

Chapter 6

Implementing Models

Outline

- Implementing Models
- Implementing 1-M Relationships
- Implementing M-N Relationships

ERD to Tables

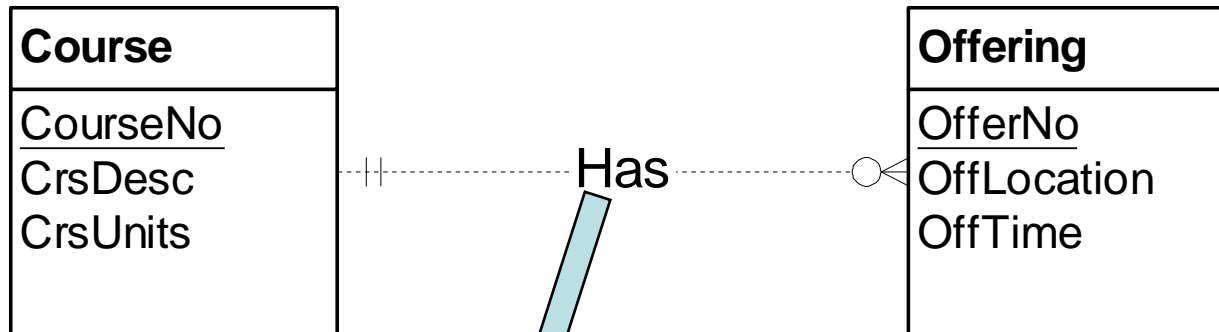
- An ERD is the design for a relational database design
- To implement an ERD, define tables in the database that reflect the structure of the ERD
- Basic process:
 1. Convert each entity to a table
 2. Implement ERD relationships using foreign keys

Implementing Relationships

■ 1-M Relationships

1. Add the primary key from the "1" side to the table on the "Many" side
2. Define the new column as a foreign key
3. Make the column NOT NULL if the "1" side is "1 and only 1"

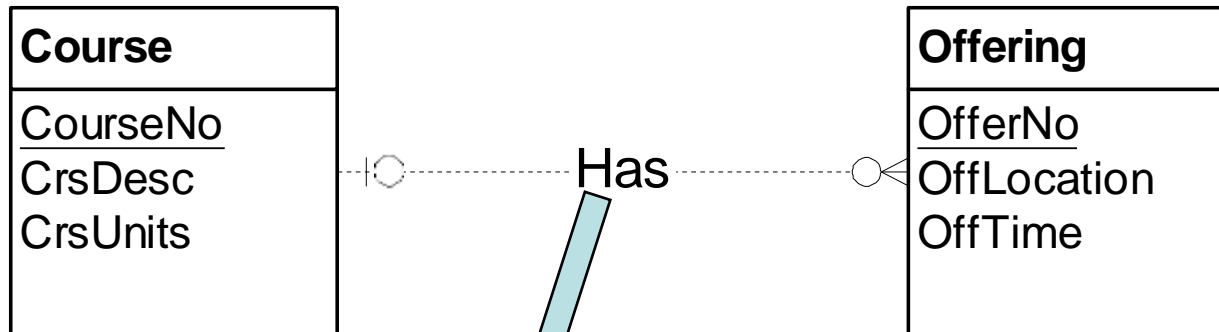
1-M Relationships



```
CREATE TABLE Course ( CourseNo INTEGER PRIMARY KEY,  
    CrsDesc VARCHAR(80),  
    CrsUnits NUMERIC(2,2) );
```

```
CREATE TABLE Offering ( OfferNo INTEGER PRIMARY KEY,  
    OffLocation VARCHAR(12),  
    OffTime VARCHAR(12),  
    CourseNo INTEGER NOT NULL,  
    FOREIGN KEY (CourseNo) REFERENCES Course(CourseNo) )
```

1-M Relationships



```
CREATE TABLE Course ( CourseNo INTEGER PRIMARY KEY,  
    CrsDesc VARCHAR(80),  
    CrsUnits NUMERIC(2,2) );
```

