

DBMS –

ENTITY

-RELATIONSHIP

MODEL

Entity-Relationship Model

- Design Process
- Modeling
- Constraints
- E-R Diagram

Modeling

- A *database* can be modeled as:
 - a collection of entities,
 - relationship among entities.
- An **entity** is an object that exists and is distinguishable from other objects.
 - Example: specific person, company, event, plant
- Entities have *attributes*
 - Example: people have *names* and *addresses*
- An **entity set** is a set of entities of the same type that share the same properties.
 - Example: set of all persons, companies, trees, holidays

Entity Sets *customer* and *loan*

customer_id customer customer customer
 name street city

loan_ amount
number

321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

customer

L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

loan

Relationship Sets

- A **relationship** is an association among several entities

Example:

Hayes depositor A-102

customer entity relationship set *account* entity

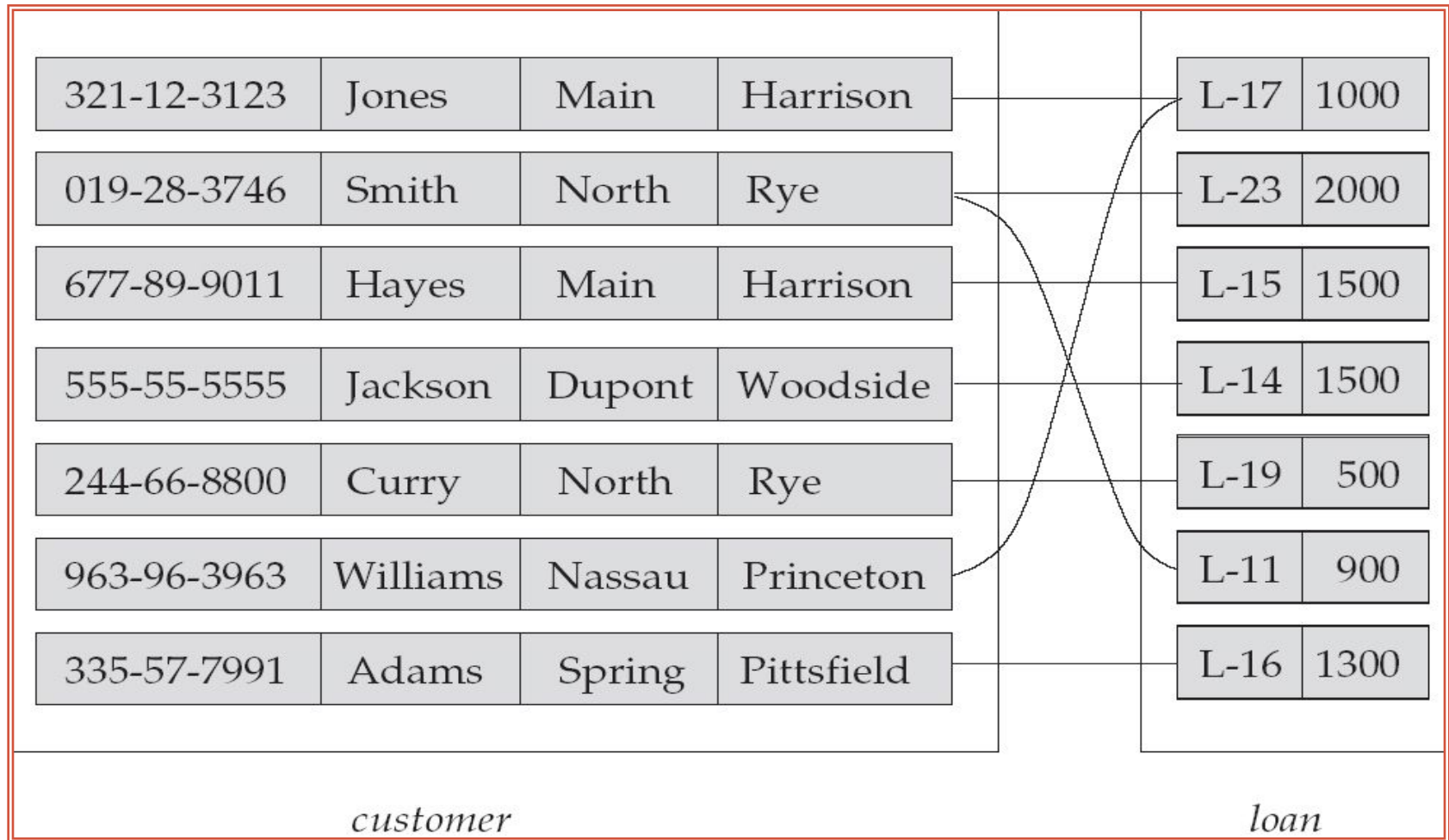
- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

