ES. 4) restituire numero e stagione degli episodi della serie TWD che hanno durata superiore a 60 minuti per i quali esistono i sottotitoli in inglese

$$\pi_{\text{NUMERO, STAGIONE, IDE}}\Big(\big(\sigma_{\text{NOME}=\text{'The walking Dead'}}(\text{SERIE})\big) \bowtie_{\text{ID}=\text{IDSERIE}} \big(\rho_{\text{IDE}\leftarrow\text{ID}}\big(\sigma_{\text{DURATA}>60}(\text{EPISODIO})\big)\big) \bowtie_{\text{IDE}=\text{IDEPISODIO}} \big(\sigma_{\text{LINGUA}=\text{'INGLESE'}}(\text{SOTTOTITOLO})\big)\Big)$$

ESAME 2

es. 1: attributi dei concerti che contengono un movimento 'Largo'

```
SELECT C.RV, C.TONALITA', C.NOME, C.DATA
FROM MOVIMENTI AS M NATURAL JOIN CONCERTI AS C
WHERE M.TEMPO = 'Largo'
```

es. 2: attributi dei movimenti dei concerti scritti dopo il 1720 che contengono più di 3 movimenti

```
SELECT C.RV, C.TONALITA', C.NOME, C.DATA
FROM MOVIMENTI AS M NATURAL JOIN CONCERTI AS C
WHERE C.DATA > 1720
GROUP BY C.RV, C.TONALITA', C.NOME, C.DATA
HAVING MAX(M.NUMERO) > 3
```

es. 3: tutti i movimenti dei concerti dove compare almeno un violino che durano più di due minuti

$$\pi_{RV, NUMERO, TEMPO, DURATA}\left(\left(\sigma_{STRUMENTO='VIOLINO'}(STRUMENTAZIONE)\right) \bowtie \left(\sigma_{DURATA>120}(MOVIMENTI)\right)\right)$$

es. 4: concerti dove tra gli strumenti compare il cembalo e hanno solo tre movimenti

$$\left(\pi_{RV}\left(\left(\sigma_{STRUMENTO='CEMBALO'}(STRUMENTAZIONE)\right) \bowtie \left(\left(\sigma_{NUMERO=3}(MOVIMENTI)\right) - \left(\sigma_{NUMERO>3}(MOVIMENTI)\right)\right)\right)\right) \bowtie CONCERTI$$

ESAME 4

es. 1: restituire tutte le famiglie con il numero minore di millesimi

$$\text{FAMIGLIA} - \pi_{\text{VIA,NUMERO,INTERNO,FAMIGLIA,MILLESIMI}}\left(\left(\rho_{M \leftarrow \text{MILLESIMI}}(\text{FAMIGLIA})\right) \bowtie_{M < \text{MILLESIMI}} (\text{FAMIGLIA})\right)$$

es. 2: restituire le famiglie che non hanno spese personali di tipo luce

$$\text{FAMIGLIA} \bowtie \left(\pi_{\text{VIA,NUMERO,INTERNO}}(\text{SPESEPERSONALI} - (\sigma_{\substack{\text{TIPOBOLLETTA} = \text{'LUCE'} \\ \wedge \\ \text{CONSUMO} > 0}}(\text{SPESEPERSONALI})))\right)$$

es. 3: query che associa ad ogni famiglia le spese personali complessive

```
SELECT F.VIA, F.NUMERO, F.INTERNO, SUM(SP.CONSUMO * SP.TARIFFA)
FROM FAMIGLIA AS F LEFT JOIN SPESEPERSONALI AS SP
WHERE SP.CONSUMO IS NOT NULL AND SP.TARIFFA IS NOT NULL
GROUP BY F.VIA, F.NUMERO, F.INTERNO
```

es. 4: query che calcola la somma dei millesimi associati ad ogni numero di scala con almeno una famiglia

```
SELECT S.SCALA, SUM (F.MILLESIMI)
FROM FAMIGLIA AS F JOIN SCALA AS S
     ON F.INTERNO = S.INTERNO
GROUP BY S.SCALA
```

ESERCIZI SLIDE 29/10/24

p.46

es. 1: nome dei musei a Londra che non hanno opere di Tiziano

```
SELECT M.NAME
FROM MUSEUM AS M
WHERE M.CITY = 'London'
      AND NOT EXISTS (SELECT *
                      FROM WORK AS W
                      WHERE M.NAMEA = 'Tiziano'
                            AND M.NAME = W.NAMEM)
```

es. 2: nomi dei musei a Londra che hanno solo opere di Tiziano

```
SELECT M.NAME
FROM MUSEUM AS M
WHERE M.NAME = 'London'
      AND 'Tiziano' = ALL (SELECT W.NAMEA
                           FROM WORK AS W
                           WHERE M.NAME = W.NAMEM)
```

es. 3: nomi dei musei che hanno almeno 20 opere di artisti italiani

```
SELECT W.NAMEM
FROM WORK AS W JOIN ARTIST AS A ON W.NAMEA = A.NAME
WHERE A.NATIONALITY = 'Italy'
GROUP BY W.NAMEM
HAVING COUNT(*) >= 20
```

EMPLOYEE(Code, Surname, Wage, Department)
DEPARTMENT(Id, Name, Location, Director)
PROJECT(Number, Name, Budget, Manager)
STAFF(Employee, Project)

Write a relational algebra expression that returns the names of the projects with a budget greater than 100K and the surnames of the employees who work on them.

Nome progetti con un budget maggiore di 100K e il cognome di chi ci lavora

$$\pi_{Nome, Surnome}((\sigma_{Budget > 100K}(PROJECT)) \bowtie_{Number = Project} STAFF) \bowtie_{Employee = Code} EMPLOYEE)$$

EMPLOYEE(Code, Surname, Wage, Department)
DEPARTMENT(Id, Name, Location, Director)
PROJECT(Number, Name, Budget, Manager)
STAFF(Employee, Project)

Write a relational algebra expression that returns the code and surname of employees not working on any project.

codice e cognome degli impiegati che non lavorano ad alcun progetto

$$EMSTF := EMPLOYEE \bowtie_{Code = Employee} STAFF$$

$$\pi_{Code, Surnome}(EMPLOYEE - (\pi_{Code, Surnome, wage, Department}(EMSTF)))$$

EMPLOYEE(Code, Surname, Wage, Department)
DEPARTMENT(Id, Name, Location, Director)
PROJECT(Number, Name, Budget, Manager)
STAFF(Employee, Project)

Codice e Cognome degli impiegati che lavorano a più progetti

Write a relational algebra expression that returns the code and surname of employees working on more than one project.

$$EMSTF := EMPLOYEE \bowtie_{Code = Employee} STAFF$$

$$\left(\Pi_{code, surname}\left(\left(\rho_{c,p \leftarrow code, project}\left(\left(\Pi_{code, surname, project}(EMSTF)\right)\right)\right) \bowtie_{c = code \wedge p \neq project} \left(\Pi_{code, surname, project}(EMSTF)\right)\right)\right)$$

