CS306 - Recitation 3

RELATIONS AND REFERENTIAL INTEGRITY
ER to RELATIONAL MODEL

DOMAIN CONSTRAINT

The type (domain) of each field is specified, and enforced by the DBMS whenever tuples are added or modified

```
CREATE TABLE students
(sid INTEGER,
name CHAR(50),
login CHAR(50),
age INTEGER,
gpa real);
```

```
INSERT INTO  Students (sid, name, login, age, gpa) VALUES  (53688, 'Shero', 'shero@cs', 18, 3.2);
```

Try this:

```
INSERT INTO  Students (sid, name, login, age, gpa)VALUES  (53689, 'Shero', 'shero@cs', '18 age', 3.2);
```

PRIMARY KEY

The PRIMARY KEY constraint uniquely identifies each record in a table.

```
ALTER TABLE Students ADD PRIMARY KEY (sid);
```

```
CREATE TABLE courses
(cid INTEGER,
cname CHAR(30),
grade char(1),
PRIMARY KEY (cid));
```

FOREIGN KEY

A FOREIGN KEY is a key used to link two tables together.
A FOREIGN KEY is a field in one table that refers to the PRIMARY KEY in another table.

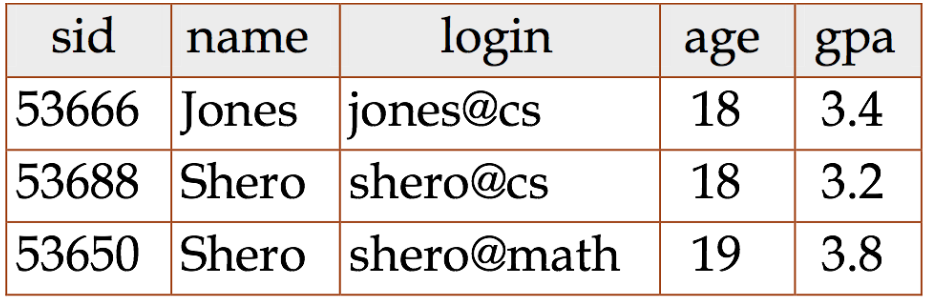
```
CREATE TABLE enrolled(
sid INTEGER,
cid INTEGER,
FOREIGN KEY (sid) REFERENCES students(sid) ON DELETE SET NULL,
FOREIGN KEY (cid) REFERENCES courses(cid) ON UPDATE CASCADE);
```

Referential Integrity

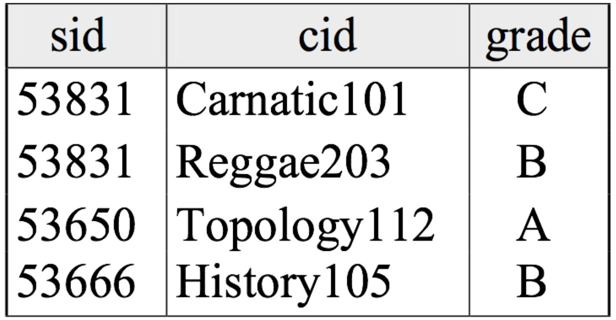
- ON DELETE CASCADE: if a row of the referenced table is DELETED, then all matching rows in the referencing table are also DELETED
- ON UPDATE CASCADE: if a row of the referenced table is UPDATED, then all matching rows in the referencing table are also UPDATED
- ON DELETE SET NULL: if a row of the referenced table is deleted, then all referencing columns in all matching rows of the referencing table to be set to null
- ON DELETE SET DEFAULT: if a row of the referenced table is deleted, then all referencing columns in all matching rows of the referencing table to be set to the column’s default value.

```
CREATE TABLE enrolled(
sid INTEGER,
cid INTEGER,
FOREIGN KEY (sid) REFERENCES students(sid) ON DELETE SET NULL,
FOREIGN KEY (cid) REFERENCES courses(cid) ON UPDATE CASCADE);
```

EXERCISE



sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Shero	shero@cs	18	3.2
53650	Shero	shero@math	19	3.8



sid	cid	grade
53831	Carnatic101	C
53831	Reggae203	B
53650	Topology112	A
53666	History105	B

Inserting some values to check referential integrity:

INSERT INTO enrolled VALUES (1,’Hasan’,’hasan@cs’,25,3.5)

Deleting some values to check referential integrity holds:

DELETE FROM students WHERE sid = 53650

Updating some values to check referential integrity holds:

UPDATE students SET sid = 60000 WHERE sid = 53666

VIEW

A *view* is just a relation, but we store a *definition*, rather than a set of tuples.

```
CREATE VIEW goodStudents (sid, gpa)
AS SELECT S.sid, S.gpa
FROM students S
WHERE S.gpa > 3;
```

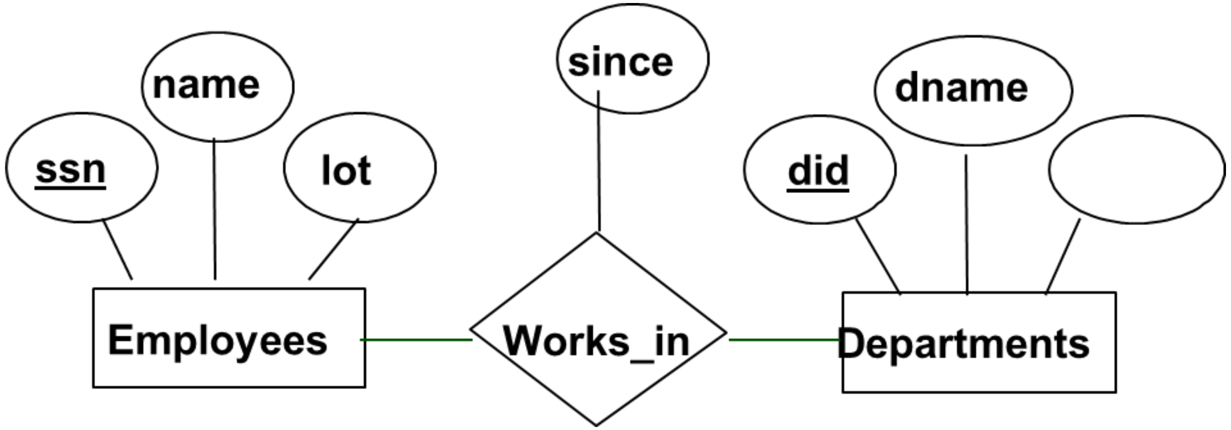
```
DROP VIEW goodStudents;
```

VIEW

A *view* is just a relation, but we store a *definition*, rather than a set of tuples.

```
CREATE VIEW goodStudents (sid, gpa)
AS SELECT S.sid, S.gpa
FROM students S
WHERE S.gpa > 3;
```

