Medical Informatics

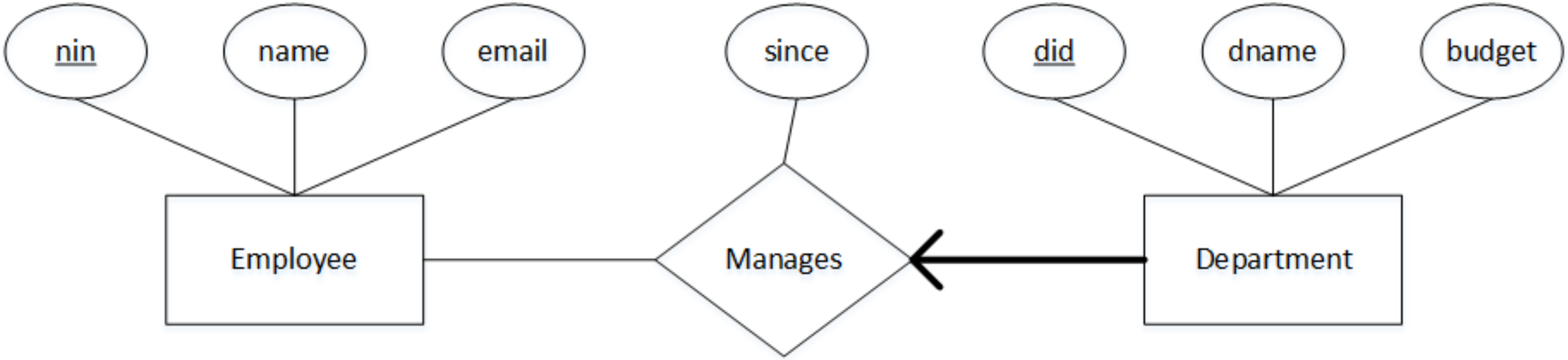
Lecture 5: Translating an ER model into the Relational Model