EMPLOYEE(Code, Surname, Wage, Department)
DEPARTMENT(Id, Name, Location, Director)
PROJECT(Number, Name, Budget, Manager)
STAFF(Employee, Project)

Codice e Cognome delle persone che lavorano su un solo progetto

Write a relational algebra expression that returns the code and surname of the employees working on a single project.

$$\Pi_{code, surname}(EMPLOYEE) - \left(\Pi_{code, surname}\left(\left(\rho_{c,p \leftarrow code, project}\left(\left(\Pi_{code, surname, project}(EMSTF)\right)\right)\right) \bowtie_{c = code \wedge p \neq project} \left(\Pi_{code, surname, project}(EMSTF)\right)\right)\right)$$

SLIDE ALGEBRA

p.30

es.8: codice dei fornitori che forniscono almeno due prodotti diversi

$$\Pi_{SupId}\left(\left(\rho_{S,P \leftarrow SupId,ProdId}(SUPPLYING)\right) \bowtie_{S=SupId \wedge P \neq ProdId} SUPPLYING\right)$$

p.46

es.13: nome delle classi che hanno 100 posti o più che non sono prenotate oggi

$$\Pi_{name}\left(\left(\sigma_{capacity>=100}(ROOM)\right) \bowtie_{code=Room} \left(\sigma_{Date <>' 29/10/24'}(BOOKING)\right)\right)$$

p. 54 n. 17

nomi delle pizzerie frequentate solo da maschi o solo da femmine

$$\pi_{Pizzeria}((\sigma_{gender='Male'}(PERSON)) \bowtie FREQUENTS) - \pi_{Pizzeria}((\sigma_{gender='Female'}(PERSON)) \bowtie FREQUENTS) \cup \pi_{Pizzeria}((\sigma_{gender='Female'}(PERSON)) \bowtie FREQUENTS) - \pi_{Pizzeria}((\sigma_{gender='Male'}(PERSON)) \bowtie FREQUENTS)$$

ESAME 3 REMAKE

es. 1

```
SELECT C.RV, C.TONALITÀ, C.NOME, C.DATA
FROM MOVIMENTI AS M NATURAL JOIN CONCERTI AS C
WHERE M.Tempo = 'Largo'
```

es. 2

```
SELECT C.*
FROM CONCERTI AS C NATURAL JOIN MOVIMENTI AS M
WHERE C.DATA > 1720
GROUP BY C.RV, C.TONALITÀ, C.NOME, C.DATA
HAVING COUNT(M.NUMERO) > 3
```

```
SELECT M.*
FROM CONCERTI AS C
     NATURAL JOIN MOVIMENTI AS M
WHERE C.DATA > 1720
      AND C.RV IN ( SELECT MO.RV
                    FROM MOVIMENTI AS MO
                    GROUP BY MO.RV
                    HAVING COUNT(M.NUMERO) > 3 )
```

es. 3

$$\pi_{RV, Numero, Tempo, Durata}\left(\sigma_{Strumento = \text{'Violino'}}(STRUMENTAZIONE)\right) \bowtie \left(\sigma_{Durata > 120}(MOVIMENTI)\right)$$

es.4

$$\pi_{\text{RV, Tonalità, Nome, Data}} \Big(\text{CONCERTI} \bowtie \Big(\big(\sigma_{\text{Strumento}='\text{Cembalo}'} (\text{STRUMENTAZIONE}) \big) \bowtie \big((\sigma_{\text{Numero}=3} (\text{MOVIMENTI})) - (\sigma_{\text{Numero}>3} (\text{MOVIMENTI})) \big) \Big) \Big)$$

$$\pi_{Nome, Nazione}(GIOCATORE) - \pi_{Nome, Nazione}(VITBIANCHI \cup VITNERI)$$

$$VITBIANCHI := (GIOCATORE \bowtie_{IdScacchista = GiocatoreBianco} (\sigma_{VITTORIA='BIANCO'}(PARTITA)))$$

$$VITNERI := (GIOCATORE \bowtie_{IdScacchista = GiocatoreNero} (\sigma_{VITTORIA='NERO'}(PARTITA)))$$

$$AK := ((\sigma_{Nome='AK'}(GIOCATORE)) \bowtie_{IdScacchiere = GiocatoreBianco \vee IdScacchiere = GiocatoreNero} (PARTITA))$$

$$NS := \left(\left(\sigma_{Nome = 'NS'}(GIOCATORE)\right) \bowtie_{IdScacchiere = GiocatoreBianco \vee IdScacchiere = GiocatoreNero} (PARTITA)\right)$$

$$\left(\left(\rho_{IdP \leftarrow IdPartita}(AK)\right) \bowtie_{IdP = IdPartita} (NS)\right)$$

ESERCIZI 22/11/24

gestione visite mediche, svolte in diverse cliniche, ogni visita e' svolta in una sola clinica

