Lecture 8

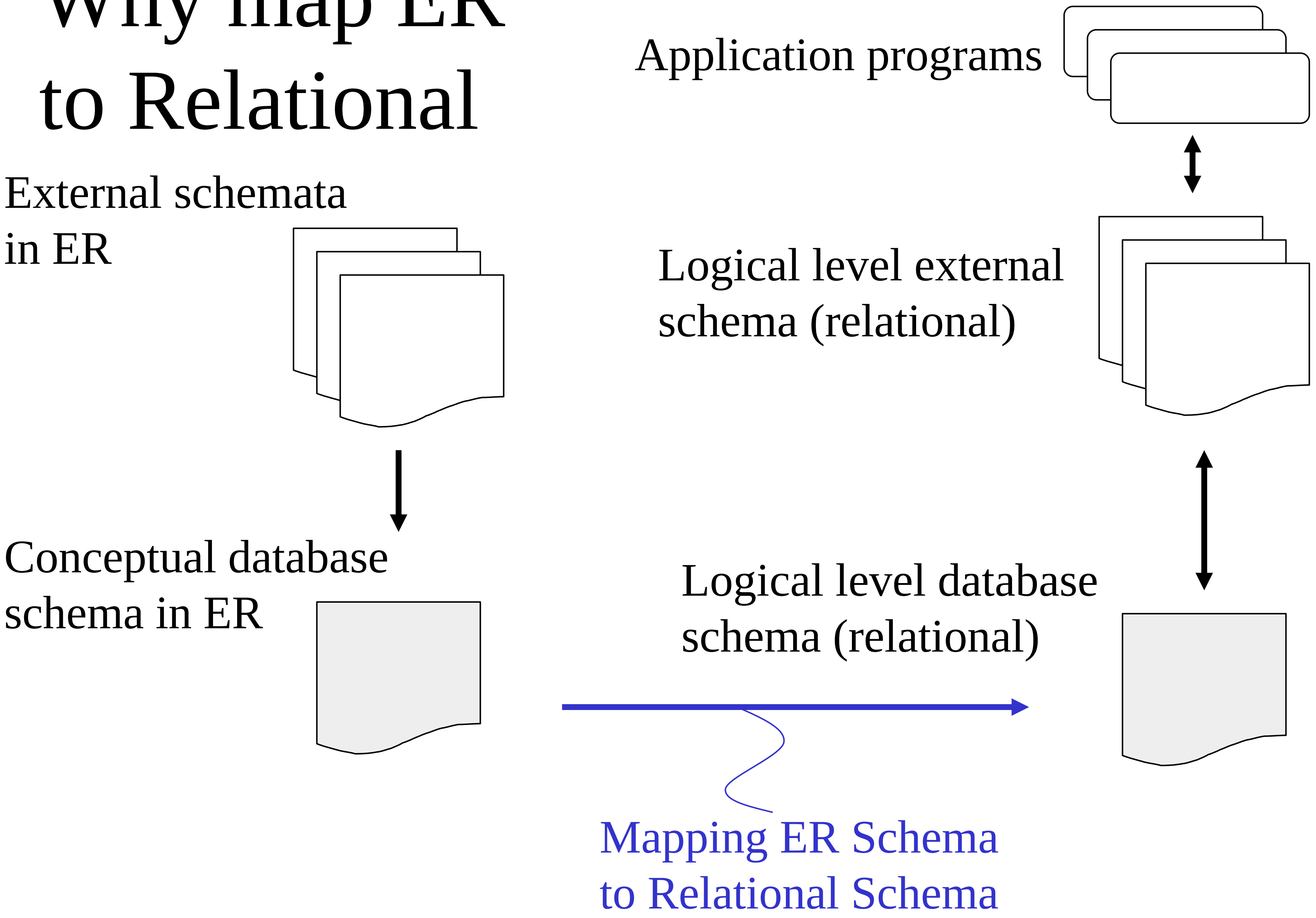
Mapping an Entity Relationship Schema to a Relational Schema

