

Mapping of ER to Relational Database Model (RDB)

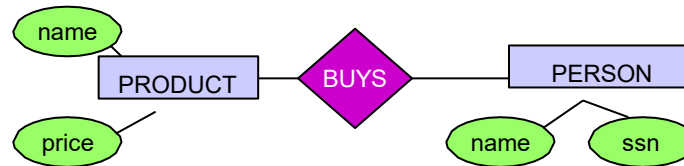
Trishla Shah

trishla.shah@smu.ca

trishla.it@gmail.com

Conceptual and Logical Design

Conceptual Model:



Relational Model:

What is Mapping?

- We cannot store data in an ER schema (there are no ER database management systems)
- We have to translate our ER schema into a relational schema.

What does this translation mean?

- We cannot import all the ER constraints into relational model, but an approximate schema can be generated.
- There are several processes and algorithms available to convert ER Diagrams into Relational Schema. Some of them are automated and some of them are manual.

Translation : Principles

- Maps
 - ER schemas to relational schemas
 - ER Instances to relational instances
- Ideally, the mapping should
 - be **one-to-one** in both directions
 - **not lose any information**

Difficulties

- what to do with ER-instances that have **identical attribute values**, but consist of **different entities**?
- **In which way** do we want to preserve information?

Mapping Entity Types to Relation

- An entity is a real-world object with some attributes.
- For every *entity type* create a *relation*.

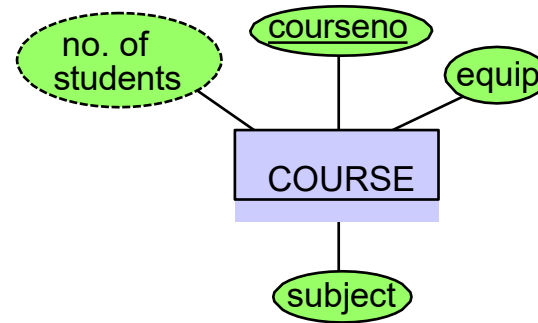
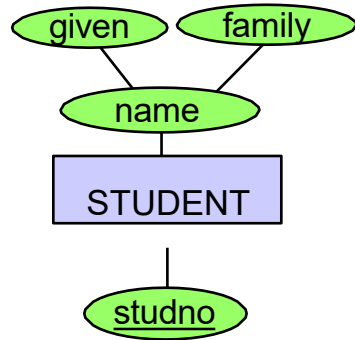
Mapping Process (Algorithm)

- Create table for each entity.
- Entity's attributes should become fields of tables with their respective data types.
- Declare primary key.

Things to NOTE in DANGER!!!

- Every *atomic attribute* of the entity type becomes a *relation attribute*
- *Composite attributes*: include *all the atomic attributes*
- *Derived attributes* are not included (*but remember their derivation rules*)
- Relation instances are subsets of the cross product of the domains of the attributes
- Attributes of the *entity key* make up the *primary key* of the relation

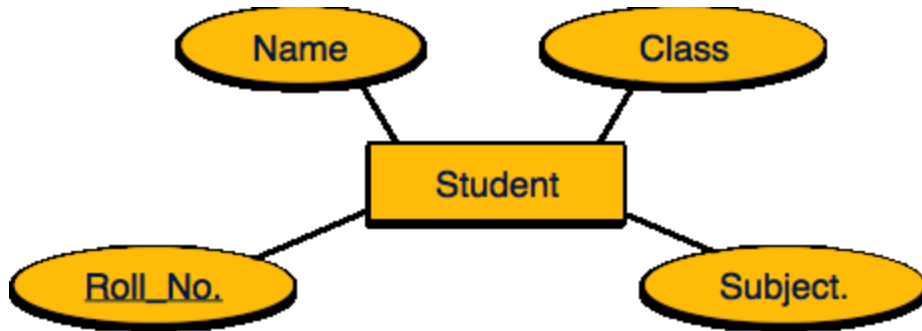
Mapping Entity Types to Relation (Cntd..)



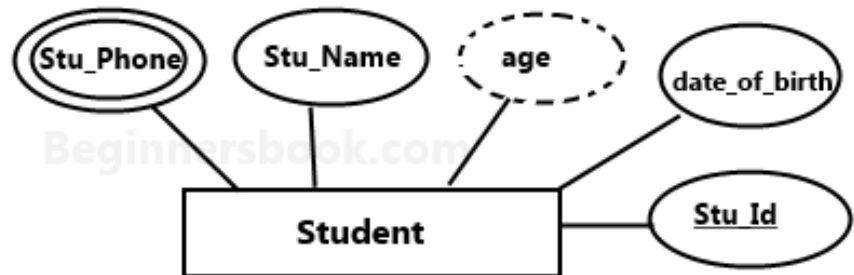
STUDENT (studno, givenname, familyname)

COURSE (courseno, subject, equip)

Can you do the same for these ER?



