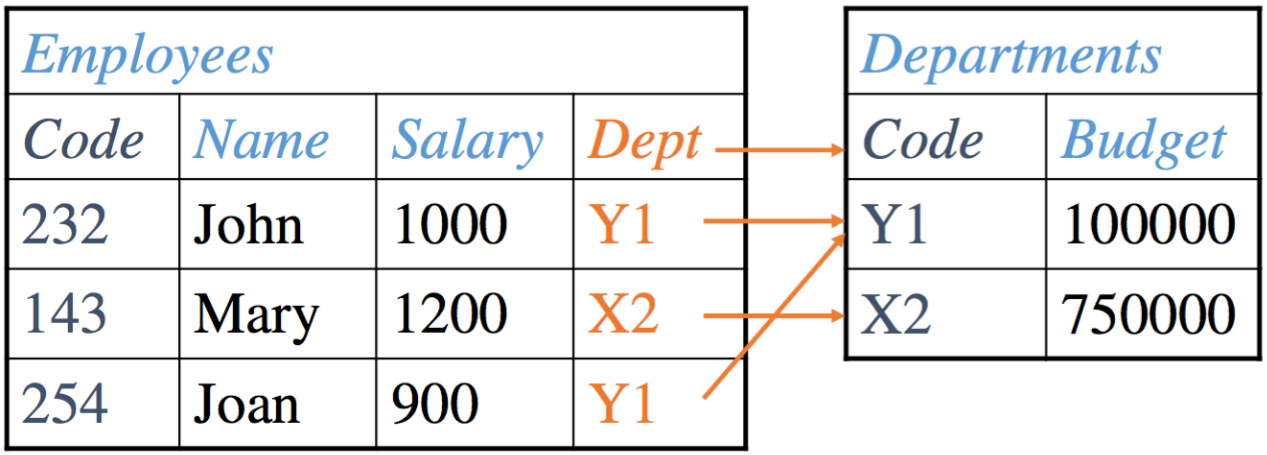
Logical Model

Anna Monreale

Relational Data Model

- Starting from the output of the Analysis of user requirements: **the Class Diagram**
- Generate the Logical Model by following a systematic process
  - organizing all data in flat **tables** of **rows** with a primary key
  - transforming associations as **foreign keys**



Employees

Code	*Name*	*Salary*	*Dept*
232	John	1000	Y1
143	Mary	1200	X2
254	Joan	900	Y1

Departments

Code	*Budget*
Y1	100000
X2	750000

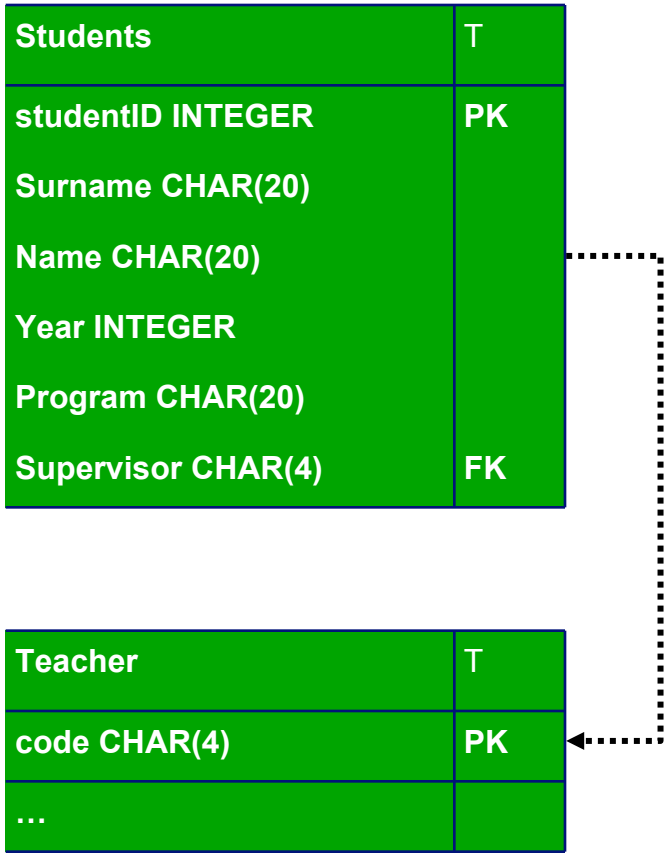
Algorithm for the Logical Design

- **Step I**: Translate all classes which are not involved in hierarchy
- **Step II**: Translate all hierarchies
- **Step III**: Translate multivalued attributes in to tables
- **Step IV** : Translate N-N relationships
- **Step V** : Translate 1-N relationships
- **Step VI** : Translate 1-1 relationships
- **Step VII** : Add other possible constraints

