CIS 8040

Introduction to Conceptual Modeling Using the Entity-Relationship Model (Chen’s Model)

Important Concepts

The Nature of Data [Revisited]

- Data asset – know what this refers to
- Different types of useful data; lots of "non-useful" data

Conceptual modeling

- What is it?
- Where it fits in the entire database design and development process. [After requirements collection/analysis]

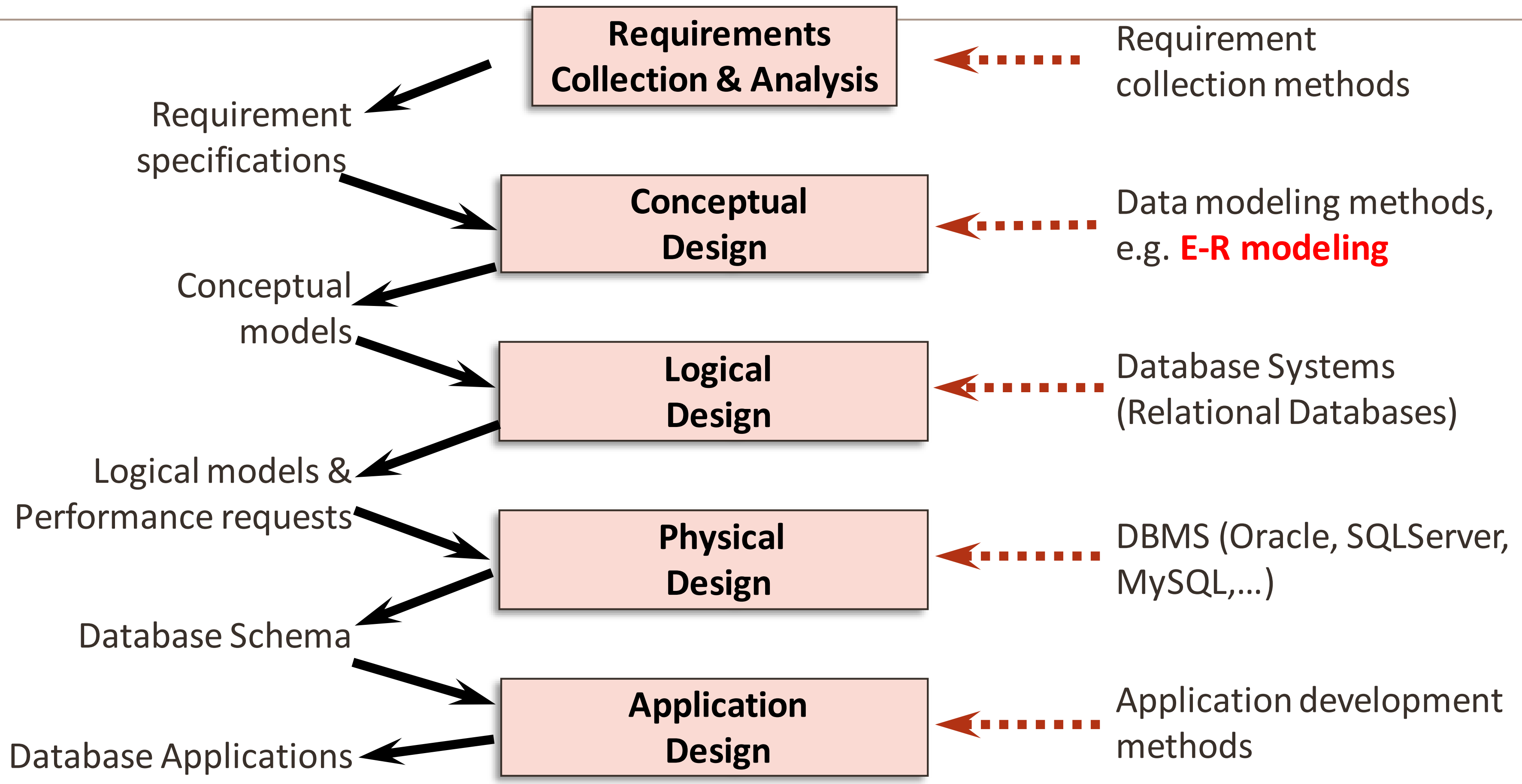
Relational databases [This is what we will implement in this course]

- Understand what they are (generally, set of tables, called relations)
- Where they fit in the entire database design and development process. [After conceptual modeling]

Terminology (Keep these handy. The list only grows …..)

