

Laurea in Ingegneria Informatica
Nuovo Ordinamento
A.A. 2005/2006

Esame di Basi di Dati

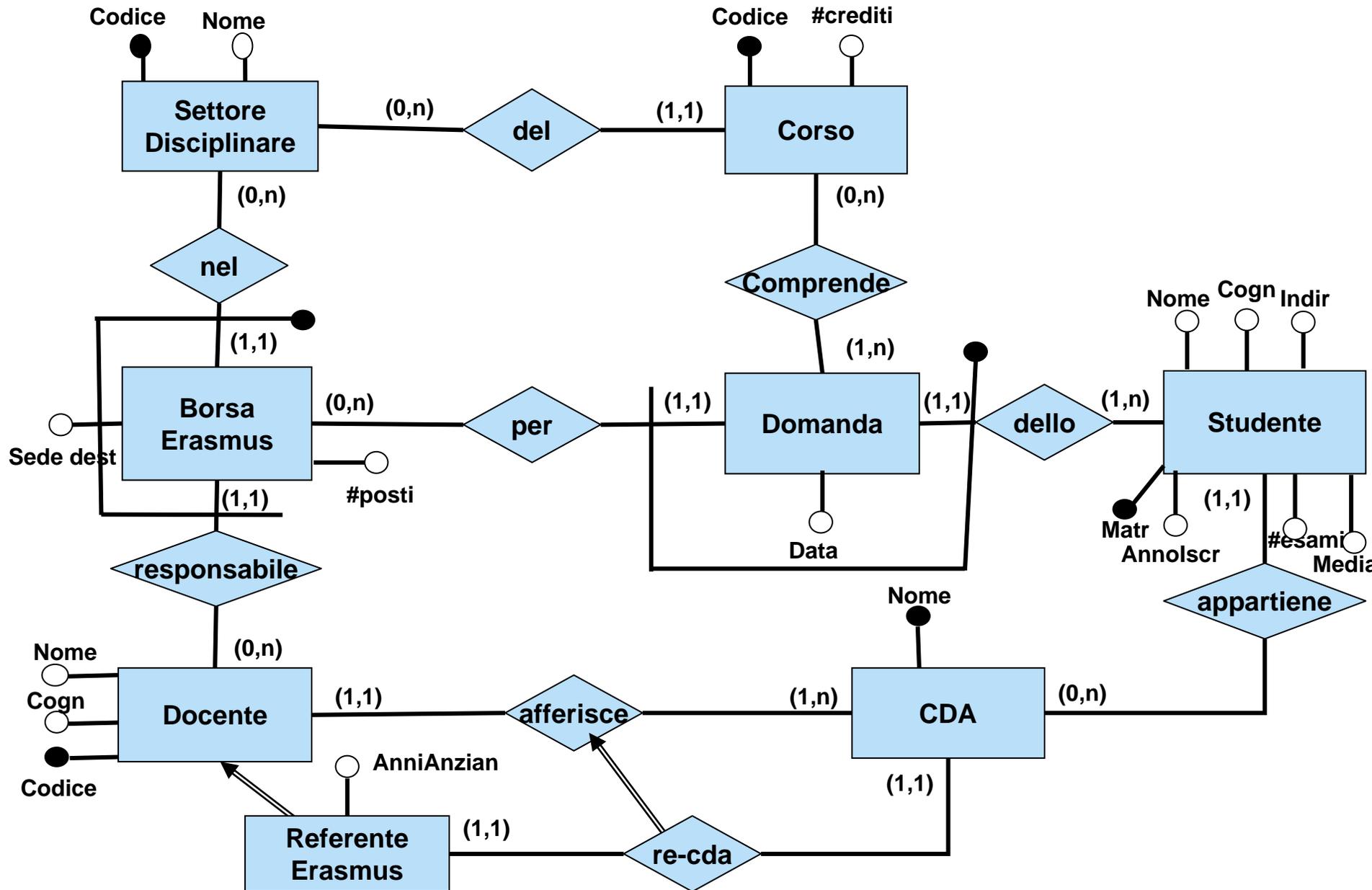
G. De Giacomo – M. Lenzerini

Appello del 19/12/2005

Soluzione del compito A

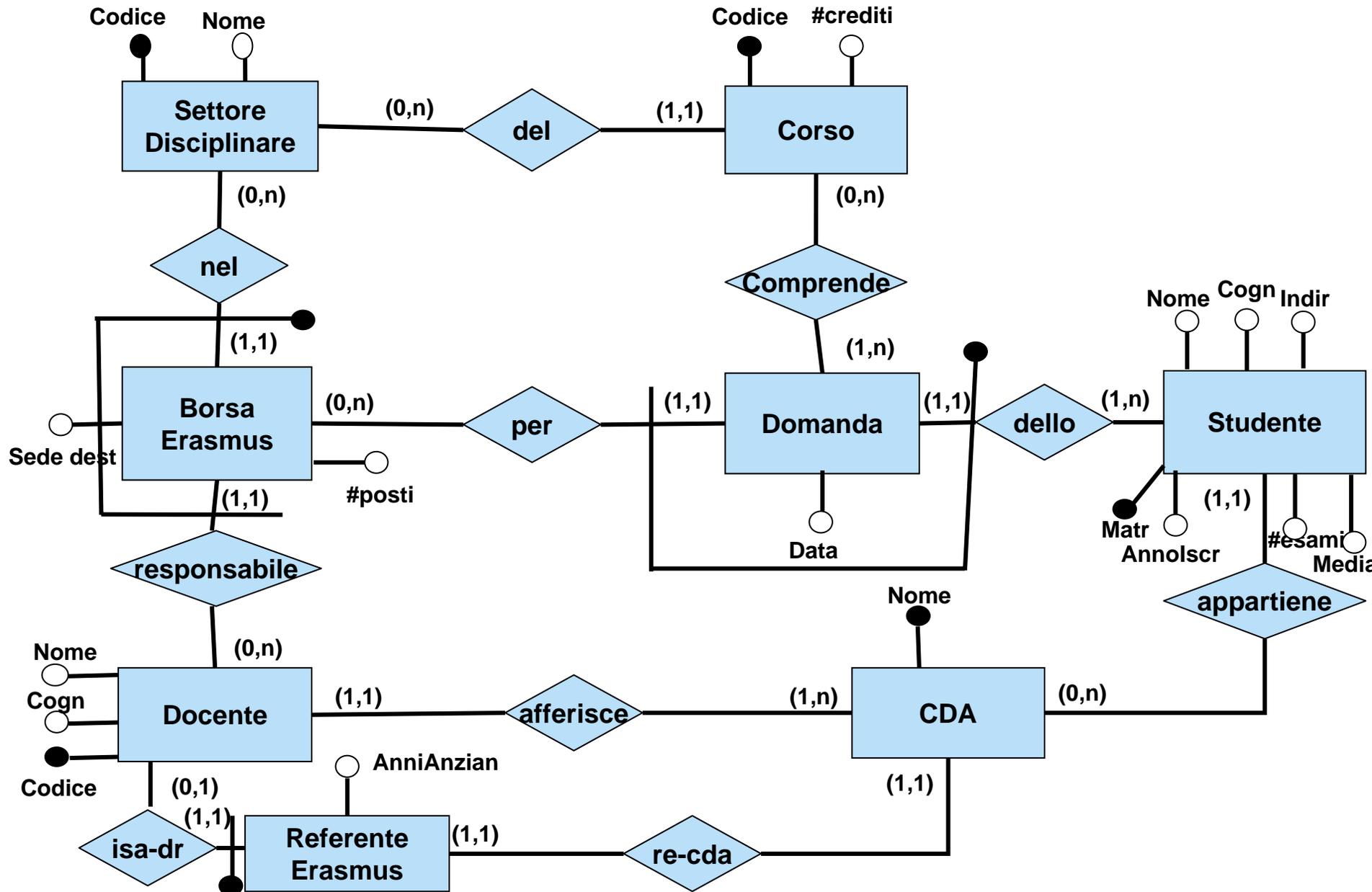
Schema Concettuale

Problema 1 – ER finale



Schema Logico

Problema 2 – ER ristrutturato



Vincoli esterni

- **forall** x,y,x' . $(re-cda(x,y) \text{ and } isa-dr(x',x))$ **implies** $afferisce(x',y)$
per modellare l'isa tra relazioni che ora non è più esprimibile
direttamente

