CMPSC 174A/174N Fundamentals of Database System

ER Diagrams and Relational Tables

Discussion Session
Friday, 9:00am-9:50am
Zexi Huang

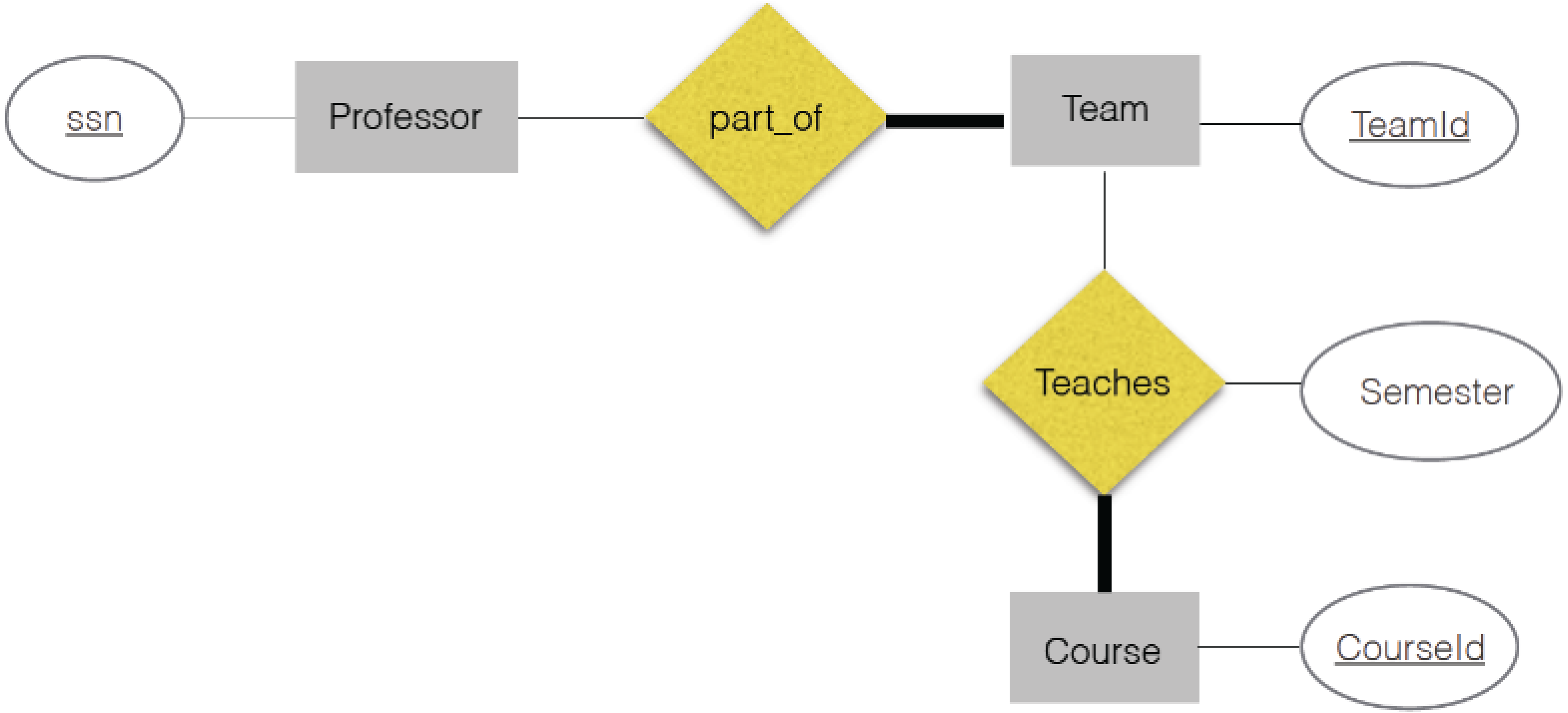
Session 4

Schedule

- Discussion on homework #1
  - Total participation
  - Multiple records
  - Hierarchy constraints
  - Equivalence constraints
- Preview Course Project

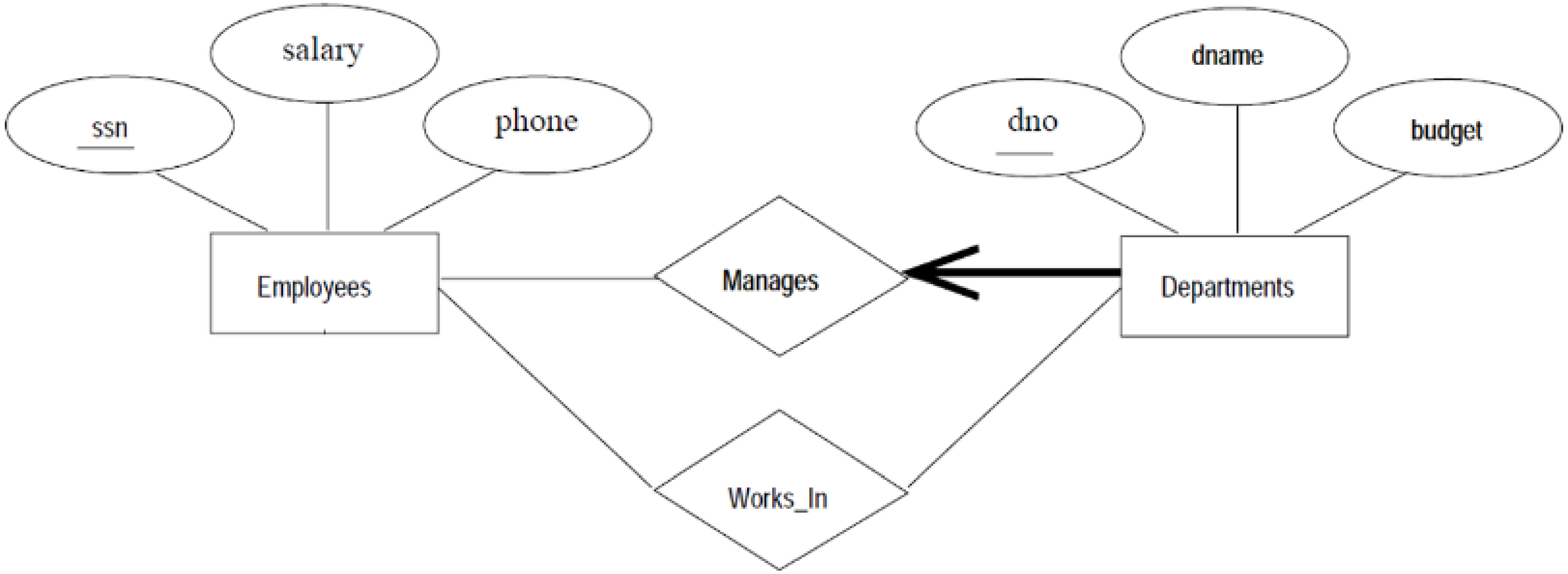
Total Participation

- **Is there a total participation?**
  - All courses are taught by team(s) of professor(s), and each team consists of at least one professor (Ex 2.2).