Week 4

Overview

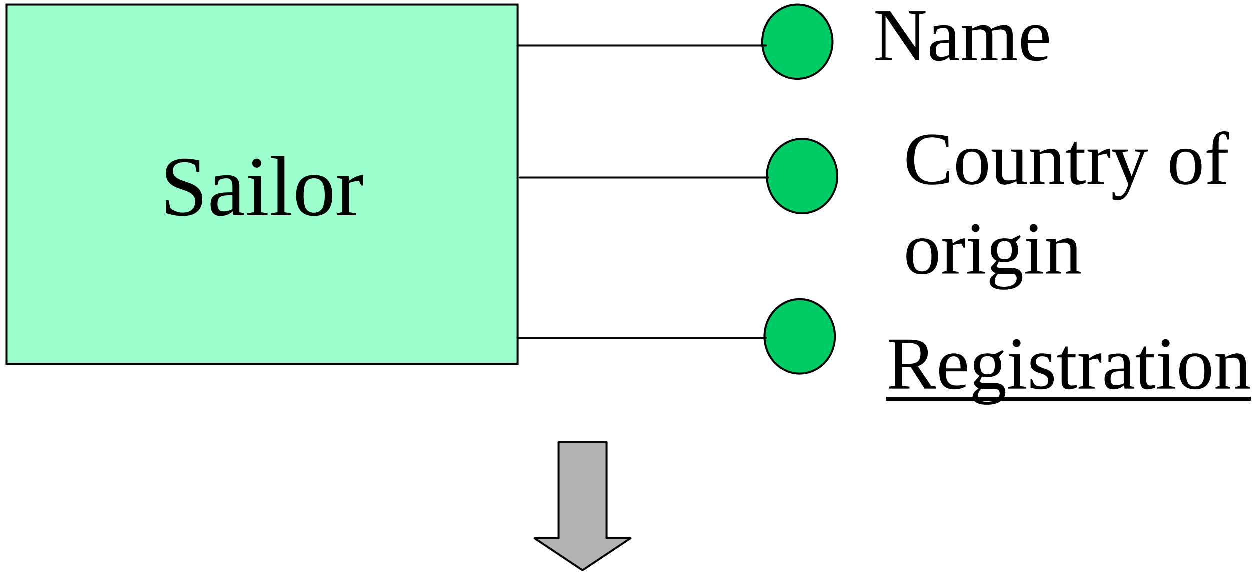
- Why map ER to Relational
- Mapping
  - Entities, and simple attributes
  - N:M relationships
  - 1:N and 1:1 relationships
  - Complex attributes
  - Multivalued attributes
  - Weak entities

Why map ER to Relational


Application programs
External schemata in ER
Logical level external schema (relational)
Conceptual database schema in ER
Logical level database schema (relational)
Mapping ER Schema to Relational Schema

1. Map entities and simple attributes

Every non-weak entity, together with its *simple* attributes is mapped to a separate relation schema



Sailor (<u>Registration</u>, Country of origin, Name)

- The candidate keys of the entity will be the candidate keys of the relation

- The primary key of the entity will be the primary key of the relation.

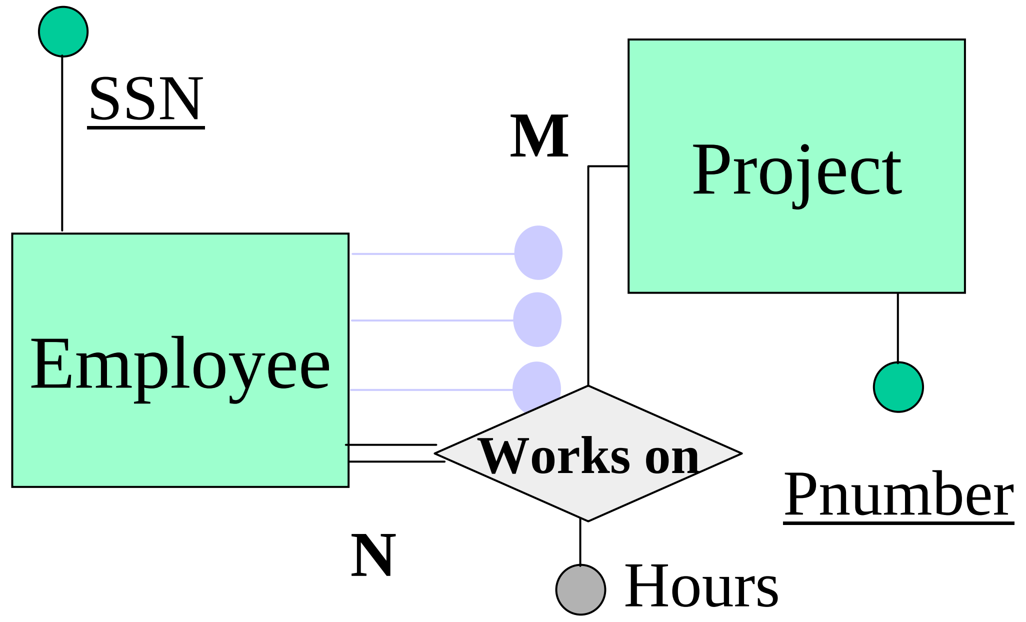
Sailor (<u>Registration</u>, Country of origin, Name)