where (e_1, e_2, \dots, e_n) is a relationship

- Example: $(\text{Hayes}, \text{A-102}) \in \text{depositor}$

Relationship Set *borrower*



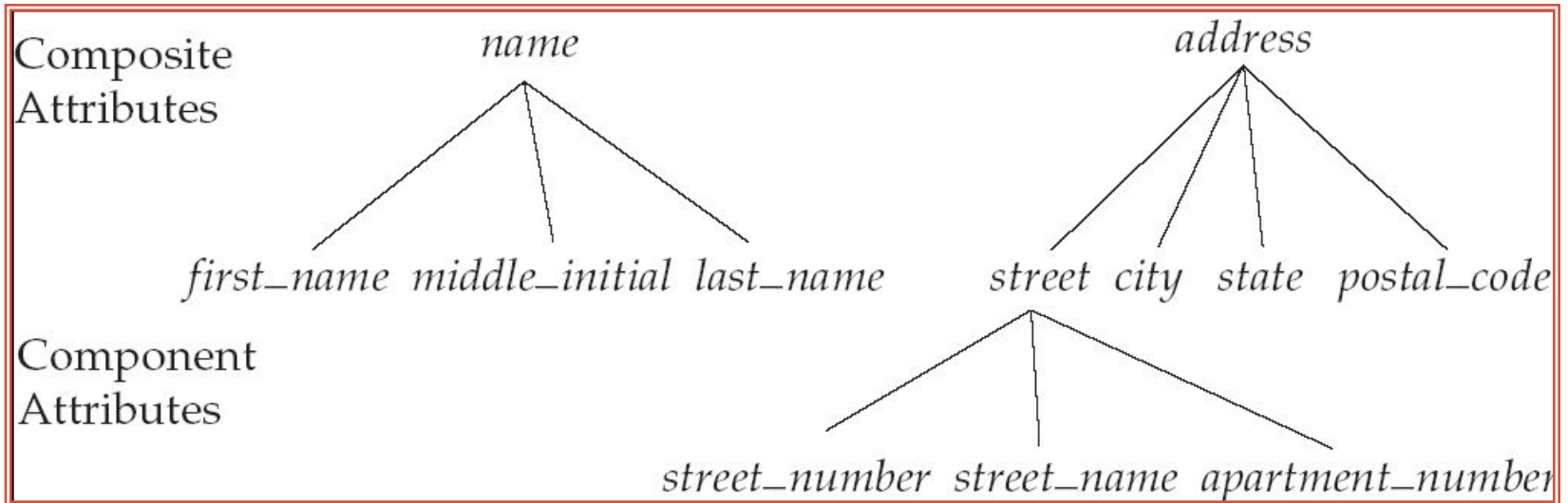
Degree of a Relationship Set

- Refers to number of entity sets that participate in a relationship set.
- Relationship sets that involve two entity sets are **binary** (or degree two). Generally, most relationship sets in a database system are binary.
- Relationship sets may involve more than two entity sets.
 - Example: Suppose employees of a bank may have jobs (responsibilities) at multiple branches, with different jobs at different branches. Then there is a ternary relationship set between entity sets *employee*, *job*, and *branch*

Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
- Example:
 - *Student (Regno, Name, DoB, Address, Phone)*
 - *Employee (EmpID, Name, DoJ, Qualification)*
- **Domain** – the set of permitted values for each attribute
- Example:
 - *Student (101, Ram, 12.06.2000, AbC Street, 123456789)*
 - *Employee (1257, Latha, 03.11.1999, PhD)*
- Attribute types:
 - *Single-valued and multi-valued attributes*
 - Example: multivalued attribute: *phone_numbers*
 - Example: Single value attribute: *Age, DoB*
 - *Derived attributes*
 - Can be computed from other attributes
 - Example: Age, given date_of_birth
 - Composite attributes
 - Combination of two or more attributes
 - Example: Name (First, Middle, Last), Address (DoorNo, Street Name, Area name, City, Pin)

