –(r-no,r-name,r-distance,r-type)**
- **Town –(t-name,t-loc)**
- **Driver –(d-no,d-name,d-age)**
- **Stage – (s-no,s-name,s-loc)**
- **Garage –(g-no,g-name,g-loc)**

For your understanding!!!!!

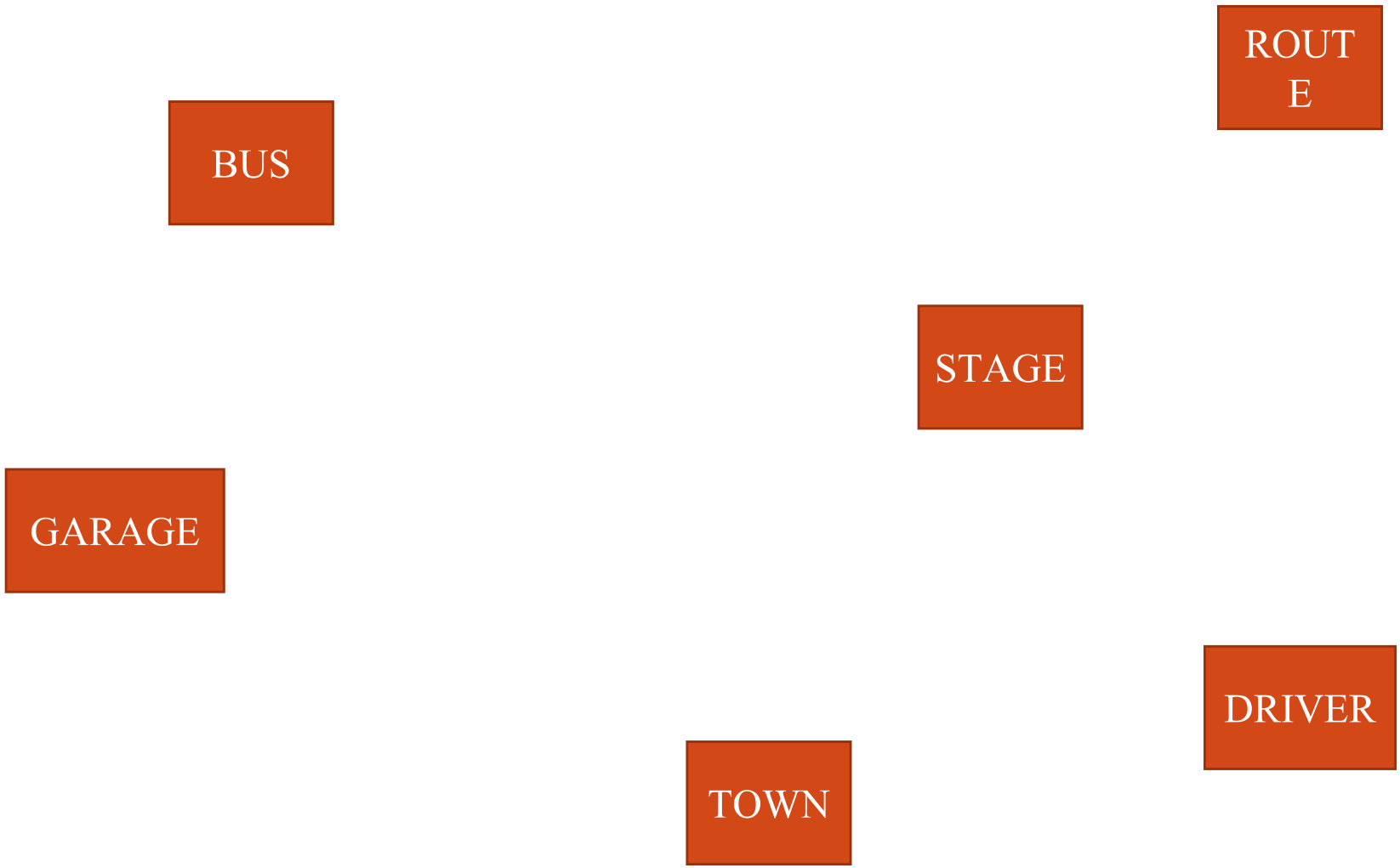
- A bus is allocated to a route and a route may have several buses.
- A route has of one or more stages.
- One or more drivers are allocated to each stage.
- A stage passes through some or all of the towns on a route.
- Some of the towns have a garage
- A bus is garaged in garage.

Find out the relation words

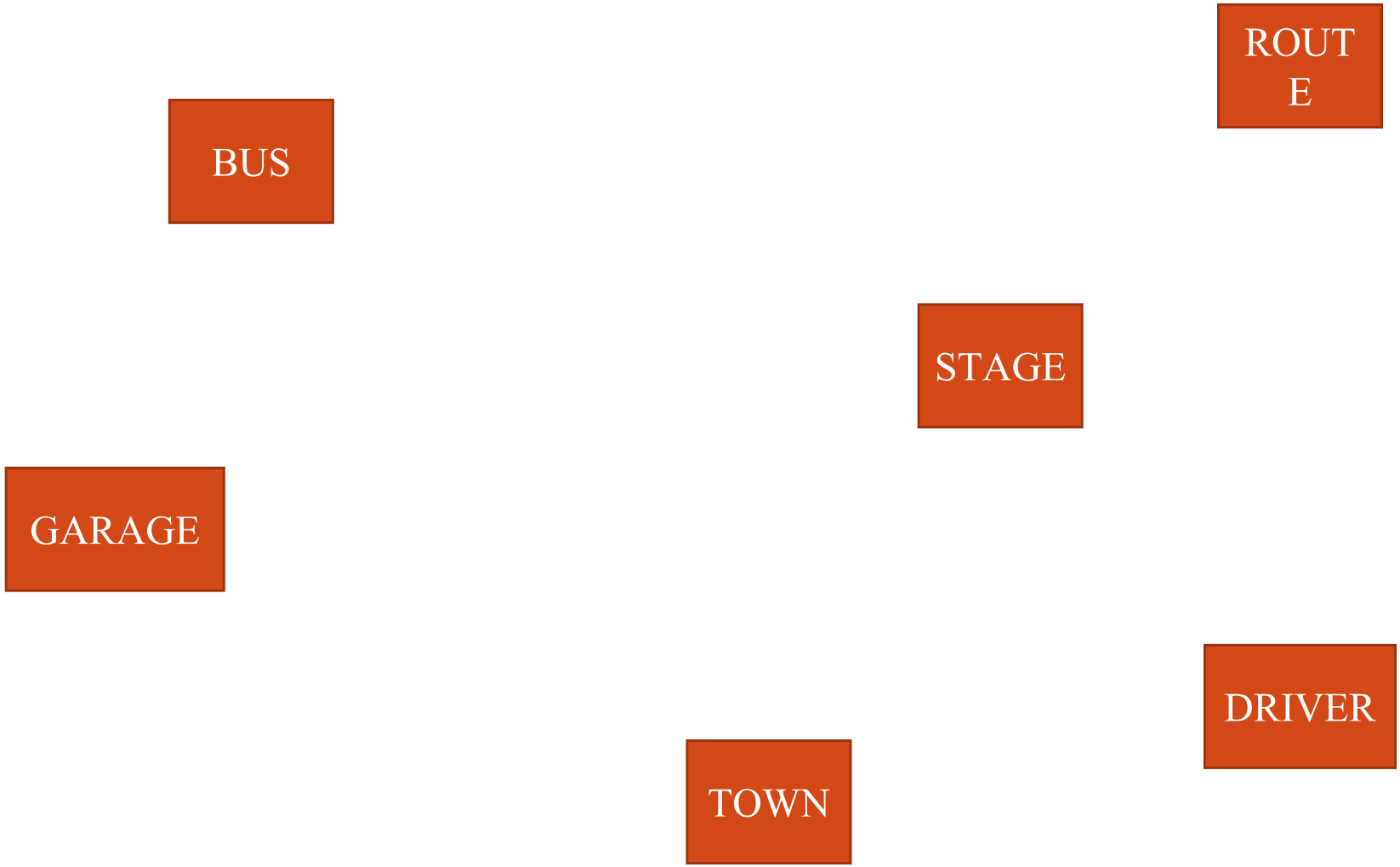
For your understanding!!!!!

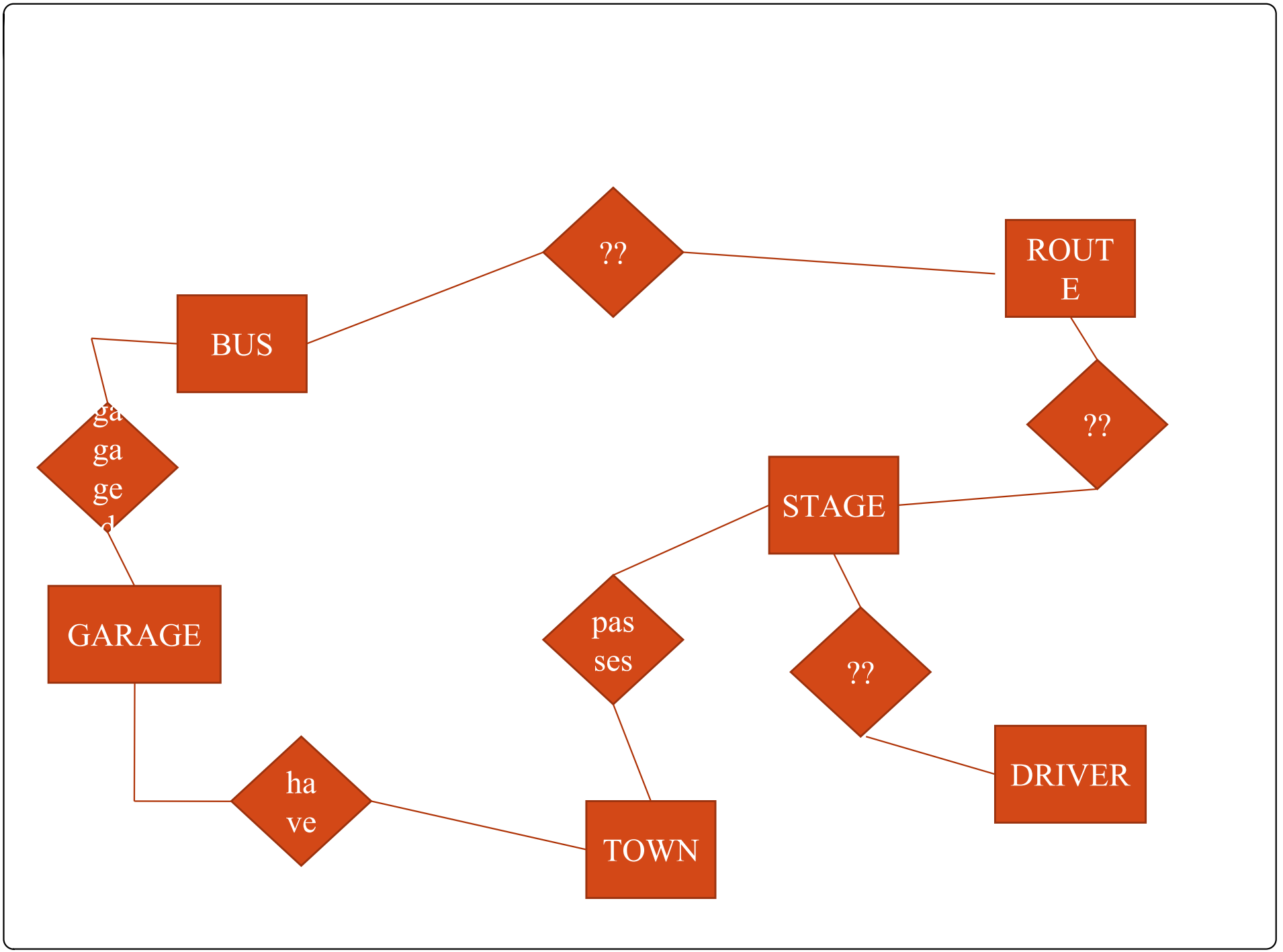
- A bus is allocated to a route and a route may have several buses.
- A route has of one or more stages.
- One or more drivers are allocated to each stage.
- A stage passes through some or all of the towns on a route.
- Some of the towns have a garage
- A bus is garaged in garage.

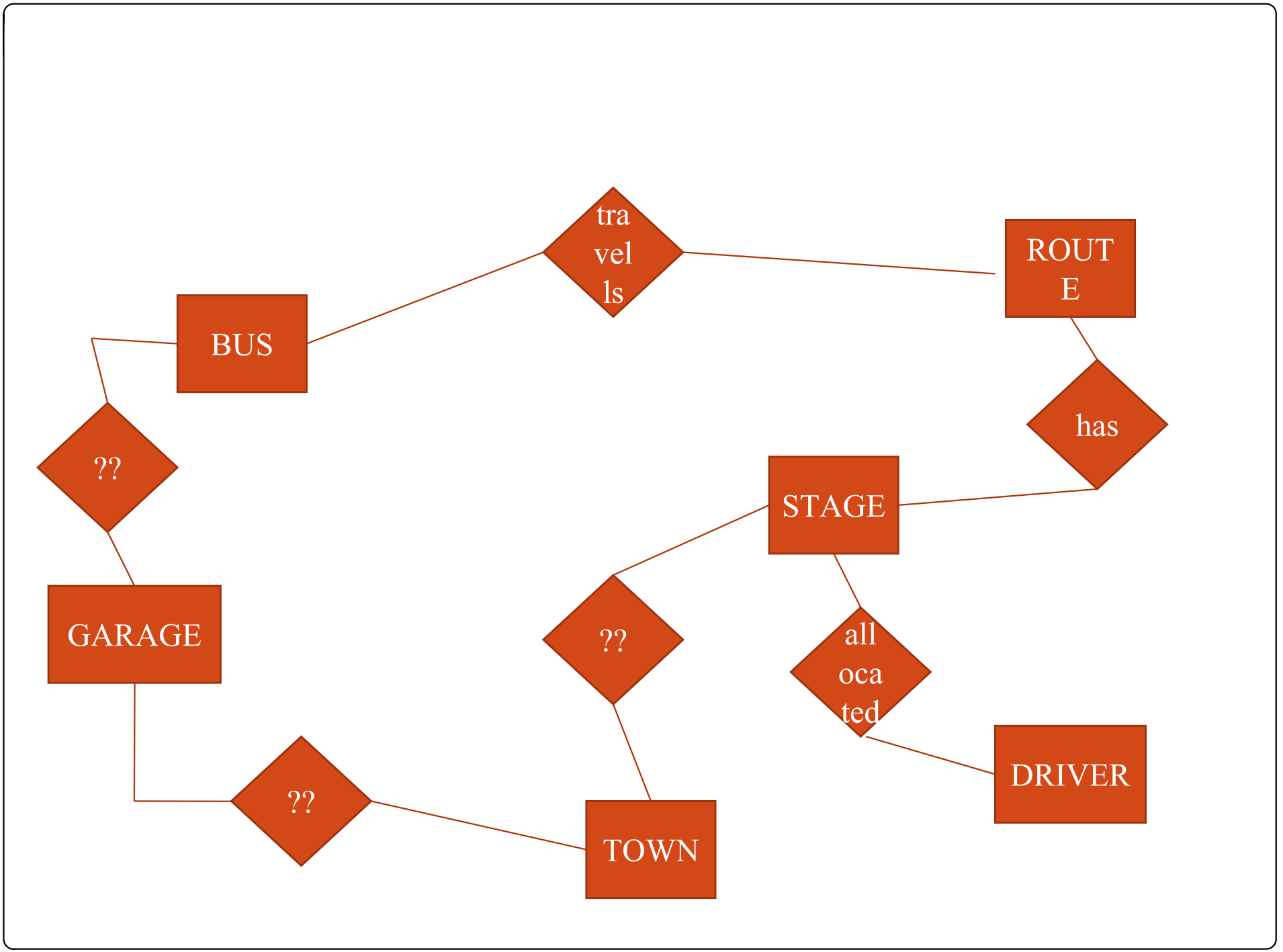
Place the entities



Place the entities.. Add relations




??
ROUTE
BUS
??
gaged
STAGE
GARAGE
passes
??
DRIVER
have
TOWN


tra
vel
ls
ROUT
E
BUS
has
??
STAGE
GARAGE
??
all
oca
ted
??
DRIVER
TOWN

- Entity and Relationship sets for the hospital called General Hospital

- Entity with attributes::
- Doctor: Specialist, num, addr, mob
- Patient: name, id, address, gender,
- Beds: room number, type, bed number, status, price
- Accounts: maintains date in date out and bill

Logical part!!

- Beds are assigned to patients….
- Doctor examines patient
- Patient has account

you people draw it now !!!!!!!!!

An ER Diagram for FACEBOOK

- **Entities**
- **LIST the entities !!!!!!**

An ER Diagram for FACEBOOK

- **Entities**
  - Users
  - School/College
  - Location
  - Album
  - Photos
  - Events

Listing of attributes

- Users (uid, fname,lname,gender,dob)
- School/College(collegeid,college_name)
- Location (loc_id,city,state,country)
- Album(aid,ownerid,createtime,aname,visible)
- Photos(pid,link,caption,ptime)
- Events(eid,estartdate,eenddate,loc,ename,cid,desp)

Mark key attributes now

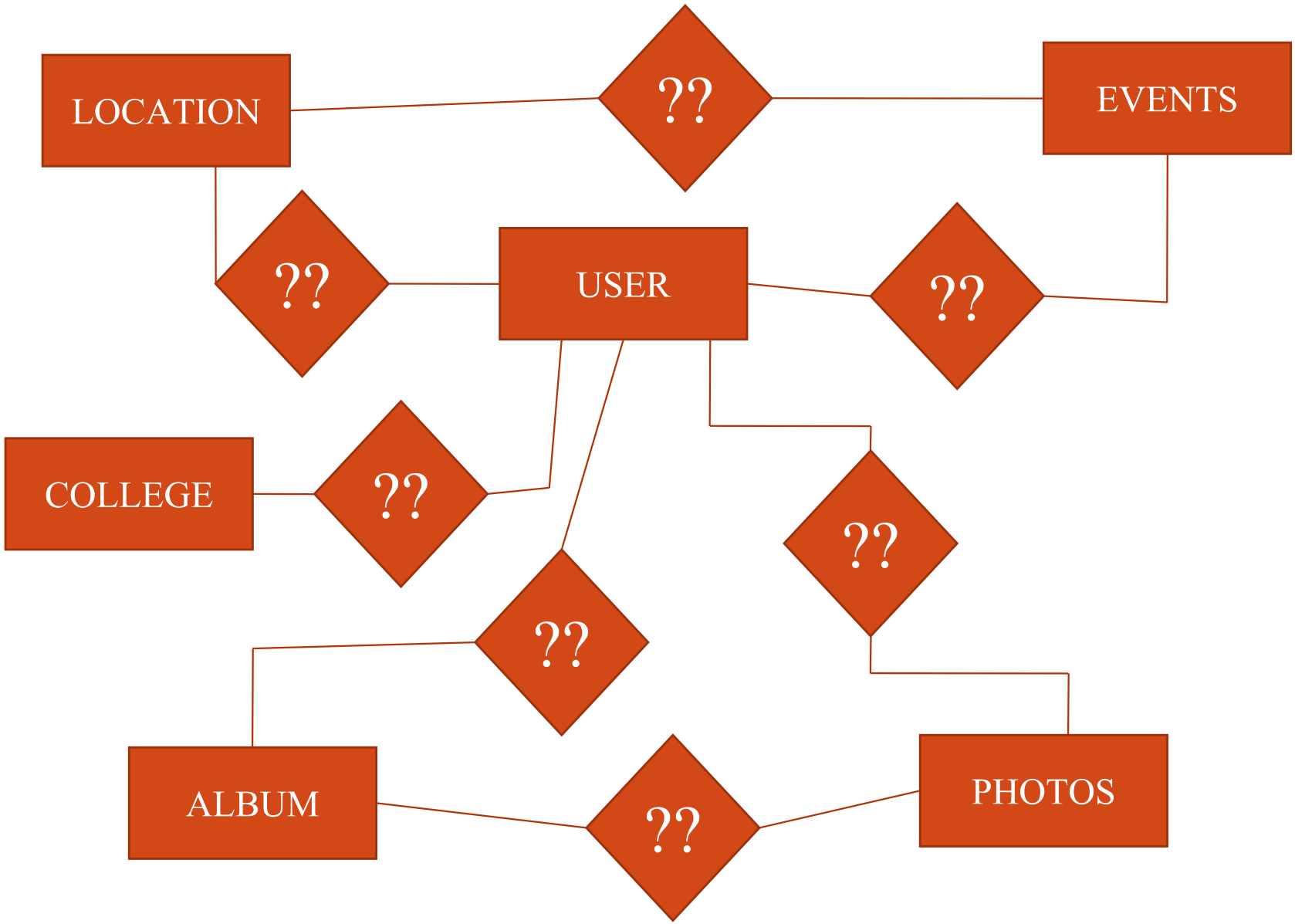
- Users (uid, fname,lname,gender,dob)
- School/College(collegeid,college_name)
- Location (loc_id,city,state,country)
- Album(aid,ownerid,createtime,aname,visible)
- Photos(pid,link,caption,ptime)
- Events(eid,estartdate,eenddate,loc,ename,cid,desp)

