LISBON
SCHOOL OF
ECONOMICS &
MANAGEMENT
UNIVERSIDADE DE LISBOA

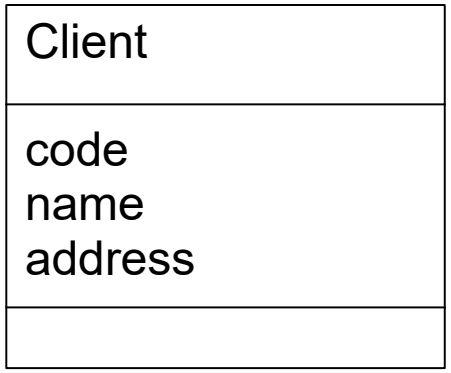
Carlos J. Costa

MAPPING A UML MODEL TO A RELATIONAL SCHEMA

A series of simple rules can be followed to map an UML Diagram into a relational schema.

Class

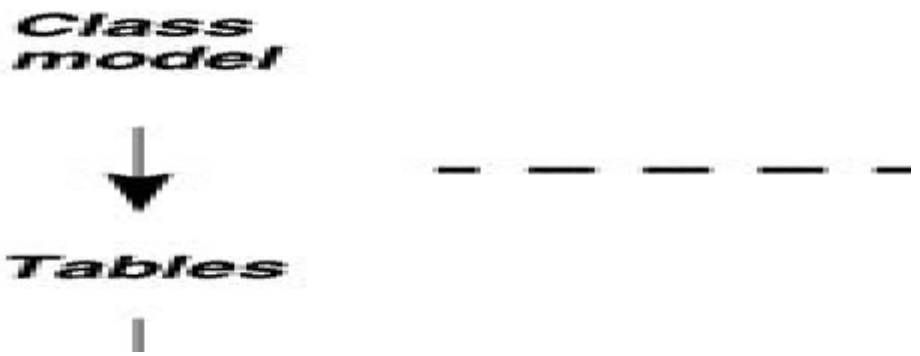
- Every UML Class becomes a Table
- Every UML Attribute in Class becomes a Column in a Table



Client(<u>code</u>, name, address)

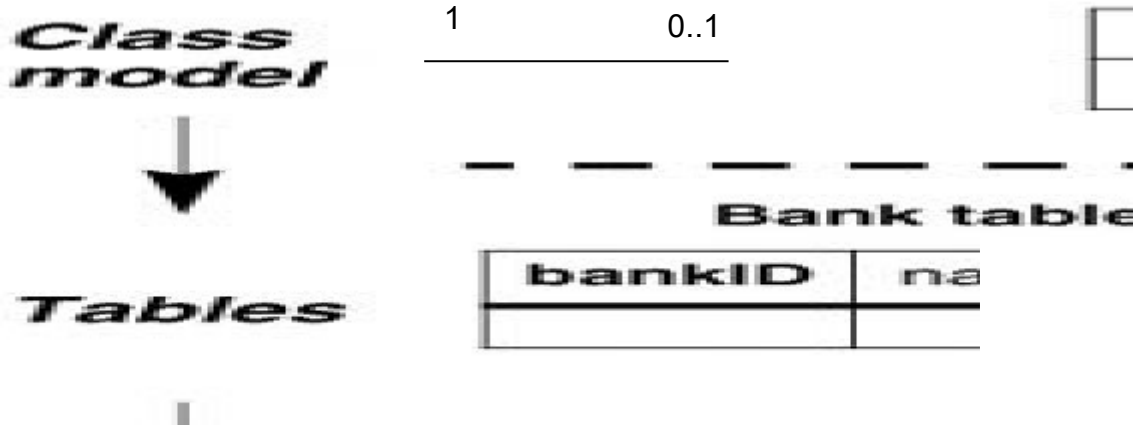
Primary key must be chosen among attributes that have the appropriate characteristics