Dr Areti Manataki

Nanjing Medical University

The story so far: ER model

- ER diagram basics

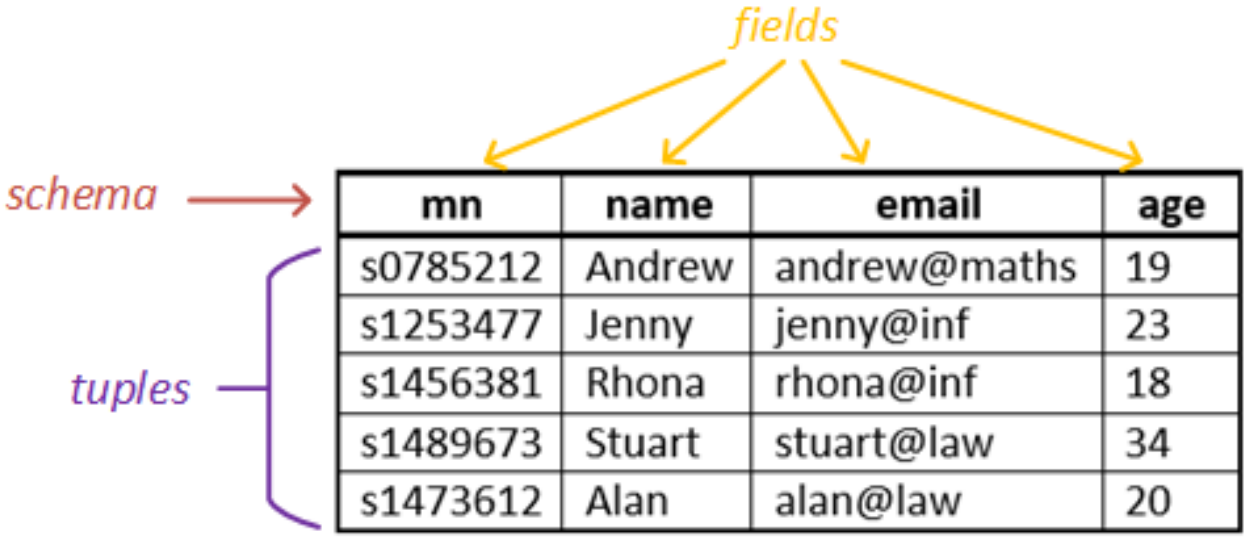


The story so far: ER model

- ER diagram basics:
  - Entity sets: rectangles
  - Relationship sets: diamonds connected to the entity sets
  - Attributes: ovals (with primary keys underlined)
  - Key constraints: arrows
  - Total participation: thick line
  - Weak entity sets: thick or double border
  - Entity hierarchies: ISA

The story so far: relational model

- Relational model basics

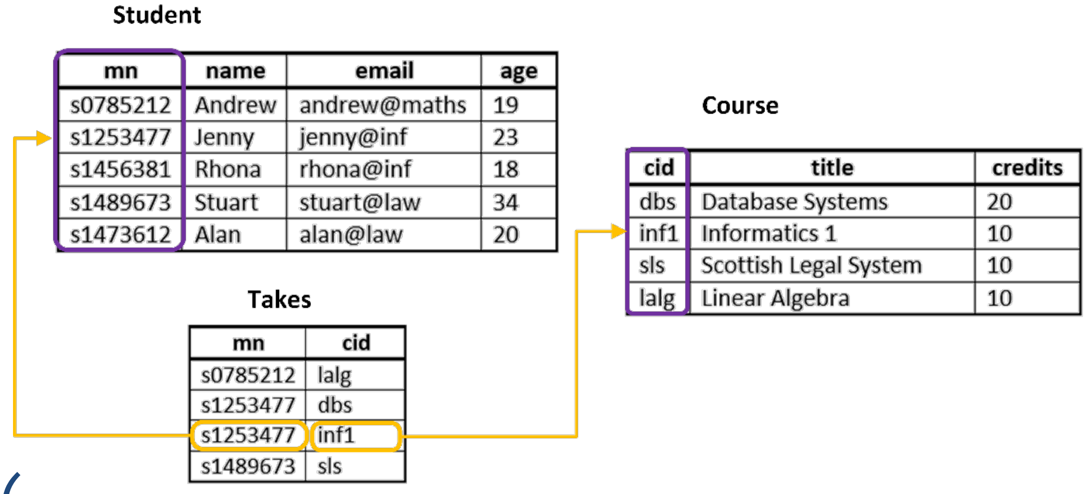


mn	name	email	age
s0785212	Andrew	andrew@maths	19
s1253477	Jenny	jenny@inf	23
s1456381	Rhona	rhona@inf	18
s1489673	Stuart	stuart@law	34
s1473612	Alan	alan@law	20

The story so far: relational model

- Relational model basics:
  - Relation schema: the format of a table, consisting of its name and the names and domains of each column
  - Relation instance: set of tuples matching the schema
  - Primary key: minimal set of fields uniquely identifying a tuple
  - Foreign key: values must be drawn from key values in the referenced table

The story so far: relational model



Student

mn	name	email	age
s0785212	Andrew	andrew@maths	19
s1253477	Jenny	jenny@inf	23
s1456381	Rhona	rhona@inf	18
s1489673	Stuart	stuart@law	34
s1473612	Alan	alan@law	20

Course

cid	title	credits
dbs	Database Systems	20
inf1	Informatics 1	10
sls	Scottish Legal System	10
lalg	Linear Algebra	10

