Database Principles: Fundamentals of Design, Implementation, and Management

Tenth Edition

Chapter 7

Data Modeling with Entity Relationship Diagrams (Cont.)

Objectives

- In this chapter, students will learn:
  - The main characteristics of weak entities and strong entities
  - How to relate weak entity to its strong entity
  - Rules about identifying relationships
  - The main characteristics of inheritance modelling
  - Constraints of inheritance relationship

Weak Entity Sets

A **weak entity** is an entity that cannot exist in a database unless another type of entity also exists in that database.

Weak entity meets two conditions

- **Existence-dependent**
  Cannot exist without entity with which it has a relationship
- **Has primary key that is partially or totally derived from parent entity in relationship**
  Weak entities cannot exist without the identifying relationship.

Weak entities do not have **primary key** attribute(s) of their own. They may have partial key.

Partial Key:

Portion of the key that comes from the weak entity. The rest of the key comes from the other entity in the relationship.

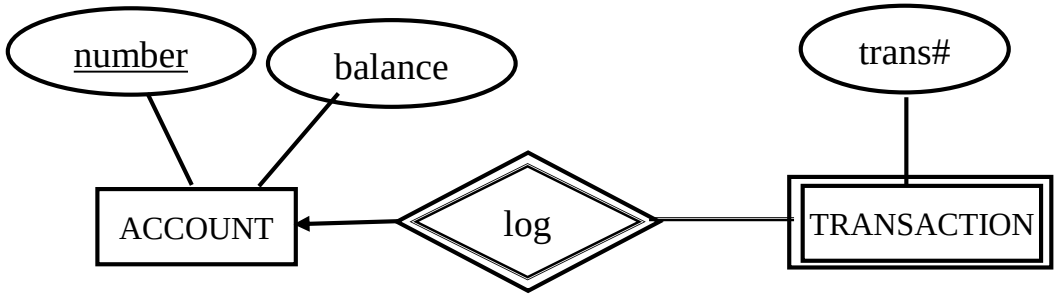
Weak Entity Sets (cont.)

- ✓ Weak Entity is represented by double rectangle
- ✓ Weak Relationship is represented by double diamonds
- ✓ Many-one (or one-one) relationship sets are required

Weak Entity Sets (cont.)

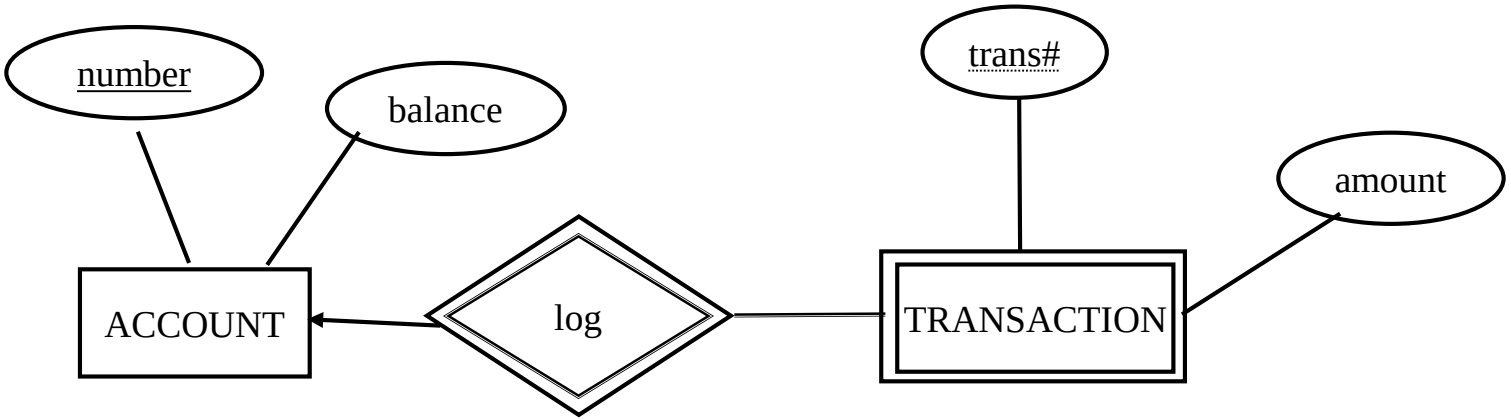
Example:

- Transactions of different accounts could have the same trans#, so "trans#" cannot be a key
- By borrowing attribute "number" from "account," we have a key for "transaction."
- "Transaction" is a weak entity set related to accounts via log relationship.

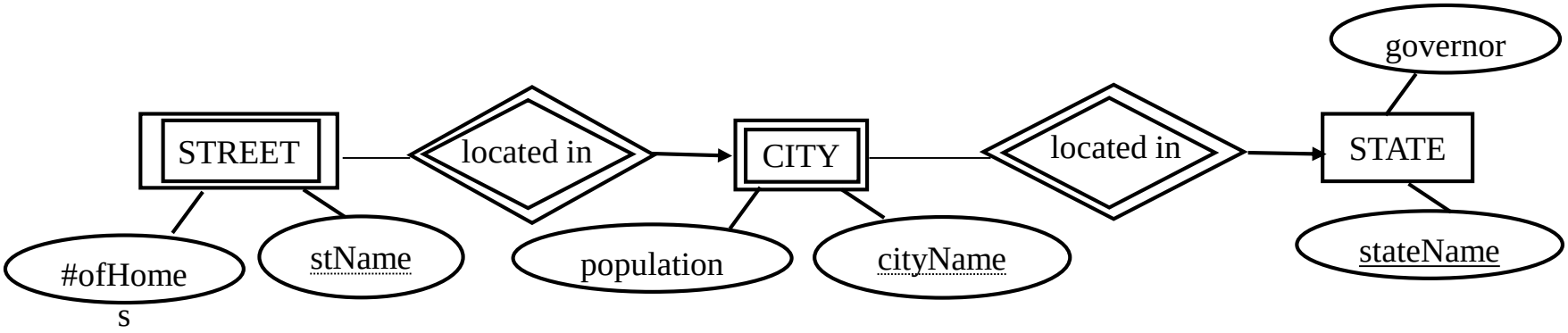


Weak Entity Sets (cont.)