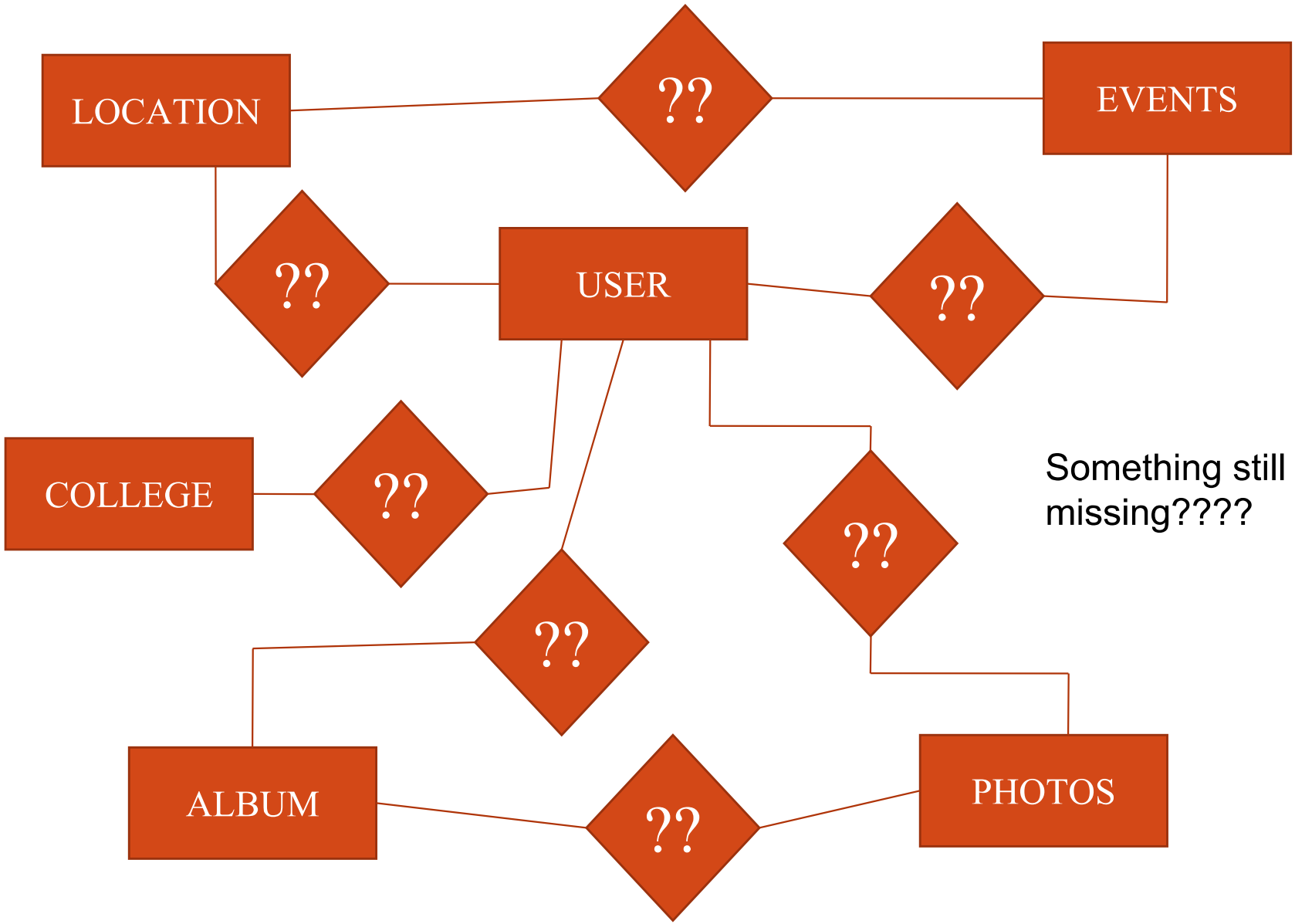
Mark key attributes now

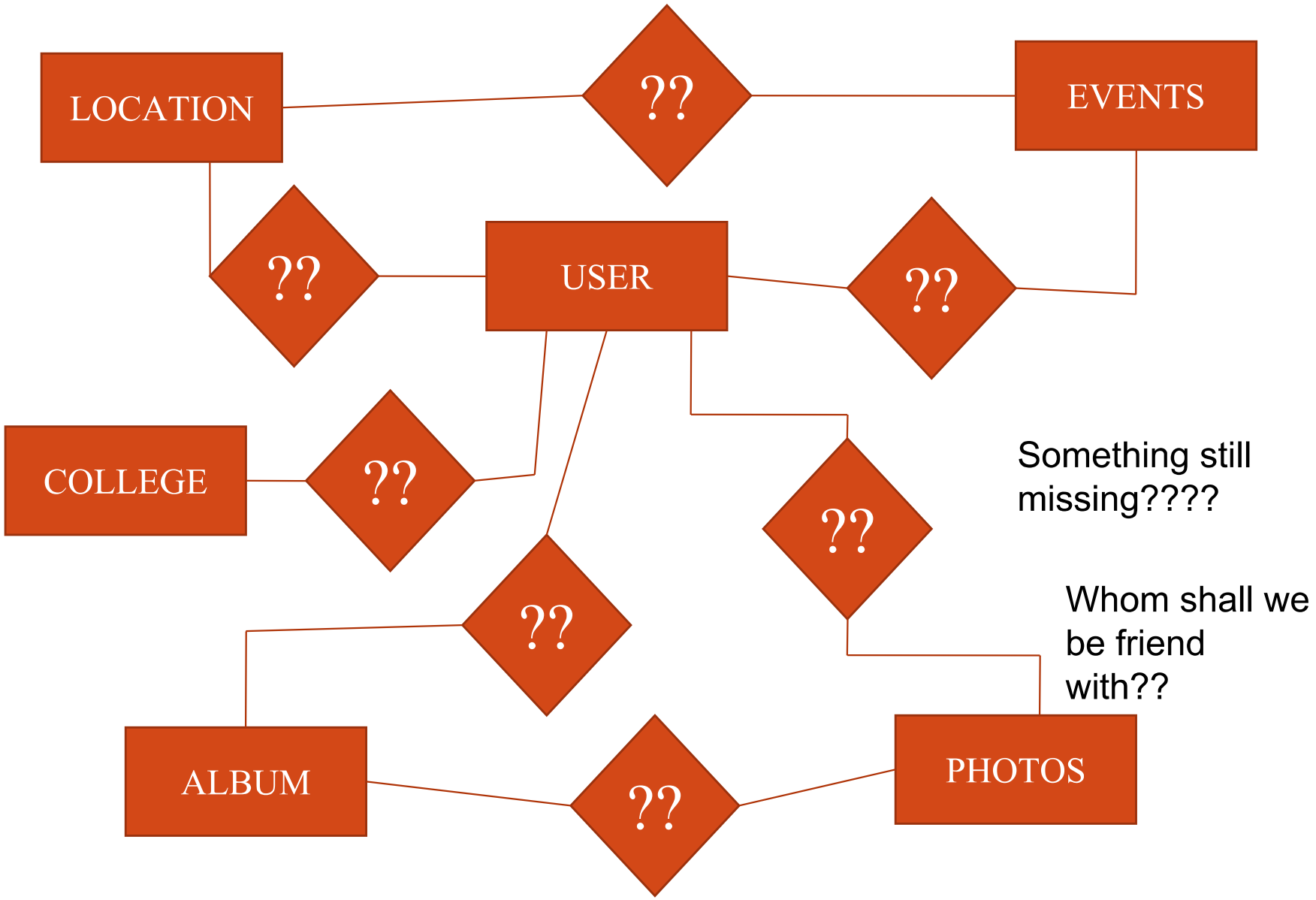
- Users (**uid**, fname,lname,gender,dob)
- School/College(**collegeid**,college_name)
- Location (**loc_id**,city,state,country)
- Album(**aid**,ownerid,createtime,aname,visible)
- Photos(**pid**,link,caption,ptime)
- Events(**eid**,estartdate,eenddate,loc,ename,cid,desp)

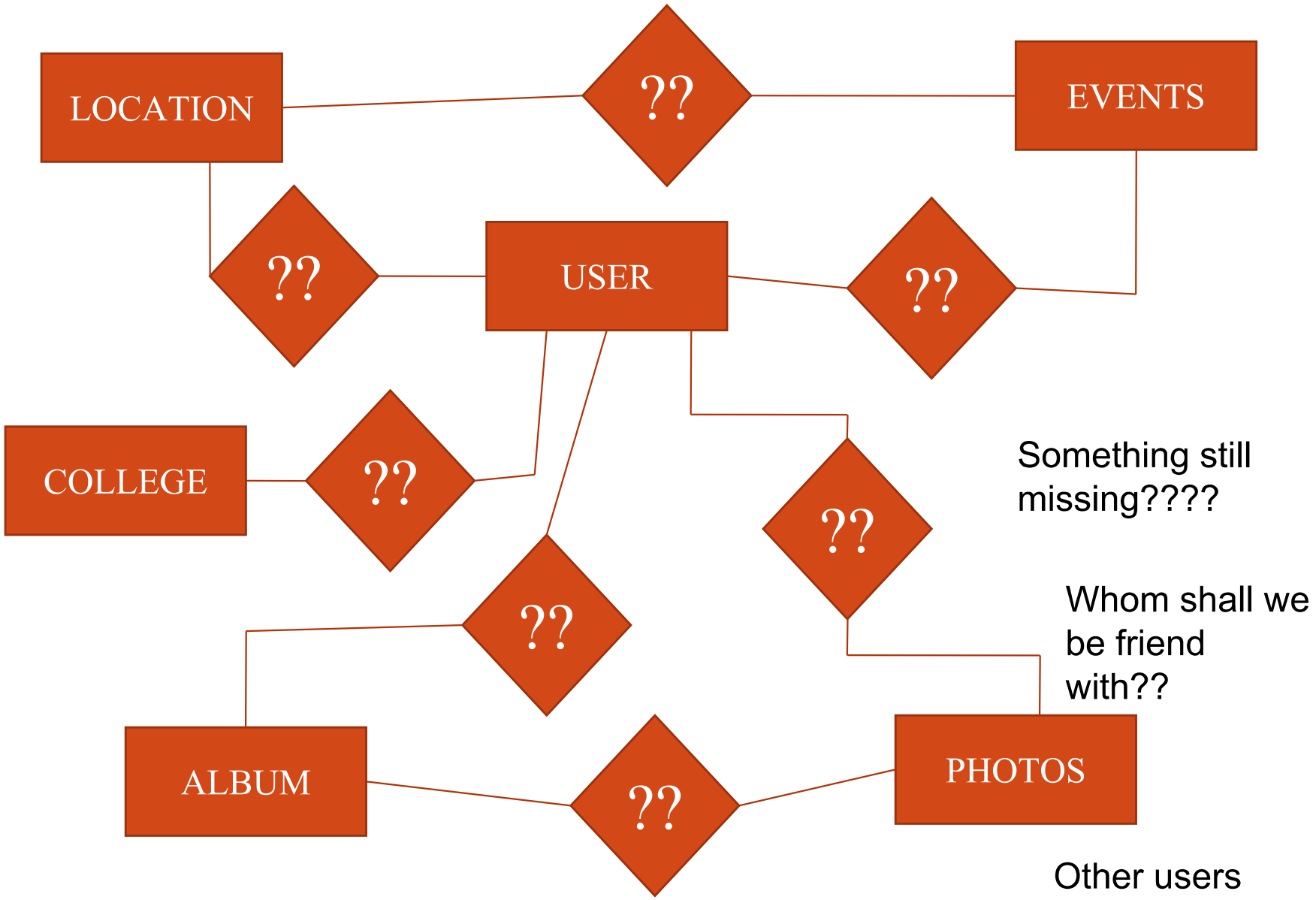
Relationship now

- User stays at Location
- User attended College
- User owns Album
- Album contains Photos
- User Tagged in Photos
- User creates Events
- Events occur at Location


LOCATION
??
EVENTS
??
USER
??
COLLEGE
??
??
??
ALBUM
??
PHOTOS

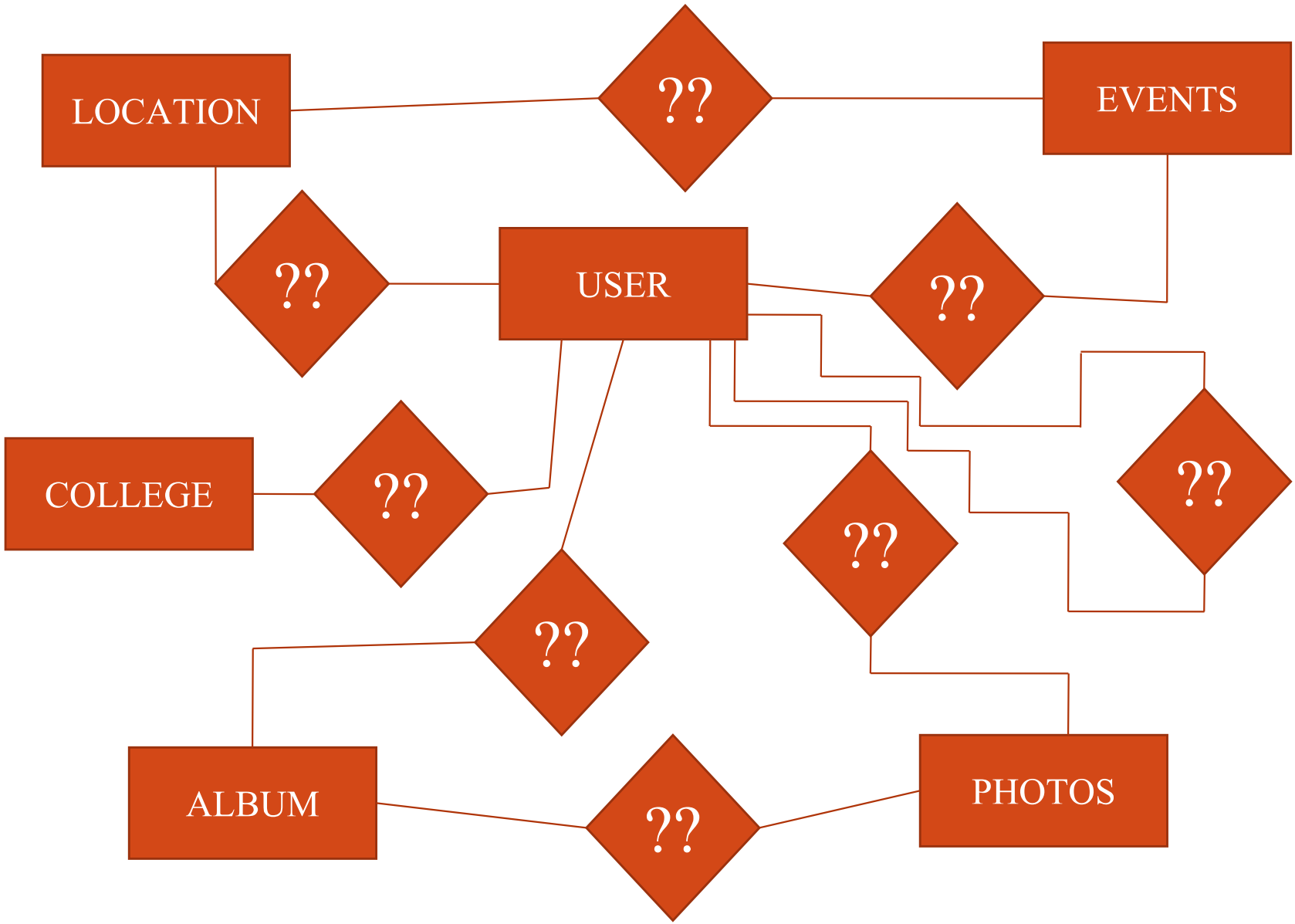

LOCATION
??
EVENTS
??
USER
??
COLLEGE
??
??
??
ALBUM
??
PHOTOS
Something still missing????

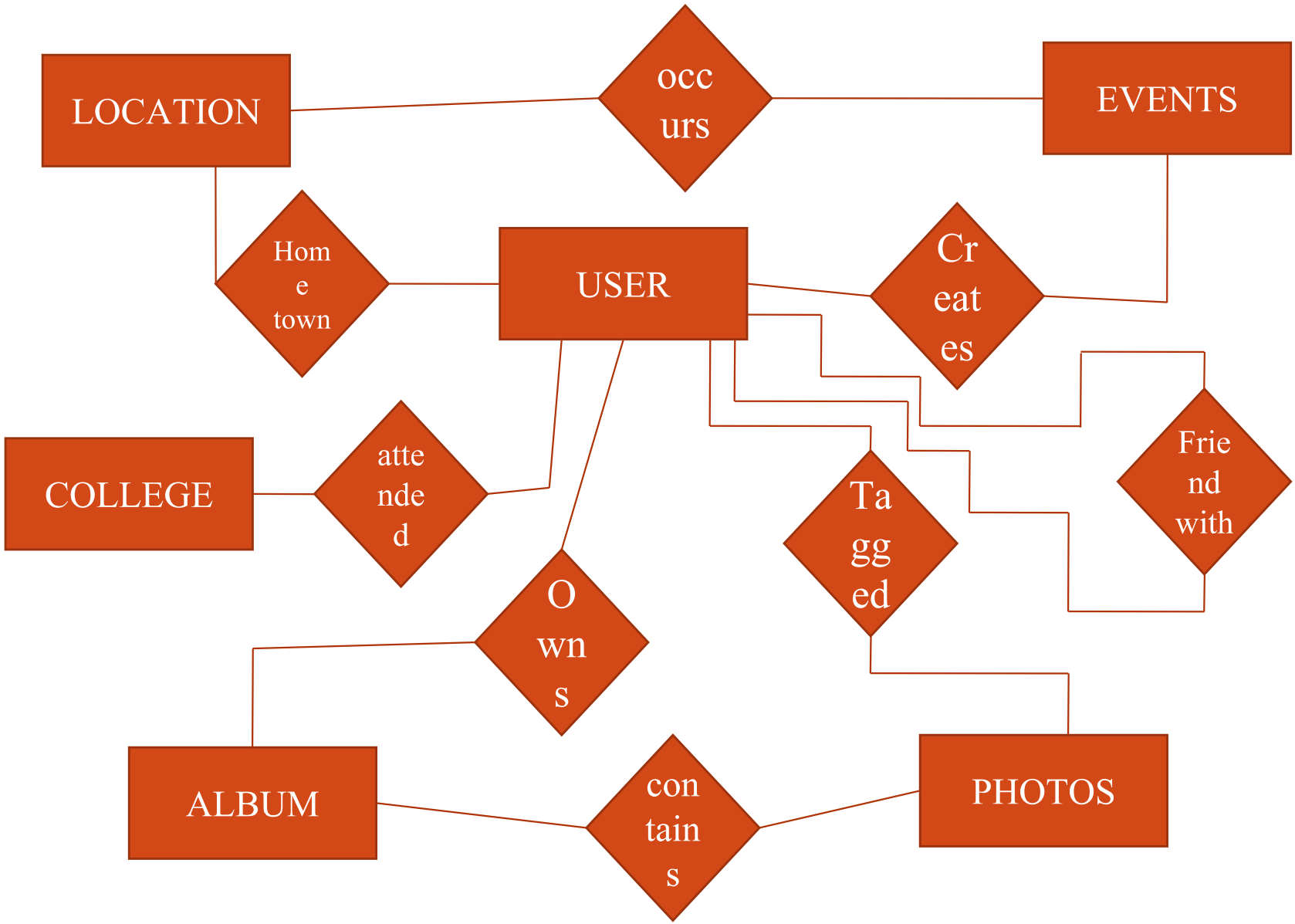

LOCATION
??
EVENTS
??
USER
??
COLLEGE
??
??
??
??
ALBUM
??
PHOTOS
Something still missing????
Whom shall we be friend with??