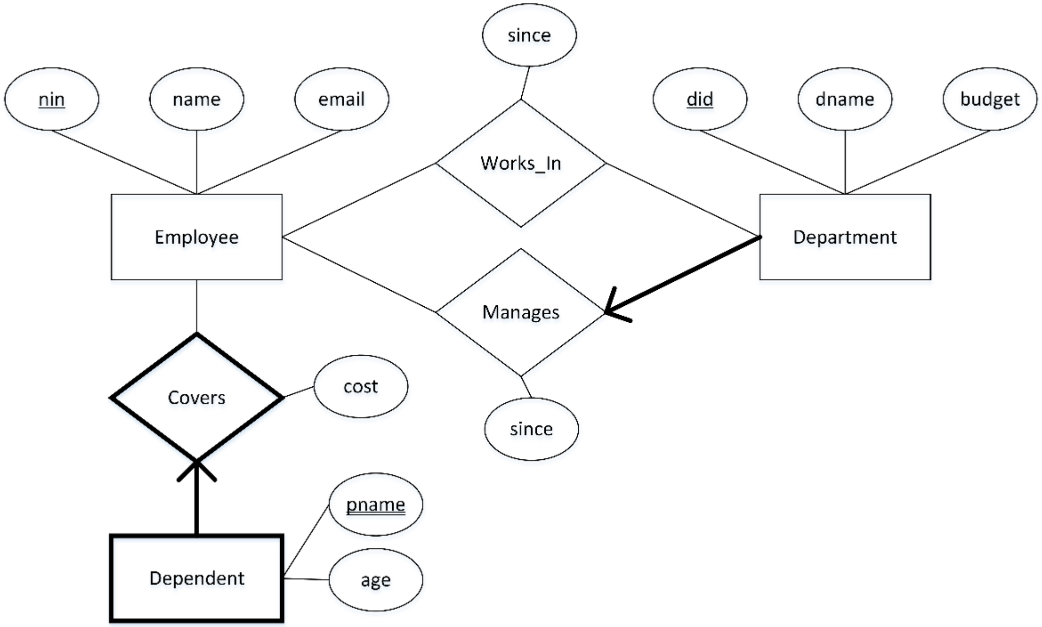
Takes

mn	cid
s0785212	lalg
s1253477	dbs
s1253477	inf1
s1489673	sls

```
CREATE TABLE Takes (
    mn CHAR(8),
    cid CHAR(20),
    mark INTEGER,
    PRIMARY KEY (mn, cid),
    FOREIGN KEY (mn) REFERENCES Student,
    FOREIGN KEY (cid) REFERENCES Course
 )
```

In this lecture

- Systematically transform an ER model (conceptual model) into a relational one (implementation in RDBMS)



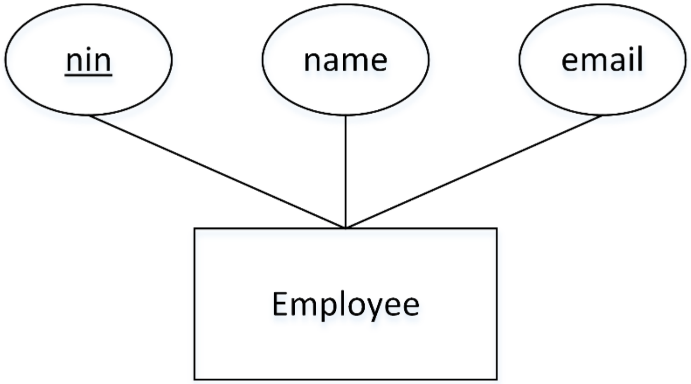
```
CREATE TABLE Employee (
    nin CHAR(9),
    name VARCHAR(20),
    email VARCHAR(35),
    PRIMARY KEY (nin) )
CREATE TABLE Works_In (
    nin CHAR(9),
    did INTEGER,
    since INTEGER,
    PRIMARY KEY (nin, did),
    FOREIGN KEY (nin) REFERENCES Employee,
    FOREIGN KEY (did) REFERENCES Department
)
CREATE TABLE ……
```

In this lecture

- Systematically transform an ER model (conceptual model) into a relational one (implementation in RDBMS)
- Transforming:
  - entity and relationship sets
  - key and participation constraints
  - weak entity sets and hierarchies

Entity sets to tables

- Create a table for the entity set.
- Each attribute becomes a field, with the corresponding domain.
- The primary key is declared.