Composite Attributes



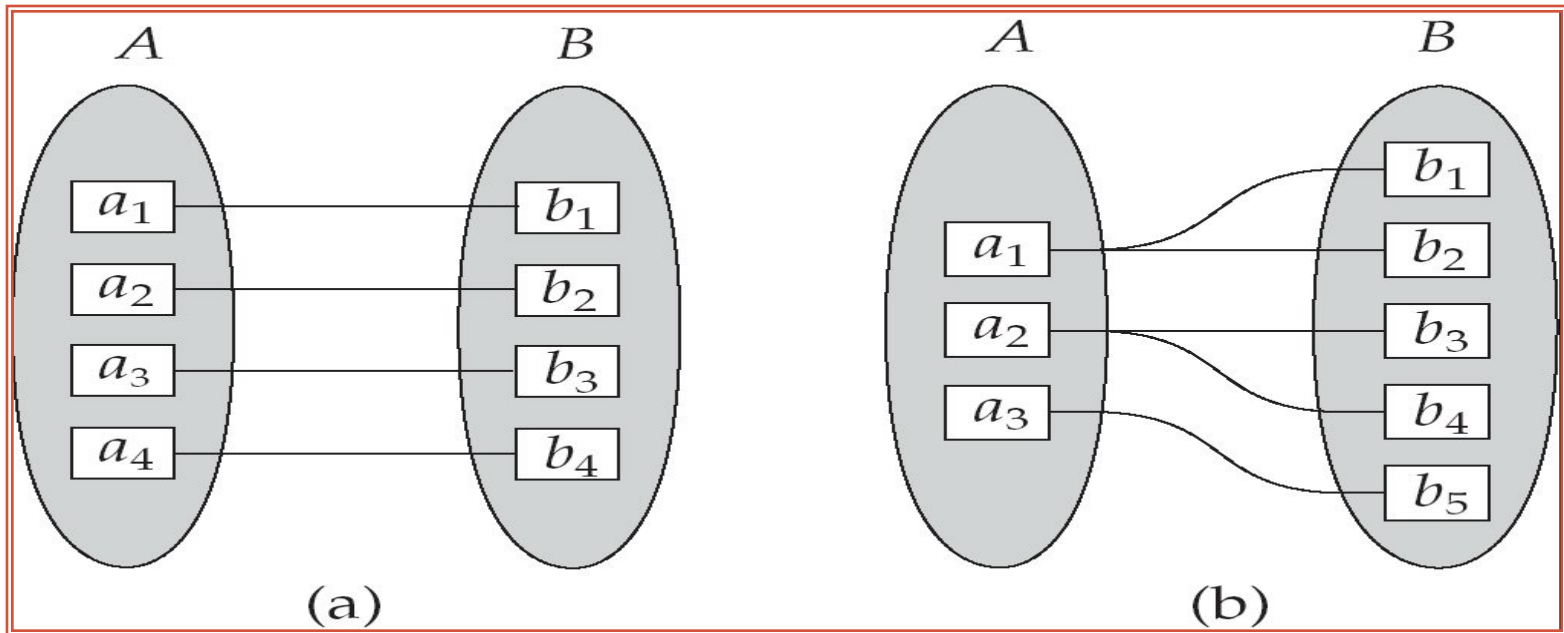
Terminologies – a visit

- Entity
- Entity Set
- Relationship
- Relationship set
- Attributes
 - Single valued
 - Multi valued
 - Composite
 - Derived

Mapping Cardinality Constraints

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
 - One to one
 - One to many
 - Many to one
 - Many to many

Mapping Cardinalities



One to one

One to many

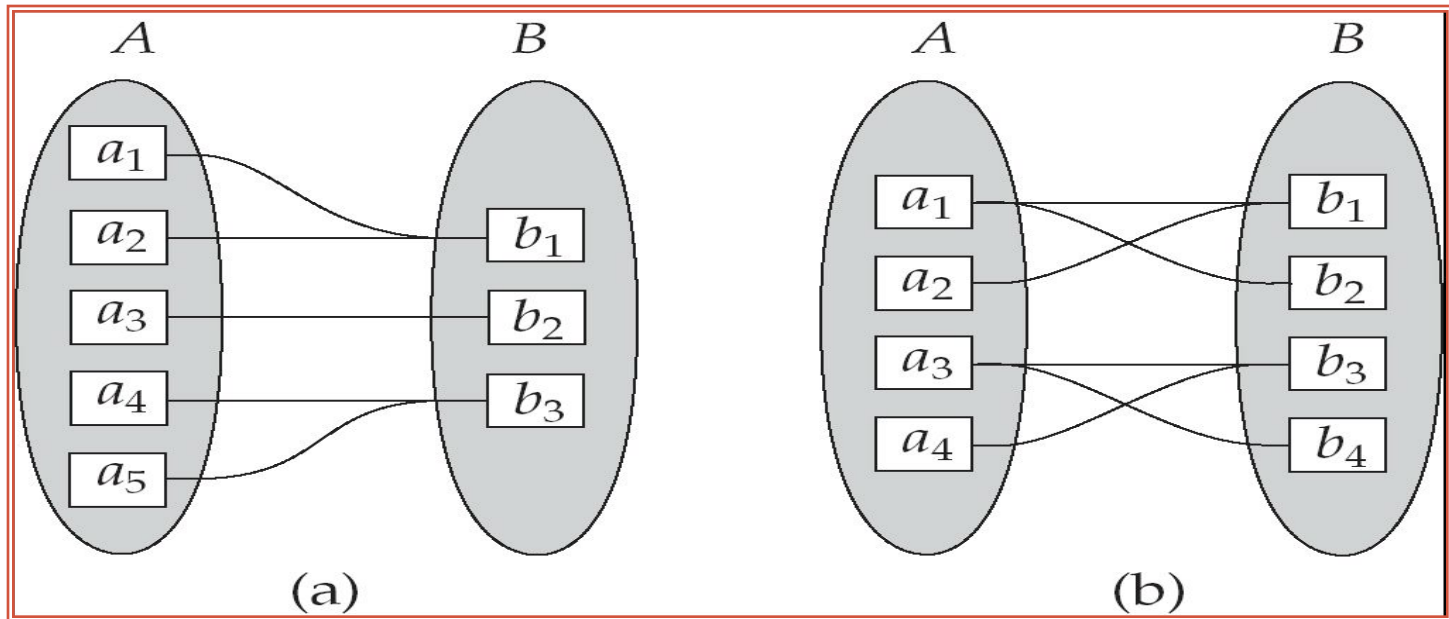
Note: Some elements in A and B may not be mapped to any elements in the other set

Example: Set A – Department details Set B – Location

Mapping A to B is one to one (CSE – AB3)

Mapping B to A is many to one (AB3 has many departments)