pk: {Registration}

ck: {Registration}

2. Map N:M relations

Map every N:M relation to a separate relation



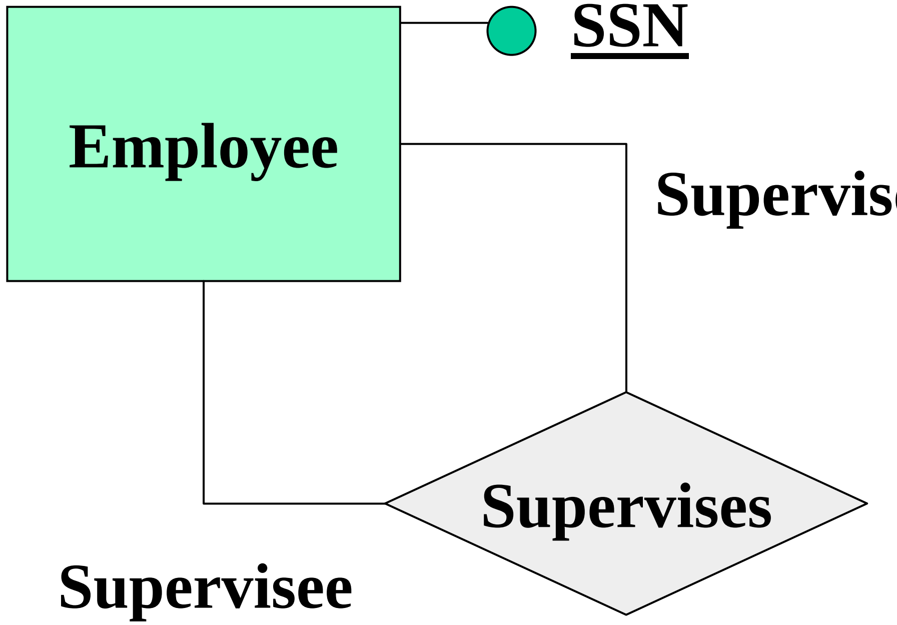
Works_on(<u>SSN, Pnumber</u>, Hours)

pk: {<u>SSN, Pnumber</u> }

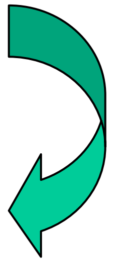
ck: {<u>SSN, Pnumber</u>}

PK is a composite of PKs of the involved entities!

In case of an entity participating in a relation more than once...



For each role the PK of the participating entity must be included in the PK of the relation (need to rename the attribute!)



Supervises (SupervisorSSN, SuperviseeSSN)

Foreign key constraints

- In any relationship (tuple) the ssn value can only be the value of an actual employee’s ssn. Similarly, Pnumber in Works_on must refer to an actual (not only potential) project number in the Project relation.
- These are called *foreign key constraints.*

We write:

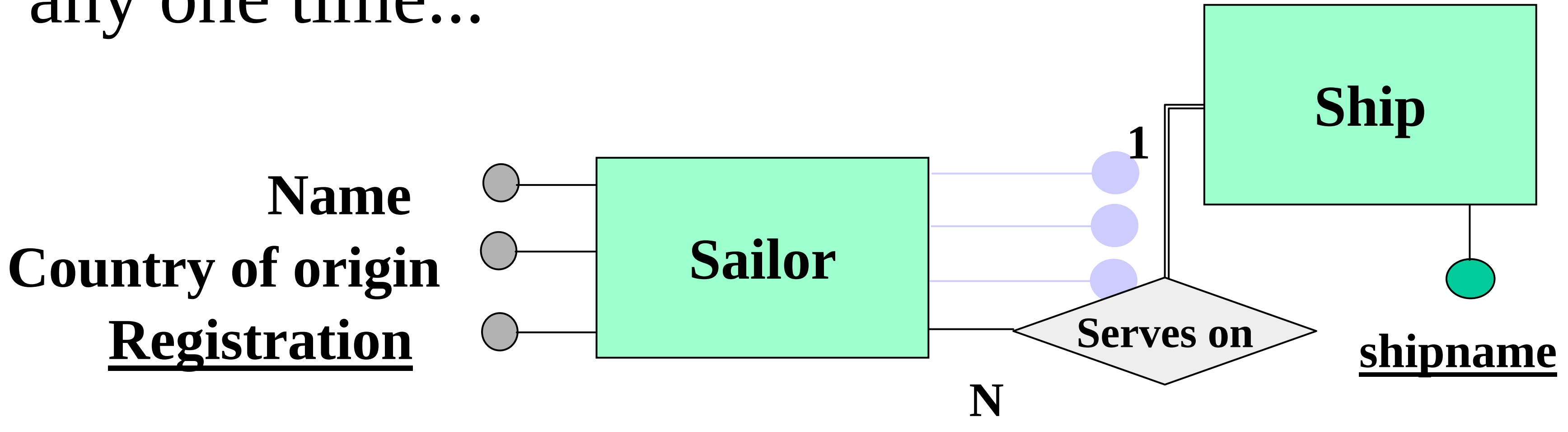
fk: SSN is SSN in Employee
fk: Pnumber is Pnumber in Project

Equivalently in graphical form:

$$\begin{tikzcd}
\textbf{Works\_on}(\underline{\textbf{SSN}, \textbf{Pnumber}}\,, \textbf{Hours}) \arrow[d] \arrow[dr] & \\
\textbf{Employee}(\underline{\textbf{SSN}},\ldots) & \textbf{Project}(\underline{\textbf{Pnumber}},\ldots)
\end{tikzcd}$$

3. Map N:1 relationship types

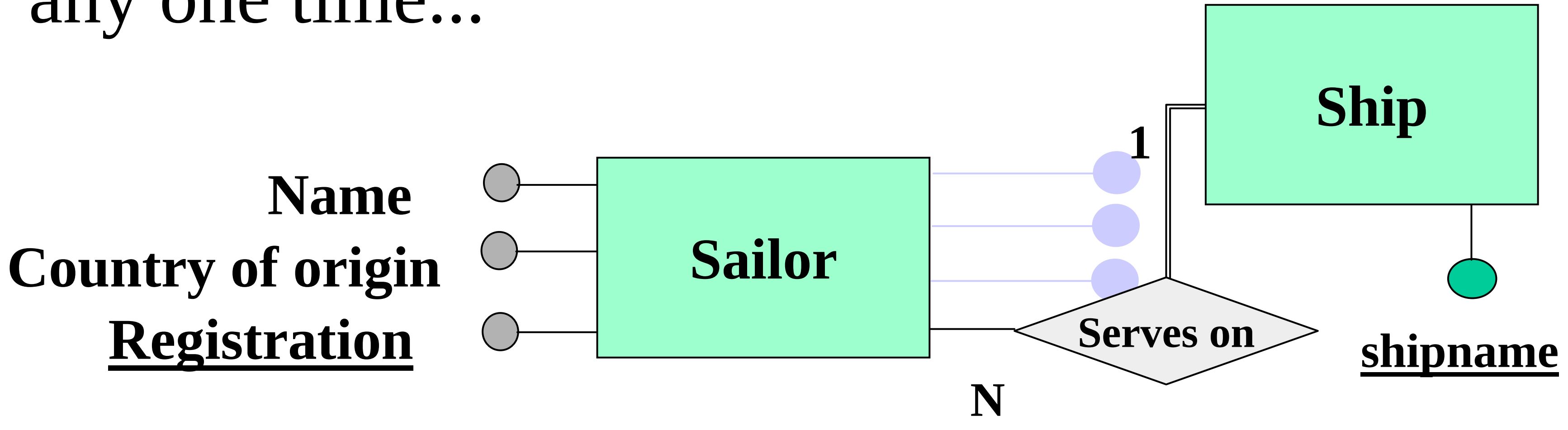
- N:1 relations *could* be treated as N:M relations (where M=1), but there is a better solution
- E.g. for every sailor there is only *one* ship at any one time...



Sailor (<u>Registration</u>, Country of origin, Name, ...

3. Map N:1 relationship types

- N:1 relations *could* be treated as N:M relations (where M=1), but there is a better solution
- E.g. for every sailor there is only *one* ship at any one time...