Step I: Translate all classes

- Classes become **tables** containing
  - Attributes
  - primary key
  - foreign keys

- **Primary key**
  - Simple to be used and compact
  - We can use set of attributes if compact
  - We can use a synthetic identifier



Step I: Translate all classes

Courses
code
title
program

Courses	T
Code CHAR(3)	PK
Title CHAR(20)	
Program CHAR(20)	

Identifier

Exams
mark
laud
date

Exams	T
Code CHAR(5)	PK
Mark INTEGER	
Laud BOOL	
Date DATE	

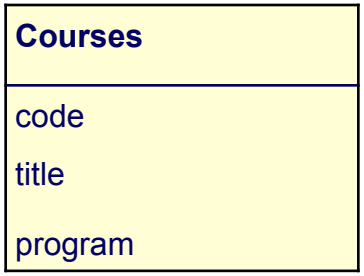
Synthetic identifier

Stages
Location
Date-start
Duration

Stages	T
studentID INTEGER	PK, FK
Location CHAR(20)	
Date-start DATE	
Duration INTEGER	

External Identifier

Step I: Translate all classes



Courses	T
Code CHAR(3)	PK
Title CHAR(20)	
Program CHAR(20)	

Code	Title	Program
PR1	Programming	Bachelor
ASD	Algorithms	Bachelor
DB1	Databases	Master

Step I: Translate all classes

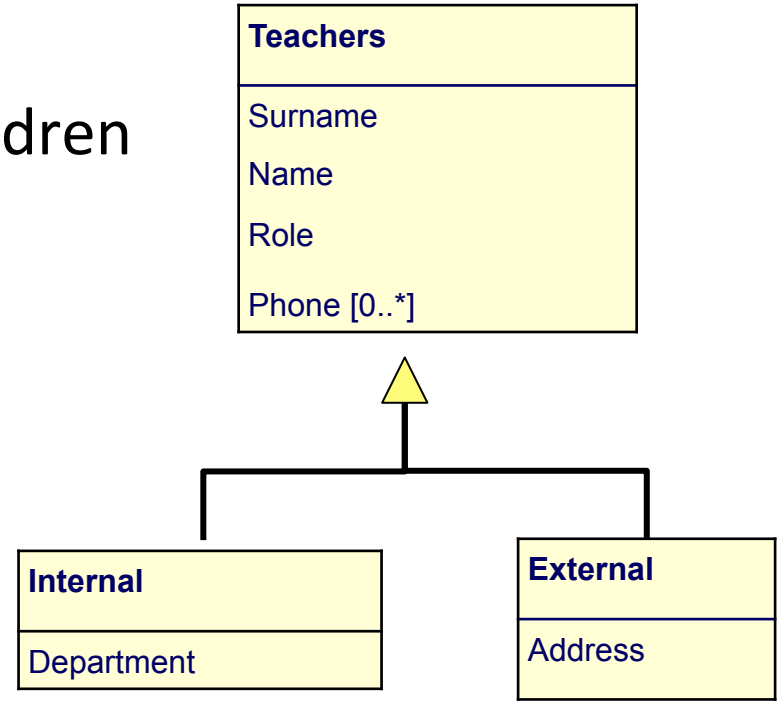
Stages
location Date-start duration

Stages	T
studentID INTEGER **Location CHAR(20)** **Date-start DATE** **Duration INTEGER**	**PK, FK**

StudentID	Location	Date-start	Duration
444	**Microsoft**	**2002-05-15**	**3**
77777	**Microsoft**	**2002-05-15**	**3**
88888	**Basica**	**2002-09-01**	**3**