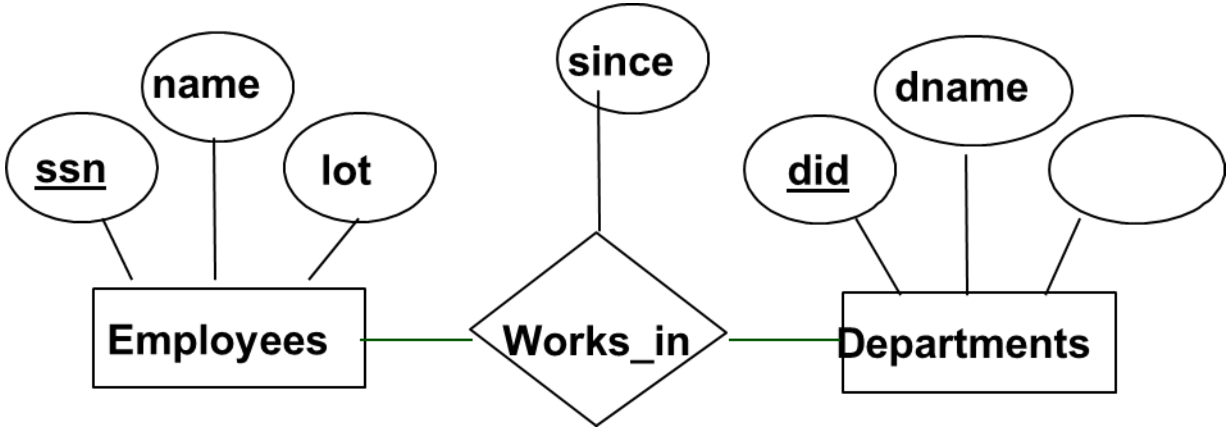
```
DROP VIEW goodStudents;
```



Many to many relationship

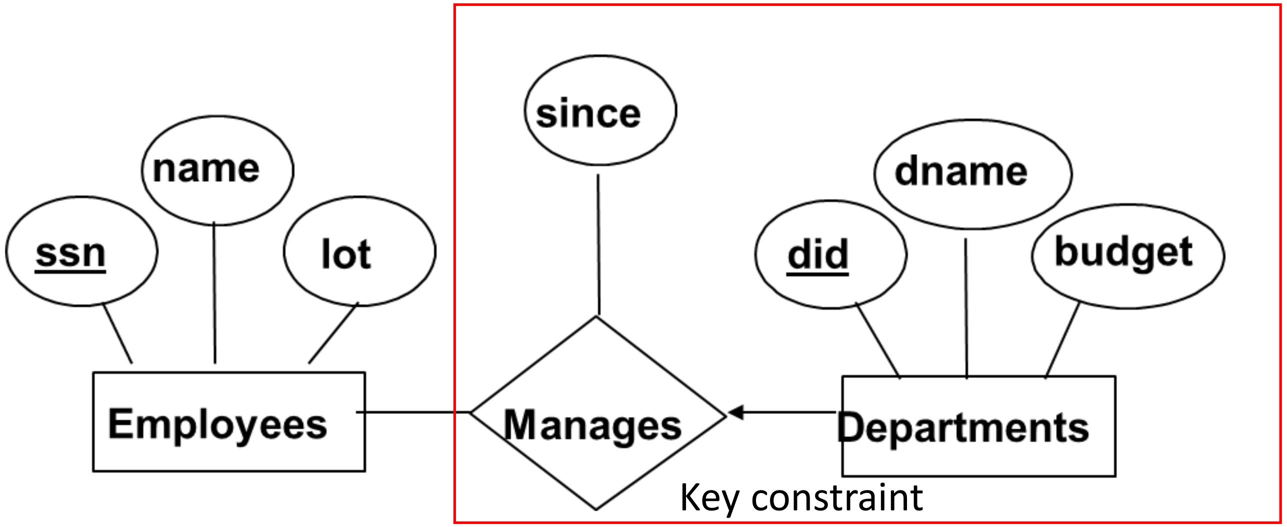
```
CREATE TABLE Employees(
ssn CHAR(11),
name CHAR(20),
lot  INTEGER,
PRIMARY KEY  (ssn));
```

```
CREATE TABLE Employees(
ssn CHAR(11),
name CHAR(20),
lot  INTEGER,
