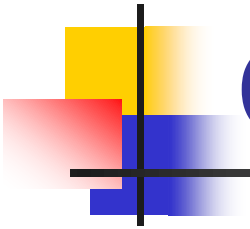
# Database Management Systems

Shamim Ahmad

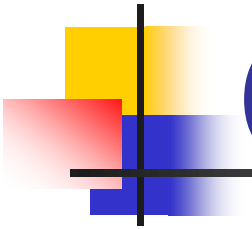
# Example ER Design



# Constructing an ER model

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

# Constructing an ER model (cont)

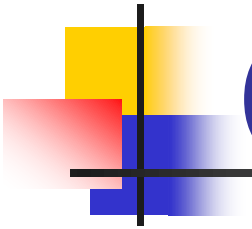


1. **Identify entities** - list all potential entity types. These are the object of interest in the system. It is better to put **too many** entities in at this stage and them **discard** them later if necessary.

# Constructing an ER model (cont)

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.
   - The library is the system, thus should not be an entity type.

# Constructing an ER model (cont)



**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?
  - if so keep them as attributes and cross them off the entity list.
- Do not have **attributes of one entity** as attributes of another entity!

Example Scenario:

# University Course Registration System

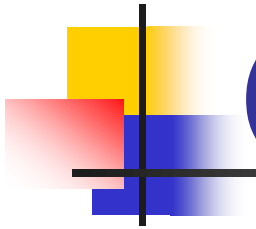
Students register for courses taught by instructors in departments.

# Step 1: Identify Possible Entities

From the description we may initially think these are entities:

- Student
- Course
- Instructor
- Department
- Address
- Phone
- Grade
- Semester

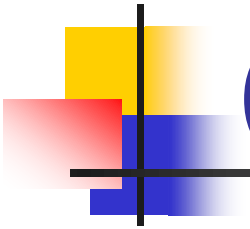
# Constructing an ER model (cont)



**4. Mark the primary keys.**

- Which attributes **uniquely identify** instances of that entity type?
- This may not be possible for some **weak entities.**

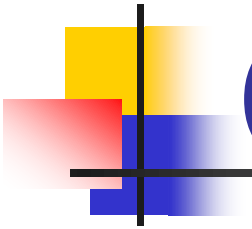
# Constructing an ER model (cont)



**5. Define the relationships**

- Examine each entity type to see its relationship to the others.

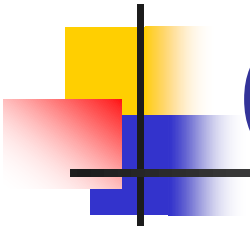
# Constructing an ER model (cont)



6. **Describe the cardinality and optionality of the relationships**

   - Examine the **constraints** between participating entities.

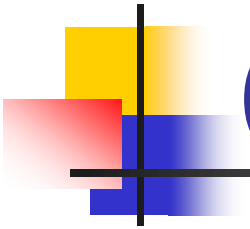
# Constructing an ER model (cont)



**7. Remove redundant relationships**

Examine the ER model for redundant relationships.

# Constructing an ER model (cont)



- ER modelling is an iterative process
- So draw **several** versions, **refining** each one until you are happy with it
- Note that there is **no one right** answer to the problem, but some **solutions are better than others**!

# Problem (story) : *National* Bus Company

- The *DeSh Travells* owns a number of busses. Each bus is allocated to a particular route, although some routes may have several busses. Each route passes through a number of towns.

# Problem (story) : *National* Bus Company

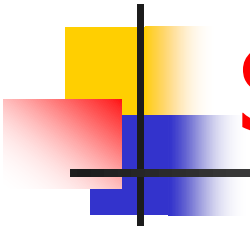
- One or more drivers are allocated to each stage of a route, which corresponds to a journey through some or all of the towns on a route.

# Problem (story) : *National* Bus Company

- Some of the towns have a garage where busses are kept and each of the busses are identified by the registration number and can carry different numbers of passengers, since the vehicles vary in size and can be single or double-decked.

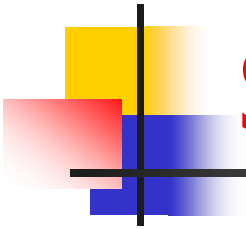
# Problem (story) : *National* Bus Company

- Each route is identified by a route number and information is available on the average number of passengers carried per day for each route. Drivers have an employee number, name, address, and sometimes a telephone number.