- Data modeling, data models, entity, relationship, attribute, abstraction, instance, instantiation, occurrence, entity type, relationship type, association, mapping ratios, identifiers, key (primary, composite, candidate), mapping ratios, min/max cardinalities, application domain

Database design methodology



Requirements Collection & Analysis

Requirement collection methods

Requirement specifications

Conceptual Design

Data modeling methods, e.g. **E-R modeling**

Conceptual models

Logical Design

Database Systems (Relational Databases)

Logical models & Performance requests

Physical Design

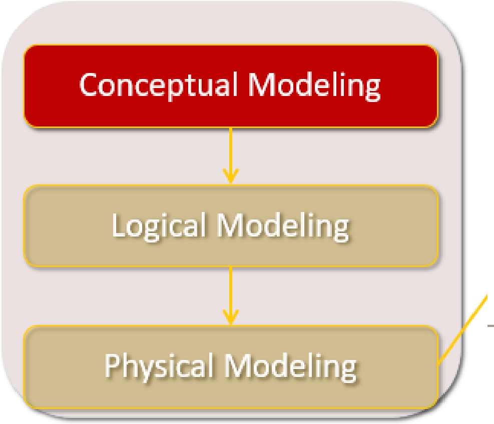
DBMS (Oracle, SQLServer, MySQL,…)

Database Schema

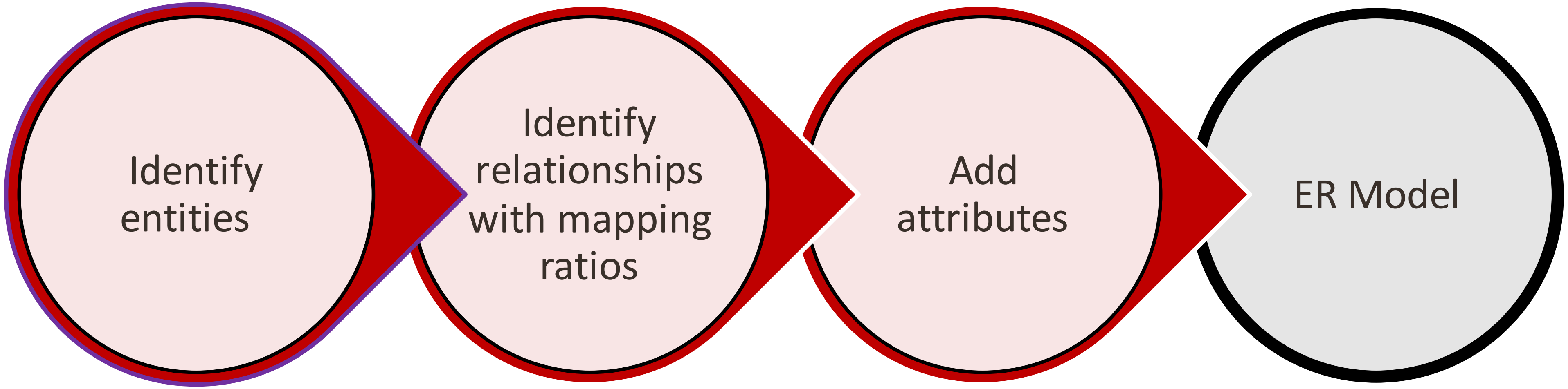
Application Design

Application development methods

Database Applications

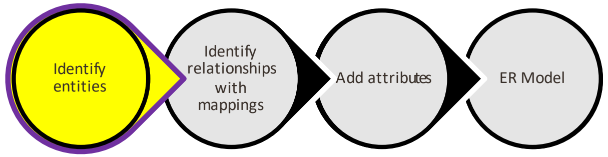


Conceptual Modeling: Creation of ER (Entity-Relationship) Model



Important terminology:

- Mapping ratios [For today]
- Min/max cardinalities [Coming – finer level of granularity]

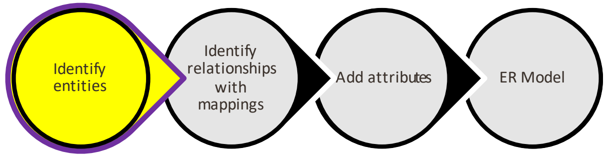


Entity: a “thing” of interest about which you want to record data in a database.