PRIMARY KEY  (ssn));
```



Many to many relationship

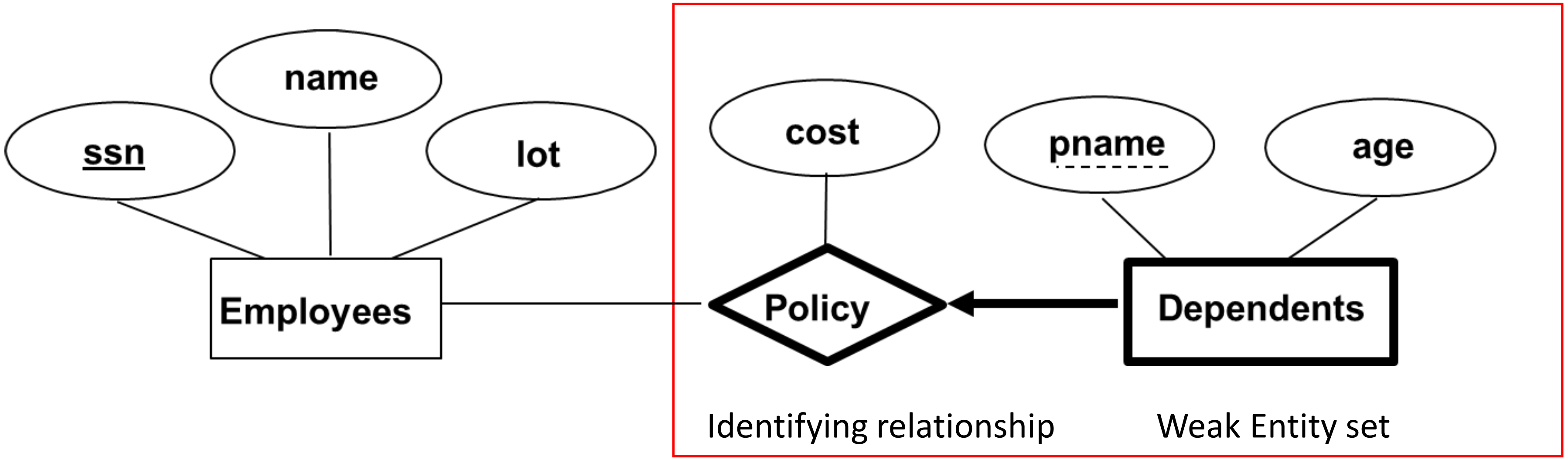
```
CREATE TABLE Works_in(
ssn  CHAR(11),
did  INTEGER,
since  DATE,
PRIMARY KEY (ssn, did),
FOREIGN KEY (ssn)        REFERENCES employees(ssn),
FOREIGN KEY (did)        REFERENCES departments(did));
```