Sailor (<u>Registration</u>, Country of origin, Name, Shipname)

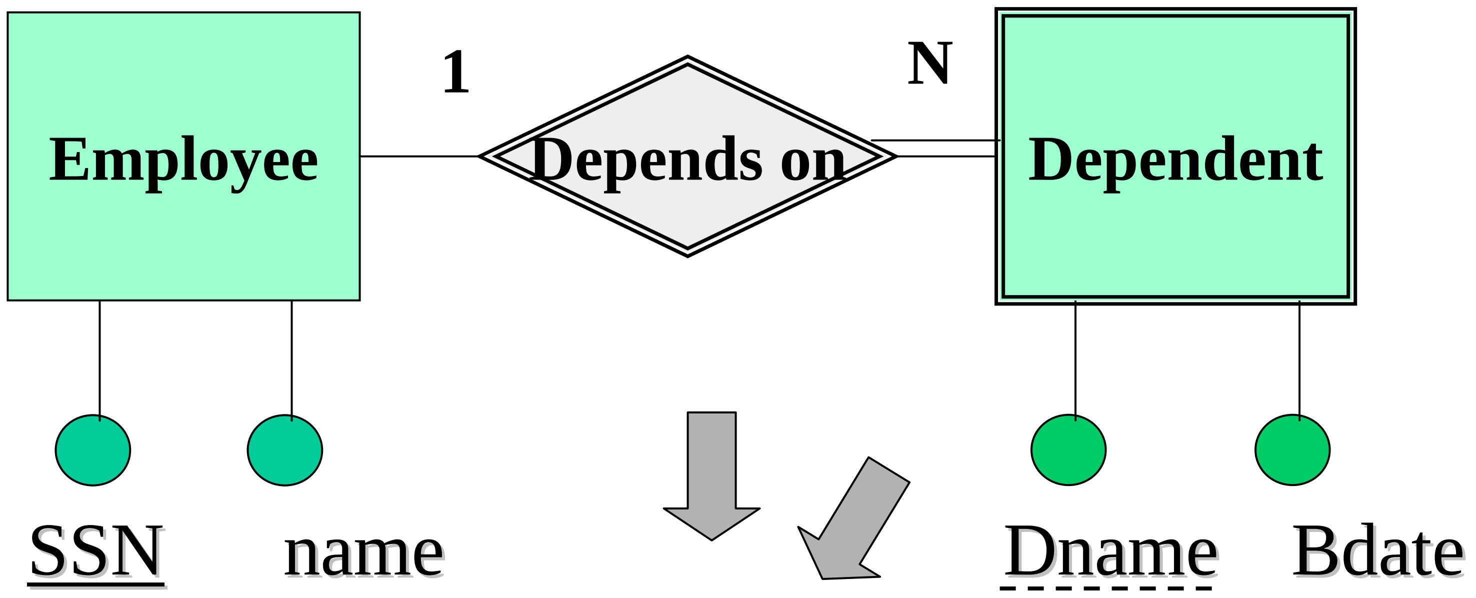
fk: Shipname is Shipname in Ship

3a. Map 1:1 relationship types

- We map 1:1 relationships as 1:N relationships (where N=1), i.e. by including the related entity’s primary key as a foreign key attribute of one of the participating entities.
- If there is a choice then we extend the schema of the entity which has total participation (to avoid NULL values)