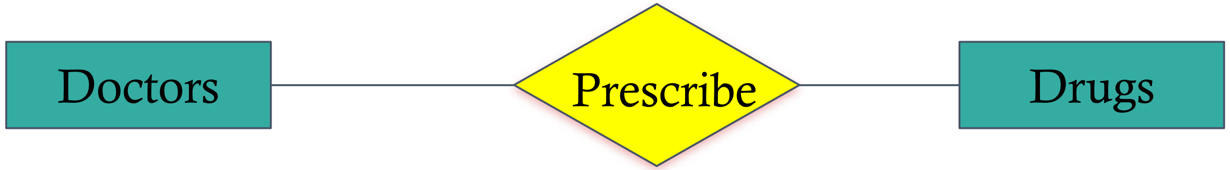
Total Participation

- **Is there a total participation?**
  - All courses are taught by team(s) of professor(s), and each team consists of at least one professor (Ex 2.2).
  - Employees work in departments; each department is managed by an employee (Ex 2.4).



Total Participation

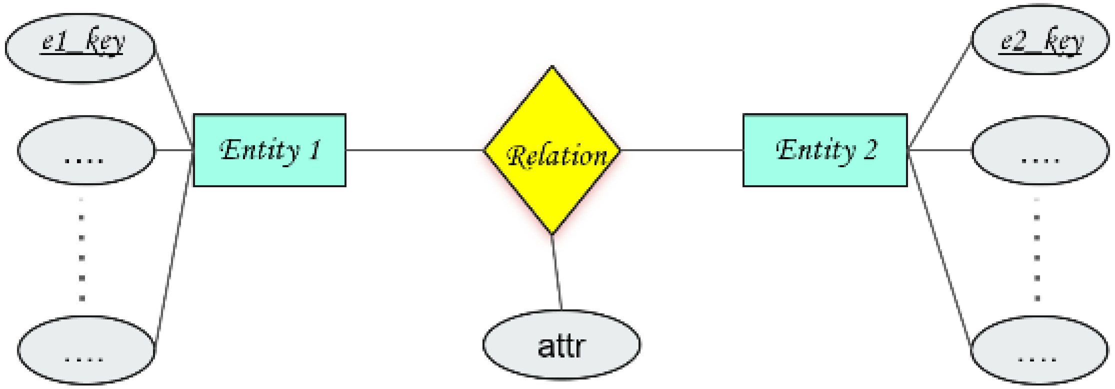
- **Is there a total participation?**
  - All courses are taught by team(s) of professor(s), and each team consists of at least one professor (Ex 2.2).
  - Employees work in departments; each department is managed by an employee (Ex 2.4).
  - Doctors prescribe drugs (Ex 2.7).

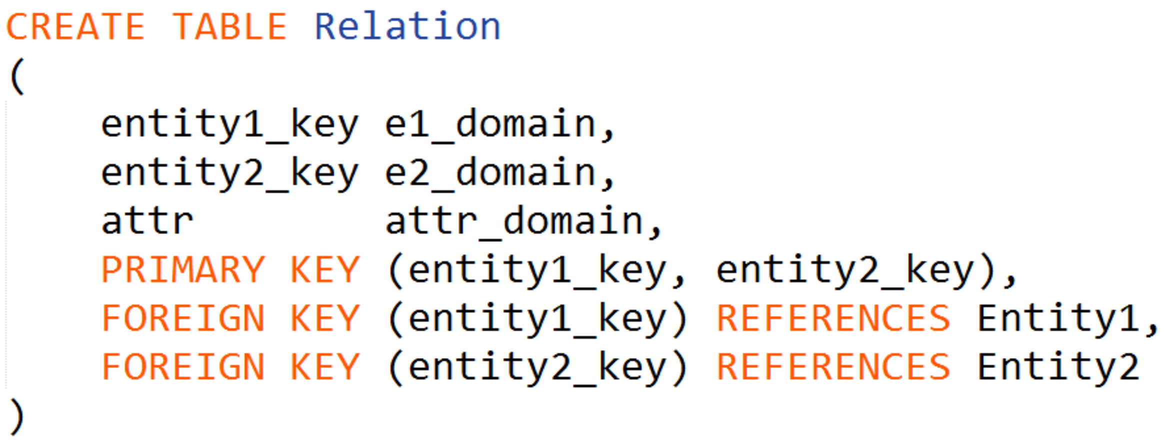


Total Participation

- **Is there a total participation?**
  - All courses are taught by team(s) of professor(s), and each team consists of at least one professor (Ex 2.2).
  - Employees work in departments; each department is managed by an employee (Ex 2.4).
  - Doctors prescribe drugs (Ex 2.7).
- **Criteria for total participation:**
  - Must, should. *Employees* **must** *work in departments.*
  - At least one, one or more, exactly one. *Employees work in* **at least one** *department.*
  - Every, each, all. **Every** *employee works in departments.*
- **Criteria for partial participation:**
  - No explicit constraints. *Employees work in departments.*
  - Can, could. *Employees* **can** *work in* **one or more** *departments.*

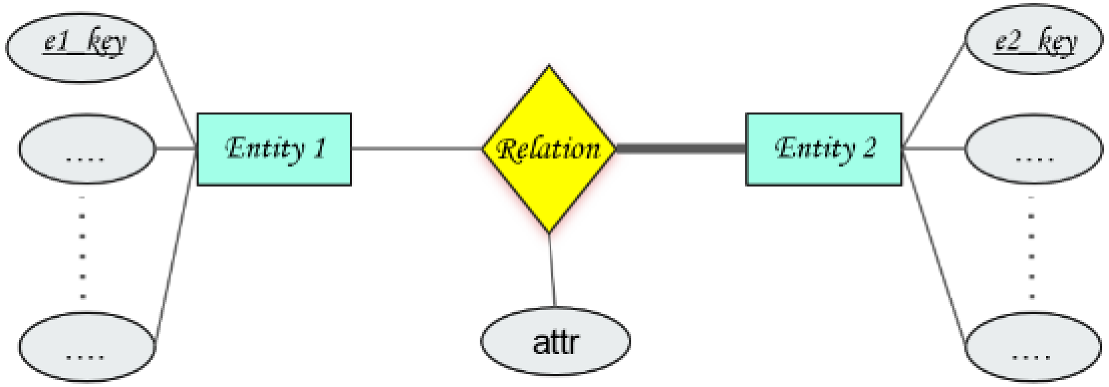
Total Participation



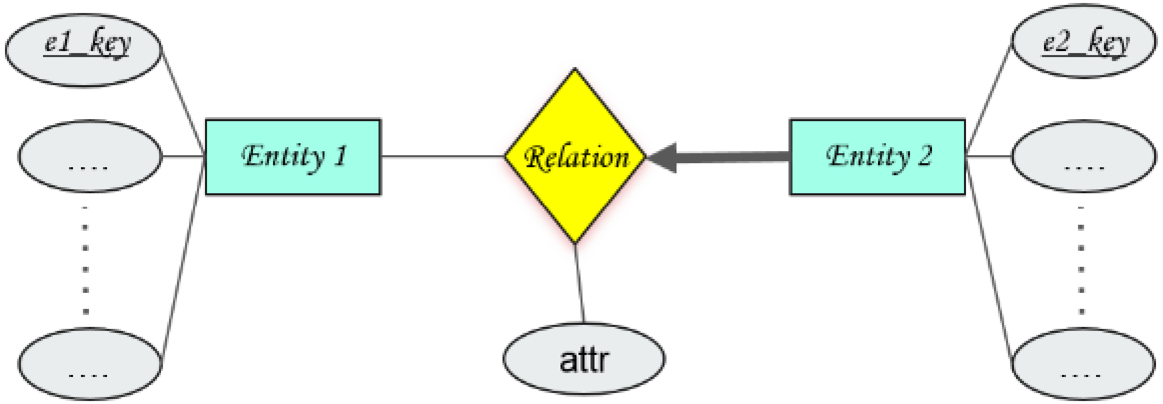


```
CREATE TABLE Relation
(
    entity1_key e1_domain,
    entity2_key e2_domain,
    attr        attr_domain,
    PRIMARY KEY (entity1_key, entity2_key),
    FOREIGN KEY (entity1_key) REFERENCES Entity1,
    FOREIGN KEY (entity2_key) REFERENCES Entity2
)
```

