CS2102
Database Systems

Slides adapted from Prof. Chan Chee Yong

LECTURE 04

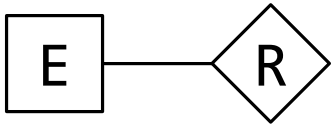
ENTITY RELATIONSHIP DATA MODEL

PREVIOUS LECTURE

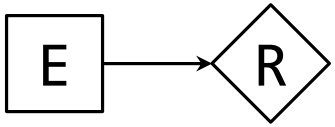
Summary

Relationship constraints

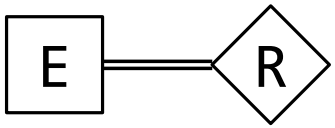
- Many-to-many — Each instance of E participates in *0 or more* instance of R



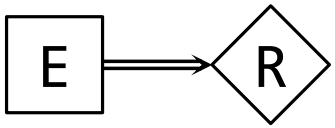
- Key — Each instance of E participates in *at most 1* instance of R



- Total — Each instance of E participates in *at least 1* instance of R



- Key & total — Each instance of E participates in *exactly one* instance of R



- Weak entity — E is a weak entity set with identifying owner E' and identifying relationship set R

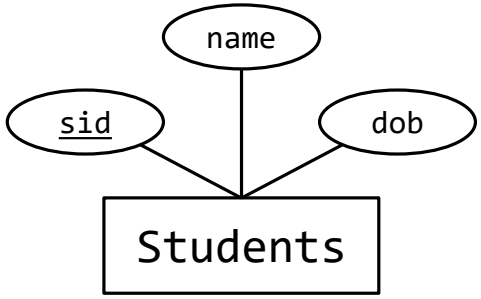
- Entity Relationship Diagram
  - ER model
  - Relationship constraints
  - Participation constraints
  - Weak entity sets
- ER to SQL
  - ER diagram to SQL
  - Additional ER concepts
  - ER design and relational mapping

ER to SQL

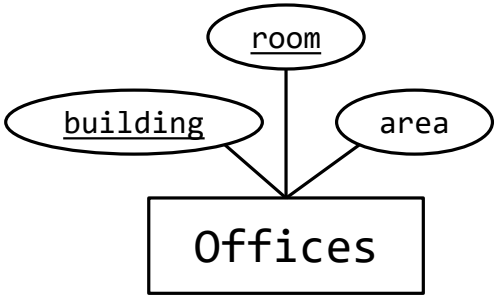
ER diagram to SQL

Entity sets

```
CREATE TABLE Students (
  sid      integer,
  name     varchar(30),
  dob      date,
  PRIMARY KEY (sid)
);
```



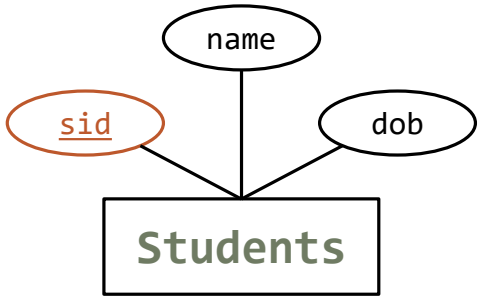

```
CREATE TABLE Offices (
  building      char(10),
  room          integer,
  area          varchar(20),
  PRIMARY KEY (building,room)
);
```



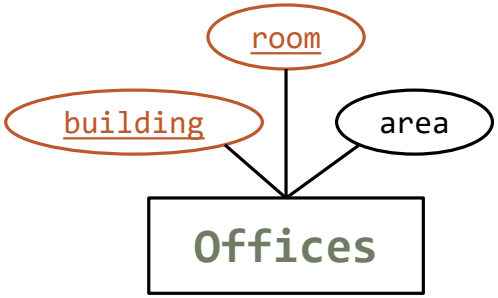
ER diagram to SQL

Entity sets

```
CREATE TABLE Students (
   sid      integer,
   name     varchar(30),
   dob      date,
   PRIMARY KEY (sid)
);
```

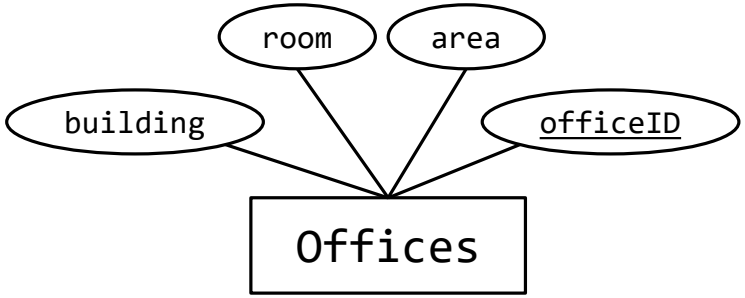



```
CREATE TABLE Offices (
   building      char(10),
   room          integer,
   area          varchar(20),
   PRIMARY KEY (building,room)
);
```



ER diagram to SQL

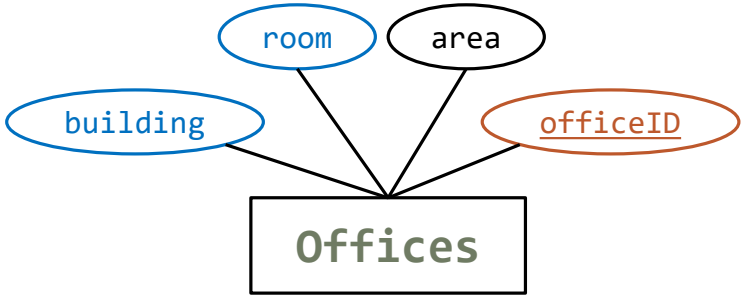
Entity sets with candidate keys




```
CREATE TABLE Offices (
  officeID      char(10) PRIMARY KEY,
  building      char(10) NOT NULL,
  room          integer  NOT NULL,
  area          varchar(20),
  UNIQUE (building,room)
);
```