4. Map weak entities

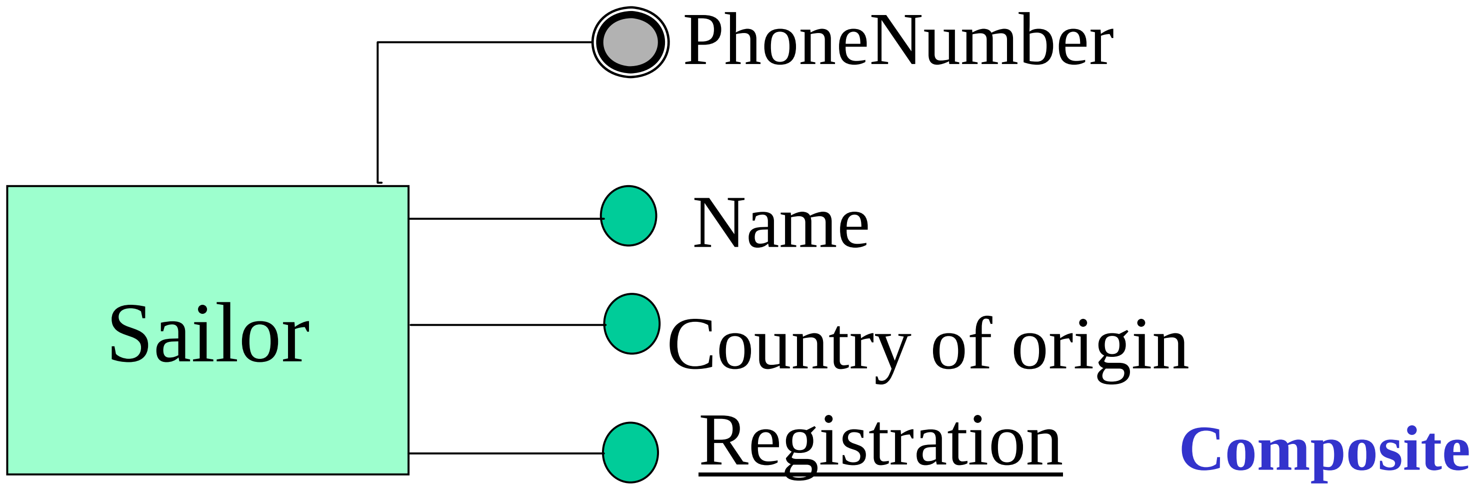
Every weak entity, together with its *simple* attributes is mapped to a separate relation schema



Dependent (<u>SSN, Dname</u>, Bdate)
fk: SSN is SSN in Employee

5. Map multivalued attributes

Every multivalued attribute of every entity is mapped to a separate schema!



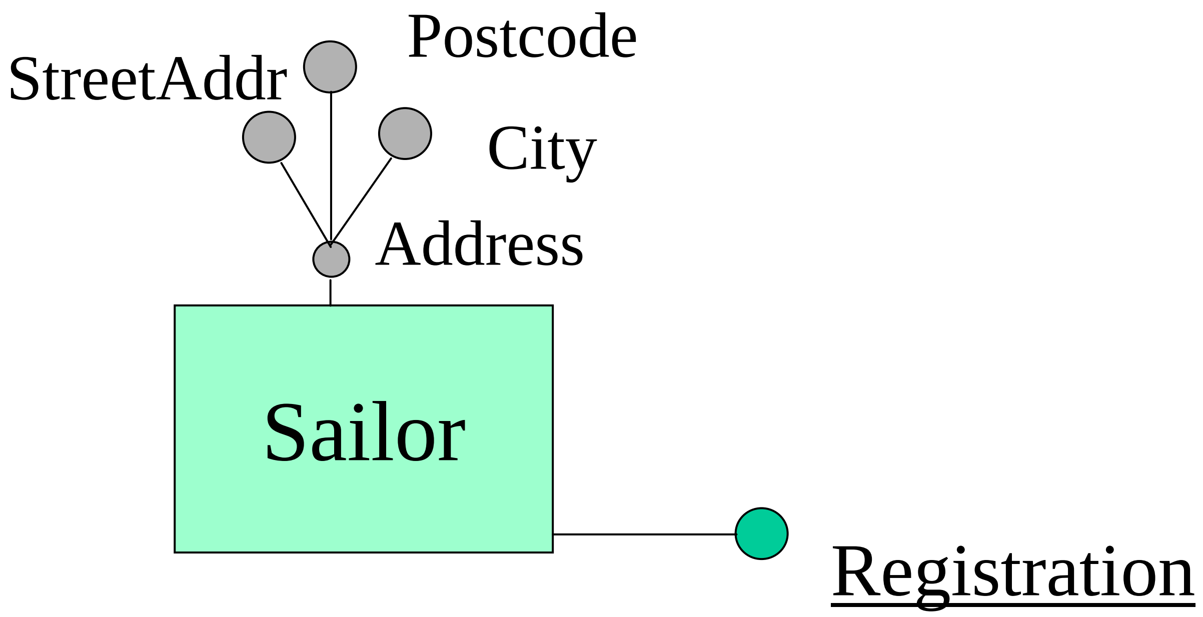
Composite key!

SailorPhoneNumber (<u>Registration, PhoneNumber</u>)

fk: Registration is Registration in Sailor

6. Mapping complex attributes...

- Complex attributes are mapped as if they were a series of attributes of the entity involved, e.g.



Sailor (<u>Registration</u>, …, StreetAddr, Postcode, City)

The end