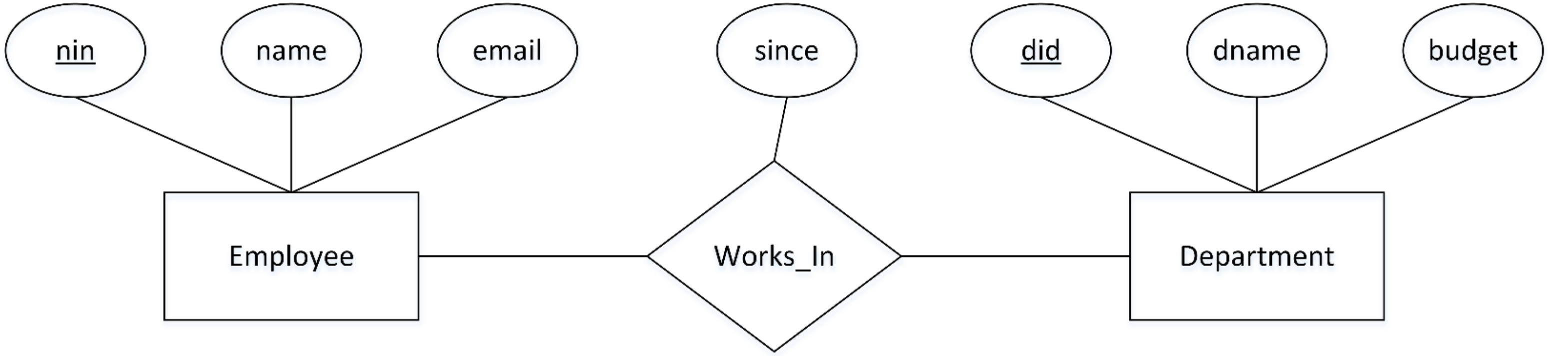
```
CREATE TABLE Employee (
    nin CHAR(9),
    name VARCHAR(20),
    email VARCHAR(35),
    PRIMARY KEY (nin) )
```

Relationship sets to tables

In the case where there are no constraints:

- Create a table for the relationship set.
- Specify a field (with the corresponding domain) for each of the following:
  - attributes of the relationship set
  - primary key attributes of each participating entity set.
- Declare primary key using all key attributes from participating entity sets.
- Declare foreign key constraints for all key attributes from participating entity sets.

Relationship sets to tables

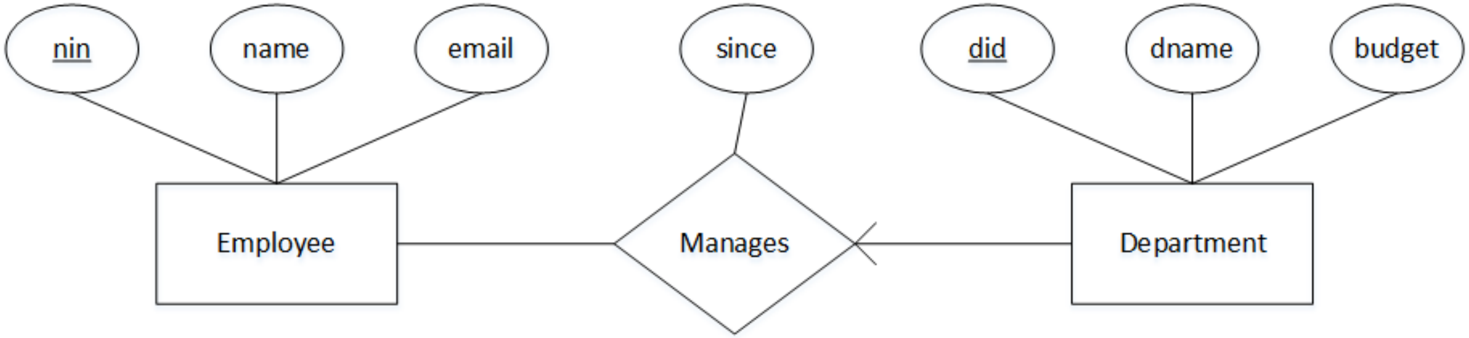


```
CREATE TABLE Works_In (
    nin CHAR(9),
    did INTEGER,
    since INTEGER,
    PRIMARY KEY (nin, did),
    FOREIGN KEY (nin) REFERENCES Employee,
    FOREIGN KEY (did) REFERENCES Department
  )
```

