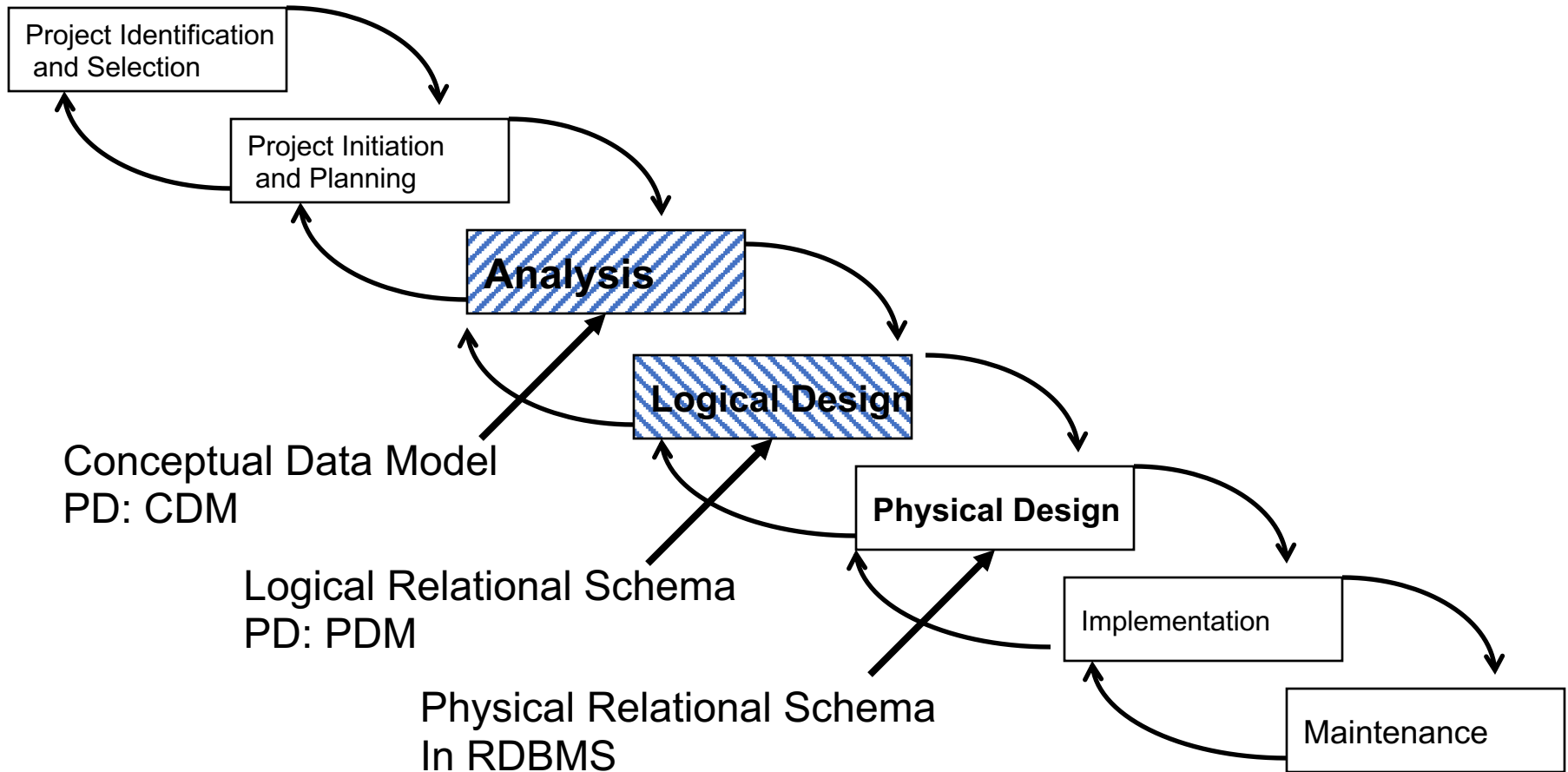


DERIVING A LRS (PDM) FROM AN ERM (CDM)