Mapping Cardinalities



Many to one

Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set

Set A – Students, Set B – Department

Many students belong to one Department – Many to one

Set A – Courses Set B – Students

Many Courses are taken by many students – Many to many

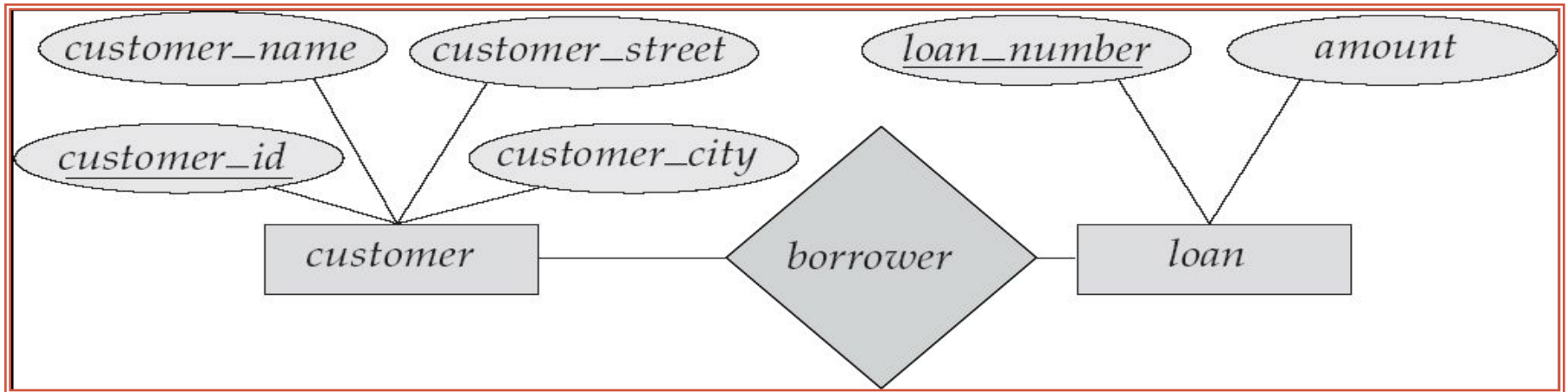
Keys

- Primary Key: Set of attributes that uniquely identifies a tuple in an entity – Should not allow duplicate values in that attribute
- Identify Primary keys in the following:
 - Student (RegNo, Name, Address, DoB)
 - Employee (EmpID, Name, Address, DoB, DoJ)
 - Department (DeptID, Name, Location)
 - Items (ProductID, Name, stock, price)
 - Account(AccNo, Name, Balance)
- Why Primary Key? -- To uniquely identify a particular tuple.
 - If no primary keys – Data integrity is lost – all students have same regno. – See the confusion.
- Aadhar No – Primary key to uniquely identify a citizen in India.

Composite Primary Key

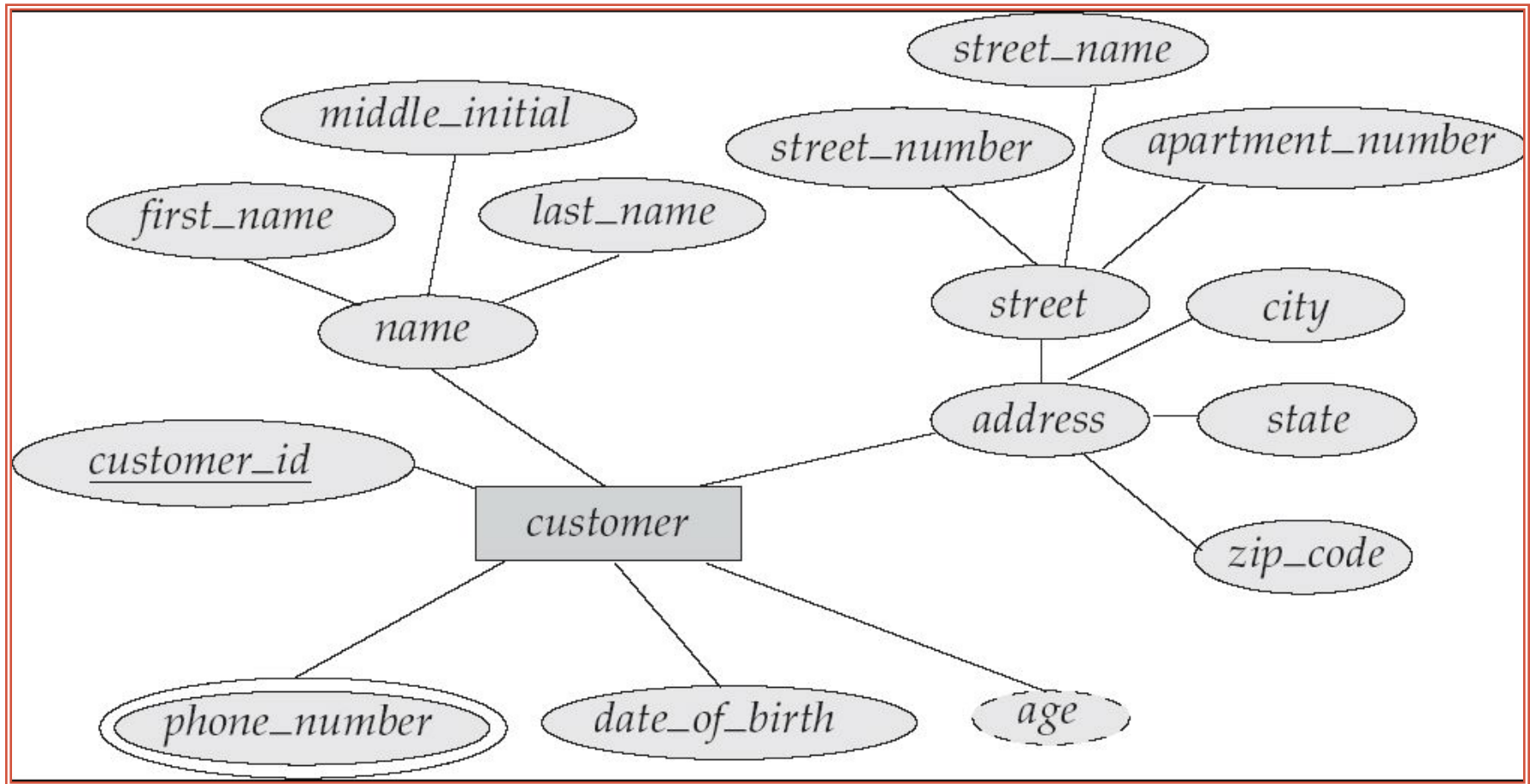
- One or more attribute is a primary key
- Example: One order has multiple products
 - Order(**OrderNo**, **ProductCode**, Quantity, price) – Only with OrderNo and ProductCode we can identify single tuple
 - Only OrderNo will give multiple tuples
 - Only ProductCode will give multiple tuples
 - Exam(**RegNo**, **CourseCode**, Grade) – Only with Regno and CourseCode we can get single tuple.
 - Only RegNo will give grade of all courses
 - Only CourseCode will give results of all students