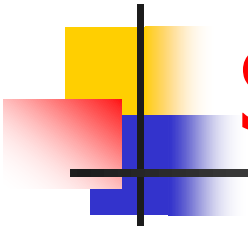
# Solution: Entities

- **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.



# Solution: Relationships

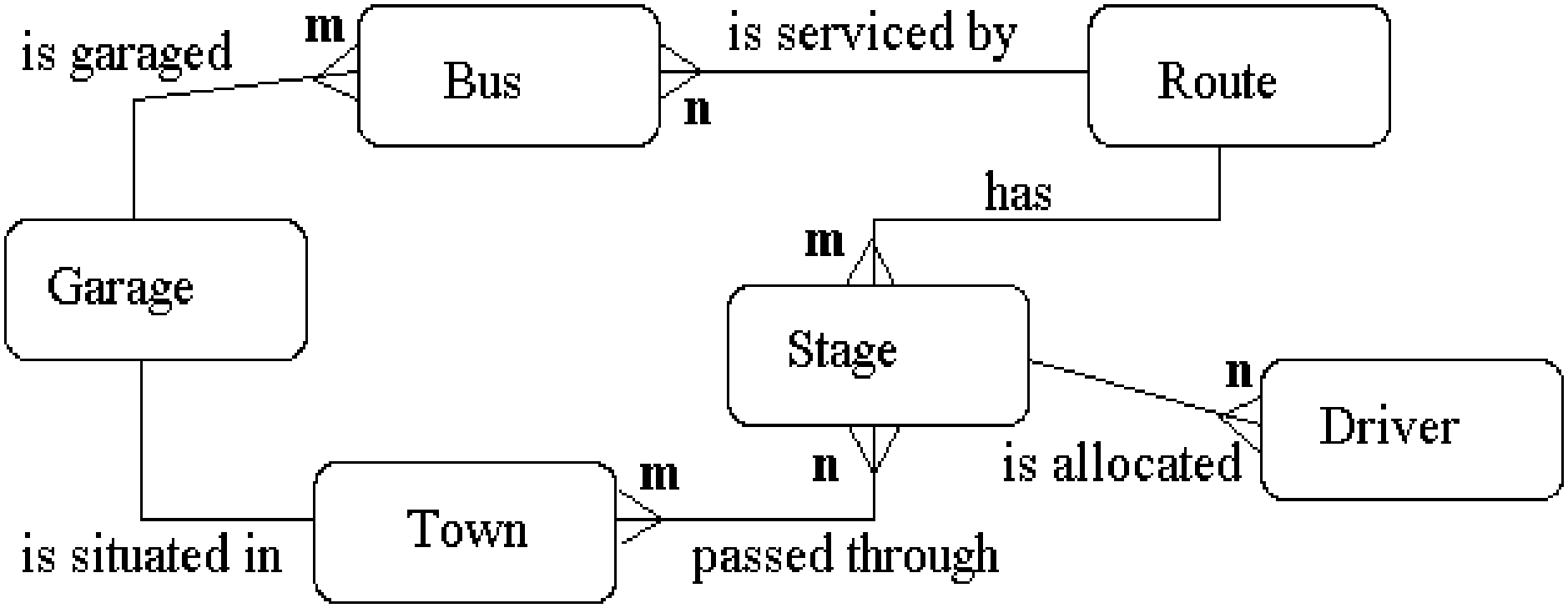
- A bus is allocated to a route and a route may have several buses.
  - Bus-route (m:1) is serviced by
- A route comprises of one or more stages.
  - route-stage (1:m) comprises
- One or more drivers are allocated to each stage.
  - driver-stage (m:1) is allocated
- A stage passes through some or all of the towns on a route.
  - stage-town (m:n) passes-through

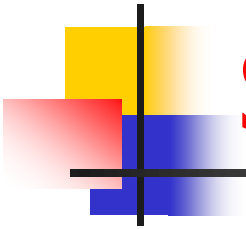


# Solution: Relationships (cont)

- A route passes through some or all of the towns
  - route-town (m:n) passes-through
- Some of the towns have a garage
  - garage-town (1:1) is situated
- A garage keeps buses and each bus has one `home' garage
  - garage-bus (m:1) is garaged