Step II: Translate all hierarchies

- Relational Data Model does not have Hierarchies
- Three possible cases
  - Translate only the parent class
  - Translate only children
  - Translate both parent and children



Step II: Translate all hierarchies

- **Translate only the parent class**
  - One table with the name of the parent class
  - The table contains the attributes of both parent and children
  - Add an attribute **TYPE** to distringuish the instance
  - Generation of **NULL values**
  - It is convenient if the children **are not used so much** in the application

Teachers	T
Code CHAR(4)	PK
Surname CHAR(20)	
Name CHAR(20)	
Department CHAR(10)	
Role CHAR(15)	
Address CHAR(20)	
Type CHAR(10)	

Step II: Translate all hierarchies

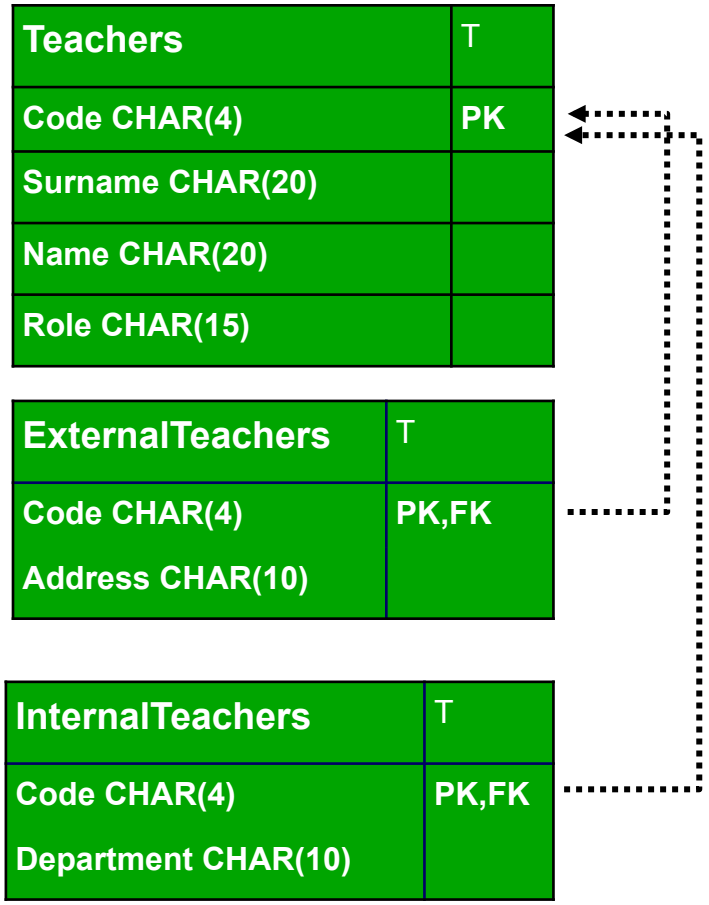
- **Translate only children classes**
  - One table for each child
  - Each child table inherits both attributes and relationships of the parent
  - Possible only when the the **hierarchy is complete**
  - It is convenient if the application **uses often the children**

InternalTeachers	T
Code CHAR(4)	PK
Surname CHAR(20)	
Name CHAR(20)	
Department CHAR(10)	
Role CHAR(15)	

ExternalTeachers	T
Code CHAR(4)	PK
Surname CHAR(20)	
Name CHAR(20)	
Address CHAR(10)	
Role CHAR(15)	