- Key of weak entity set = key of strong entity set(s) + partial key



Chain of Weak Entity Sets

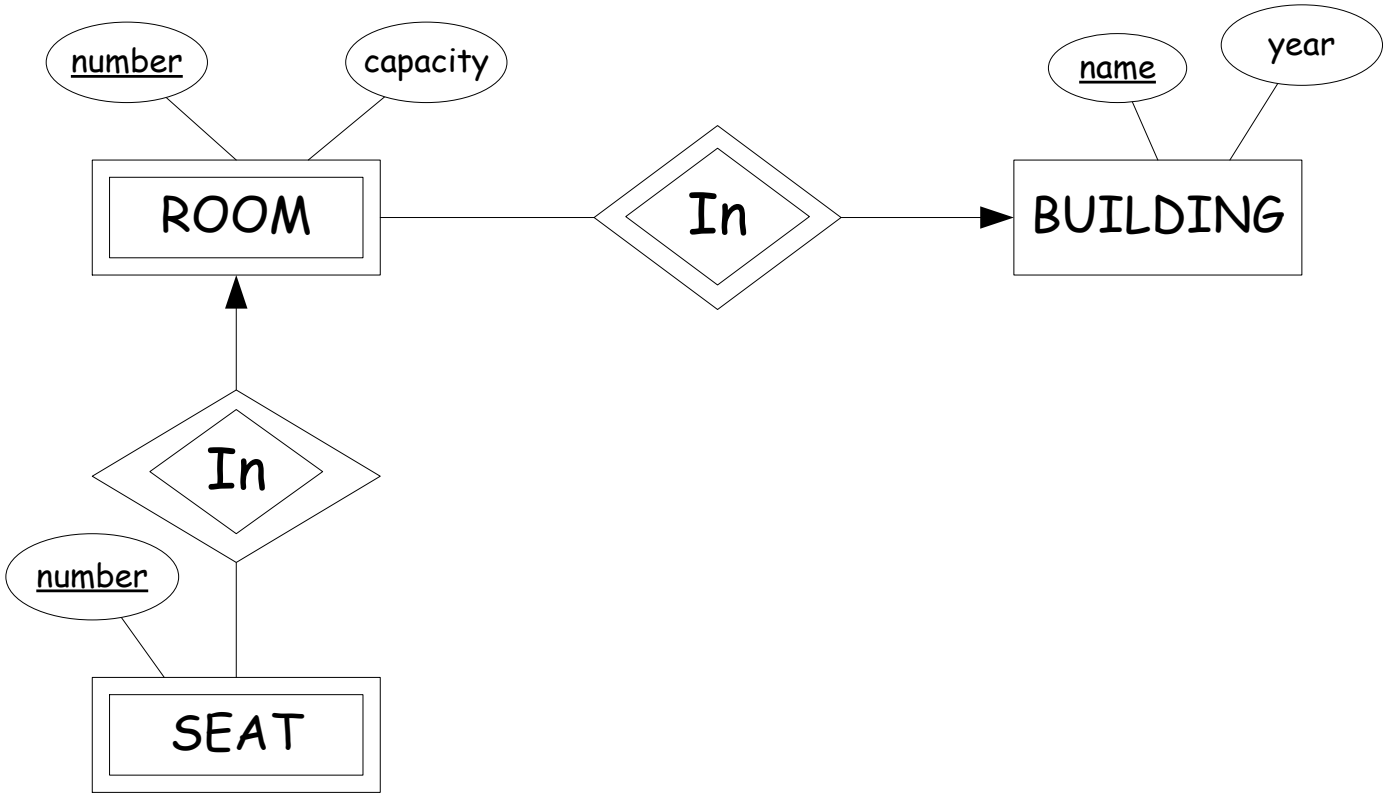


- State name forms a key.
- City names are unique only within a state (e.g., 24 Springfield's within the 50 states).
- Street names are unique within a city. Multiple cities could have streets with the same name (e.g., "Main").
- A weak entity set might itself participate as owner in an identifying relationship with another weak entity set.

EXAMPLE

Weak entity set examples!

Seats in rooms in buildings



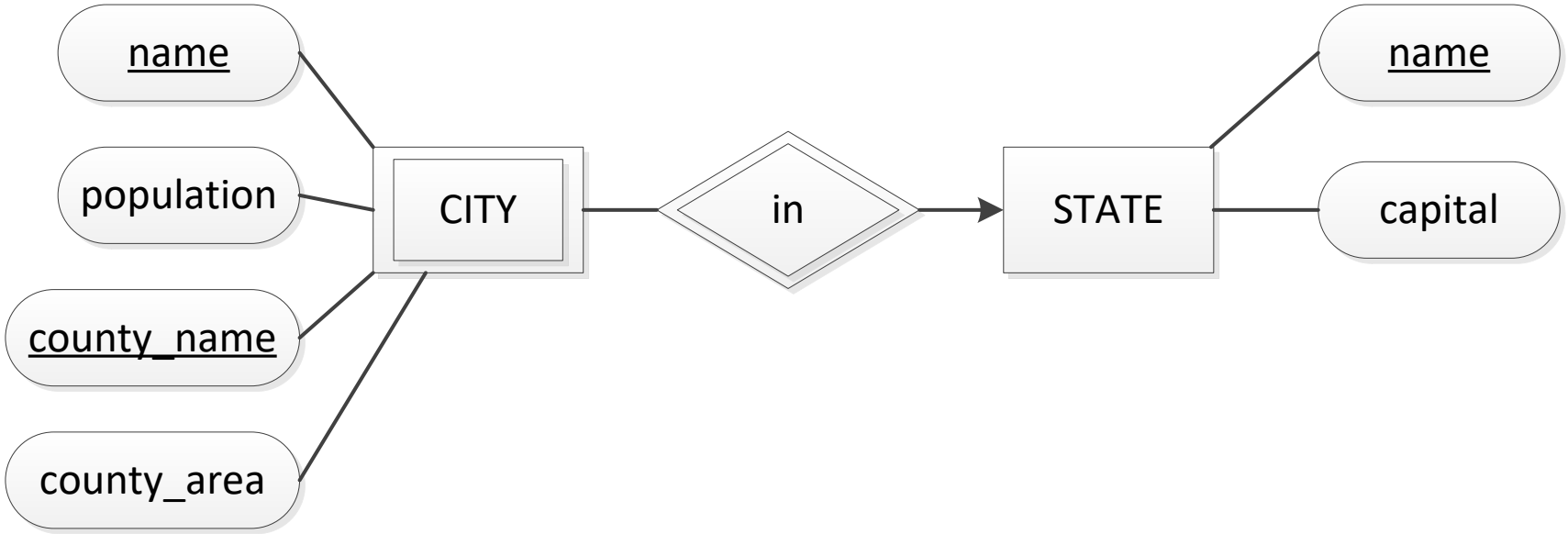
CASE STUDY

Design a database representing cities, counties, and states

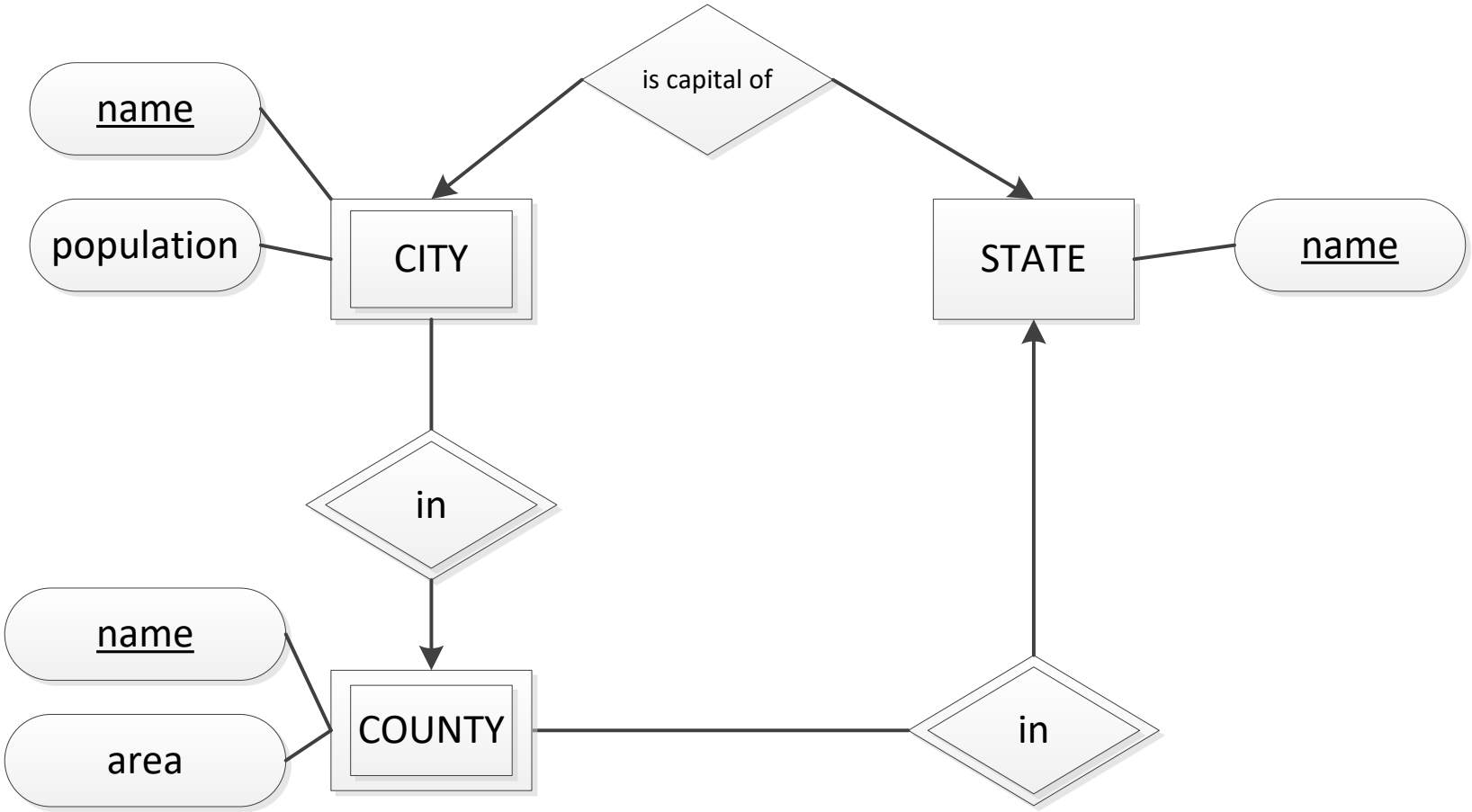
- For states, record name and capital (city)
- For counties, record name, area, and location (state)
- For cities, record name, population, and location (county and state)

- Assume the following:
  - Names of states are unique
  - Names of counties are only unique within a state
  - Names of cities are only unique within a county
  - A city is always located in a single county
  - A county is always located in a single state

DESIGN 1


name
population
county_name
county_area
CITY
in
STATE
name
capital

DESIGN 2


is capital of
name
population
CITY
STATE
name
in
name
area
COUNTY
in

Practice 1

- The company organizes many different types of parties for its customers. For all parties, the company assigns a unique id, a party date, start time, finish time and duration. Duration is calculated as the number of hours between start-time and finish-time. Each party is hosted (owned) by exactly one customer. A customer may host many parties. Each customer has a name, an address, a phone number and a unique id. There are many employees working at the company. Each employee has a name, salary and a unique id. A number of meetings are arranged for each party during the organization. Each meeting belongs to exactly one party. For each meeting a meeting date, a comment and a meeting number is recorded. Each meeting is identified by using the meeting number and party id together. Each meeting may contain one or more employees. An employee may be included in zero or more meetings.