Class



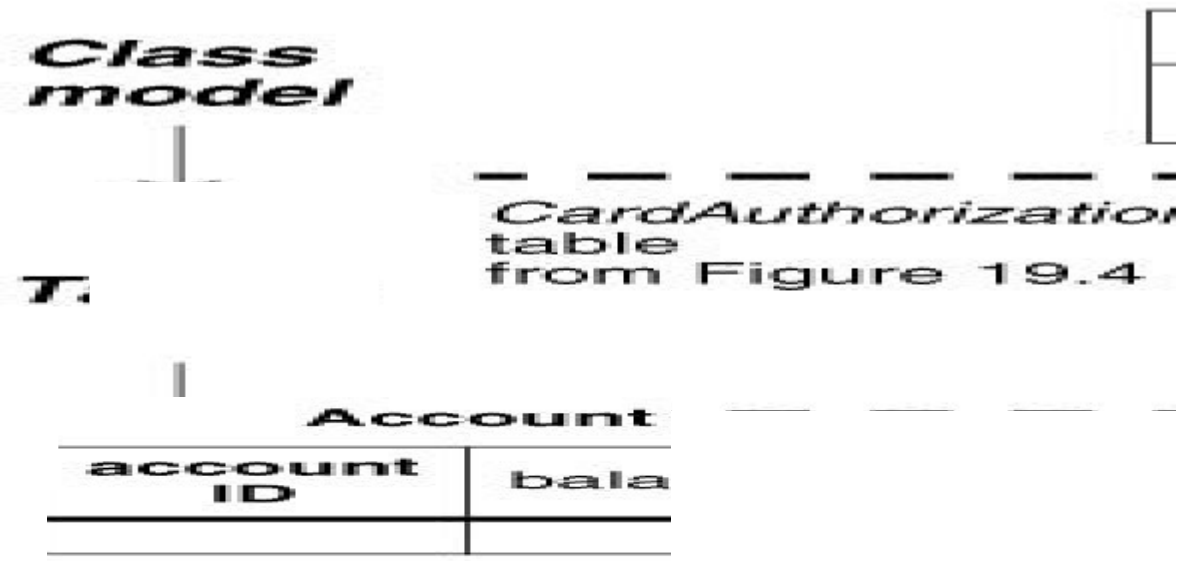
Customer(<u>customerID</u>,name,tempAmount)

One to one



- Bank(<u>bankID</u>, name)
- Account(<u>accountID</u>, balance, creditLimit, *bankID*)

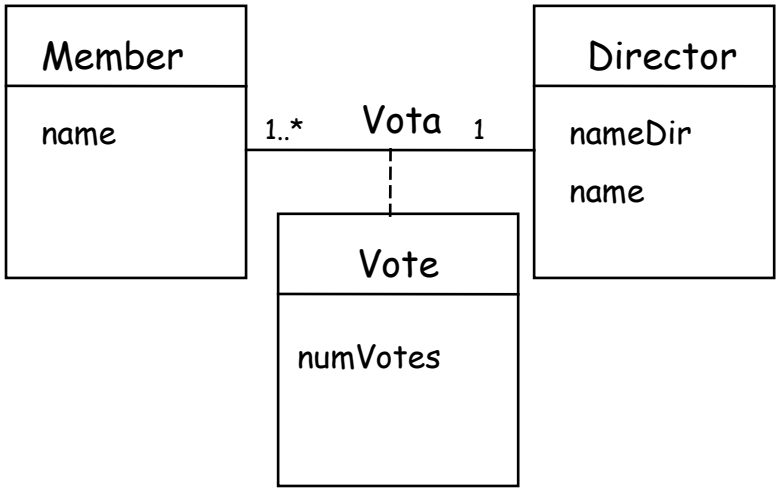
Association “1 to many”



CardAuthorization(cardAuthorizationID, password, limit)

CashCard(cashCardId, serial, *cardAuthorizationID*)