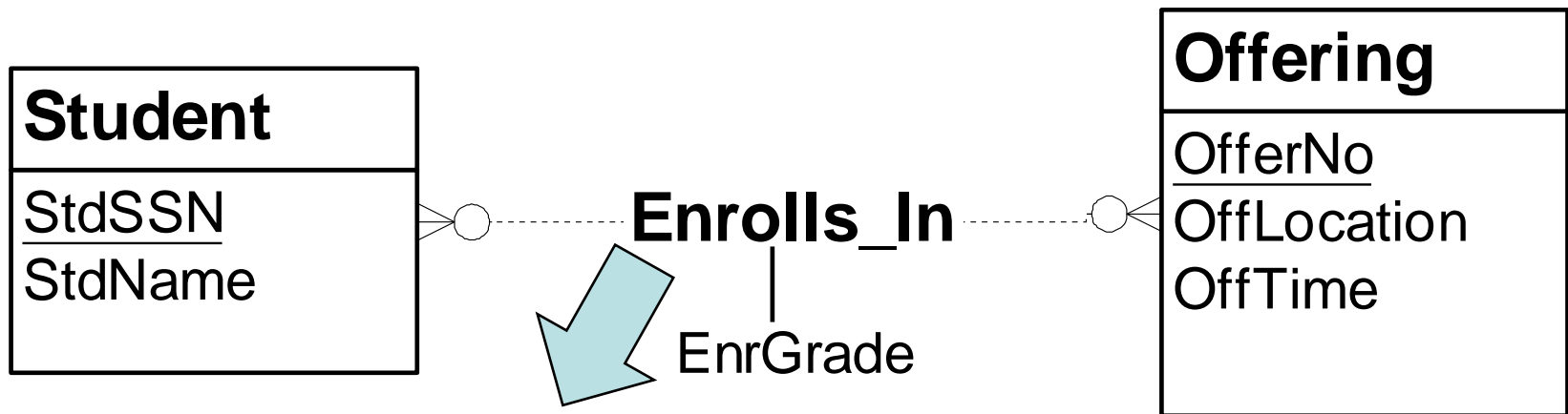
```
CREATE TABLE Offering ( OfferNo INTEGER PRIMARY KEY,  
    OffLocation VARCHAR(12),  
    OffTime VARCHAR(12),  
    CourseNo INTEGER,  
    FOREIGN KEY (CourseNo) REFERENCES Course(CourseNo) )
```

Implementing Relationships

■ M-N Relationships

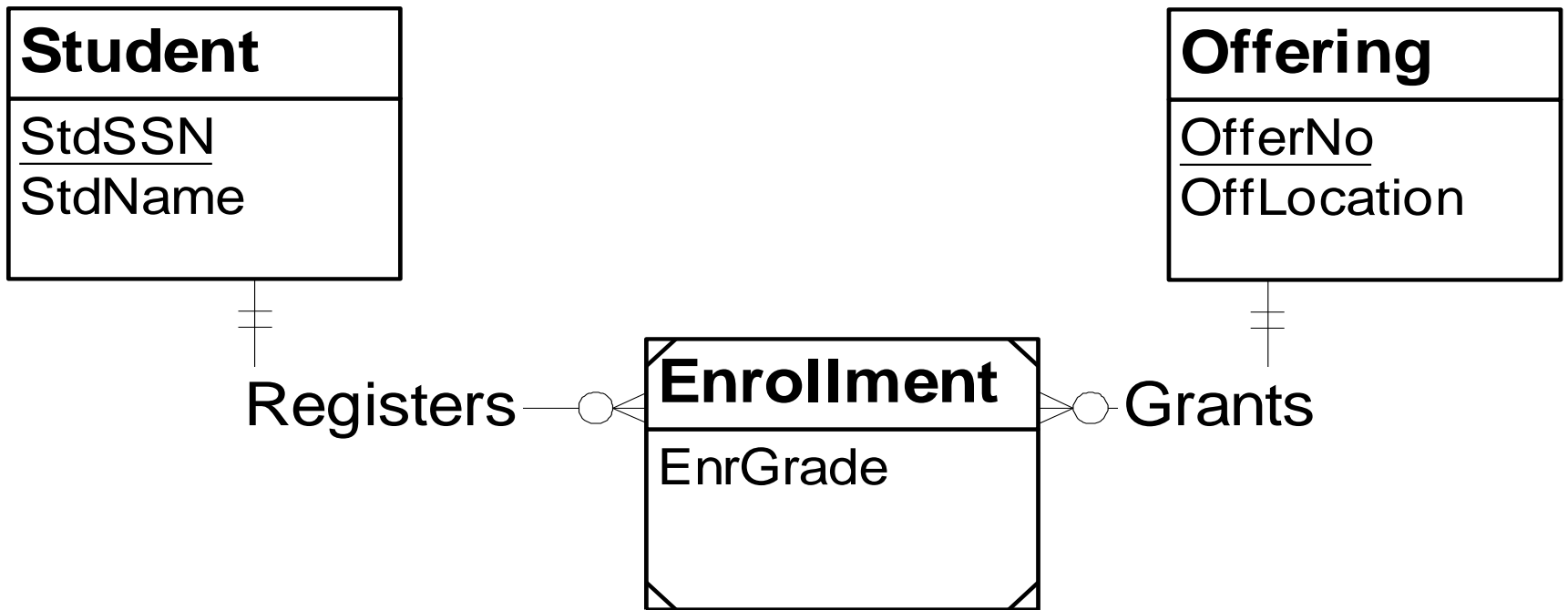
1. Create a new "association" table to hold the relationship
 - Name it with relationship verb name, or using the names of the tables in the relationship
2. Put the primary key from each entity involved in the relationship into the association table as foreign keys
3. The primary key of the new table is the combination of the two foreign keys
4. If the relationship has attributes, put them in the association table