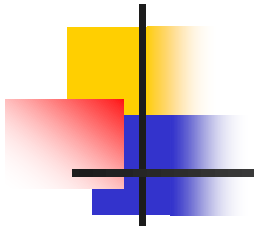
# Solution: Draw E-R Diagram



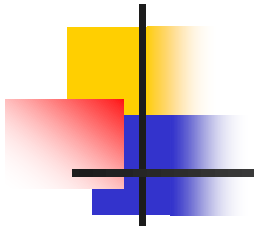


# Solution: Relations

- Bus (reg-no,make,size,deck,no-pass)
- Route (route-no,avg-pass)
- Driver (emp-no,name,address,tel-no)
- Town (name)
- Stage (stage-no)
- Garage (name,address)



# Banking System Design

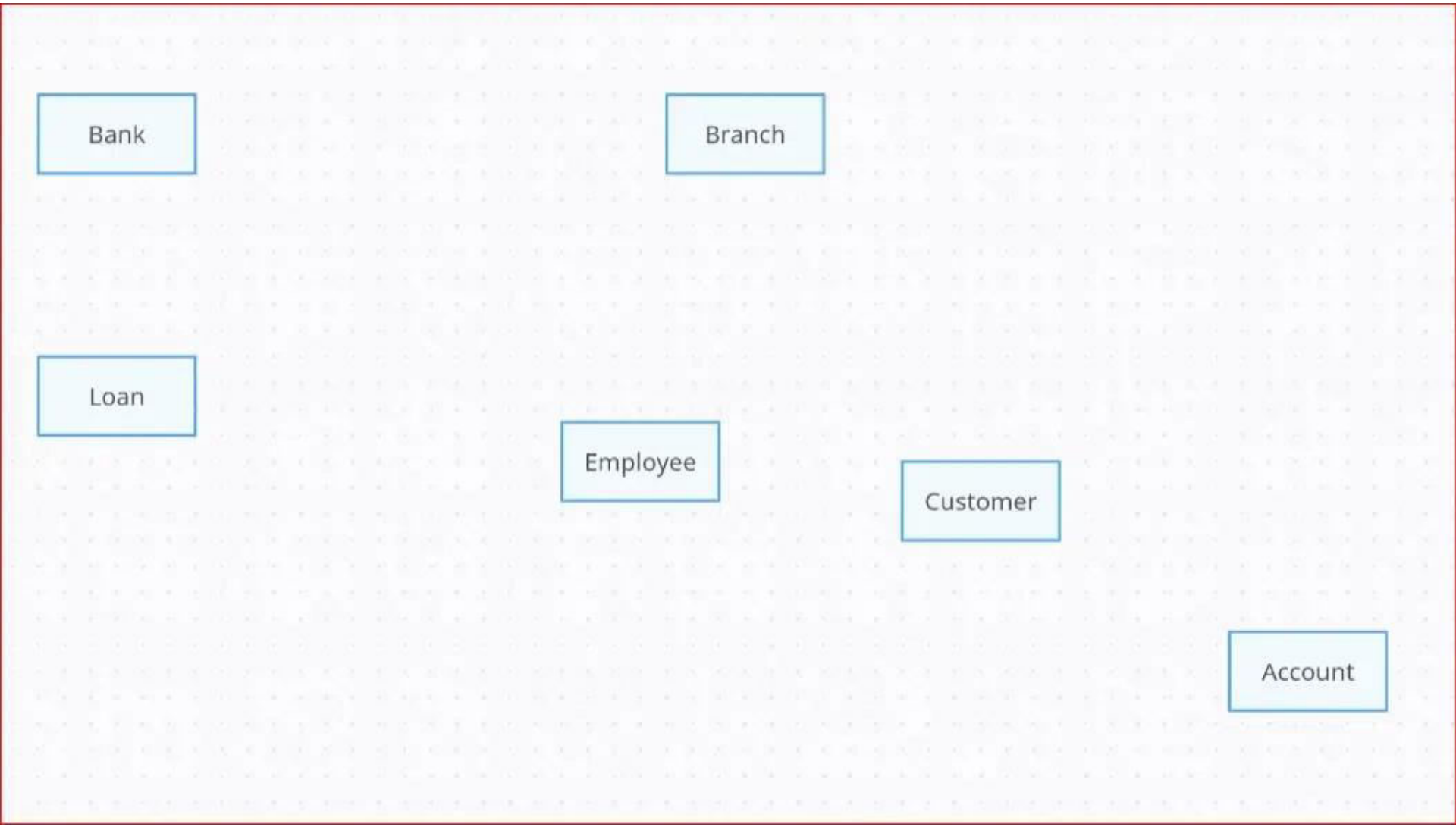


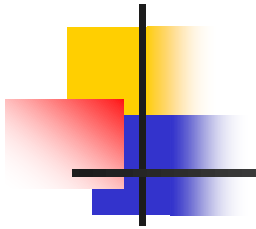
## Defining Entities

A thing in the real world with an independent existence. It is may be an object with physical existence (ex: house, person) or with a conceptual existence (ex: course, job). The are represented by rectangle.

Entities for Bank are:

Bank, Branch, Employee, customer, loan, account.


