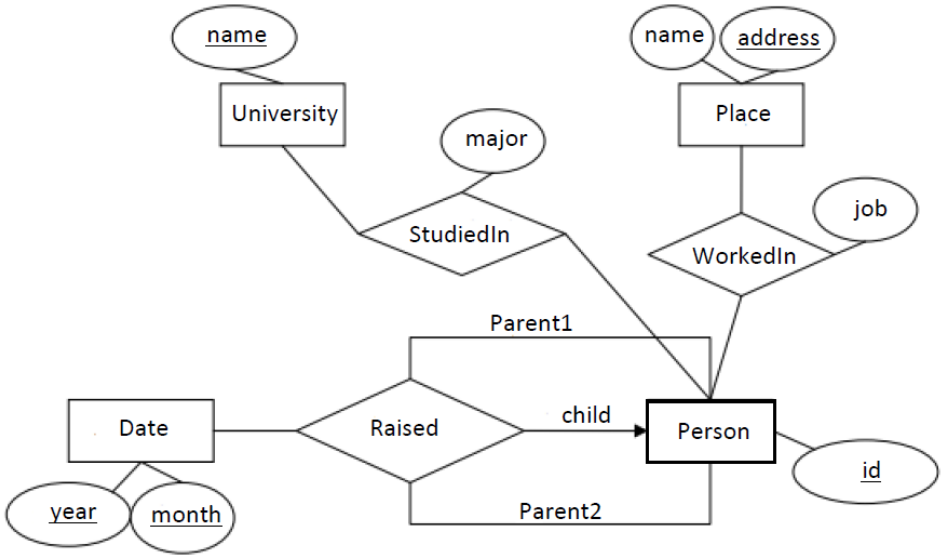
Introduction to Operating Systems and SQL for Data Science

Practice 11 – Entity Relation

ERD



Reichman University

Introduction

- ERD is a formal modeling of data
- Conceptual database design
- The data is often consists of:
  - *Entities*
  - *Relationships*
- Examples:
  - Movies, actors, directors, roles, awards
  - Students, courses, lecturers, rooms
  - Persons, statuses, friendships, messages, likes

Reichman University

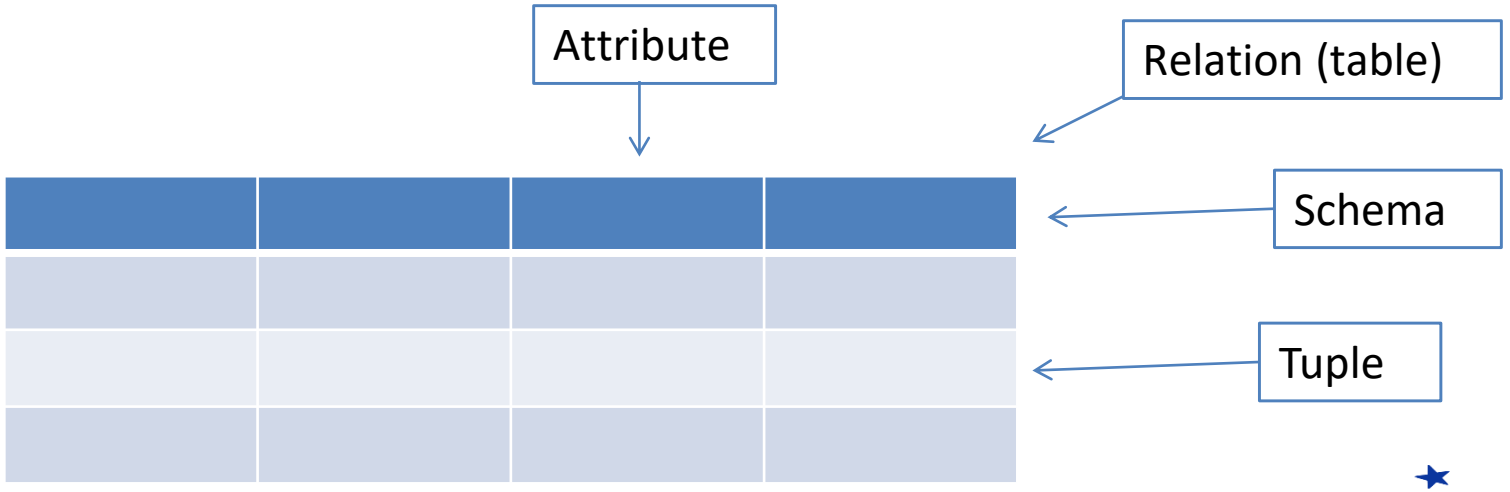
ERD

- Part 1 - Extracting Tables from ERD.
- Part 2 – ERD questions.

Reichman University

The relational model

- Logical database design
- We may deduce the relational model from the conceptual model
  - For instance, by extracting tables from an ERD
- Basic definitions:



Reichman University

A table example

- t1 = (foo,bar,baz,{x,y})
- t2 = (quz,bar,foo,{y,z})

- t1[a1] = (baz)
- t2[a2] = ({y,z})

- t1[$\bar{a}$] = (baz,{x,y})
- t2[$\underline{\bar{k}}$] = (quz,bar)

	<u>k1</u>	<u>k2</u>	a1	a2
t1	foo	bar	baz	{x,y}
t2	quz	bar	foo	{y,z}

Vector of all **key** attributes

$\underline{\bar{k}}$ = (k1,k2)

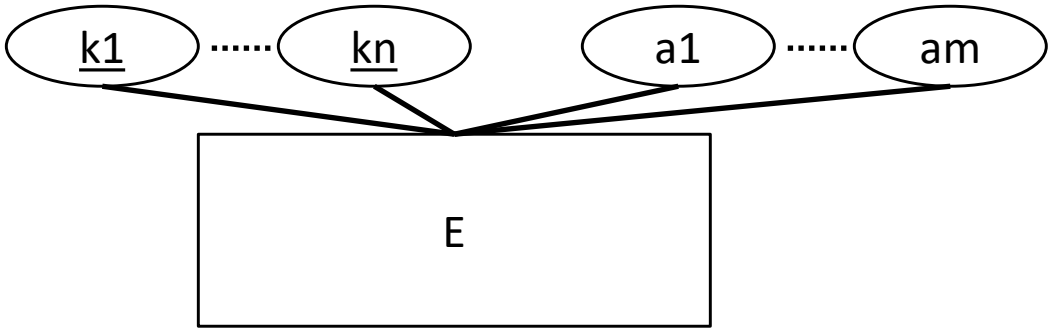
$\bar{a}$ = (a1,a2)

Vector of all **non-key** attributes

Reichman University

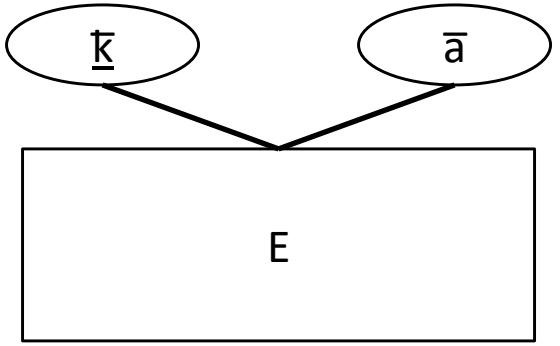
Extracting Tables from ERD

Entities



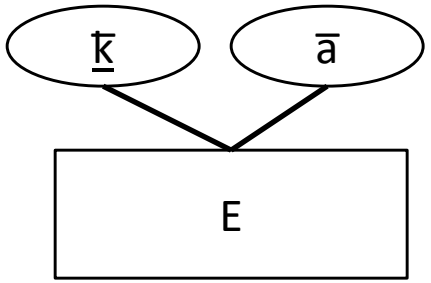
$\underline{\bar{k}}$ = (k1,…,kn) $\bar{a}$ = (a1,…,am)

- A short representation:



Reichman University

Entities



- Translation to a table:

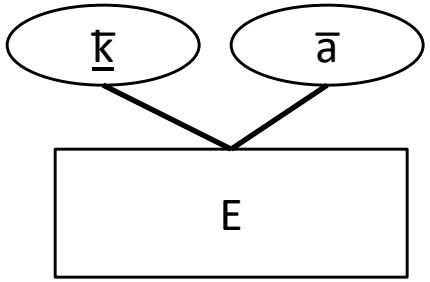
k̲			ā		
k1	…	kn	a1	…	am

- Constraints:

$$t_1[\underline{k}] = t_2[\underline{k}] \Rightarrow t_1[\bar{a}] = t_2[\bar{a}]$$