Key constraint

```
CREATE TABLE Employees(
ssn CHAR(11),
name CHAR(20),
lot  INTEGER,
PRIMARY KEY  (ssn));
```

```
CREATE TABLE Dept_Manages(
did CHAR(11),
ssn CHAR(11),
dname CHAR (20),
budget INTEGER,
since DATE,
PRIMARY KEY (did),
FOREIGN KEY ssn REFERENCES Employees(ssn)
);
```

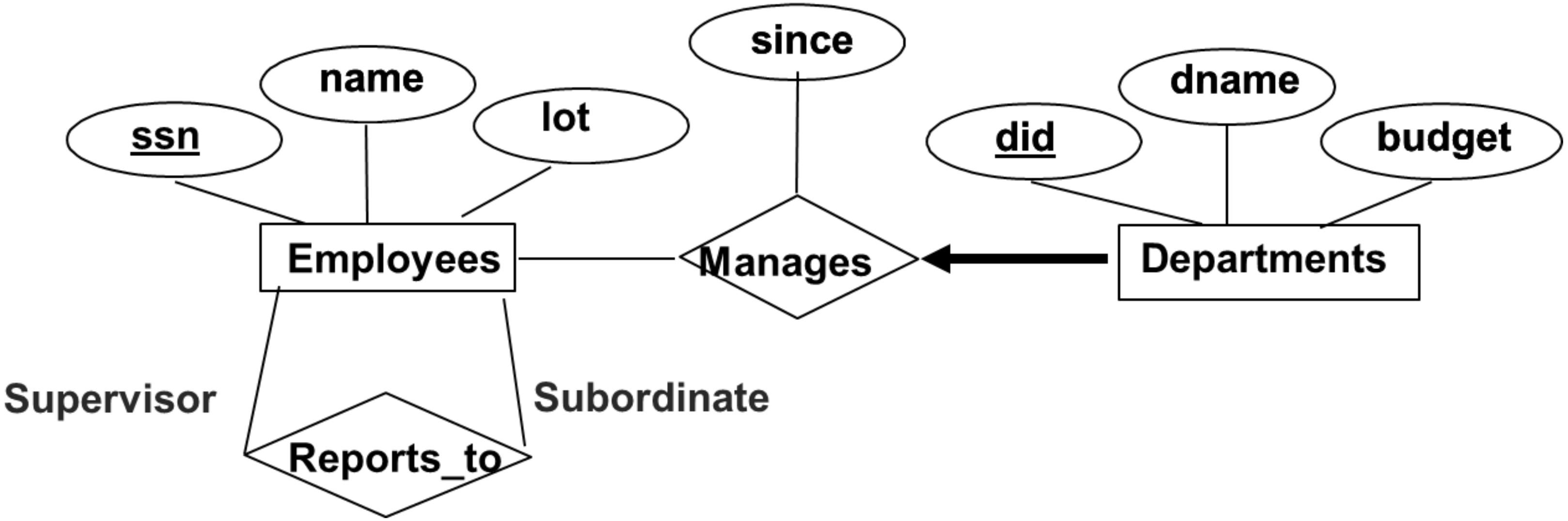


Identifying relationship

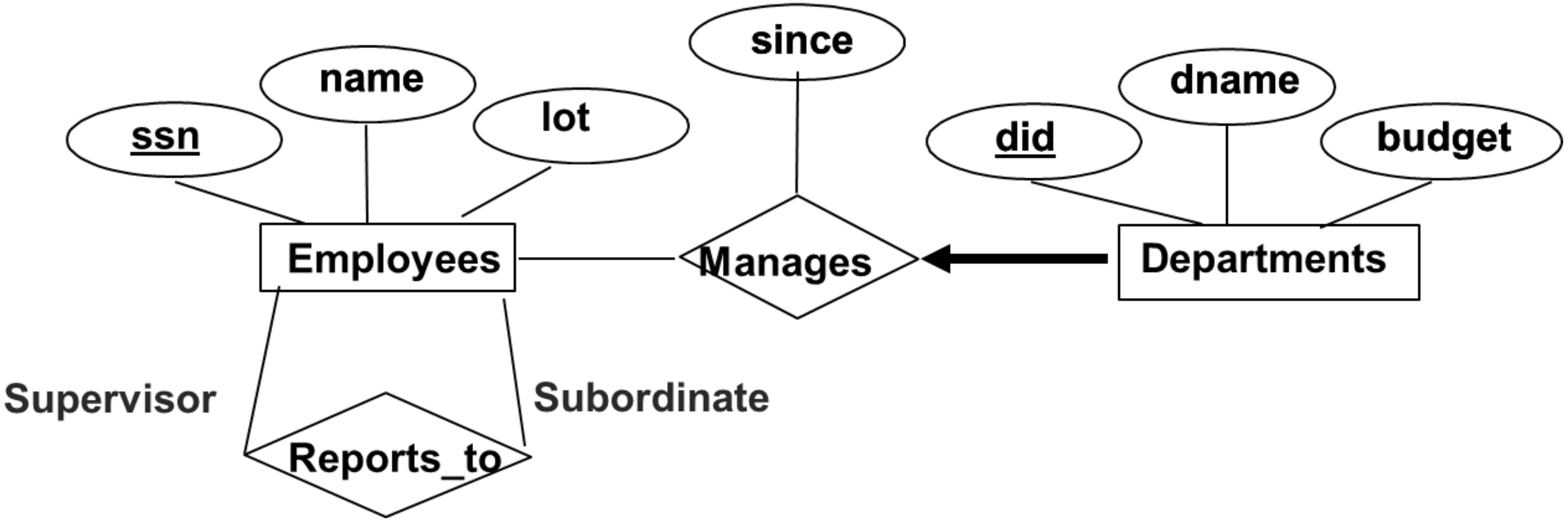
Weak Entity set

```
CREATE TABLE Employees(
ssn CHAR(11),
name CHAR(20),
lot  INTEGER,
PRIMARY KEY  (ssn));
```

```
CREATE TABLE Dep_Policy(
pname CHAR(11),
age INTEGER,
cost DECIMAL,
ssn CHAR(11),
PRIMARY KEY (ssn,pname)
FOREIGN KEY ssn REFERENCES Employees(ssn)  ON DELETE CASCADE
);
```