Total Participation

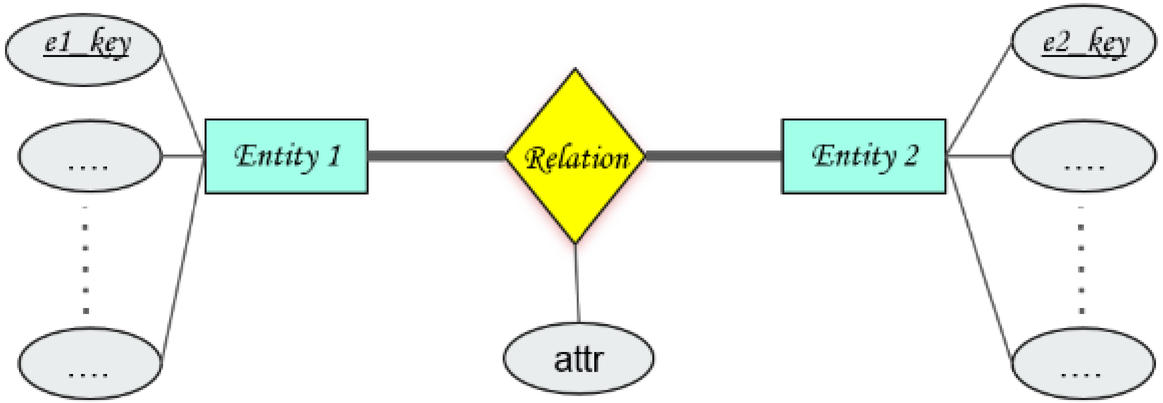


```
CREATE TABLE Relation
(
    entity1_key e1_domain,
    entity2_key e2_domain,
    attr        attr_domain,
    PRIMARY KEY (entity1_key, entity2_key),
    FOREIGN KEY (entity1_key) REFERENCES Entity1,
    FOREIGN KEY (entity2_key) REFERENCES Entity2,
    CHECK (...) --All entity2_key must appear in table
)
```

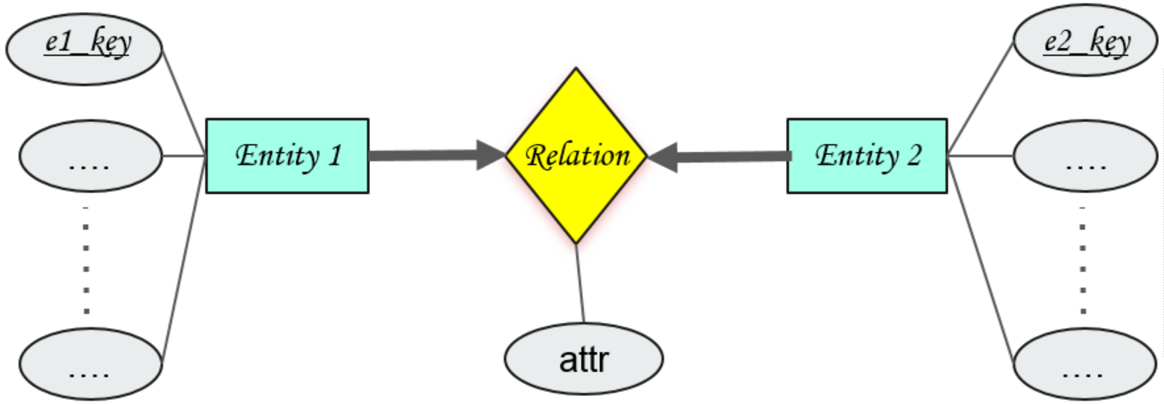


```
CREATE TABLE Entity2_Rel
(
    entity2_key e2_domain,
    attr        attr_domain,
    attr2       attr2_domain,
    entity1_key e1_domain NOT NULL,
    PRIMARY KEY (entity2_key),
    FOREIGN KEY (entity1_key) REFERENCES Entity_1
)
```

Total Participation



```
CREATE TABLE Relation
(
    entity1_key     e1_domain,
    entity2_key     e2_domain,
    attr            attr_domain,
    PRIMARY KEY (entity1_key, entity2_key),
    FOREIGN KEY (entity1_key) REFERENCES Entity1,
    FOREIGN KEY (entity2_key) REFERENCES Entity2,
    CHECK (...) --All entity1,2 keys must appear in table
)
```



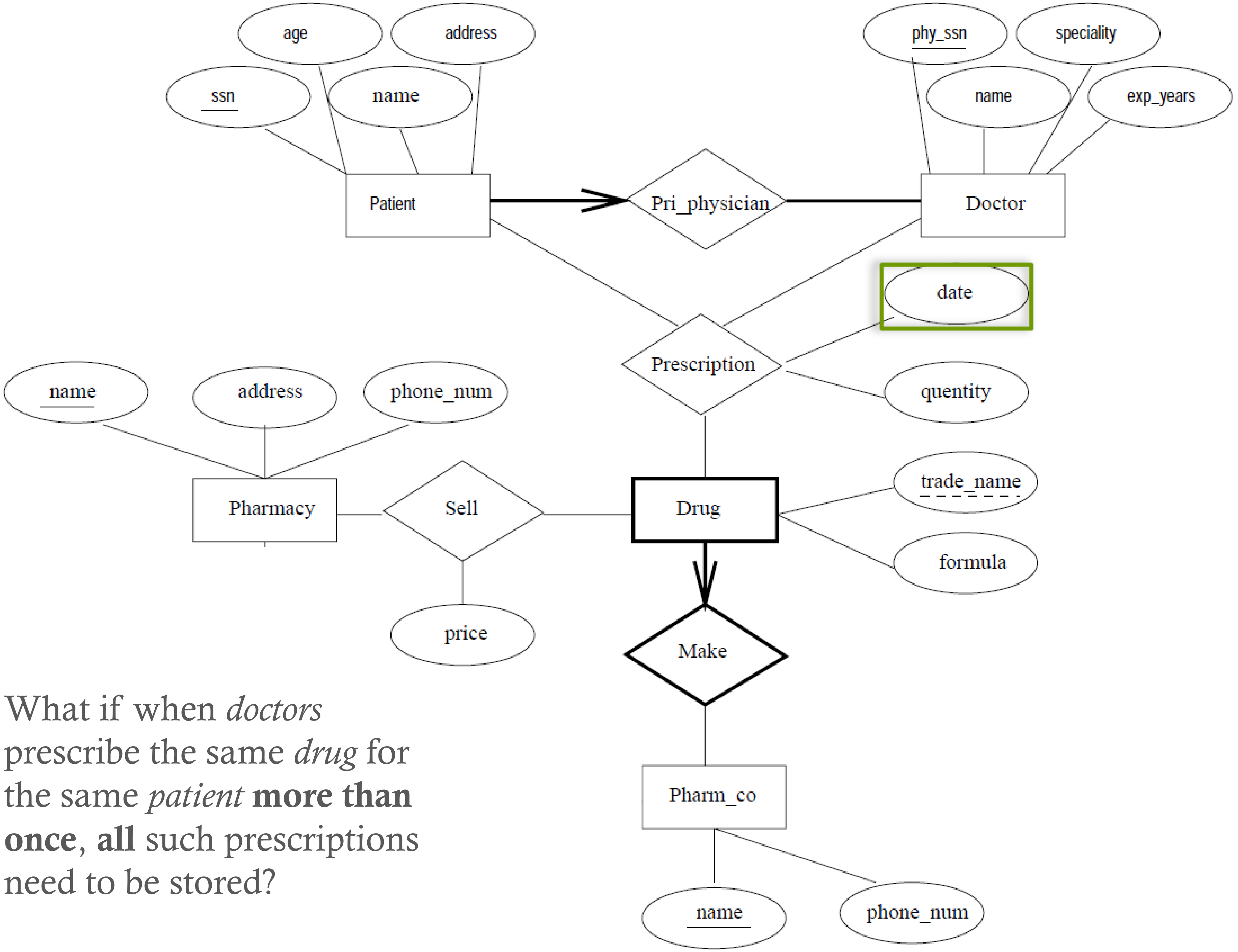
```
CREATE TABLE Relation
(
    entity1_key     e1_domain,
    entity2_key     e2_domain NOT NULL UNIQUE,
    attr            attr_domain,
    attr1           attr1_domain,
    attr2           attr2_domain,
    PRIMARY KEY (entity1_key)--or entity2_key
)
```