1



2

Mapping Many:many Relationship Types to Relations

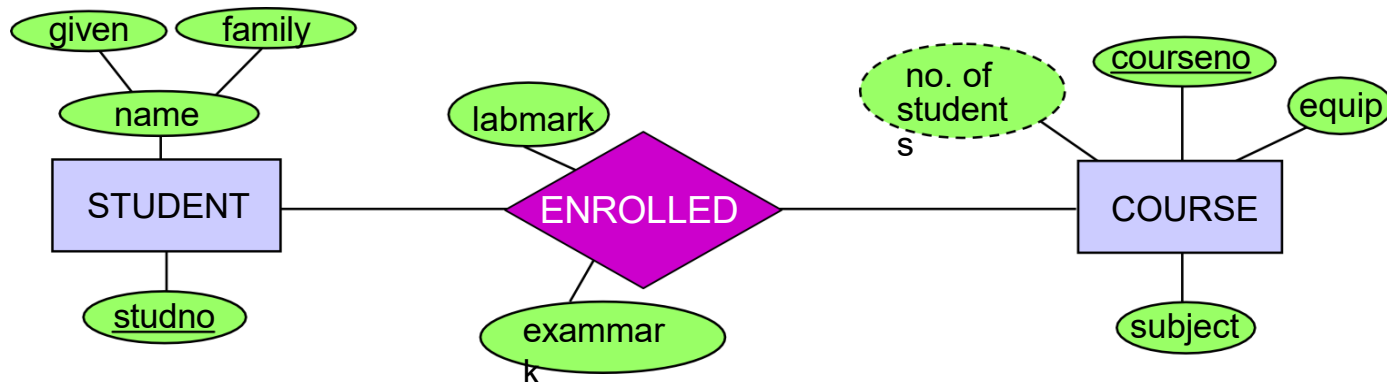
Create a relation with the following set of attributes:

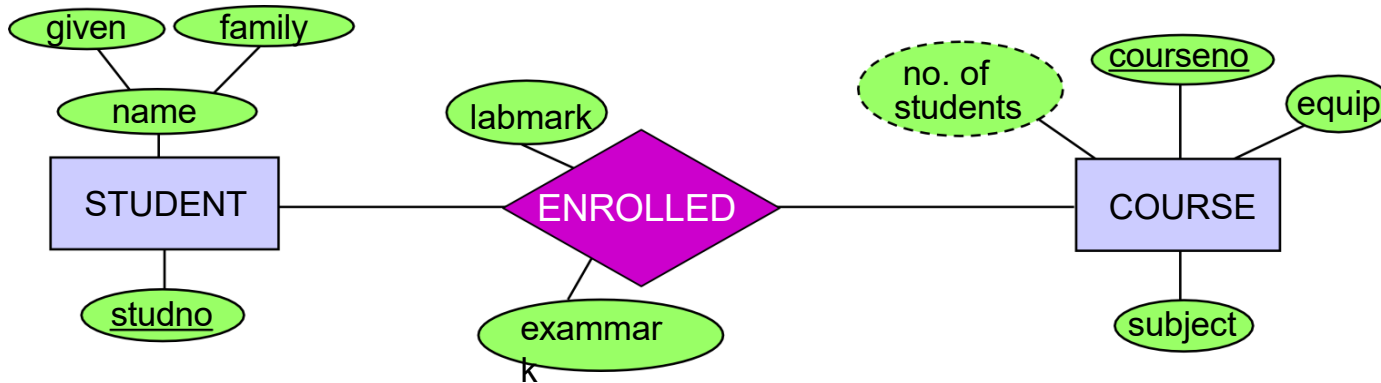
N (degree of relationship)

$$\bigcup_{i=1}^N \text{primary_key}(E_i) \cup \{a_1, \dots, a_M\}$$

*primary keys of each
entity type participating
in the relationship*

*attributes of the
relationship type (if any)*



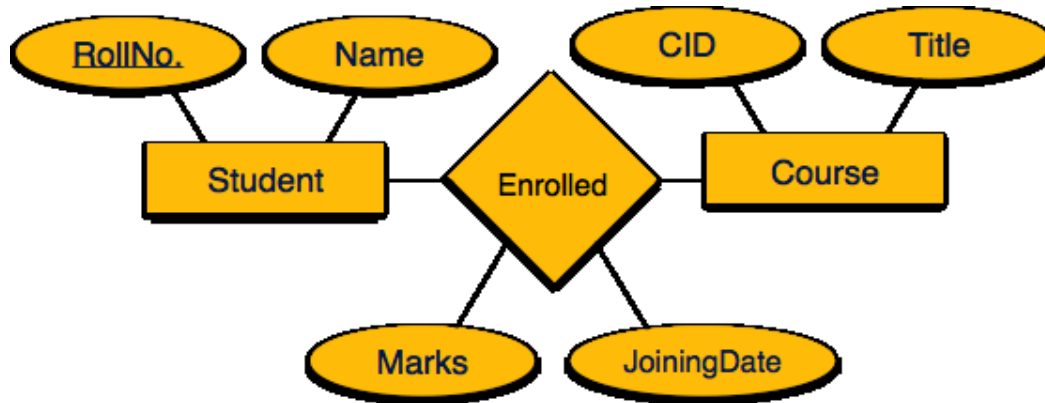


ENROL(studno, courseno, labmark, exammark)

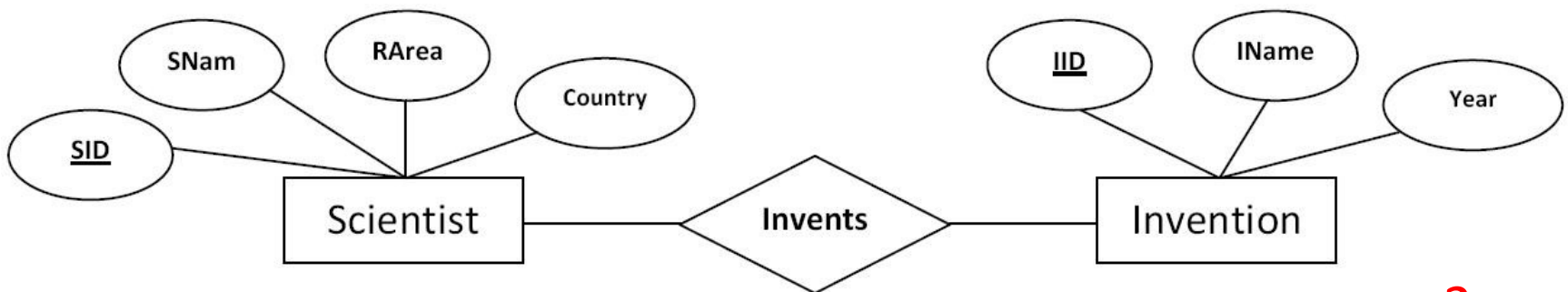
Foreign Key ENROL(studno) references STUDENT(studno)

Foreign Key ENROL(courseno) references COURSE(courseno)

Can you do the same for these ER?