Bank
Branch
Loan
Employee
Customer
Account

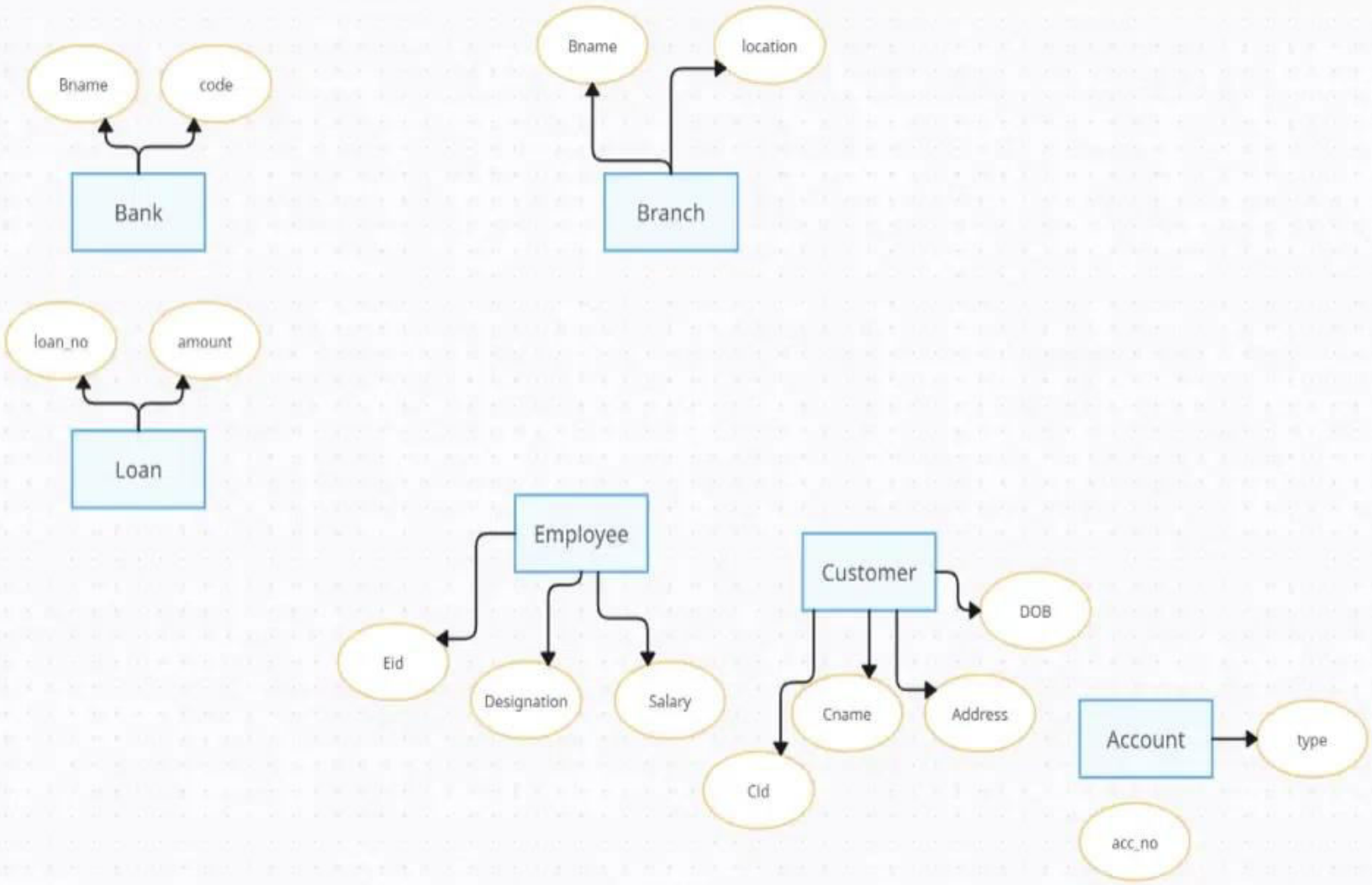


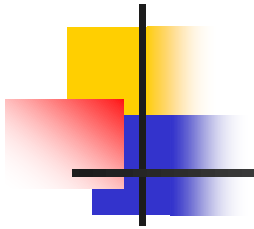
**Adding Attributes**

Attributes are the kind of properties that describe the entities. They are represented by **ovals**.

**Attributes for Bank are:**

- For **Bank Entity** the Attributes are **Bname, code.**
- For **Branch Entity** the Attributes are **Blocation, Bname.**
- For **Employee Entity** the Attributes are **Eid, Designation, salary.**
- For **Customer Entity** the Attributes are **Cid, Cname, Address, DOB.**
- For **Loan Entity** the Attributes are **Loan_no, amount, rate.**
- For **Account Entity** the Attributes are **acc_no, type.**


Bname
code
Bank
Bname
location
Branch
loan_no
amount
Loan
Employee
Eid
Designation
Salary
Customer
DOB
Cname
Address
Cid
Account
type
acc_no

