Principles of Databases

Translating Entity-Relationship Data Models to Relations

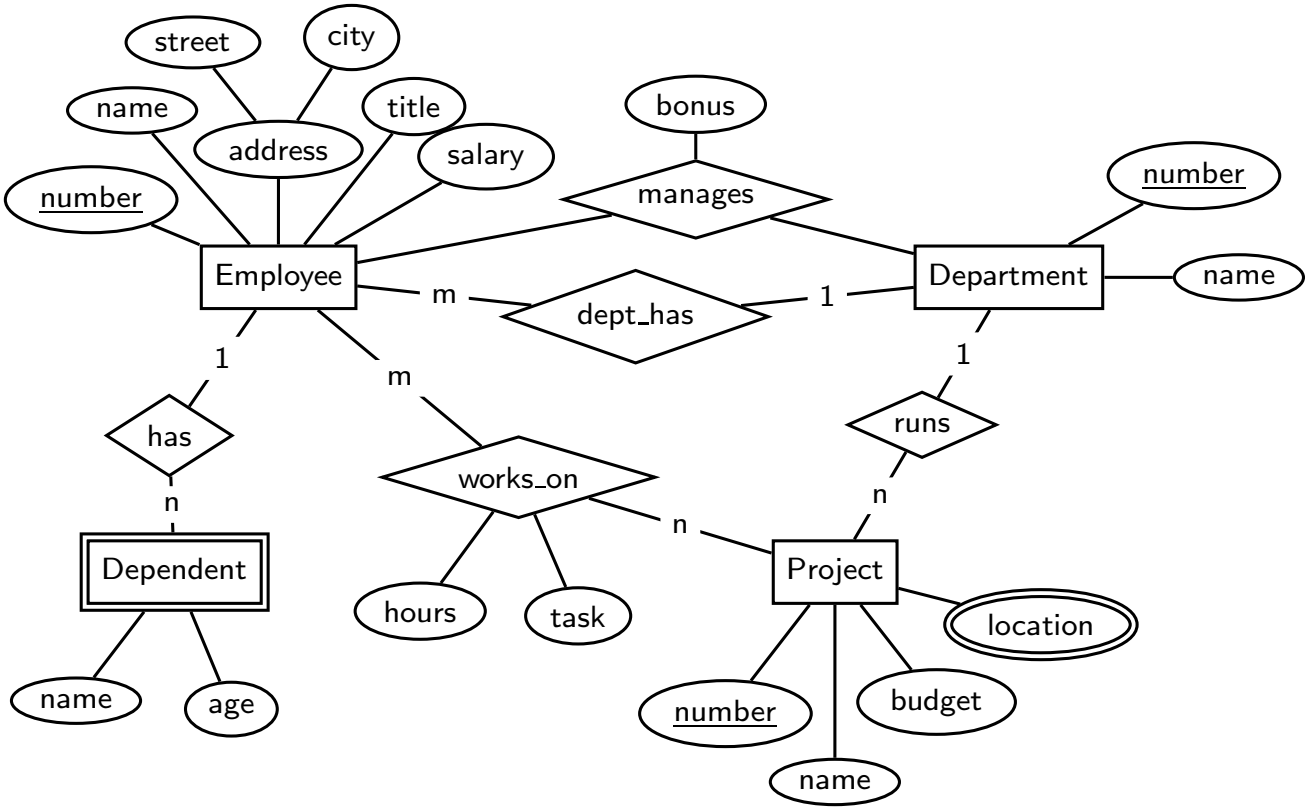
David Sinclair

Translating ER Models to Relational Schema

- An Entity-Relationship model is a conceptual model of the system we are modelling.
- We need to convert the conceptual model into a logical model that can be used to define the relations (tables) in a relational database.
- The resulting relations may need to be normalised and optimised.

Example

Consider the following ER model.