Multiple Records

- **Exercise 2.6** The FAA requires the airport to keep track of **each time** a given airplane is tested by a given technician using a given test. For each testing event, the information needed is the date, the number of hours the technician spent doing the test, and the score the airplane received on the test.

Exercise 2.7

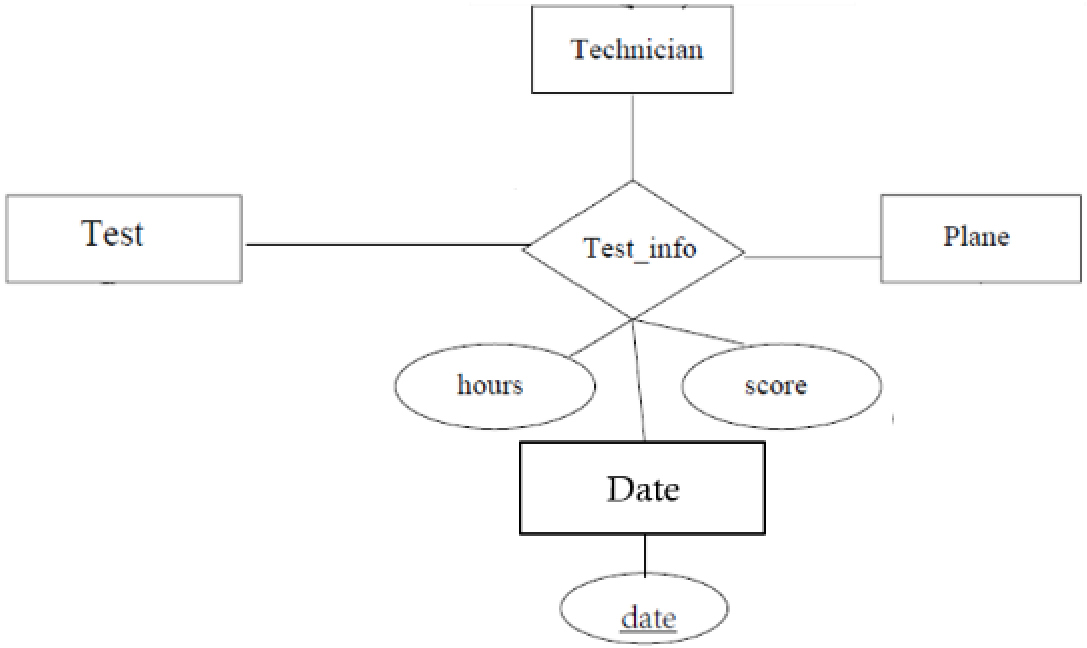
- *Doctors* prescribe *drugs* for *patients*.

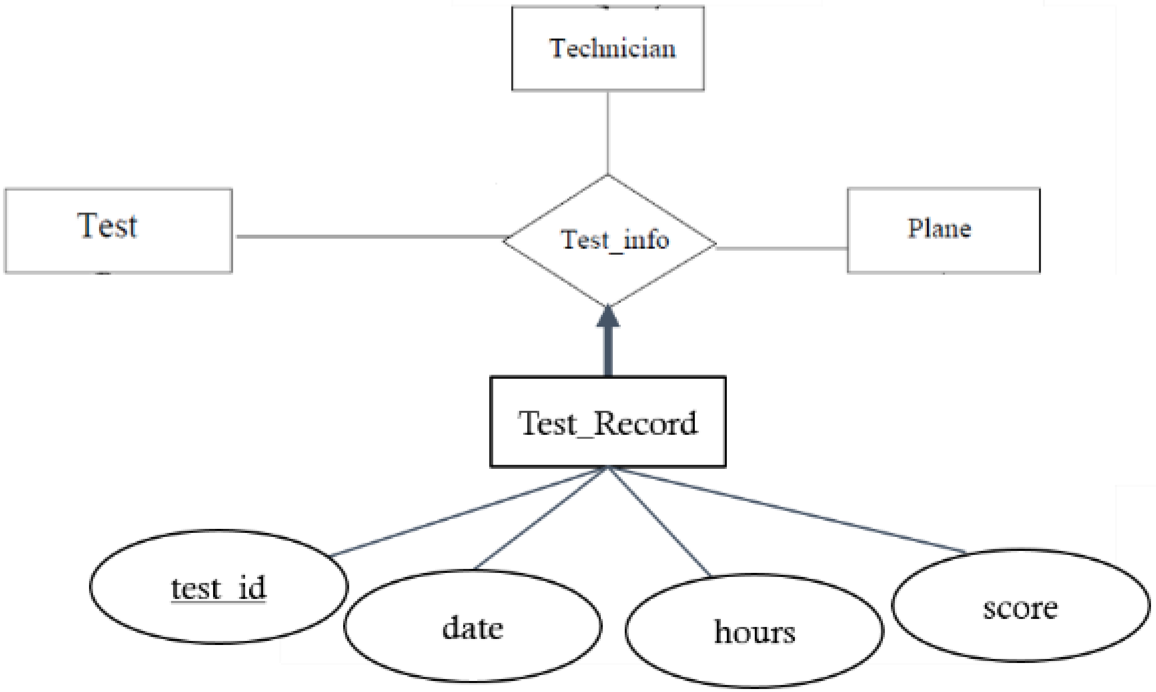


- What if when *doctors* prescribe the same *drug* for the same *patient* **more than once**, **all** such prescriptions need to be stored?