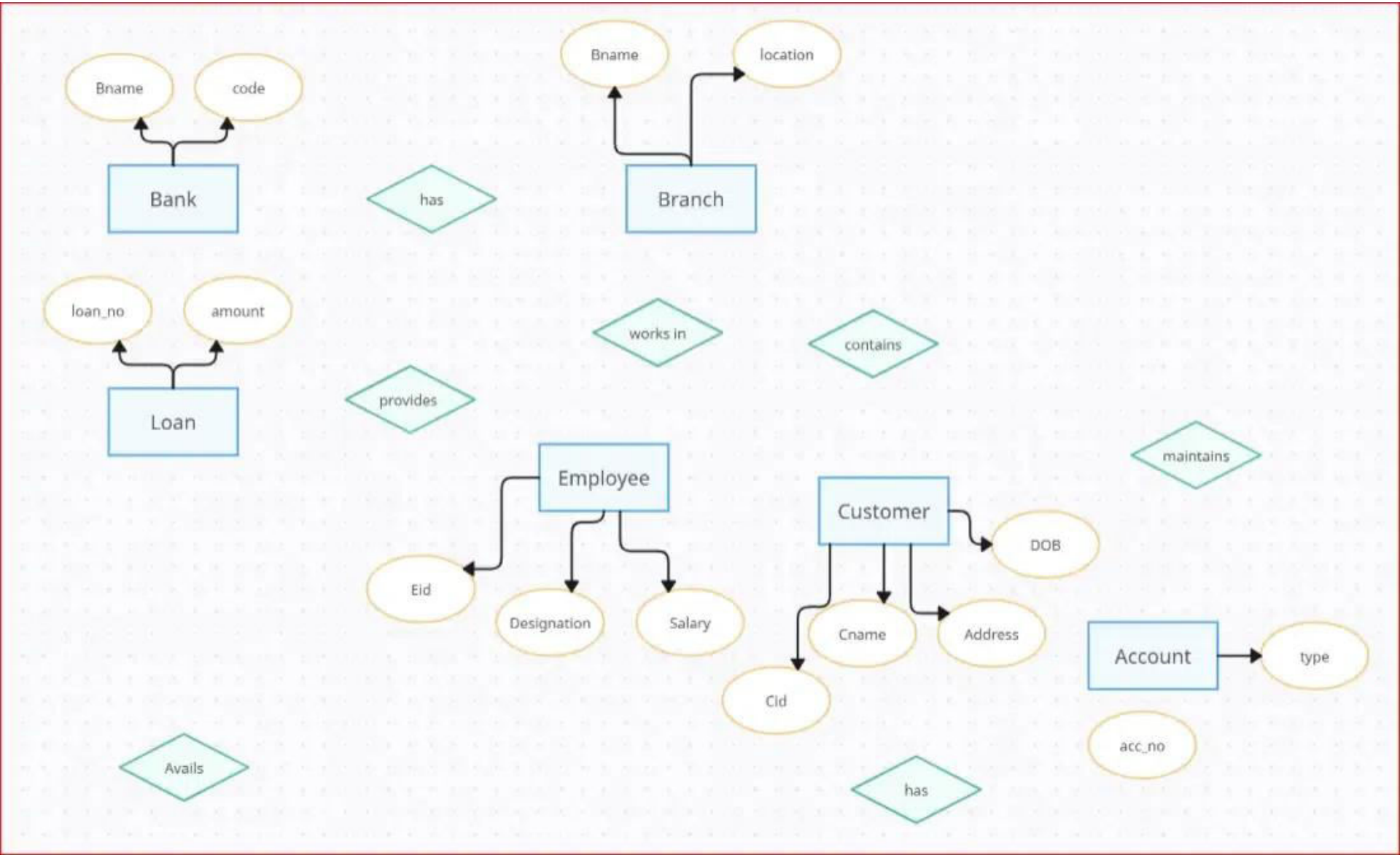


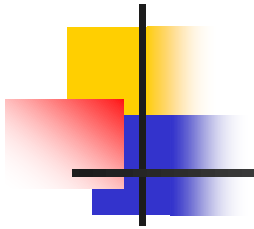
**Establishing Relationships**

Entities have some relationships with each other. Relationships define **how entities are associated with each other**.

**Let's Establishing Relationships between them are:**

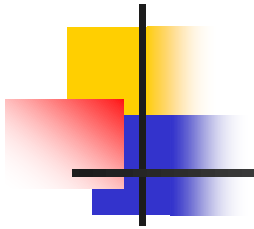
- The **Bank** has **branches**.
- The **Branch** provides **loan**.
- The **Employee** works in **branch**.
- The **Branch** contains **customers**.
- The **Customers** has **account**.
- The **Branch** maintains **account**.
- The **Customer** avails **loan**.


Bname
code
Bank
has
Bname
location
Branch
loan_no
amount
Loan
works in
contains
provides
Employee
maintains
Customer
DOB
Eid
Designation
Salary
Cname
Address
Cid
Account
type
acc_no
Avails
has

