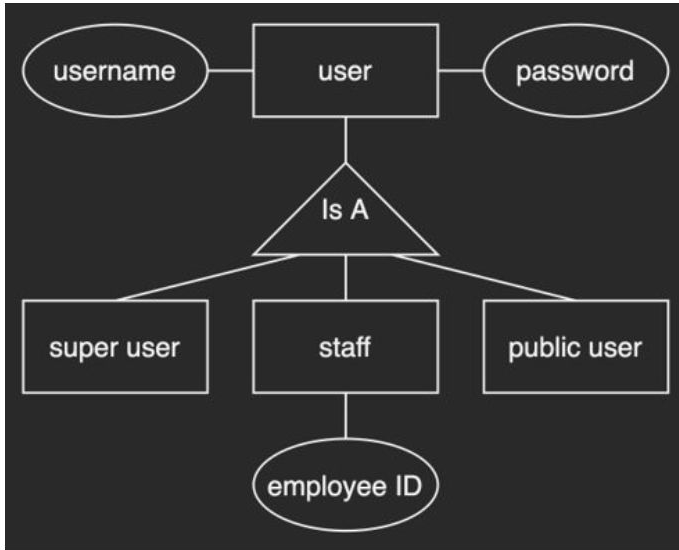


Is-A Relationships

Is-a relationships

Some entities are one of several different types
The types may share some, but not necessarily all attributes

Some entities are one of several different types
The types may share some, but not necessarily all attributes



Represented with triangles

Unidirectional

Read in direction that the triangle points

All users have username

All users have password

Staff users also have Employee ID

Enum data type

```
from enum import Enum

class UserType(Enum):
    SUPER = 1
    STAFF = 2
    PUBLIC = 3

user1 = {
    "username": "admin",
    "password": "admin123",
    "user_type": UserType.SUPER,
    "employee_id": None
}

user2 = {
    "username": "worker1",
    "password": "pass4321",
    "user_type": UserType.STAFF,
    "employee_id": "748393"
}
```

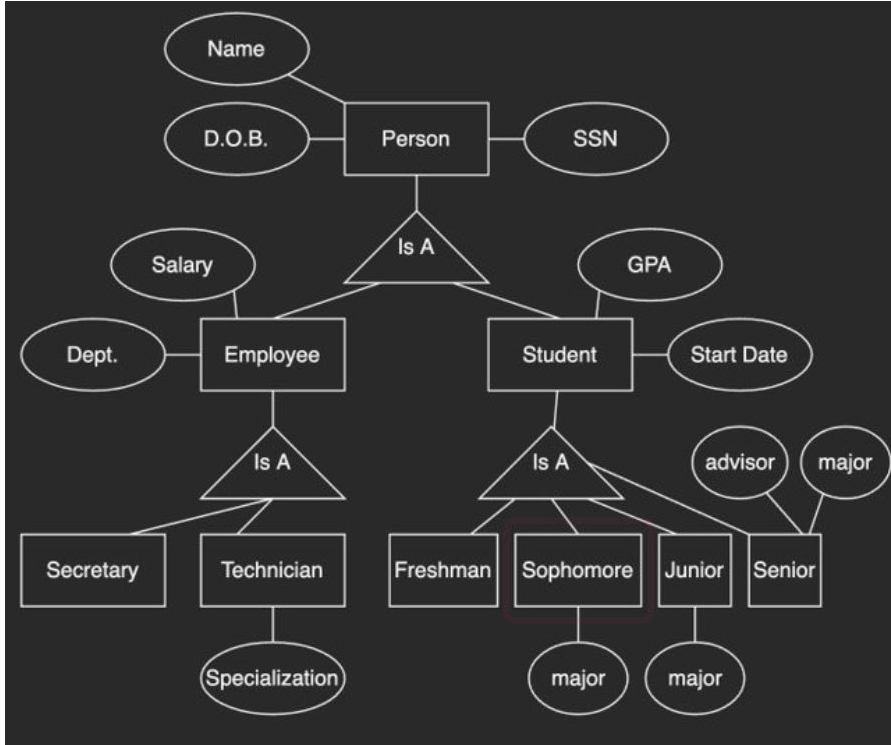
Short for Enumeration

Restricts possible values of data

Prevents "bad data" e.g. typos, incorrect formatting

Improves maintainability of code

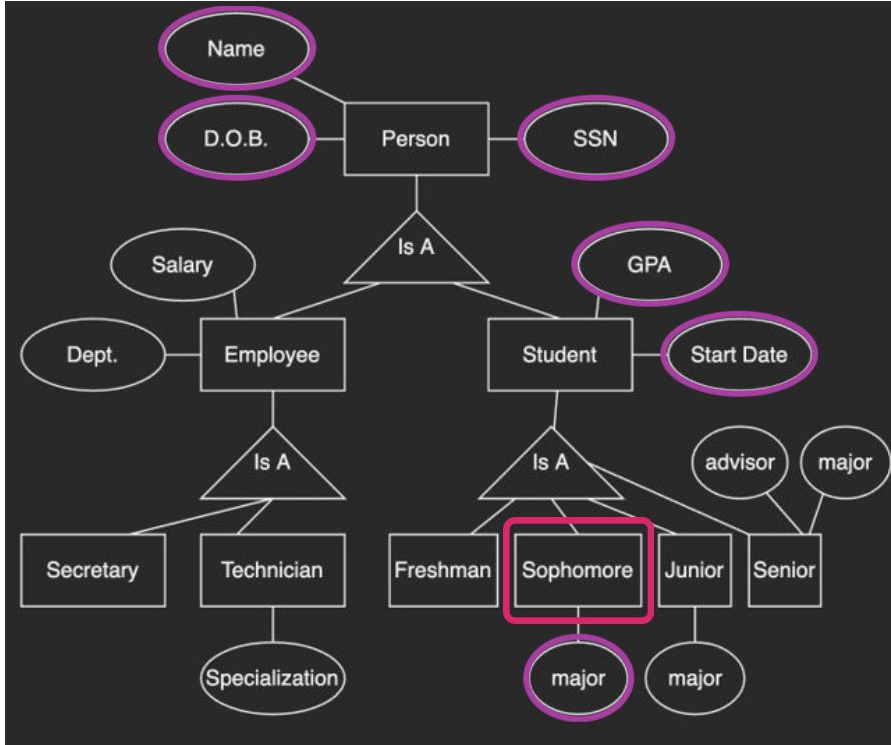
Example



Inheritance is ***transitive***

This means that if A inherits from B, and B inherits from C, then A inherits from C

Sophomore is a Student
Student is a Person



Thus, Sophomore is a Person

Sophomore has major, GPA, start date, name, SSN, and D.O.B