- Code
- indirizzo
- telefono

ENTITÀ

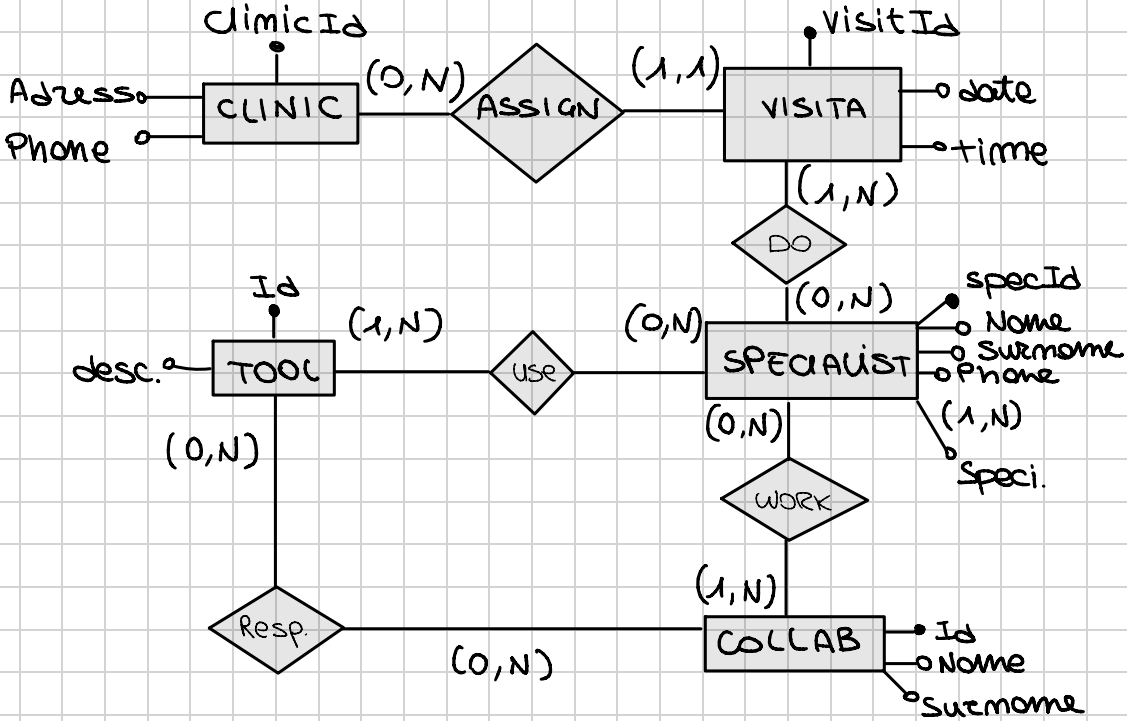
VISITA

SPECIALISTA

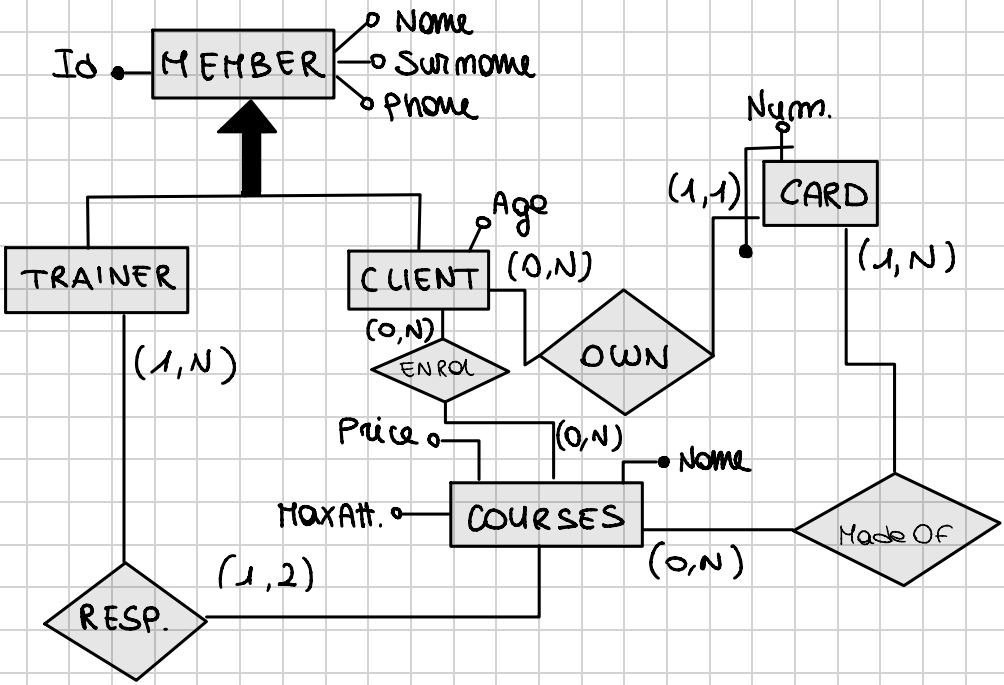
CLINICA

COLLABORATORE

STRUMENTO



ES. 2



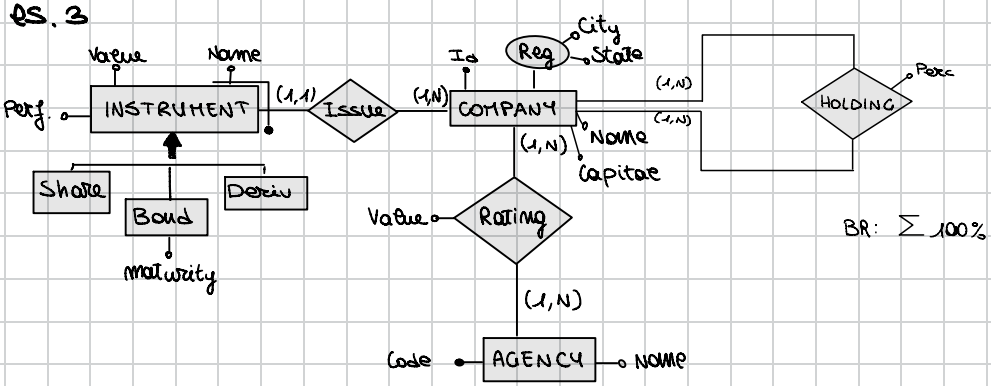
BR0 progressive number

BR1
TOT = Σ Price

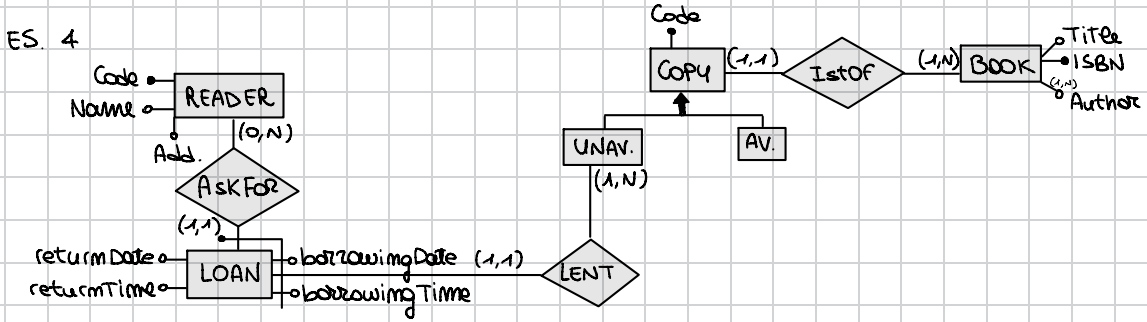