ER diagram to SQL

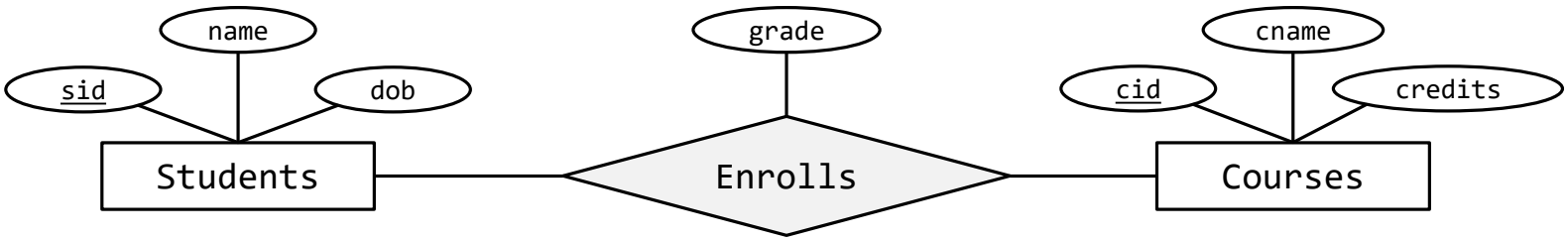
Entity sets with candidate keys




```
CREATE TABLE Offices (
  officeID      char(10) PRIMARY KEY,
  building      char(10) NOT NULL,
  room          integer  NOT NULL,
  area          varchar(20),
  UNIQUE (building,room)
);
```

ER diagram to SQL

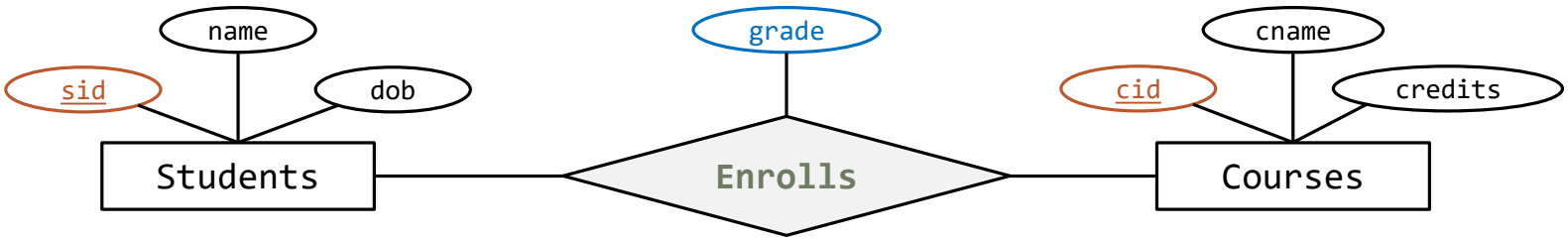
Relationship sets without constraints




```
CREATE TABLE Enrolls (
  sid     integer REFERENCES Students,
  cid     char(5) REFERENCES Courses,
  grade   numeric,
  PRIMARY KEY (sid,cid)
);
```

ER diagram to SQL

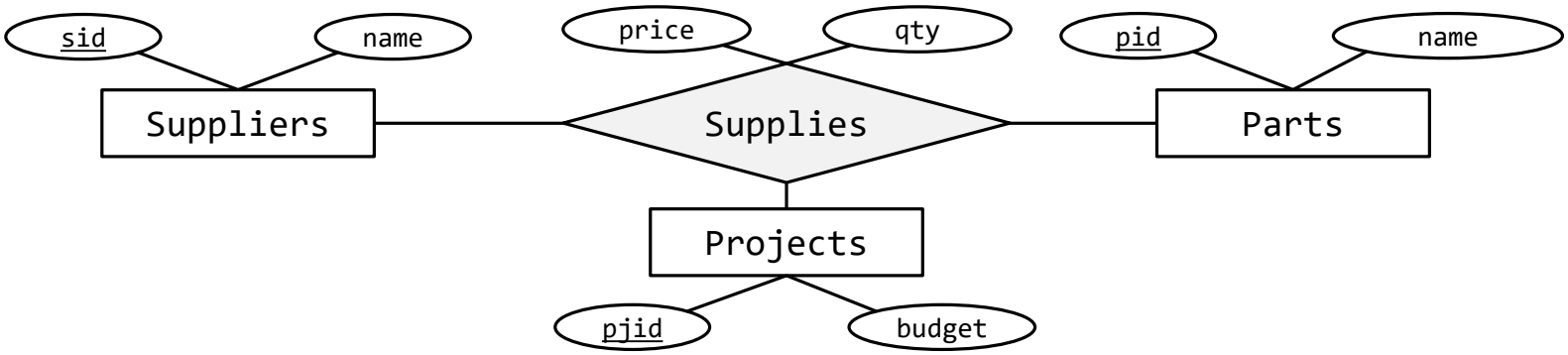
Relationship sets without constraints




```
CREATE TABLE Enrolls (
  sid     integer REFERENCES Students,
  cid     char(5) REFERENCES Courses,
  grade   numeric,
  PRIMARY KEY (sid,cid)
);
```

ER diagram to SQL

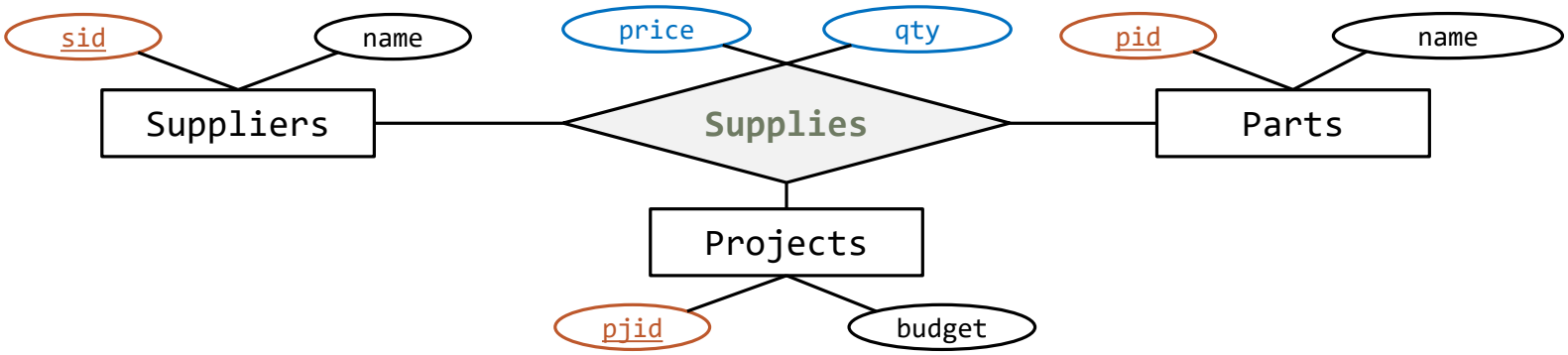
Relationship sets without constraints




```
CREATE TABLE Supplies (
  sid     char(10) REFERENCES Suppliers,
  pid     char(10) REFERENCES Parts,
  pjid    char(10) REFERENCES Projects,
  price   numeric,
  qty     integer,
  PRIMARY KEY (sid,pid,pjid)
);
```

