ER MODEL

Instructor : Nitesh Kumar Jha

niteshjha@soa.ac.in

ITER,S'O'A(DEEMED TO BE UNIVERSITY)

July 2018

Review

- The ER Model

Content

- The ER Model
- The ER Diagram

Keys - I

- The values of the attributes of an entity must be such that, they uniquely identify the entity.
- No two entities in an entity set are allowed to have exactly the same value for all its attributes.
- A Key makes one entity distinguishable from other entity in the same entity set.
- A key for an entity set is a set of attributes, whose value uniquely determines each and every entity in an entity set.
- Different keys used are
  - Candidate key
  - Super key
  - Primary Key
  - Alternate key

Keys - II

- Ex. Consider the entity set
  - Student(reg_no, name, branch, address, ph_no, DoB)
- Candidate key: is an attribute who's value uniquely determine each and every entity in an entity set.
  - Ex. Candidate keys {name, ph_no}, {name, address} or {reg_no}
  - For a given entity set, more than one candidate keys can be designed.
- Super Key: any superset of an candidate key is referred as super key
  - i.e. A candidate key is the minimal super key, of which no proper subset can again act as a key.
  - Ex. Super Keys: {reg_no, name}, {reg_no, name, ph_no}, {reg_no, name, branch}, {reg_no, name, address} or {reg_no, name, DoB}, . . .

Keys - III

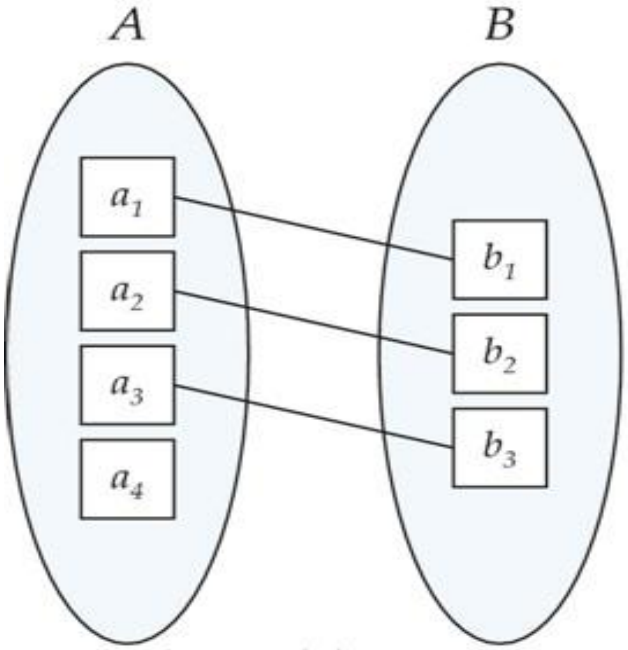
- Primary key: Out of multiple candidate keys, one of the key is chosen by the database engineer as the primary key.
- Primary key is used as principal means to uniquely identify each and every entity in the entity set.
  - Ex. Primary key {reg_no}
- Alternate Key: Rest of the candidate keys (excluding primary key) are called alternate keys
  - Ex. Alternate Keys {name, ph_no} and {name, address}
- Keys are represented in ER Diagram by underlining the key attribute(s)

ER Model: Constraints

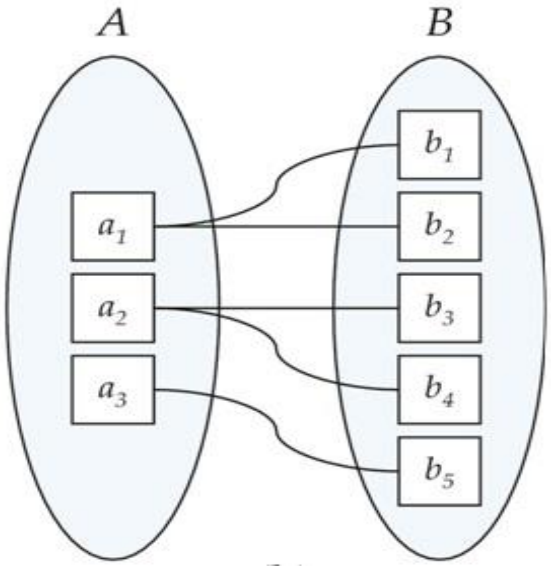
- Constraints are the reflections of business rules/logic to which the database design must comply to.
- These are the characteristics of relationship set.
- ER Model supports two types of constraints:
  - Mapping Cardinalities or Cardinality constraints
    - How many entities in one entity set is associated with entities of another entity set?
  - Participation Constraints.
    - Tells about the total or *partial* participation of an entity set in a relationship set.
    - i.e. Whether all entities or few of them are participating in a relationship set.
    - Ex. Participation of Student entity set in the relationship advisor is total. But participation of Instructor may be partial