Entity –Relationship Diagrams (ERD)

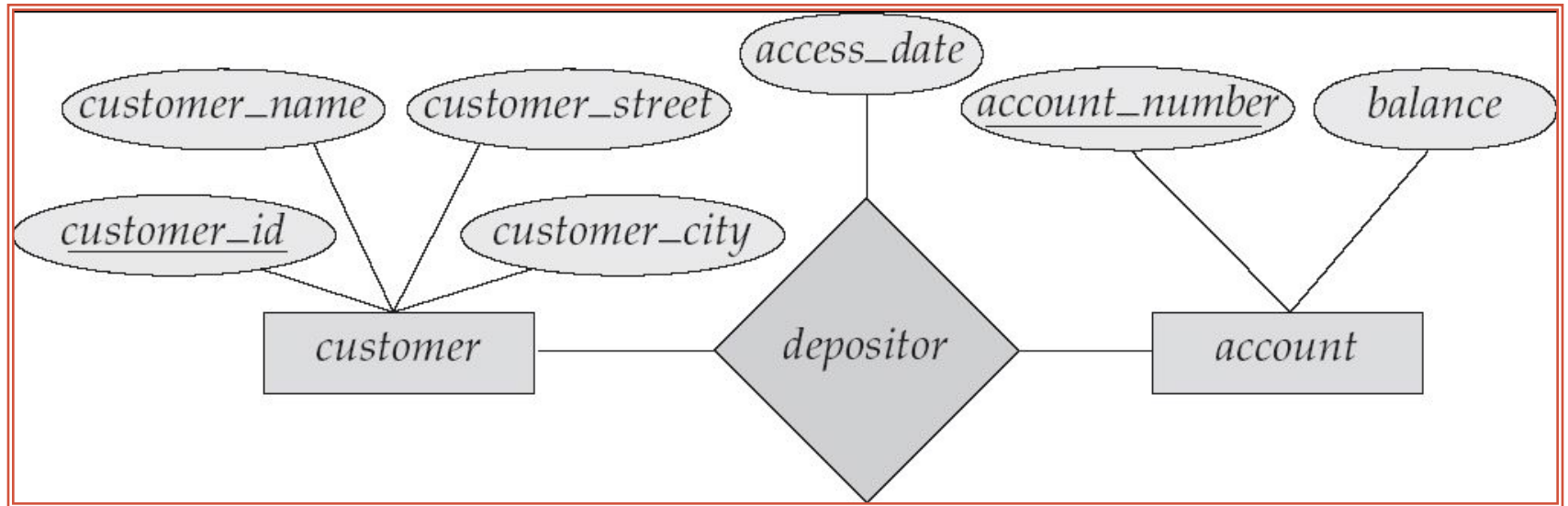


- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
 - Double ellipses represent multivalued attributes.
 - Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes

E-R Diagram With Composite, Multivalued, and Derived Attributes

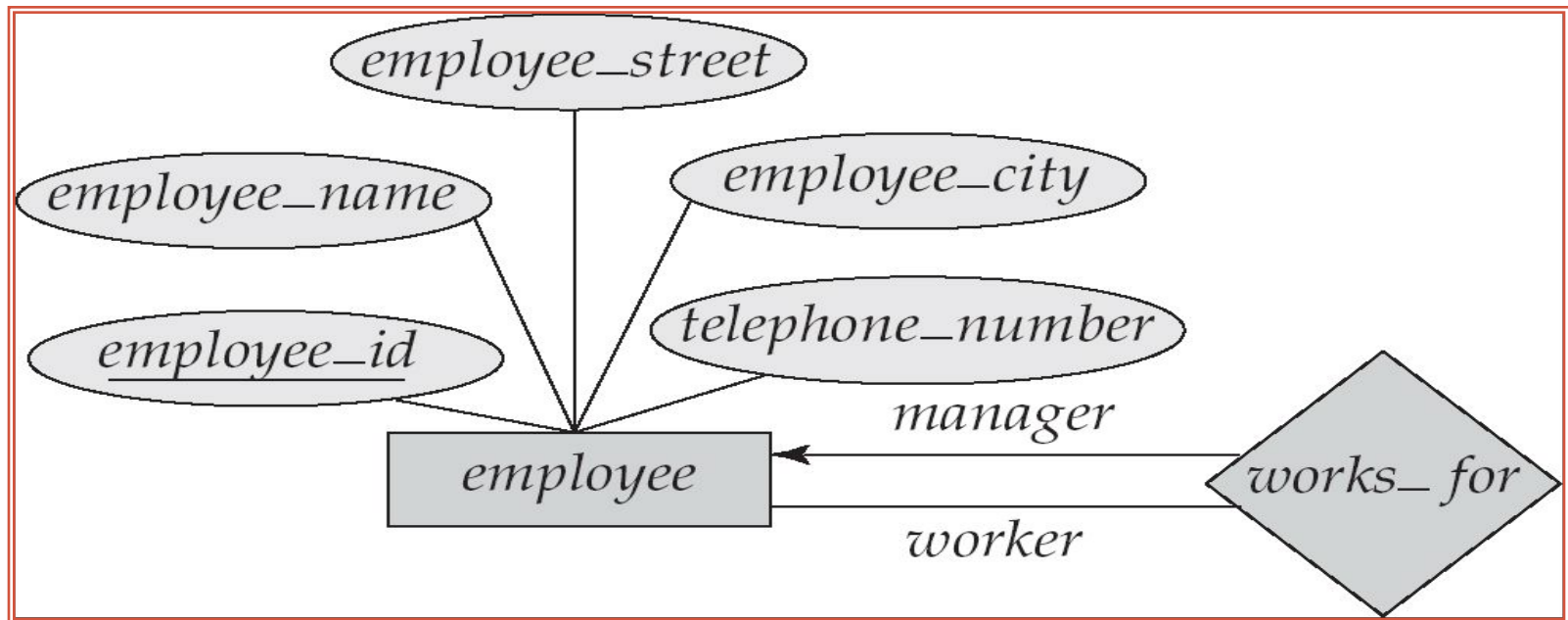


Relationship Sets with Attributes



Roles

- Entity sets of a relationship need not be distinct
- The labels “manager” and “worker” are called **roles**; they specify how employee entities interact via the works_for relationship set.
- Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship

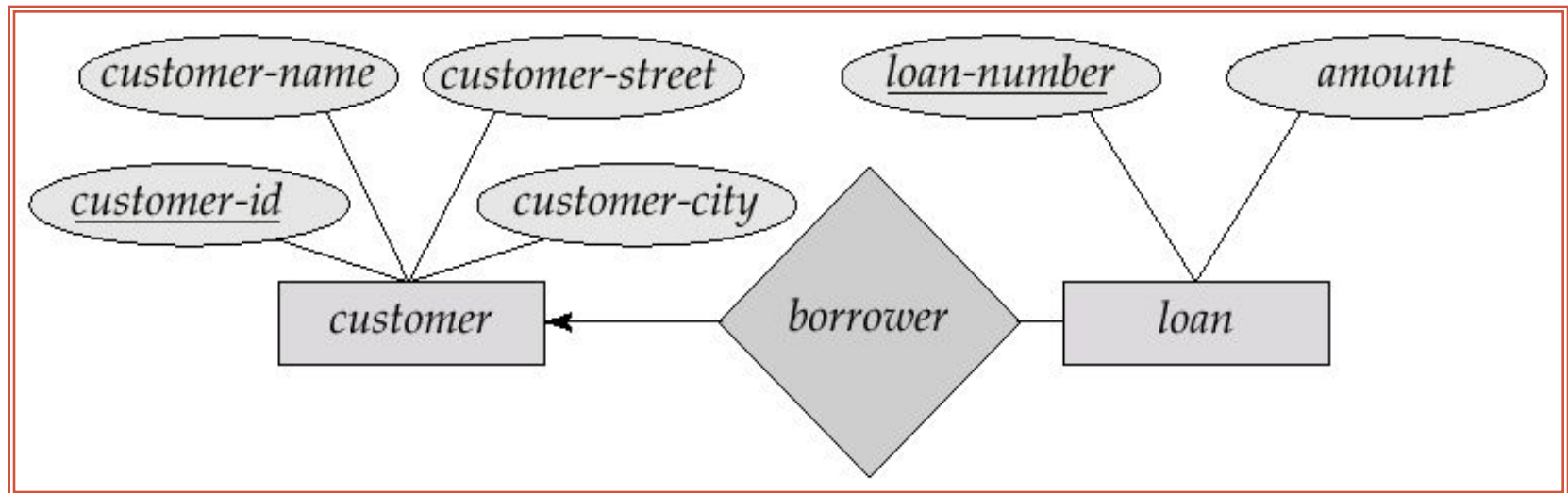


Cardinality Constraints

- We express cardinality constraints by drawing either a directed line (\rightarrow), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.
- One-to-one relationship:
 - A customer is associated with at most one loan via the relationship *borrower*
 - A loan is associated with at most one customer via *borrower*