BR2
MaxAtt. $\geq \Sigma$ Clients

27/11/24

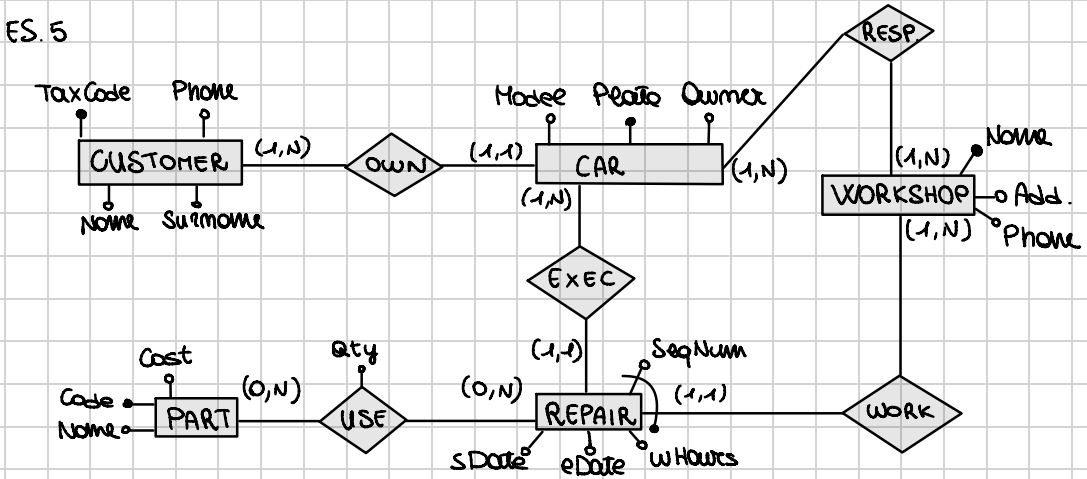
ES. 3



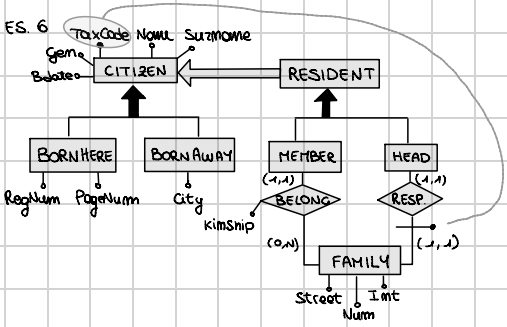
ES. 4



ES. 5



ES. 6



ES. 7

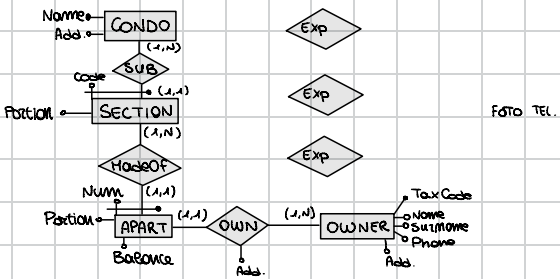
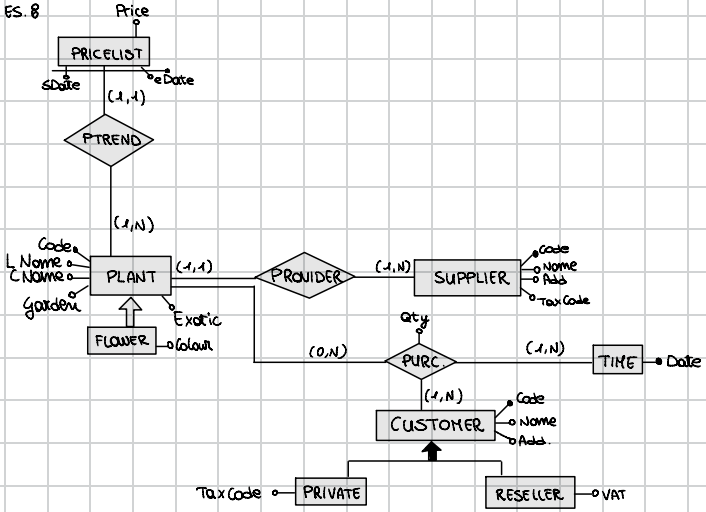


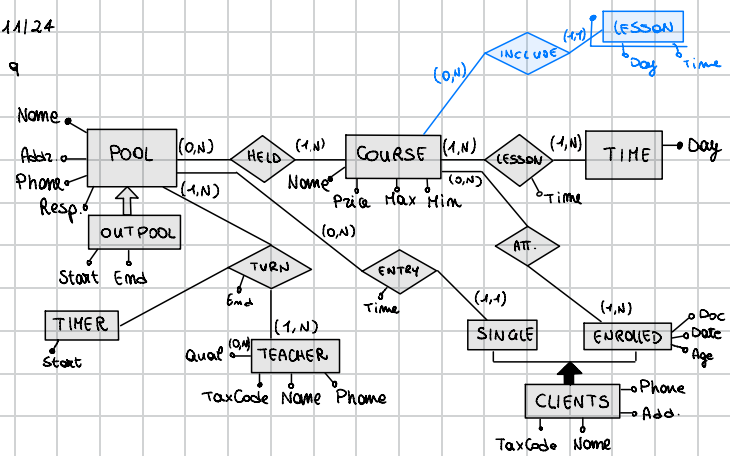
FOTO TEL.