ER Model: Mapping Cardinalities - I

- Mapping cardinalities or cardinality ratios tells the number of entities to which another entity can be associated via a relationship set.
  - One-to-One: An entity in entity set A is associated with at most one entity in another entity set B.
  - Ex. One Student enrolls in One course

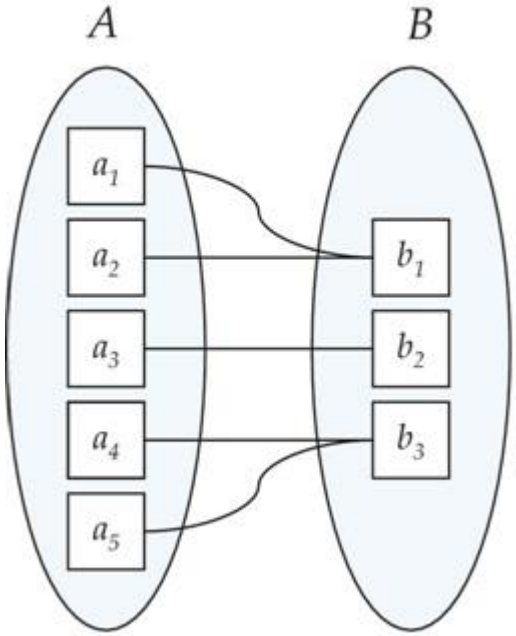


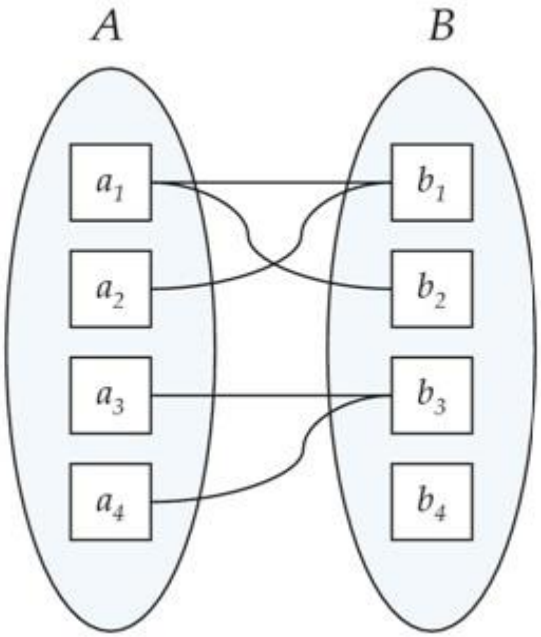
- One-to-Many: An entity in entity set A is associated with any number of entities in entity set B.
  - Ex. One Teacher advises Multiple students



ER Model: Mapping Cardinalities - II

- Many-to-One: An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number of entities in A.
  - Ex. Many Students are enrolled in One Course

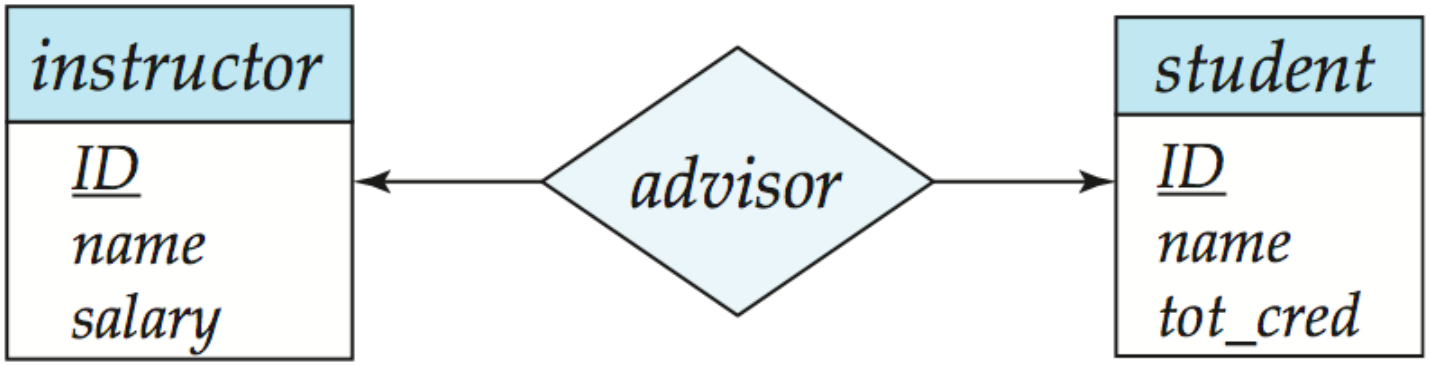




- Many-to-Many: An entity in A is associated with any number of entities in B and an entity in B is associated with any number of entities in A.
  - Ex. One Customer have Many bank Accounts and One Account can have Multiple number of Customers.