PART 1

SOFTWARE LIFE CYCLE REVISITED



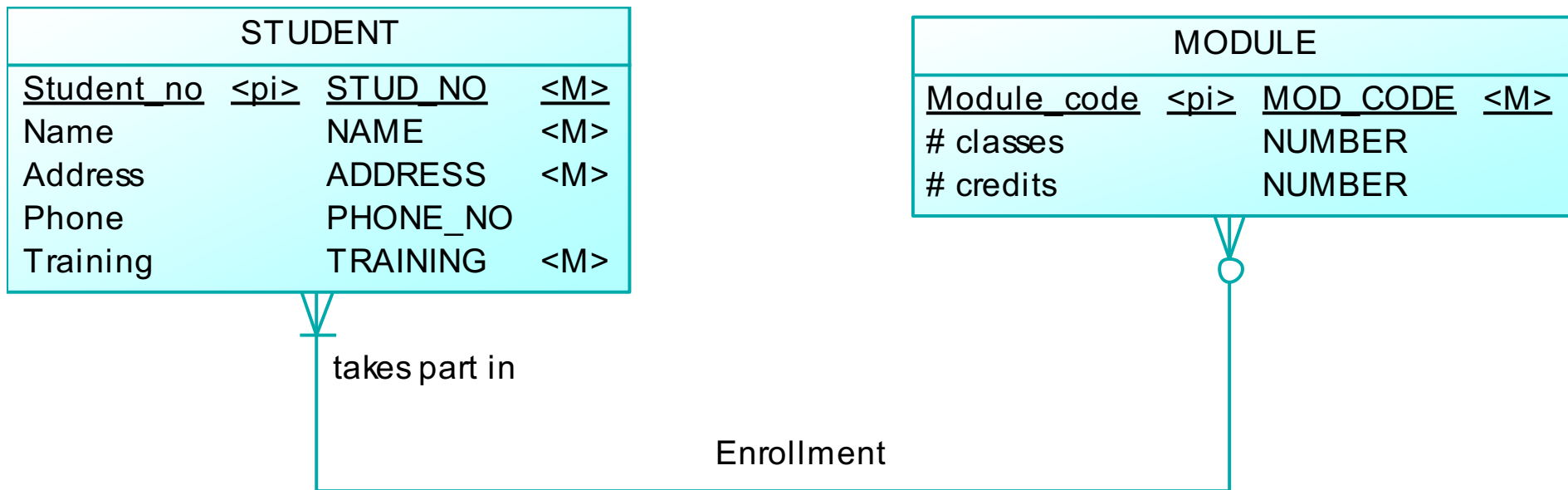
FROM ERM (CDM) TO LRS (PDM)

Six main steps
A-F

- A Replace each n-m relationship type (RT) with a weak entity type (ET)
- B Choose the dominant side in each 1-1 RT
- C ET → Table Att → Column
- D Process each 1-n RT (FK, Ref(s))
- E Process constraints
- F Add predicates and example populations