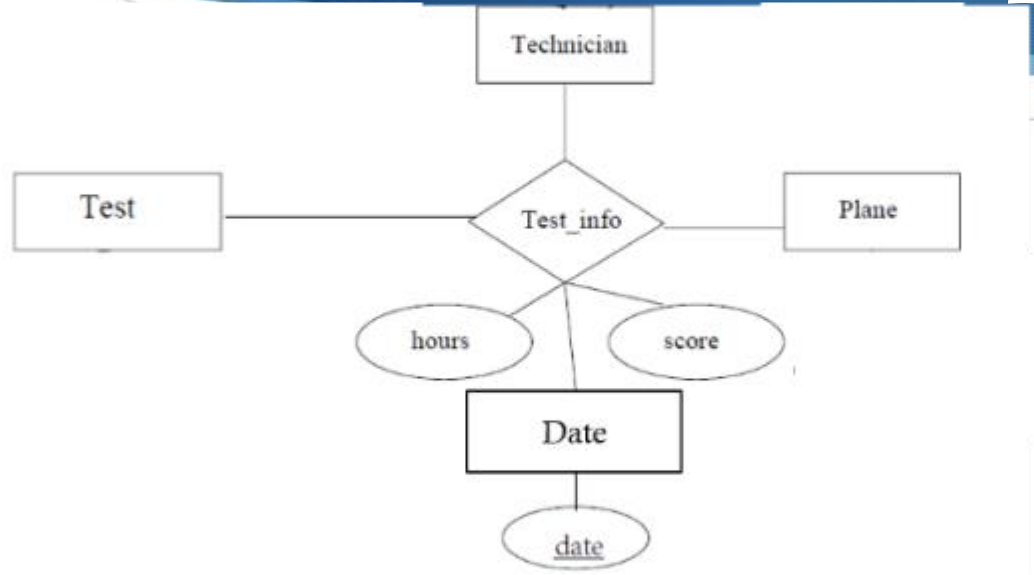
Multiple Records

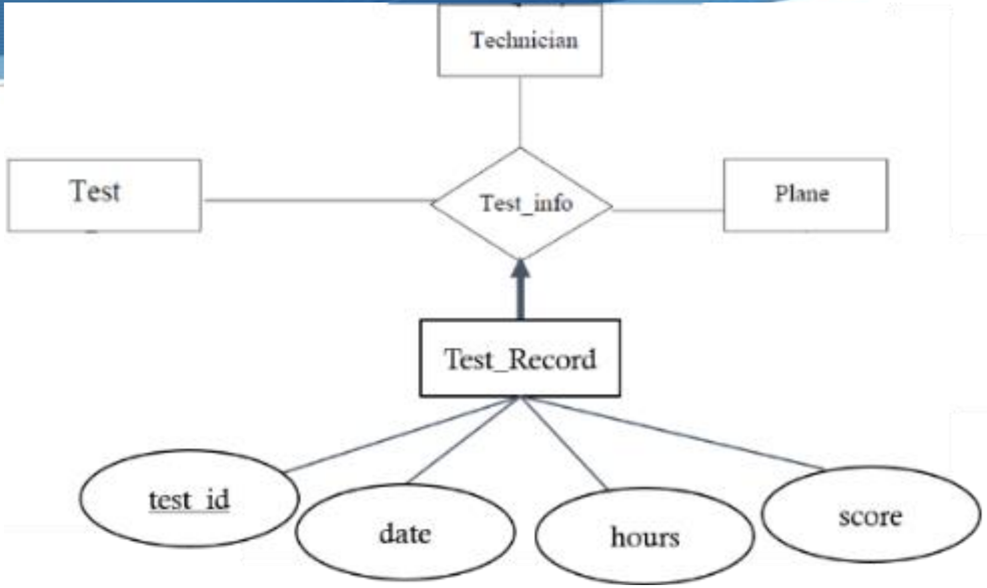
- **Exercise 2.6** The FAA requires the airport to keep track of **each time** a given airplane is tested by a given technician using a given test. For each testing event, the information needed is the date, the number of hours the technician spent doing the test, and the score the airplane received on the test.





Multiple Records





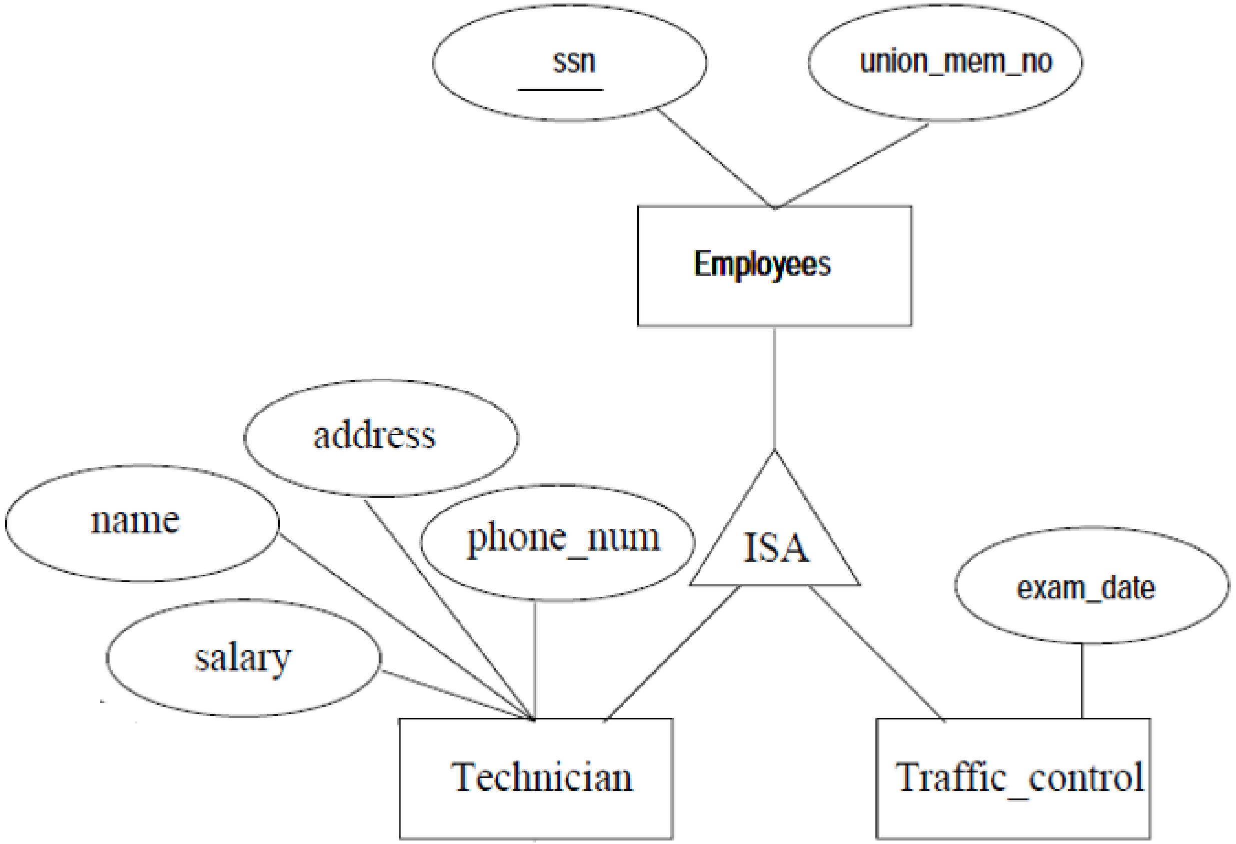
```
CREATE TABLE Test_info
(
    test_no     CHAR(20),
    ssn         CHAR(20),
    plane_no    CHAR(20),
    the_date    DATE,
    hours       INTEGER,
    score       INTEGER,
    PRIMARY KEY (test_no,ssn,plane_no,the_date),
    FOREIGN KEY (test_no) REFERENCES Test,
    FOREIGN KEY (ssn) REFERENCES Technician,
    FOREIGN KEY (plane_no) REFERENCES Plane,
    ~~FOREIGN KEY (the_date) REFERENCES Date_entity~~
)
```

```
CREATE TABLE Test_info
(
    test_id     CHAR(20),
    test_no     CHAR(20) NOT NULL,
    ssn         CHAR(20) NOT NULL,
    plane_no    CHAR(20) NOT NULL,
    the_date    DATE,
    hours       INTEGER,
    score       INTEGER,
    PRIMARY KEY (test_id),
    FOREIGN KEY (test_no) REFERENCES Test,
    FOREIGN KEY (ssn) REFERENCES Technician,
    FOREIGN KEY (plane_no) REFERENCES Plane
)
```