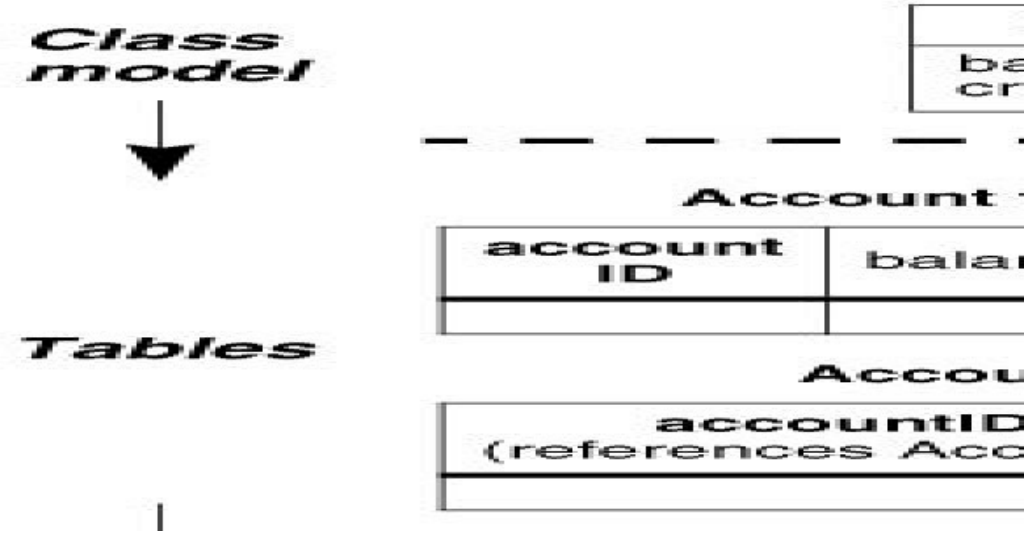
Association “1 to many” with association class



Director(idDirector, name)

Member(idMember, name, numVotes,*idDirector*)

Association “many to many”