A REPLACE EACH N-M RT WITH A WEAK ET

Example 1: Students / Modules



A REPLACE EACH N-M RT TYPE WITH A WEAK ET

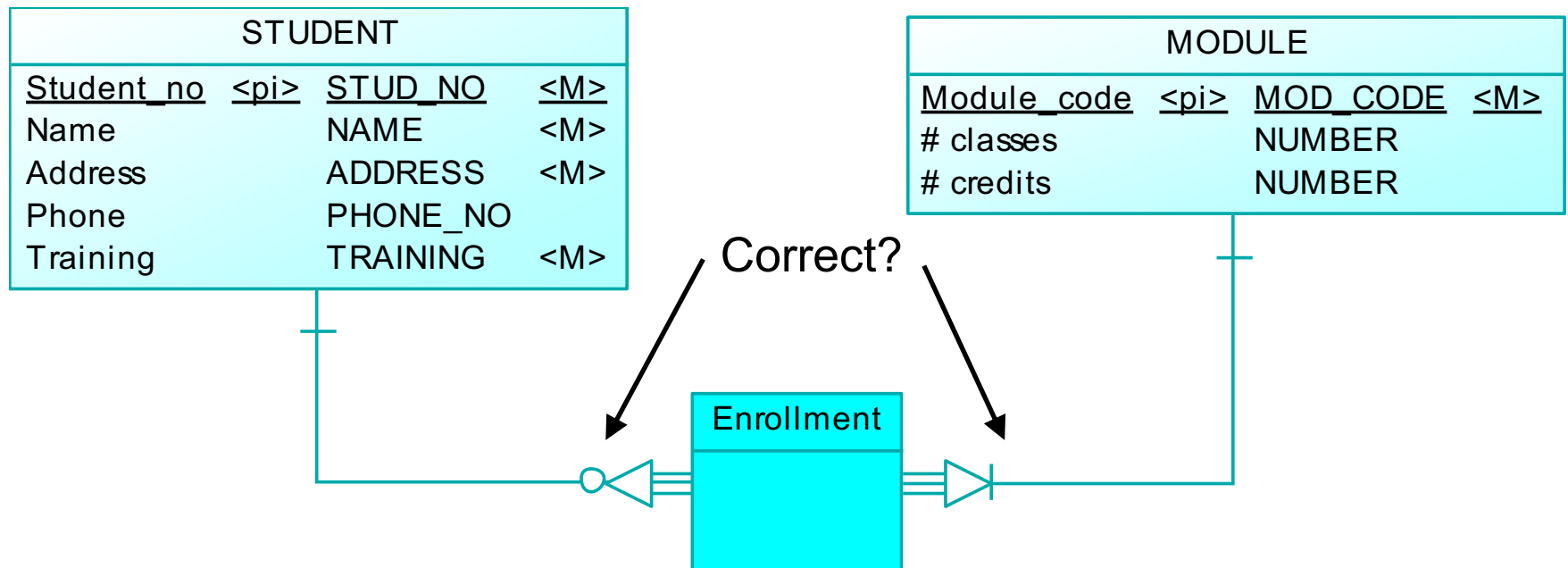
Six substeps in
step A

Numbers
correspond with
those
in Reader
DM-RDS

1. Replace the RT with an ET
2. Add two new 1-n RTs
3. Make the new ET weak,
dependent on the two other ETs
4. Determine the minimum cardinalities near
the new ET
Check your tool!
5. Give names to the new RTs
6. Add at least one role to each new RT

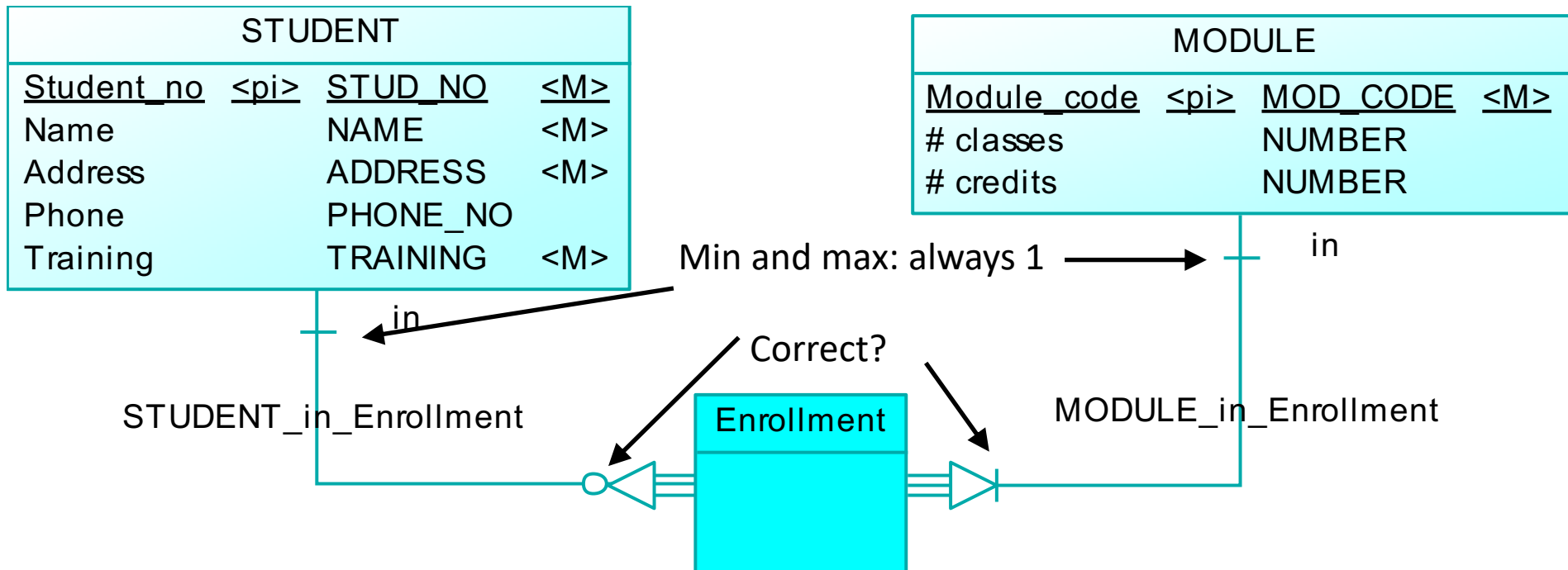
A REPLACE EACH N-M RTTYPE WITH A WEAK ET

1. Replace the RT with an ET
2. Add two new 1-n RTs
3. Make the new ET weak, dependent on the two other Ets



A REPLACE EACH N-M RT TYPE WITH A WEAK ET

4. Determine the min. card. near the new ET. Check your tool!
5. Give names to the new RTs
6. Add at least one role to each new RT



FROM ERM (CDM) TO LRS (PDM)

Six main steps
A-F

Step B,
substep 7, to
be discussed
later (Deriving
PDM from
CDM 2)

- A Replace each n-m relationship type (RT) with a weak entity type (ET)
- B Choose the dominant side in each 1-1 RT
- C ET → Table Att → Column
- D Process each 1-n RT (FK, Ref(s))
- E Process constraints
- F Add predicates and example populations

FROM ERM (CDM) TO LRS (PDM)

Six main steps
A-F

- A Replace each n-m relationship type (RT) with a weak entity type (ET)
- B Choose the dominant side in each 1-1 RT
- C ET → Table Att → Column
- D Process each 1-n RT (FK, Ref(s))
- E Process constraints
- F Add predicates and example populations

C ET → Table Att → Column

8. ET: table with same name. Att: column with same name.
 <pi>: becomes <pk>. <ai>: becomes <ak>.
 <M>: becomes 'not null'. Otherwise: 'null'.
 Domains (and data types): copied.