Strong Entities

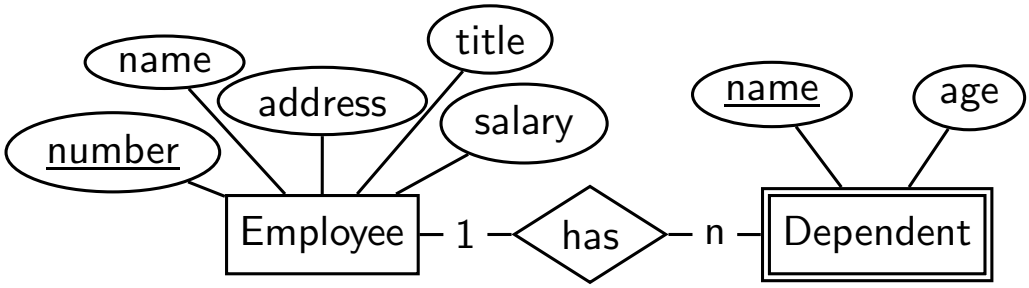
- Strong entities are easy to convert.
- Each strong entity becomes a relation (table).
- The attributes of the strong entity become the attributes of the relation.
- Composite attributes should be flattened to simple attributes. In our example, we should flatten address into street and city.
- The key of the strong entity becomes the primary key of the relation.

`Employee(`<u>`enum`</u>`, name, street, city, title, salary)`

Weak Entities

- Convert each weak entity into a relation with a foreign key to its identifying entity.
  - Identify the owning entities $E_1, E_2, \ldots, E_n$.
  - Create a relation R for the weak entity.
  - The primary key of R is the primary keys of the owning entities plus the partial key of the weak entity.
  - Create foreign keys in R for the primary keys in the owning entities.
  - Attributes are converted in a similar manner as strong entities.

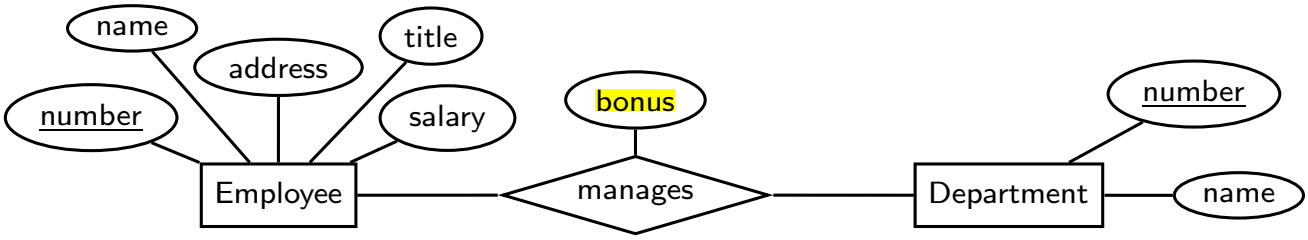
`Employee(`<u>`enum`</u>`, name, street, city, title, salary)`
`Dependent(`**<u>eno</u>**`,` <u>`name`</u>`, age)`



1:1 Relationships

- Convert a 1:1 relationship, R, between entities E_i and E_j into *unique* foreign keys references from the relations representing both entities.
  - Identify the relations R_i and R_j that correspond to the entities E_i and E_j.
  - Choose one of the relations, say R_i.
    - Choose the entity/relation that will always participate in the relationship.
  - Add the attributes of R to R_i.
  - Add the primary key attributes of R_j to R_i and create a foreign key reference.
  - Make sure the primary key attributes of R_j are *unique*.

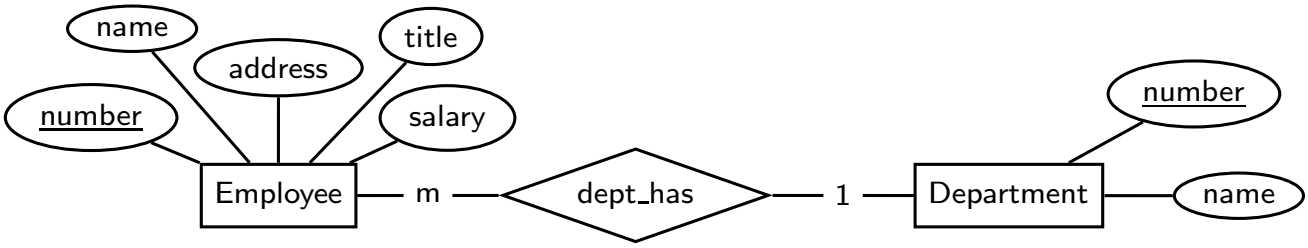
`Employee(`<u>`enum`</u>`, name, street, city, title, salary)`
`Department(`<u>`dnum`</u>`, name, bonus,` **`enum`**`)`