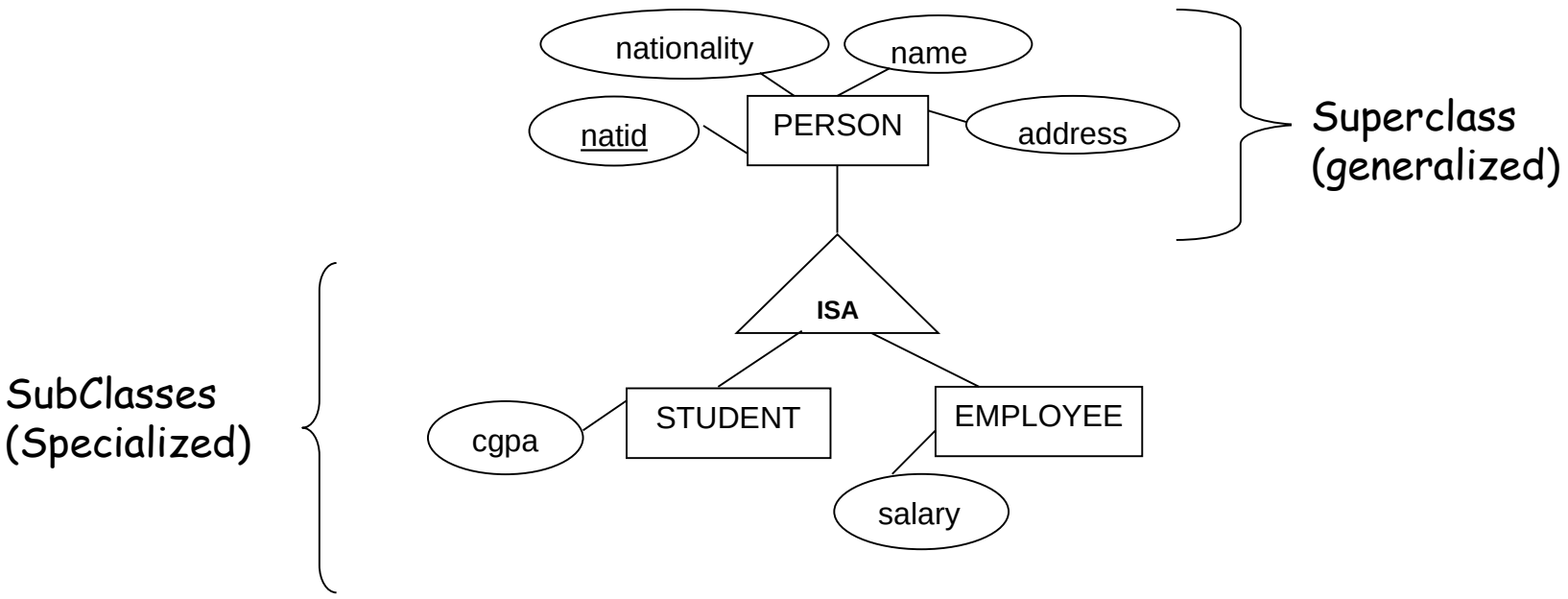
Practice 2

A college course is taught by many instructors, and an instructor may teach many courses. A course may have one or more scheduled sections, or may not have a scheduled section. Attributes of COURSE include courseID, coursename, and credits. Courses are identified by courseID. Attribute of a SECTION of a course include courseID, sectionID, semesternumber, year, and instructorID. A Section is belong to only one course. Sections are identified by courseID and sectionID together. Only one instructor is responsible for a given section of a course. An INSTRUCTOR is identified by an instructorID. Additional information we want to store about an instructor includes the lastname, firstname, email address and officenumber.

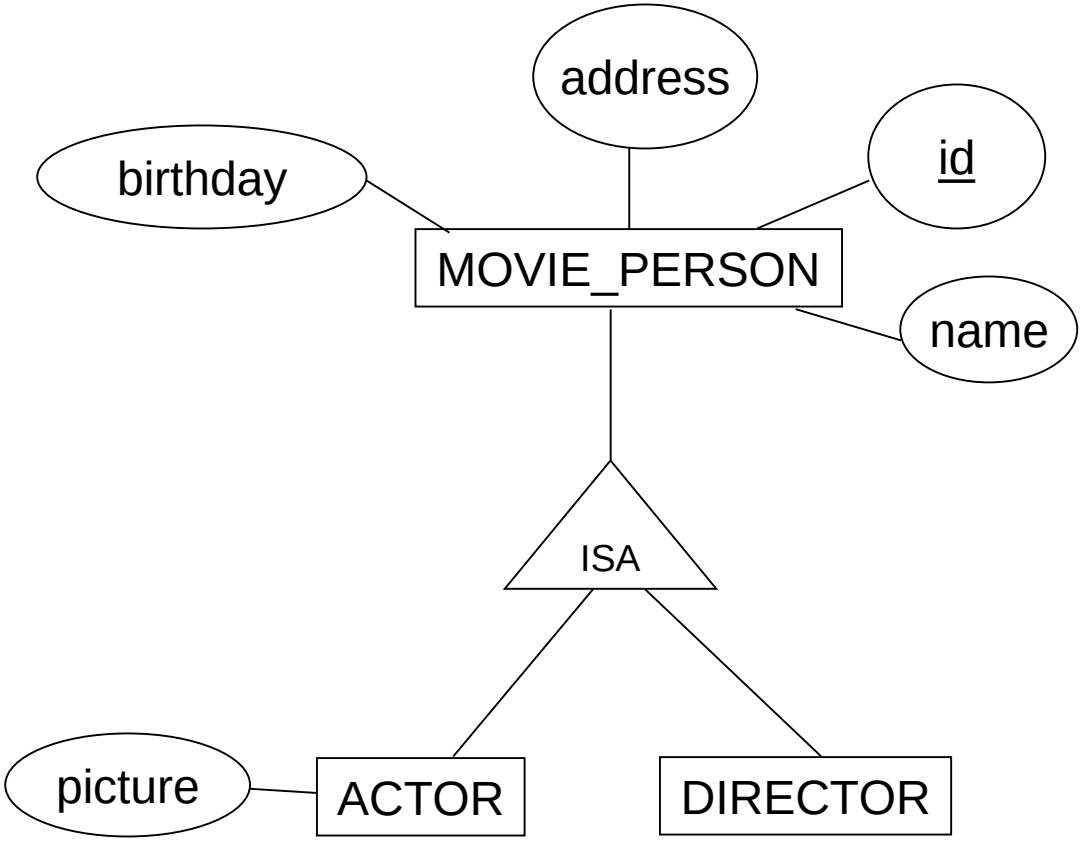
INHERITANCE MODELLING

ISA relationship METHOD I

In some problems, we see entities that are types of or specializations/generalized of other entities. For these situations we use inheritance relationship. (Also known as ISA relationship).



