Introduction to Relational Databases

- Bachelor CS, Lille 1 University
- Oct. 5th, 2011 (lecture 6/12)
- Today's lecturer: A. Bonifati
- Topic: Introduction to Database Modeling (aka Logical Modeling, transformation from Conceptual Modeling to Logical Modeling)

Prologue: Logical Model

- Today's lecture: where are we?
  - we have now the conceptual model; we need to enforce a transformation from the conceptual model to a logical model
- Goals of the logical model
  - allows to get closer to the implementation of the database
  - starting from a more abstract (conceptual) model that was conceived before

Summary

- Introduction
  - The design process of a database
- An Algorithm for Logical Modeling
  - Translation of the classes
  - Translation of m-m associations
  - Translation of 1-m associations
  - Translation of 1-1 associations
- The resulting database

Introduction

- We are designing a database
  - we have already completed (at least one iteration of) the analysis phase
- Activities to carry out
  - define the architecture of the application
  - define the structure and the attributes of the classes
  - define the structure of the database
- Next phase: development

The design process of a database

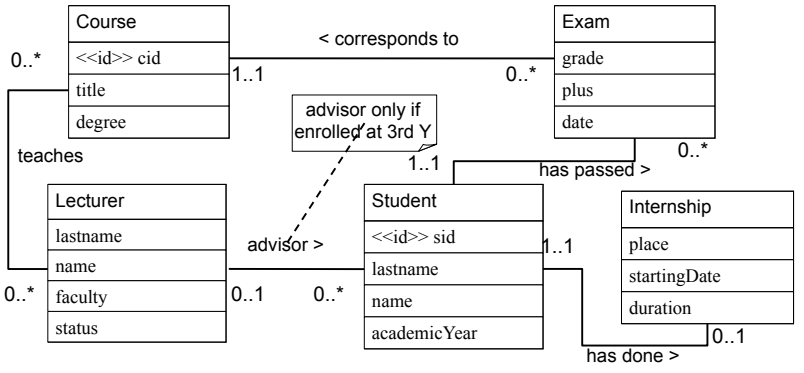
- Starting point
  - the conceptual modeling for the data
- Logical design
  - derive the logical schema from the conceptual model and, eventually, external schemas (views on the database)
- Physical design
  - verify and, possibly, optimize the logical schema

The design process of a database

- Logical design
  - it relies on a standard algorithm
- Physical design
  - mixed mode activity: design and "tuning"
  - too difficult to standardize
- In this lecture
  - we focus on the logical design

Translation Algorithm

- For now
  - we will study a simplified version
- Steps of the algorithm
  - initial translation of the classes
  - translation of m-m associations
  - translation of 1-m associations
  - translation of 1-1 associations



Conceptual Schema

Translation Algorithm

- Graphical notations for the relations
  - we still use a suitable stereotype of UML
  - that lets encode a table
  - its attributes
  - the primary keys
  - the possible foreign keys

Translation Algorithm

- Example:

```
CREATE TABLE Student (
   sid integer PRIMARY KEY,
   lastname char(20),
   name char(20),
   year integer,
   degree char(20),
   advisor char(4) REFERENCES Lecturer(pid));
```

Student	T
sid INTEGER	PK
lastname CHAR(20)	
name CHAR(20)	
year INTEGER	
degree CHAR(20)	
advisor CHAR(4)	FK

Lecturer	T
pid CHAR(4)	PK
…	

Translation of classes

- Idea
  - each class is converted into a table
  - initially, the same set of attributes
  - other attributes can be added afterwards
- It is mandatory
  - identify the type of attributes
  - identify the primary key
  - identify possible foreign keys

Translation of classes

- Primary key
  - it has to be easy to use and concise
  - it typically corresponds to an explicit identifier (e.g.: sid for Student, cid for Course)
  - otherwise, it is a synthetic identifier

Translation of classes

Course
<<id>> cid
title
degree

Exam
grade
plus
date

Course	T
cid CHAR(3)	PK
title CHAR(20)	
degree CHAR(20)	

explicit identifier

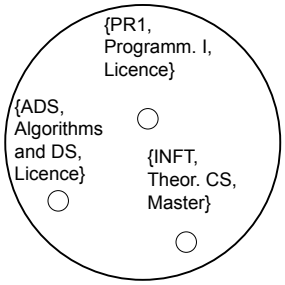
Exam	T
eid CHAR(5)	PK
grade INTEGER	
plus BOOL	
date DATE	

synthetic identifier (same for internship)

Translation of classes

Course
<<id>> cid
title
degree

Course	T
cid CHAR(3)	PK
title CHAR(20)	
degree CHAR(20)	



cid	title	degree	…
PR1	Programming I	Licence	…
ASD	Algorithms and DS	Licence	…
INFT	Theoretical CS	Master	…