- Person – employee, student, patient, professor
- Place – store, university, warehouse
- Object – machine, vehicle
- Event – sale, registration
- Concept – course, account

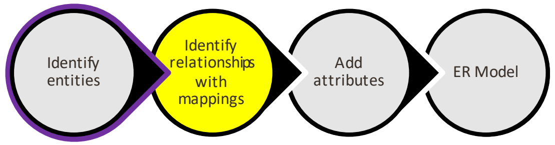
What do these have in common?
What are the implications for database design?
Note: Can be concrete or conceptual.

Source: Hoffer et al.



Entities

- Suppose you want to design a database for managing football recruiting and games at Georgia State University.
- Identify 2-3 reasonable entities to include in your database.


Identify entities

Identify relationships with mappings

Add attributes

ER Model

Relationship: association between [among] entities

Examples (of binary relationship):

STUDENT *takes* CLASS

PROFESSOR *teaches* CLASS

DEPARTMENT *employs* PROFESSOR

DIVISION *managed by* EMPLOYEE

AIRCRAFT *flown by* CREW

Questions: What do these have in common?

Why would you want to represent these in a database?

How would you represent these in a database? [Later …]

Relationship: operates in both directions

Concept of mappings (reflects real world situation).

Mapping ratios (M:N (or N:M), 1:N (or 1:M), 1:1)

Capture / represent some business rules.

STUDENT *takes* CLASS
M N
PROFESSOR *teaches* CLASS
1 N
DEPARTMENT *employs* PROFESSOR
1 N
DIVISION *managed by* EMPLOYEE
N 1
AIRCRAFT *flown by* CREW
N 1

Note: 1:N versus N:1
N:M versus M:N
1:1
Important to be correct
(influences organization of a database)

Relationships: Mappings

Customer buys Car

Chef prepares Menu-Items

Team has Player

Relationships: Mappings

QUESTION: WHAT ARE YOU MODELING?

Customer buys Car

N M

Chef prepares Menu-Items

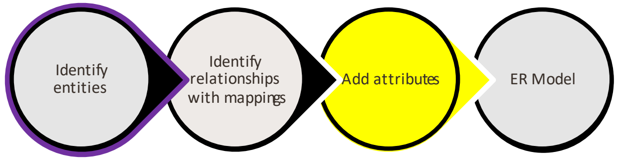
1 N

Chef prepares Menu-Items

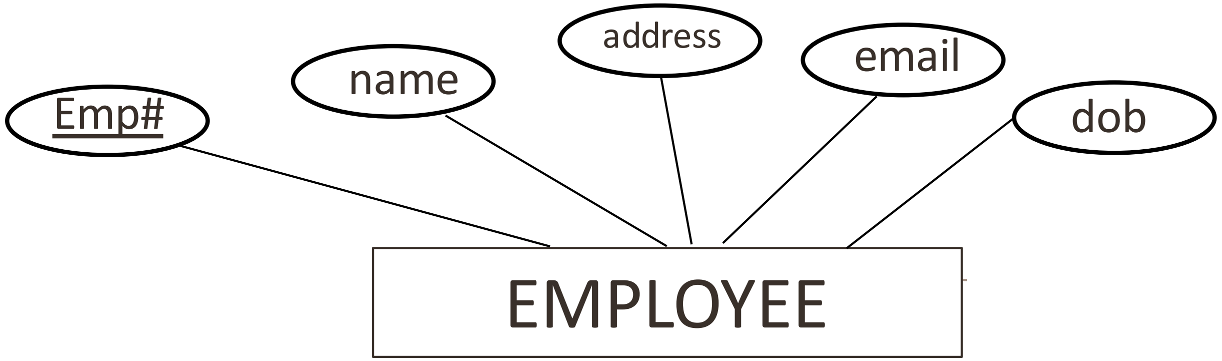
N M

Team has Player

1 N



Entity and its attributes



Employee: (Emp#, name, address, email, dob)

Attributes are characteristics of entities that are important to include in a database.

Attributes

Domain

- Important concept for attributes
- Implications for physical design (implementation) where you must specify data type (later)

Domain -- possible values for a given attribute

Domain for grade point average (GPA) [0,4.3]

Domain for answer – T or F (or some other equivalent code).

Attributes may share a domain.

E.g.: student address and professor address share the same
domain of all possible addresses.

NOTE: Also the concept of "application domain."

Entity Student

What are reasonable attributes?

Can you justify them?

Entity and its Attributes

Student: (StudentID, name, address, major, full-time, …)

Context: StudentID at GSU is a Panther#

What other data?

How much do we collect and store? Why?