EXAMPLE



What are the keys of:

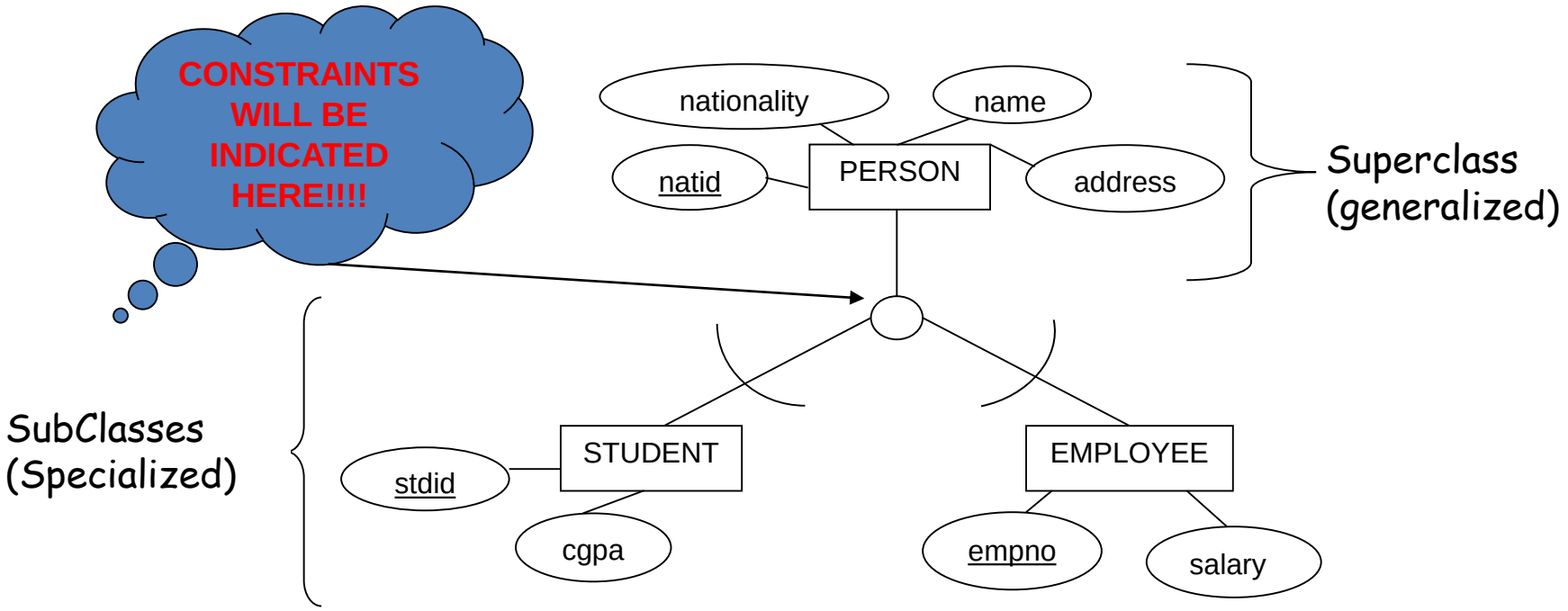
1. Movie Person
2. Actor
3. Director

INHERITANCE MODELLING

ISA relationsip

METHOD II

We might have Constraints for the inheritances such as Disjoint or Overlap.



Specialization & Generalization

- **Specialization**
  - process of taking an entity and creating several *specialized* subclasses

- **Generalization**
  - process of taking several related entities and creating a *general* superclass

Specialization constraints

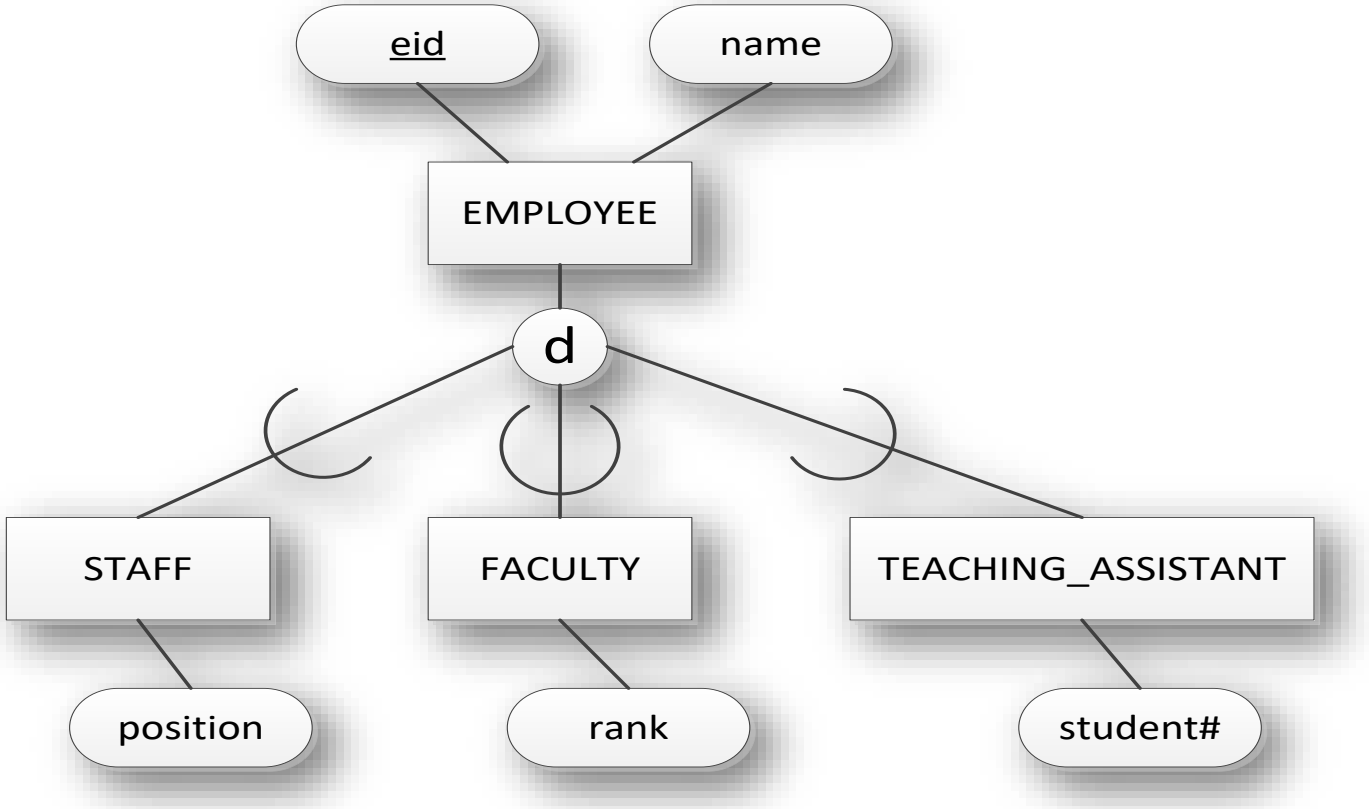
- ***Disjointness constraint*** – specialization is disjoint or overlapping

- ***Completeness constraint*** – specialization is total or partial

Disjointness constraint

- Specifies that an entity can be a member of **at most** one subclass
- There can be no overlap between the subclasses
- We use the notation of a **d** in a circle to symbolize that the subclasses are disjoint