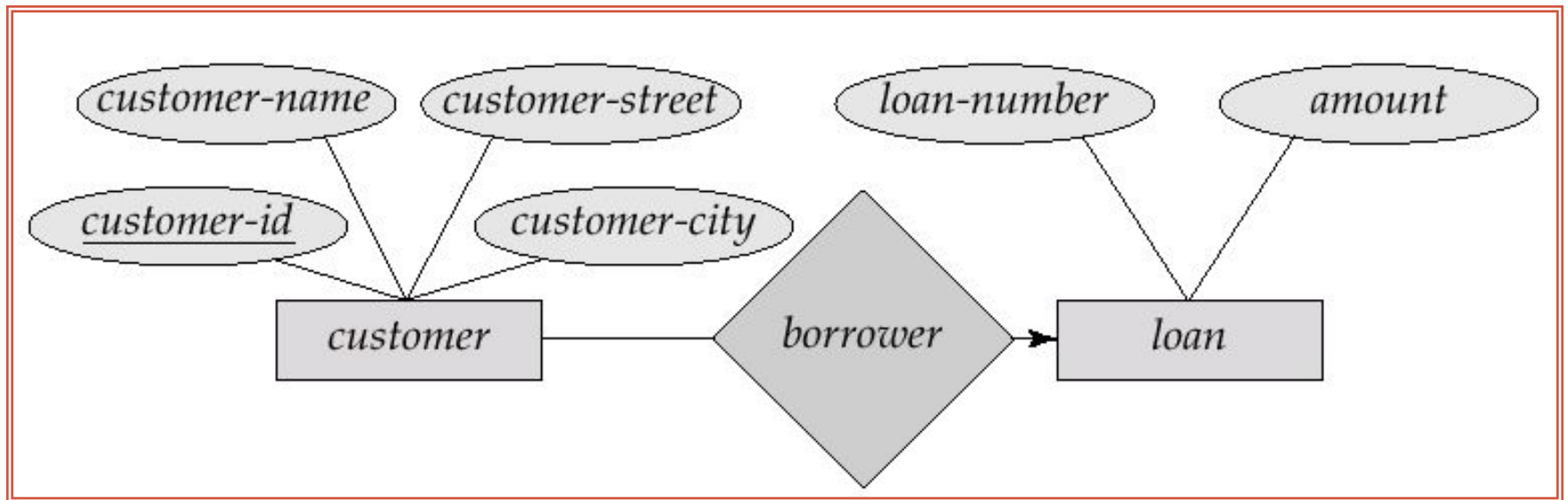
One-To-Many Relationship

- In the one-to-many relationship a loan is associated with at most one customer via *borrower*, a customer is associated with several (including 0) loans via *borrower*