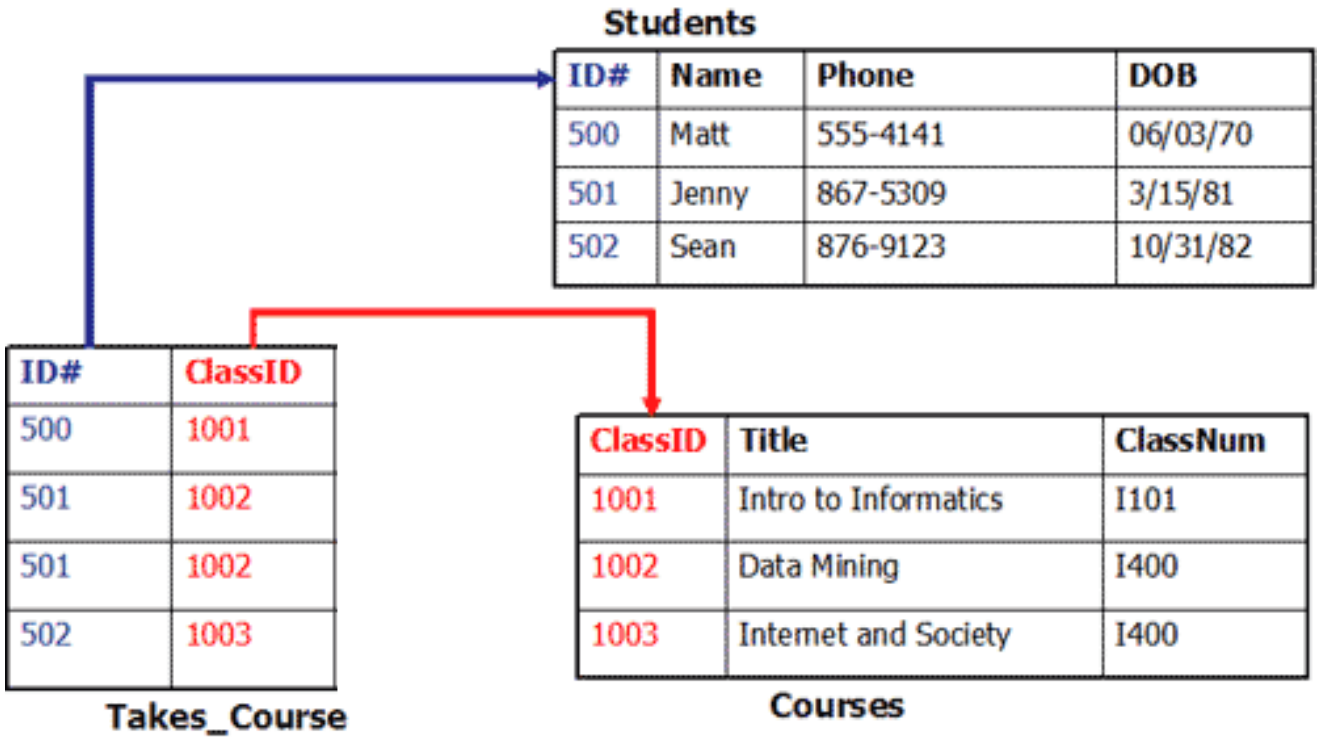


Account(accountID,balance,creditLimit)

CardAuthorization(cardAuthorizationID, password, limit)

Account_CardAthorization(*accountID,cardAthorizationID*)

Association “many to many”



Students

ID#	Name	Phone	DOB
500	Matt	555-4141	06/03/70
501	Jenny	867-5309	3/15/81
502	Sean	876-9123	10/31/82

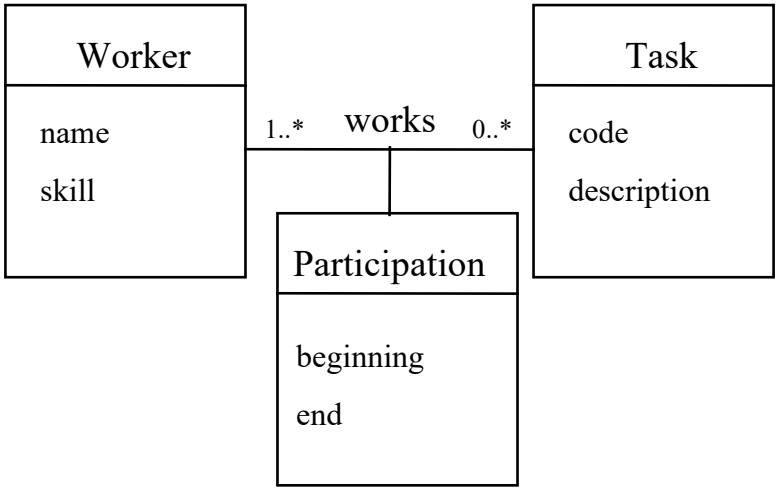
Takes_Course

ID#	ClassID
500	1001
501	1002
501	1002
502	1003

Courses

ClassID	Title	ClassNum
1001	Intro to Informatics	I101
1002	Data Mining	I400
1003	Internet and Society	I400

Association “many to many” with association class

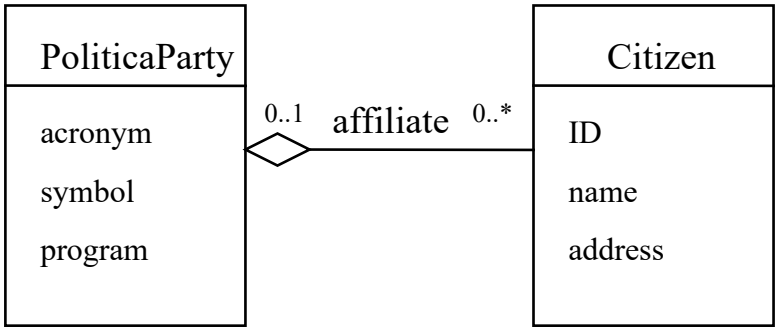


Worker(idWorker, name, skill)

Task(code, description)

Work_Task(*idWorker,code*, beginning, end)