LOCATION
??
EVENTS
??
USER
??
COLLEGE
??
??
??
ALBUM
??
PHOTOS
Something still missing????
Whom shall we be friend with??
Other users

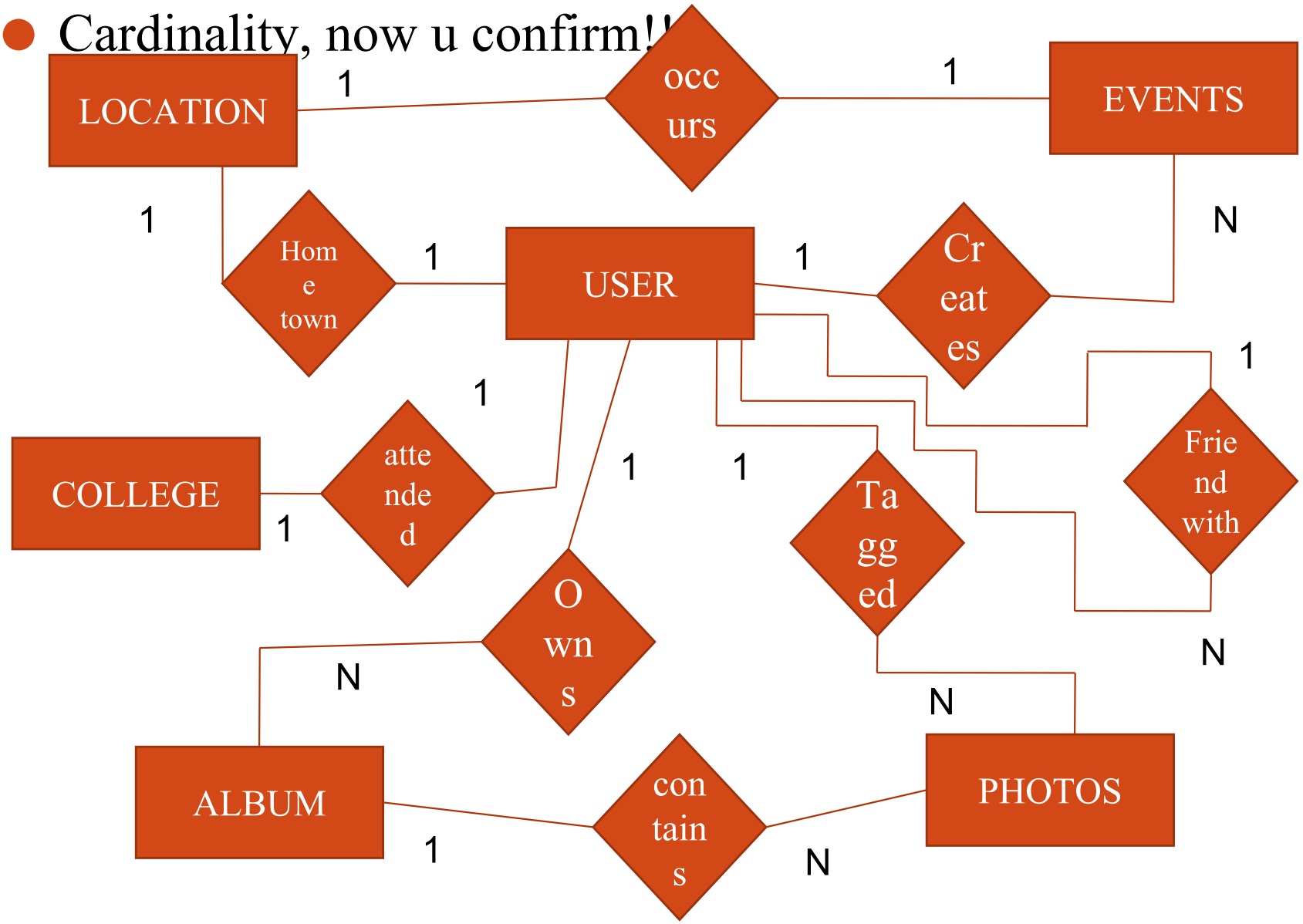
Relationship now

- User stays at Location
- User attended School
- User owns Album
- Album contains Photos
- User Tagged in Photos
- User creates Events
- Events occur at Location
- User friendwith User


LOCATION
??
EVENTS
??
USER
??
COLLEGE
??
??
??
??
ALBUM
??
PHOTOS

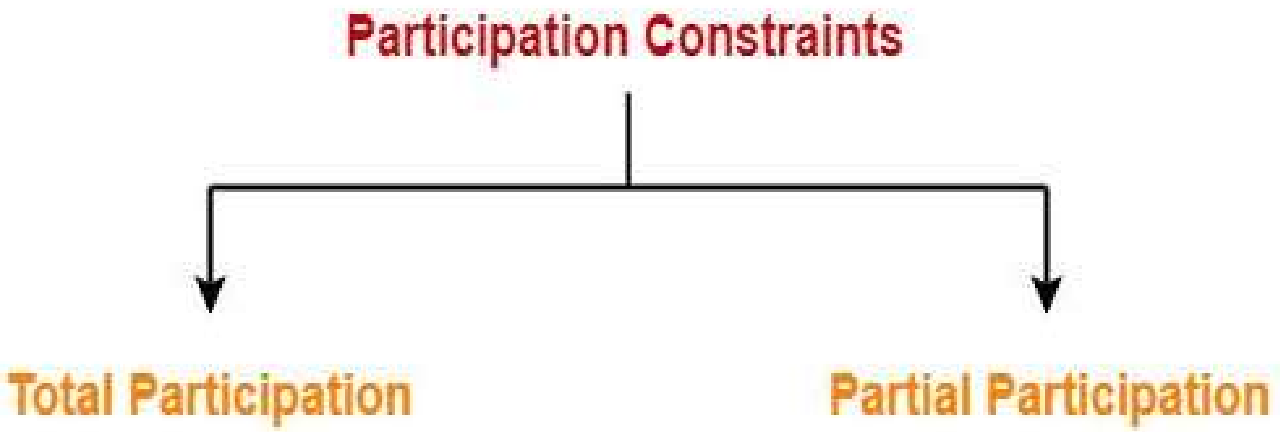

LOCATION
occ
urs
EVENTS
Hom
e
town
USER
Cr
eat
es
COLLEGE
atte
nde
d
Ta
gg
ed
Frie
nd
with
O
wn
s
ALBUM
con
tain
s
PHOTOS

- Cardinality, now u confirm!!



Types of Participation Constraints-

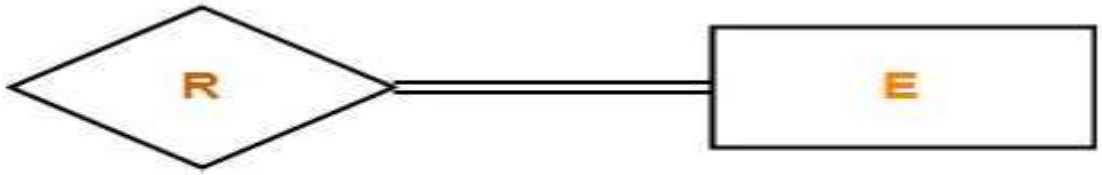
There are two types of participation constraints-



1. Total participation
2. Partial participation

1. Total Participation-

- It specifies that each entity in the entity set must compulsorily participate in at least one relationship instance in that relationship set.
- That is why, it is also called as **mandatory participation.**
- Total participation is represented using a double line between the entity set and relationship set.



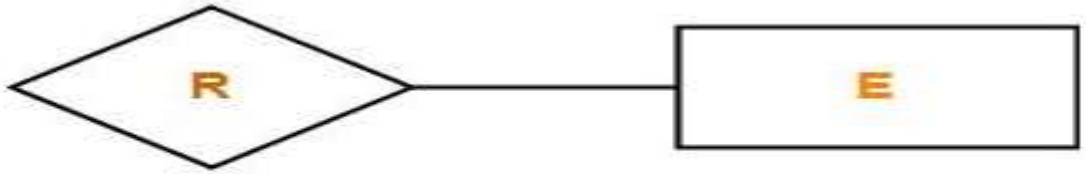
Total Participation

- **Example-**

- Here,
- Double line between the entity set “Student” and relationship set “Enrolled in” signifies total participation.
- It specifies that each student must be enrolled in at least one course.

2. Partial Participation-

- It specifies that each entity in the entity set may or may not participate in the relationship instance in that relationship set.
- That is why, it is also called as **optional participation.**
- Partial participation is represented using a single line between the entity set and relationship set.



Partial Participation

- **Example-**

- Here,
- Single line between the entity set “Course” and relationship set “Enrolled in” signifies partial participation.
- It specifies that there might exist some courses for which no enrollments are made.

Few more examples: Example - 1

- Customers borrows loan.

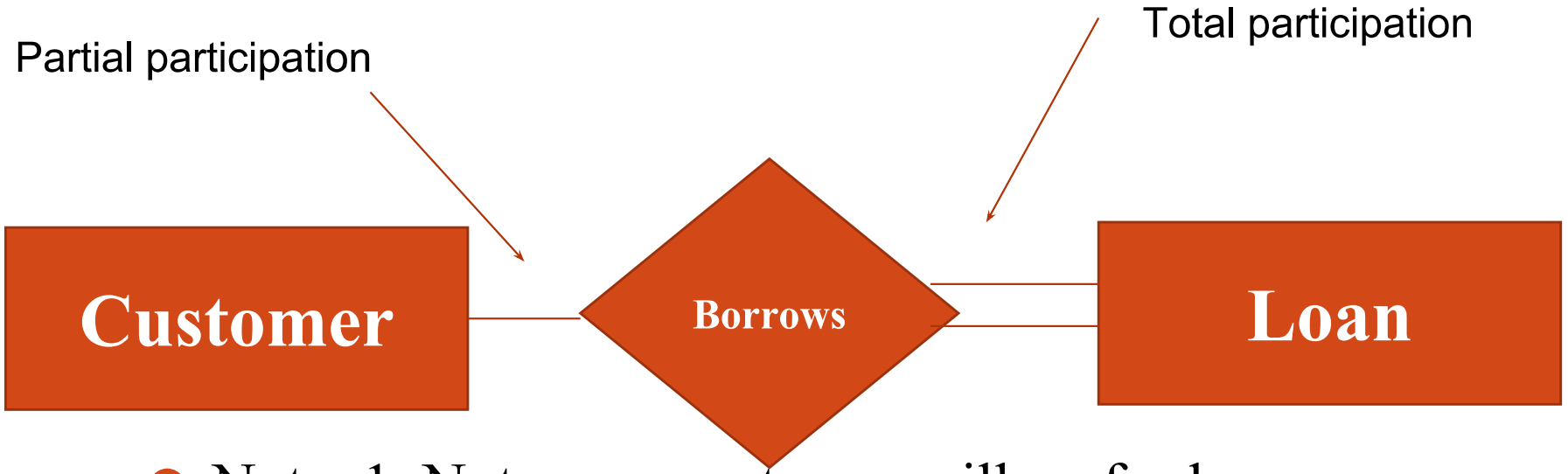
Few more examples : Example - 1

- Customers borrows loan.

- Note: 1. Not every customer will go for loan.
- 2. But every loan is associated with a customer

Few more examples: Example - 1

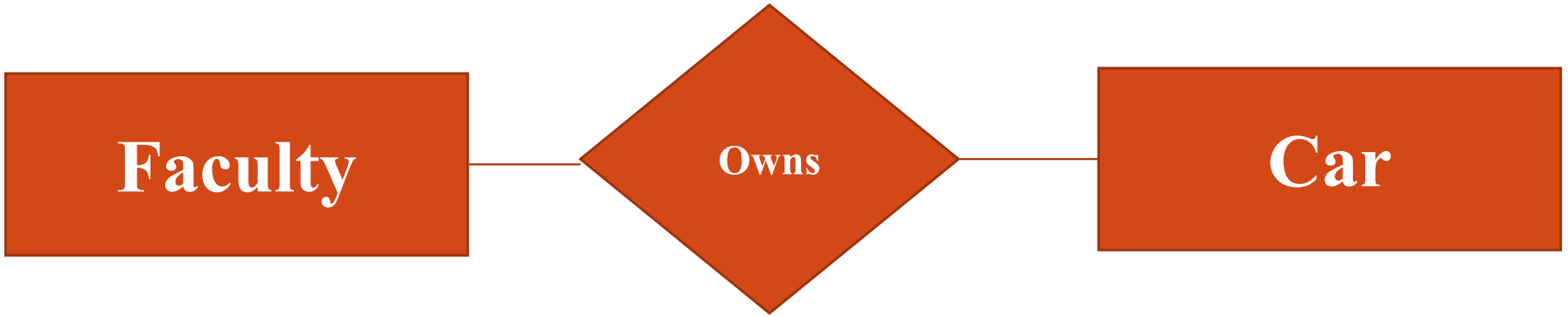
- Customers borrows loan.



- Note: 1. Not every customer will go for loan.
- 2. But every loan is associated with a customer

Example:2

- Consider the relationship owns between FACULTY and CAR.

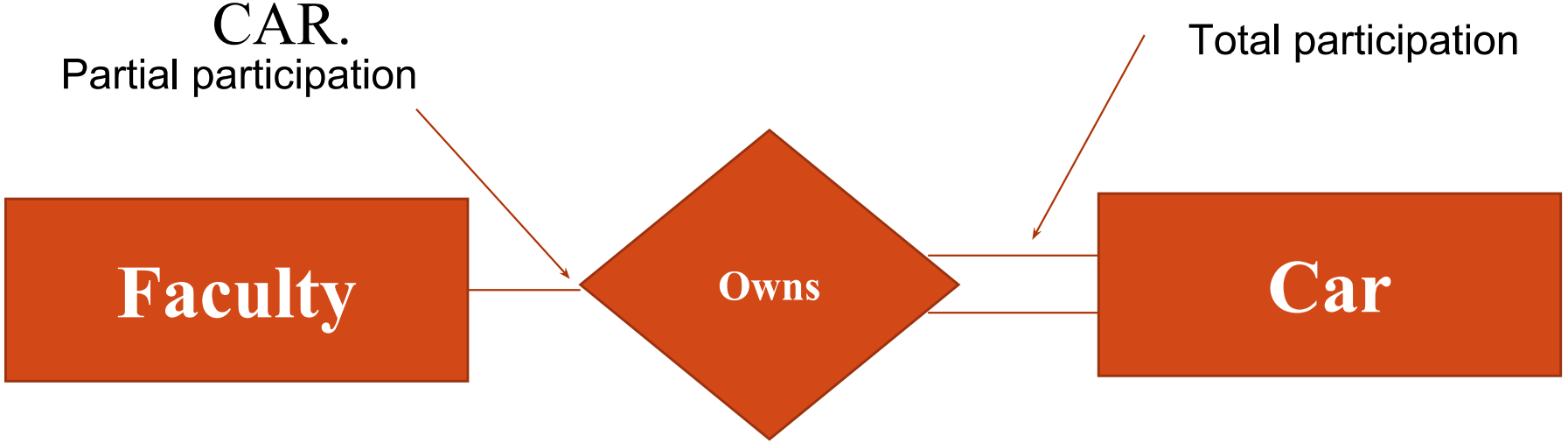


Example:2

- The CAR entity *totally participate* in the relationship indicated by double line from CAR to relationship and the FACULTY is *partially participate* in the relationship indicated by single line from FACULTY to relationship.
- Total participation indicates that a CAR must associated to atleast one FACULTY. If a CAR is in database then there must be a FACULTY associate with it.

Example:2

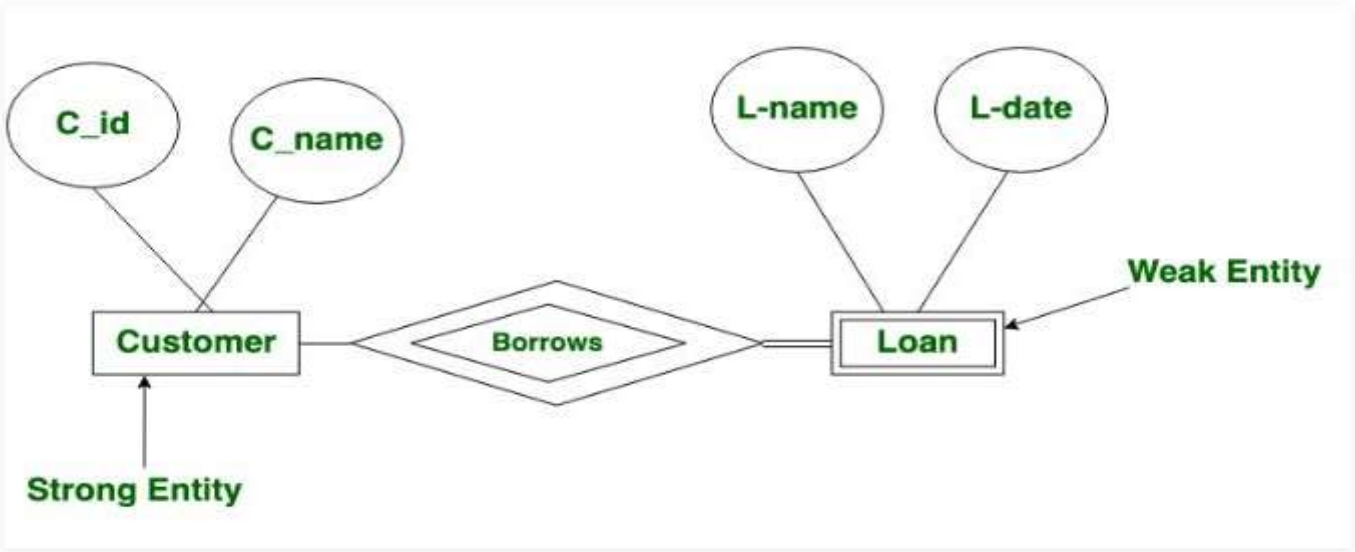
- Consider the relationship owns between FACULTY and CAR.



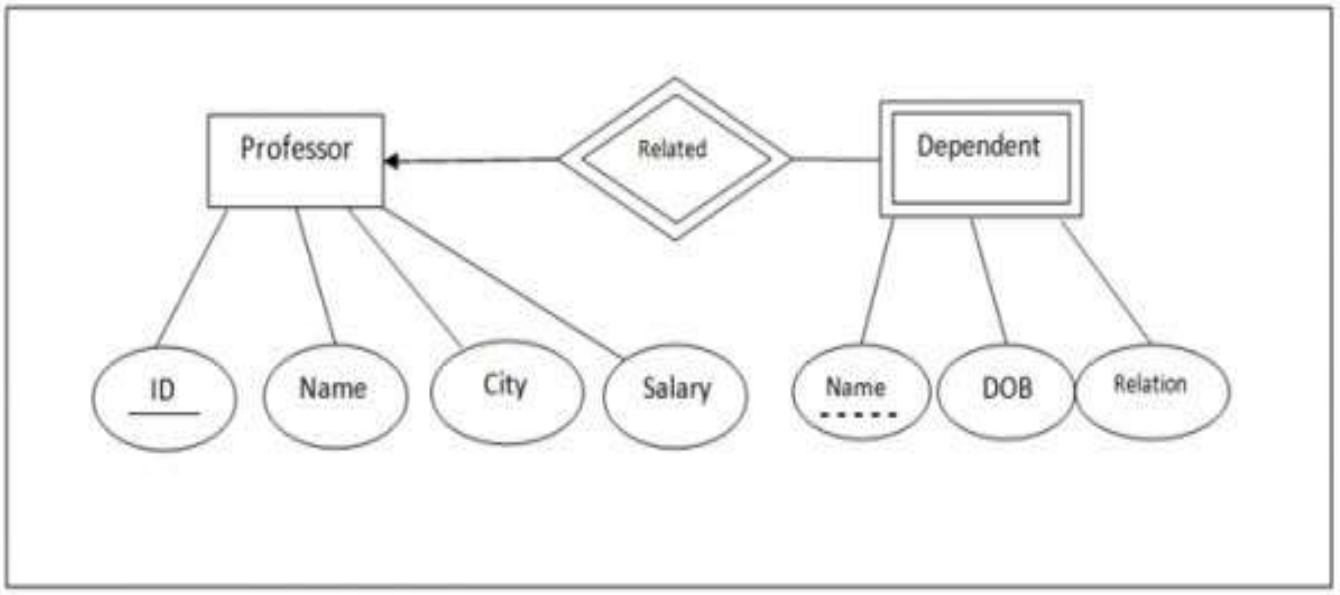
Weak Entity

- A weak entity is an entity that depends on the existence of another entity. In more technical terms it can defined as an entity that cannot be identified by its own attributes.
- E.g. A weak entity is represented by a double rectangle.

ity is



- **Professor** is a strong entity, and the primary key is **Professor_ID**. However, another entity is **Professor_Dependents**, which is our Weak Entity.
- This is a weak entity since its existence is dependent on another entity **Professor**