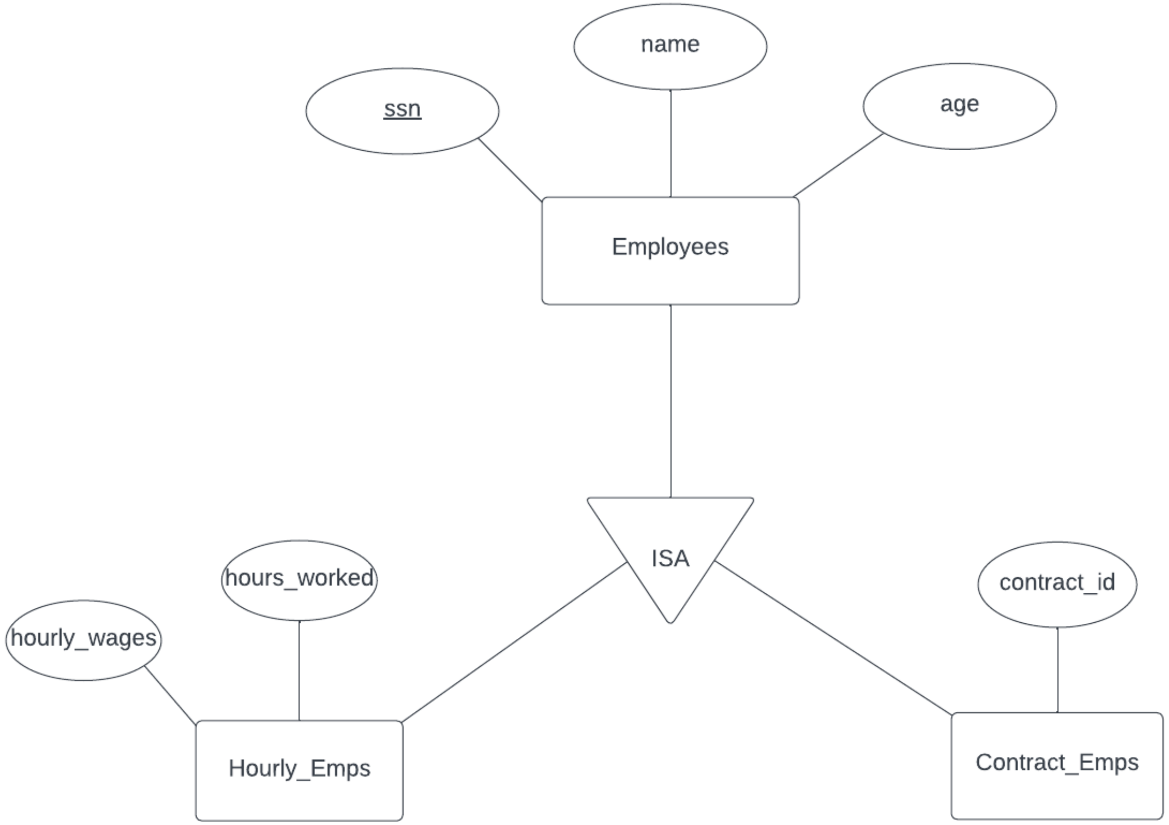
How many relations will there be, after we convert the following ER diagram to relational model?



```
CREATE TABLE Employees(
ssn CHAR(11),
name CHAR(20),
lot  INTEGER,
PRIMARY KEY  (ssn));
```

```
CREATE TABLE Reports_To(
supervisor_ssn CHAR(11),
subordinate_ssn CHAR(11),
PRIMARY KEY
(supervisor_ssn,subordinate_ssn),
FOREIGN KEY supervisor_ssn REFERENCES
Employees(ssn) ON DELETE CASCADE,
FOREIGN KEY subordinate_ssn REFERENCES
Employees(ssn) ON DELETE CASCADE,
);
```

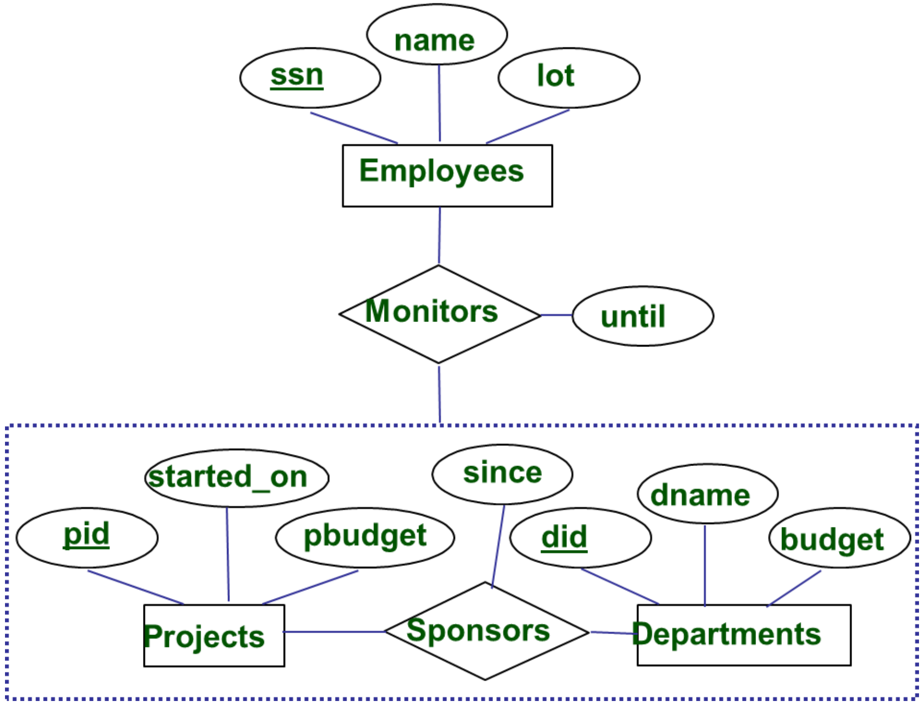
```
CREATE TABLE Dept_Manages(
did CHAR(11),
ssn CHAR(11) NOT NULL,
dname CHAR (20),
budget INTEGER,
since DATE,
PRIMARY KEY (did),
FOREIGN KEY ssn REFERENCE Employees(ssn));
```



```
CREATE TABLE Employees(
ssn CHAR(11),
name CHAR(20),
age INTEGER,
PRIMARY KEY (ssn)
)
```