ER Diagram: Cardinality Constraints-I

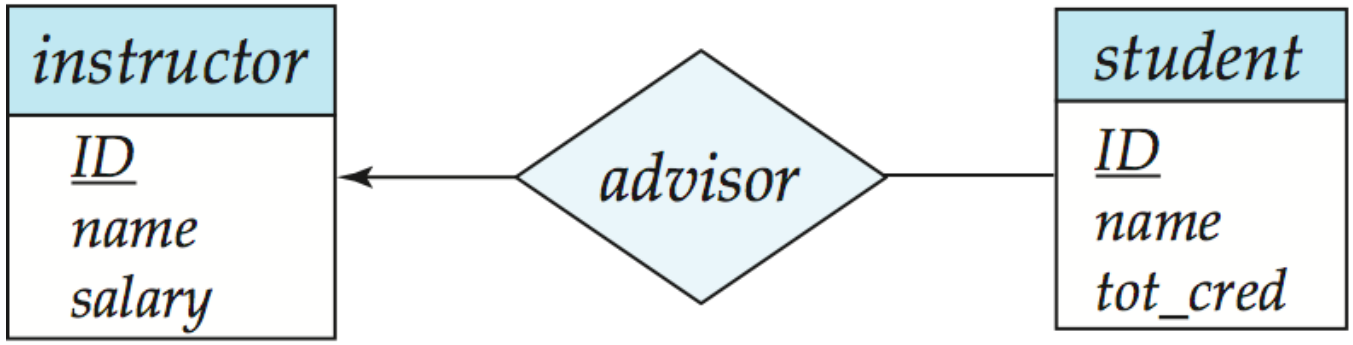
- One: A directed line (→) from relationship set to entity set
- Many: an undirected line (—)



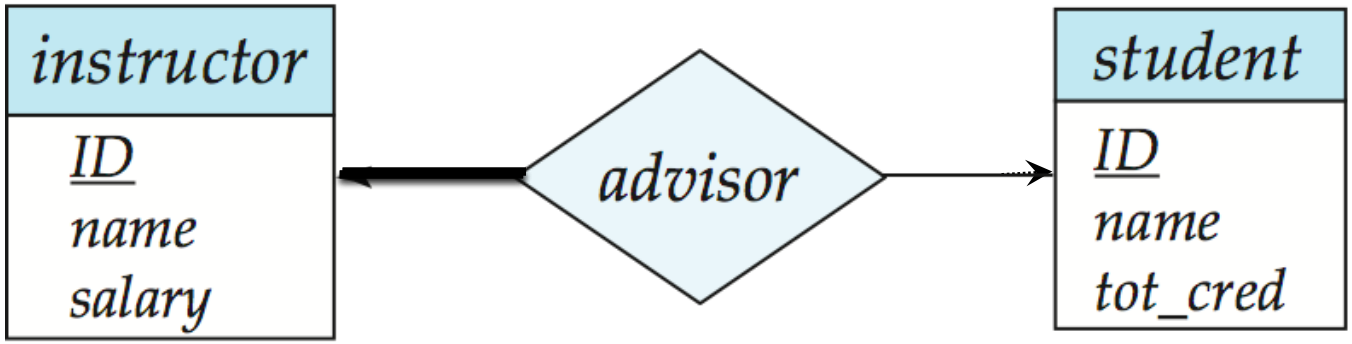
- One-to-One relationship between an *instructor* and a *student* :
  - A student is associated with at most one *instructor* via the relationship *advisor*
  - Task: Draw ER diagram where, A *student* is associated with at most one *department* via *stud_dept*

ER Diagram: Cardinality Constraints-II

- One-to-Many: an instructor is associated with several (including 0) students via *advisor*