Disjoint constraint

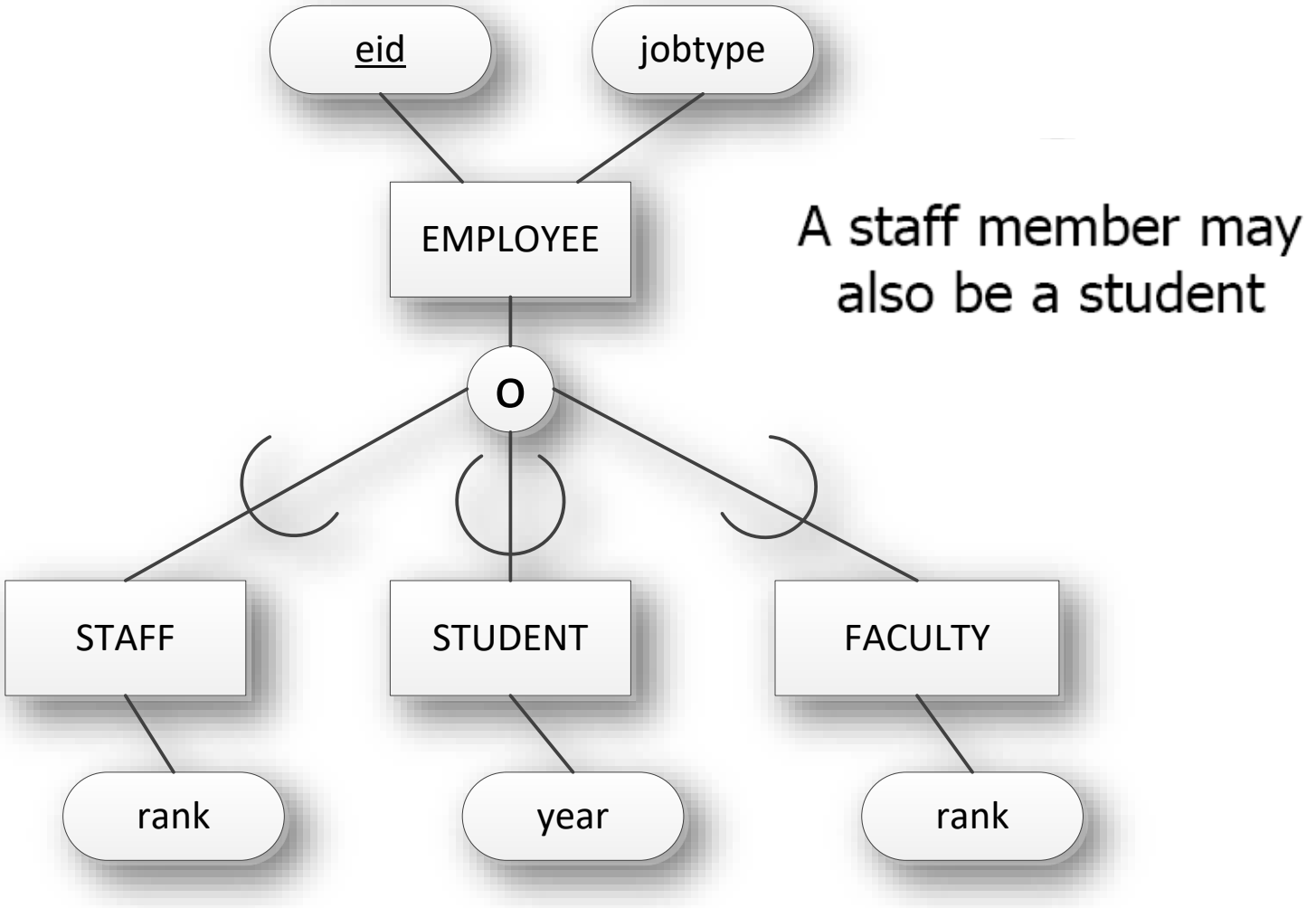


Overlap

- Entities are able to belong to more than one subclass

- Notation is an **o** inside of a circle

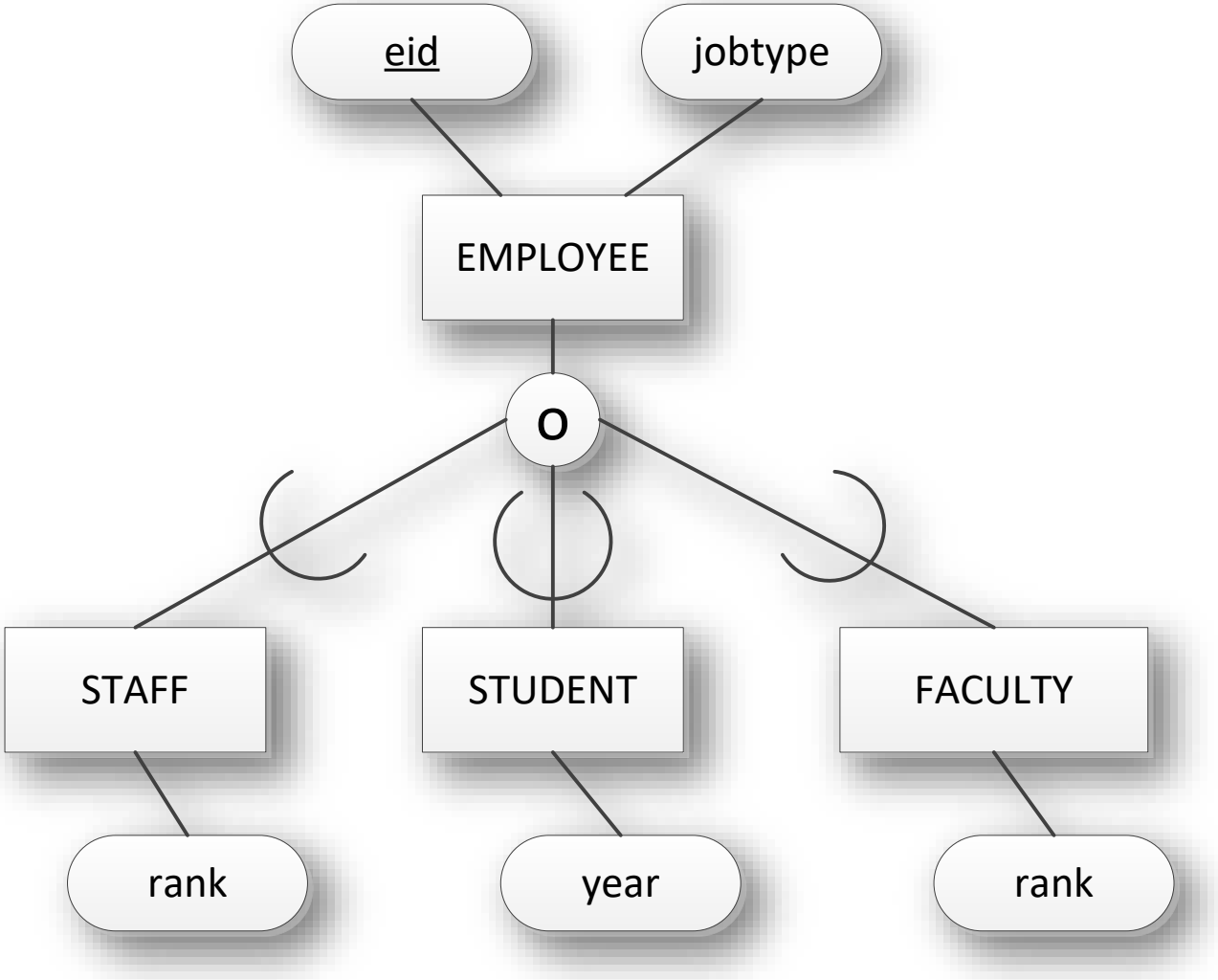
Overlap constraint



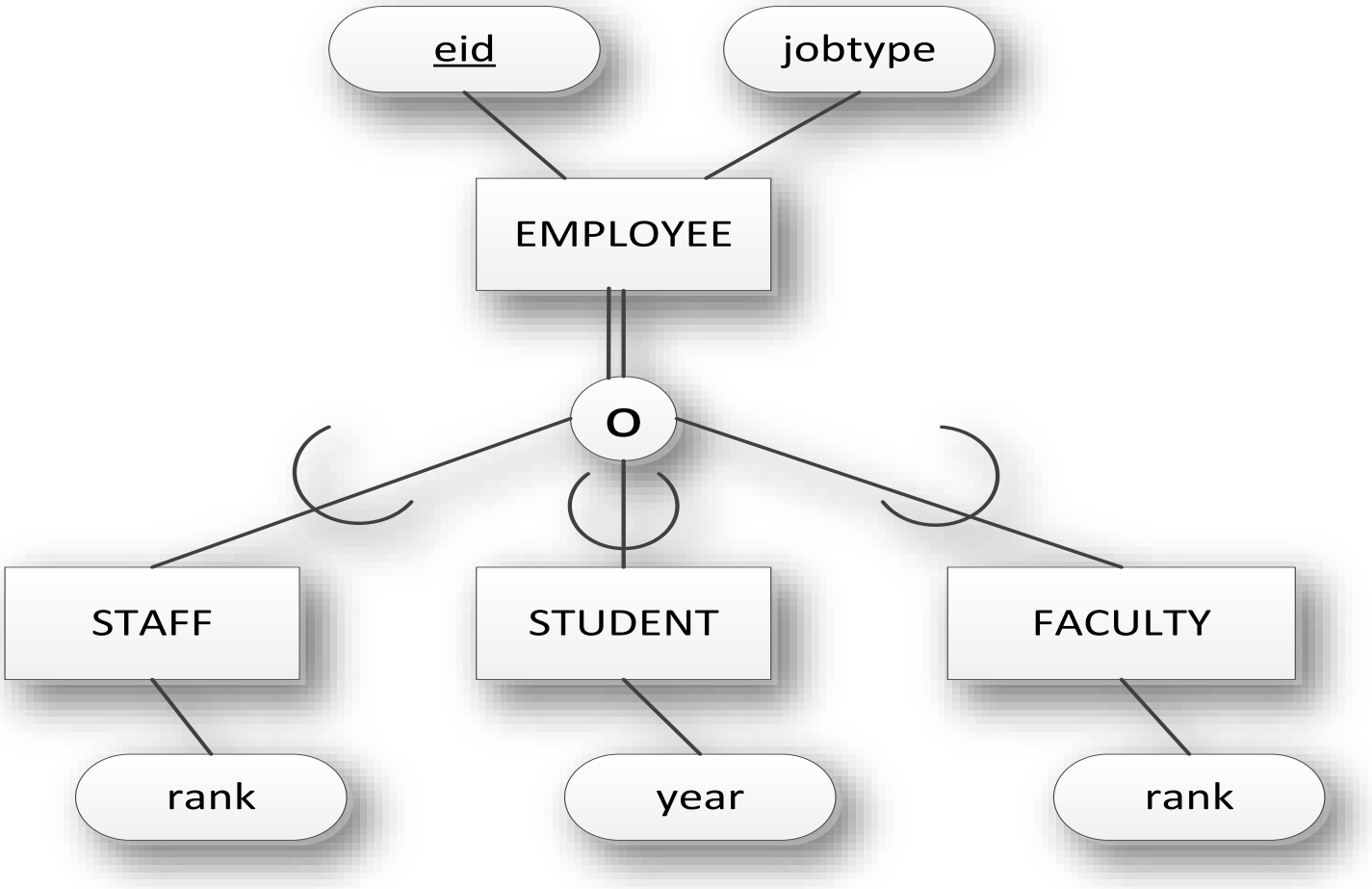
Completeness Constraint

- May be *total* or *partial*
- For *total*, every entity in the superclass **must** belong to a subclass
- For *partial*, entities in the superclass do not need to be part of any subclass