ER diagram to SQL

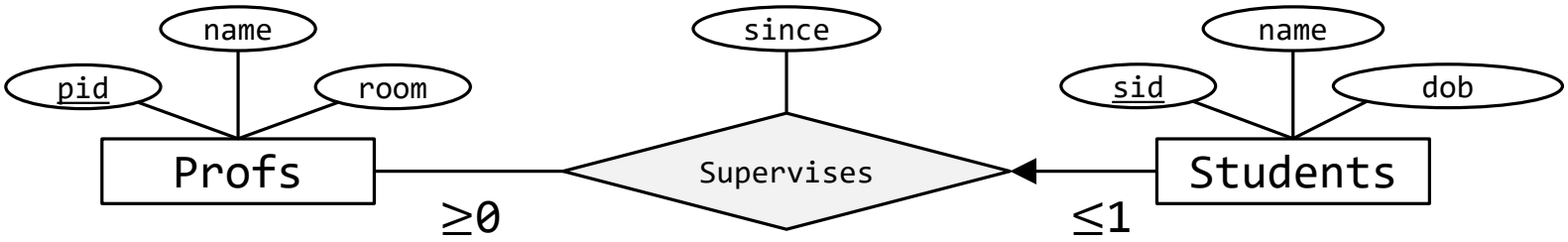
Relationship sets without constraints




```
CREATE TABLE Supplies (
  sid     char(10) REFERENCES Suppliers,
  pid     char(10) REFERENCES Parts,
  pjid    char(10) REFERENCES Projects,
  price   numeric,
  qty     integer,
  PRIMARY KEY (sid,pid,pjid)
);
```

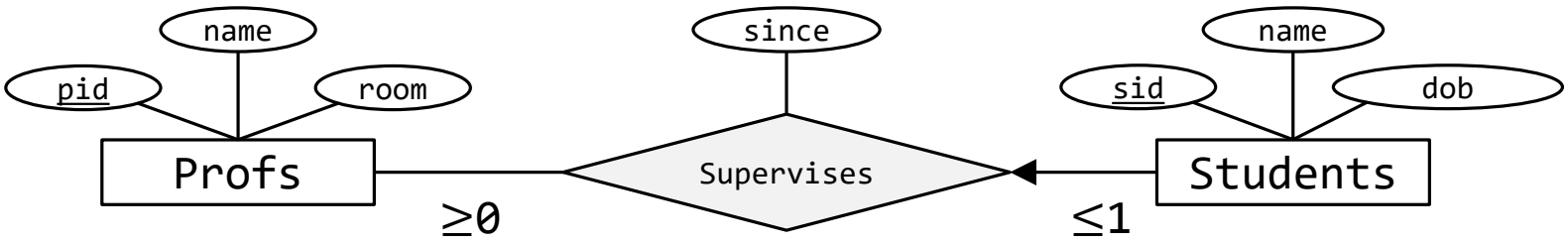
ER diagram to SQL

Relationship sets with key constraints



ER diagram to SQL

Relationship sets with key constraints



First approach

Represent `Supervises` with a *separate table*

- `Profs (pid, name, room)`
- `Students (sid, name, dob)`
- `Supervises (sid, pid, since)`