STUDENT

<u>Student_no</u>	<u>STUD_NO</u>	<u><pk></u>	<u>not null</u>
Name	NAME		not null
Address	ADDRESS		not null
Phone	PHONE_NO		null
Training	TRAINING		not null

MODULE

<u>Module_code</u>	<u>MOD_CODE</u>	<u><pk></u>	<u>not null</u>
# classes	NUMBER		null
# credits	NUMBER		null

Enrollment

FROM ERM (CDM) TO LRS (PDM)

Six main steps
A-F

- A Replace each n-m relationship type (RT) with a weak entity type (ET)
- B Choose the dominant side in each 1-1 RT
- C ET → Table Att → Column
- D Process each 1-n RT (FK, Ref(s))
- E Process constraints
- F Add predicates and example populations

D PROCESS EACH 1-N RT (FK, REF(S))

Five substeps
in step D

9. Each 1-n RT becomes a reference
from the n-side to the 1-side

Numbers
correspond
with those
in Reader
DM-RDS

10. Add FK-columns
with <pk> and <fk> indicators

11. Process inheritance links for subtypes

Step 11 to be
discussed
later

12. Add join-expressions to each
reference

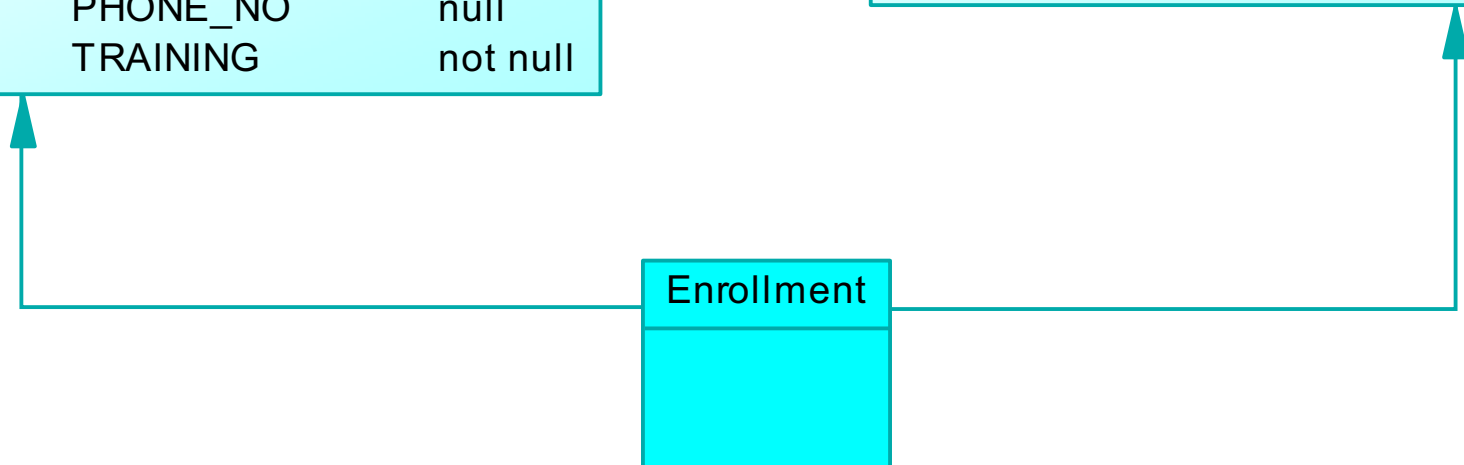
13. Show cardinalities
at the foot of each reference

D PROCESS EACH 1-N RT (FK, REF(S))

9. Each 1-n RT becomes a reference from the n-side to the 1-side

STUDENT			
<u>Student_no</u>	<u>STUD_NO</u>	<u><pk></u>	<u>not null</u>
Name	NAME		not null
Address	ADDRESS		not null
Phone	PHONE_NO		null
Training	TRAINING		not null

MODULE			
<u>Module_code</u>	<u>MOD_CODE</u>	<u><pk></u>	<u>not null</u>
# classes	NUMBER		null
# credits	NUMBER		null



D Process each 1-n RT (FK, Ref(s))

10. Add FK-columns with <pk> and <fk> indicators

STUDENT			
<u>Student_no</u>	<u>STUD_NO</u>	<pk>	not null
Name	NAME		not null
Address	ADDRESS		not null
Phone	PHONE_NO		null
Training	TRAINING		not null

MODULE			
<u>Module_code</u>	<u>MOD_CODE</u>	<pk>	not null
# classes	NUMBER		null
# credits	NUMBER		null

ALWAYS <FK>, AND ALSO <PK> IF THE RT WAS DEPENDENT.