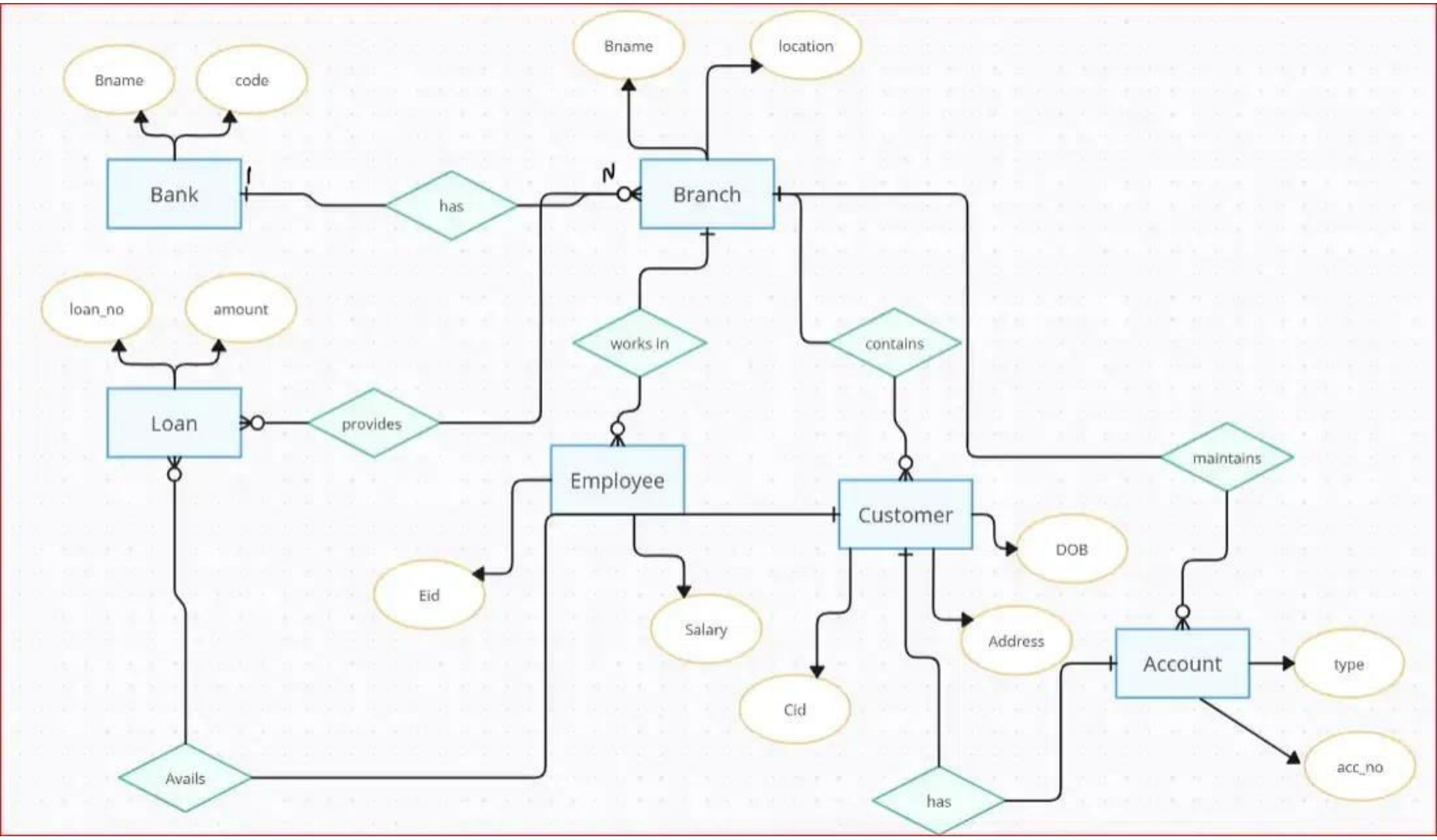


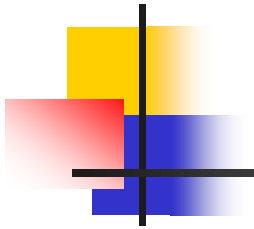
# Specify cardinality for Bank:

- **Bank** and **branch** has **One to Many** relationship (a bank has multiple branches).
- **Branch** and **loan** has also **One to Many** relationship(one branch can provide multiple loans).
- **Branch** and **employee** has **One to Many** relationship(one branch employs multiple employees).



- **Branch** and **account** has **One to Many** relationship(one branch maintains multiple accounts).
- **Branch** and **customer** has **One to Many** relationship(one branch has multiple customers; each customer belongs to one primary branch).
- **Customer** and **account** has **One to Many** relationship(one customer can have multiple accounts, each account belongs to exactly one customer).
- **Customer** and **loan** has **One to Many** relationships(each customer can have multiple loans; each loan belongs to exactly one customer)