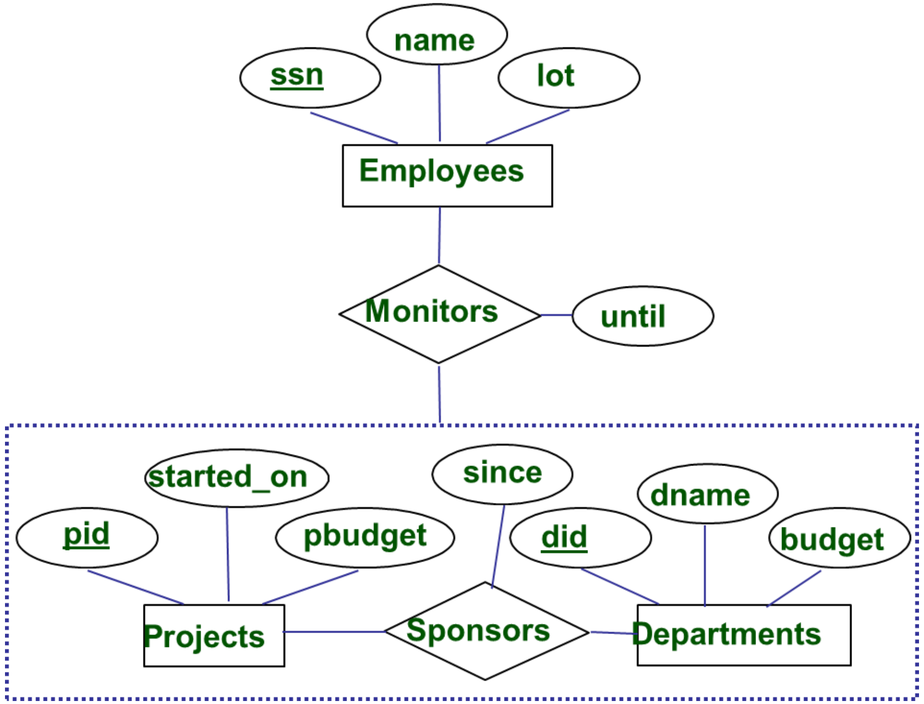
```
CREATE TABLE Hourly_Emps(
ssn CHAR(11),
Hours_worked INTEGER,
Hourly_wages DECIMAL,
PRIMAY KEY (ssn),
FOREIGN KEY (ssn) References
Employees(ssn) ON DELETE CASCADE
)
```

```
CREATE TABLE Contract_Emps(
ssn CHAR(11),
contract_id INTEGER,
PRIMAY KEY (ssn),
FOREIGN KEY (ssn) References
Employees(ssn) ON DELETE CASCADE
)
```



- Have to record the descriptive attributes of Sponsors relationship.
- Not every sponsorship has a monitor, some (pid, did) pairs in the
- Sponsors relation may not appear in the Monitors relation.

```
CREATE TABLE Employees(
ssn CHAR(11),
name CHAR(20),
lot INTEGER,
PRIMARY KEY (ssn)
)
```



```
CREATE TABLE Monitors(
ssn CHAR(11),
pid CHAR(11),
did CHAR(11),
until DATE,
PRIMARY KEY (ssn,pid,did)
FOREIGN KEY ssn REFERENCES
Employees,
FOREIGN KEY pid REFERENCES
Projects,
FOREIGN KEY did REFERENCES
Departments
)
```

```
CREATE TABLE Projects(
pid CHAR(11),
started_on DATE,
Pbudget DECIMAL,
PRIMARY KEY (pid)
)
```

```
CREATE TABLE Sponsors(
pid CHAR(11),
did CHAR(11),
Since DATE,
PRIMARY KEY (pid,did),
FOREIGN KEY pid REFERENCES Projects(pid),
FOREIGN KEY (did) REFERENCES
Departments(did)
)
```

```
CREATE TABLE Departments(
did CHAR(11),
dname CHAR(20),
budget DECIMAL,
PRIMARY KEY (pid)
)
```