Enrollment			
<u>Student_no</u>	<u>STUD_NO</u>	<pk.fk1>	not null
<u>Module_code</u>	<u>MOD_CODE</u>	<pk.fk2>	not null

D PROCESS EACH 1-N RT (FK, REF(S))

Five substeps in step D

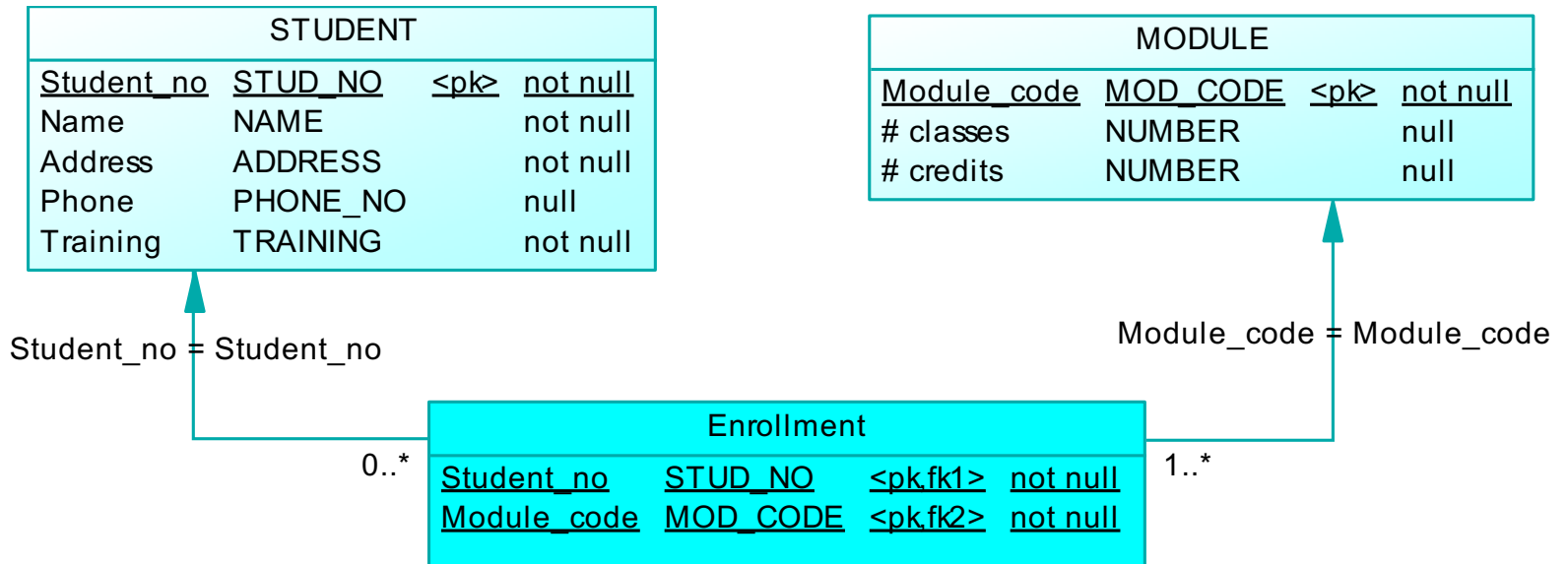
Numbers correspond with those in Reader DM-RDS

Step 11 to be discussed later

9. Each 1-n RT becomes a reference from the n-side to the 1-side
10. Add FK-columns with <pk> and <fk> indicators
11. Process inheritance links for subtypes
12. Add join-expressions to each reference
13. Show cardinalities at the foot of each reference

D Process each 1-n RT (FK, Ref(s))

12. Add join-expressions to each reference
13. Show cardinalities at the foot of each reference



ENROLLMENT: 'ordinary' child of STUDENT, mandatory child of MODULE.
Can **only** be seen from the cardinalities at the foot of the references.

Three refs: ENROLLMENT(Studentno) → STUDENT(Studentno)
 ENROLLMENT(Module code) ← → MODULE(Module code)

FROM ERM (CDM) TO LRS (PDM)

Six main steps
A-F

- A Replace each n-m relationship type (RT) with a weak entity type (ET)
- B Choose the dominant side in each 1-1 RT
- C ET → Table Att → Column
- D Process each 1-n RT (FK, Ref(s))
- E Process constraints
- F Add predicates and example populations

E PROCESS CONSTRAINTS

Most constraints ($\langle pi \rangle$, $\langle ai \rangle$, $\langle M \rangle$, cardinalities) translate easily ($\langle pk \rangle$, $\langle ak \rangle$, not null)

Exceptions:

- Mandatory child references (write procedures)
- Miscellaneous constraints, usually not represented graphically, see reader DM-RDS for some examples

FROM ERM (CDM) TO LRS (PDM)

Six main steps
A-F

- A Replace each n-m relationship type (RT) with a weak entity type (ET)
- B Choose the dominant side in each 1-1 RT
- C ET → Table Att → Column
- D Process each 1-n RT (FK, Ref(s))
- E Process constraints
- F Add predicates and example populations

F Add predicates and example populations

CDM with predicates and example population:

The name of student <Student_ no> is: <Name>.