ES. 8

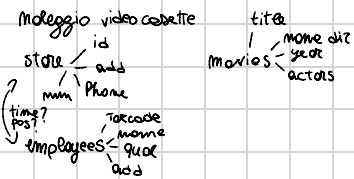


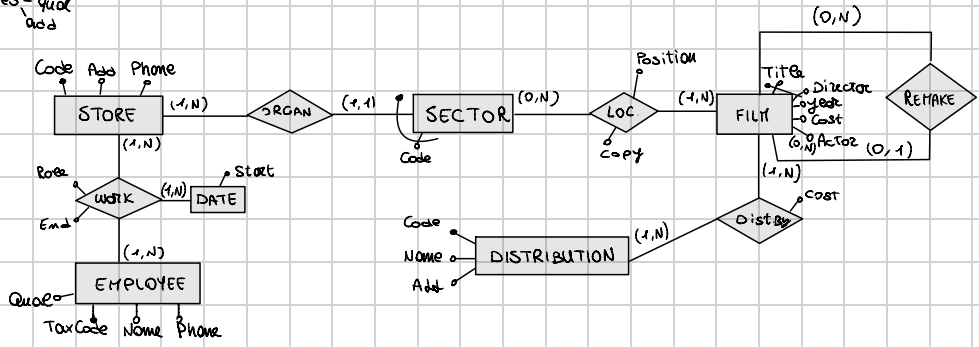
29/11/24

ES. 9

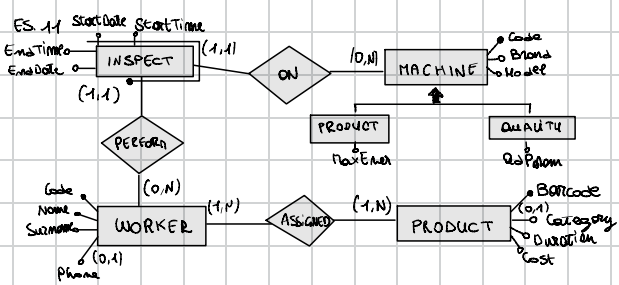


ES. 10

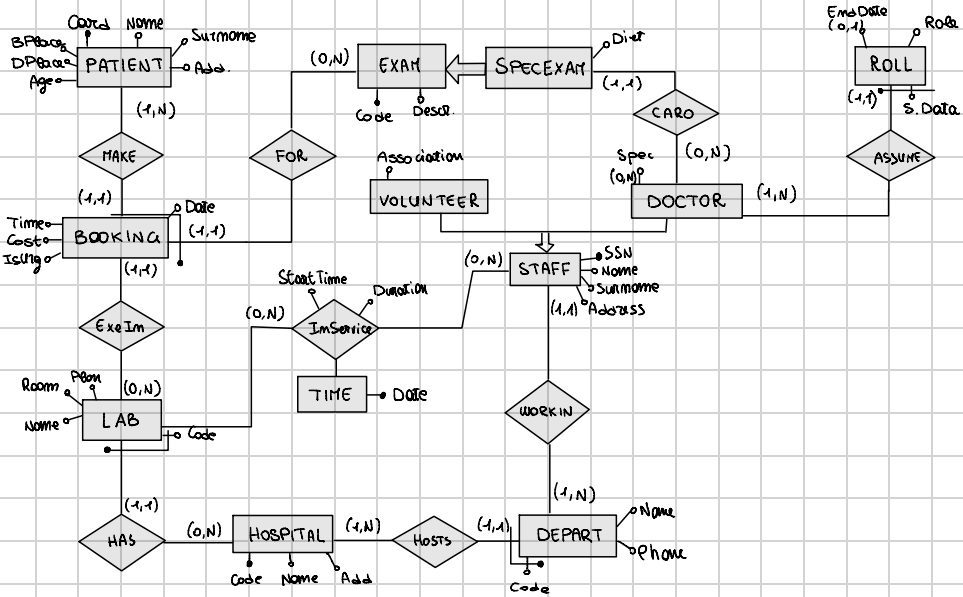




ES. 11



ES. 12

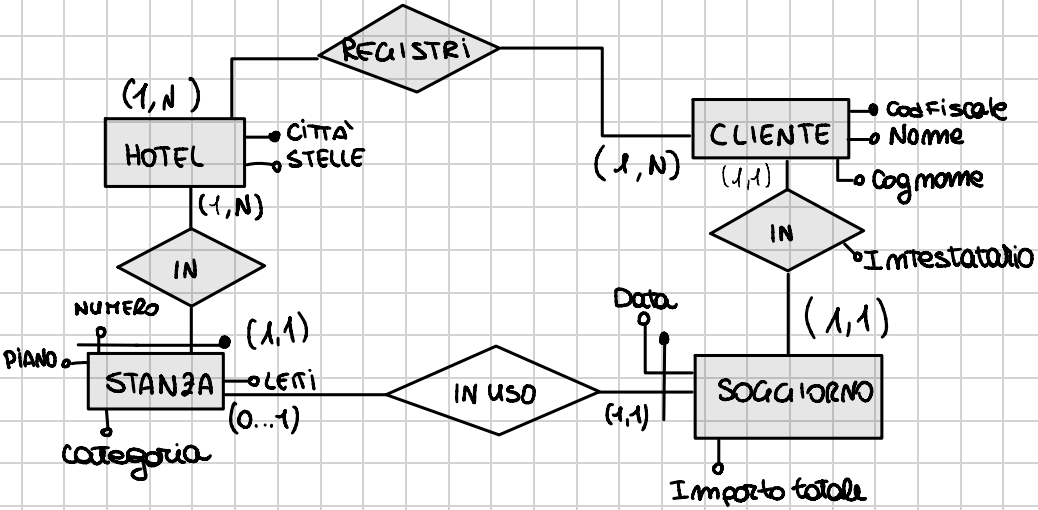

Card
Nome
Surnome
BPlace
DPlace
Age
PATIENT
Add.
(1,N)
MAKE
(1,1)
Date
Time
Cost
Isling
BOOKING
(1,1)
(1,1)
ExeIm
(0,N)
Room
Plan
Nome
LAB
Code
(1,1)
HAS
(0,N)
HOSPITAL
(1,N)
Code
Name
Add
(0,N)
EXAM
Code
Descr.
FOR
Association
VOLUNTEER
Dirr
SPECEXAM
(1,1)
CARO
Spec
(0,N)
(0,N)
DOCTOR
(1,N)
StartTime
Duration
(0,N)
ImService
TIME
Date
(0,N)
STAFF
SSN
Nome
Surnome
Address
(1,1)
WORK IN
(1,N)
HOSTS
(1,1)
DEPART
Name
Phone
Code
EndDate
(0,1)
Role
ROLL
(1,1)
S. Date
ASSUME

TRAINING TEST 1

3) PROGETTAZIONE

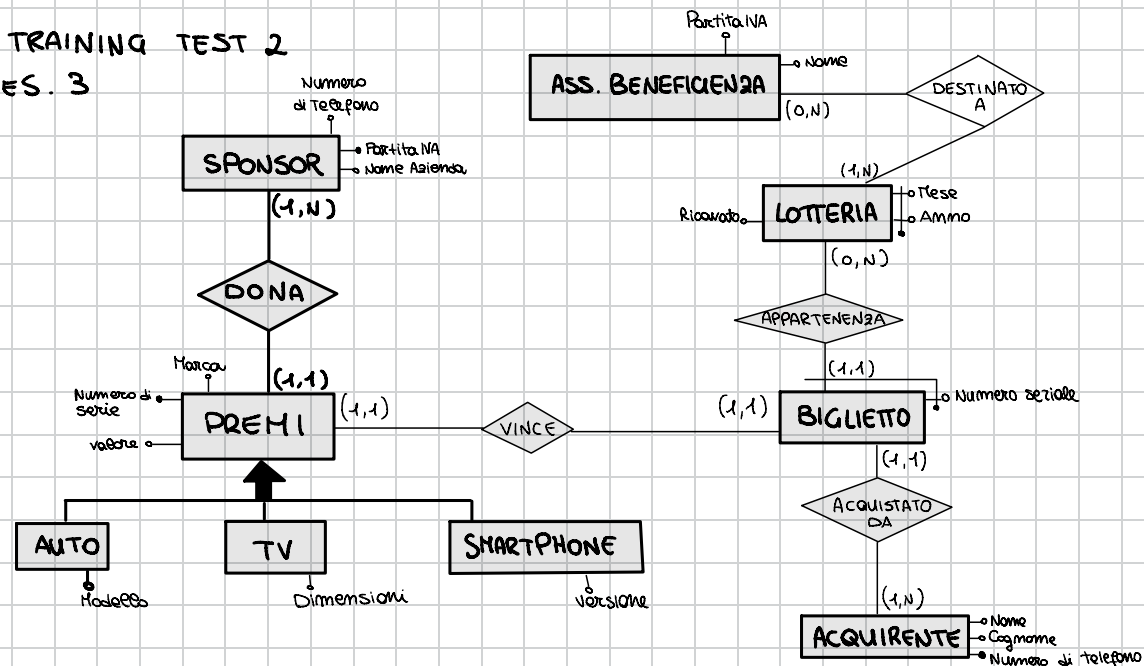
B.R.