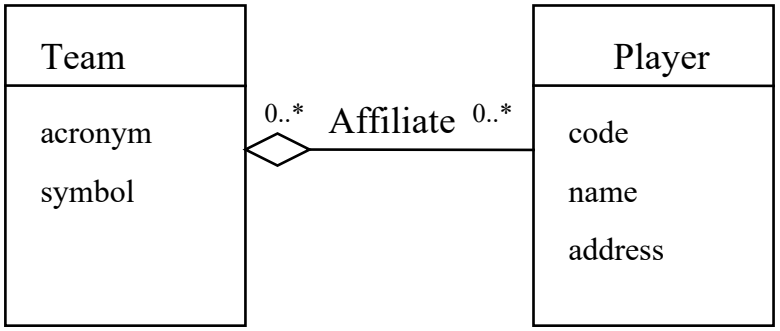
Aggregating



PoliticalParty(<u>acronym</u>,symbol, program)

Citizen(<u>BI</u>,name,address,*acronym*)

Aggregating

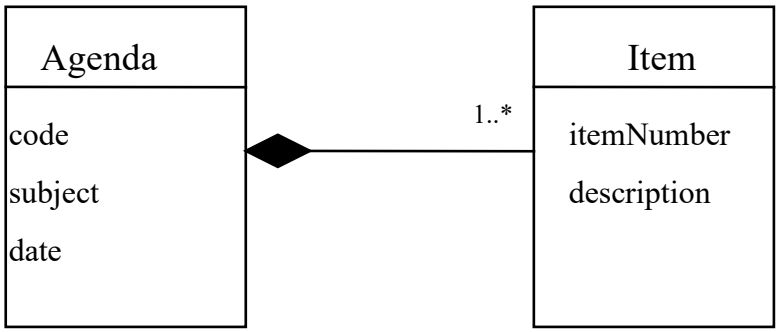


Team(<u>acronym</u>,symbol)

Player(<u>code</u>,name,address)

Tema_Player(<u>*acronym,code*</u>)

Composite Aggregation



Agenda(code,subject,date)

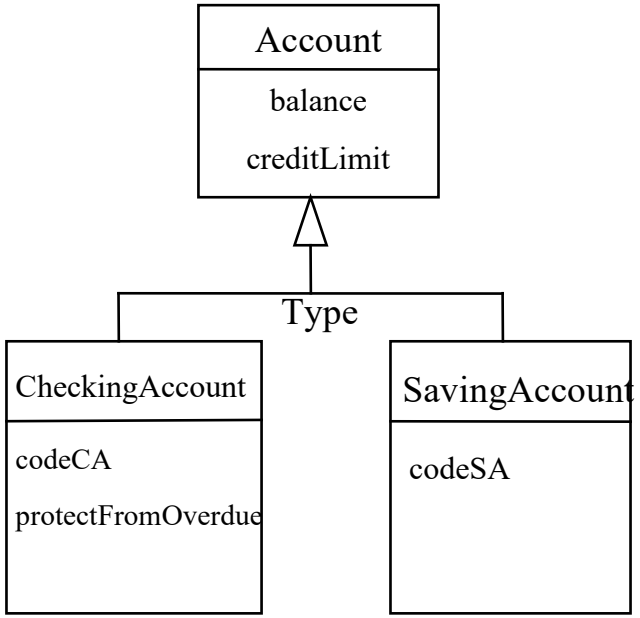
Item(*code*,itemNumber,description)

Generalization

Account(<u>accountID</u>, balance, creditLimit, accountType{CheckingAccount, SavingAccount})
CheckingAccount(<u>*accountID*</u>, protectFromOverdraft)
SavingsAccount(<u>*accountID*</u>)

Generalization

identity



Account(accountID, balance, creditLimit, accountType{CheckingAccount, SavingAccount})
CheckingAccount(codeCA, protectFromOverdraft, *accountID*)
SavingsAccount(codeSA,*accountID*)

Referencias

Ambler, S. (2000) *Mapping objects to relational databasesWhat you need to know and why*, developerWorks, IBM

Blaha, M, Rumbaugh, J. (2004) *Object Oriented Modeling and Design with UML*, Pearson; 2 edition

Costa, C (2004) ; *Transposição de UML para relacional*, GSI ISCTE.