Reichman University

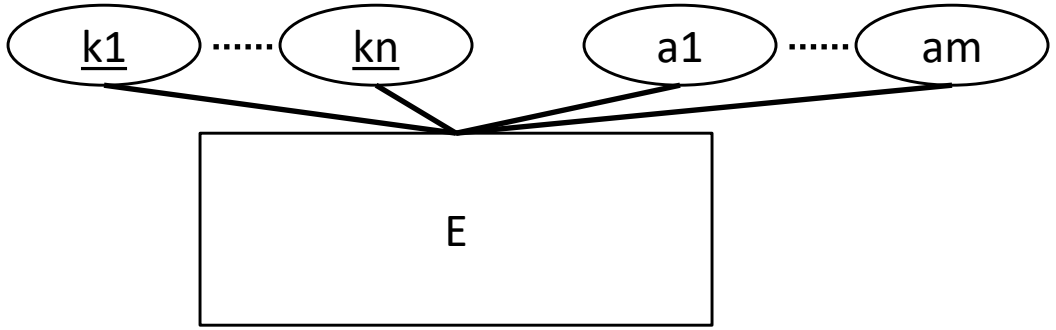
Entities

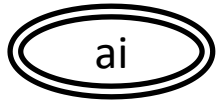


- We assume $\underline{k}$ can not be empty
- $\bar{a}$ may be empty

Reichman University

Entities

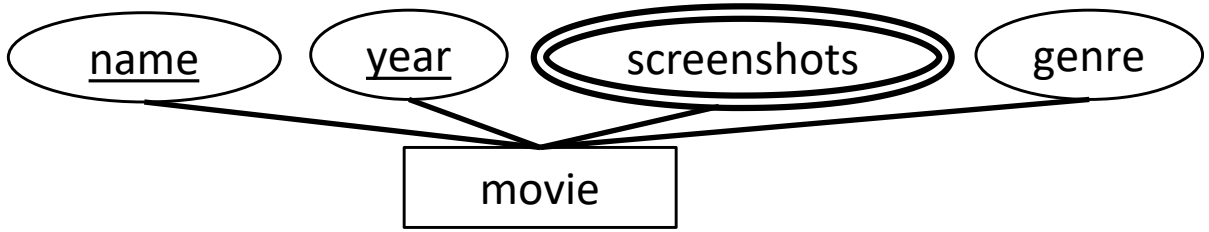




- Any ai may be multi-valued
  - Multi-valued attributes cannot be a part of the key
- In domain D, for a table row t:
  - for an attribute a_i: $t[a_i] \in D$
  - for a multi-valued attribute a_j: $t[a_j] \in P(D)$
    - a powerset.

Reichman University

Another Representation of a Multi-Valued Attribute



- A table without multi-valued attributes:

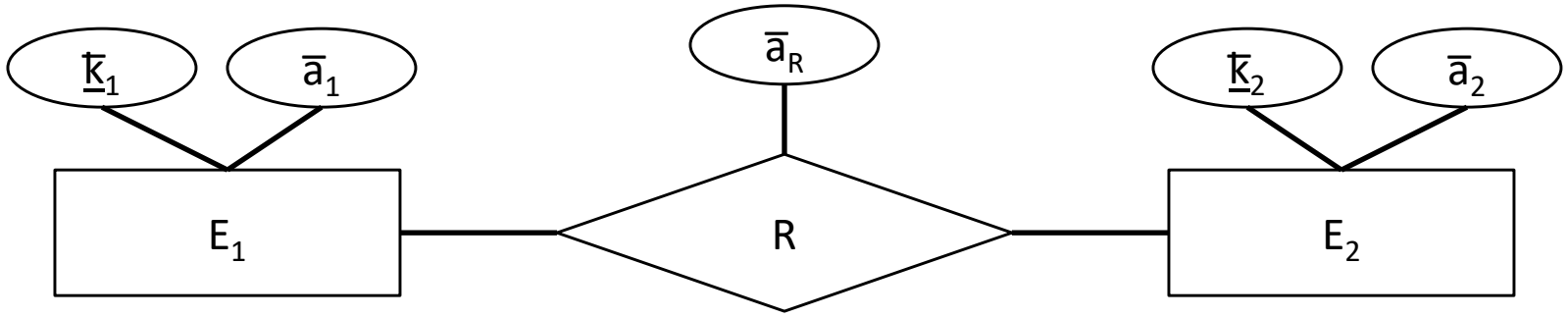
k		a2
name	year	genre
Cold Mountain	2003	Drama

- Tables - one for each multi-valued attribute:

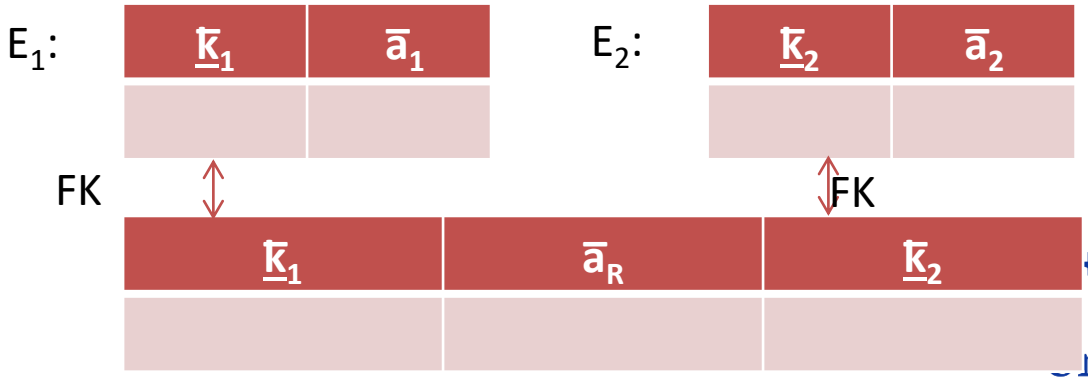
name	year	screenshots
Cold Mountain	2003	screenshot1

Reichman University

Relationships

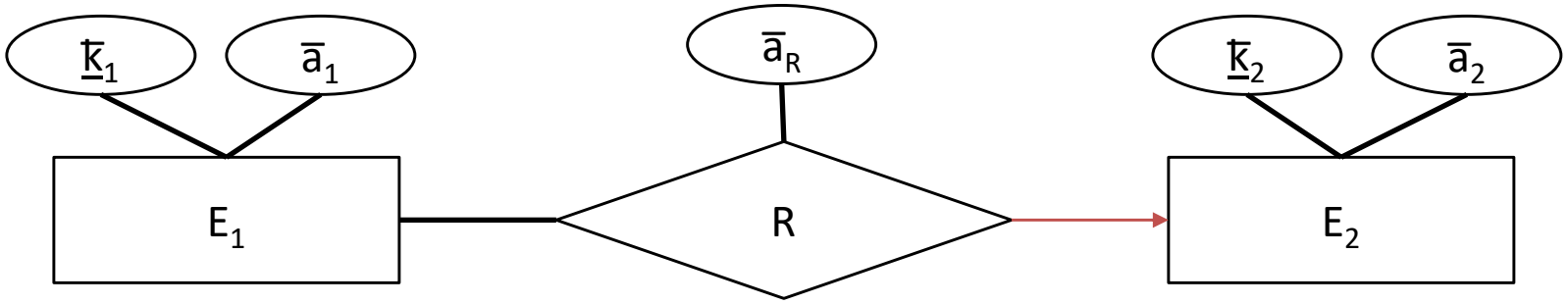


- For each $\underline{k}$, $\bar{a}$ may be empty
- Translation to a table:

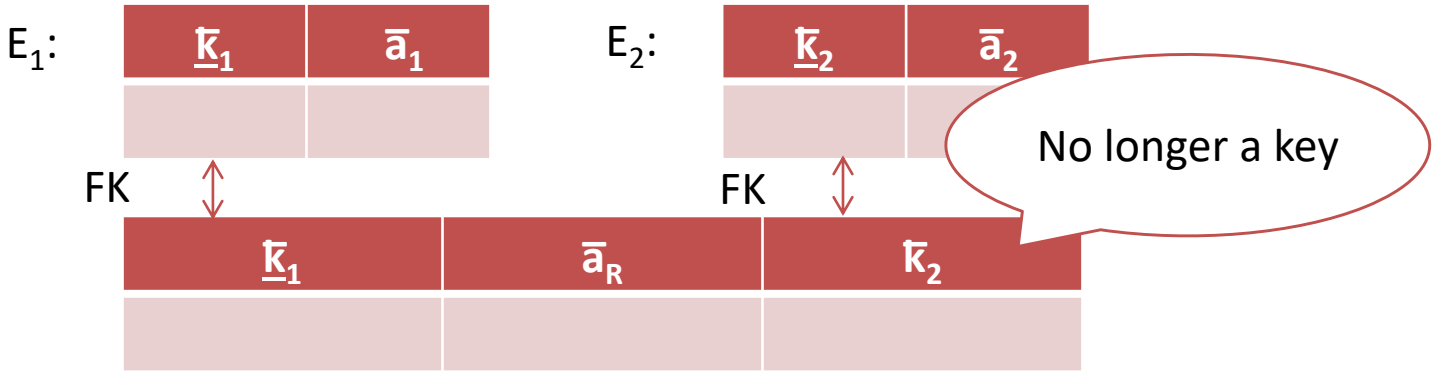


Reichman University

Relationships – many to one

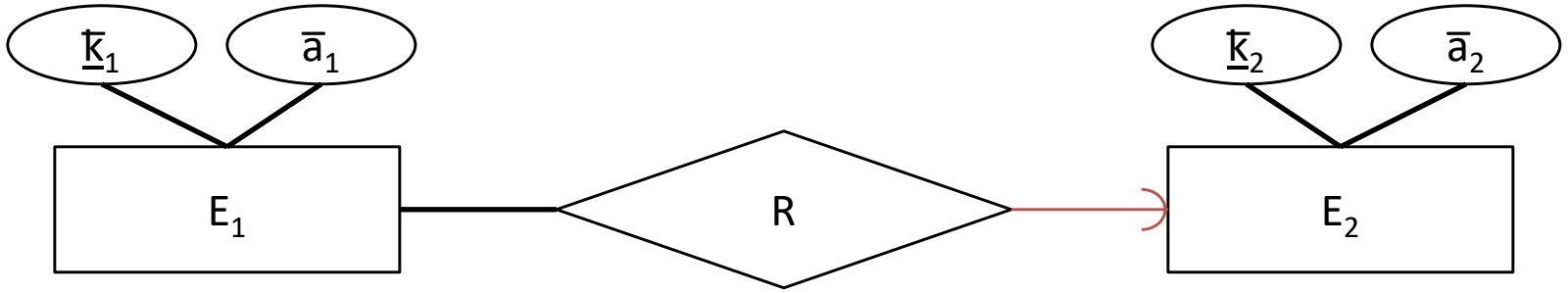


- An E_1 can relate to at most one E_2
- Translation to a table:



Reichman University

Relationships – unique reference



- An E_1 relates to precisely one E_2
- No table is extracted for R

E_1:

$\underline{k}_1$	$\bar{a}_1$	$\underline{k}_2$

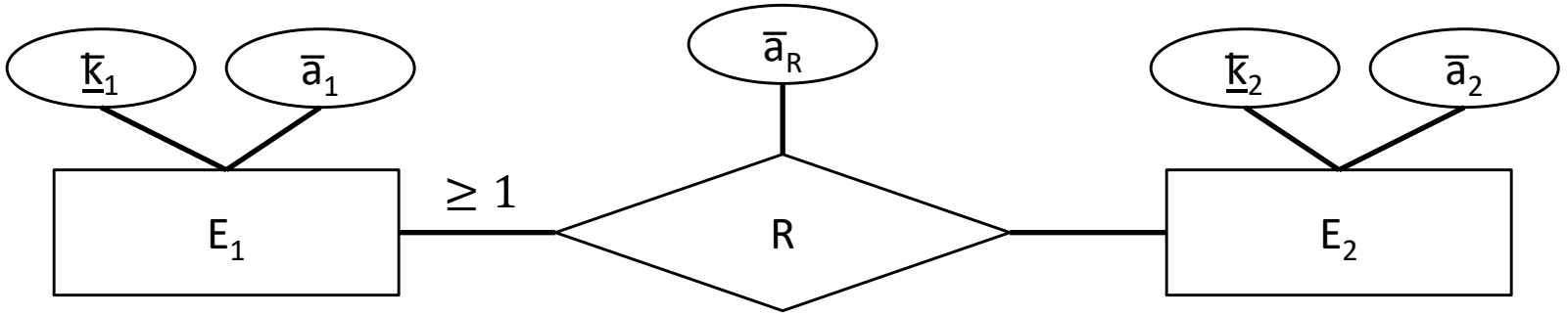
FK

E_2:

$\underline{k}_2$	$\bar{a}_2$

Reichman University

Relationships – degree constraint



- Each entity of type E_2 must relate (with R) to at least one entity of type E_1
- k_1 and k_2 in R are foreign keys

E_1:

$\underline{k}_1$	$\bar{a}_1$

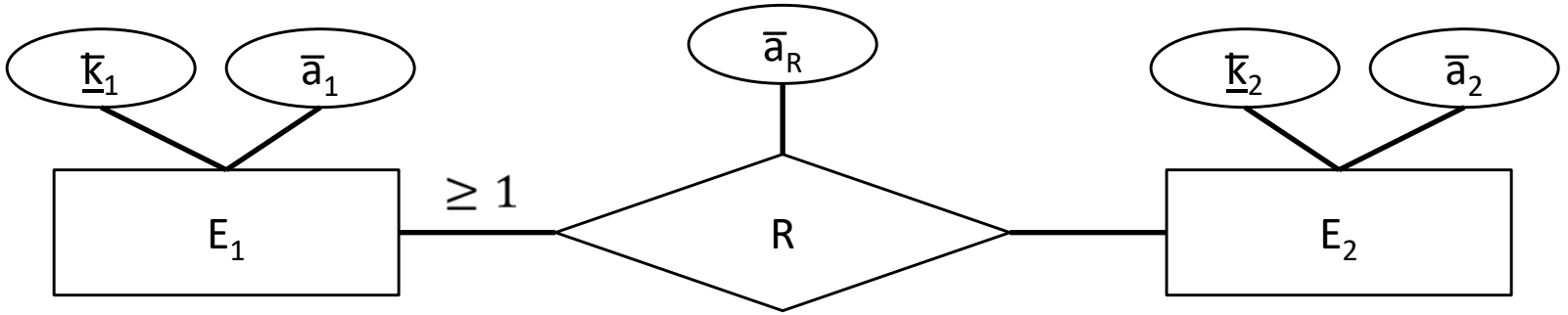
E_2:

$\underline{k}_2$	$\bar{a}_2$

R:

$\underline{k}_1$	$\bar{a}_R$	$\underline{k}_2$

Relationships – degree constraint



- Note that it always holds that:
  - $\pi_{k2}(R) \subseteq \pi_{k2}(E_2)$
  - $\pi_{k1}(R) \subseteq \pi_{k1}(E_1)$

E_2:

$\pi_{k2}(E_2)$	
$\underline{k}_2$	$\bar{a}_2$

E_1:

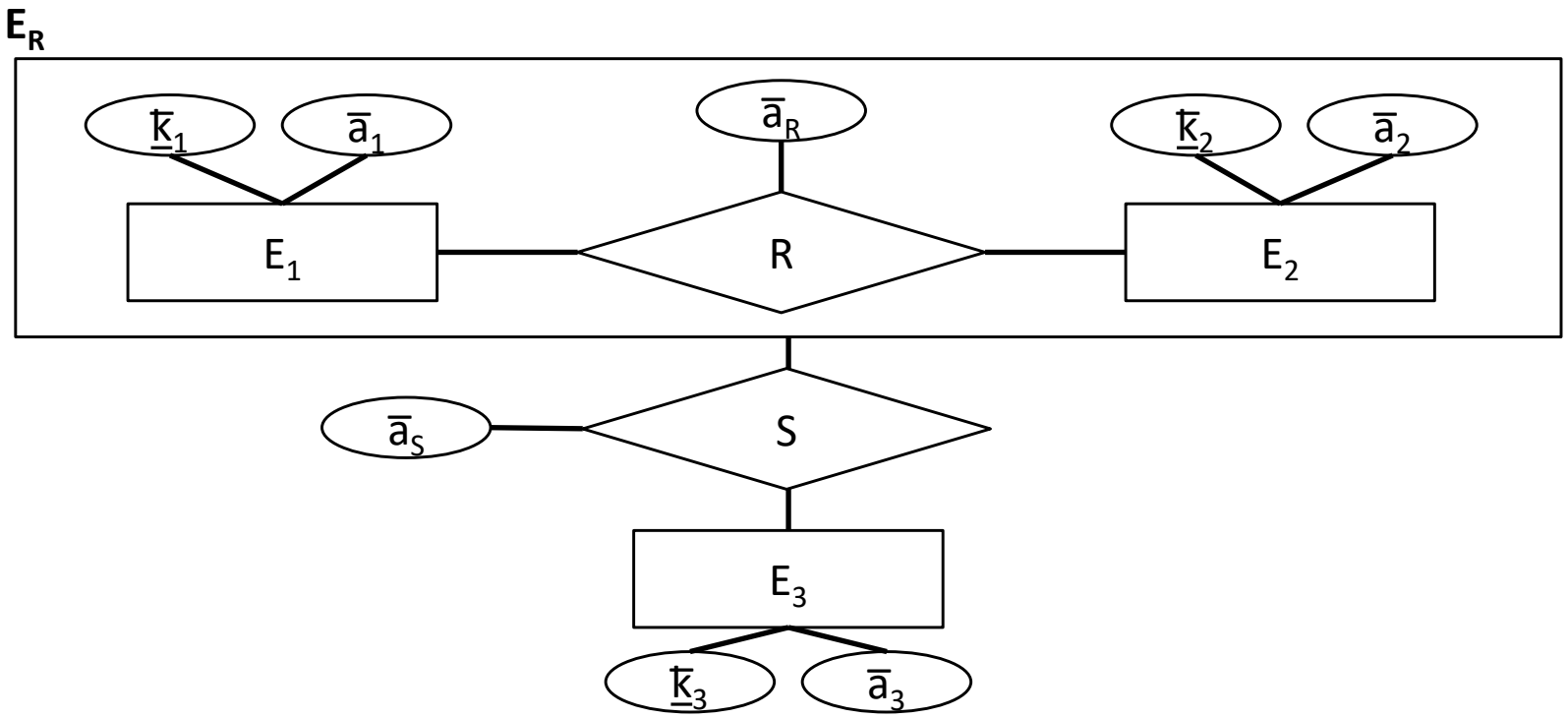
$\pi_{k1}(E_1)$	
$\underline{k}_1$	$\bar{a}_1$

R:

$\pi_{k1}(R)$		$\pi_{k2}(R)$
$\underline{k}_1$	$\bar{a}_R$	$\underline{k}_2$

Reichman University

Aggregations



- Turns the relationship into an entity with attributes of the relationship

Aggregations

E1:

$\underline{k}_1$	$\bar{a}_1$

E2:

$\underline{k}_2$	$\bar{a}_2$

E3:

$\underline{k}_3$	$\bar{a}_3$

R:

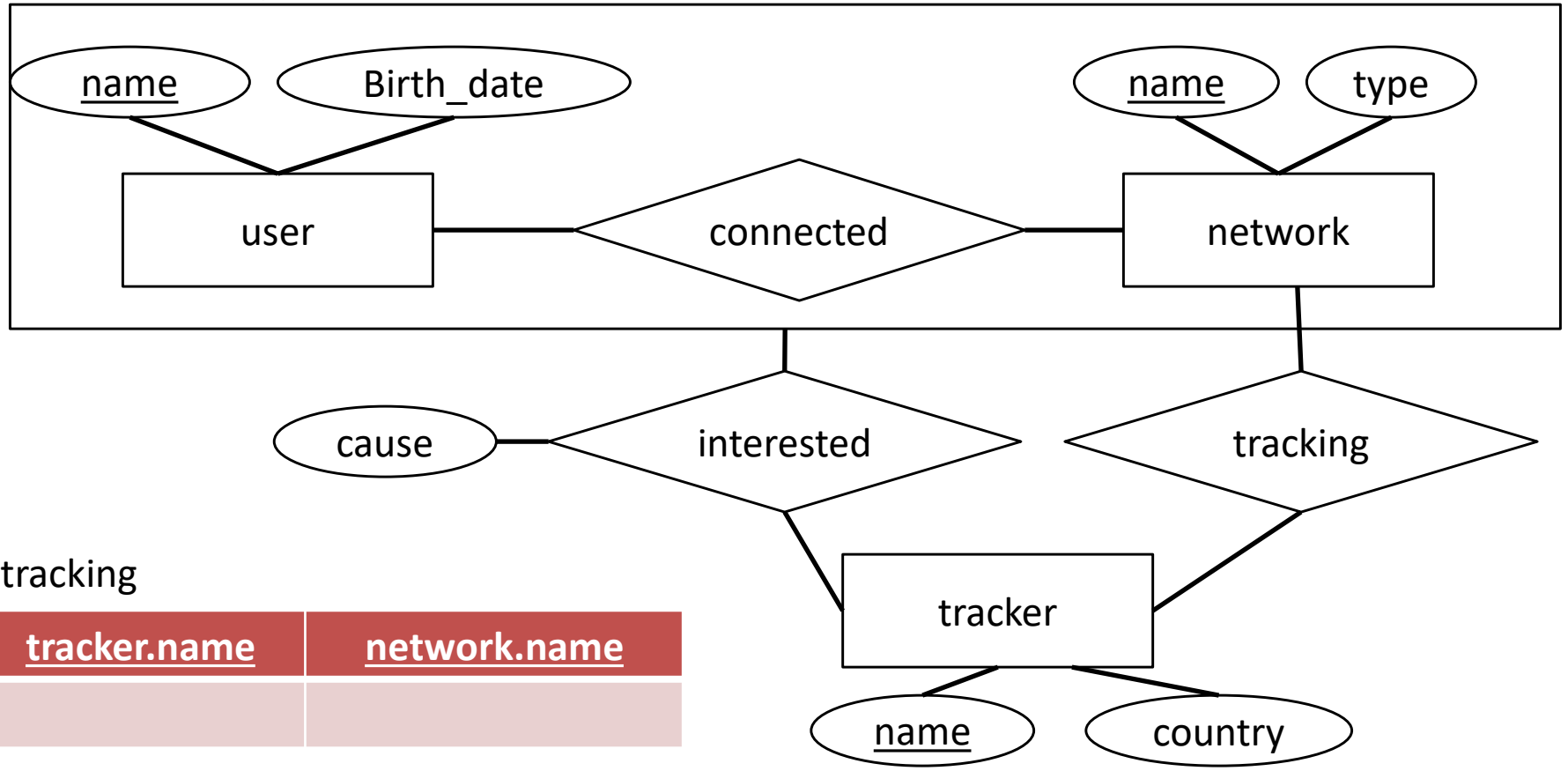
$\underline{k}_1$	$\underline{k}_2$	$\bar{a}_R$

S:

$\underline{k}_1$	$\underline{k}_2$	$\underline{k}_3$	$\bar{a}_S$

Reichman University

Example



tracking

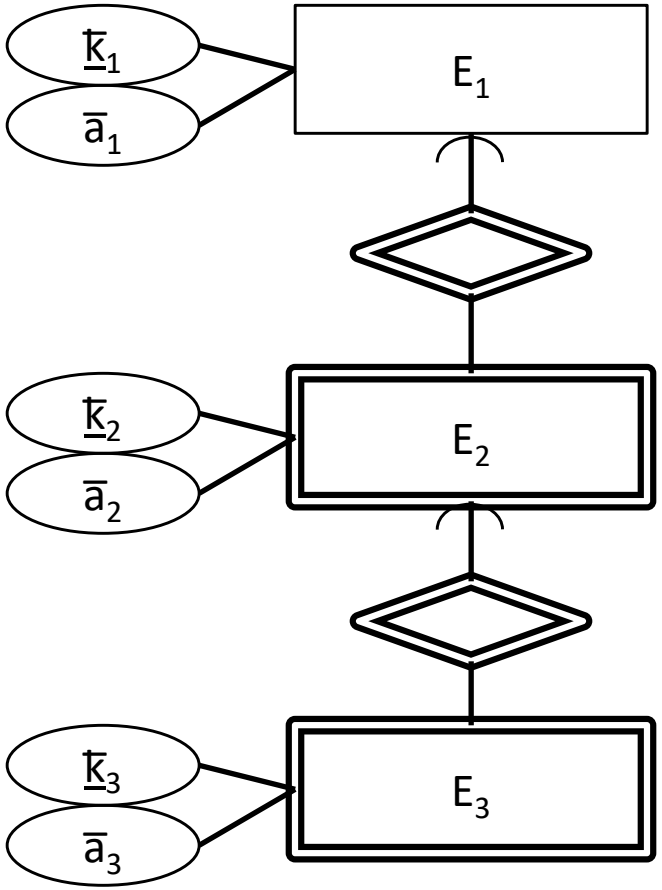
tracker.name	network.name

interested

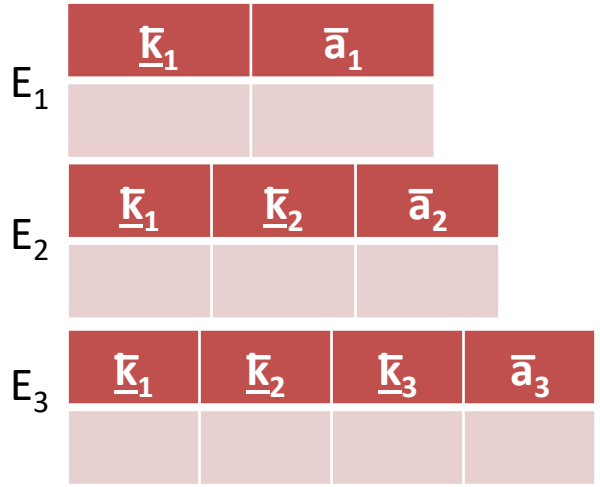
tracker.name	network.name	user.name	cause

Reichman University

Weak Entities



Translation to tables:



Constraints:

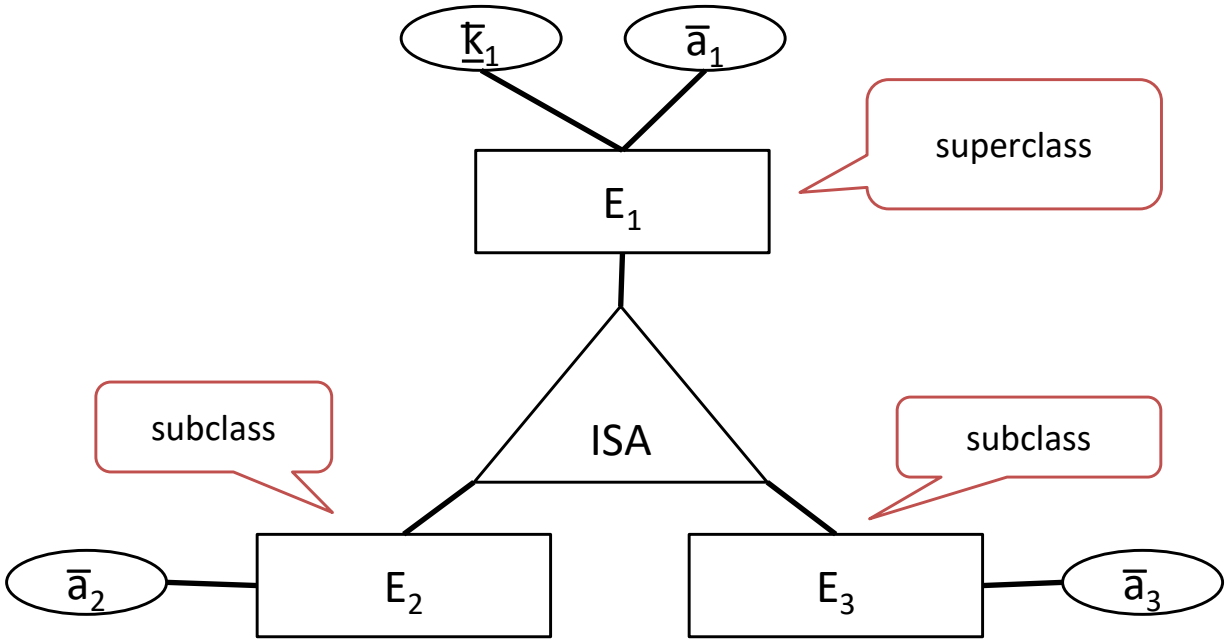
$\pi_{k1}(E_2) \subseteq \pi_{k1}(E_1)$

$\pi_{k1,k2}(E_3) \subseteq \pi_{k1,k2}(E_2)$

Reichman University

ISA

- ISA – a branching weak entity without key components in the *subclass*



Reichman University

ISA – Translations and Constraints

Dessert:

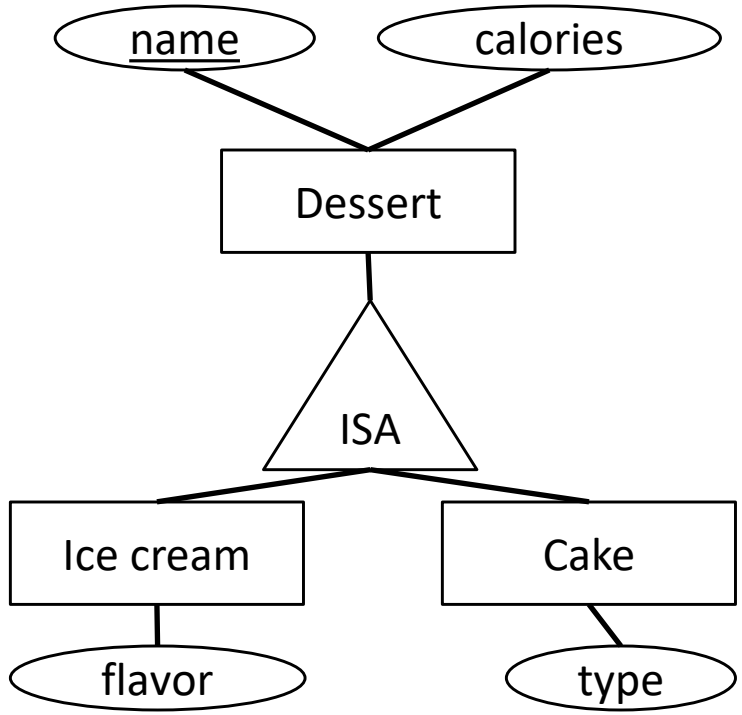
name	calories

Ice cream:

name	flavor

Cake:

name	type



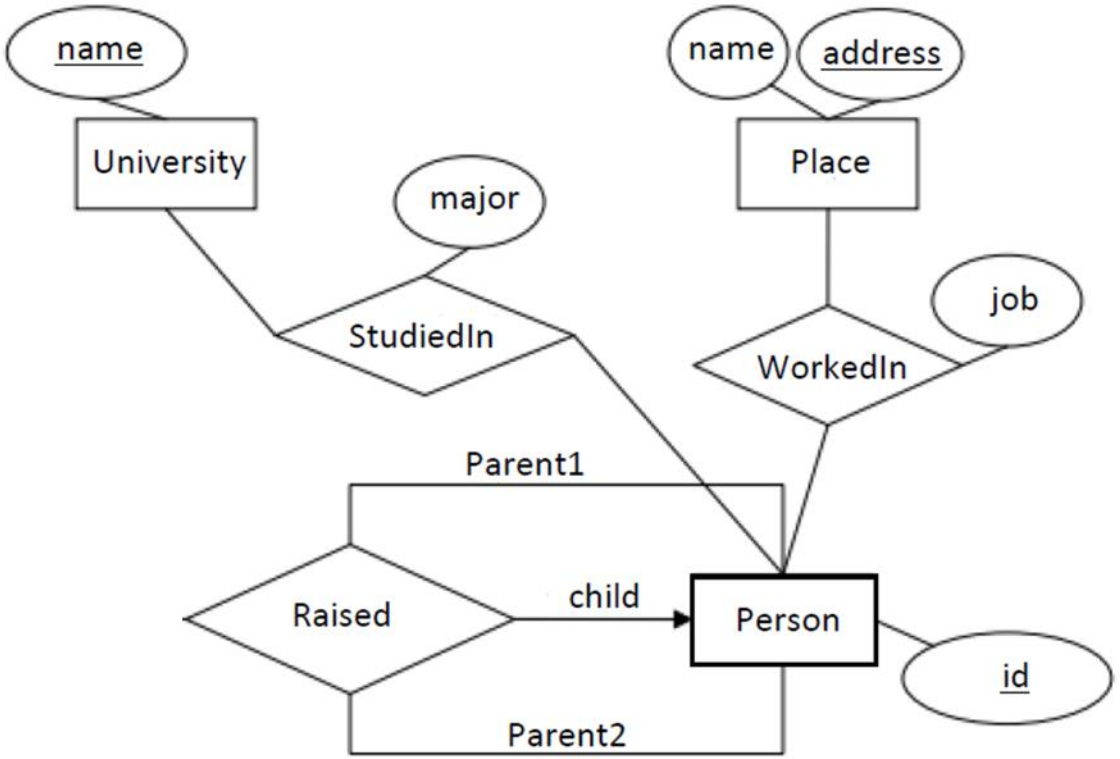
Constraints:

$\pi_{name}(Cake) \subseteq \pi_{name}(Dessert)$

$\pi_{name}(Ice\ cream) \subseteq \pi_{name}(Dessert)$

Reichman University

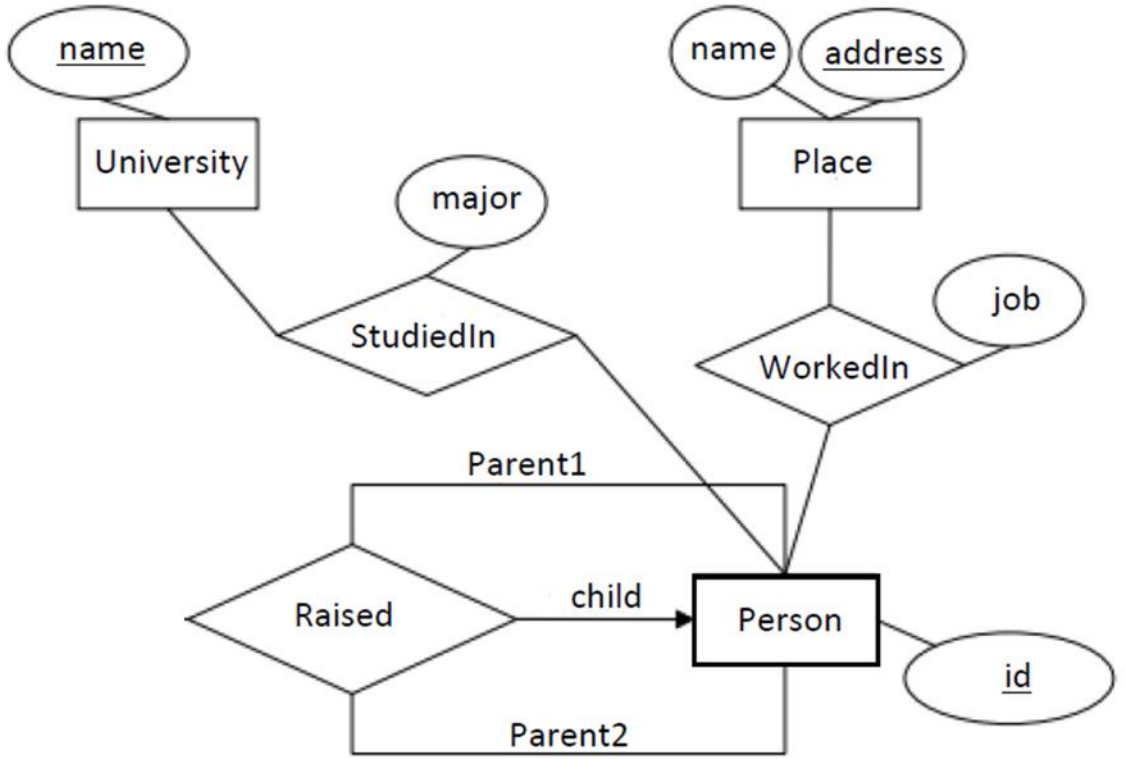
Question 1 - ERD



Reichman University

Question 1a

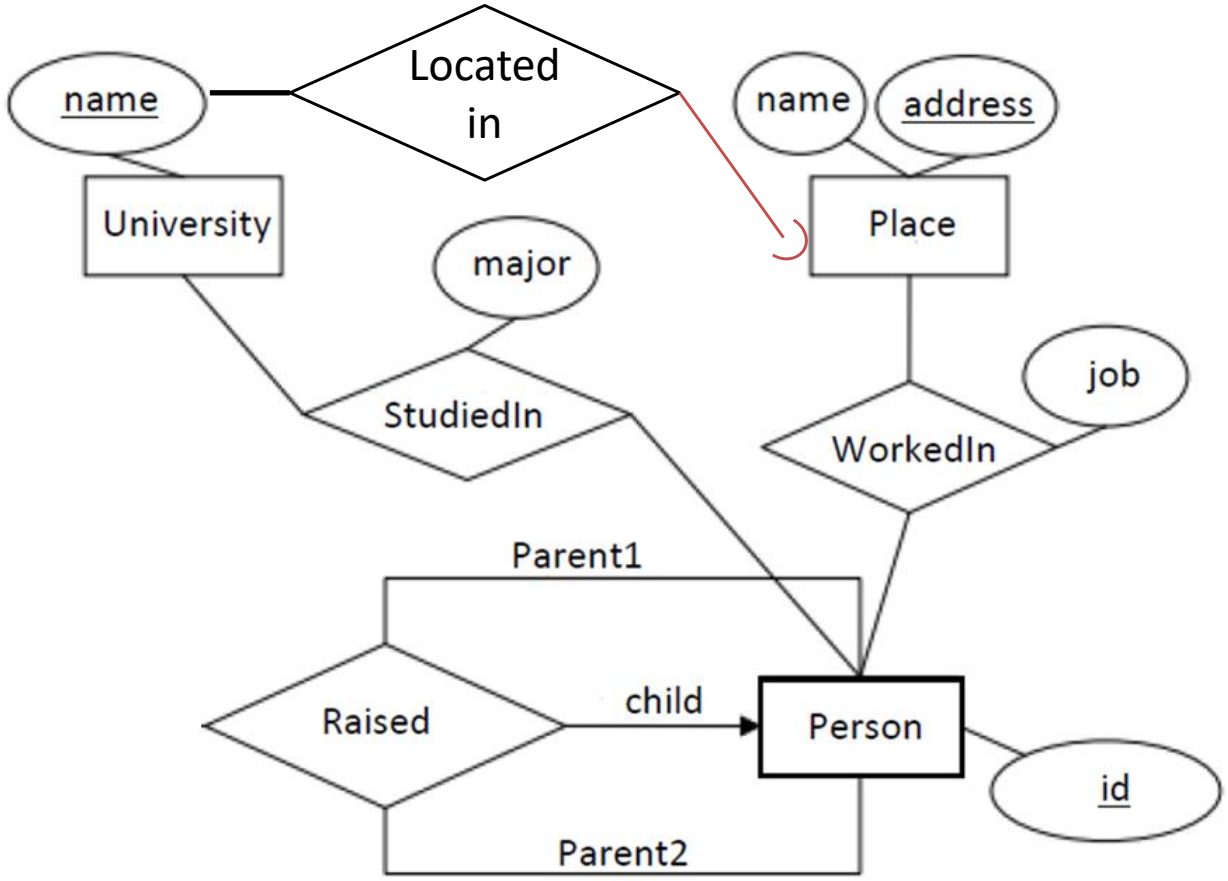
Modify the ERD diagram such that for each university its place will be stored (without adding new entities).



Reichman University

Question 1a

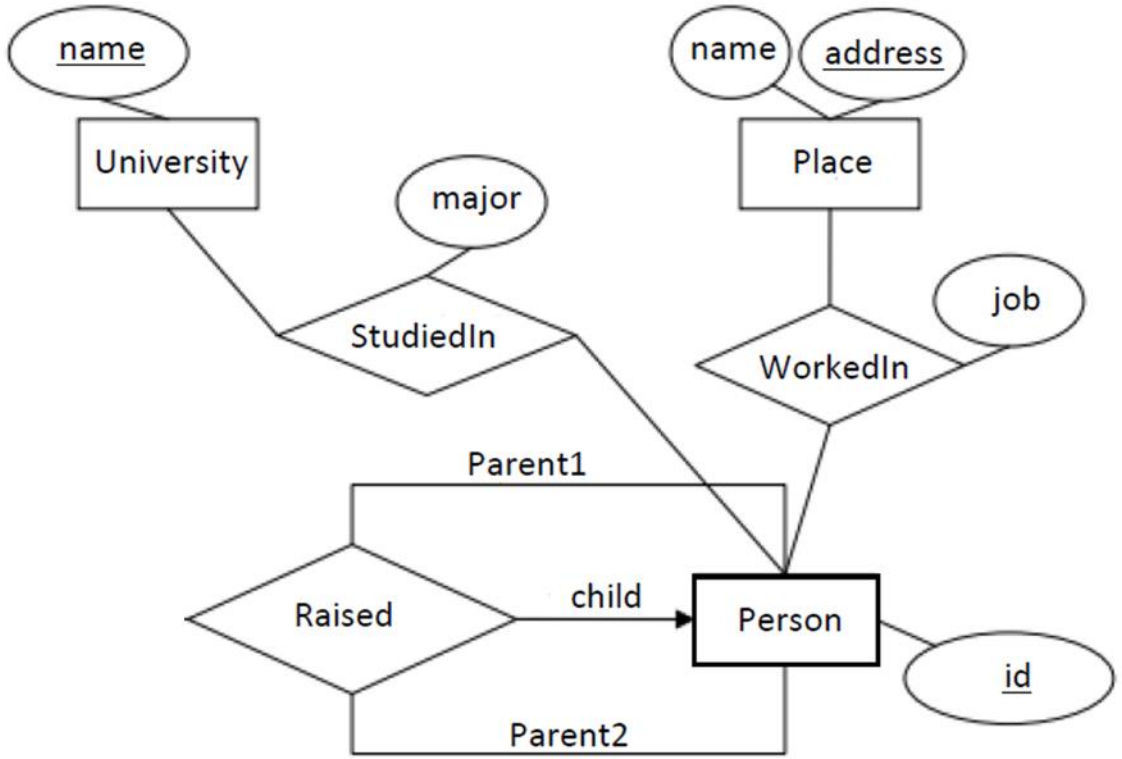
Modify the ERD diagram such that for each university its place will be stored (without adding new entities).



Reichman University

Question 1b

Can parents raise 2 children?



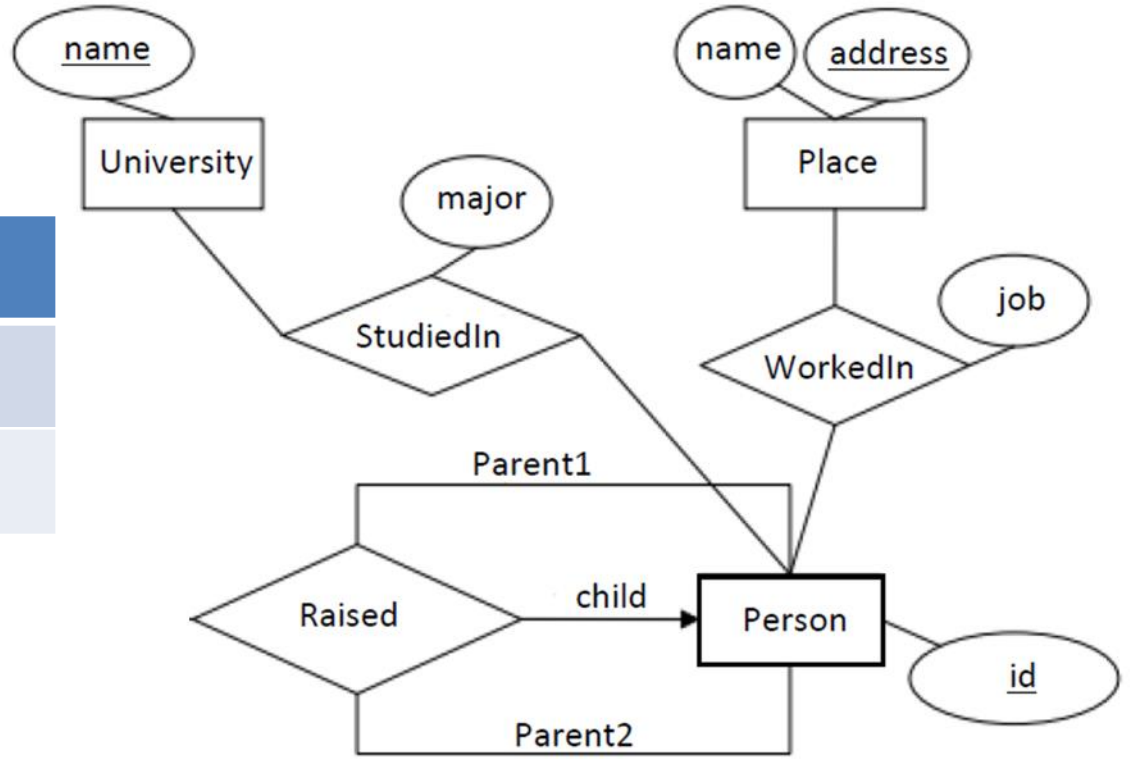
Reichman University

Question 1b

Can parents raise 2 children?