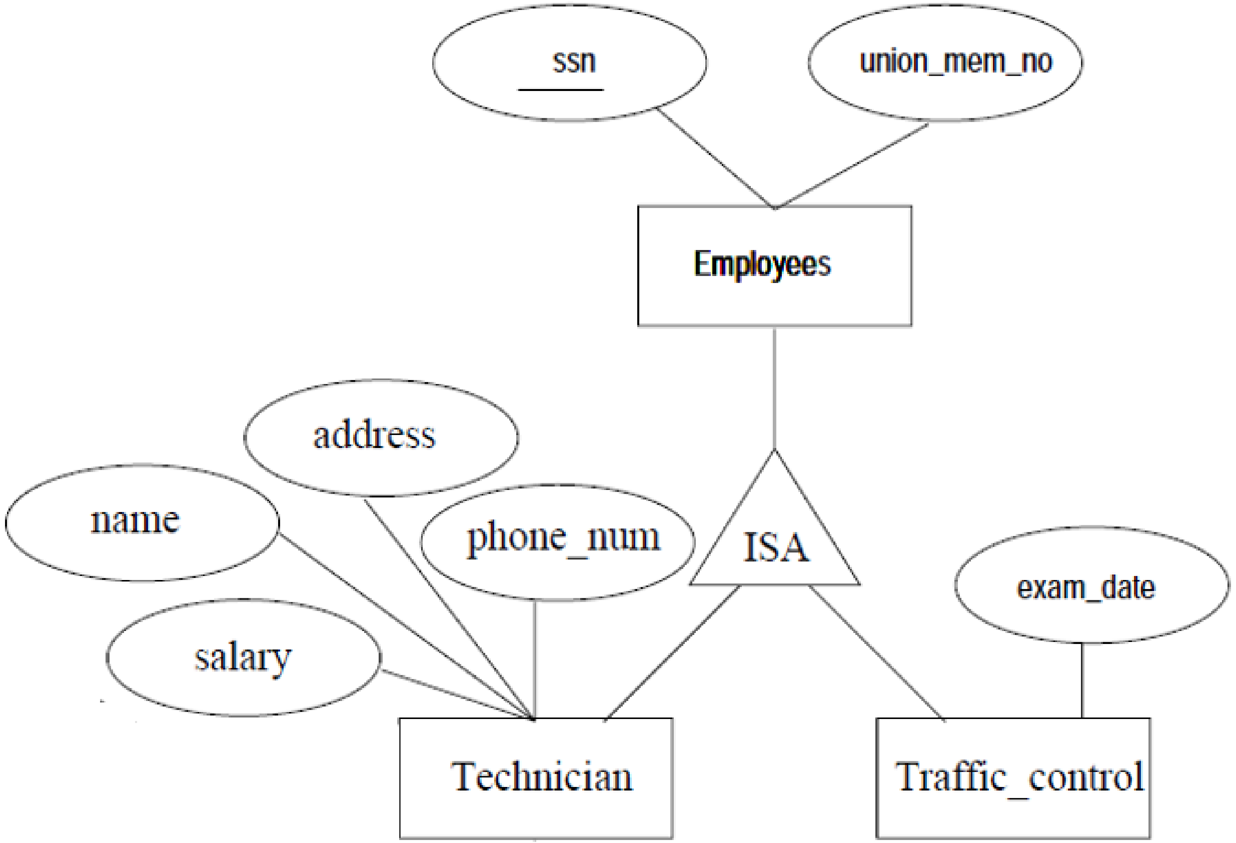
Hierarchy Constraints

- **Exercise 2.6.1 Specify any necessary covering and overlap constraints as well (in English).**
  - A number of technicians work at the airport. You need to store the name, SSN, address, phone number, and salary of each technician.
  - Traffic controllers must have an annual medical examination. You must store the date of the most recent exam.
  - All airport employees (including technicians) belong to a union. You must store the union membership number of each employee. Assume that each employee is uniquely identified by SSN.

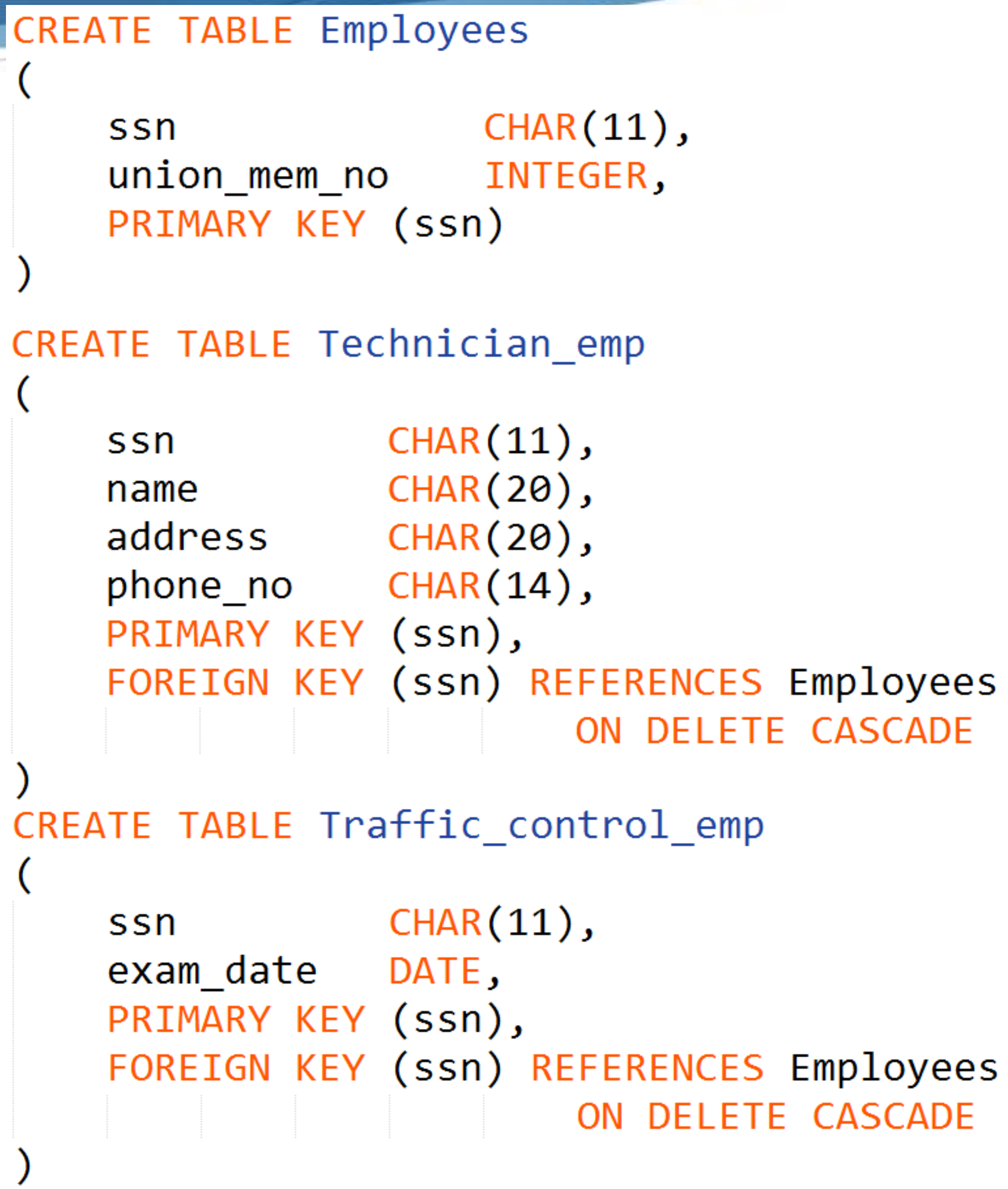
- **Covering: subclasses collectively include all entities in superclass**
- **Overlap: Subclasses contain the same entity**