M-N Relationship



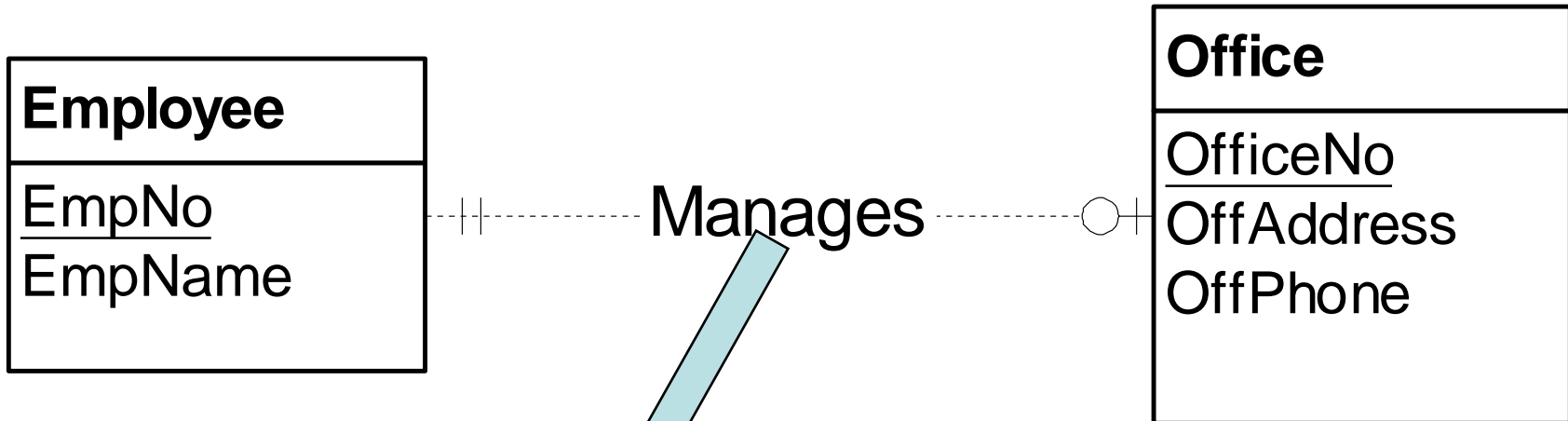
```
CREATE TABLE Enrolls_In (  
    StdSSN VARCHAR(11),  
    OfferNo INTEGER,  
    EnrGrade CHAR(1),  
    PRIMARY KEY (StdSSN, OfferNo),  
    FOREIGN KEY (StdSSN) REFERENCES Student(StdSSN),  
    FOREIGN KEY (OfferNo) REFERENCES Offering(OfferNo) )
```