Translation of m-m associations

- Each m-m association is converted into
  - a new table
  - with foreign keys to the classes connected by the association
  - with the attributes of the association, if any
  - the primary key of the association is composed of both foreign keys of the connecting classes (each instance of the association is identified by a pair of elements)

Translation of m-m associations

Course
<<id>> cid
title
degree

0..* teaches 0..*

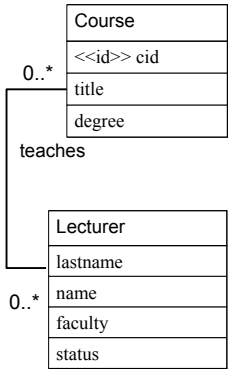
Lecturer
lastname
name
faculty
status

Course	T
cid CHAR(3)	PK
title CHAR(20)	
degree CHAR(20)	

Teaching	T
cid CHAR(3)	PK, FK
pid CHAR(4)	PK, FK

Lecturer	T
pid CHAR(4)	PK
lastname CHAR(20)	
name CHAR(20)	
…	…

Translation of the m-m associations



cid	title	degree
PR1	Programming I	Licence
ASD	Algorithms and DS	Licence
INFT	Theoretical CS	Master

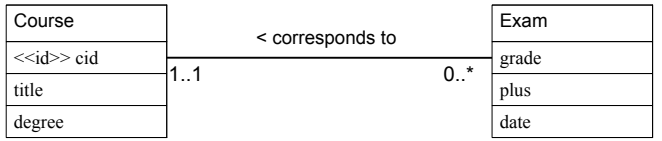
lecturer	course
FT	PR1
CV	ASD
FT	ASD
…	…

pid	lastname	name	…
FT	Totti	Francesco	…
CV	Vieri	Christian	…
ADP	Del Piero	Alessandro	…

Translation of the 1-m associations

- In principle
  - could introduce new tables (as for m-m case)
  - it would be inefficient: we would need to join the obtained tables
- They generate foreign keys
  - each instance of the association is identified by the object on the side of the association with cardinality 1
  - it enforces a foreign key corresponding to the side of the association with cardinality 1 into the table with cardinality m

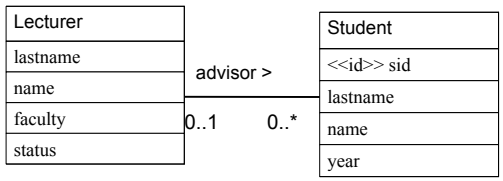
Translation of the 1-m associations



Course	T
cid CHAR(3)	PK
title CHAR(20)	
degree CHAR(20)	

Exam	T
eid CHAR(5)	PK
grade INTEGER	
plus BOOL	
date DATE	
course CHAR(3)	FK

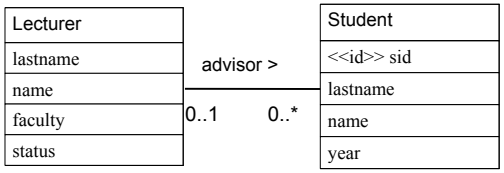
Translation of the 1-m associations



Lecturer	T
pid CHAR(4)	PK
lastname CHAR(20)	
name CHAR(20)	
faculty CHAR(10)	
status CHAR(15)	
type CHAR(10)	

Student	T
sid INTEGER	PK
lastname CHAR(20)	
name CHAR(20)	
year INTEGER	
degree CHAR(15)	
advisor CHAR(4)	FK

Translation of the 1-m associations



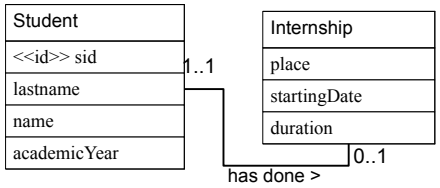
pid	lastname	name	…
FT	Totti	Francesco	…
CV	Vieri	Christian	…
ADP	Del Piero	Alessandro	…

sid	lastname	name	…	advisor
111	Rossi	Mario	…	null
222	Neri	Paolo	…	null
333	Rossi	Maria	…	null
444	Pinco	Palla	…	FT
77777	Bruno	Pasquale	…	FT
88888	Pinco	Pietro	…	CV

Translation of the 1-1associations

- Similar discussion as previously
  - I can choose where to put the foreign key
  - it is preferable to use as foreign key the primary key of the class which is on the side of the association with minimum cardinality equal to 1
- Example
  - Student has done an Internship

Translation of the 1-1associations



Student	T
sid INTEGER	PK
…	…

Intership	T
id INTEGER	PK
sid INTEGER	FK
place CHAR(20)	
startingDate DATE	
duration INTEGER	