Hierarchy Constraints



- Can we use two tables to capture the class hierarchy relationship?

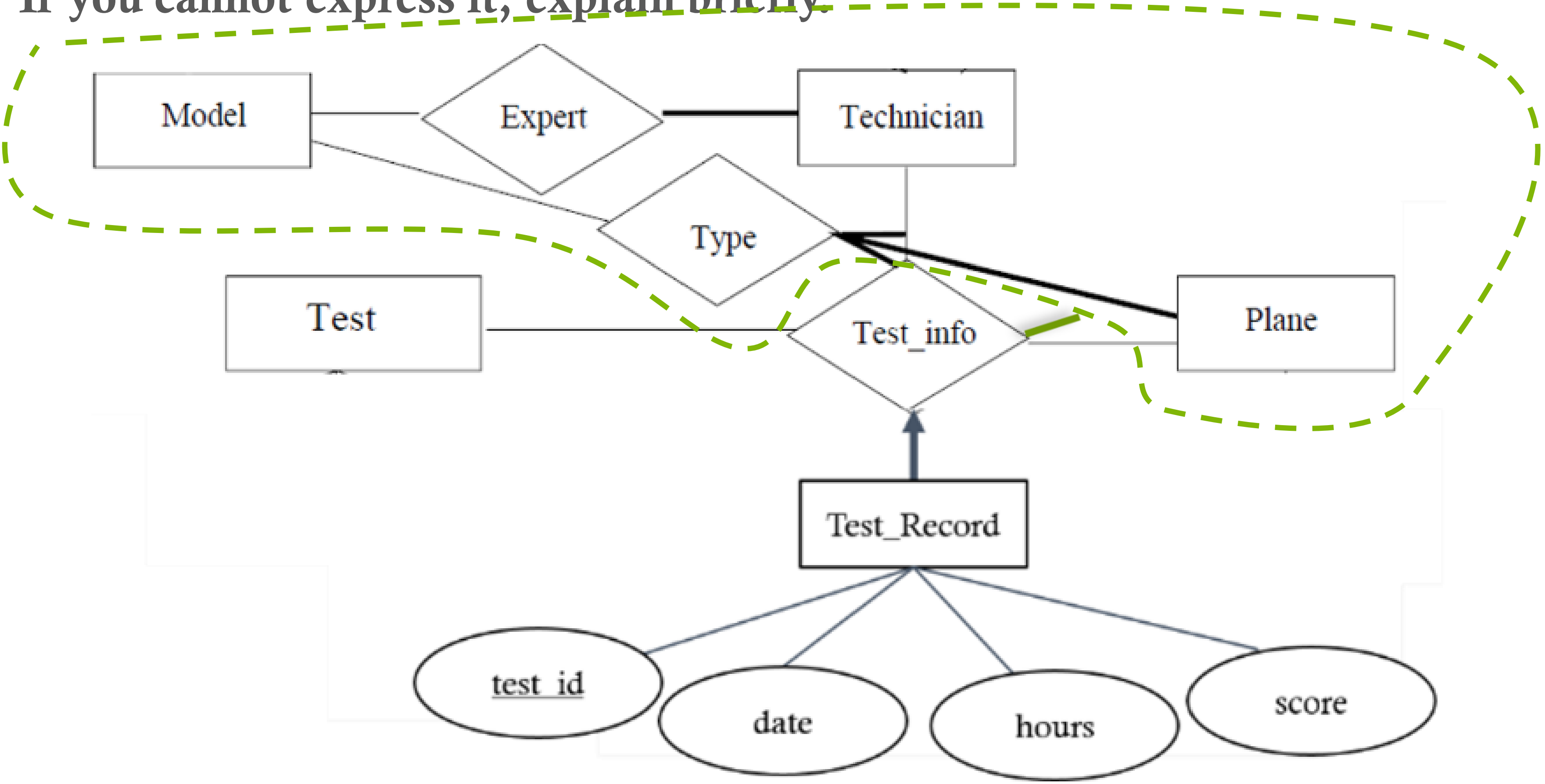


```
CREATE TABLE Employees
(
    ssn                CHAR(11),
    union_mem_no       INTEGER,
    PRIMARY KEY (ssn)
)

CREATE TABLE Technician_emp
(
    ssn           CHAR(11),
    name          CHAR(20),
    address       CHAR(20),
    phone_no      CHAR(14),
    PRIMARY KEY (ssn),
    FOREIGN KEY (ssn) REFERENCES Employees
                        ON DELETE CASCADE
)
CREATE TABLE Traffic_control_emp
(
    ssn           CHAR(11),
    exam_date     DATE,
    PRIMARY KEY (ssn),
    FOREIGN KEY (ssn) REFERENCES Employees
                        ON DELETE CASCADE
)
```