...
Student <Student_ no> has a <Training> diploma.

1234	5678
John Doe	Lisa Doyle

...	...
HAVO	VWO

Module <Module_ code> consists
of <# classes> classes.

Module <Module_ code> yields
<# credits> credit points.

SQL1	ERMadv
10	5
8	5

STUDENT

<u>Student_no</u>	<pi>	<u>STUD_NO</u>	<M>
Name		NAME	<M>
Address		ADDRESS	<M>
Phone		PHONE_NO	
Training		TRAINING	<M>

MODULE

<u>Module_code</u>	<pi>	<u>MOD_CODE</u>	<M>
# classes		NUMBER	
# credits		NUMBER	

takes part in

Enrollment

Student <Student_ no> takes part
in module <Module_ code>.

1234	5678	5678
SQL1	SQL1	ERMadv

F Add predicates and example populations

The name of student <Student_ no> is: <Name>.

...
Student <Student_ no> has a <Training> diploma.

1234	5678
John Doe	Lisa Doyle

...
HAVO VWO

Module <Module_ code> consists of <# classes> classes.

Module <Module_ code> yields <# credits> credit points.

SQL1	ERMadv
10	5
8	5

STUDENT			
<u>Student_no</u>	<u>STUD_NO</u>	<pk>	not null
Name	NAME		not null
Address	ADDRESS		not null
Phone	PHONE_NO		null
Training	TRAINING		not null

MODULE			
<u>Module_code</u>	<u>MOD_CODE</u>	<pk>	not null
# classes	NUMBER		null
# credits	NUMBER		null

Student_no = Student_no

Module_code = Module_code

0..*

1..*

Enrollment			
<u>Student_no</u>	<u>STUD_NO</u>	<pkfk1>	not null
<u>Module_code</u>	<u>MOD_CODE</u>	<pkfk2>	not null

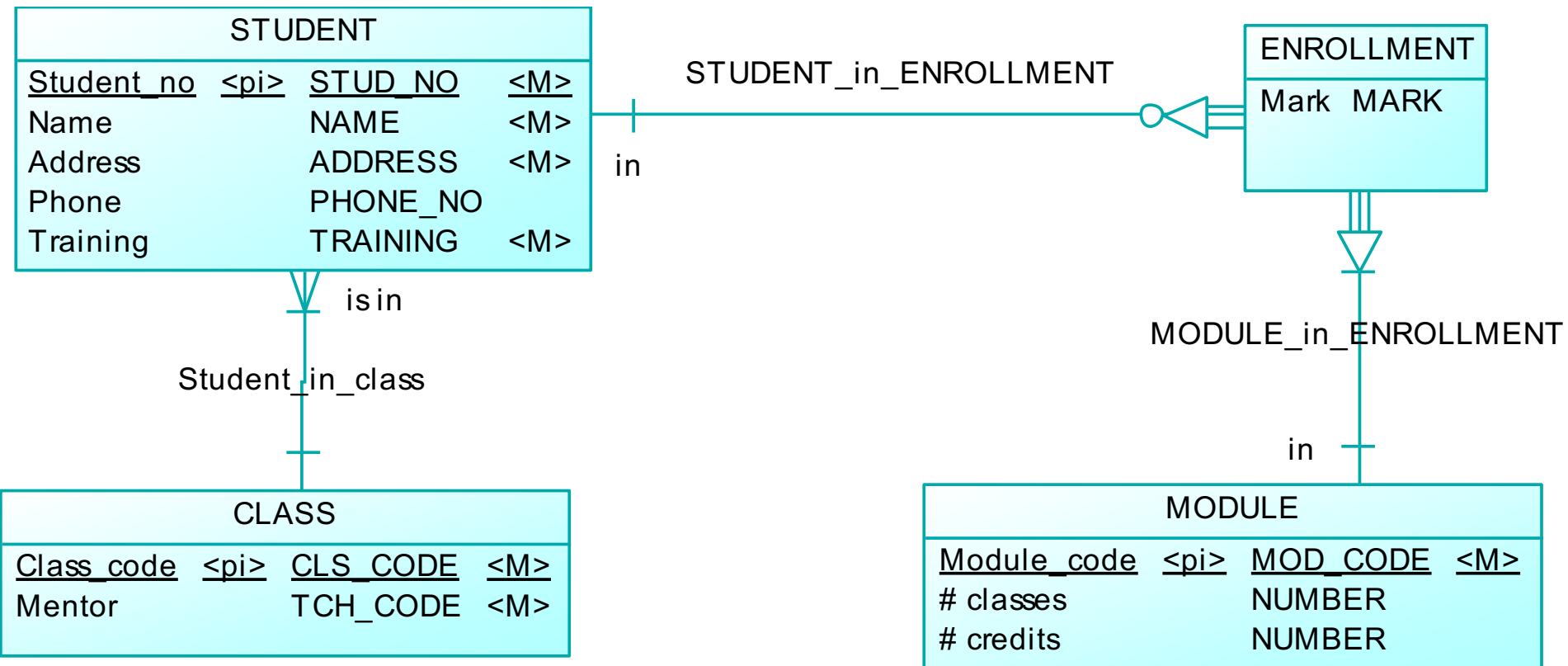
Elementary facts
are independent of
modelling technique