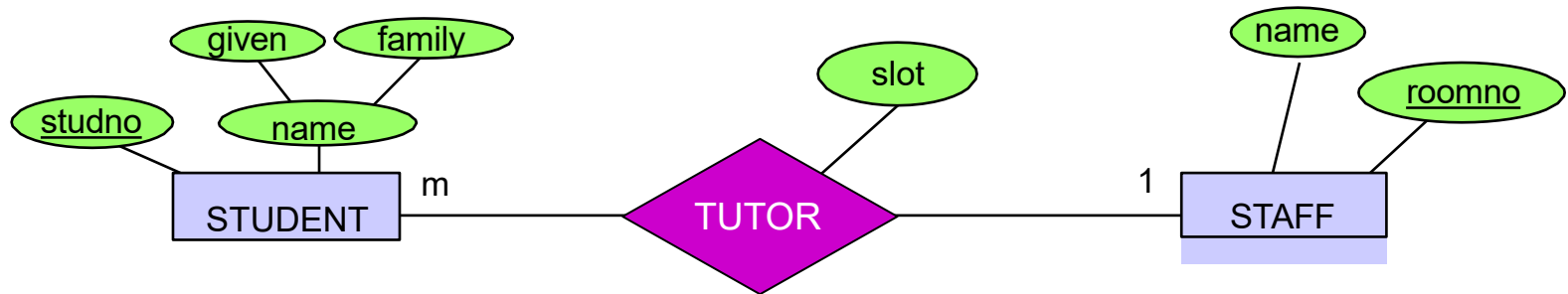


1



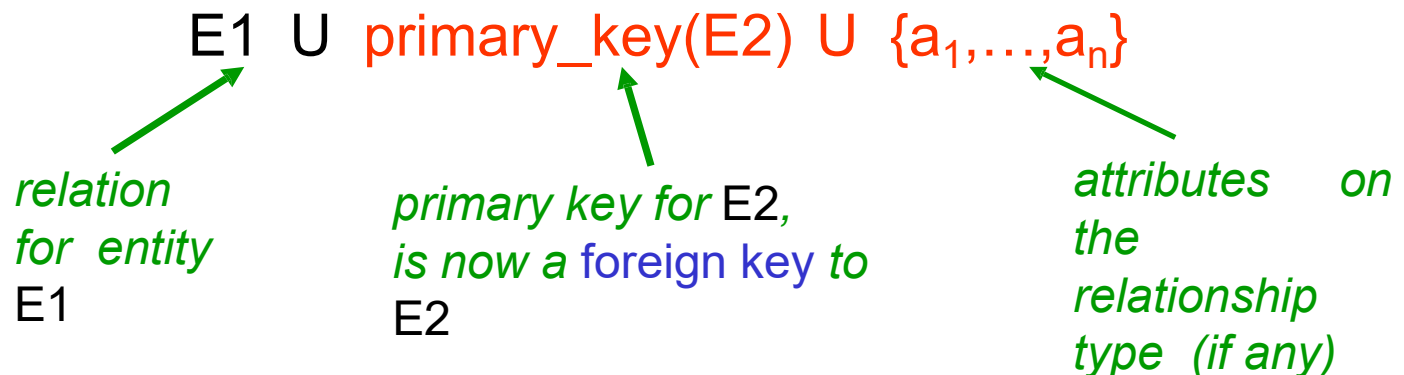
2

Mapping Many:one Relationship Types to Relations



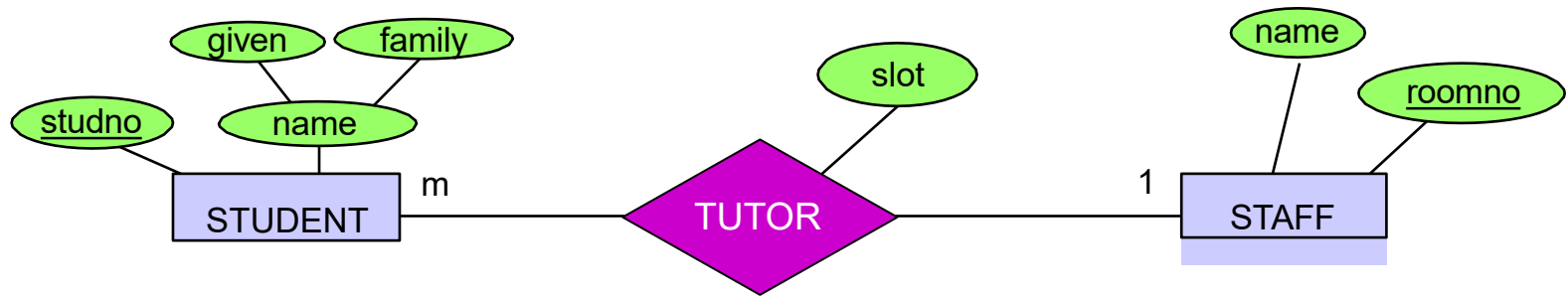
Idea: “*Post the primary key*”