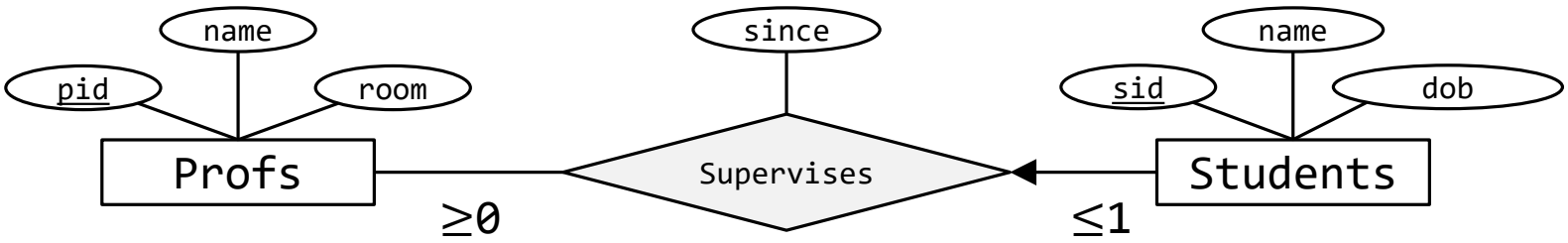
Second approach

Combine `Supervises` & `Students` into a *single table*

- `Profs (pid, name, room)`
- `SupervisedStudents (sid, name, dob, pid, since)`

ER diagram to SQL

Relationship sets with key constraints



First approach

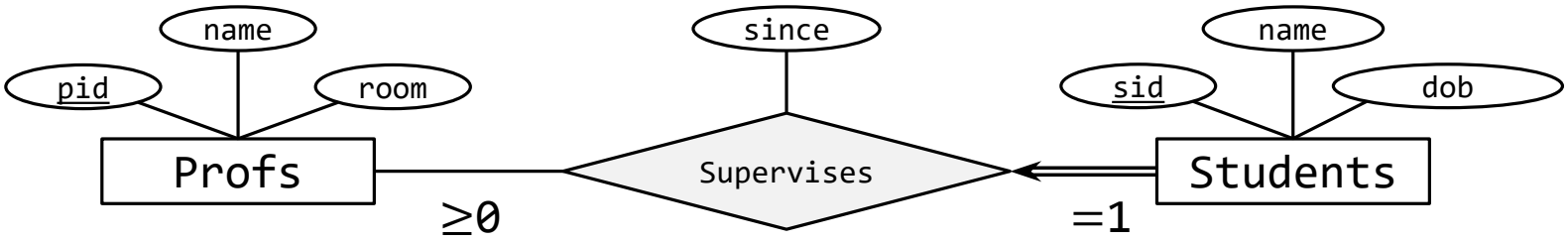
```
CREATE TABLE Supervises (
  sid   integer,
  pid   char(7),
  since date,
  PRIMARY KEY(sid),
  FOREIGN KEY(sid) REFERENCES
    Students,
  FOREIGN KEY(pid) REFERENCES
    Profs
);
```

Second approach

```
CREATE TABLE SupervisedStudents (
  sid   integer,
  name  varchar(30),
  dob   date,
  pid   char(7),
  since date,
  PRIMARY KEY(sid),
  FOREIGN KEY(pid) REFERENCES
    Profs
);
```

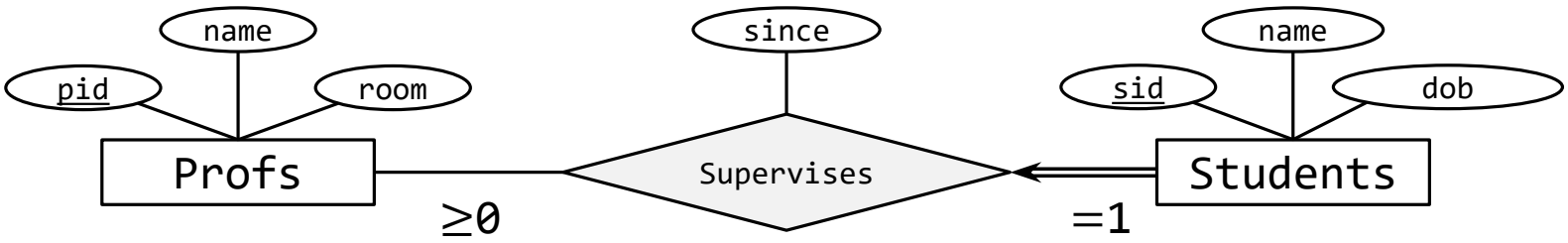
ER diagram to SQL

Relationship sets with key & total constraints



ER diagram to SQL

Relationship sets with key & total constraints



First approach

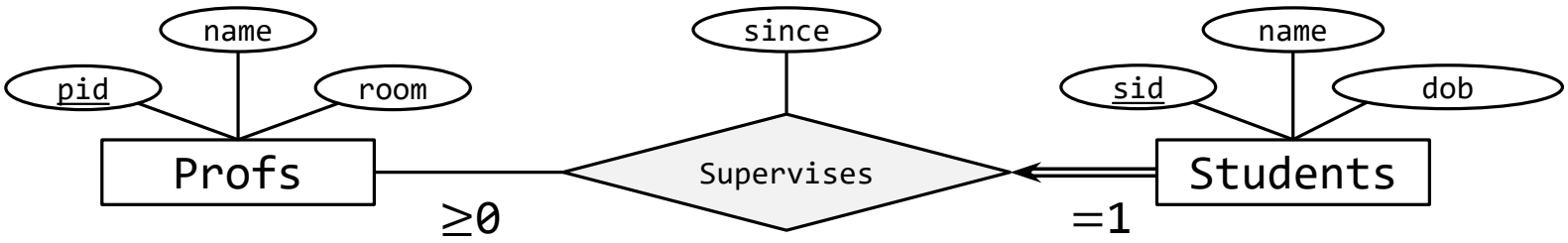
```
CREATE TABLE Supervises (
  sid   integer,
  pid   char(7),
  since date,
  PRIMARY KEY(sid),
  FOREIGN KEY(sid) REFERENCES
    Students,
  FOREIGN KEY(pid) REFERENCES
    Profs
);
```

Second approach

```
CREATE TABLE SupervisedStudents (
  sid   integer,
  name  varchar(30),
  dob   date,
  pid   char(7) NOT NULL,
  since date,
  PRIMARY KEY(sid),
  FOREIGN KEY(pid) REFERENCES
    Profs
);
```

ER diagram to SQL

Relationship sets with key & total constraints



First approach

```
CREATE TABLE Supervises (
  sid   integer,
  pid   char(7),
  since date,
  PRIMARY KEY(sid),
  FOREIGN KEY(sid) REFERENCES
    Students,
  FOREIGN KEY(pid) REFERENCES
    Profs
);
```

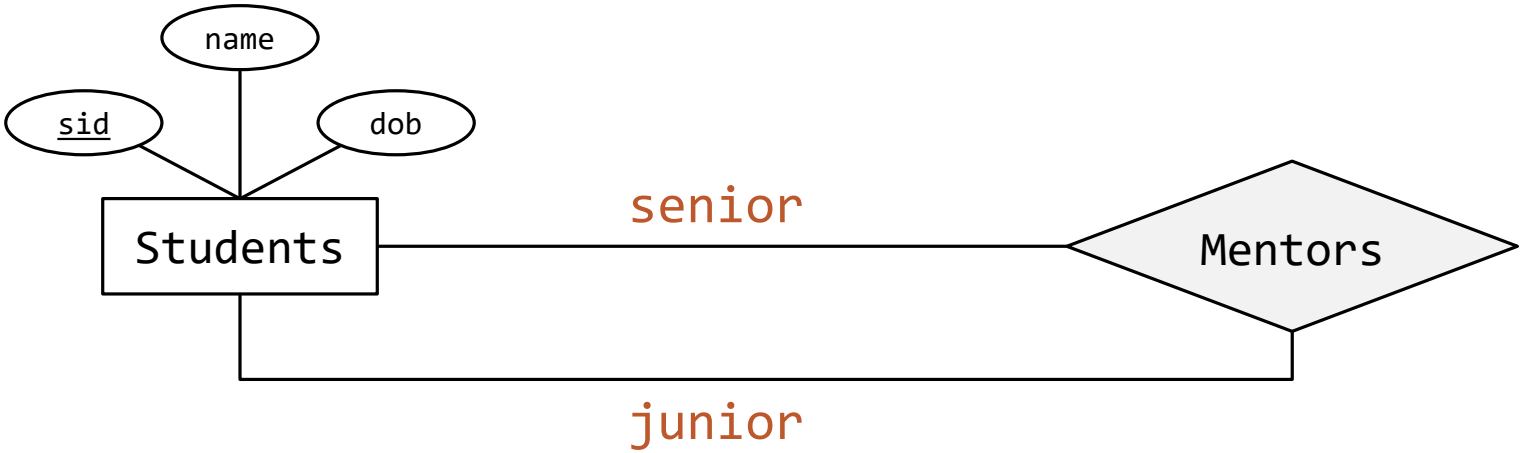
Total participation constraint of Students w.r.t Supervises is not captured!