Raised table:

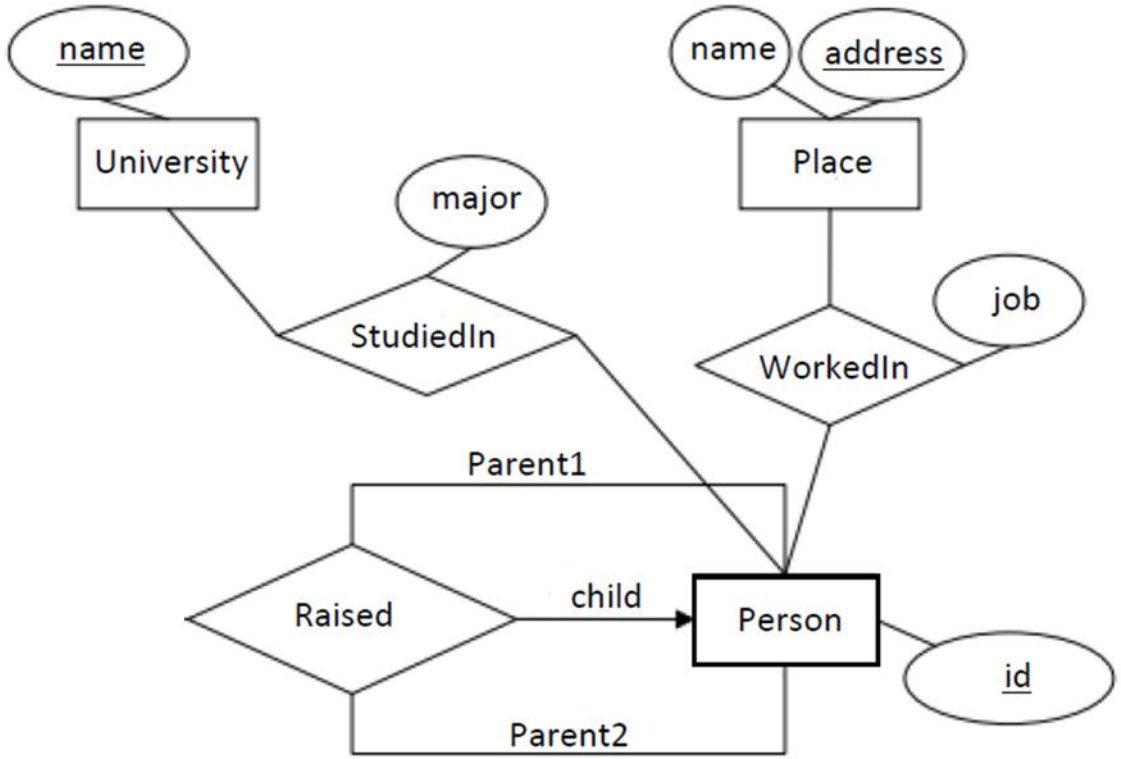
Parent1.id	Parent2.id	Child.id
11	22	33
22	11	44



Reichman University

Question 1c

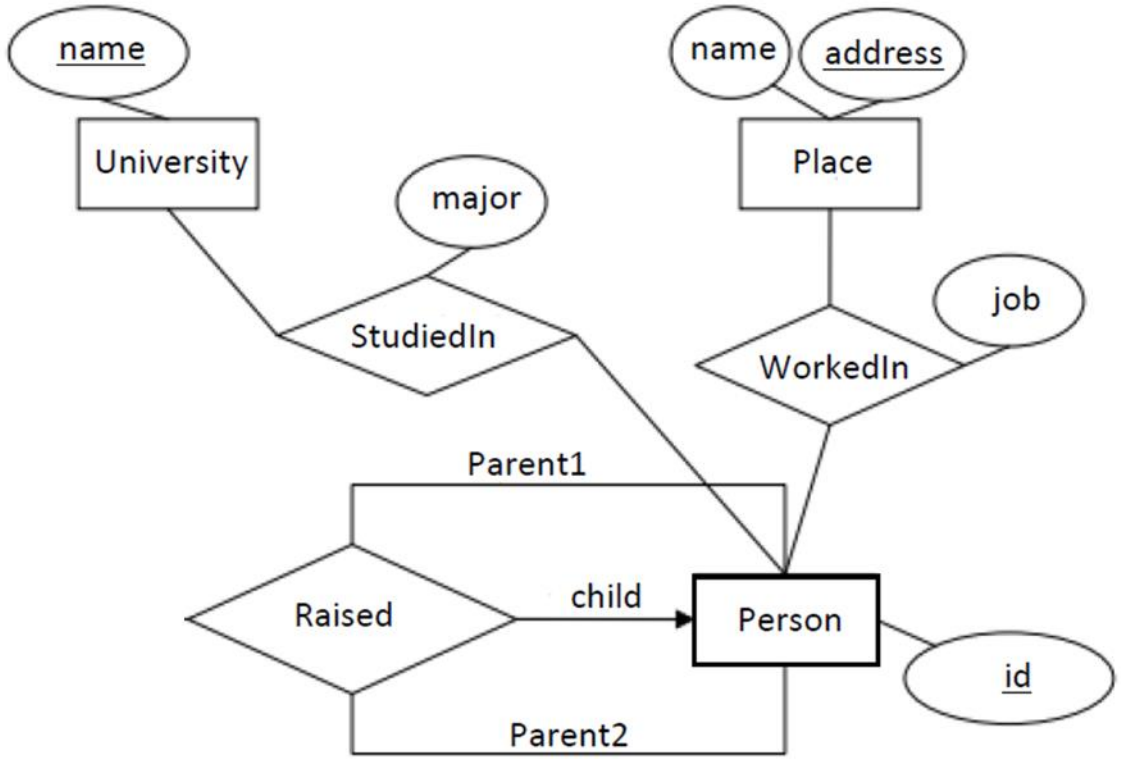
Can parents raise 3 children?



Reichman University

Question 1c

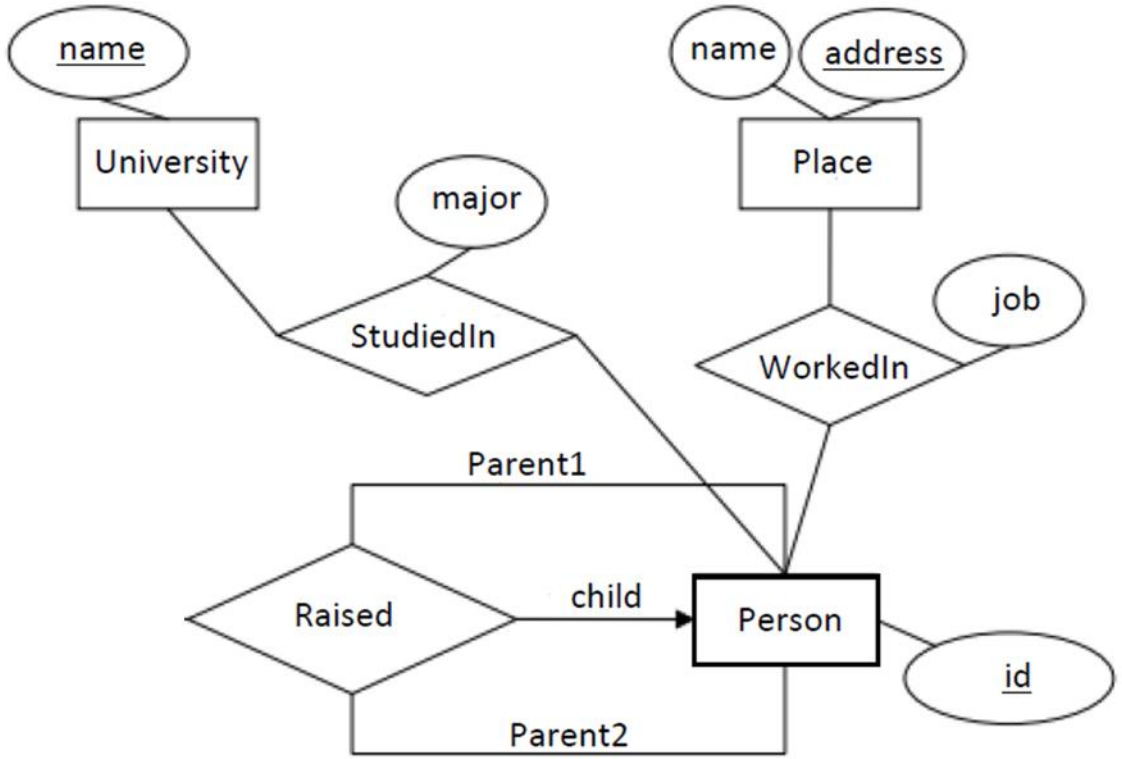
Can parents raise 3 children? no



Reichman University

Question 1d

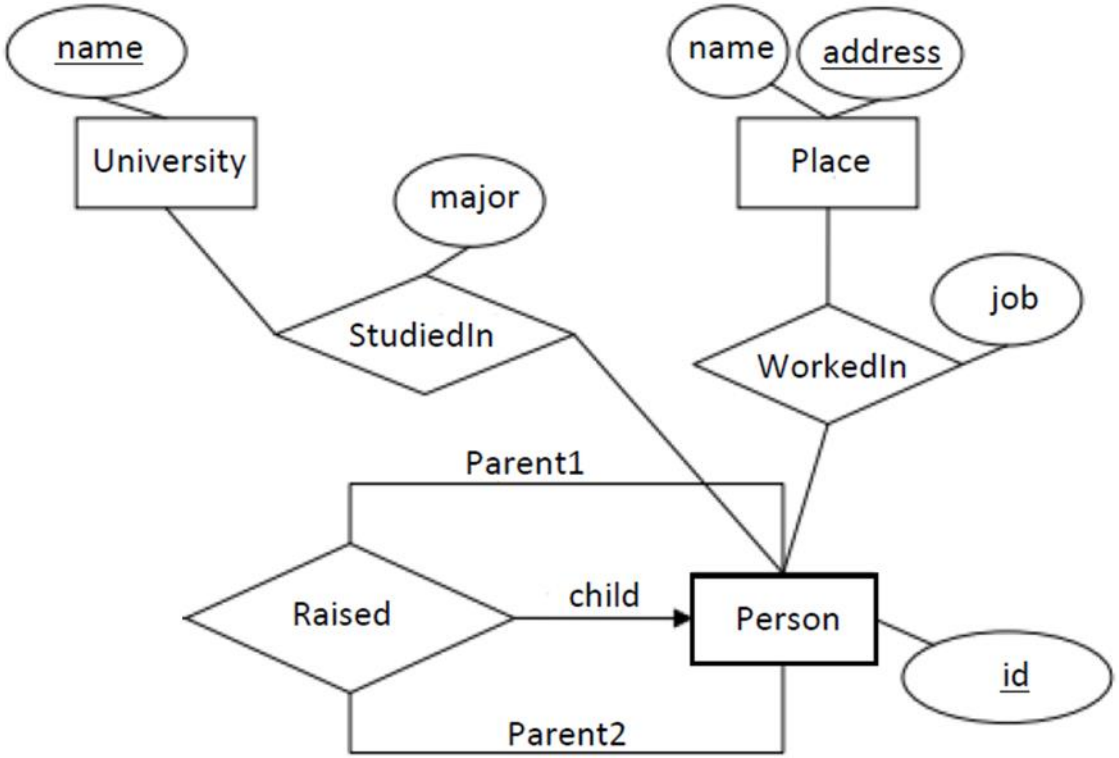
How many parents can raise one child?