Mapping relationship sets with key constraints – 1st approach

- Create a table for the relationship set.
- Specify a field (with the corresponding domain) for each of the following:
  - attributes of the relationship set
  - primary key attributes of each participating entity set.
- Declare primary key using only key attributes of the source entity set.
- Declare foreign key constraints for all key attributes from participating entity sets.

Mapping relationship sets with key constraints – 1st approach

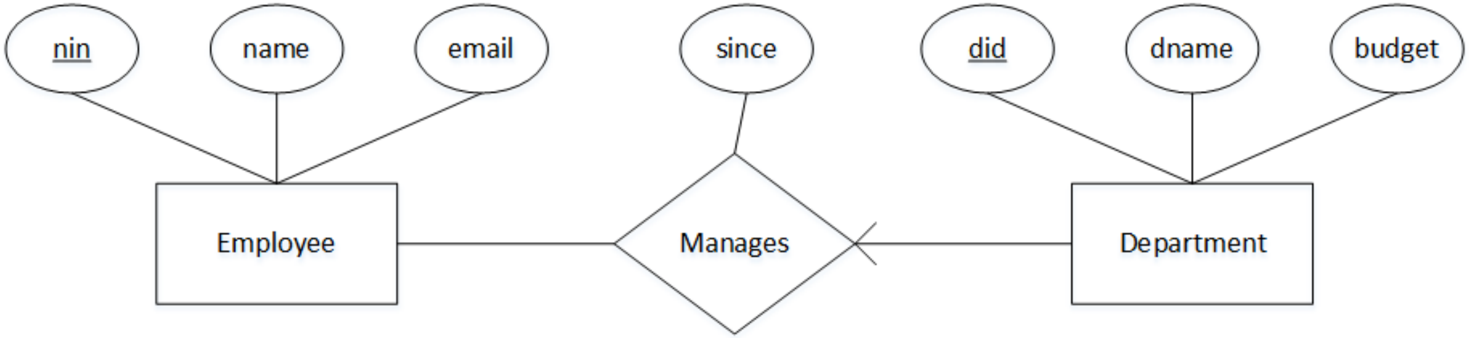


```
CREATE TABLE Manages (
    nin CHAR(9),
    did INTEGER,
    since INTEGER,
    PRIMARY KEY (did),
    FOREIGN KEY (nin) REFERENCES Employee,
    FOREIGN KEY (did) REFERENCES Department
  )
```

Mapping relationship sets with key constraints – 2nd approach

- Main idea: no separate table, just include the information about the relationship set in the table for the entity set with the key (source entity set).
- In the table for the source entity set:
  - Specify a field (with the corresponding domain) for each of the following:
    - attributes of the relationship set
    - primary key attributes of the target entity set.
  - Declare foreign key constraints for the primary key attributes of the source entity set.