Teaching	T
course CHAR(3)	PK, FK
lecturer CHAR(4)	PK, FK

Course	T
cid CHAR(3)	PK
title CHAR(20)	
degree CHAR(20)	

Exam	T
eid CHAR(5)	PK
grade INTEGER	
plus BOOL	
date DATE	
course CHAR(3)	FK
stud INTEGER	FK

Intership	T
id INTEGER	PK
sid INTEGER	FK
place CHAR(20)	
startingDate DATE	
duration INTEGER	

Lecturer	T
pid CHAR(4)	PK
lastname CHAR(20)	
name CHAR(20)	
faculty CHAR(10)	
status CHAR(15)	
type CHAR(10)	

Student	T
sid INTEGER	PK
lastname CHAR(20)	
name CHAR(20)	
year INTEGER	
degree CHAR(15)	
advisor CHAR(4)	FK

The final database schema in SQL

```
CREATE TABLE Lecturer (
   pid char(4) PRIMARY KEY,
   lastname varchar(20) NOT NULL,
   name varchar(20) NOT NULL,
   status char(15),
   faculty char(10),
   type char(10) NOT NULL
);

CREATE TABLE Student (
   sid integer PRIMARY KEY,
   lastname varchar(20) NOT NULL,
   name varchar(20) NOT NULL,
   degree char(20),
   year integer,
   advisor char(4) REFERENCES Lecturer(pid)
);
```


The final database schema in SQL

```
CREATE TABLE Course (
   cid char(3) PRIMARY KEY,
   title varchar(20) NOT NULL,
   degree char(20)
);

CREATE TABLE Exam (
    eid char(5) PRIMARY KEY,
    student integer NOT NULL REFERENCES Student(sid)
      ON DELETE cascade ON UPDATE cascade,
    course char(3) NOT NULL REFERENCES Course(cid),
    grade integer,
    plus bool,
    date date,
    UNIQUE (studente, corso)
);
```


The final database schema in SQL

```
CREATE TABLE Internship (
   id INTEGER PRIMARY KEY,
   student integer REFERENCES Student(sid),
   place char(20) NOT NULL,
   startingDate date,
   duration integer
);


CREATE TABLE Teaching (
   lecturer char(4) REFERENCES Lecturer(pid),
   course char(3) REFERENCES Course(cid),
   PRIMARY KEY (lecturer, course)
);
```


A Possible Instance

Lecturer

pid	lastname	name	status	faculty	type
FT	Totti	Francesco	full	Engineering	staff
CV	Vieri	Christian	associate	Sciences	staff
ADP	Del Piero	Alessandro	null	null	external

Student

sid	lastname	name	degree	year	advisor
111	Rossi	Mario	Licence	1	null
222	Neri	Paolo	Licence	2	null
333	Rossi	Maria	Licence	1	null
444	Pinco	Palla	Licence	3	FT
77777	Bruno	Pasquale	Master	1	FT
88888	Pinco	Pietro	Master	1	CV

Course

cid	title	degree
PR1	Programmazione I	Licence
ASD	Algoritmi e Str. Dati	Licence
INFT	Informatica Teorica	Master

Exam

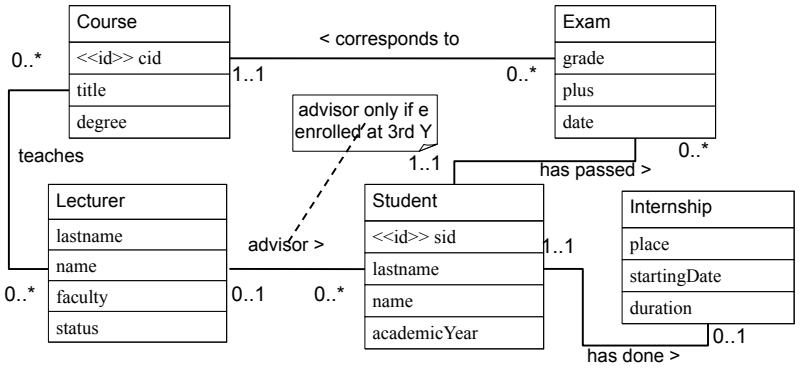
eid	student	course	grade	plus	date
pr101	111	PR1	C	false	2002-06-12
asd01	222	ASD	A	true	2001-12-03
inft1	111	INFT	C	false	2001-09-30
pr102	77777	PR1	D	false	2002-06-12
asd02	77777	ASD	C	false	2001-12-03
asd03	88888	ASD	B	false	2002-06-13
pr103	88888	PR1	A	false	2002-07-01
inft2	88888	INFT	A	true	2001-09-30