- Given E1 at the ‘many’ end of relationship and E2 at the ‘one’ end of the relationship, add information to the relation for E1
- The primary key of the entity at the ‘one’ end (the *determined* entity) becomes a foreign key in the entity at the ‘many’ end (the *determining* entity). Include any relationship attributes with the foreign key entity



Mapping Many:one Relationship Types

Example



The relation

STUDENT(studno, givenname, familyname)

is extended to

STUDENT(studno, givenname, familyname, **tutor**, **roomno**, **slot**)

and the constraint

Foreign Key STUDENT(tutor,roomno) references STAFF(name,roomno)

Mapping Many:one Relationship Types

Example

STUDENT

<u>studno</u>	given	family	tutor	roomno	slot
s1	fred	jones	bush	2.26	12B
s2	mary	brown	kahn	IT206	12B
s3	sue	smith	goble	2.82	10A
s4	fred	bloggs	goble	2.82	11A
s5	peter	jones	zobel	2.34	13B
s6	jill	peters	kahn	IT206	12A

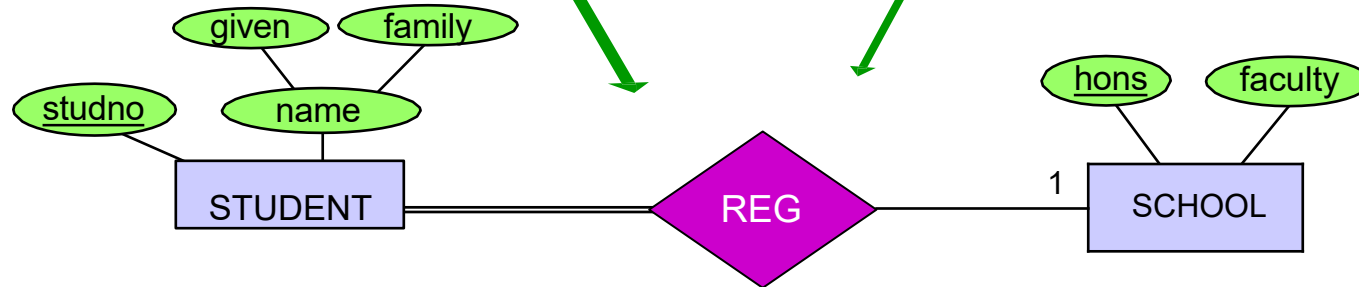
STAFF

<u>name</u>	<u>roomno</u>
kahn	IT206
bush	2.26
goble	2.82
zobel	2.34
watson	IT212
woods	IT204
capon	A14
lindsey	2.10
barringer	2.125

Optional Participation of the Determined Entity ('one end')

A student entity instance must participate in a relationship instance of REG

A school entity instance need not participate in a relationship instance of REG



SCHOOL (hons, faculty)

STUDENT (studno, givenname,
familyname,

???)

Optional Participation of the Determined Entity

STUDENT

<u>studno</u>	given	family	hons
s1	fred	jones	ca
s2	mary	brown	cis
s3	sue	smith	cs
s4	fred	bloggs	ca
s5	peter	jones	cs
s6	jill	peters	ca

“hons” cannot be NULL because it is mandatory for a student to be registered for a school

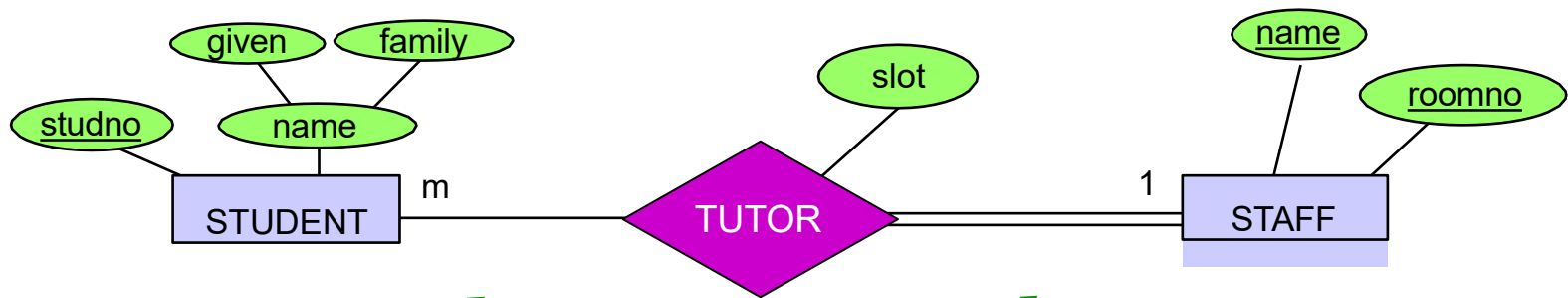
“not null” constraint

SCHOOL

<u>hons</u>	faculty
ac	accountancy
is	information systems
cs	computer science
ce	computer science
mi	medicine
ma	mathematics

No student is registered for “mi”, so “mi” doesn’t occur as a foreign key value (but that’s no problem)

Optional Participation of the Determinant Entity ('many end')



A student entity instance need not participate in a relationship instance of TUTOR

A staff entity instance must participate in a relationship instance of TUTOR

Optional Participation of the Determinant Entity

1. STUDENT (studno, givenname, familyname, tutor, roomno, slot)
STAFF(name, roomno)

Integrity constraint:

$$\pi_{\text{name, roomno}} \text{STAFF} \setminus \pi_{\text{tutor, roomno}} \text{STUDENT} = \emptyset$$

2. STUDENT(studno, givenname, familyname)
STAFF(name, roomno)
TUTOR(studno, tutor, roomno, slot)

Do we also need an integrity constraint?