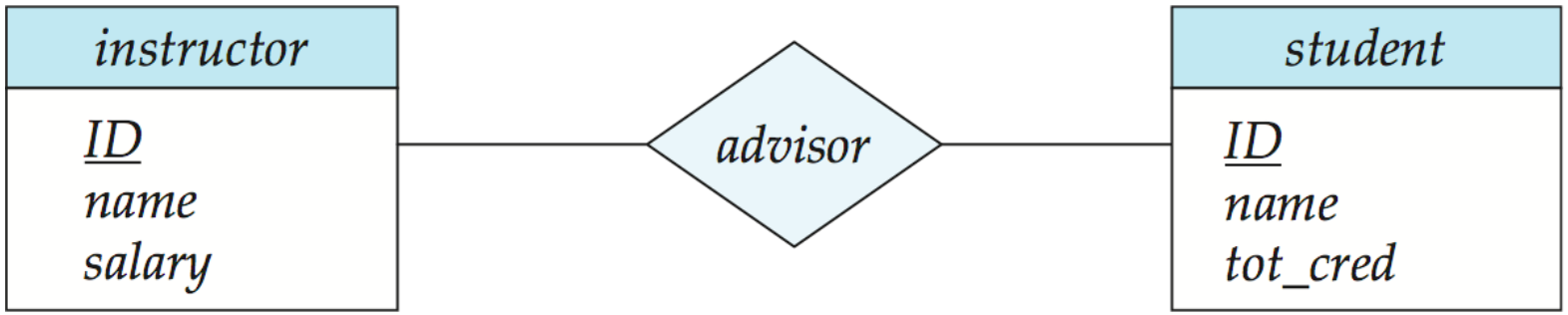


- Many-to-One: a student is associated with several (including 0) instructors via advisor



ER Diagram: Cardinality Constraints-III

- Many-to-Many:
  - An instructor is associated with several (possibly 0) students via *advisor*
  - A student is associated with several (possibly 0) instructors via *advisor*



ER Diagram: Total and Partial Participation

- Total participation is indicated by double line:
  - i.e. every entity in the entity set participates in at least one relationship in the relationship set

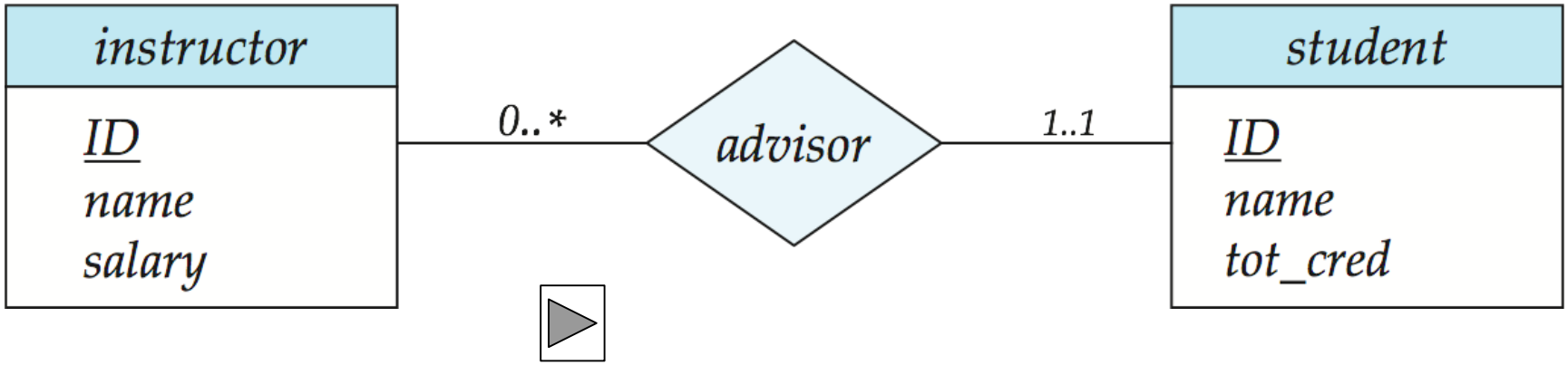
Ex. participation of *student* in *advisor* relation is *total*

- every *student* must have an associated instructor

- Partial participation: some entities may not participate in any relationship in the relationship set
  - Ex. participation of *instructor* in *advisor* is partial

ER Diagram: Complex Constraints

- A line may have an associated minimum and maximum cardinality, shown in the form *l..h*, where
  - A minimum value of 1 indicates total participation
  - A maximum value of 1 indicates that the entity participates in at most one relationship
  - A maximum value of * indicates no limit.