$1 < \#\text{LETTI} \leq 3$



TRAINING TEST 2

ES. 3



B.R.: i ricavati sono uguali al prodotto tra il numero di biglietti associati alla lotteria e una costante equivalente al loro prezzo

T.T. 1

ES 4)

1	2
read(x)	
	write(x)
	commit
write(x)	
commit	

=> lost update

l'aggiornamento t_2 è perso dalla sovrascrittura di t_1

1	2
read(x)	
write(x)	
	read(x)
	commit
abort	

=> write-read conflict

t_2 legge un dato potenzialmente invalido

1	2
read(x)	
	read(x)
write(x)	
read(y)	
commit	commit

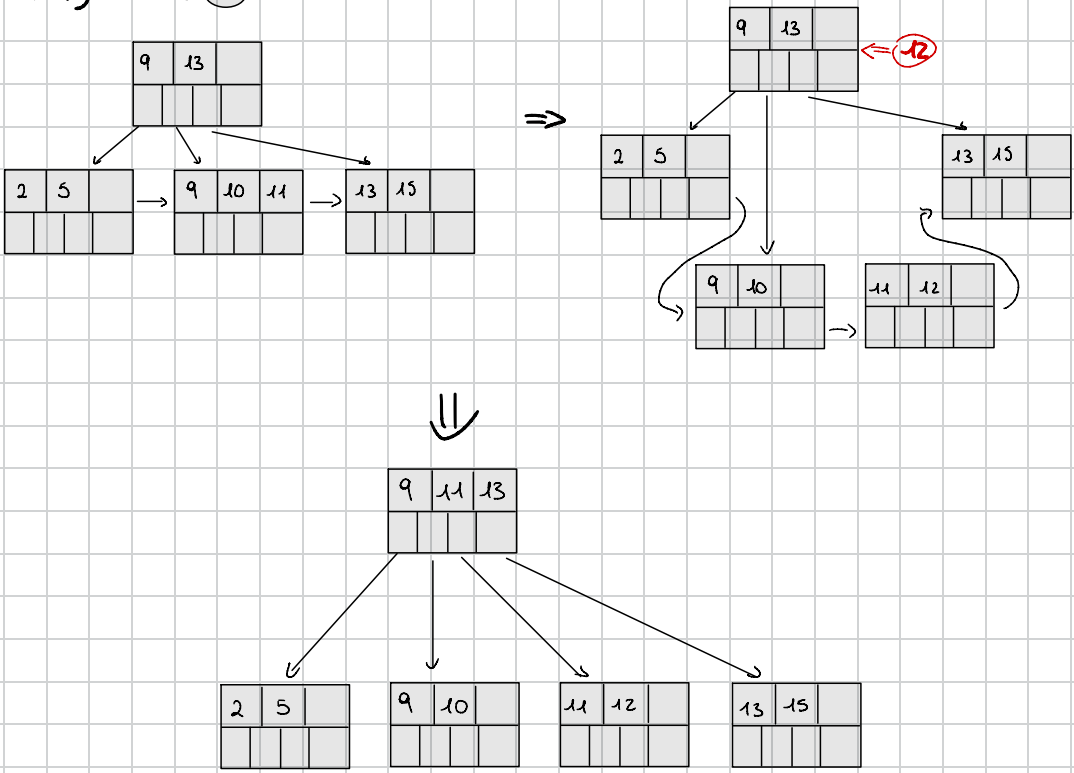
=> ok!

1	2
read(x)	
write(x)	
	read(x)
	write(x)
commit	
	commit

=> ok!

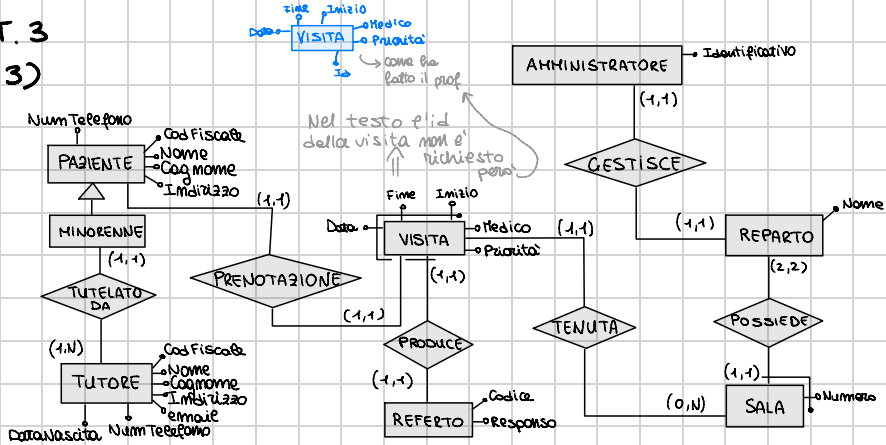
T.T. 2

es. 9) inserire (12)



T.T. 3

es 3)



es 4)

1	2	
write(x)		
	read(x)	dirty reading
	write(y)	inserire lock tra write(x)
abort		e read(x) => se t_1 non e'
	commit	commit non eseguire t_2

1	2
read(x)	
	read(x)
	write(x)
read(y)	
commit	
	commit

=> OK!

con anomalie

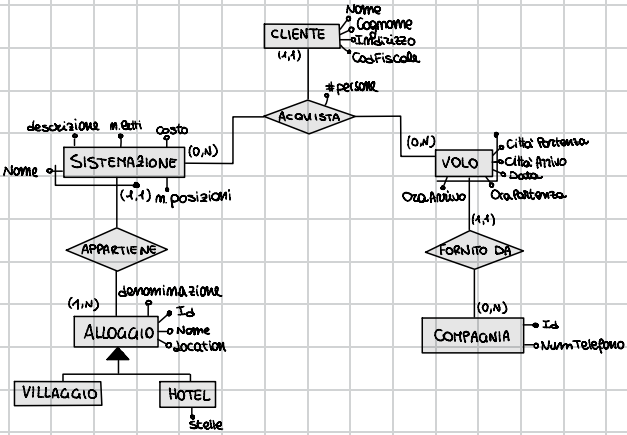
1	2
read(x)	
write(x)	
	read(x)
	write(x)
abort	
	commit

LOCK → prima controllo che t_1 finisca con successo

=> dirty reading: t_2 legge un dato potenzialmente invalido

T.T.4

es 3)


CLIENTE
Nome
Cognome
Indirizzo
CodFiscale
(1,1)
#persone
ACQUISTA
(0,N)
descrizione
m.letti
costo
SISTEMAZIONE
Nome
(1,1)
m.posizioni
APPARTIENE
(1,N)
denominazione
ALLOGGIO
Id
Nome
location
VILLAGGIO
HOTEL
Stelle
(0,N)
VOLO
Citta' Partenza
Citta' Arrivo
Data
Ora Arrivo
Ora Partenza
(1,1)
FORNITO DA
(0,N)
COMPAGNIA
Id
NumTelefono

T.T. 1

es 4)

1.

1	2
read(x)	
	write(x)
	commit
write(x)	
commit	

lost update (write-write)

2.

1	2
read(x)	
write(x)	
	read(x)
	commit
abort	

dirty reading (write-read)

3.

1	2
read(x)	
	read(x)
	write(x)
read(y)	
commit	
	commit

=) OK!