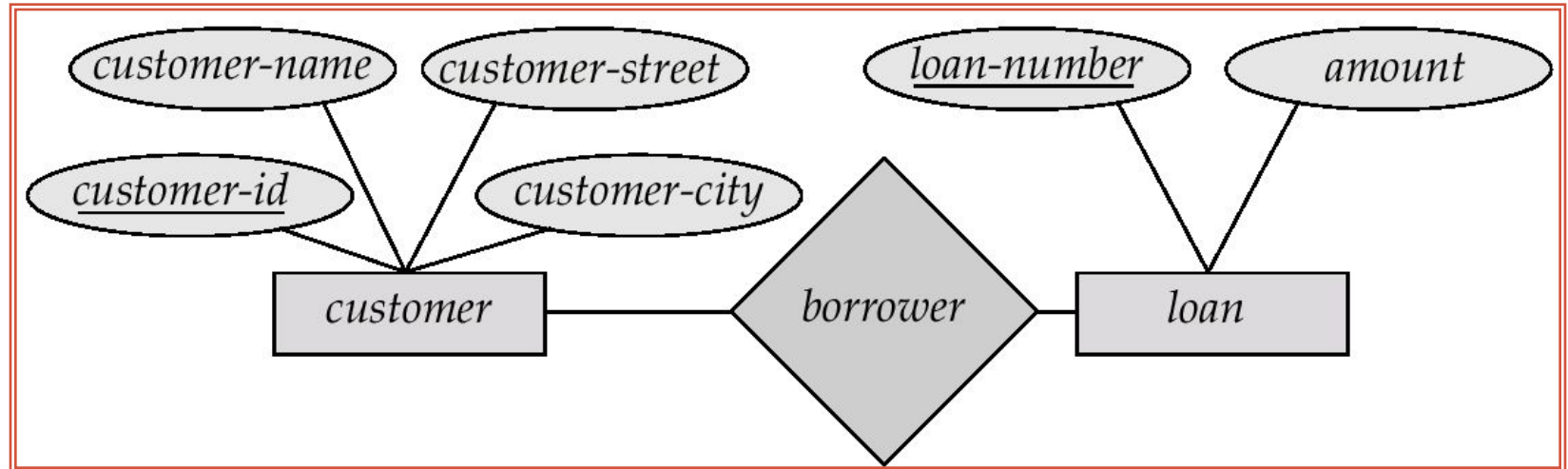
Many-To-One Relationships

- In a many-to-one relationship a loan is associated with several (including 0) customers via *borrower*, a customer is associated with at most one loan via *borrower*



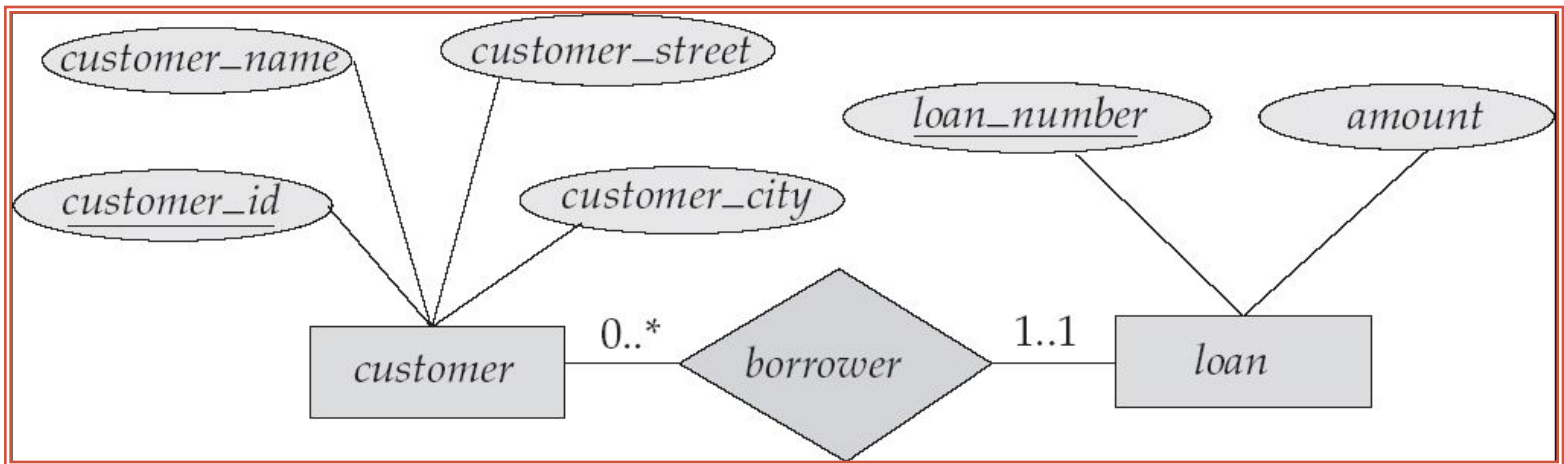
Many-To-Many Relationship

- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower



Alternative Notation for Cardinality Limits

- Cardinality limits can also express participation constraints



Case Study

- For the following Entities, Identify Primark Key, Identify relationship, cardinality and Draw ER Diagram
- Case 1:
 - Department (DeptId, Name, YearOfStart, Location)
 - Student (Regno, Name, DoB, Address, Branch, Batch)
 - Courses (DeptId, CourseCode, Title, EvalPattern)
- Case 2:
 - City (CityId, CityName, Remarks)
 - Product (ProdCode, ProdName, Stock, Price, CityId)
 - Customer (CustomerId, CustName, Address, OrderId)
 - Orders (OrderId, ProdCode, Quantity)
 - Bills (OrderId, Cost, Tax, ShipCharges, Total, Remarks)

Reference Key or Foreign Key

- A **Foreign key** (or reference key) is a key borrowed from another related table (that's why its foreign) in order to make the relationship between two entities
- A key to connect two entities – Allows duplicate values
- Parent Entity – An entity that does not depend for its existence on other entities – Has a Primary Key (may or may not have foreign keys)
- Child Entity (referential entity) – An entity that depends for its existence on other entities – Should have a reference (foreign) key

Identify all keys in the following, Draw ER Diagram

- Case 1:
 - Department (**DeptId**, Name, YearOfStart, Location) - No Foreign Key
 - Student (**Regno**, Name, DoB, Address, Branch, Batch)
 - Courses(**DeptId**, **CourseCode**, Title, EvalPattern)
 - Grades(**Regno**, CourseCode, Grade)
- Case 2: -- Do this now.
 - City(CityId, CityName, Remarks)
 - Product(ProdCode, ProdName, Stock, Price, CityId)
 - Customer(CustomerId, CustName, Address, OrderId)
 - Orders(OrderId, ProdCode, Quantity)
 - Bills(OrderId, Cost, Tax, ShipCharges, Total, Remarks)