M-N Relationship using Association Entity



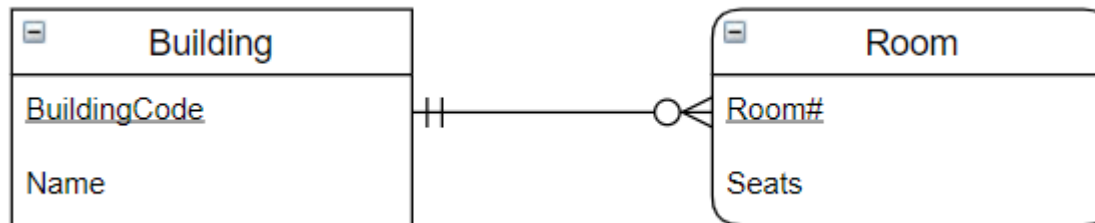
- Same conversion result as the previous slide

1-1 Relationships



```
CREATE TABLE Office (  
    OfficeNo INTEGER PRIMARY KEY,  
    OffAddress VARCHAR(20),  
    OffPhone VARCHAR(20),  
    EmpNo INTEGER NOT NULL,  
    FOREIGN KEY(EmpNo) REFERENCES Employee,  
    UNIQUE (EmpNo) )
```

Identifying Relationships

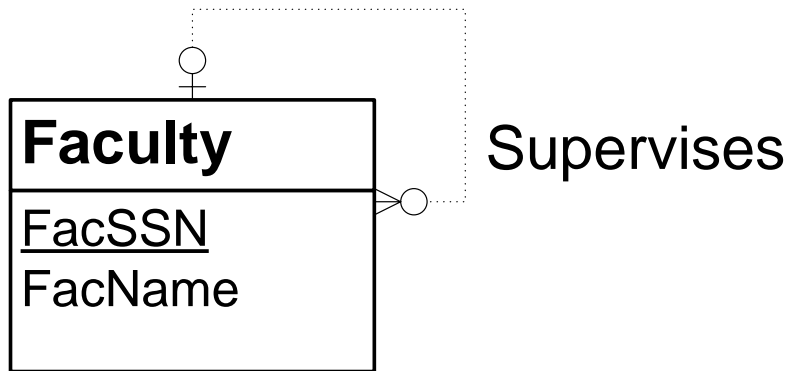


```
CREATE TABLE Building (  
    BuildingCode VARCHAR(3) PRIMARY KEY,  
    Name VARCHAR(20))
```

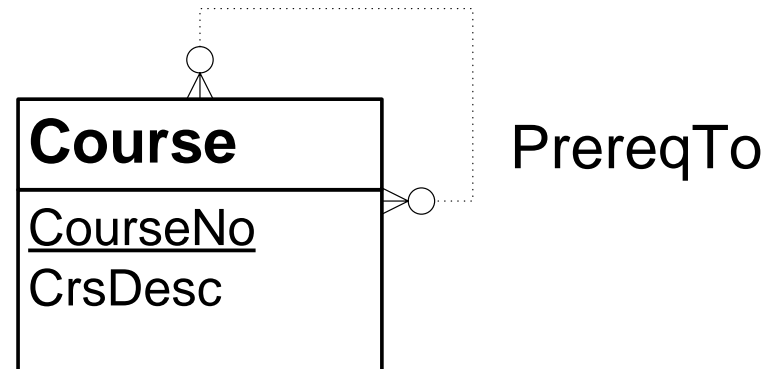
```
CREATE TABLE Room (  
    BuildingCode VARCHAR(3) NOT NULL,  
    RoomNo INTEGER NOT NULL,  
    PRIMARY KEY(BuildingCode, RoomNo),  
    FOREIGN KEY(BuildingCode) REFERENCES Building)
```

Self-Referencing Relationships

a) manager-subordinate



b) course prerequisites



```
CREATE TABLE Faculty (  
    FacSSN CHAR(11) PRIMARY KEY,  
    FacName VARCHAR(20),  
    FacSupervisor CHAR(11),  
    FOREIGN KEY(FacSupervisor) REFERENCES Faculty(FacSSN))
```

Summary of Basic Conversion Rules

- Each entity type becomes a table.
- Each 1-M relationship becomes a foreign key in the table corresponding to the child entity type (the entity type near the crow's foot symbol).
- Each M-N relationship becomes an associative table with a combined primary key.
- Each identifying relationship adds a column to a primary key.