- Notation for *total* and *partial* are the same as in a regular E-R diagram – single and double lines

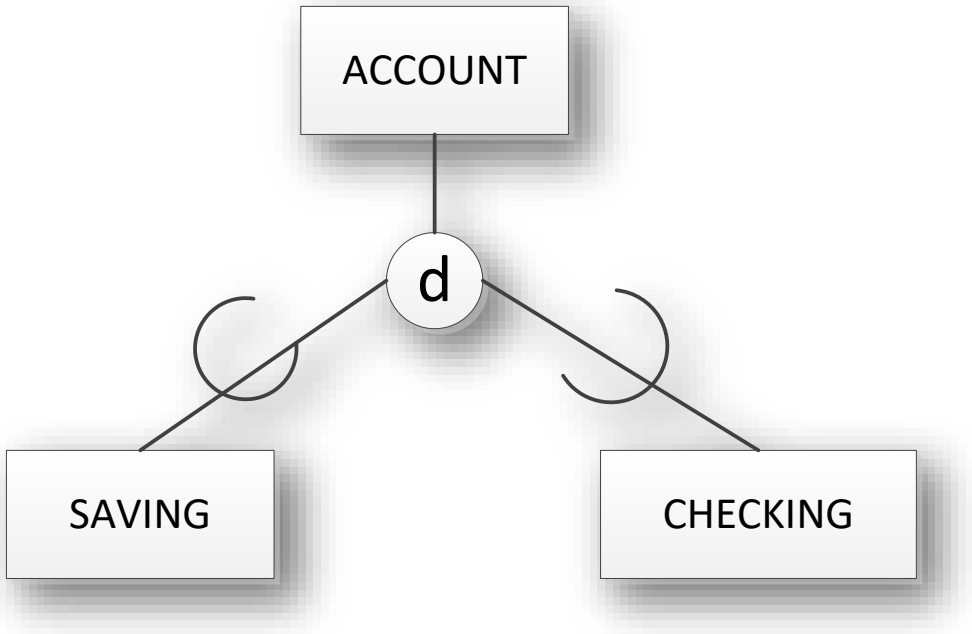
Partial


eid
jobtype
EMPLOYEE
O
STAFF
STUDENT
FACULTY
rank
year
rank

Total

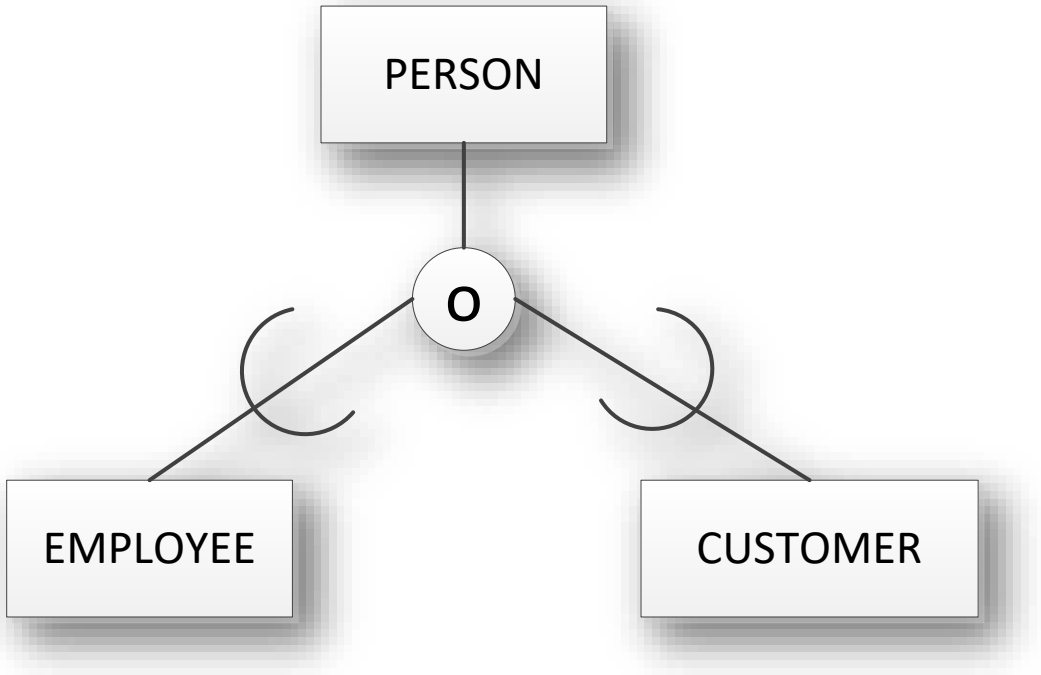

eid
jobtype
EMPLOYEE
O
STAFF
STUDENT
FACULTY
rank
year
rank

Examples



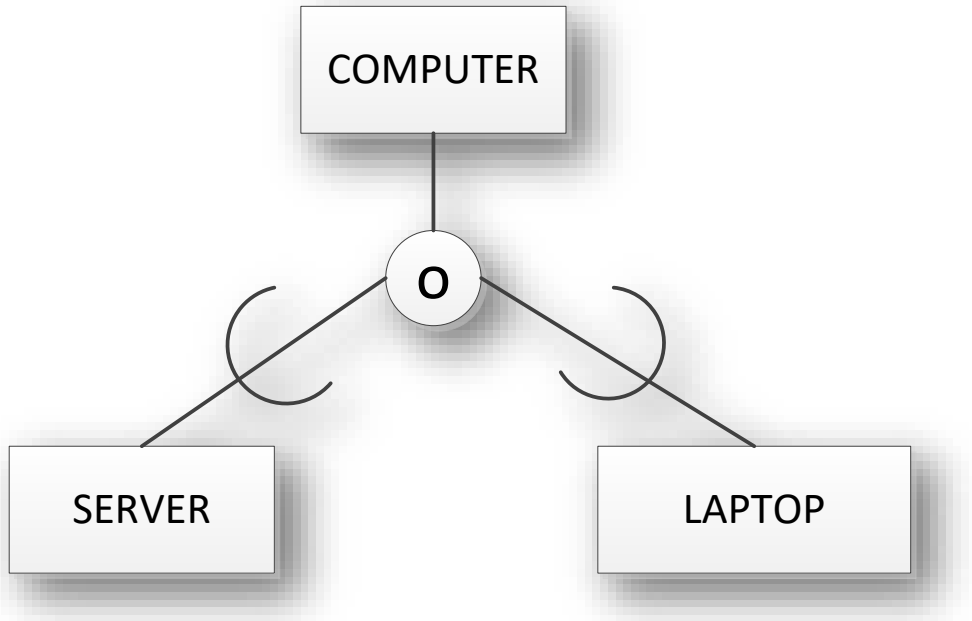
An Entity can be either a savings account or a checking account, but cannot be both.