Student <Student_ no> takes part in module <Module_ code>.

1234	5678	5678
SQL1	SQL1	ERMadv

From ERM (CDM) to a LRS (PDM)

Example 2: Students / Modules extended



EXAMPLE 2: STUDENTS / MODULES EXTENDED

Result of steps A, B and C (1-8):

STUDENT			
<u>Student_no</u>	<u>STUD_NO</u>	<u><pk></u>	<u>not null</u>
Name	NAME		not null
Address	ADDRESS		not null
Phone	PHONE_NO		null
Training	TRAINING		not null

ENROLLMENT		
Mark	MARK	null

CLASS			
<u>Class_code</u>	<u>CLS_CODE</u>	<u><pk></u>	<u>not null</u>
Mentor	TCH_CODE		not null

MODULE			
<u>Module_code</u>	<u>MOD_CODE</u>	<u><pk></u>	<u>not null</u>
# classes	NUMBER		null
# credits	NUMBER		null

EXAMPLE 2: STUDENTS / MODULES EXTENDED

Result of step 9 (Each 1-n RT becomes a reference)

STUDENT			
<u>Student_no</u>	<u>STUD_NO</u>	<pk>	not null
Name	NAME		not null
Address	ADDRESS		not null
Phone	PHONE_NO		null
Training	TRAINING		not null