Second approach

```
CREATE TABLE SupervisedStudents (
  sid   integer,
  name  varchar(30),
  dob   date,
  pid   char(7) NOT NULL,
  since date,
  PRIMARY KEY(sid),
  FOREIGN KEY(pid) REFERENCES
    Profs
);
```

ER diagram to SQL

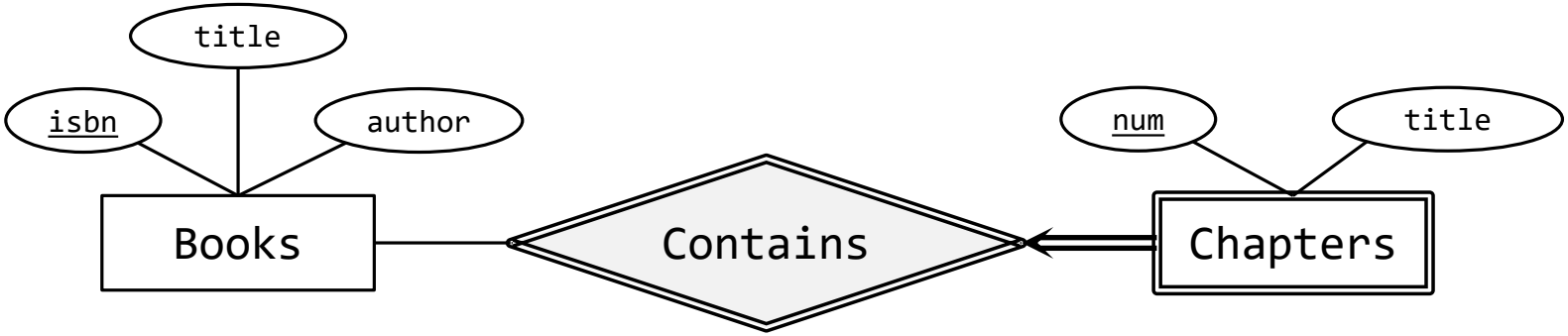
Roles in relationships




```
CREATE TABLE Mentors (
  seniorSID  integer,
  juniorSID  integer,
  PRIMARY KEY(seniorSID, juniorSID),
  FOREIGN KEY(seniorSID) REFERENCES Students(sid),
  FOREIGN KEY(juniorSID) REFERENCES Students(sid)
);
```

ER diagram to SQL

Weak entity sets



Books table

```
CREATE TABLE Books (
  isbn    char(30),
  title   char(50),
  author  char(60),
  PRIMARY KEY(isbn)
);
```

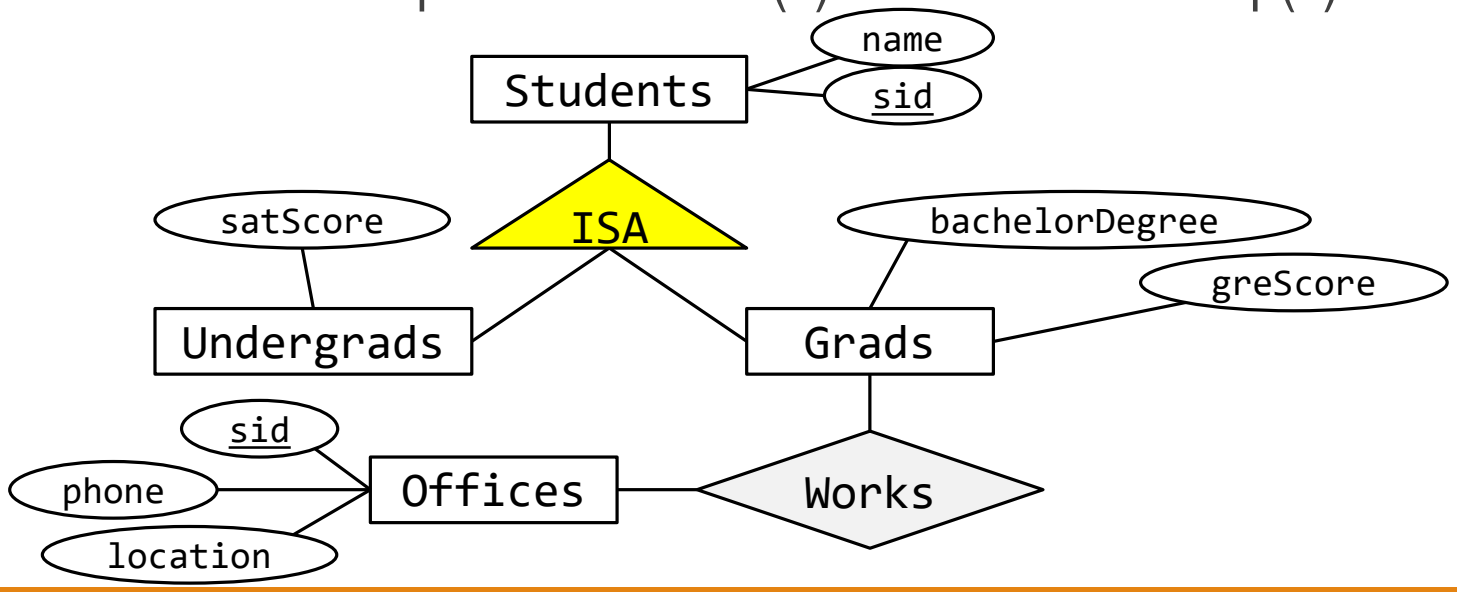
BookChapters table

```
CREATE TABLE BookChapters (
  num     char(30),
  title   char(50),
  isbn    char(30),
  PRIMARY KEY(num,isbn),
  FOREIGN KEY(isbn)
    REFERENCES Books
    ON DELETE cascade
);
```