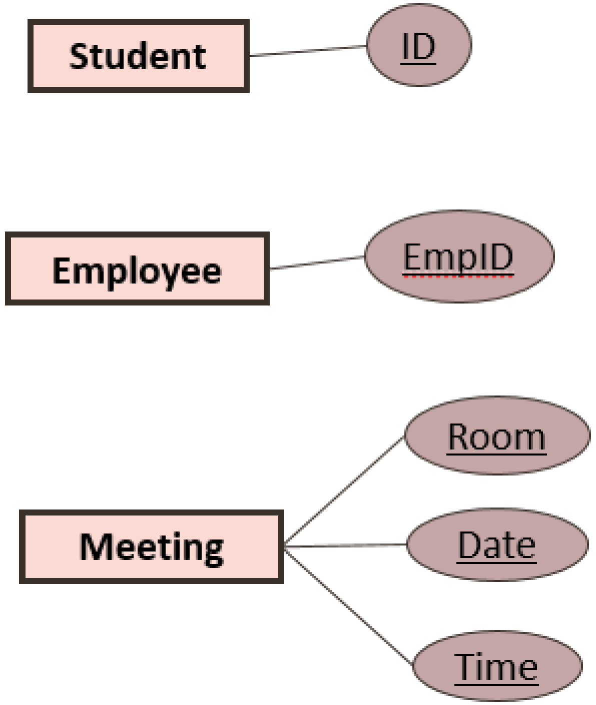
Each entity type must have unique identifier (key)

An entity type may have several qualified keys

- Candidate keys
- Single key or composite key

Primary key – One of the candidate keys

Primary key attribute(s) is(are) underlined in an entity-relationship diagram



Definition:
Identifier -- one or more attributes that uniquely identify each entity instance

Attribute: single valued, multi-valued, derived

Single-valued attribute (best)

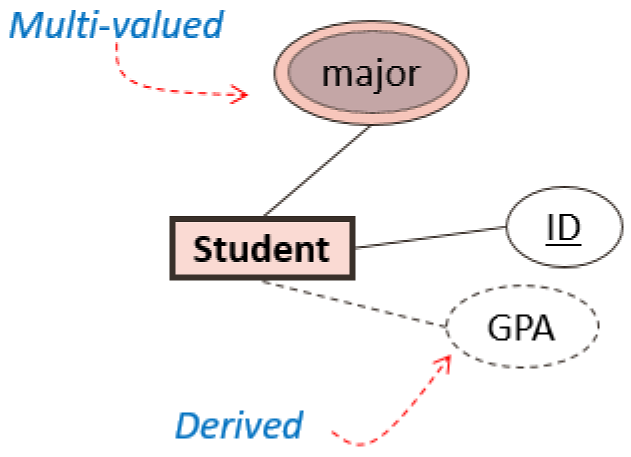
- Has exactly one value per entity
- Represented as a single lined oval

Multi-valued attribute (best not to use)

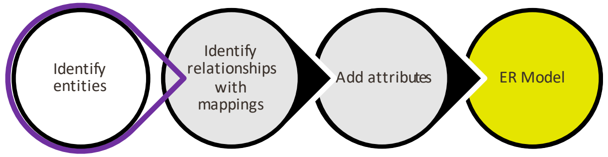
- Contains repeating values per entity
- Represented as a double lined oval

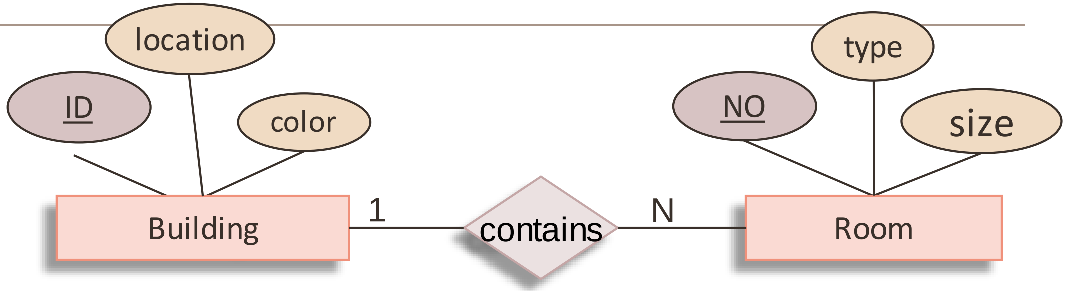
Derived attribute (best in application)

- Calculated based on other attributes
- Represented as a dotted line and a dotted line oval



Note: terminology / representation can differ depending on modeling approach used.
This is an entity-relationship model representation.
Multi-values attributes – need to be modeled separate entities. Do not include in your final model.
Derived – obtained in application. Why? Do not include on your final model.