Examples



A Person can be both an Employee and a Customer.

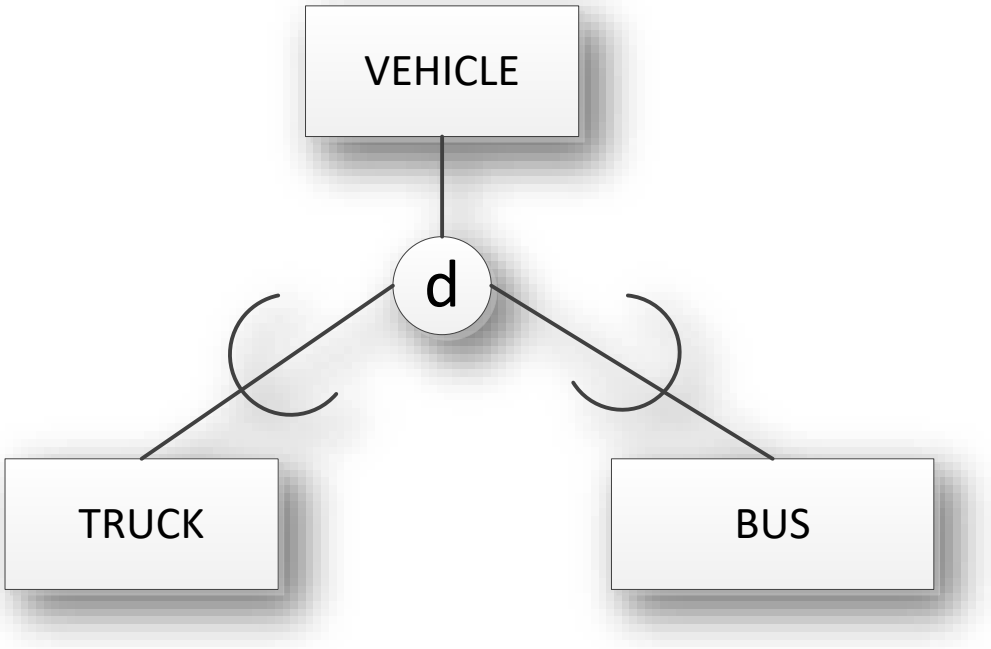
Examples



Is it Partial? If yes why?

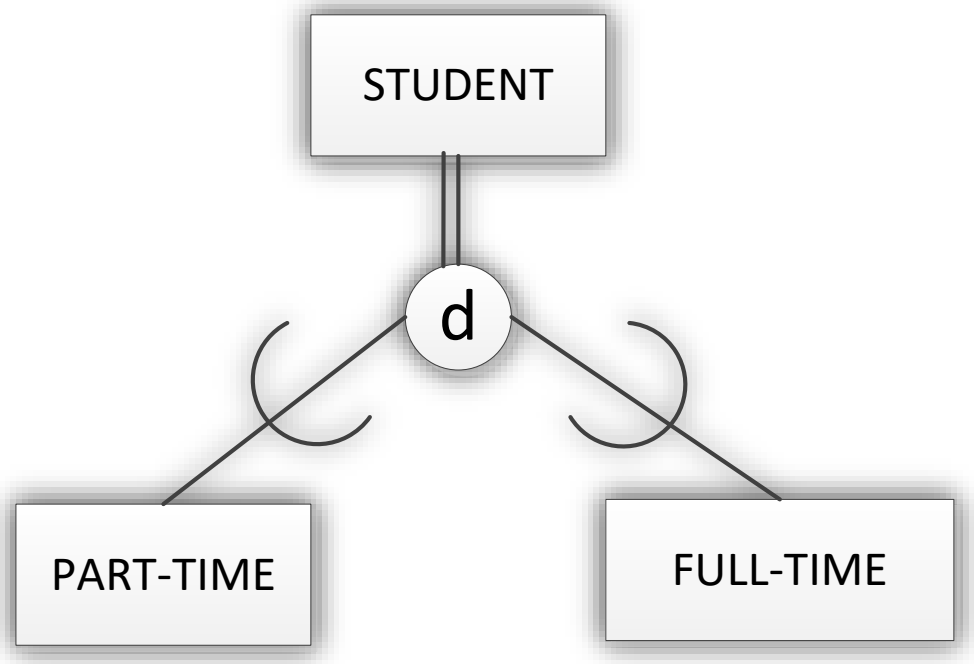
Yes it is Partial. Because, there are some computers that they are neither Server nor Palmtop.

Examples



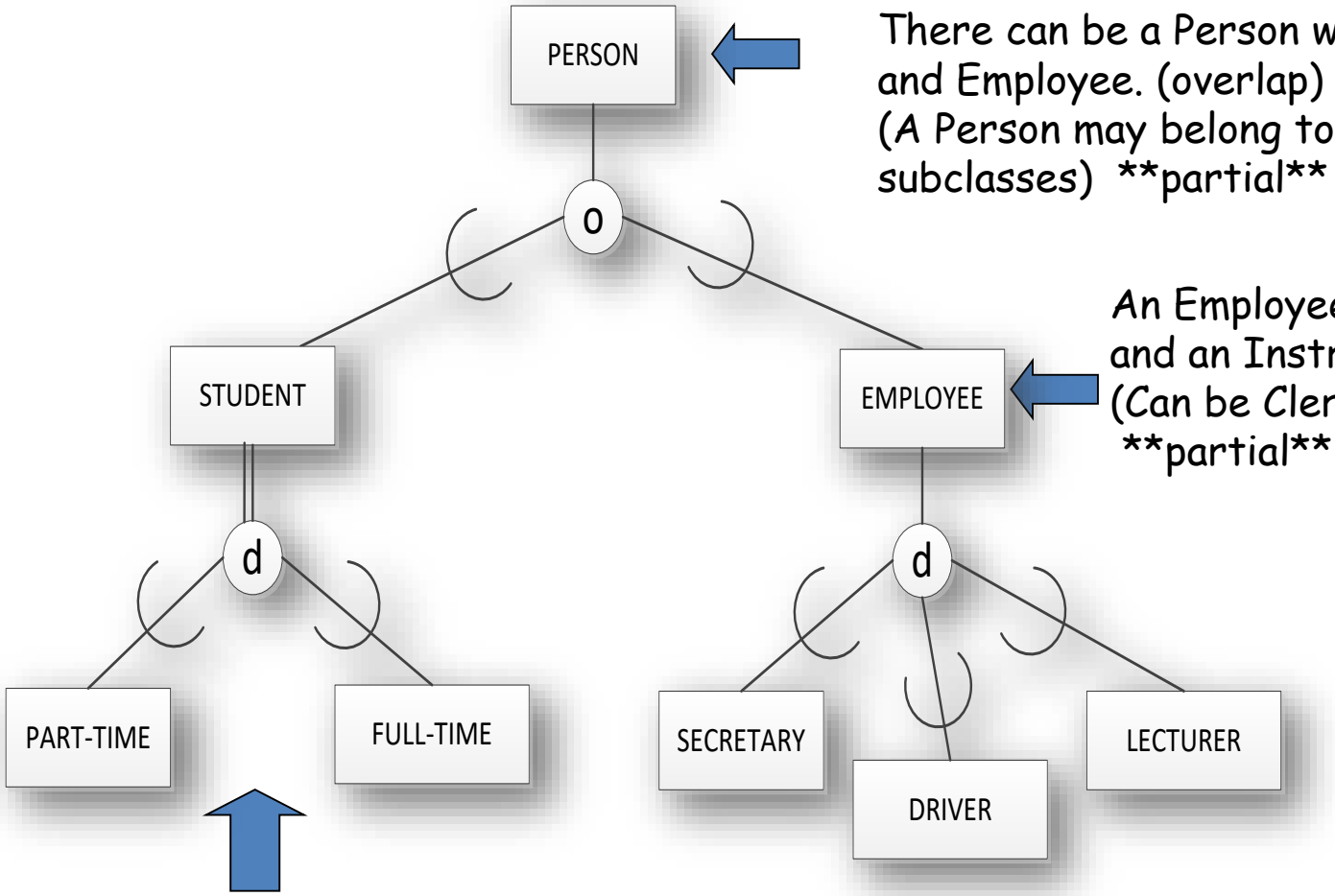
It is Partial, because we can have cars, bike and etc.