ENROLLMENT		
Mark	MARK	null

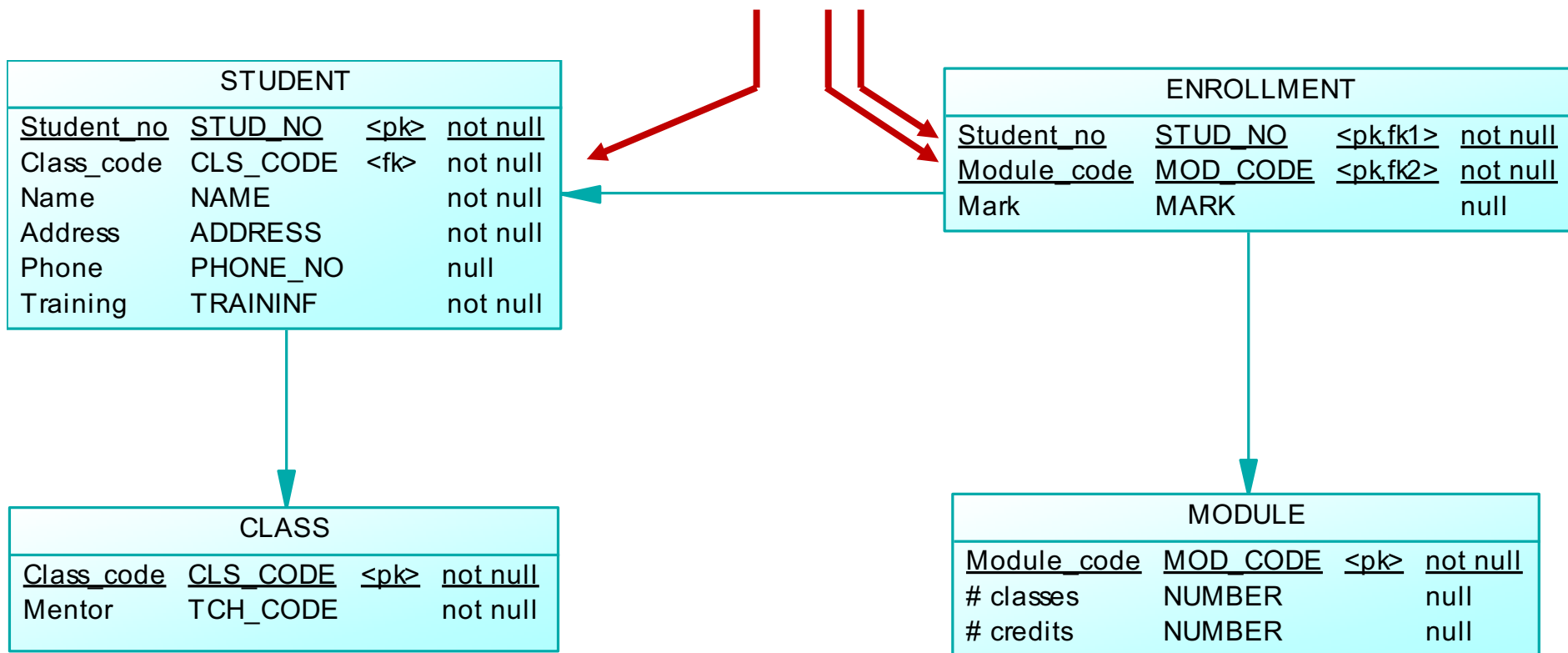


CLASS			
<u>Class_code</u>	<u>CLS_CODE</u>	<pk>	not null
Mentor	TCH_CODE		not null

MODULE			
<u>Module_code</u>	<u>MOD_CODE</u>	<pk>	not null
# classes	NUMBER		null
# credits	NUMBER		null

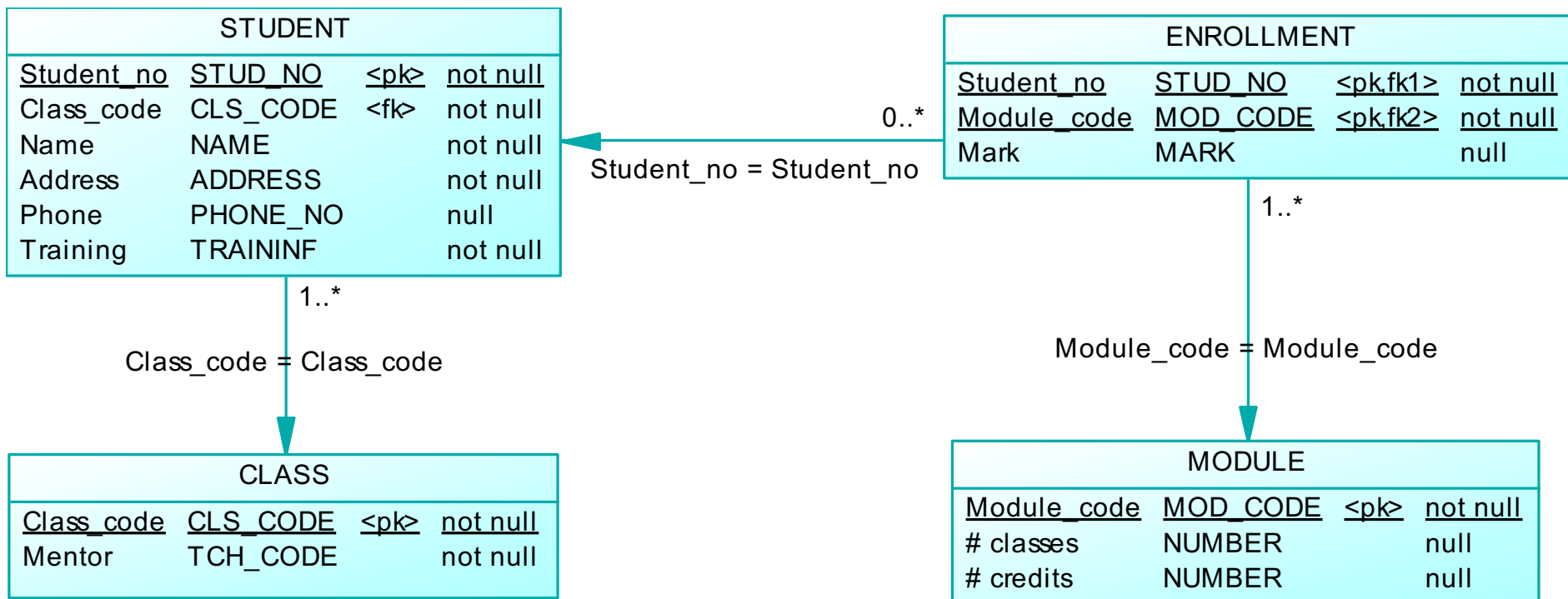
EXAMPLE 2: STUDENTS / MODULES EXTENDED

Result of step 10 (add FK-columns with <pk> and <fk>):



EXAMPLE 2: STUDENTS / MODULES EXTENDED

Result of steps 12-13 (Add cardinalities and join conditions):



Note: 5 references in total, 3 FK refs, 2 mand. child refs