ER Diagram for placing Amazon order

- Entities
  - Customer
  - Orders
  - Order items
  - Products

ER Diagram for placing Amazon order

- Entities & Attributes
  - Customer(Cust_no, C_Name, C-Addr, Email, Mob)
  - Orders(Order_n0,Order_date, Payment_info,Shipp_addr)
  - Order items(Item_no, quantity, Price, discount)
  - Products(Pro_num, P_price, P_type, P_description)

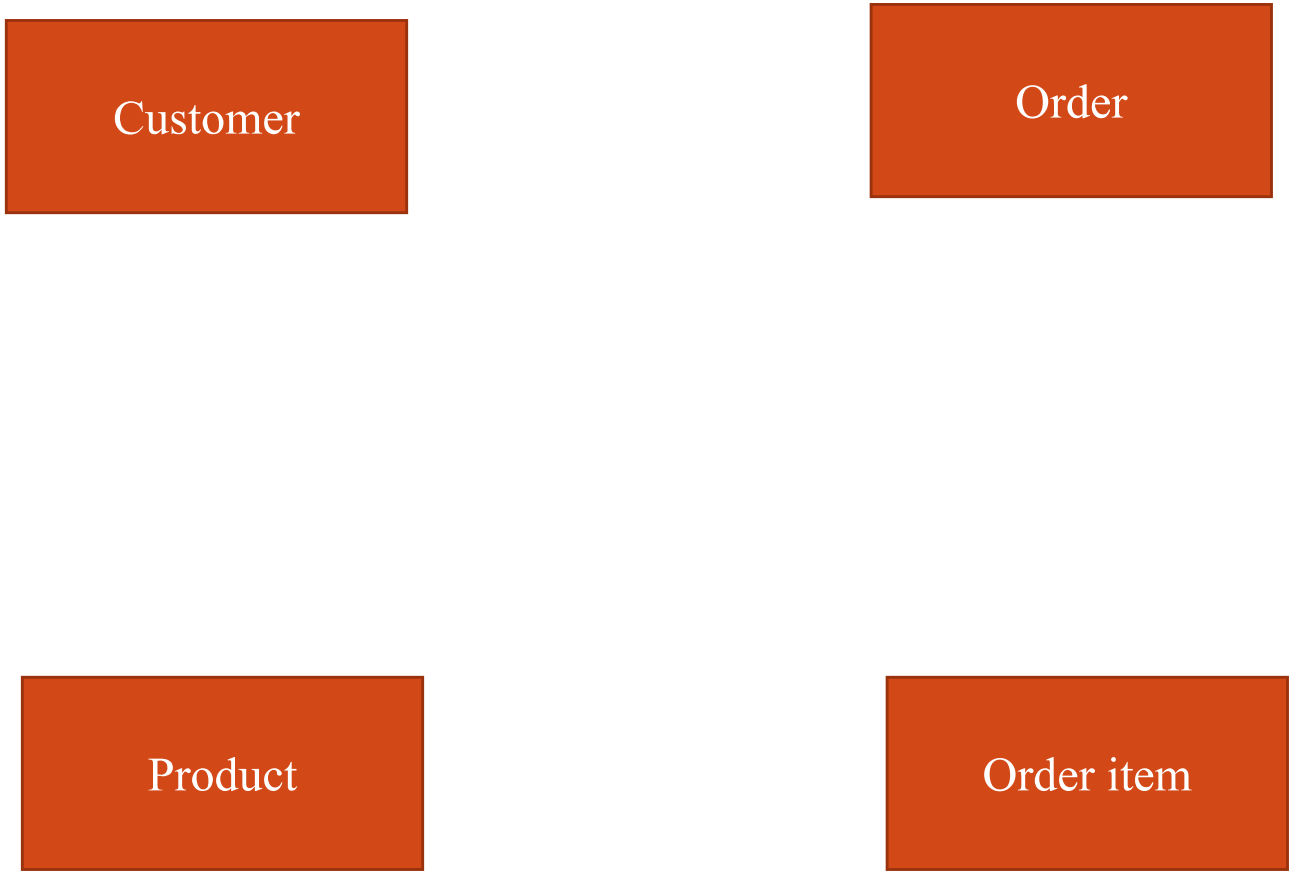
ER Diagram for placing Amazon order

- Entities & Attributes & Key attributes
  - Customer(**Cust_no**, C_Name, C-Addr, Email, Mob)
  - Orders(**Order_n0**,Order_date, Payment_info,Shipp_addr)
  - Order items(**Item_no**, quantity, Price, discount)
  - Products(**Pro_num**, P_price, P_type, P_description)

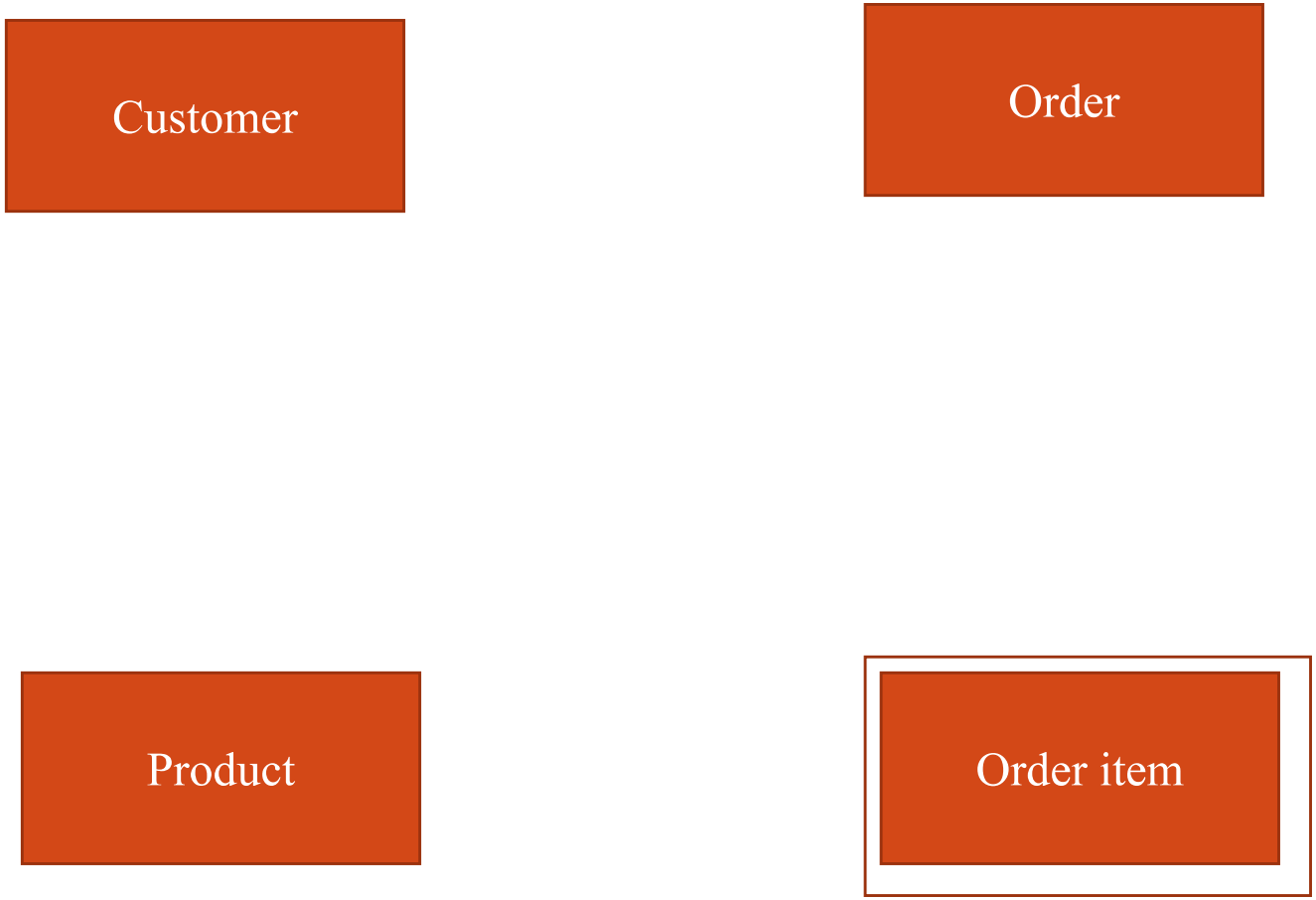
ER Diagram for placing Amazon order

- Relations
  - Customer places Order
  - Order contains Order items
  - Order items contains products