Optional Participation of the Determinant Entity (Cntd..)

STUDENT

<u>studno</u>	given	family	tutor	roomno	slot
s1	fred	jones	bush	2.26	12B
s2	mary	brown	kahn	IT206	12B
s3	sue	smith	goble	2.82	10A
s4	fred	bloggs	goble	2.82	11A
s5	peter	jones	zobel	2.34	13B
s6	jill	peters	kahn	IT206	12A

STAFF

<u>name</u>	<u>roomno</u>
kahn	IT206
bush	2.26
goble	2.82
zobel	2.34
watson	IT212
woods	IT204
capon	A14
lindsey	2.10
barringer	2.125

Optional Participation of the Determinant Entity (Cntd..)

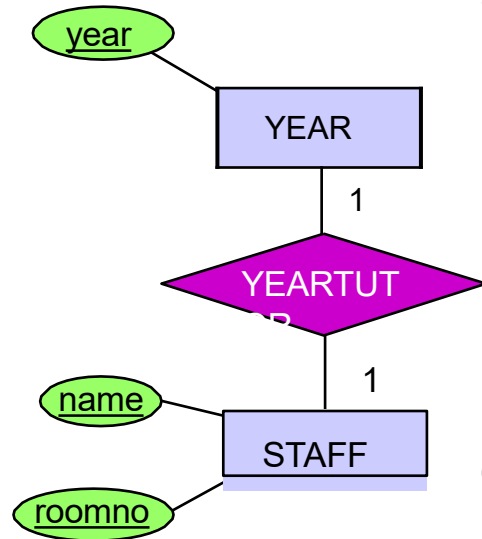
STUDENT

<u>studno</u>	given	family	tutor	roomno	slot
s1	fred	jones	bush	2.26	12B
s2	mary	brown	kahn	IT206	12B
s3	sue	smith	goble	2.82	10A
s4	fred	bloggs	goble	2.82	11A
s5	peter	jones	NULL	NULL	NUL L
s6	jill	peters	kahn	IT206	12A

STAFF

<u>name</u>	<u>roomno</u>
kahn	IT206
bush	2.26
goble	2.82
zobel	2.34
watson	IT212
woods	IT204
capon	A14
lindsey	2.10
barringer	2.125

Mapping One: one relationship types to Relations



- Post the primary key of one of the entity types into the other entity type as a foreign key, including any relationship attributes with it

or

- Merge the entity types together

YEAR

<u>year</u>	yeartutor
1	zobel
2	bush
3	capo n

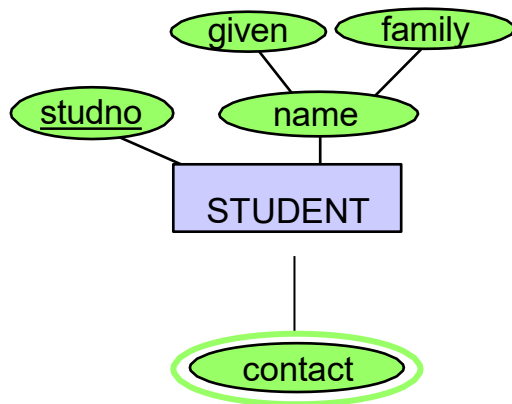
STAFF

<u>name</u>	<u>roomno</u>	year
kahn	IT206	NULL
bush	2.26	2
goble	2.82	NULL
zobel	2.34	1
watson	IT212	NULL
woods	IT204	NULL
capon	A14	3
lindsey	2.10	NULL
barringer	2.125	NULL ₂₁

Multi Valued Attributes

For each multi-valued attribute of E_i , create a relation with the attributes
 $\text{primary_key}(E_i) \cup \text{multi-valued attribute}$.

The primary key comprises *all* attributes



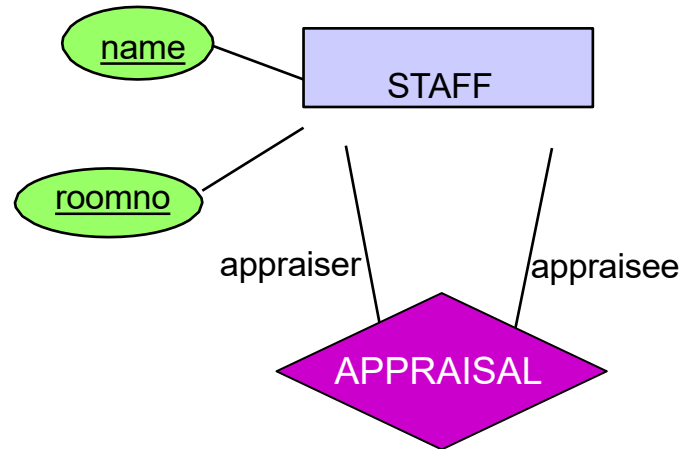
STUDENT

<u>studno</u>	given	family
s1	fred	jones
s2	mary	brown

STUDENT_CONTACT

<u>studno</u>	<u>contact</u>
s1	Mr. Jones
s1	Mrs Jones
s2	Bill Brown
s2	Mrs Jones
s2	Billy-Jo Woods