Examples



It's Total, because we have either part time or full time students.
No other options!!!!

Examples



There can be a Person who is both Student and Employee. (overlap)
(A Person may belong to more than one of its subclasses) **partial**

An Employee can't be a Secretary and an Instructor. (disjoint)
(Can be Clerk or Doctor or etc…)
partial

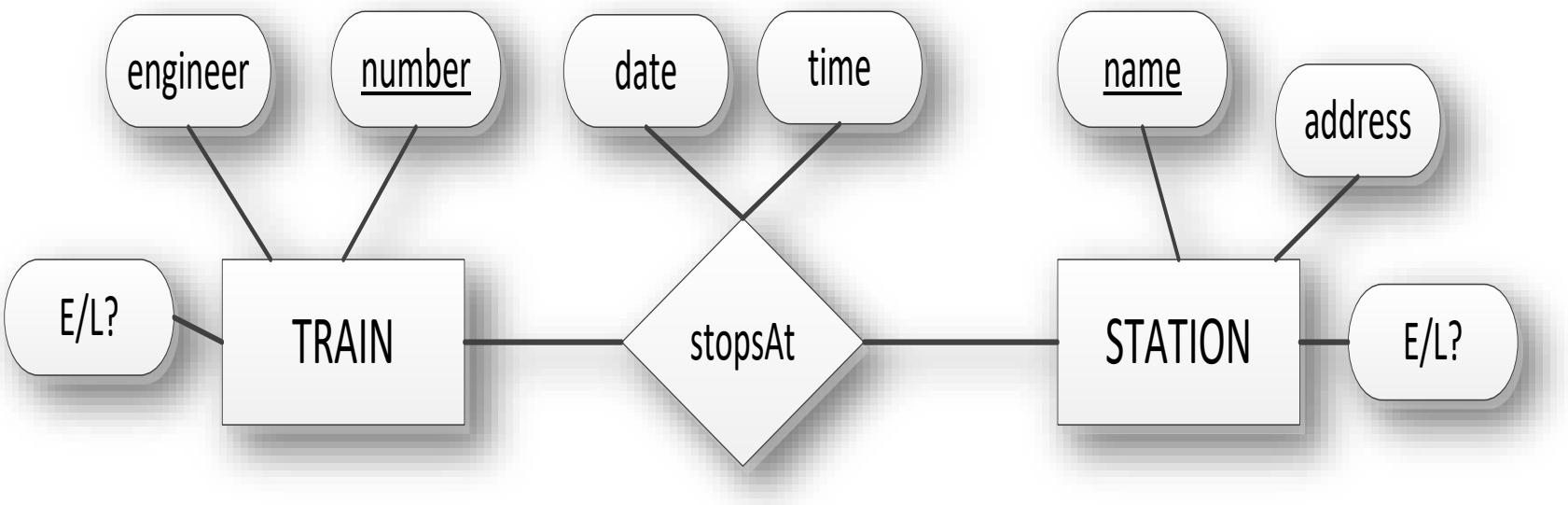
The Student has to be Parttime or Fulltime. Not BOTH!!! (disjoint)
(A Student has to belong to any of its subclasses) **total**

Case Study

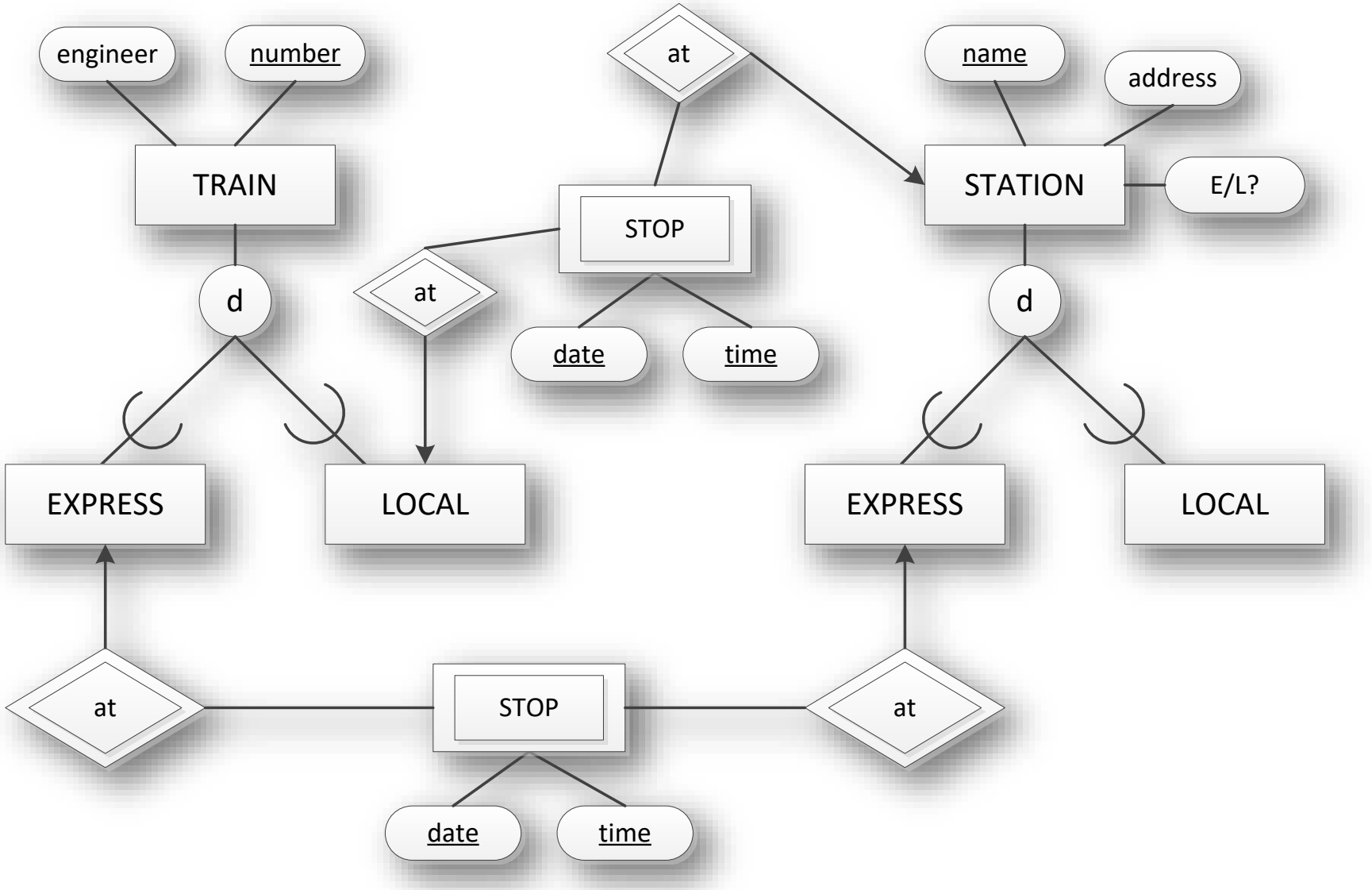
Design a database consistent with the following:

- A station has a unique name and an address, and is either an express station or a local station
- A train has a unique number and an engineer, and is either an express train or a local train
- A local train can stop at any station
- An express train only stops at express stations
- A train can stop at a station for any number of times during a day
- Train schedules are the same everyday

DESIGN 1


engineer
number
date
time
name
address
E/L?
TRAIN
stopsAt
STATION
E/L?

DESIGN 2


engineer
number
TRAIN
d
EXPRESS
LOCAL
at
STOP
at
date
time
name
address
STATION
E/L?
d
EXPRESS
LOCAL
at
STOP
at
date
time

Practice 1:

There are many students at the university. Students are subdivided into graduate and undergraduate students. Students take a course in a particular semester and receive a grade for their performances. Sometimes students take the same course again in a different semester. There are no limits on how many courses a student can take. Each graduate student has exactly one advisor, who must be a professor, whereas each professor is allowed to be the advisor of at the most 20 students. Courses have a unique course number and a course title. Students and professors have a name and a unique identification no (assume that both professor and student are subclasses of Person Class); students additionally have a gpa, and undergraduate students have many majors. Professors can be students and take courses, but graduate students cannot be undergraduate students.

Clearly indicate the Primary Keys.