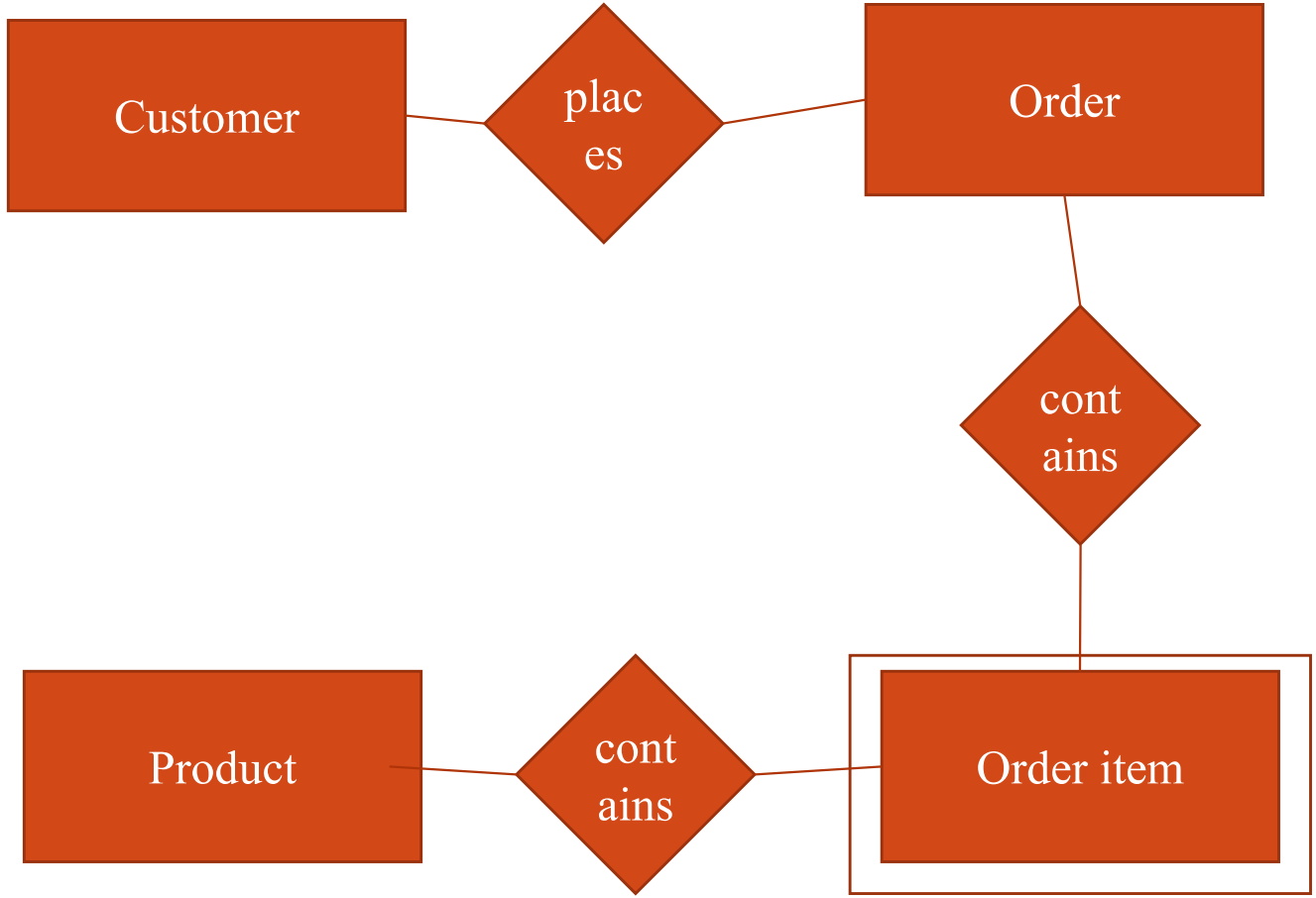
- Entities



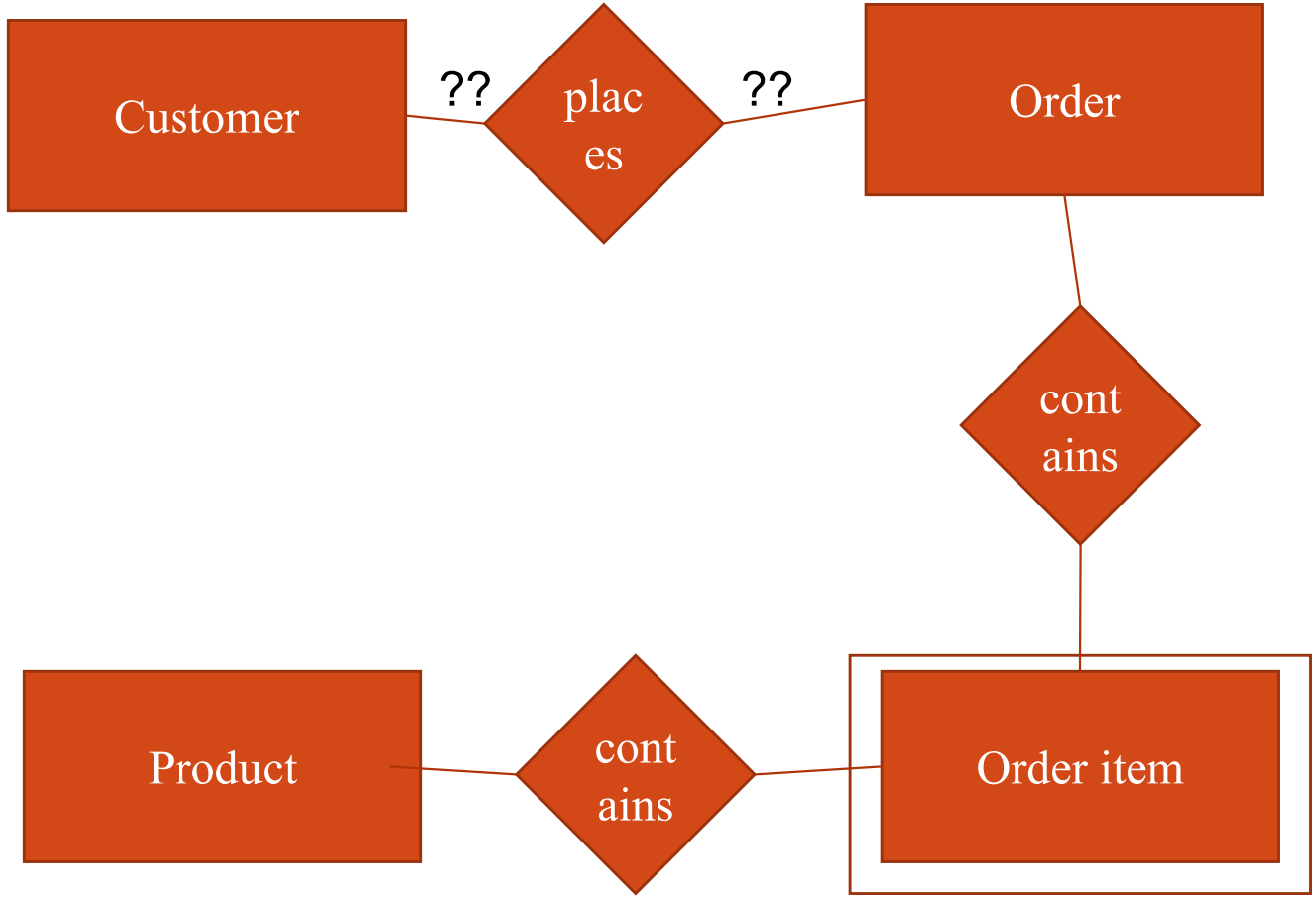
- Entities



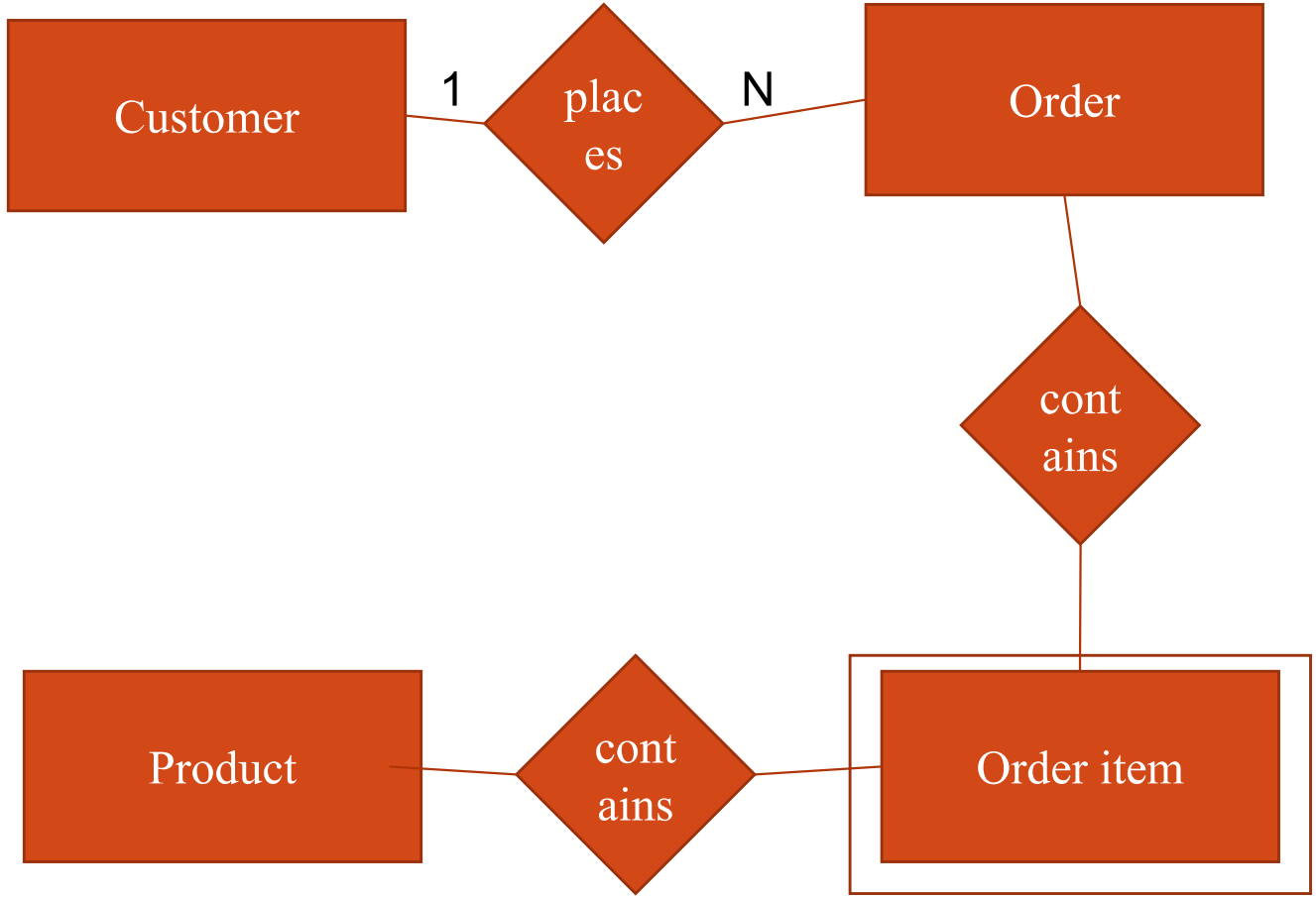
- Entities & Relations



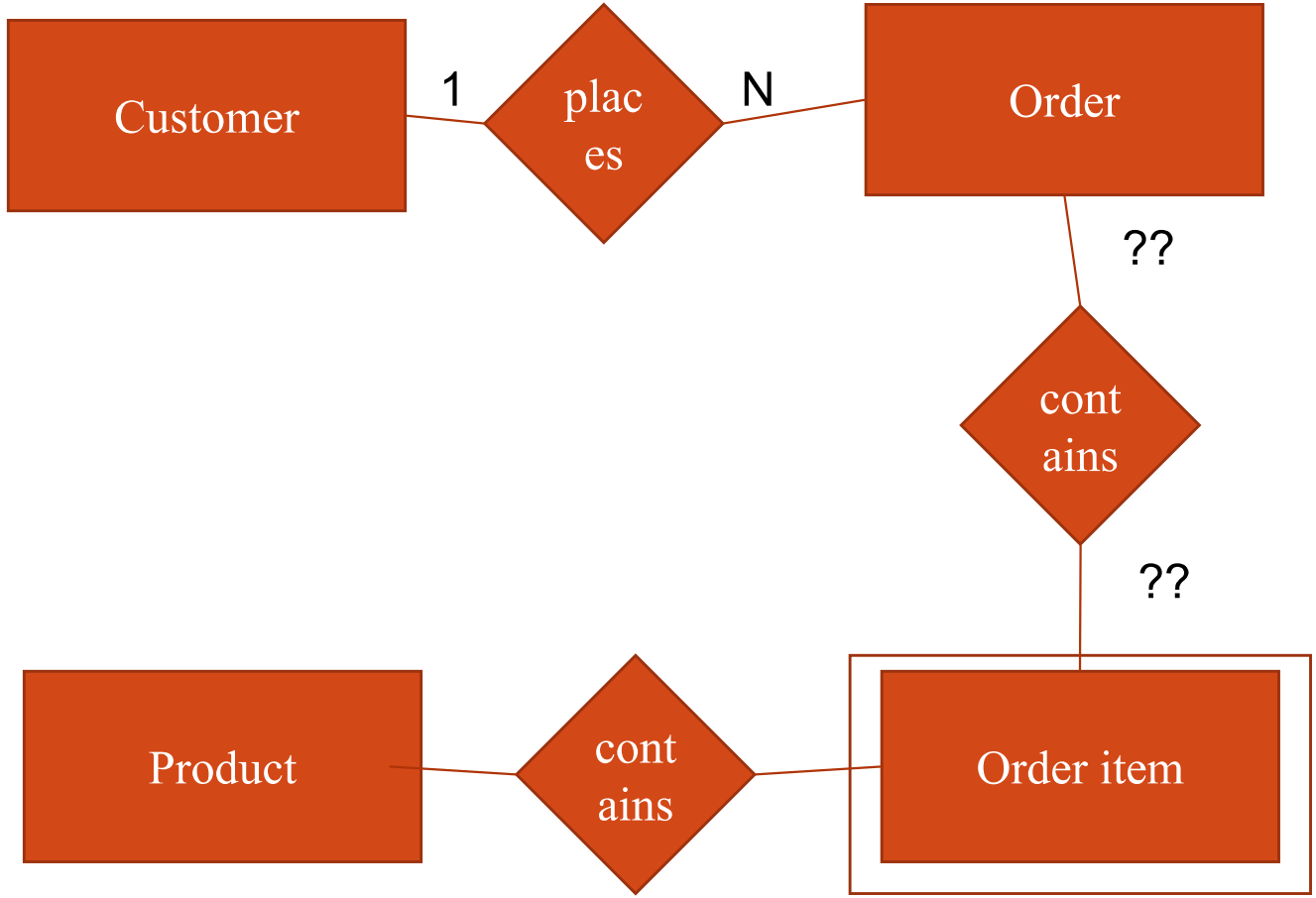
- Entities & Relations



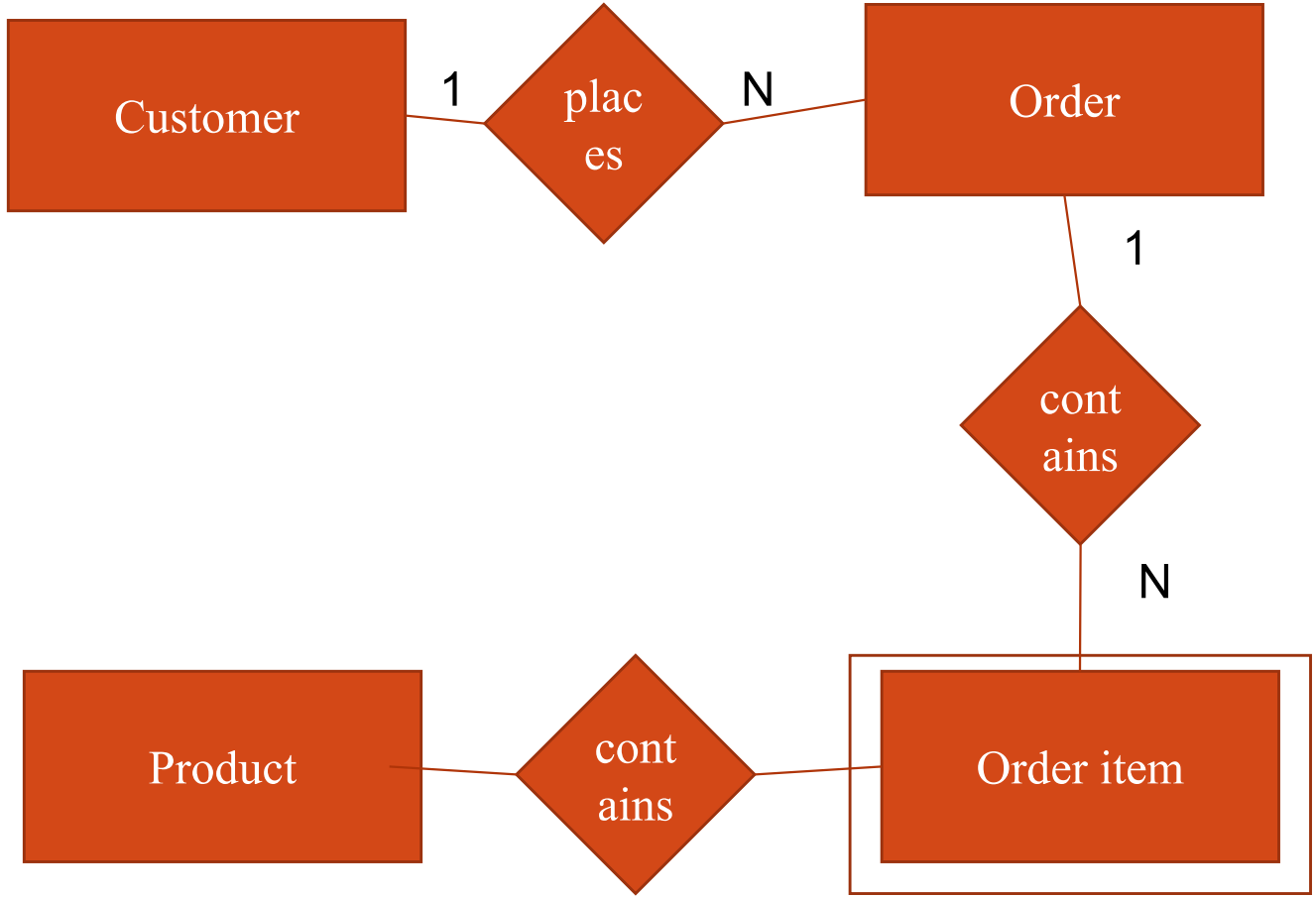
- Entities & Relations



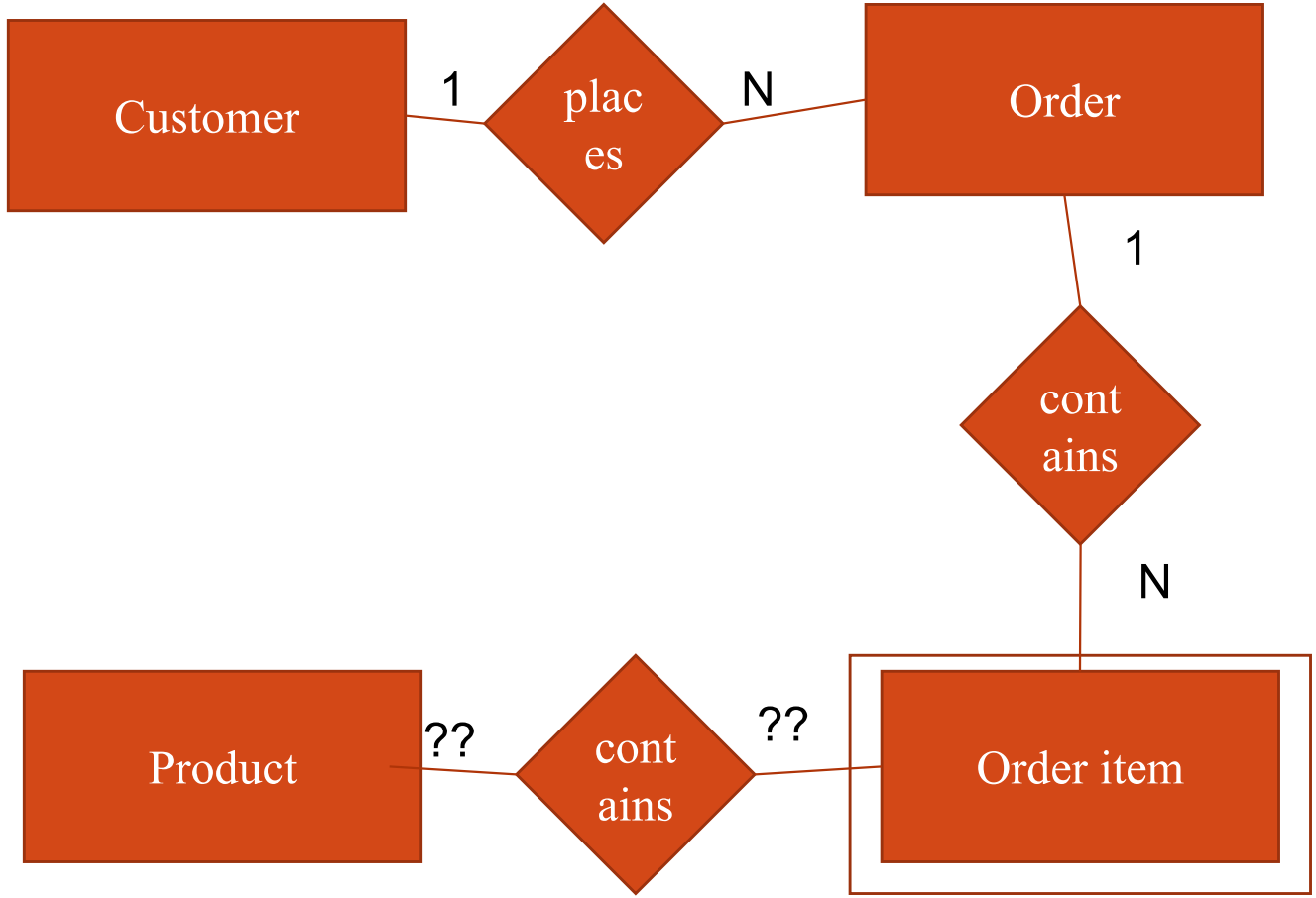
- Entities & Relations



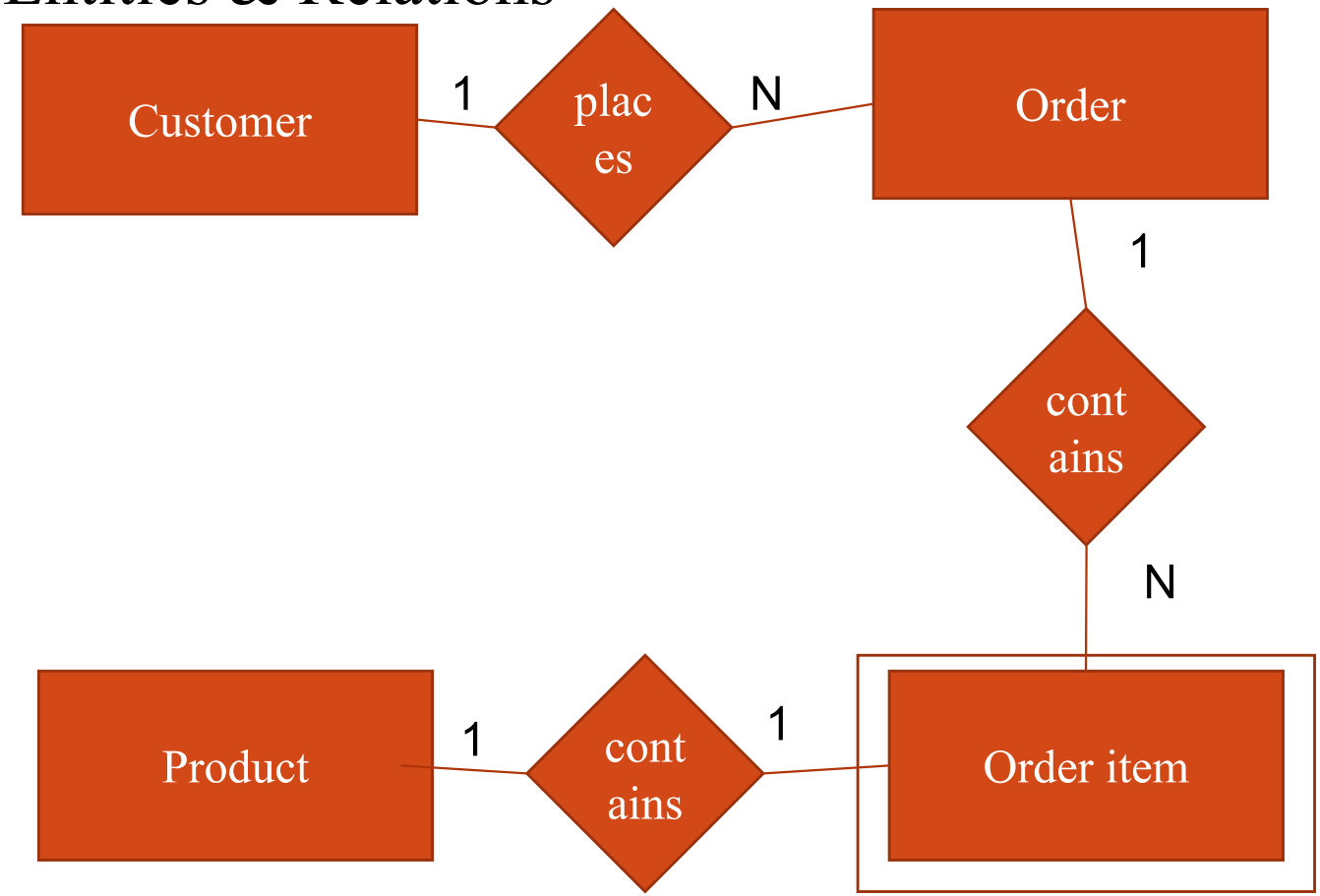
- Entities & Relations



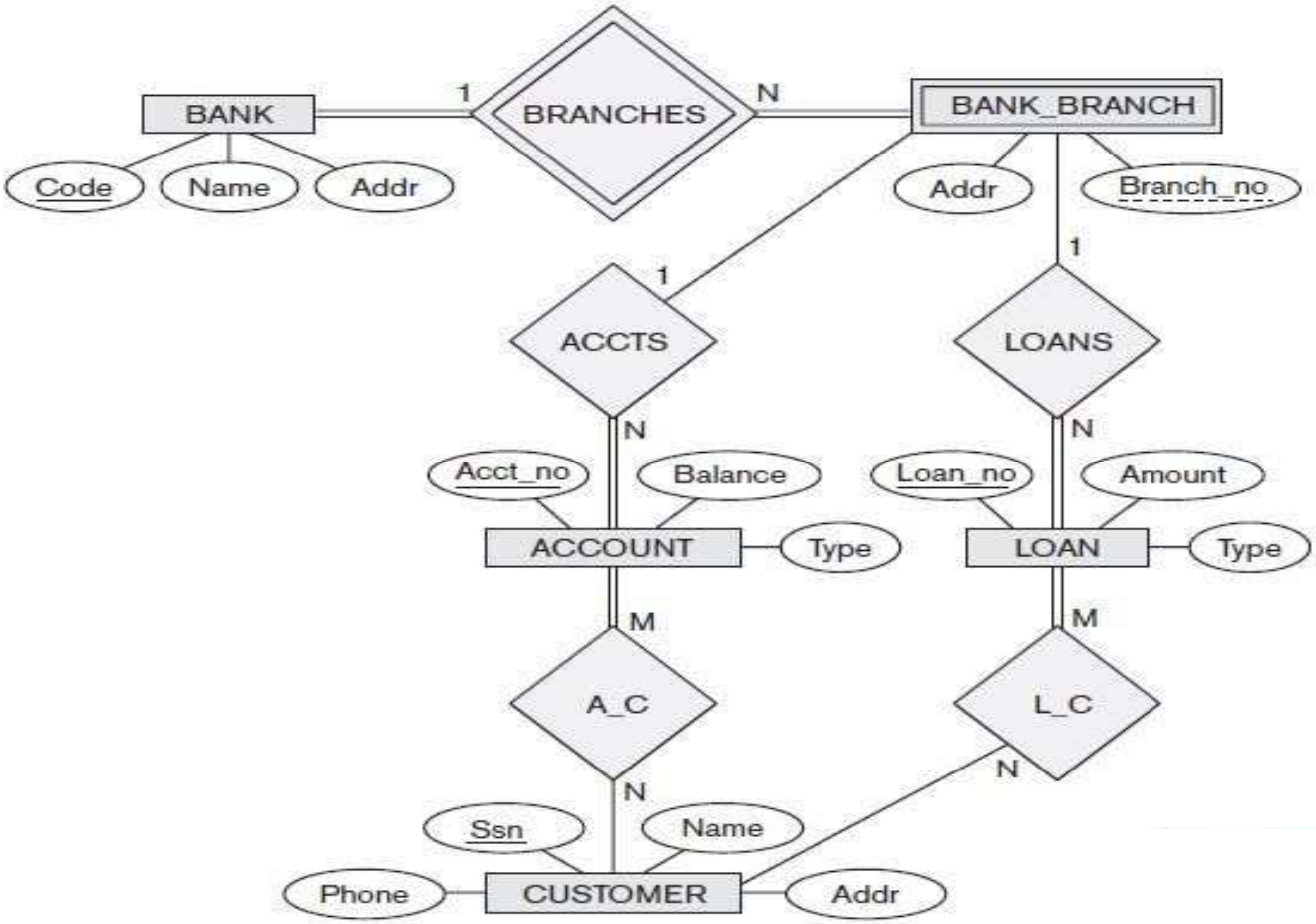
- Entities & Relations



- Entities & Relations



ER Diagram for Bank Database



ER DIAGRAM FOR LIBRARY DATABASE

ER DIAGRAM FOR LIBRARY DATABASE

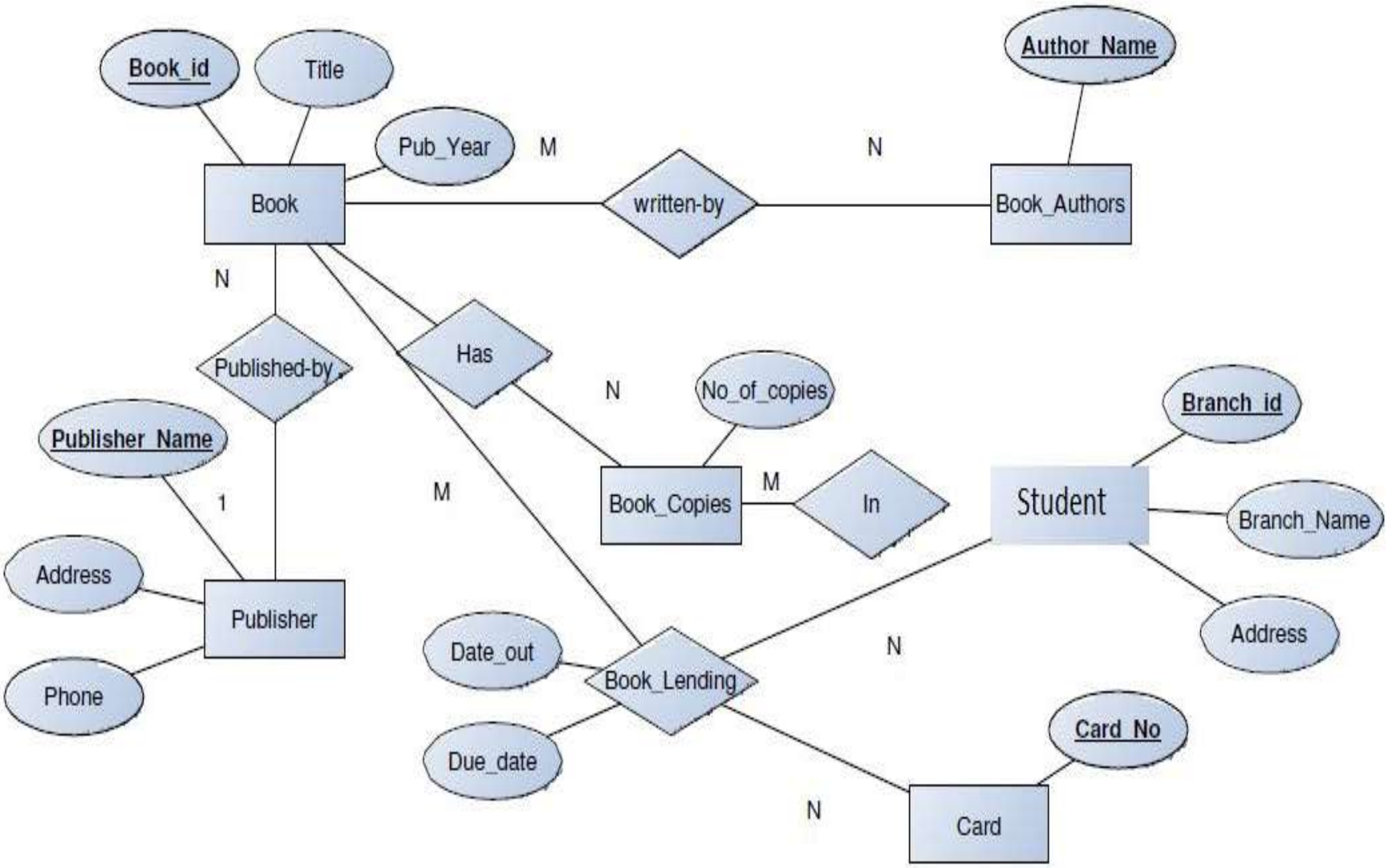
- List the entities
  - Books
  - Author
  - Publisher
  - Student
  - Card
  - Copies