1:N Relationships

- Convert a 1:N relationship (or N:1 relationship), R, between entities E_i and E_j into a foreign key references from the N-sided entity to the 1-sided entity.
  - Identify the relations R_i and R_j that correspond to the entities E_i and E_j.
  - Choose the N-sided relation, say R_i.
  - Add the attributes of R to R_i.
  - Add the primary key attributes of R_j to R_i and create a foreign key reference to R_j from R_i.

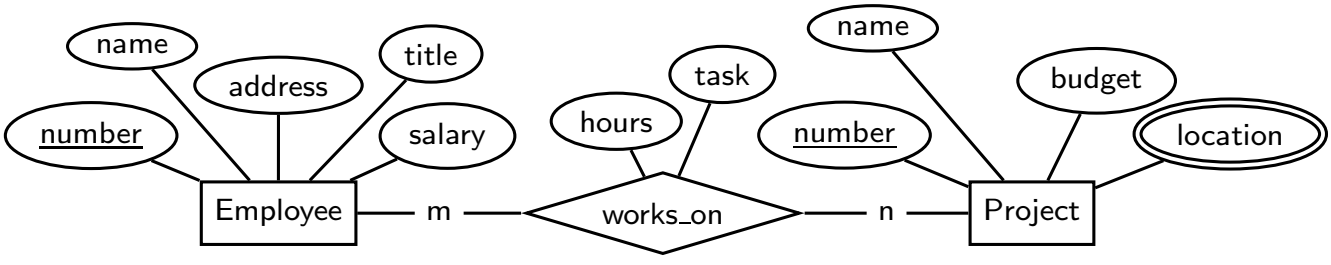
`Employee(`<u>`enum`</u>`, name, street, city, title, salary,` **dnum**`)`
`Department(`<u>`dnum`</u>`, name, bonus)`



M:N Relationships

- Convert a M:N relationship, R, between entities E_i and E_j into a new relation R with foreign key references to the two participating entities.
  - Identify the relations R_i and R_j that correspond to the entities E_i and E_j.
  - Create a new relation R.
  - Add the attributes of the relationship to R.
  - The primary key of R is a composite of the primary keys of R_i and R_j and create a foreign key reference from R to R_j and to R_i.

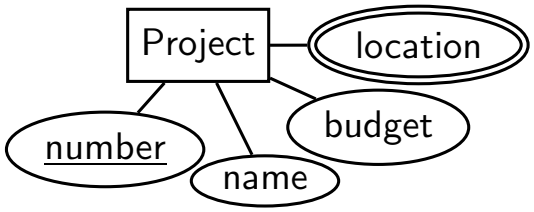
`Employee(`<u>`enum`</u>`, name, street, city, title, salary,` **dnum**`)`
`Project(`<u>`pnum`</u>`, name, budget, location)`
`WorksOn(`<u>**enum**</u>`,` <u>**pnum**</u>`, hours, task)`



Multi-valued Attributes

- Convert multi-values attributes into a relation with a composite key of the attribute value and the primary key of the attribute's entity.
  - Identify the entity's relation, R_i.
  - Create a relation R for the multi-value attribute where R contains:
    - a single-value attribute; and
    - R_i's primary key, which is a foreign key from R to R_i.
  - The primary key of R is a composite key of the attribute value and R_i's primary key

`Project(`<u>`pnum`</u>`, name, budget)`
`ProjectLocation(`**<u>pnum</u>**`, `<u>`location`</u>`)`



N-ary Relationships

- Convert an N-ary relationship, R, between the entities into a new relation R with foreign key references to the related entities.
  - Identify the relations $R_1, R_2, \ldots, R_n$ that correspond to the entities $E_1, E_2, \ldots, E_n$.
  - Create a new relation R.
  - Add the attributes of the relationship to R.
  - The primary key of R is a composite of the primary keys of $R_1, R_2, \ldots, R_j$ and create foreign key references from R to $R_1, R_2, \ldots, R_i$.