Additional ER concepts

ISA hierarchies

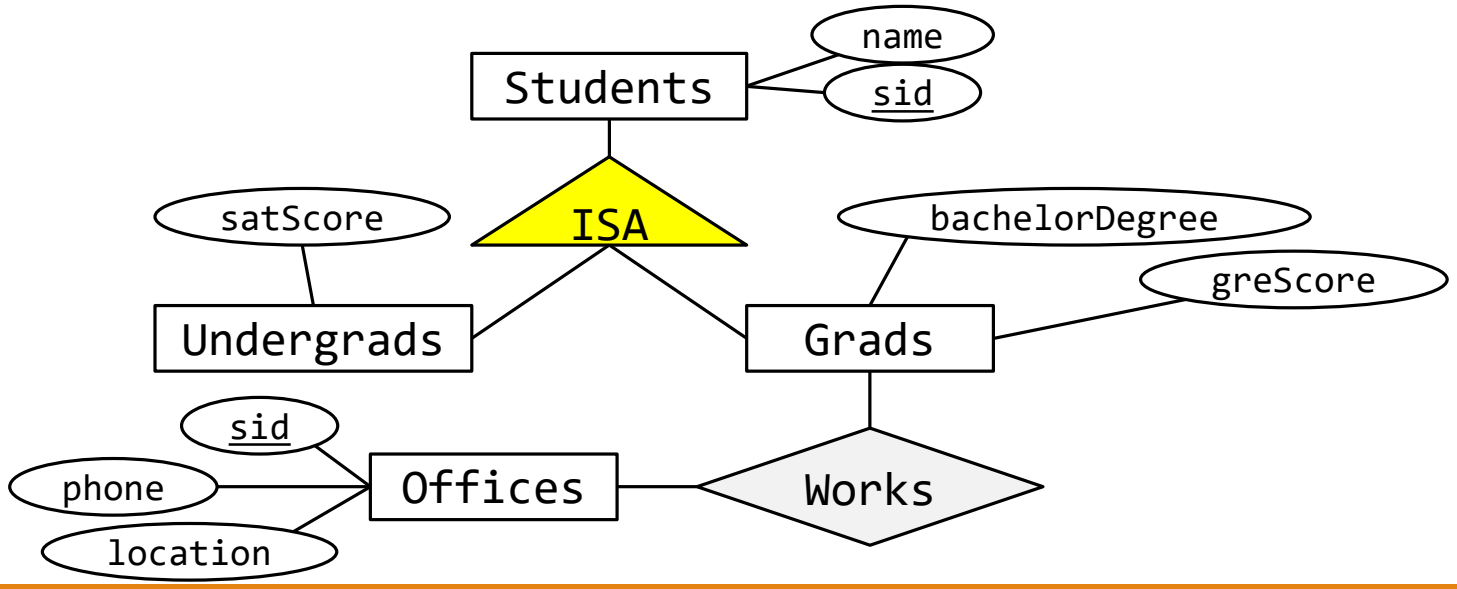
- Based on “is-a” relationship in OOP
  - Subclass-superclass relationship
  - Describing an entity sets into subclasses
- Every entity in a subclass entity set is an entity in its superclass entity set
- Each subclass has specific attribute(s) and/or relationship(s)



Additional ER concepts

ISA hierarchies

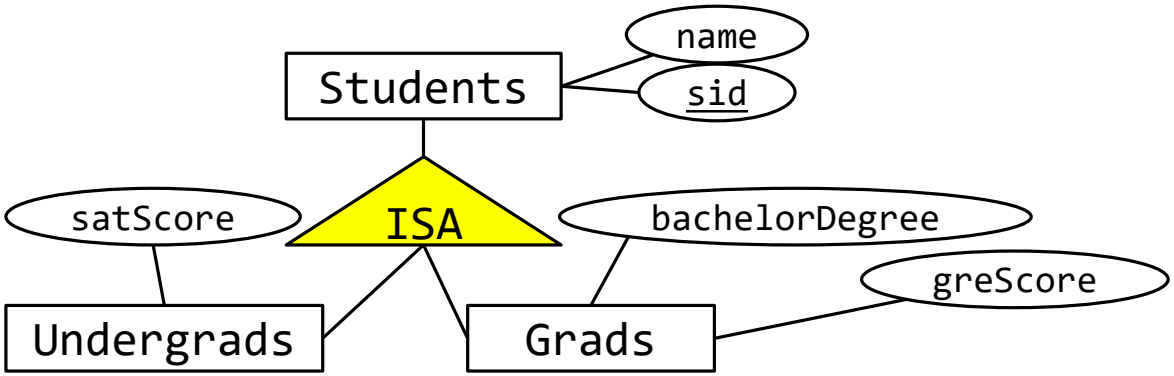
- Constraints:
  - Overlap constraints: can entity belong to multiple subclasses?
    - Satisfied if entity in superclass *could belong to multiple subclasses*
  - Covering constraints: does an entity in a superclass have to belong to some subclass?
    - Satisfied if every entity in a superclass *has to belong to some subclass*