ER DIAGRAM FOR LIBRARY DATABASE

- List the Relations
  - Book written by author
  - Book published by publisher
  - Book has copies
  - Book lending to student
  - Book lending on card

  - Books borrowed by student / book issued to student
  - Book given on card

Entity-Relationship Diagram for LIBRARY DATABASE



ER diagram for Train database

- Entities
  - Passengers
  - Train
  - Train Status
  - Stations
  - Routes

ER diagram for Train database

- Entities with attributes
  - Passengers(PNR, Pname, Age, Gender, reserve_Status)
  - Train (trainid, Tname,Ttype, Class)
  - Train Status(Status_id, wait_seat, Booked_seat, Avl_seat)
  - Stations(St_id, St_name)
  - Routes(Stop_no, A_time, D_time)

ER diagram for Train database

- Entities with attributes & key values
  - Passengers(**PNR**, Pname, Age, Gender, reserve_Status)
  - Train (**trainid**, Tname,Ttype, Class)
  - Train Status(**Status_id**, wait_seat, Booked_seat, Avl_seat)
  - Stations(**St_id**, St_name)
  - Routes(**Stop_no**, A_time, D_time)

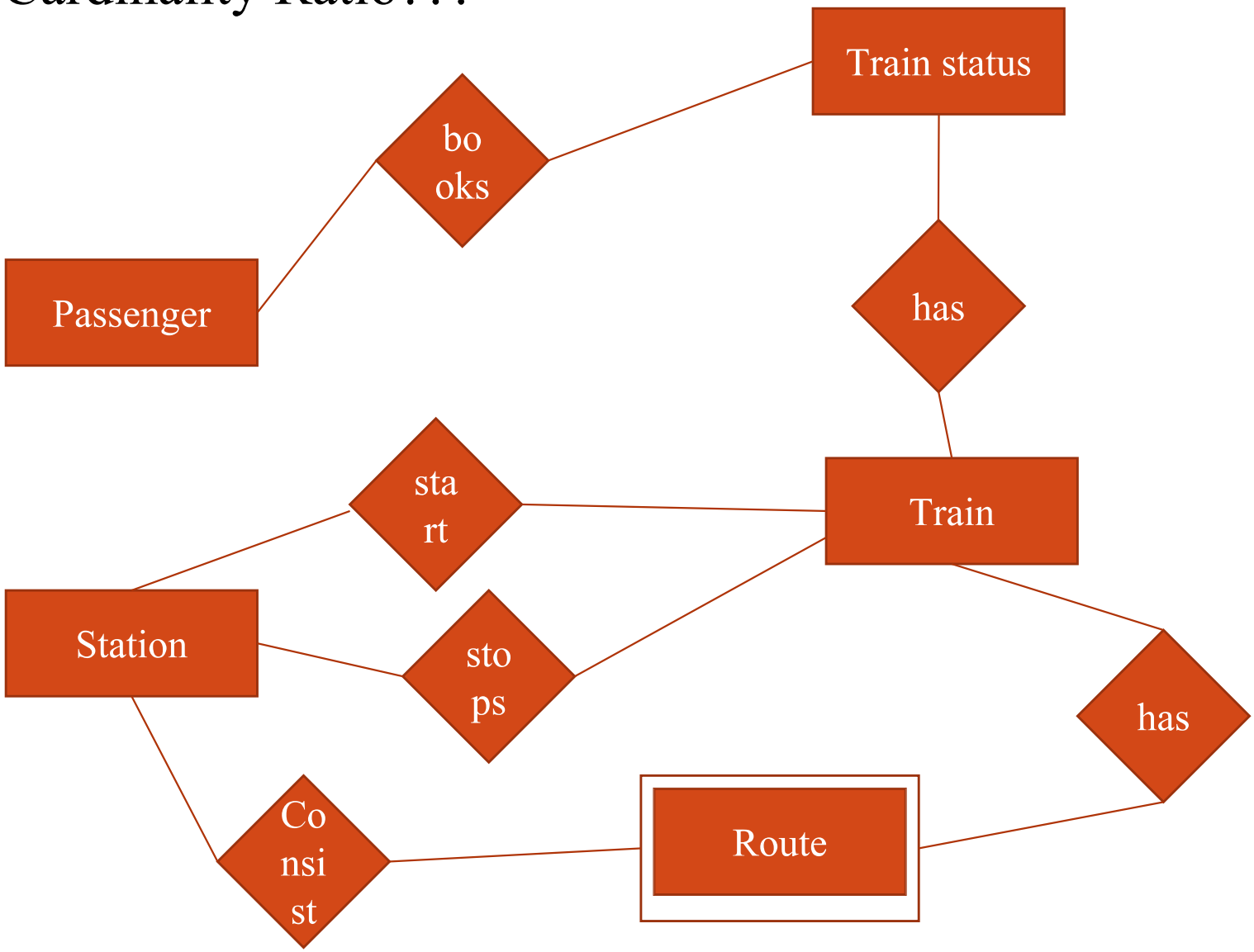
ER diagram for Train database

- Relations
  - Train has train status
  - Train starts from station
  - Train stops at station
  - Station consists of route (route is weak entity)
  - Route has train
  - Passenger books ????

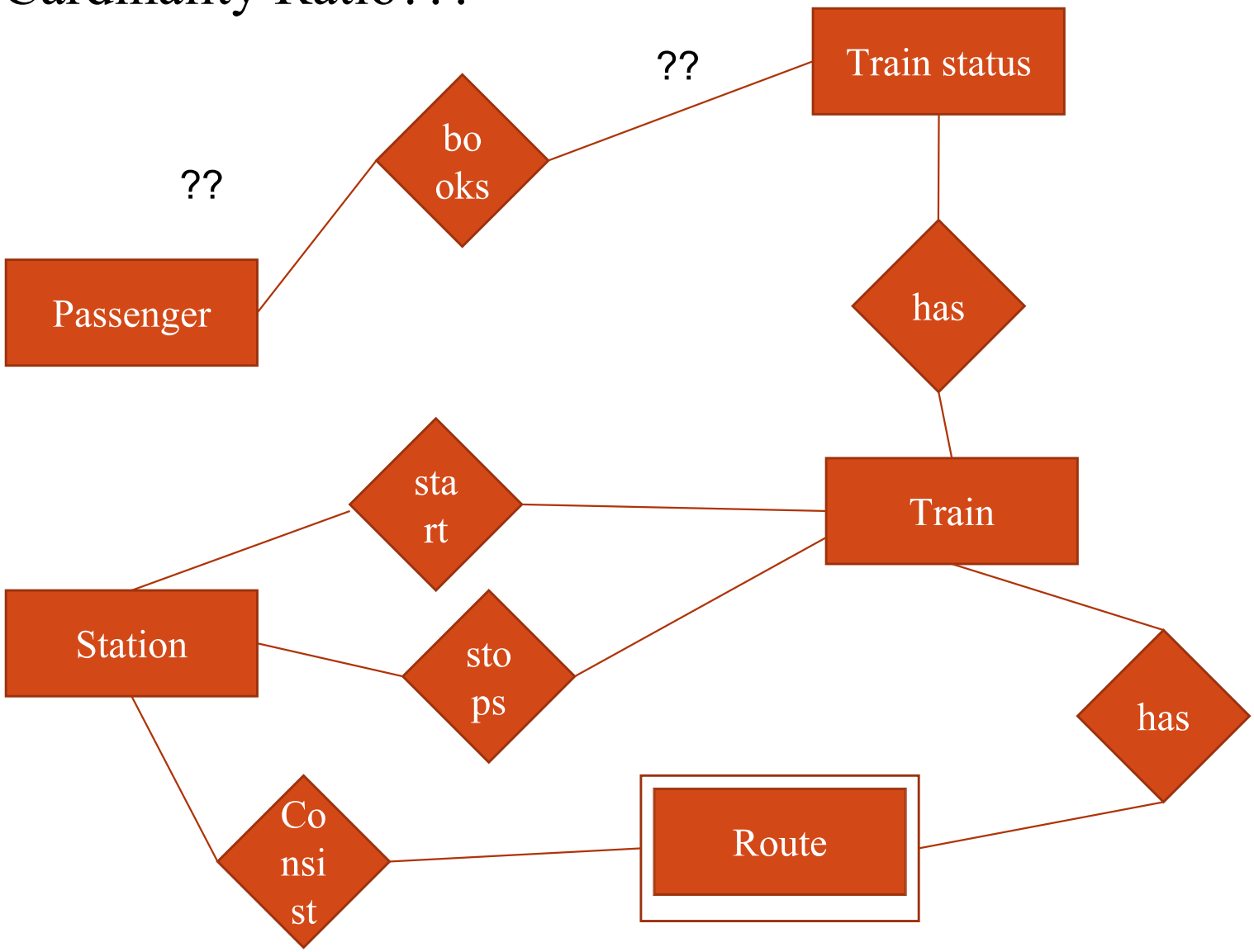
ER diagram for Train database

- Relations
  - Train has train status
  - Train starts from station
  - Train stops at station
  - Station consists of route (route is weak entity)
  - Route has train
  - Passenger books train status

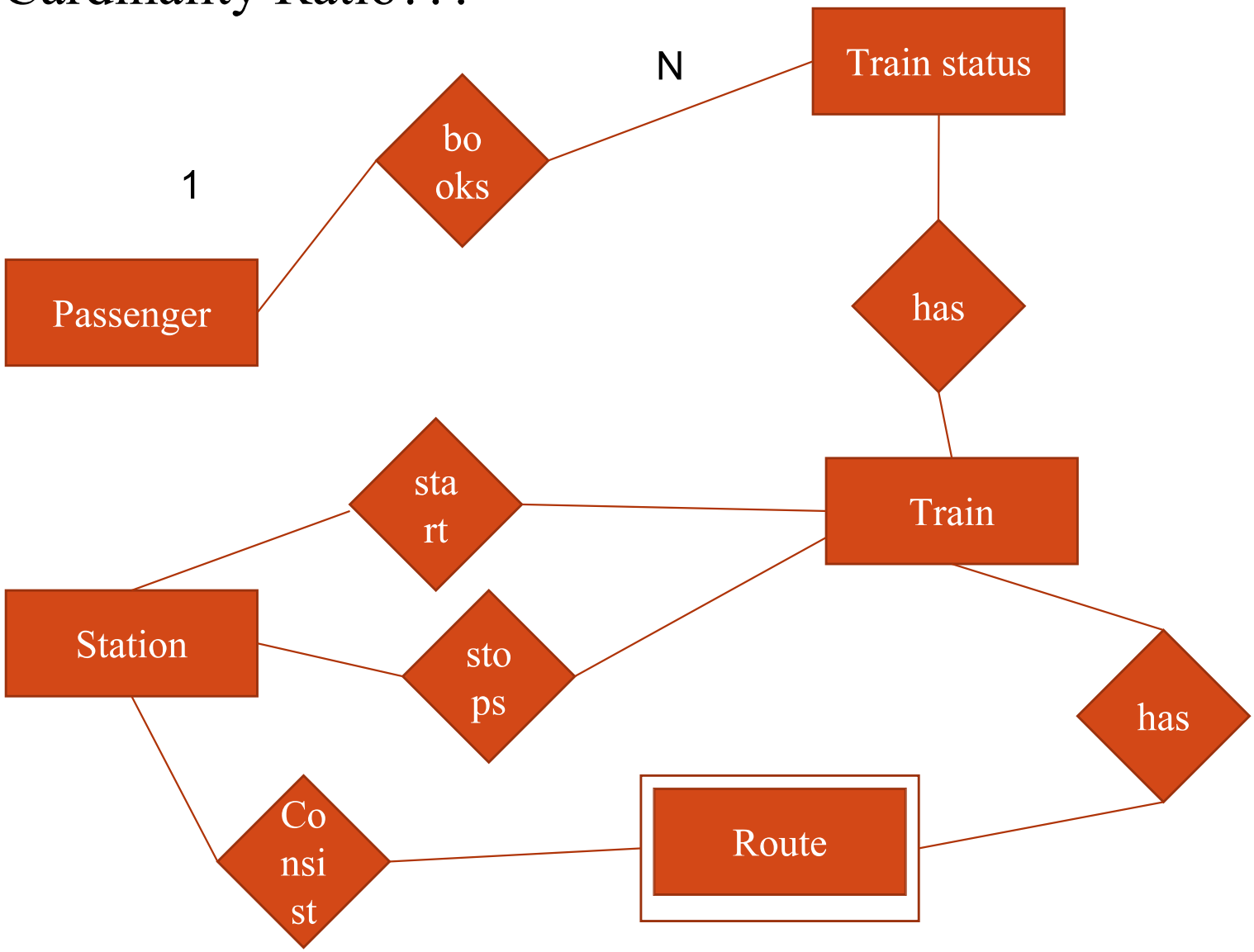
- Cardinality Ratio???



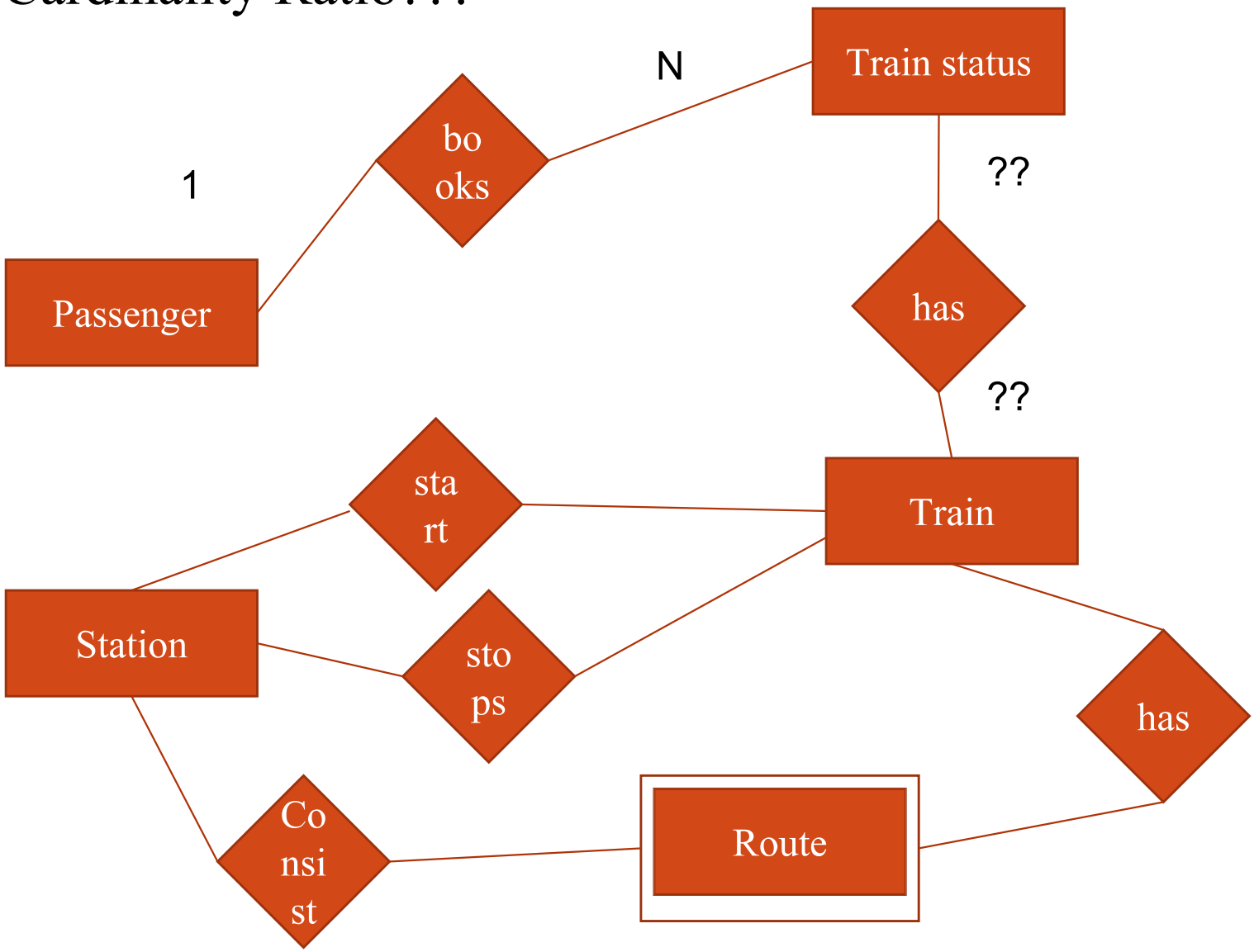
- Cardinality Ratio???



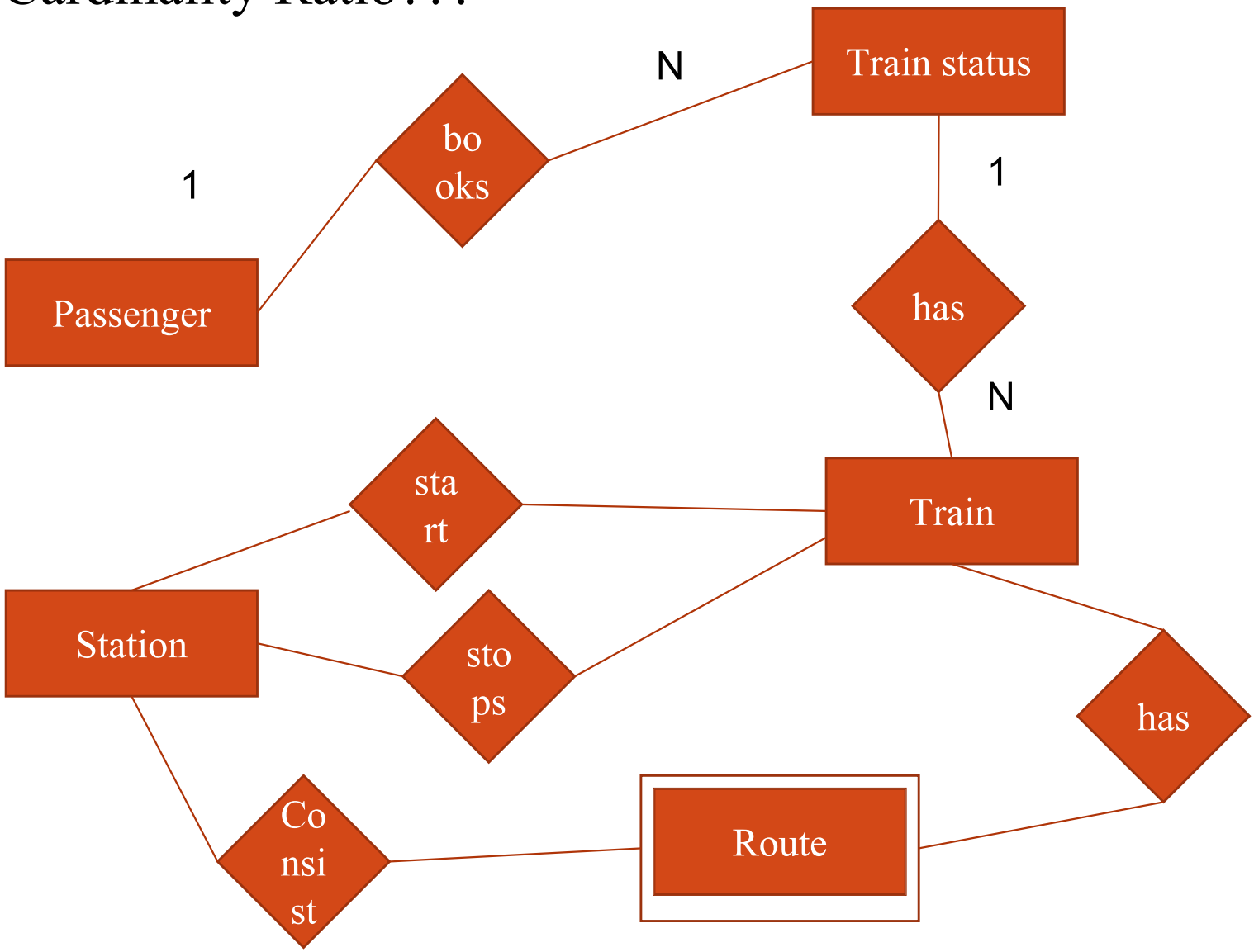
- Cardinality Ratio???