Reichman University

Question 1d

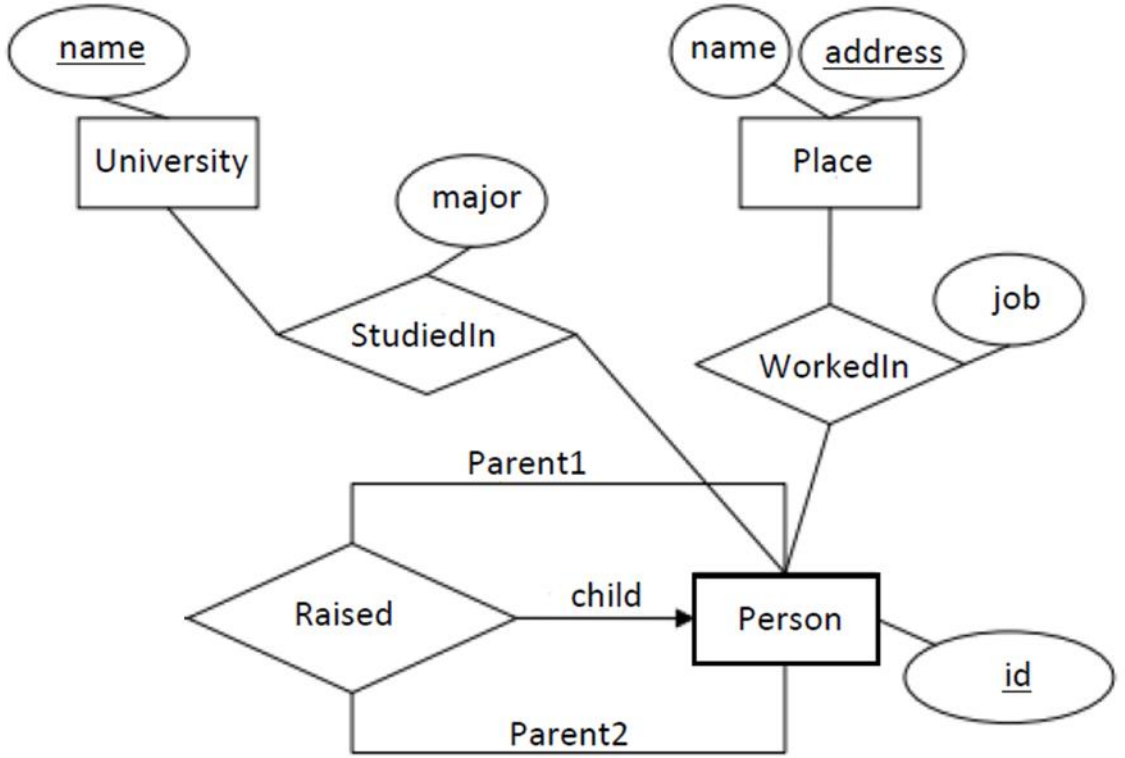
How many parents can raise one child? No limit for now



Reichman University

Question 1e

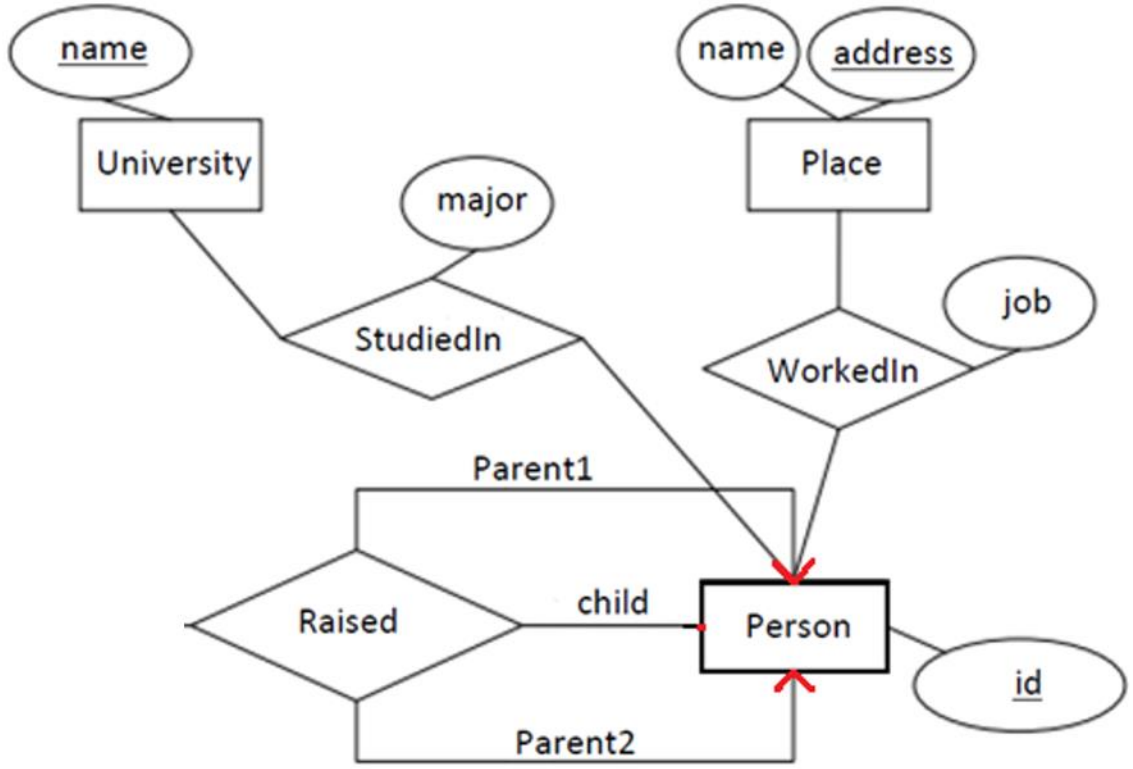
How can we enforce that each child will be raised by at most one pair of parents?



Reichman University

Question 1e – 1st try

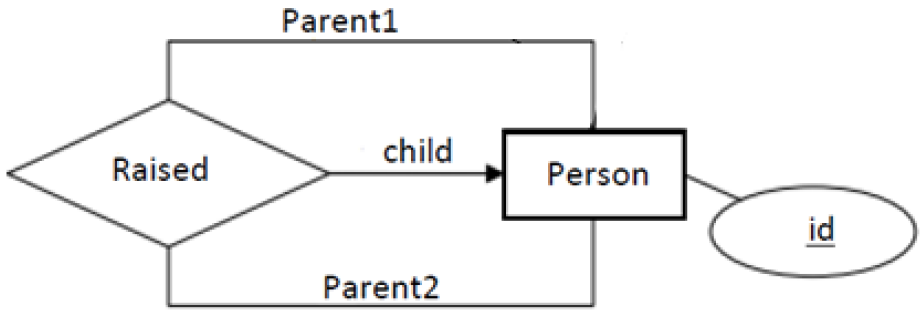
How can we enforce that each child will be raised by at most one pair of parents?

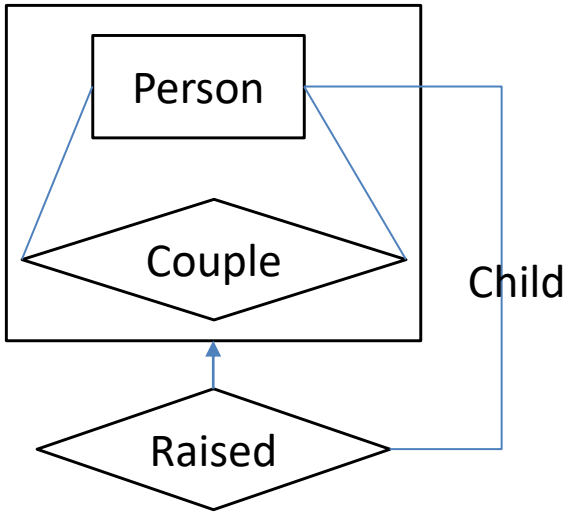


Reichman University

Question 1e – 2nd try

How can we enforce that each child will be raised by at most one pair of parents?





Reichman University