Simple Conceptual Model for Buildings

Interpretation: A building has 1 to many rooms.

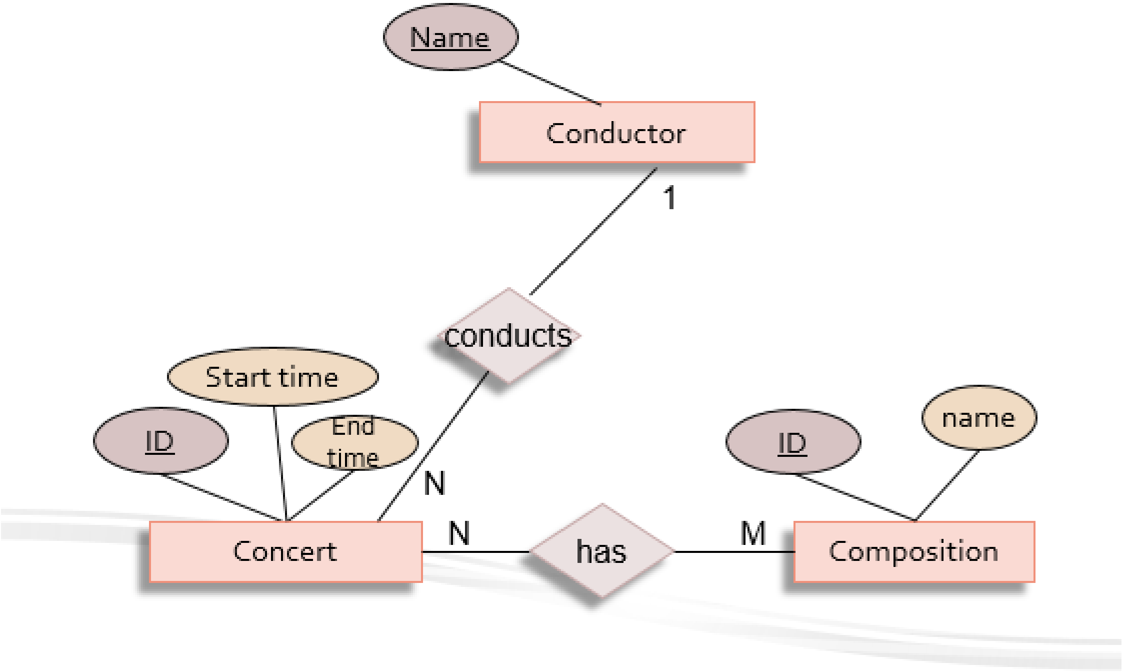
Note: key and non-key attributes.

Example: Symphony Orchestra

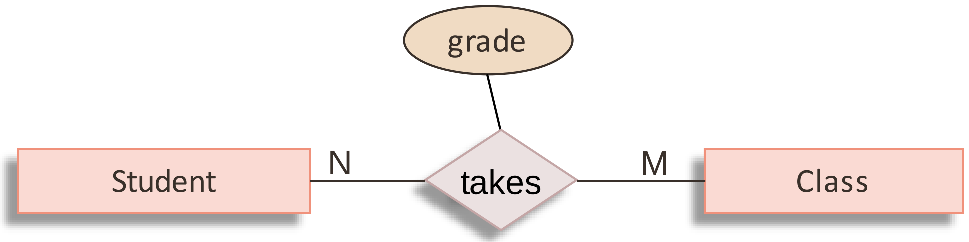
You want to develop a database for a symphony orchestra. A concert is conducted by one conductor and includes the performance of one or more compositions.

Example: Symphony Orchestra

You want to develop a database for a symphony orchestra. A concert is conducted by one conductor and includes the performance of one or more compositions.



Relationship attributes



Interpretation?

Note: Only many-to-many relationships can have relationship attributes.

Some exceptions (but not for this course).