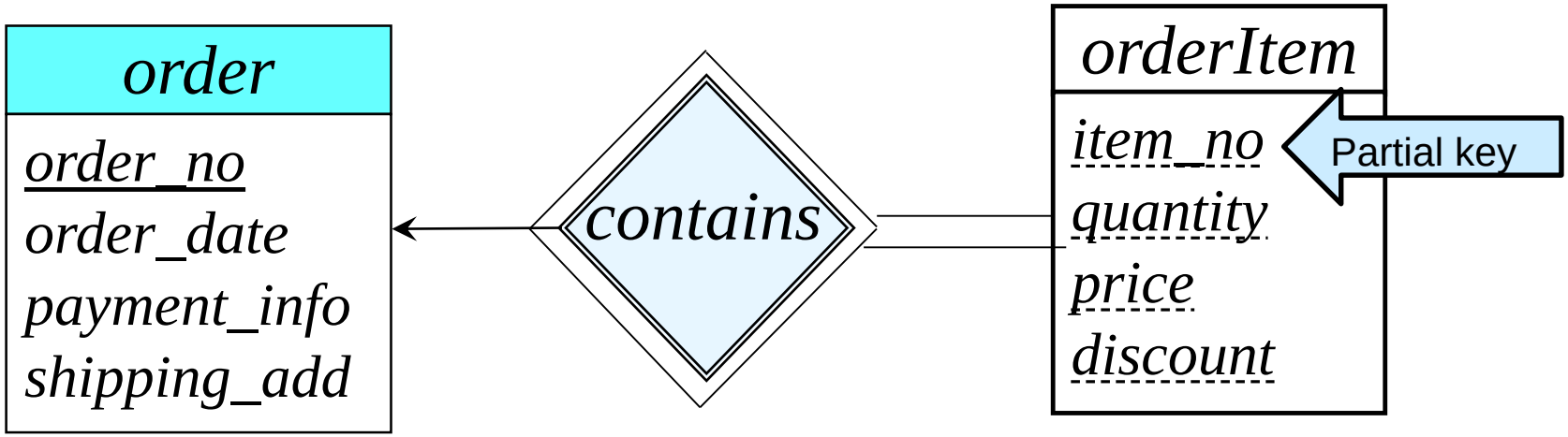
Instructor can advise 0 or more students. A student must have 1 advisor; cannot have multiple advisors

ER Model: Weak Entity Set - I

- An entity set that cannot be uniquely identified by its attributes alone are called weak entity sets.
  - i.e. the entity set does not have sufficient attributes to form a primary key and are dependent on other entity sets for their existence.
- An entity having own primary key is called a strong entity.
- The strong entity upon which the weak entity depends is referred as identity entity
- The relationship set through which the weak entity is connected to its identity (strong) entity set is called identifying relationship set.

ER Diagram: Weak Entity Set

- Ex. orderItem(item_no, quantity, price, discount)