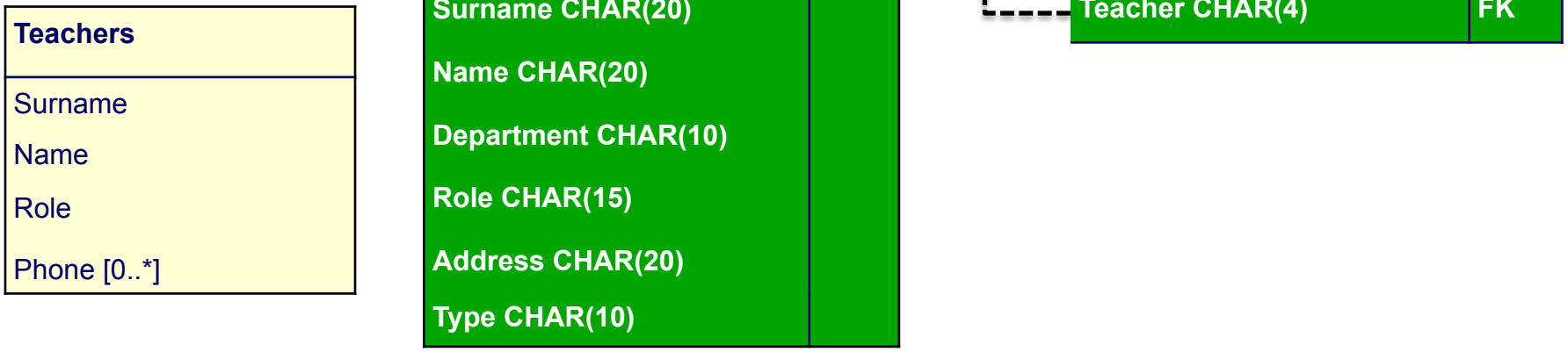
Step II: Translate all hierarchies

- **Translate both parent and children classes**
  - One table for each child and for the parent
  - Each class has its attributes
  - It is necessary if the application **uses often both children and parent**