Mapping relationship sets with key constraints – 2nd approach

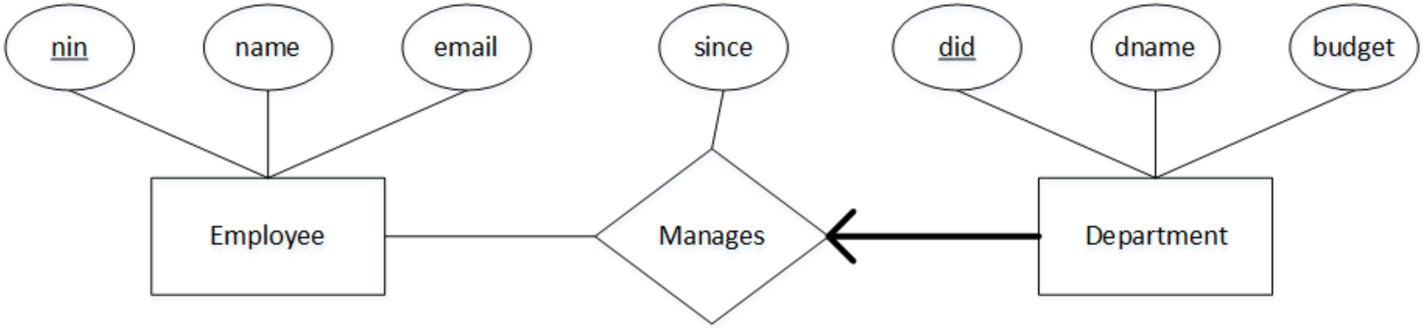


```
CREATE TABLE Department (
    did INTEGER,
    dname VARCHAR(20),
    budget INTEGER,
    nin CHAR(9),
    since INTEGER,
    PRIMARY KEY (did),
    FOREIGN KEY (nin) REFERENCES Employee )
```

Mapping key and participation constraints

- In the table for the source entity set:
  - Specify a field (with the corresponding domain) for each of the following:
    - attributes of the relationship set
    - primary key attributes of the target entity set.
  - Declare foreign key constraints for the primary key attributes of the source entity set.
  - Declare those fields as NOT NULL.

Mapping key and participation constraints



```
CREATE TABLE Department (
    did INTEGER,
    dname VARCHAR(20),
    budget INTEGER,
    nin CHAR(9) NOT NULL,
    since INTEGER,
    PRIMARY KEY (did),
    FOREIGN KEY (nin) REFERENCES Employee )
```

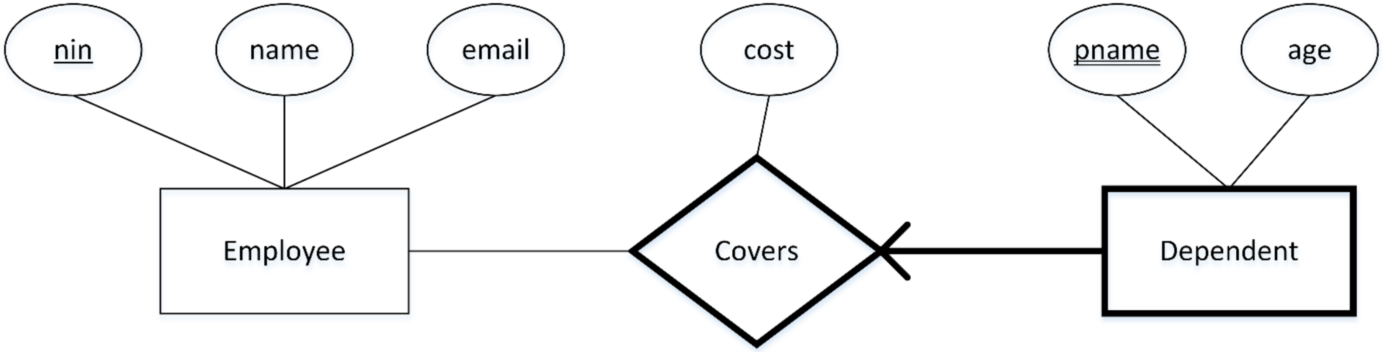
Mapping participation constraints

When not combined with a key constraint, a participation constraint cannot be captured in SQL, unless we use more advanced SQL.