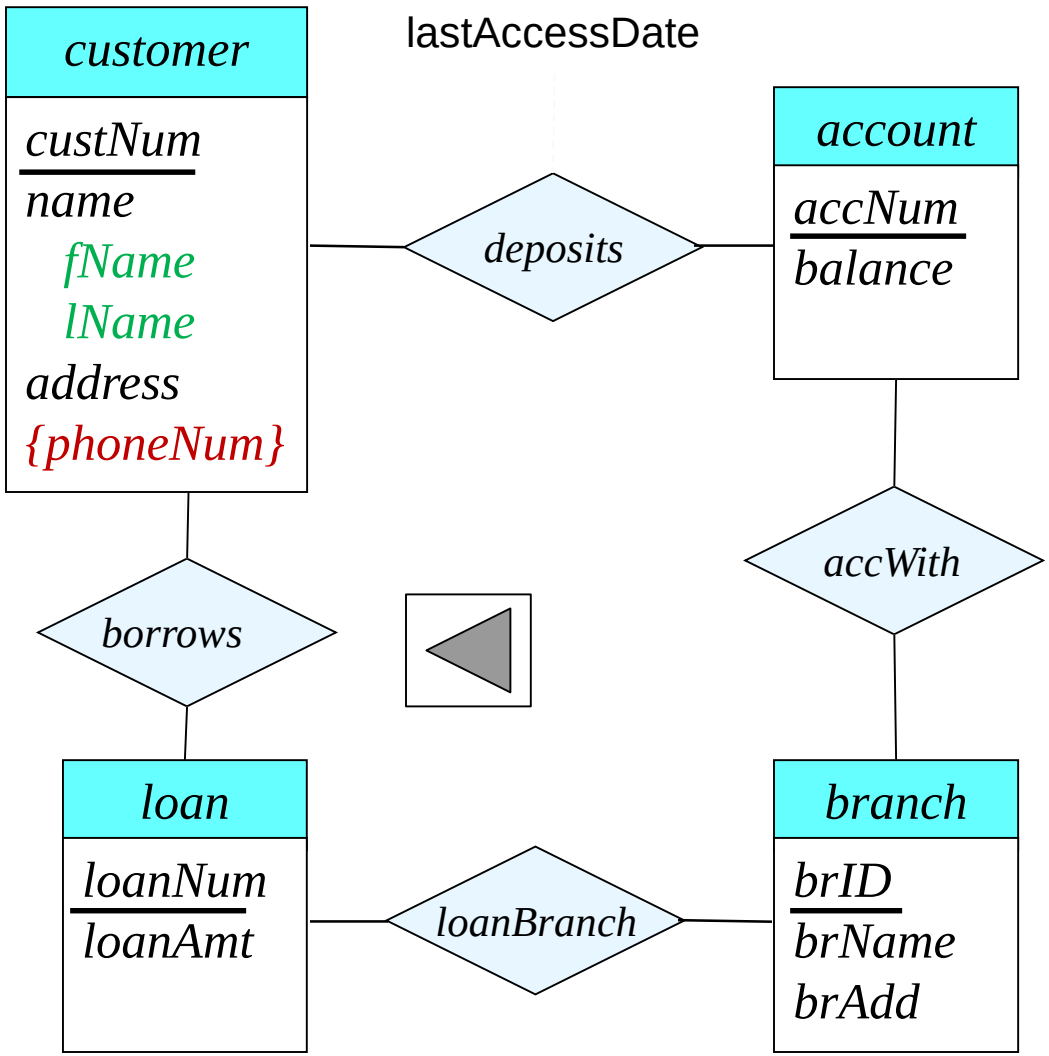


- The relationship set with its identity entity (strong) set is represented using double line diamond
- The line from weak entity to identifying relationship set is double lined
- A set of attributes of a weak entity set is referred as the *discriminator* (or *partial key*) that is used as a means of distinguishing among all those entities in the weak entity set that depend on one particular strong entity

Weak Entity Set - III

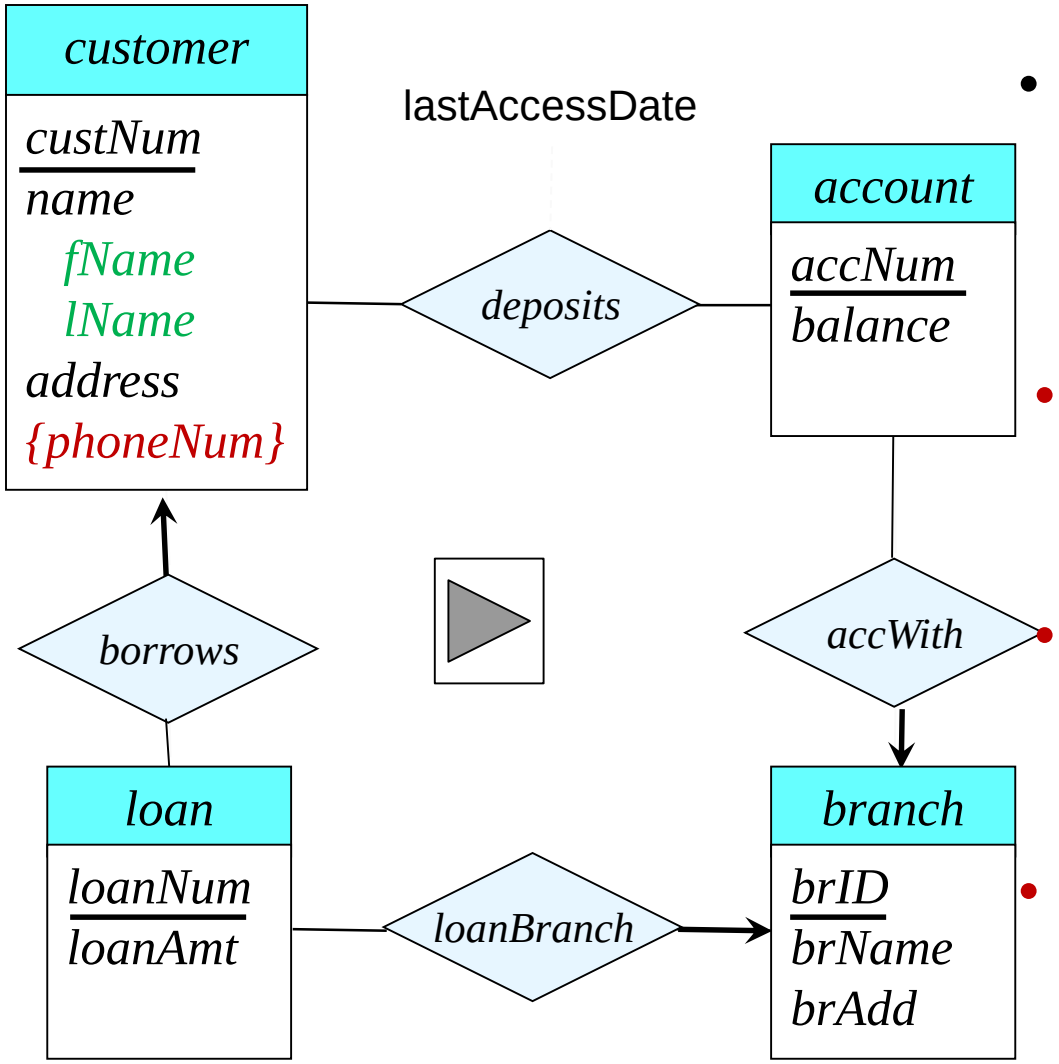
- The primary key of a weak entity set is formed by
  - the primary key of the identifying entity set, plus
  - the weak entity set's discriminator.
  - Example: {order_no, item_no} is the primary key in the table **orderItem**
- In the case of the entity set *section*, *its primary key is {course_id, sec_id, year, semester}*
- The participation of the weak entity set with its identifying relationship set is always total.
- The cardinality of the identifying relationship set is always many-to-one from the weak entity set to its identifying entity set
- The identifying relationship set is not allowed to have any descriptive attribute
- Weak entity can participate in any other relationship set after forming the primary key.