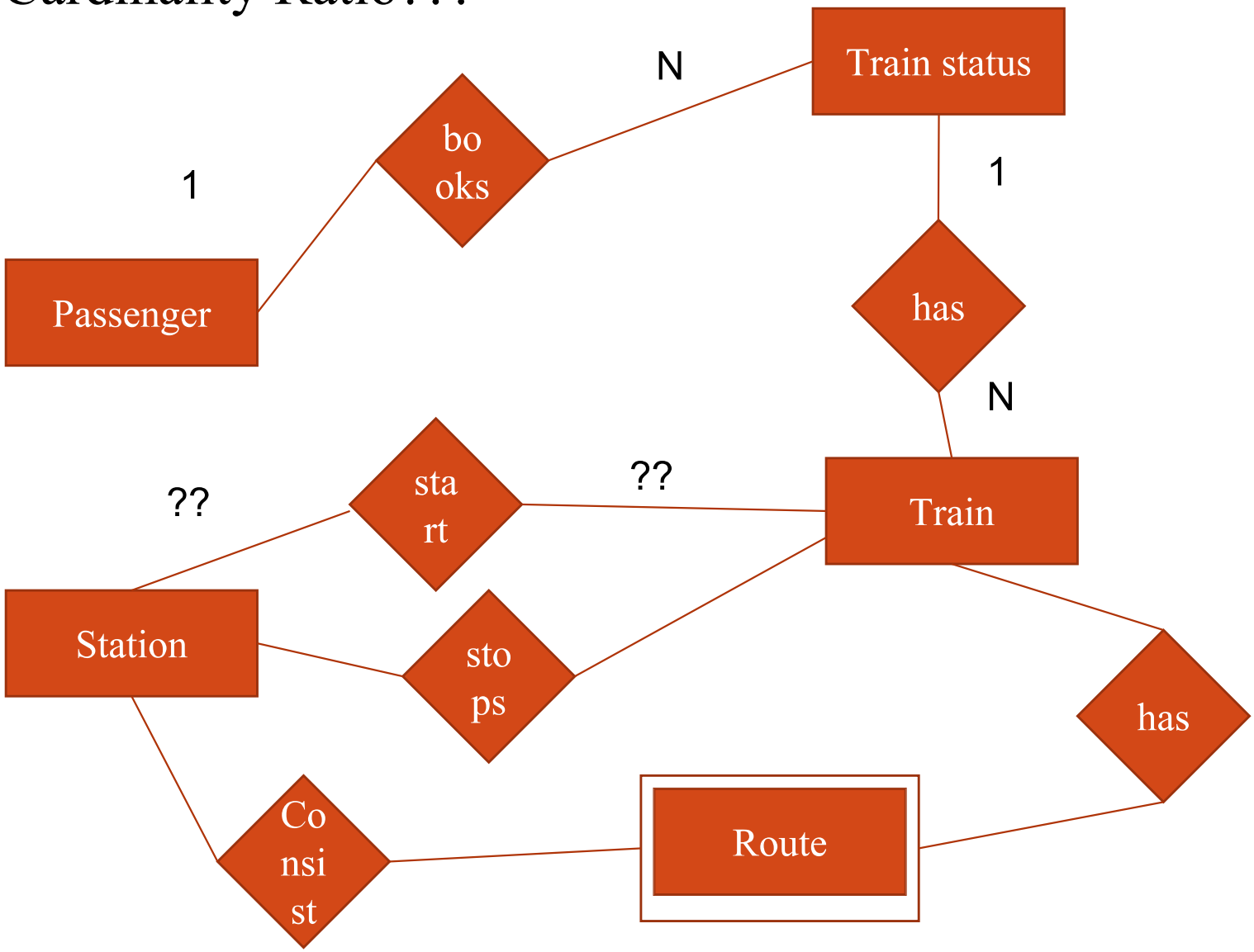
- Cardinality Ratio???



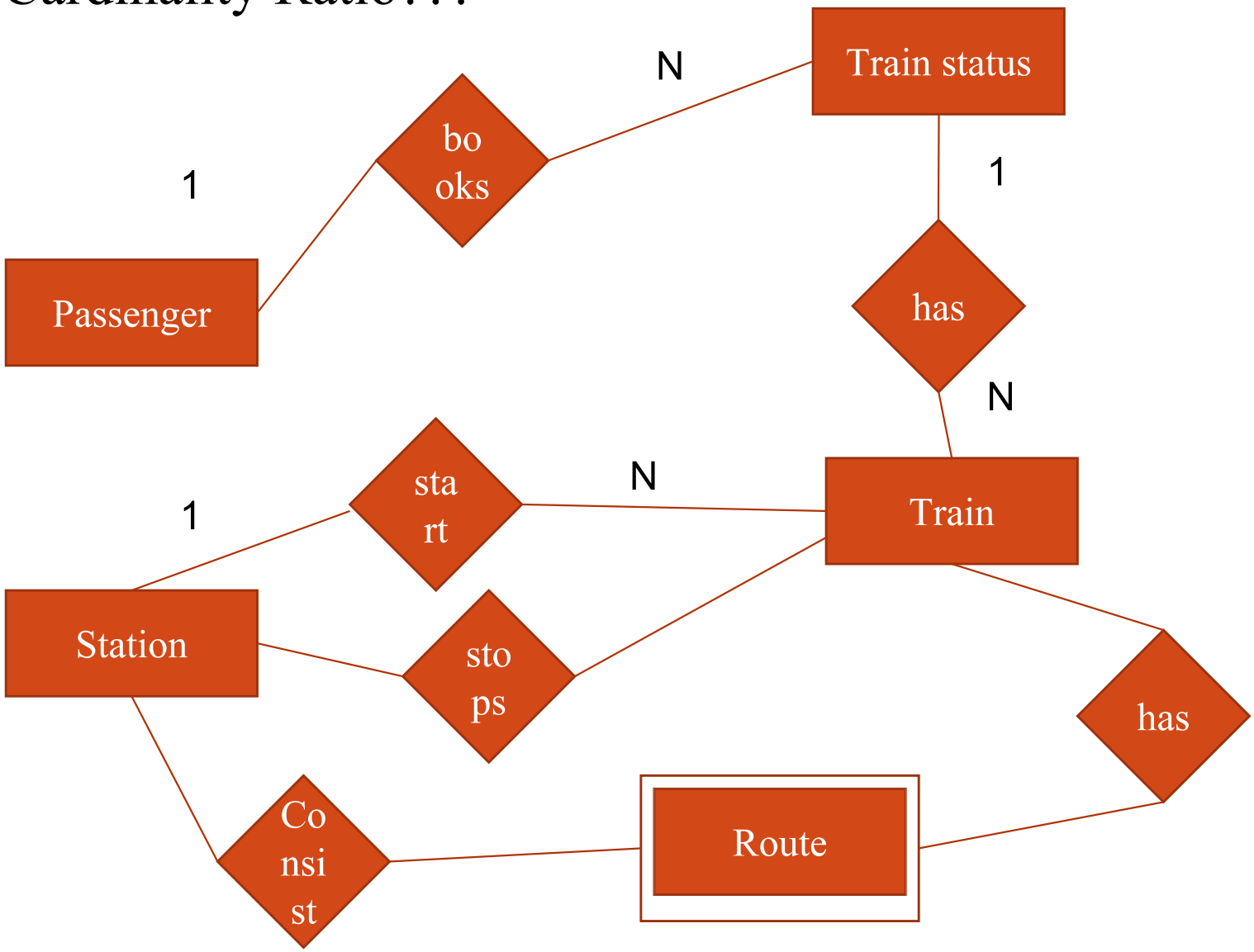
- Cardinality Ratio???



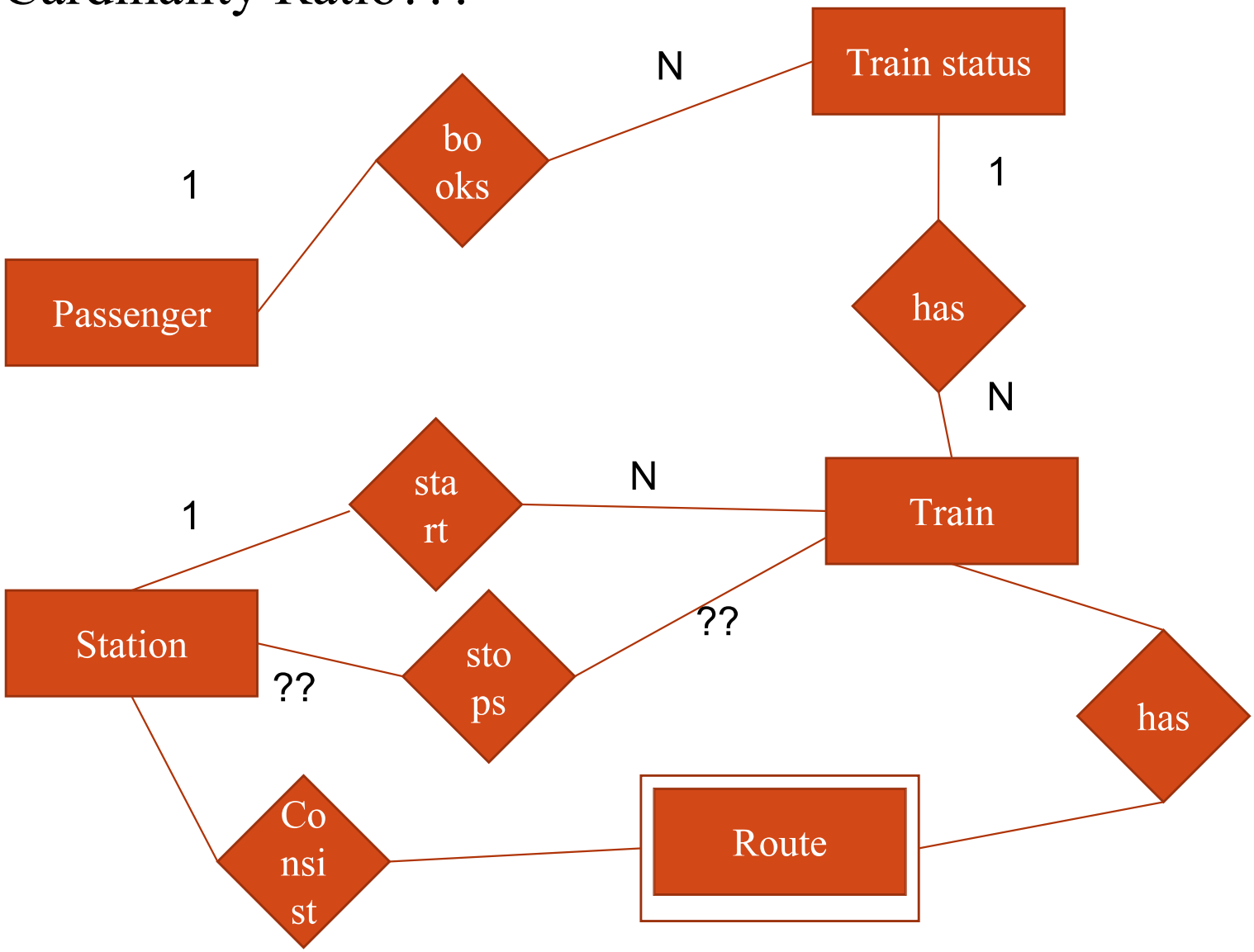
- Cardinality Ratio???



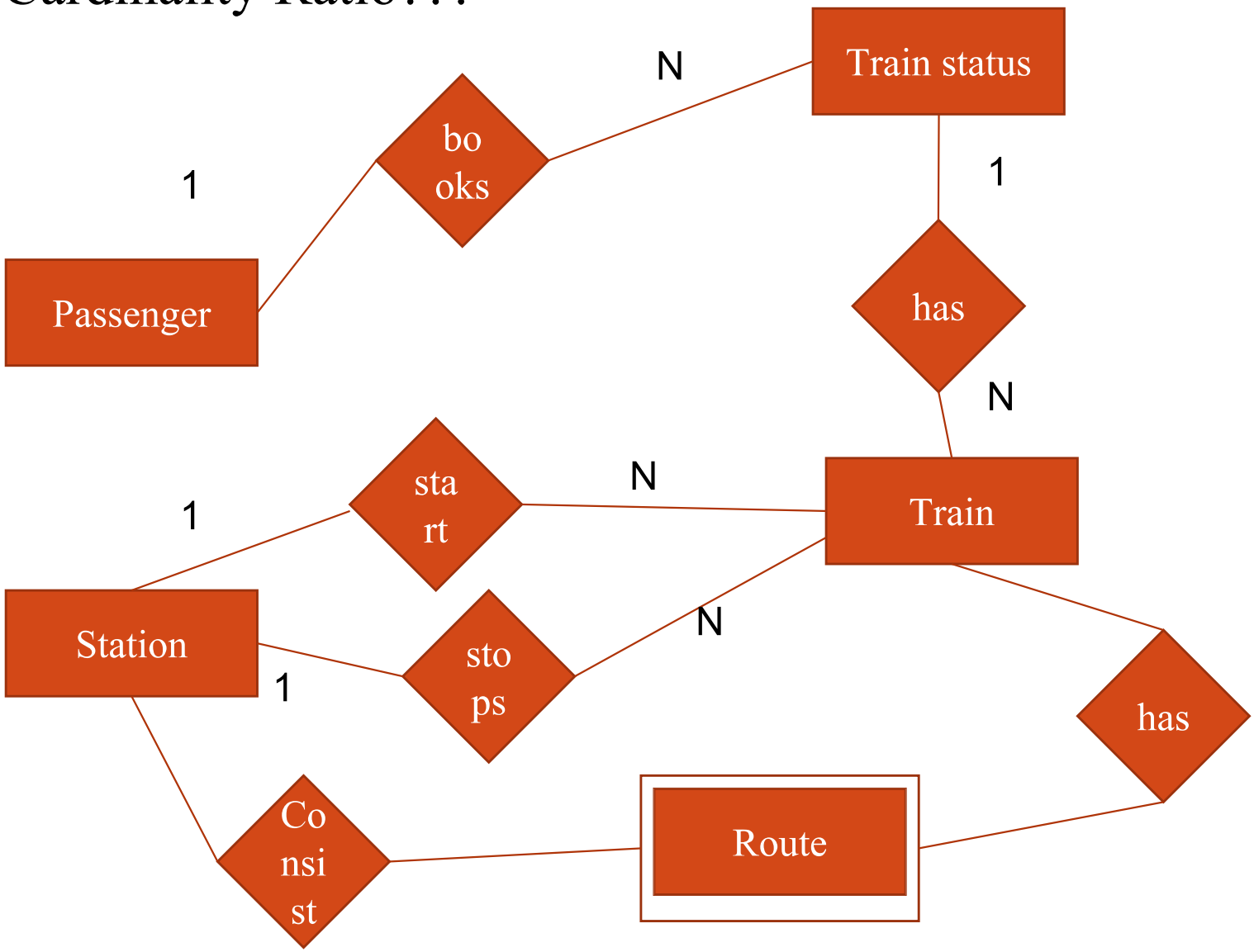
- Cardinality Ratio???



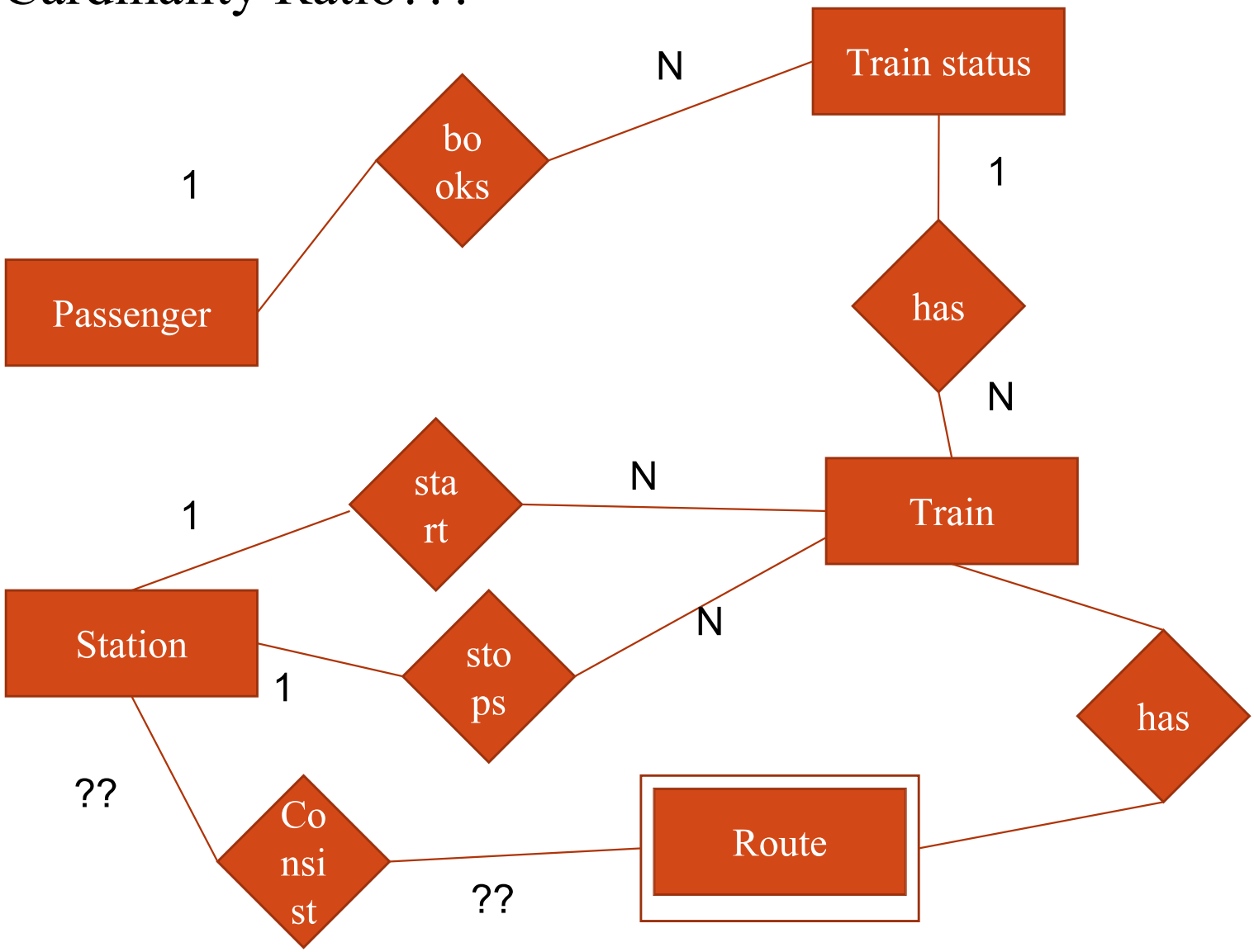
- Cardinality Ratio???