Additional ER concepts

ISA hierarchies

- **Approach #1**: one relation per subclass or superclass

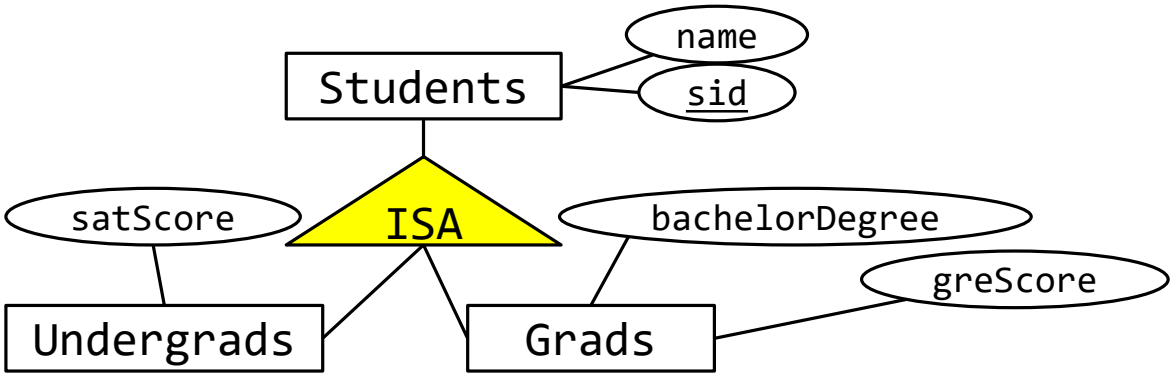



```
CREATE TABLE Students (
  sid       integer PRIMARY KEY,
  name      char(30) );

CREATE TABLE Undergrads (
  sid       integer PRIMARY KEY REFERENCES Students
            ON DELETE cascade,
  satScore  numeric );

CREATE TABLE Grads (
  sid       integer PRIMARY KEY REFERENCES Students
            ON DELETE cascade,
  greScore  numeric ); -- bachelorDegree omitted due to space
```

Additional ER concepts

ISA hierarchies

- **Approach #2**: one relation per subclass



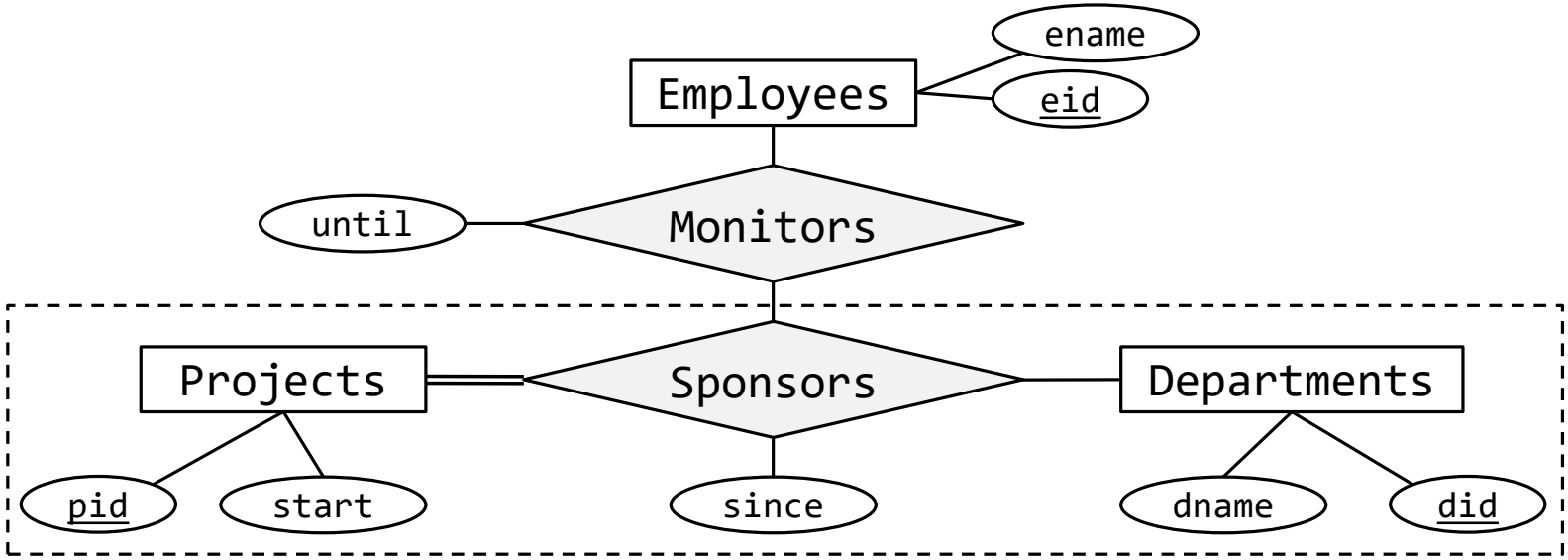
Applicable if covering constraint is satisfied

```
CREATE TABLE Undergrads (
  sid        integer PRIMARY KEY,
  name       char(30),
  satScore   numeric );

CREATE TABLE Grads (
  sid        integer PRIMARY KEY,
  name       char(30),
  greScore   numeric ); -- bachelorDegree omitted due to space
```

Additional ER concepts

Aggregation

- How to model a relationship between entities & relationships?
- Example:
  - Every project is sponsored by at least one department
  - Each sponsorship has a “since” attribute & might be monitored by 0 or more employees
  - Each monitoring has an “until” attribute



Additional ER concepts

Aggregation

- Relational mapping

CREATE TABLE Projects (
  pid char(20),
  start date,
  PRIMARY KEY (pid)
);
	CREATE TABLE Departments (
	  did char(20),
	  dname char(30),
	  PRIMARY KEY (did)
	);