Translating weak entity sets

- Create a table for the weak entity set.
- Specify a field (with the corresponding domain) for each of the following:
  - attributes of the weak entity set
  - attributes of the identifying relationship set
  - primary key attributes of the identifying owner.
- Declare a primary key that consists of the partial key of the weak entity set and the key attributes of the identifying owner.
- Declare foreign key constraints for the primary key attributes of the identifying owner, and set them ON DELETE CASCADE.

Translating weak entity sets

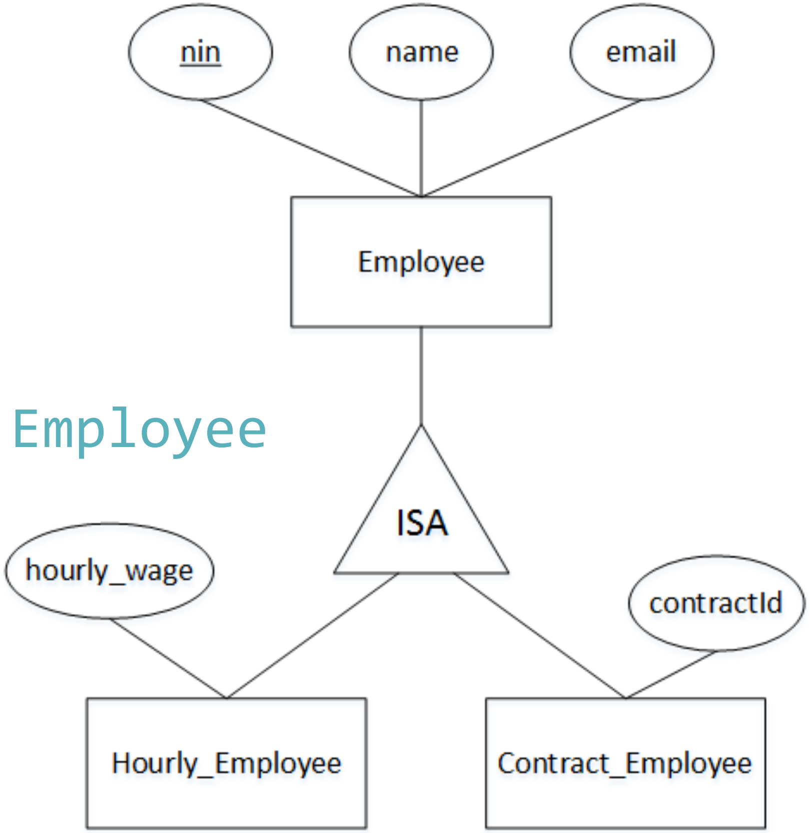


```
CREATE TABLE Dependent (
    pname VARCHAR(20),
    age INTEGER,
    cost INTEGER,
    nin CHAR(9),
    PRIMARY KEY (pname, nin),
    FOREIGN KEY (nin) REFERENCES Employee
                      ON DELETE CASCADE )
```

Translating entity hierarchies

- Create a table for the superclass, as usual.
- For each subclass:
    - Create a table.
    - Specify a field (with the corresponding domain) for each of the following:
        - attributes of the subclass
        - primary key attributes of the superclass.
    - Declare the primary key to be the primary key of the superclass.
    - Declare foreign key constraints for the primary key attributes of the superclass.

Translating entity hierarchies



```
CREATE TABLE Hourly_Employee (
    nin CHAR(9),
    hourly_wage INTEGER,
    PRIMARY KEY (nin),
    FOREIGN KEY (nin) REFERENCES Employee
  )
CREATE TABLE Contract_Employee (
    nin CHAR(9),
    contract_id INTEGER,
    PRIMARY KEY (nin),
    FOREIGN KEY (nin) REFERENCES Employee
  )
```

Conclusions

- We’ve seen how to systematically transform an ER model into a relational one.
- Some constraints expressed in the ER model might not be that easy to capture in the relational model.
- In some cases, there might be more than one possible transformations.
- In the next lecture we’ll learn how to populate tables. We’ll also get to write some simple queries in SQL.

Acknowledgements