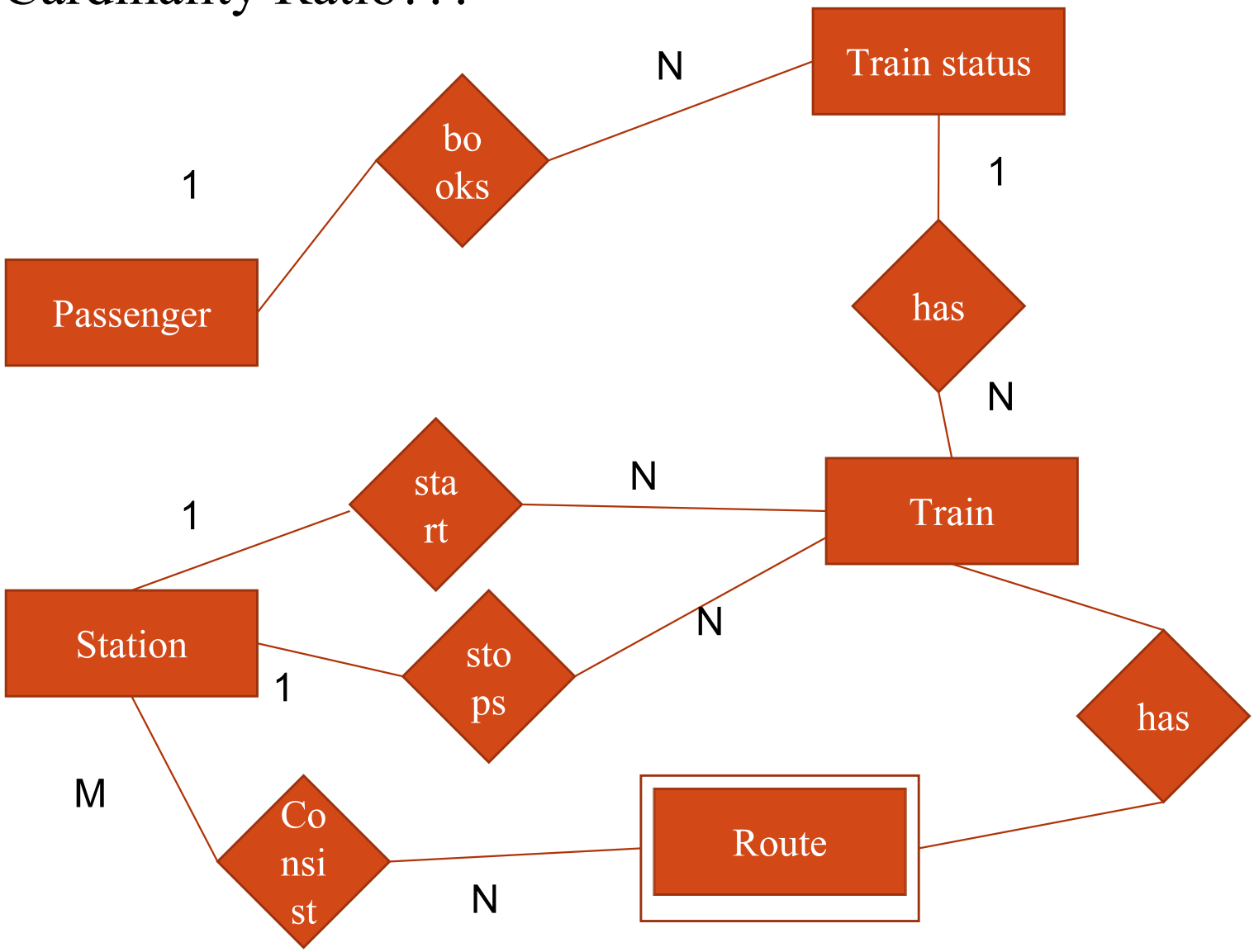
- Cardinality Ratio???



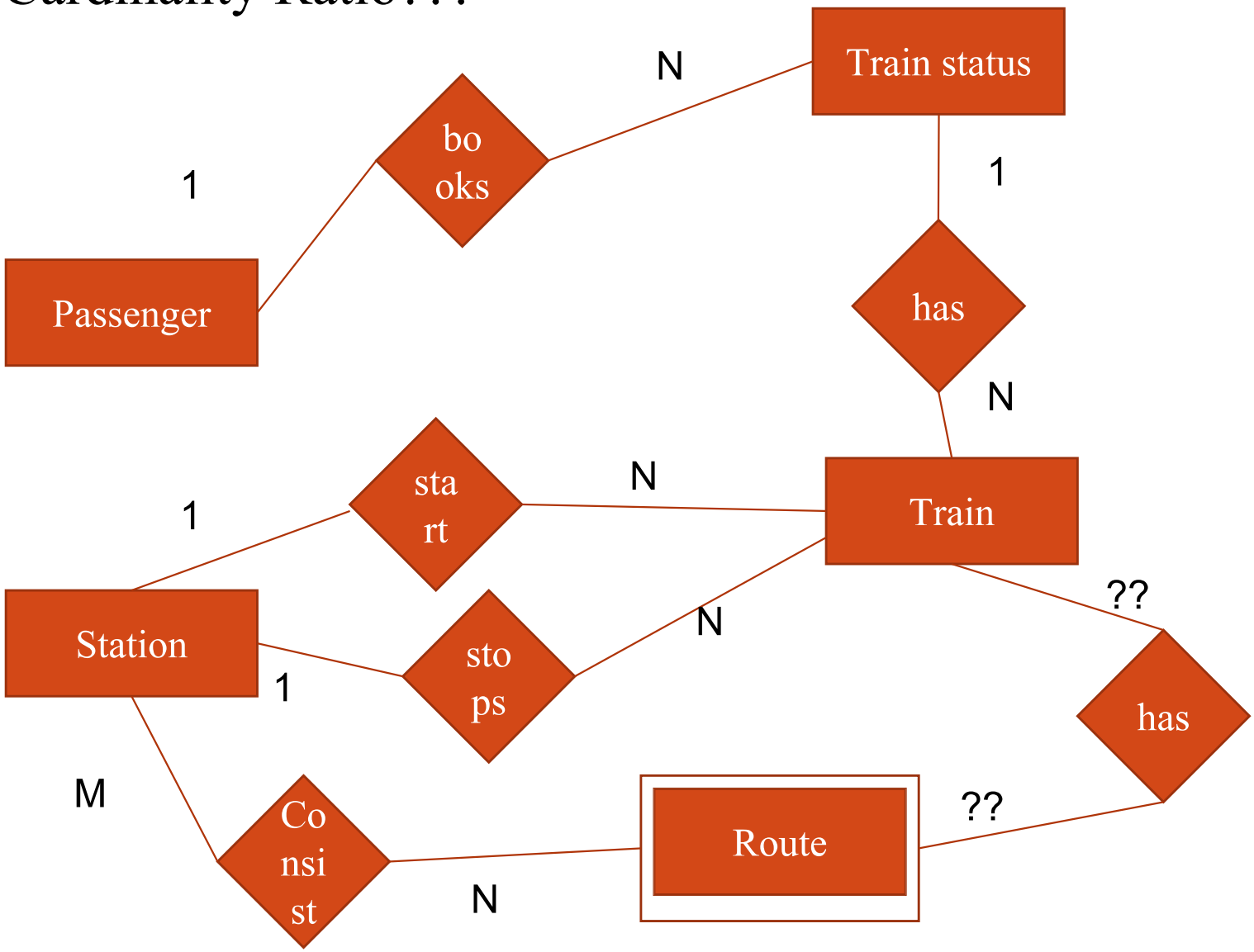
- Cardinality Ratio???



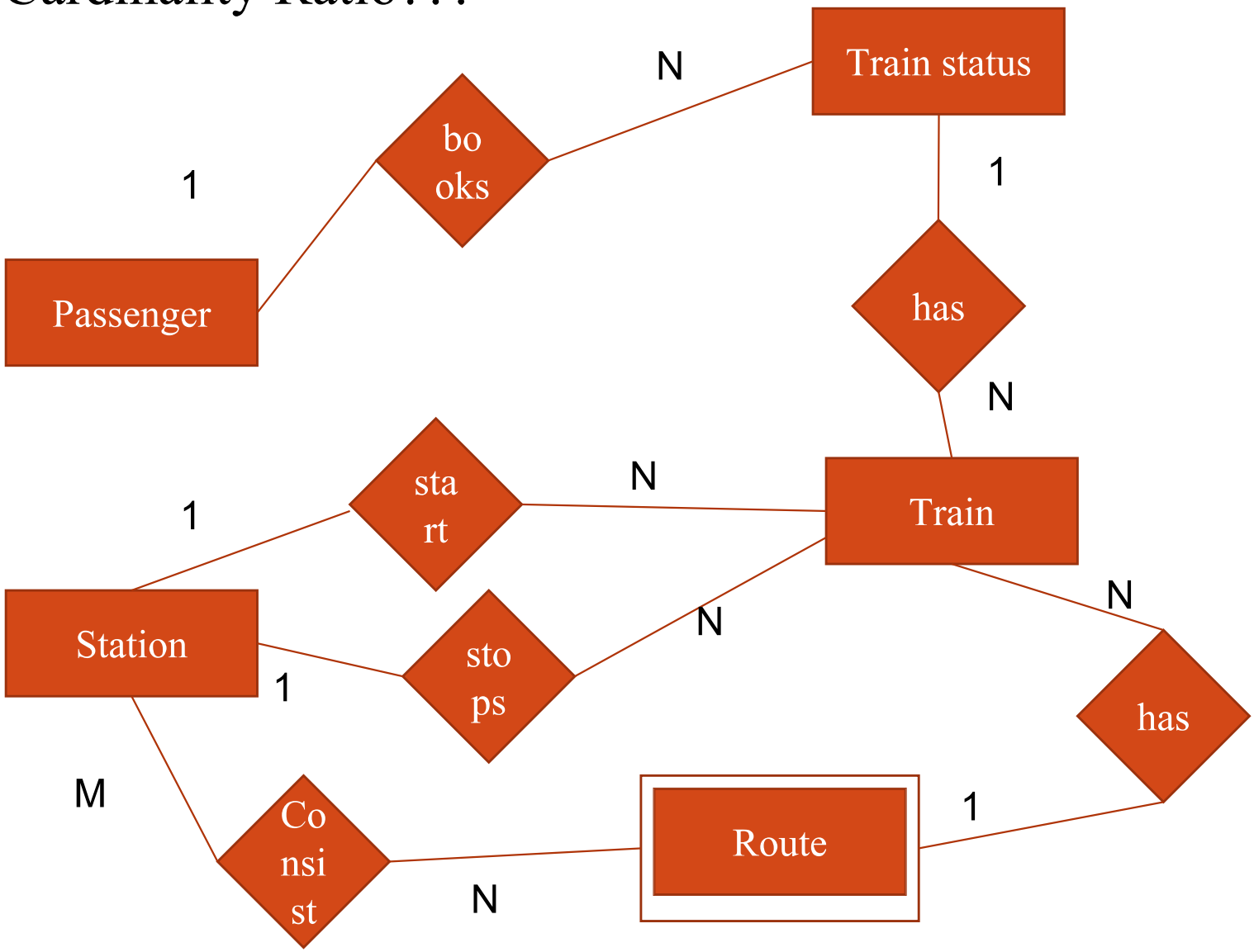
- Cardinality Ratio???



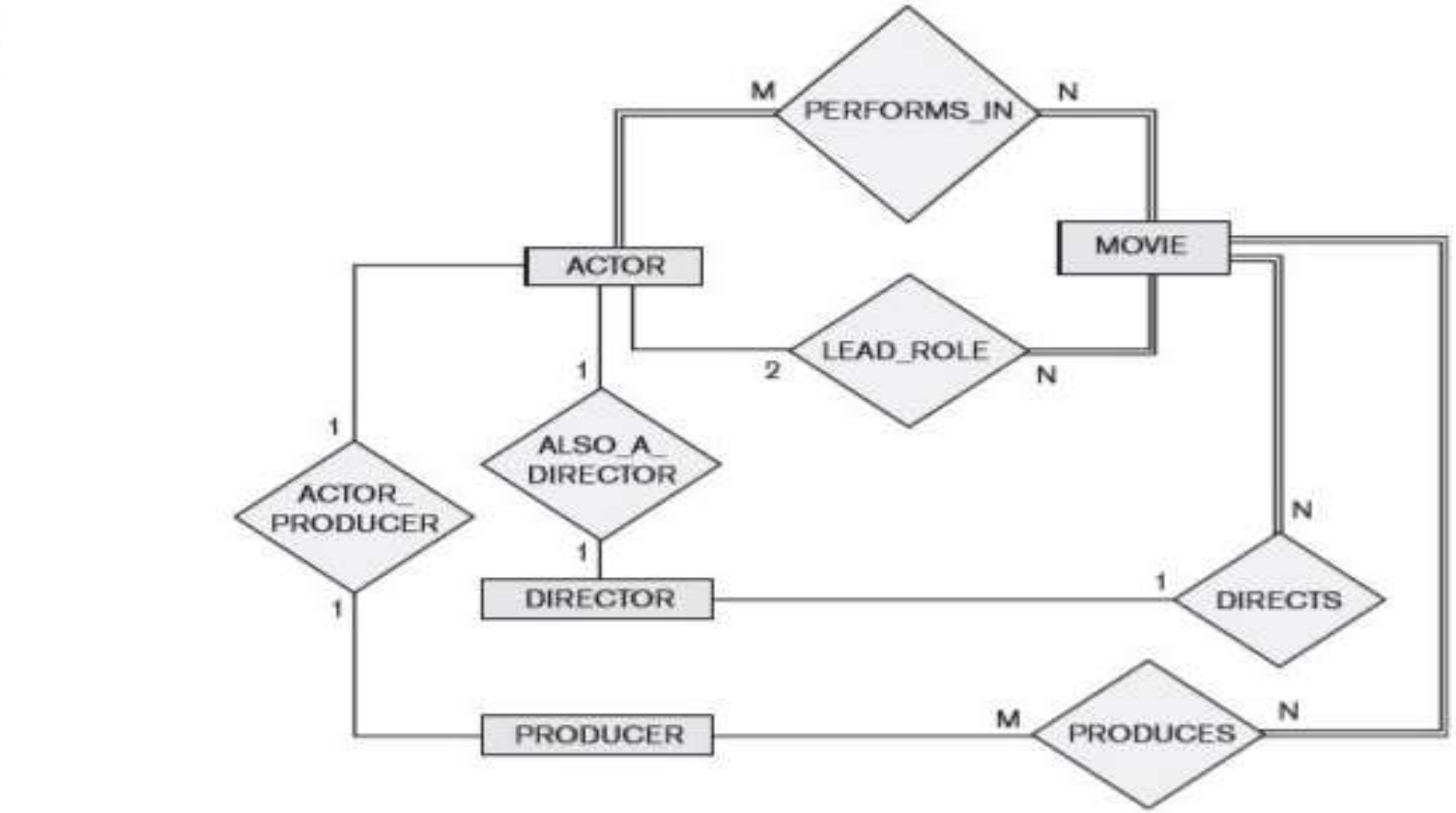
- Cardinality Ratio???



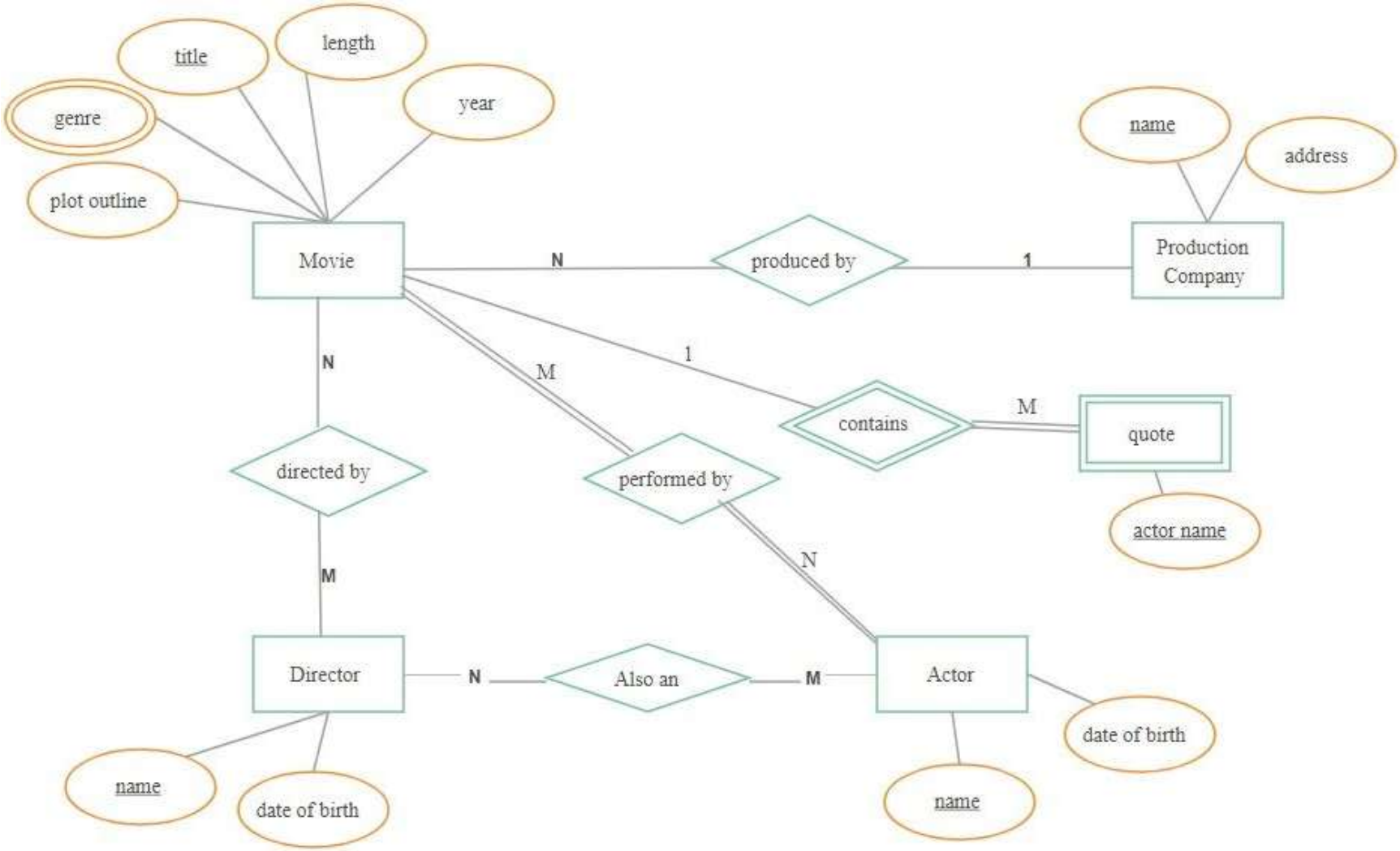
- Cardinality Ratio???



ER diagram for Movie Database

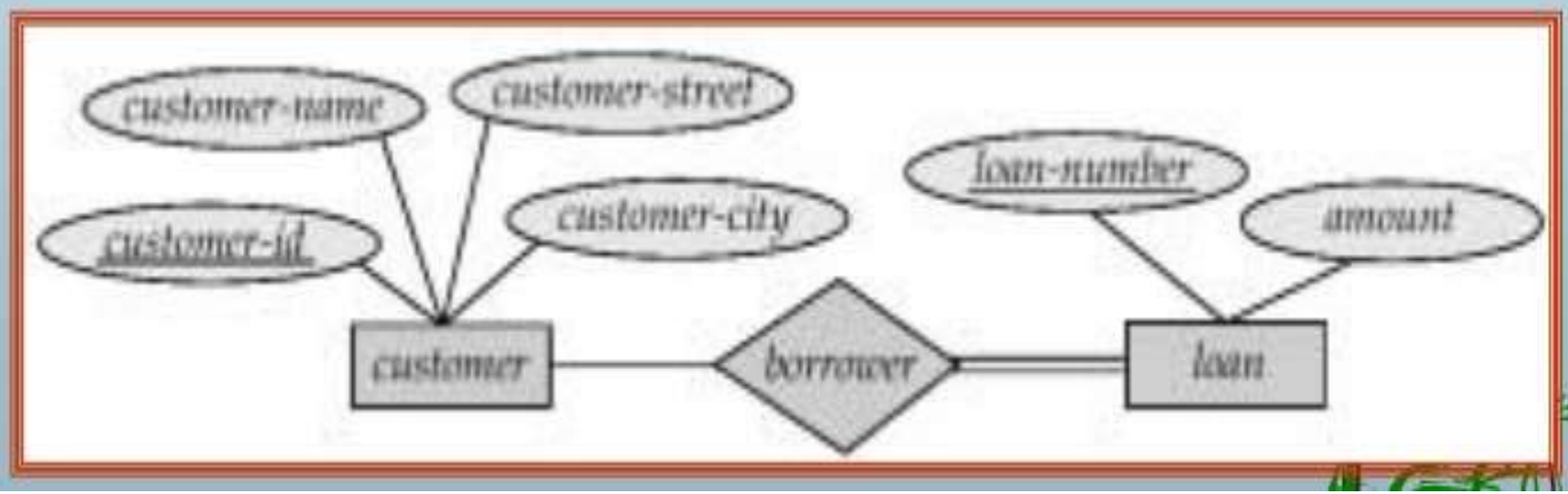


ER diagram for Movie Database



Participation of an Entity Set in a Relationship Set

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
  - E.g. participation of *loan* in *borrower* is total
    - every loan must have a customer associated to it via borrower
- Partial participation: some entities may not participate in any relationship in the relationship set
  - E.g. participation of *customer* in *borrower* is partial



Most Frequently asked Questions

1.Explain the various notations in ER modeling.

2. List different types of attributes with example.

3. Explain the various types of relationship with an example.

4. Explain the main phases of database design with a diagram.

5. ER modeling for a given scenario.

6. Define entity, entity type and entity set.

7. What is an weak entity with an example explain.

8. Explain total and partial participation.