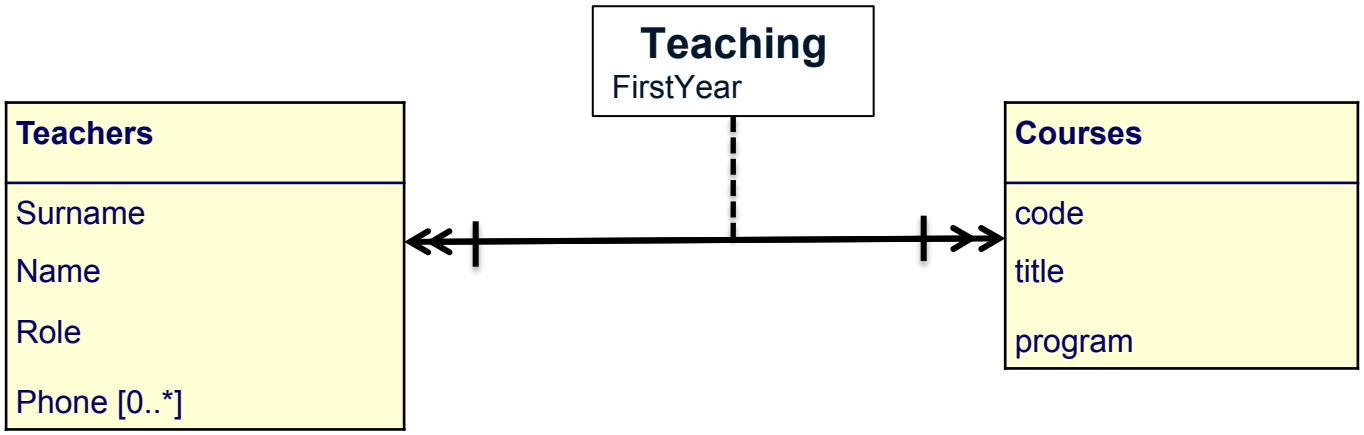


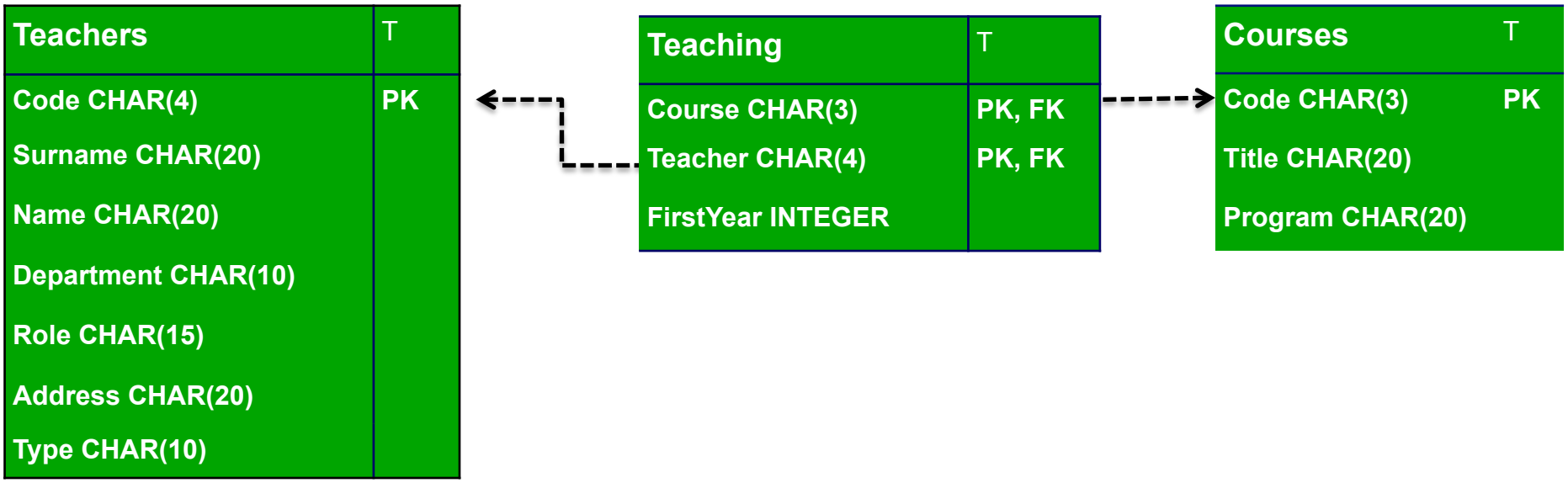
Step III: Translate multivalued attributes

- Each multivalued attribute generates a new table
- The new table containing a foreign key vs the original class



Step IV : Translate N-N relationships





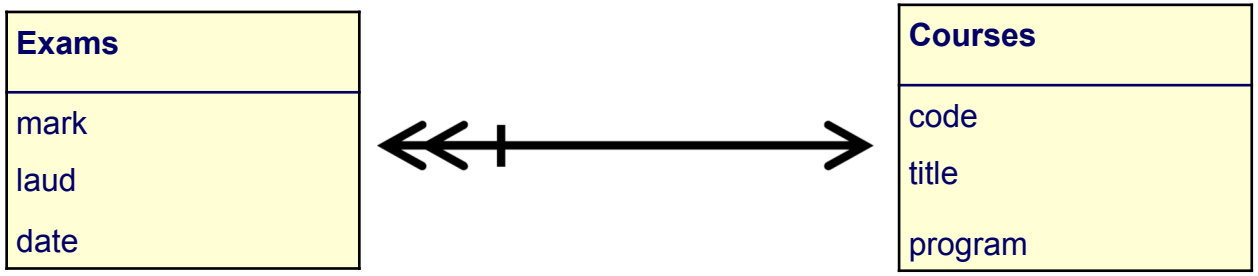
Step IV : Translate N-N relationships

codice	cognome	nome	…
FT	Pedreschi	Dino	…
CV	Monreale	Anna	…
ADP	Giannotti	Fosca	…

docente	corso	primoAnnoTit
FT	PR1	2001
CV	ASD	2002
FT	ASD	1999
…	…	

Code	Title	Program
PR1	Programming	Bachelor
ASD	Algorithms	Bachelor
DB1	Databases	Master