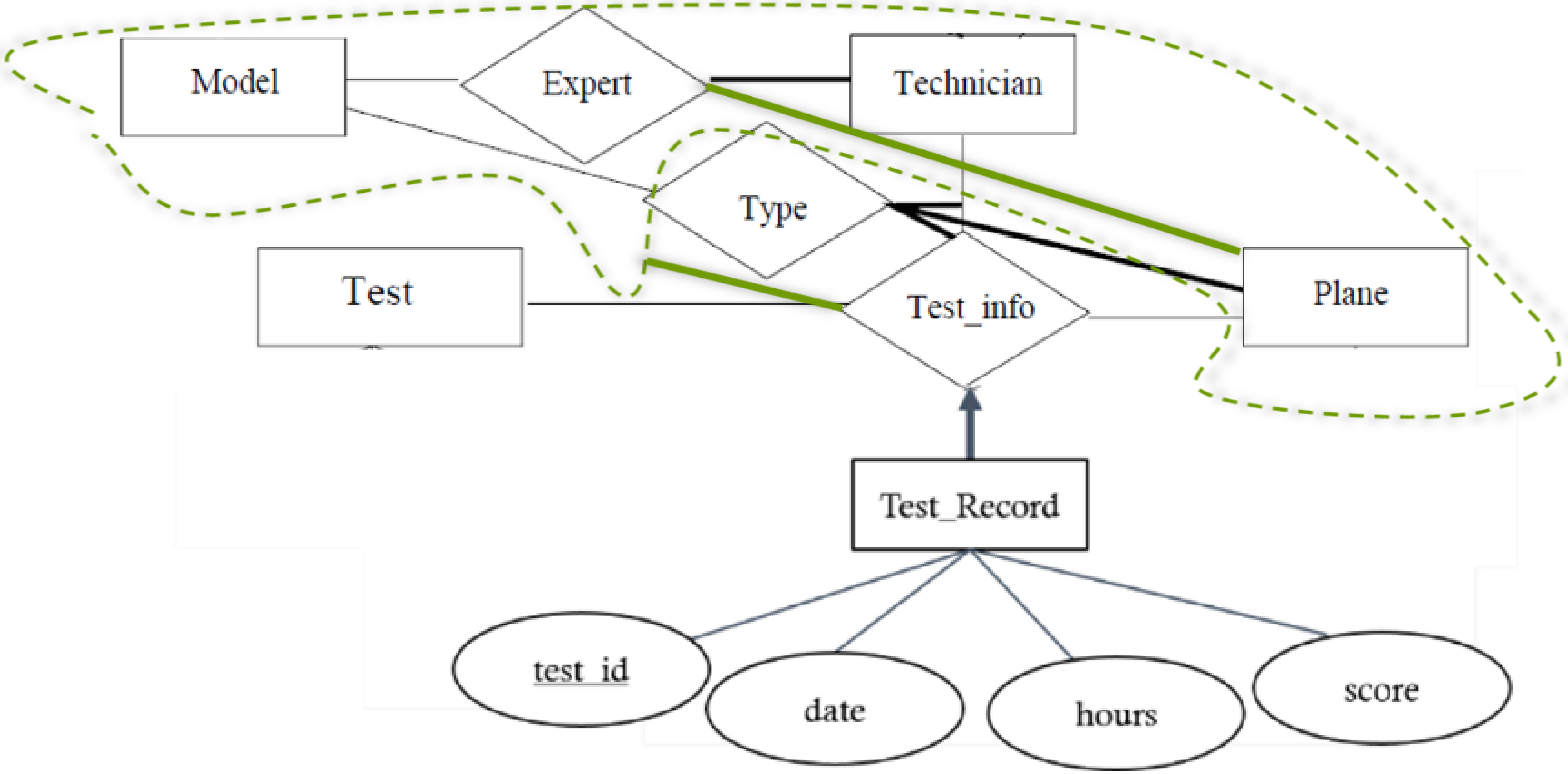
Equivalence Constraints

- **Exercise 2.6.2 The FAA passes a regulation that tests on a plane must be conducted by a technician who is an expert on that model. How would you express this constraint in the ER diagram? If you cannot express it, explain briefly.**



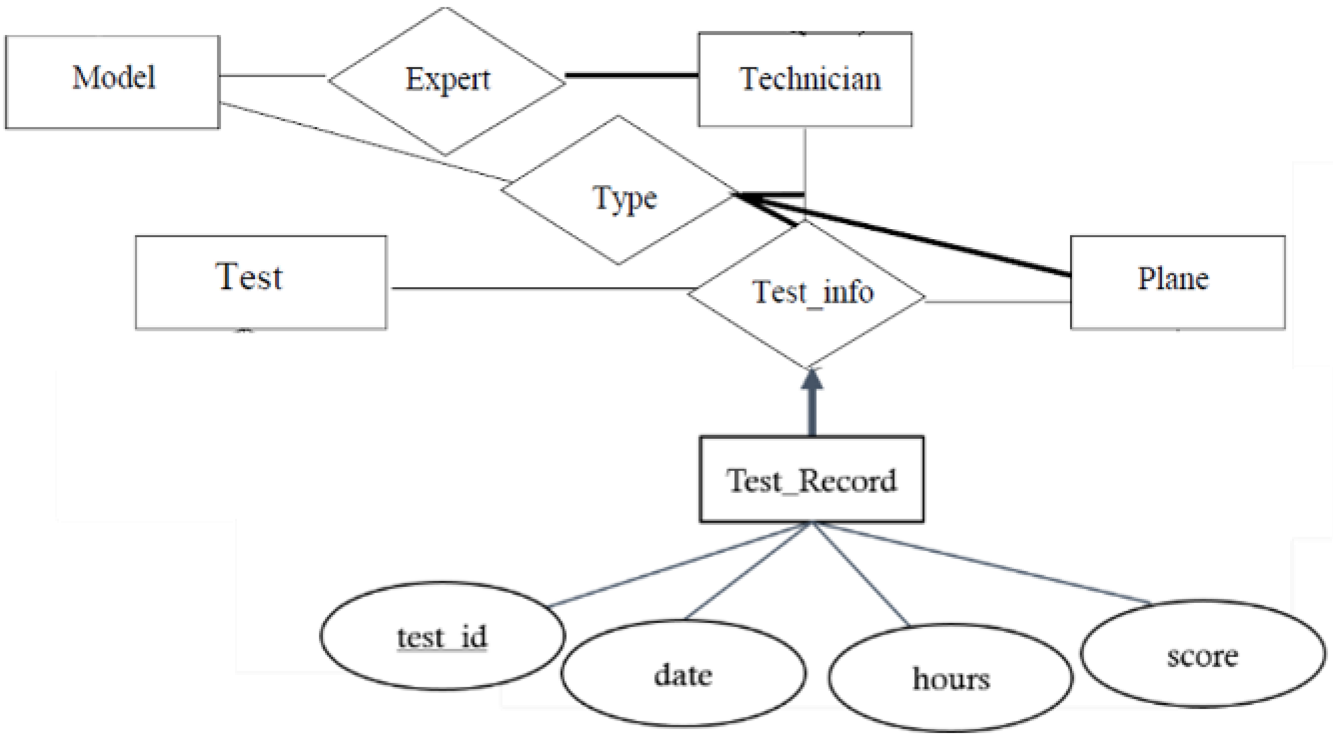
Equivalence Constraints

- **Exercise 2.6.2 The FAA passes a regulation that tests on a plane must be conducted by a technician who is an expert on that model. How would you express this constraint in the ER diagram? If you cannot express it, explain briefly.**



Equivalence Constraints

- **Exercise 3.16.2 Use SQL statements to capture the equivalence constraint.**



```
CREATE TABLE Test_info
(
    test_id     CHAR(20),
    test_no     CHAR(20) NOT NULL,
    ssn         CHAR(20) NOT NULL,
    plane_no    CHAR(20) NOT NULL,
    the_date    DATE,
    hours       INTEGER,
    score       INTEGER,
    PRIMARY KEY (test_id),
    FOREIGN KEY (test_no) REFERENCES Test,
    FOREIGN KEY (ssn) REFERENCES Technician,
    FOREIGN KEY (plane_no) REFERENCES Plane,
)
CONSTRAINT Model
    CHECK
    (
        SELECT * FROM Expert, Type
        WHERE Expert.ssn = ssn AND
        Type.plane_no = plane_no AND
        Expert.model_no = Type.model_no
    )
```

Course Project

- **Bank Management System:**
  - Maintaining balance information for all customer accounts and customers themselves,
  - Processing transactions (deposits, withdrawals, payments, etc.),
  - Generating monthly reports and updating accounts with monthly interest, and
  - ATM-App and bank teller interfaces (GUIs).
- **Milestones:**
  - Early project design report (ER diagrams, relational tables, constraints) due Nov 2.
  - Half an hour demo on CSIL computers (running Linux) with sample data entered by you.
    - Early demo at 9th week, Nov 26-30, 10% extra credits.
    - Regular demo at 10th week, Dec 3-7.