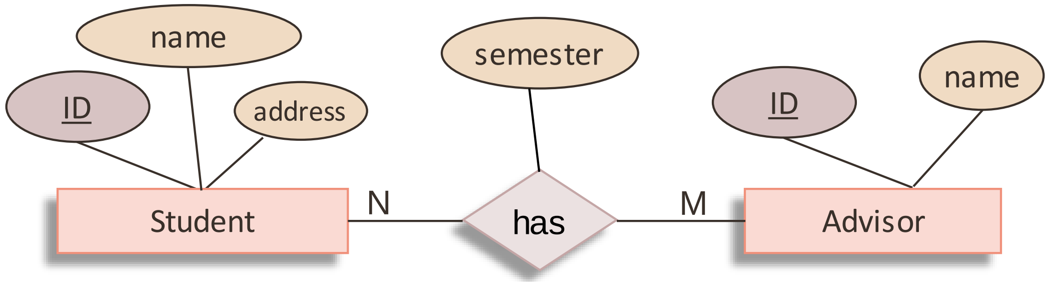
Example: Student-Advisor

Each semester, every student is assigned an advisor who helps the student understand degree requirements and register for classes. The advisor can change depending upon the semester. A database is needed to keep track of the students and their advisors for each semester.

Draw the conceptual model (entity-relationship model) for this application.

Identify the entities, relationship, mappings (mapping ratios), attributes

Example: Student-Advisor



What are the inherent assumptions in modeling the application in this way?

Example: Roger’s Catering Service

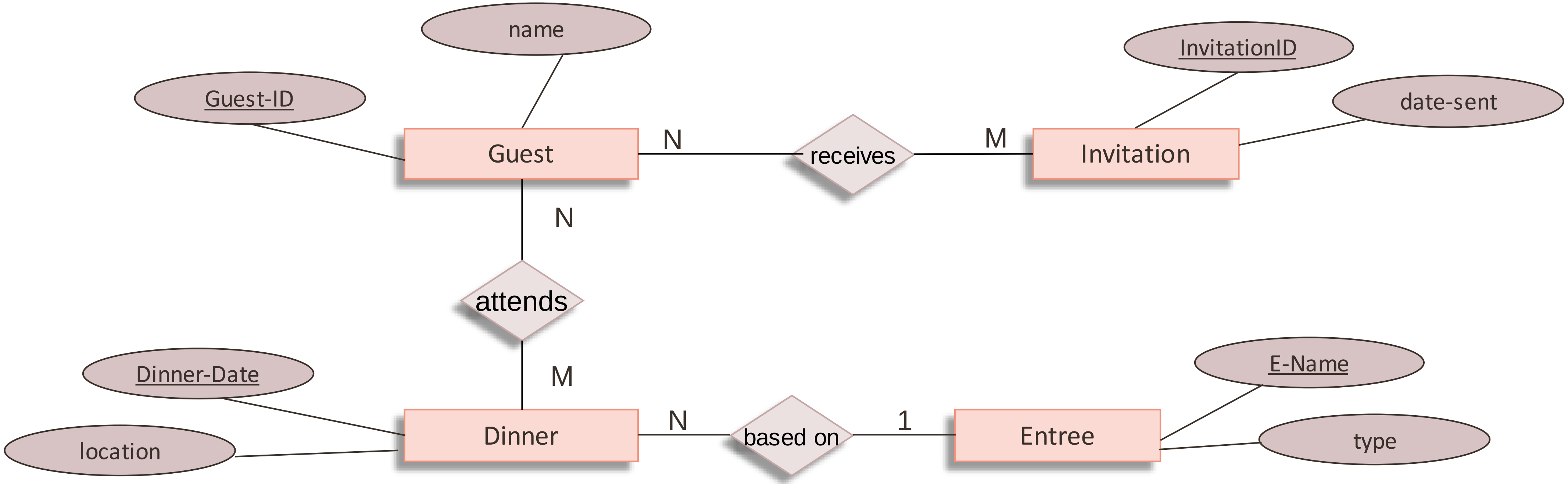
Each dinner is based on a single entree, but each entree can be served at many dinners. A guest can attend many dinners. Each dinner can be attended by many guests. Each dinner invitation can be mailed to many guests, and each guest can receive many invitations.

Develop a conceptual model for this application.

- Identify entities
- Identify relationships
- Identify attributes (based on your own knowledge)

Example: Roger’s Catering Service

Each dinner is based on a single entree, but each entree can be served at many dinners. A guest can attend many dinners. Each dinner can be attended by many guests. Each dinner invitation can be mailed to many guests, and each guest can receive many invitations.



Conclusion

Conceptual modeling – important phase of database design

Entity-relationship model

- Entities
- Attributes
- Relationships

Important notion of keys

- Entities have keys. Must have primary keys (may be composite).
- Relationships are associations among entities but do not, by themselves, have keys.

Examples

- Capture essence of what is important in the real world
  - (e.g., attributes to include in a student database)
- Important to identify correct mapping ratio. Why?