Internship

id	student	place	startingDate	duration
1	444	Microsoft	2002-05-15	3
2	77777	Microsoft	2002-05-15	3
3	88888	SOGEI	2002-09-01	3

Teaching

lecturer	course
FT	PR1
CV	ASD
ADP	INFT
ADP	PR1
FT	ASD



Conceptual Schema

The complete Algorithm

- Up to now
  - we may want to add more details
- In particular
  - translation of multi-valued attributes
  - translation of the minimum cardinalities

The Complete Algorithm

- I step
  - initial translation of classes
- II step
  - translation of hierarchies (omitted in our course)
- III step
  - translation of multi-valued attributes
- IV step
  - translation of m-m associations

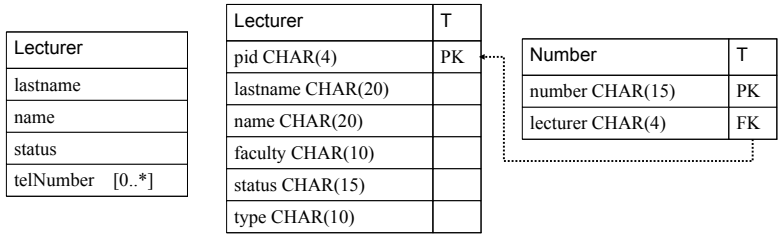
The Complete Algorithm

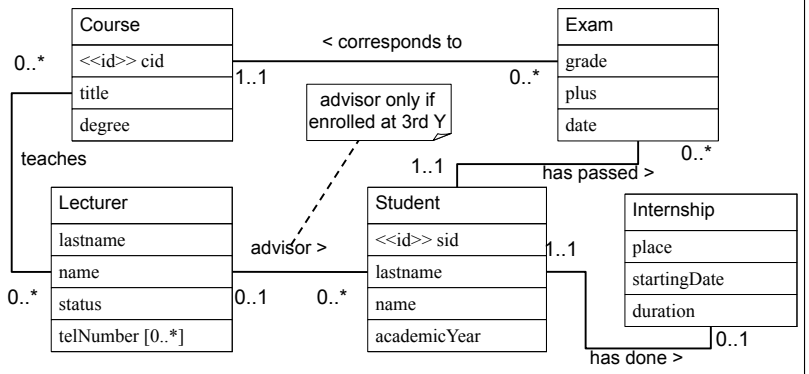
- V step
  - translation of 1-m associations
- VI step
  - translation of 1-1 associations
- VII step
  - introduction of further constraints (omitted)
- VIII step
  - design of external schemas (omitted)

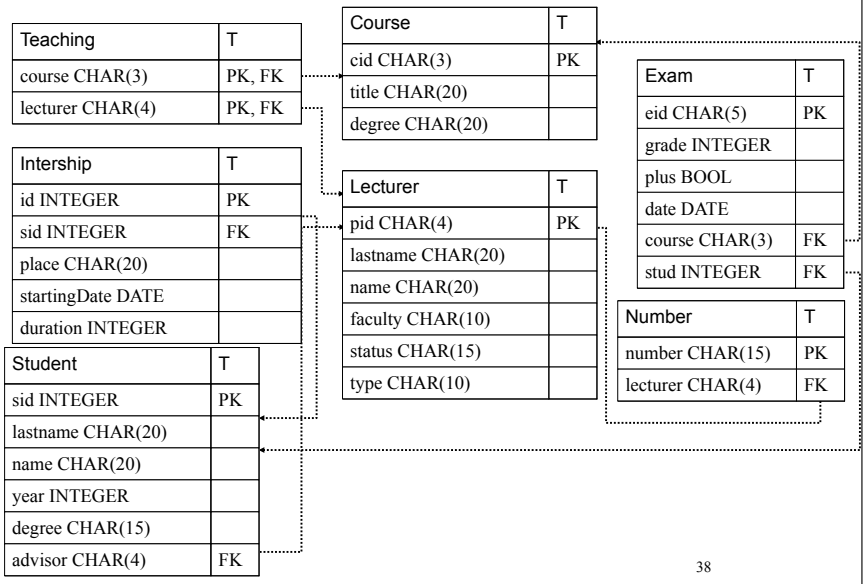
III Step: Multi-valued attributes

- Multi-valued attributes
  - do not have ‘atomic’ values
  - require extra tables
- A table for a multi-valued attribute
  - has an attribute that represents the values
  - a foreign key to the class to which the attribute belongs to

III Step: Multi-valued Attributes







Summary of today's lecture

- Introduction
  - The design process of a database
- An algorithm for the design of the logical model
  - Translation of classes
  - Translation of multi-valued attributes
  - Translation of m-m associations
  - Translation of 1-m associations
  - Translation of 1-1 associations

- The resulting database