Mapping Roles and Recursive Relationships

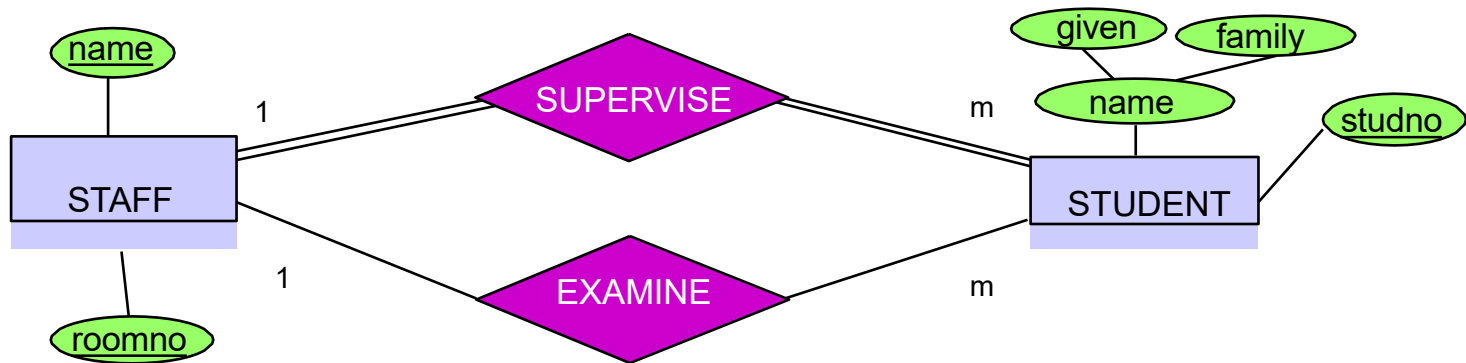


How can the entity STAFF appear in both of its roles ?

STAFF(name, roomno, appraiser, approomno)

Multiple Relationship between Entity types

1. Treat each relationship type separately
2. Represent distinct relationships by different foreign keys drawing on the same relation



STAFF(name, roomno)

STUDENT(studno, given, family, name)

STUDENT(studno, given, family, name)

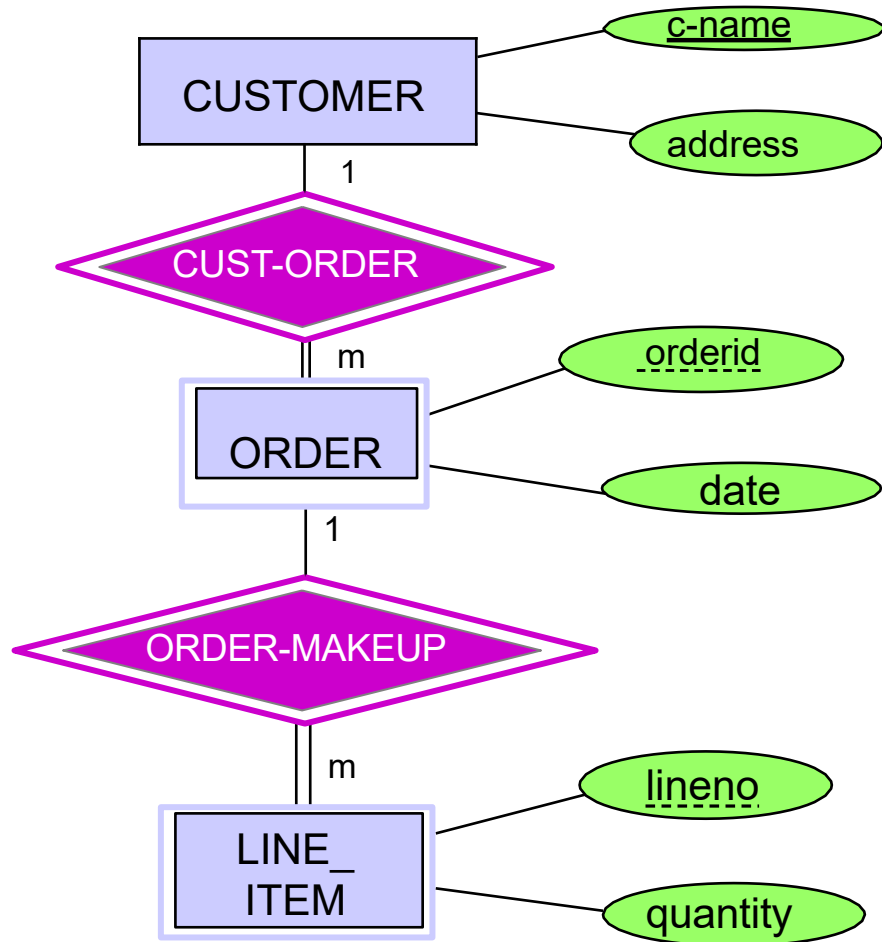
EXAMINER(studno, given, family, name)

SUPERVISOR(studno, given, family, name)

EXAMINER-SUPERVISOR(studno, given, family, name)

Weak Entities

- Strong entity type
- Identifying entity for ORDER
- Identifying entity for LINE_ITEM
- Weak entity type
- Identifying entity for LINE_ITEM
- Weak entity type

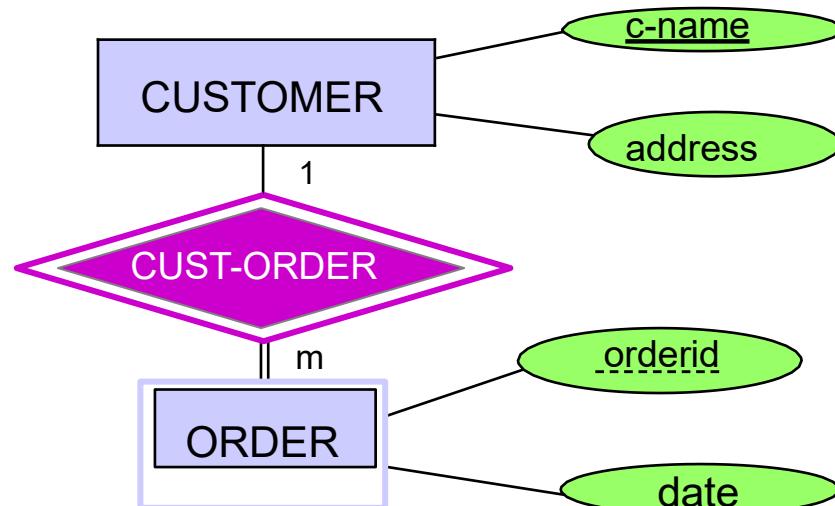


Mapping Weak entities to a relation

Create a relation with the attributes:

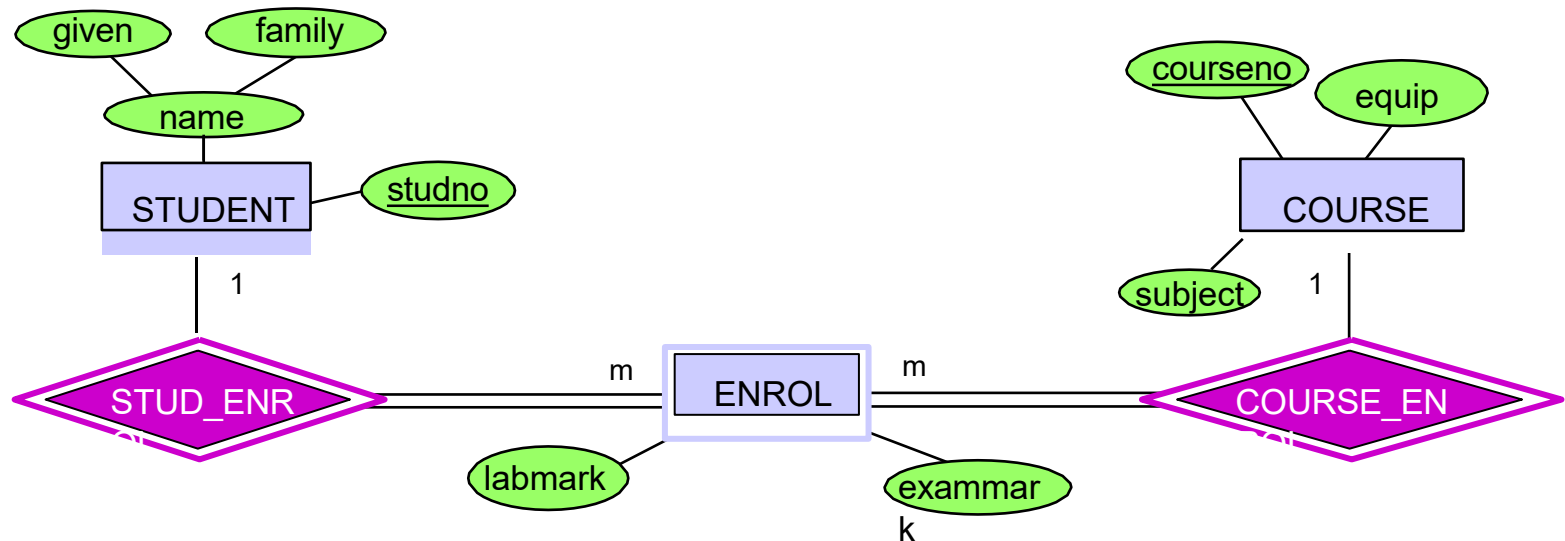
$$\text{primary_key}(E_0) \cup \bigcup_{i=1}^n \text{discriminator}(E_i) \cup \{a_1, \dots, a_n\}$$

Primary key of identifying strong entity type *Discriminators of identifying weak entity types* *Attributes of the weak entity type*



Association Entity Types

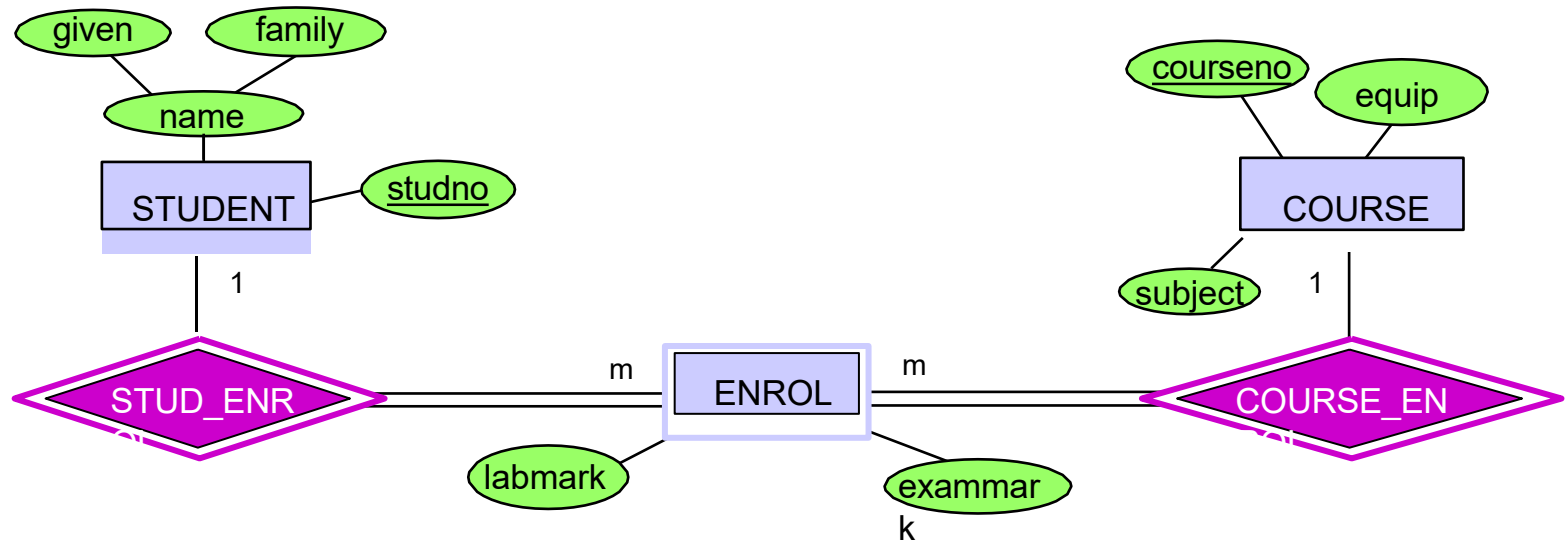
An entity type that represents a relationship type:



Association Entity Types

We have:

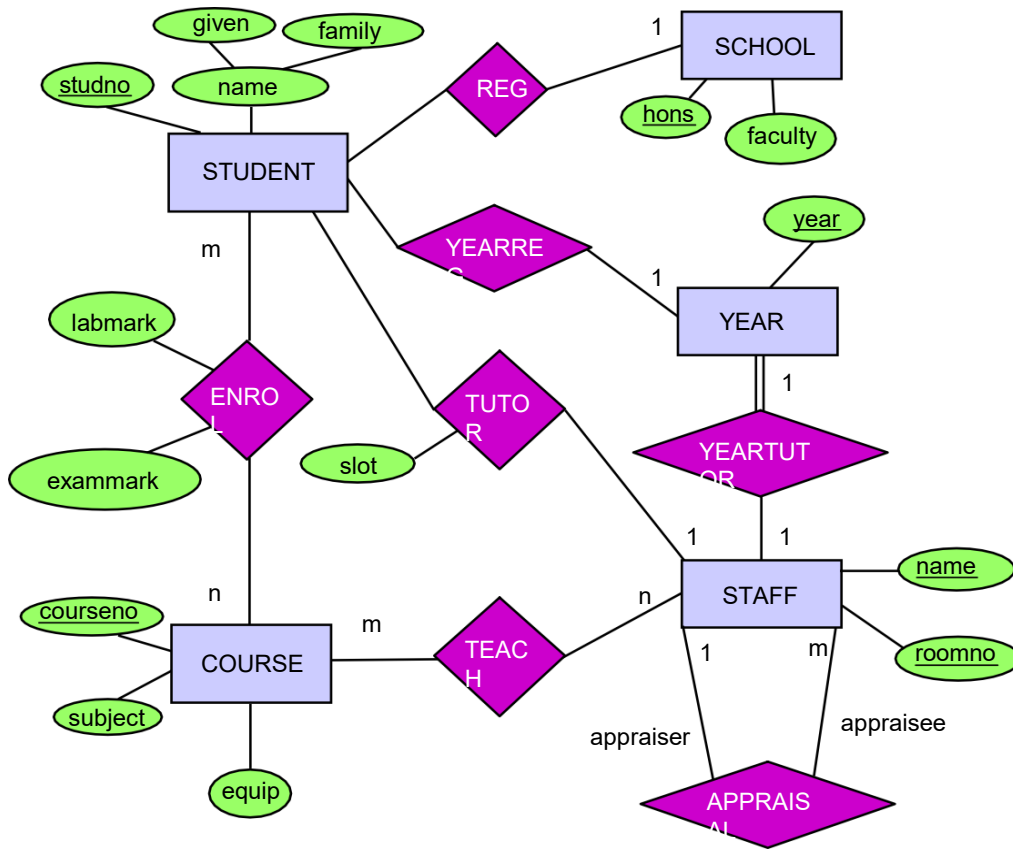
- COURSE(courseno, subject, equip)
- STUDENT(studno, givenname, familyname)



Then:

- ENROL(courseno, studno, labmark, exammark)

Translation of a University Diagram



STUDENT

(studno, givenname,
familyname, hons,
tutor, tutorroom, slot, year)

ENROL(studno, courseno,
labmark, exammark)

COURSE(courseno, subject, equip)

STAFF(lecturer, roomno,
appraiser, approom)

TEACH(courseno, lecturer, lecroom)

YEAR(year, yeartutor, yeartutorroom)

SCHOOL(hons, faculty)

Exercise: Supervision of PhD Students

A database needs to be developed that keeps track of PhD students:

- For each student store the name and matriculation number. Matriculation numbers are unique.
- Each student has exactly one address. An address consists of street, town and post code, and is uniquely identified by this information.
- For each lecturer store the name, staff ID and office number. Staff ID's are unique.
- Each student has exactly one supervisor. A staff member may supervise a number of students.

Exercise Contd...

- The date when supervision began also needs to be stored.
- For each research topic store the title and a short description. Titles are unique.
- Each student can be supervised in only one research topic, though topics that are currently not assigned also need to be stored in the database.
- Tasks:
 - a) Design an entity relationship diagram that covers the requirements above. Do not forget to include cardinality and participation constraints.
 - b) Based on the ER-diagram from above, develop a relational database schema. List tables with their attributes. Identify keys and foreign keys.

References

Books:

- A First Course in Database Systems, by J. Ullman and J. Widom
- Fundamentals of Database Systems, by R. Elmasri and S. Navathe

Slides from Database courses held by the following people:

- Enrico Franconi (Free University of Bozen-Bolzano)
- Carol Goble and Ian Horrocks (University of Manchester)