Complex Attributes



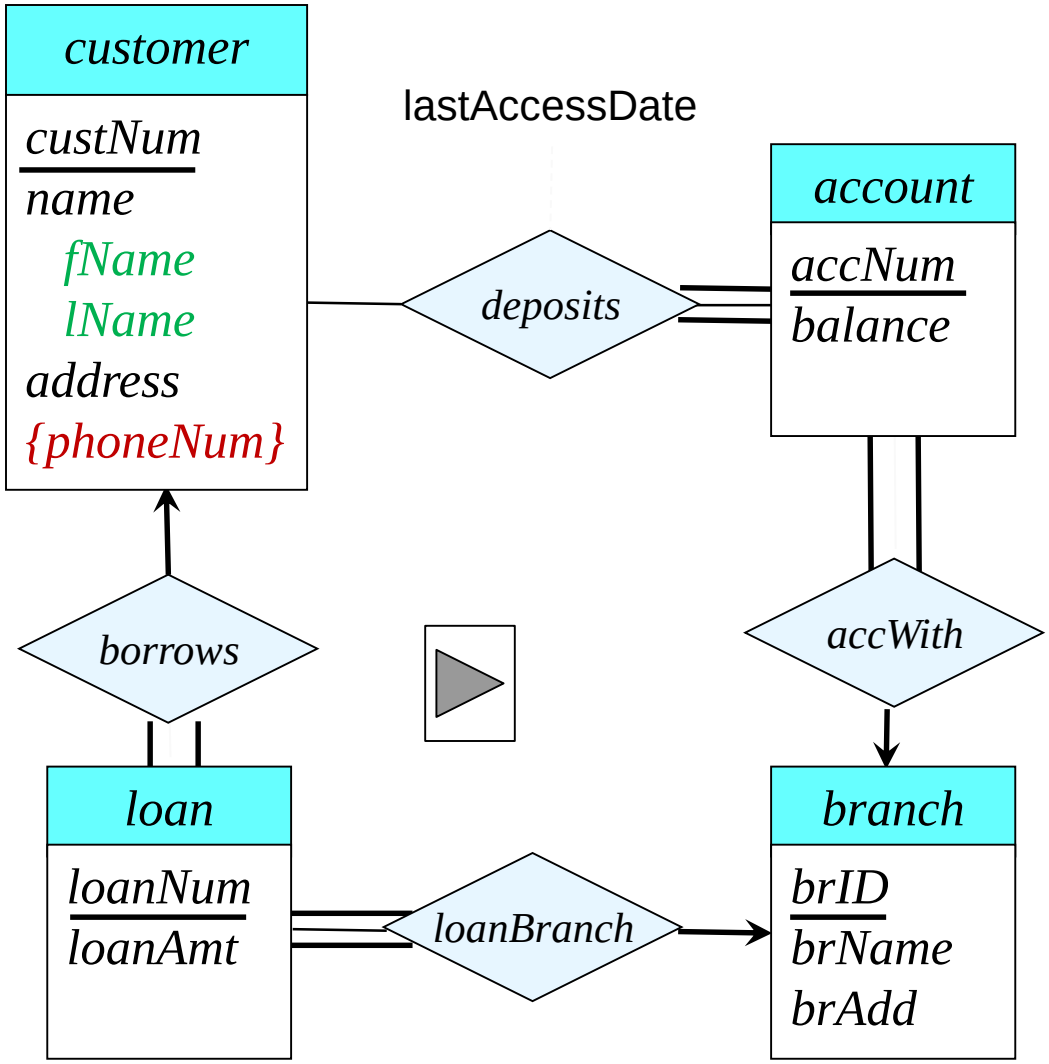
- Make name attribute of customer entity set as composite attribute with component attributes as
  - fName & lName
- Make phoneNum attribute of customer entity set as multivalued attribute as one customer can have multiple phone numbers