Step V : Translate 1-N relationships



Exams	T
Code CHAR(5)	PK
Course CHAR(3)	FK
Mark INTEGER	
Laud BOOL	
Date DATE	

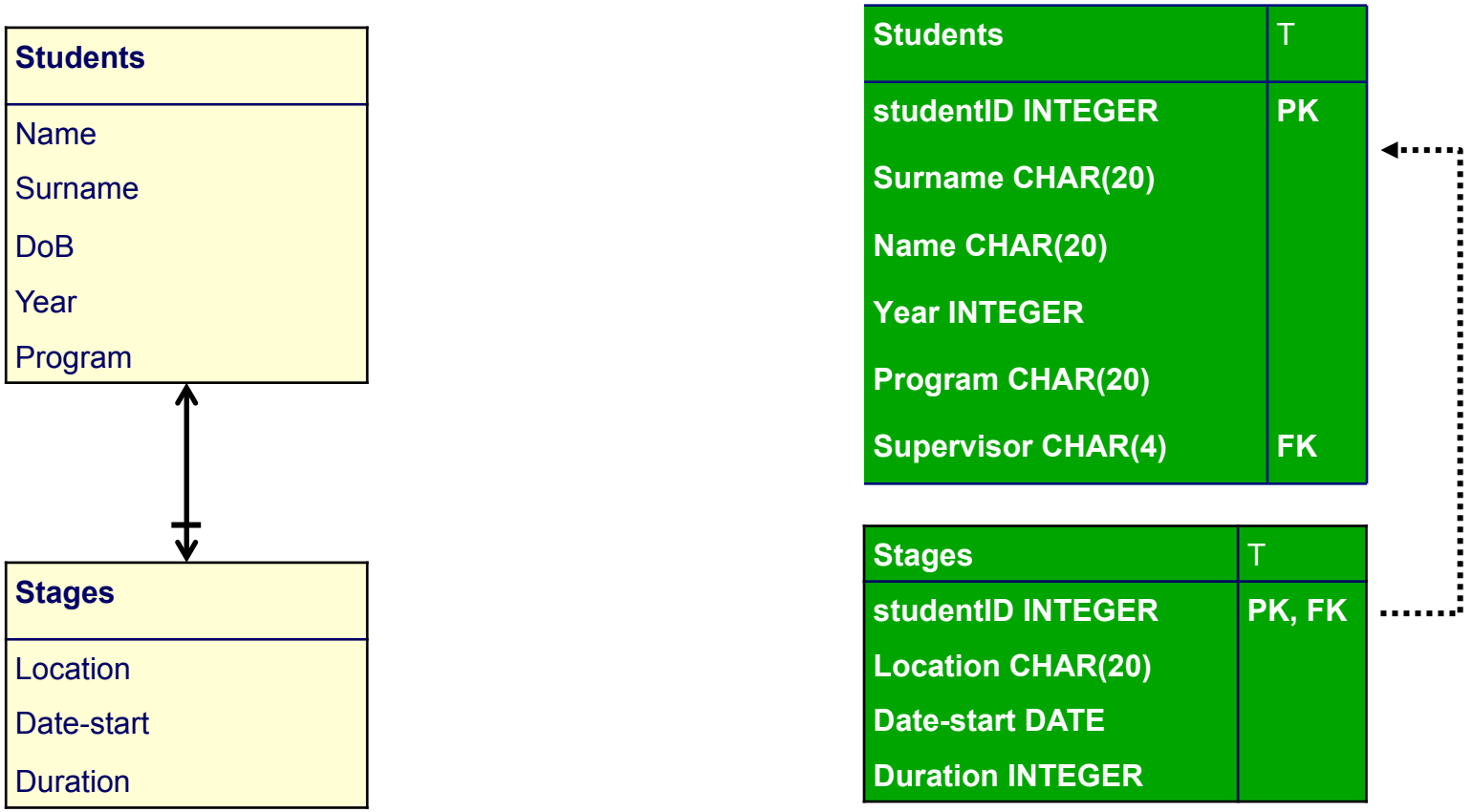
Courses	T
Code CHAR(3)	PK
Title CHAR(20)	
Program CHAR(20)	

Code	Title	Program
PR1	Programming	Bachelor
ASD	Algorithms	Bachelor
DB1	Databases	Master

Code	Course	Mark	Laud	Date
pr101	PR1	27	false	2002-06-12
asd01	ASD	30	true	2001-12-03
BD101	INFT	24	false	2001-09-30
pr102	PR1	21	false	2002-06-12
asd02	ASD	20	false	2001-12-03
asd03	ASD	28	false	2002-06-13

Step VI : Translate 1-1 relationships

- Similar to 1-N relationship but we can choose where putting the foreign key
- It is better in the class where we have total relationship



Step VII: Add other possible constraints

- Now we have:
  - Tables
  - Attributes
  - Primary keys
  - Foreign keys
- We need to add other constraints
  - NOT NULL
  - DEFAULT
  - CASCADE
  - CHECK
  - …..