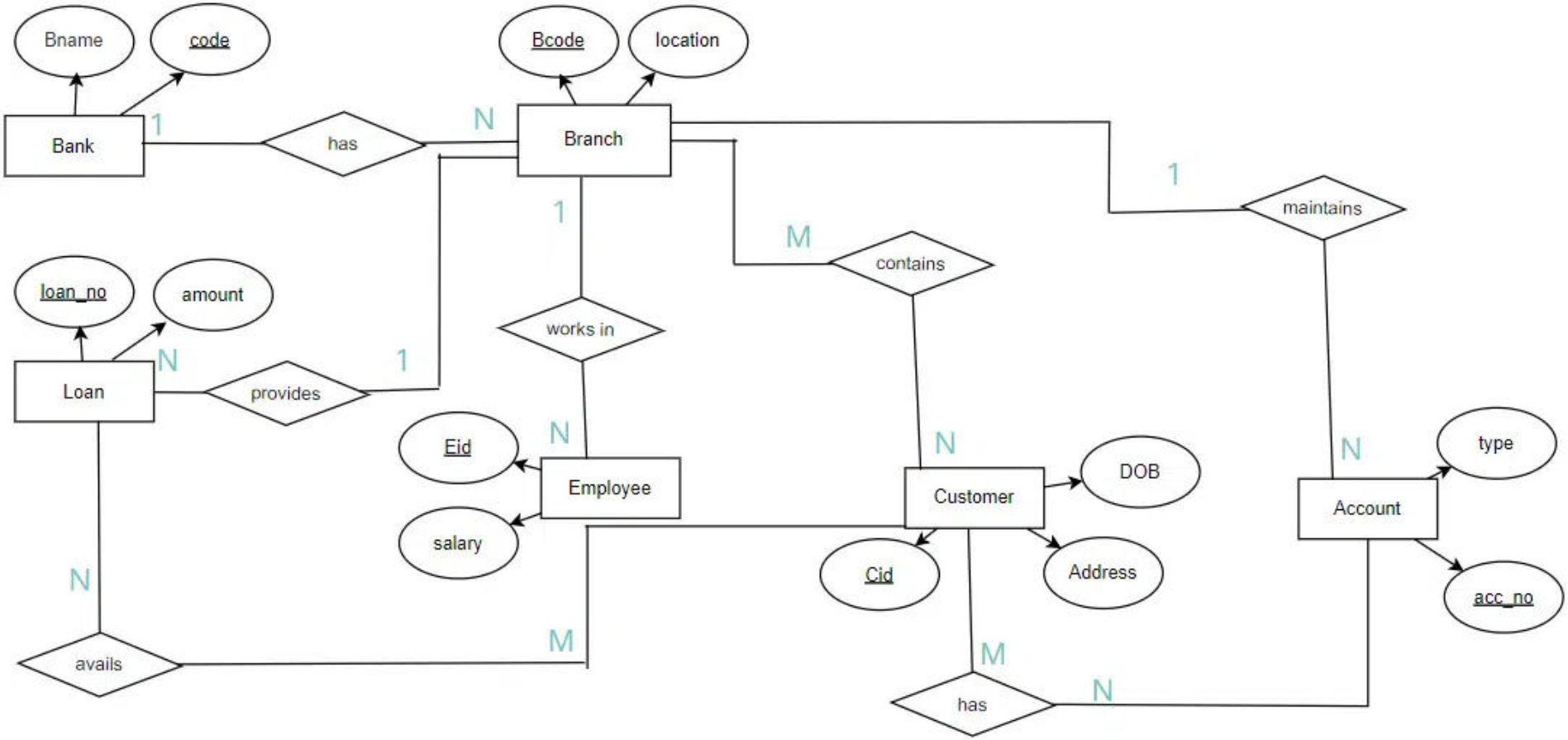

Bname
code
Bank
1
has
N
Bname
location
Branch
loan_no
amount
Loan
provides
works in
contains
Employee
Customer
maintains
Eid
Salary
DOB
Address
Cid
Account
type
acc_no
Avails
has



**Identify Primary Keys**

Primary keys are the **unique** identifier for each record in database table. It is denoted by an underline under the attribute name.

- The Primary key of **Bank** is **code**.
- The Primary key of **Branch** is **branch_code.**
- The Primary key of **Employee** is **Eid.**
- The Primary key of **Customer** is **Cid.**
- The Primary key of **Loan** is **loan_no**.
- The Primary key of **Account** is **acc_no**


Bname
code
Bcode
location
Bank
1
has
N
Branch
1
maintains
M
contains
loan_no
amount
works in
Loan
N
provides
1
Eid
N
Employee
salary
N
Customer
DOB
Cid
Address
N
Account
type
acc_no
N
avails
M
M
has
N