Cardinality Constraints



- There is a many-to-many relationship between customer and account via deposit
- many-to-one relationship between account and branch
- many-to-one relationship between loan and branch
- many-to-one relationship between loan and customer

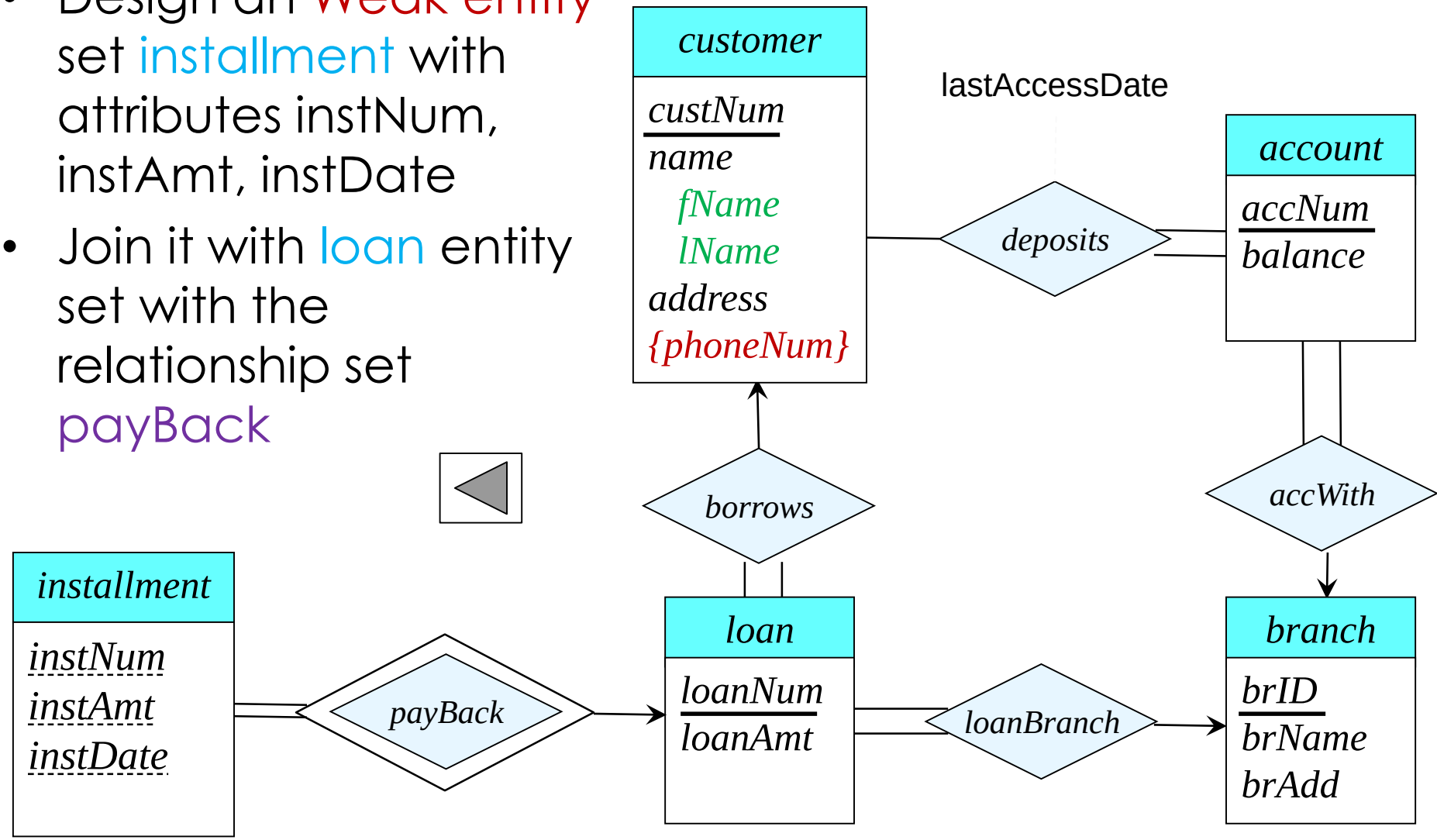
Participation Constraints



- Participation of entity set account in deposit relationship & accWith relationship is total
- Participation of loan entity set in loanBranch relationship & borrows relationship is total

Weak Entity Set

- Design an Weak entity set installment with attributes instNum, instAmt, instDate
- Join it with loan entity set with the relationship set payBack



Thank You