The arity of a relation is the number of fields.
1. True
2. False

A super key is...
1. A set of all candidate keys
2. A set of fields that contains a candidate key
3. A candidate key that is also the primary key
4. A set of fields that contains more than one candidate key
An integrity constraint limits the data that can be viewed by a database user

1. True
2. False

If the (a) _______ is deleted, then a (the) _______ is (are) deleted.

1. Student record; Enrolled record
2. Enrolled record; student record;
3. Enrolled table; Student table
4. Enrolled record; all student records

ER to Relational

- Entity sets to tables

```
SSN  Name  HDate
```

Employees
ER to Relational

- Entity sets to tables

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>HDate</th>
</tr>
</thead>
</table>

CREATE TABLE Employees (SSN CHAR(11),
 Name CHAR(30),
 HDate Date,
 PRIMARY KEY (SSN))

ER to Relational

- Relation sets to tables (no constraints)

CREATE TABLE WorksIn (SSN CHAR(11),
 DID INTEGER,
 Since DATE,
 PRIMARY KEY (SSN, DID),
 FOREIGN KEY (SSN) REFERENCES Employees,
 FOREIGN KEY (DID) REFERENCES Departments)
ER to Relational

• Relation sets to tables (no constraints)

CREATE TABLE ReportsTo (supervisor_SSN CHAR(11), subordinate_SSN CHAR(11), PRIMARY KEY (supervisor_SSN, subordinate_SSN), FOREIGN KEY (supervisor_SSN) REFERENCES Employees(SSN), FOREIGN KEY (subordinate_SSN) REFERENCES Employees(SSN))

ER to Relational

• Relation sets to tables (no constraints)

ER to Relational

• Relation sets to tables (with key constraints)
ER to Relational

• Relation sets to tables (with key constraints)

CREATE TABLE Manages (SSN CHAR(11),
DID INTEGER,
Since DATE,
PRIMARY KEY (DID),
FOREIGN KEY (SSN) REFERENCES Employees,
FOREIGN KEY (DID) REFERENCES Departments)

CREATE TABLE Dept_Mgr (DID INTEGER,
DName CHAR(20),
Budget REAL,
SSN CHAR(11),
Since DATE,
PRIMARY KEY (DID),
FOREIGN KEY (SSN) REFERENCES Employees)

ER to Relational

• Which way is better?

• Departments + Manages
• Dept_Mgr
ER to Relational

- Which way is better?

- Departments + Manages
  - Often more space efficient
  - Can be slow when generating the combined view
- Dept_Mgr
  - Can waste space
  - Remove the need for another table and therefore faster to query

CREATE TABLE Dept_Mgr (DID INTEGER,
  DName CHAR(20),
  Budget REAL,
  SSN CHAR(11) NOT NULL,
  Since DATE,
  PRIMARY KEY (DID),
  FOREIGN KEY (SSN) REFERENCES Employees
  ON DELETE NO ACTION
)

Can we still capture the total participation constraint?
ER to Relational

- Relation sets to tables (with participation constraints)

```
CREATE TABLE Manages (SSN CHAR(11) NOT NULL, 
    DID INTEGER, 
    Since DATE, 
    PRIMARY KEY (DID), 
    FOREIGN KEY (SSN) REFERENCES Employees, 
    FOREIGN KEY (DID) REFERENCES Departments 
)
```

Does this fix the problem?

ER to Relational

- Relation sets to tables (with participation constraints)

```
CREATE TABLE Dep_Policy (PName CHAR(20), 
    Age INTEGER, 
    Cost REAL, 
    SSN CHAR(11), 
    PRIMARY KEY (PName, SSN), 
    FOREIGN KEY (SSN) REFERENCES Employees 
    ON DELETE CASCADE 
)
```

ER to Relational

- Weak entities to tables

```
CREATE TABLE Dep_Policy (PName CHAR(20), 
    Age INTEGER, 
    Cost REAL, 
    SSN CHAR(11), 
    PRIMARY KEY (PName, SSN), 
    FOREIGN KEY (SSN) REFERENCES Employees 
    ON DELETE CASCADE 
)
```

Participation constraints in general cannot be enforced with only table definitions.

We need other more powerful tools, such as table constraints and assertions.

What can be captured?

1. In a binary relationship, there is exactly one of the entity sets has a key constraint, and the total participation constraint is expressed on that entity set.
2. A relationship in which all participating entity sets have key constraints and total participation.
ER to Relational

- Class hierarchies
  - First approach: Dog, Trained_Dog, In_Training_Dog

ER to Relational

- Class hierarchies
  - Second approach: Trained_Dog and In_Training_Dog which common attributes of Dog.

ER to Relational

- Class hierarchies
  - Which way is better?