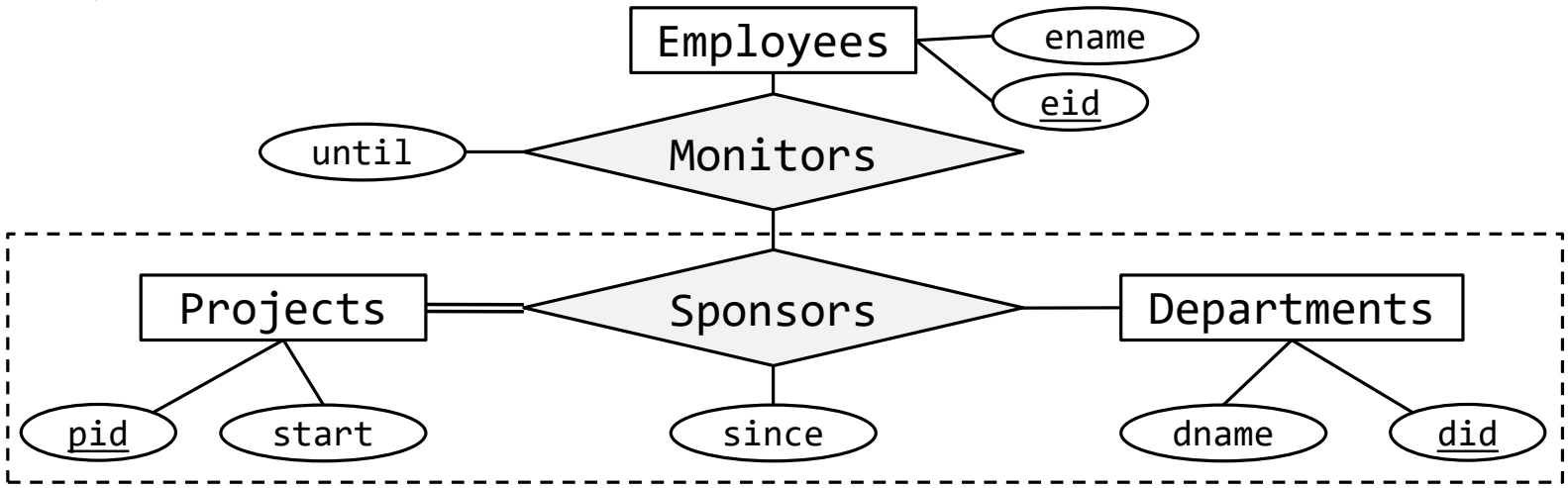
CREATE TABLE Sponsors (
  pid char(20)
  REFERENCES Projects,
  did char(30)
  REFERENCES Departments,
  since date,
  PRIMARY KEY (pid,did)
);
	CREATE TABLE Employees (
	  eid char(20),
	  ename char(30),
	  PRIMARY KEY (eid)
	);

Additional ER concepts

Aggregation

- Relational mapping

```
CREATE TABLE Monitors (
  eid       char(20) REFERENCES Employees,
  pid       char(30),
  did       char(30),
  until     date,
  PRIMARY KEY (eid,pid,did),
  FOREIGN KEY (pid,did) REFERENCES Sponsors (pid,did)
);
```

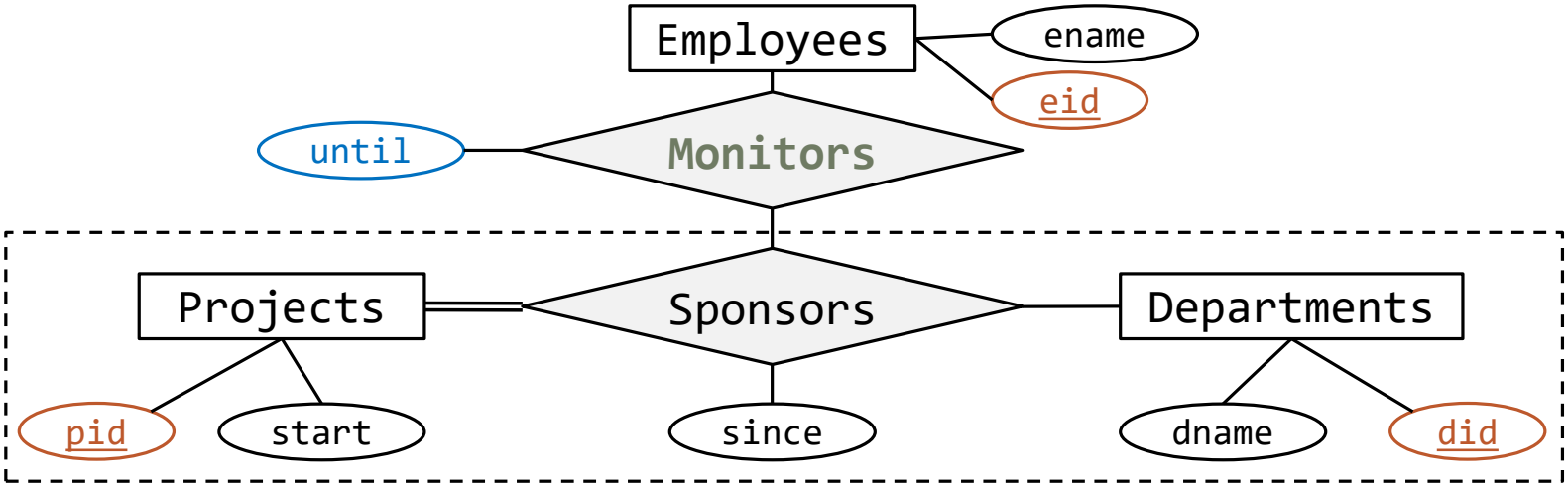


Additional ER concepts

Aggregation

- Relational mapping

```
CREATE TABLE Monitors (
  eid       char(20) REFERENCES Employees,
  pid       char(30),
  did       char(30),
  until     date,
  PRIMARY KEY (eid,pid,did),
  FOREIGN KEY (pid,did) REFERENCES Sponsors (pid,did)
);
```



ER design and relational mapping

Guidelines for ER design

- ER design should capture as many of the application's constraints as possible
- ER design must not impose any constraint that is not required in the application

Guidelines for relational mapping

- Relational schema should enforce as many of the application's constraints as possible using column/table constraints
- Relational schema must not impose any constraint that is not required in the application

Summary

- ER model has expressive constructs for conceptual data design
  - Concepts: *entities; relationships; attributes; weak entities; ISA hierarchies; aggregation*
  - Constraints: *key constraints; participation constraints*
- ER design is subjective
- Rules for mapping entity-relationship model to relational model
  - Entity & relationship sets
  - Key constraints
  - Participation constraints
  - Relationship roles
  - Weak entity sets
  - ISA hierarchies
  - Aggregation