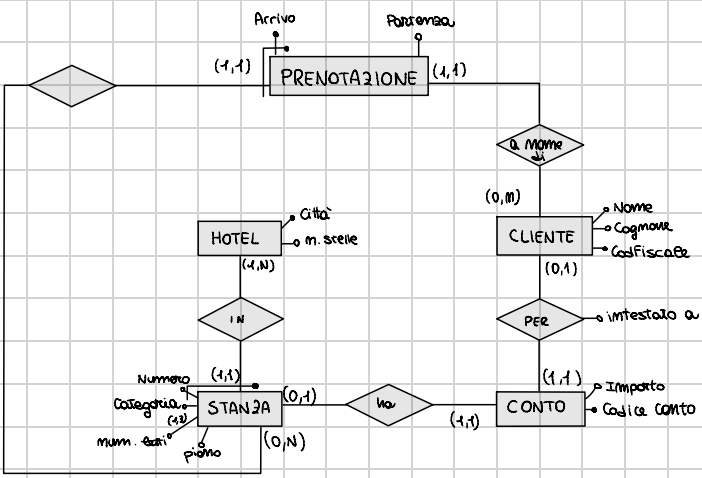
4.

1	2
read(x)	
write(x)	
	read(x)
	write(x)
commit	
	commit

=) OK!

T.T.1

es.3) (di nuovo!)



We want to model a system for the management of a gym and its members.

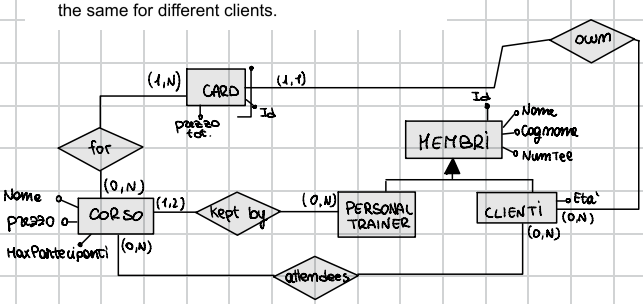
- Each course has a unique name and a fixed price.
- We want to store name, surname and the phone number for all members involved in the gym (both trainers and members).
- Each member has an unique id.
- We want to know the clients' age to fit the exercises workload.
- Each course is taught by one or two trainers.
- Each course has a maximum number of attendees.
- Each member can buy one or more cards, and each card unlocks some courses that are thus purchased at a global price. The final price for each card must not exceed the sum of the prices of the included courses.
- Each client can enrol in a course without buying a card.
- Each card is identified by a progressive number, that is unique for each client. The number can be the same for different clients.

entità:
corso

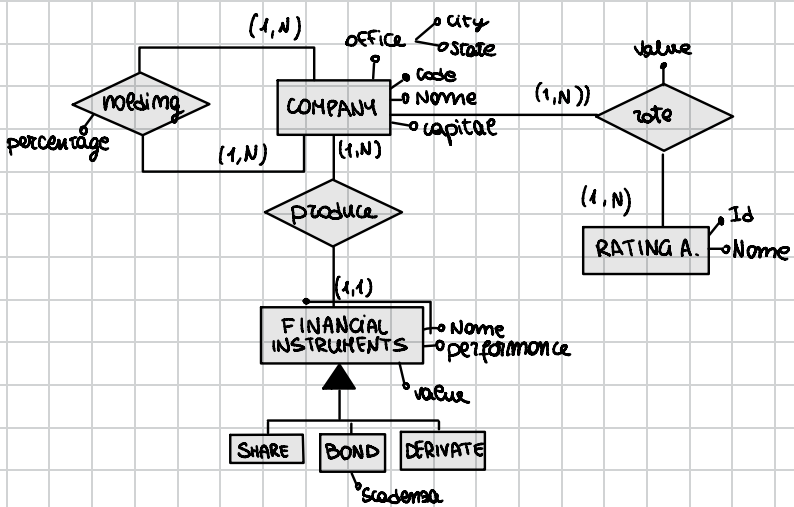
membri (pt e clienti)

pt - corso

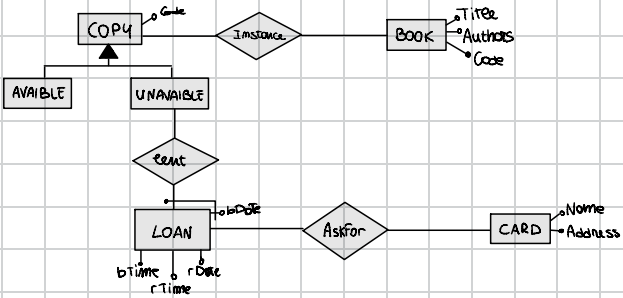
card



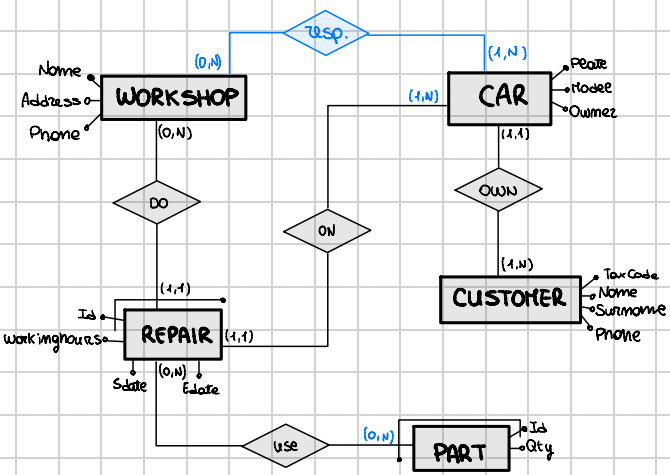
p.33 es.3



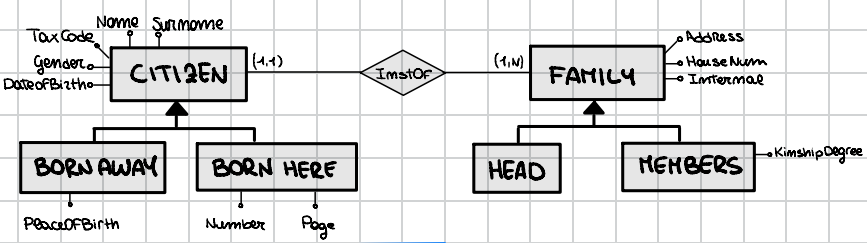
es. 4 p.47

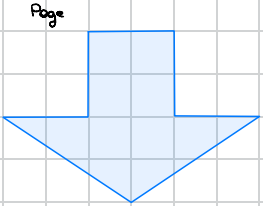


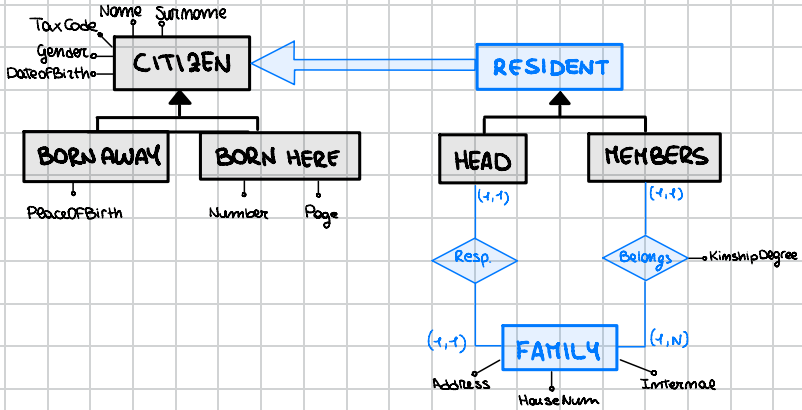
es. 5 p. 50



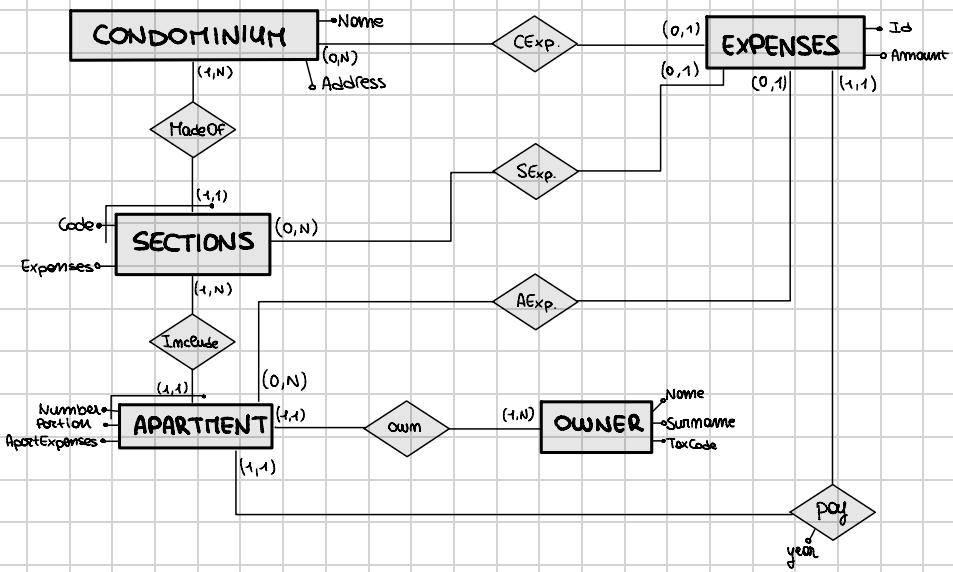
ES. 6 p. 52



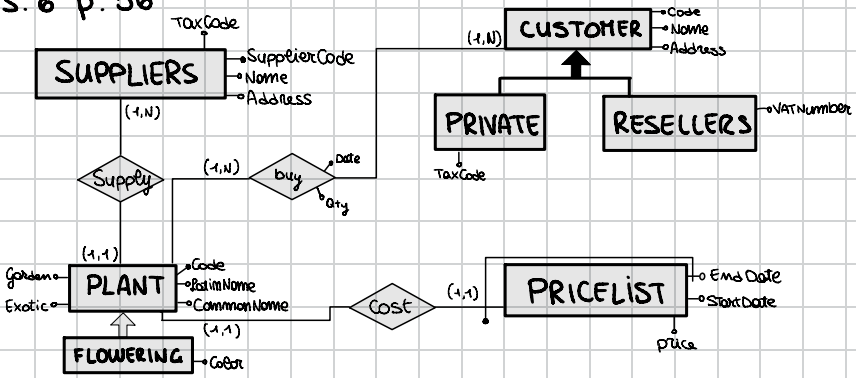



TaxCode
Name
Surname
Gender
DateOfBirth
CITIZEN
RESIDENT
BORN AWAY
BORN HERE
HEAD
MEMBERS
PeaceOfBirth
Number
Page
(1,1)
(1,1)
Resp.
Belongs
KinshipDegree
(1,1)
FAMILY
(1,N)
Address
HouseNum
InternNum

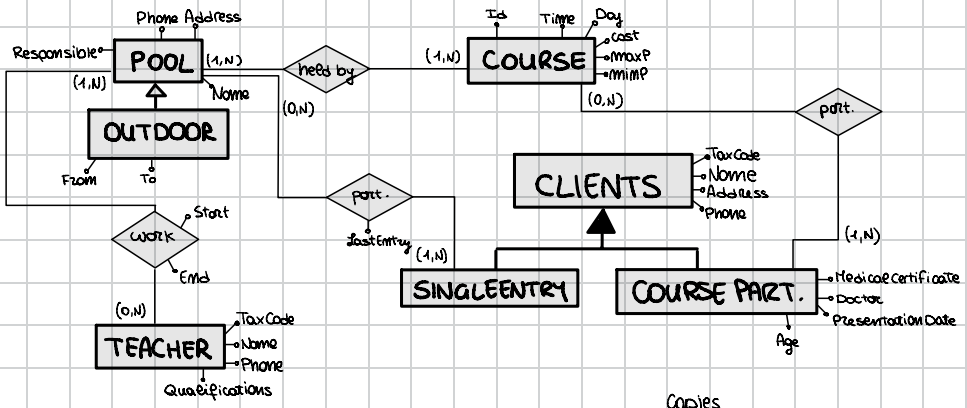
es. 7 p. 54


CONDOMINIUM
Name
(0,N)
Address
CExp.
(0,1)
EXPENSES
Id
Amount
(1,N)
MadeOf
(0,1)
(0,1)
(1,1)
SExp.
(1,1)
Code
SECTIONS
(0,N)
Expenses
(1,N)
AExp.
Include
(1,1)
(0,N)
Number
Portion
ApartExpenses
APARTMENT
(1,1)
own
(1,N)
OWNER
Name
Surname
TaxCode
(1,1)
pay
year

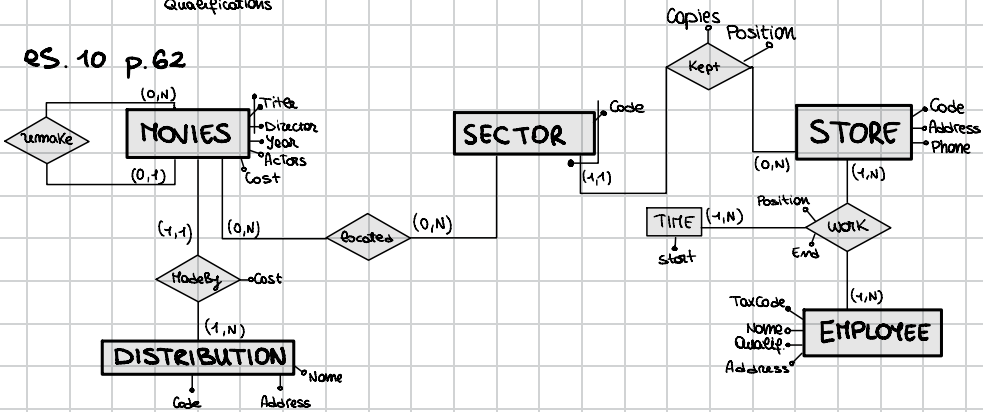
es. 8 p. 56



es. 9 p. 58



es. 10 p. 62



SIMULAZIONE