Problema 2 – Schema logico

SettoreDisciplinare(Codice, Nome)

Docente(Codice, Nome, Cognome)

foreign key: Docente[Codice] \subseteq Afferisce[Docente]

BorsaErasmus(SedeDestinazione, SettoreDisciplinare, Responsabile, #posti)

foreign key: BorsaErasmus [SettoreDisciplinare] \subseteq SettoreDisciplinare[Codice]

foreign key: BorsaErasmus [Responsabile] \subseteq Docente[Codice]

CDA(Nome)

foreign key: CDA[Nome] \subseteq Afferisce[CDA]

foreign key: CDA[Nome] \subseteq re-cda[CDA]

Afferisce(Docente, CDA)

foreign key: Afferisce[Docente] \subseteq Docente[Codice]

foreign key: Afferisce[CDA] \subseteq CDA[Nome]

ReferenteErasmus(Codice, AnniAnzianita)

foreign key: ReferenteErasmus[Codice] \subseteq Docente[Codice]

foreign key: ReferenteErasmus[Codice] \subseteq re-cda[ReferenteErasmus]

re-cda(ReferenteErasmus, CDA)

foreign key: re-cda[ReferenteErasmus] \subseteq ReferenteErasmus[Codice]

foreign key: re-cda[CDA] \subseteq CDA[Nome]

unique (CDA)

include: re-cda[ReferenteErasmus, CDA] \subseteq Afferisce[Docente, CDA] coglie il vincolo esterno dovuto all'isa tra relazioni

Domanda(BorsaSede, BorsaSettoreDisciplinare, BorsaResponsabile, Studente, Data)

foreign key: Domanda[BorsaSede, BorsaSettoreDisciplinare, BorsaResponsabile] \subseteq

BorsaErasmus[SedeDestinazione, SettoreDisciplinare, Responsabile]

foreign key: Domanda[Studente] \subseteq Studente[Matricola]

include: Domanda[BorsaSede, BorsaSettoreDisciplinare, BorsaResponsabile, Studente] \subseteq

Comprende[BorsaSede, BorsaSettoreDisciplinare, BorsaResponsabile, Studente]

Studente(Matricola, Nome, Cognome, Indirizzo, AnnoIscrizione, #Esami, Media)

include: Studente[Matricola] \subseteq Domanda[Studente]

foreign key: Studente[Matricola] \subseteq Appartiene[Studente]

Appartiene(Studente, CDA)

foreign key: Appartiene[Studente] \subseteq Studente[Matricola]

foreign key: Appartiene[CDA] \subseteq CDA[Nome]

Corso(Codice, #crediti)

foreign key: Corso[Codice] \subseteq Del[Corso]

Del(Corso, SettoreDisciplinare)

foreign key: Del[Corso] \subseteq Corso[Codice]

foreign key: Del[SettoreDisciplinare] \subseteq SettoreDisciplinare[Corso]

Comprende(BorsaSede, BorsaSettoreDisciplinare, BorsaResponsabile, Studente, Corso)

foreign key: Comprende[Corso] \subseteq Corso[Codice]

foreign key: Comprende[BorsaSede, BorsaSettoreDisciplinare, BorsaResponsabile, Studente] \subseteq

Domanda[BorsaSede, BorsaSettoreDisciplinare, BorsaResponsabile, Studente]

Problema 2 – Ristrutturazione dello schema logico

- Poiché si accede ai dati anagrafici dello studente (nome, cognome, indirizzo) separatamente rispetto a quelli curriculari (anno di iscrizione, media e numero esami superati), effettuiamo la decomposizione verticale di **Studente** in:

Studente(Matricola, AnnoIscrizione, #esami, Media)

StudenteAnagrafica(Matricola, Nome, Cognome, Indirizzo)

foreign key: **StudenteAnagrafica**[Matricola] \subseteq **Studente**[Matricola]

Il resto dello schema logico rimane immutato (anche se ovviamente ora si fa riferimento alla nuova tabella **Studente**)

- Poiché quando si accede ai dati curriculari si vuole anche accedere al CDA di appartenenza dello studente, effettuiamo l'accorpamento di **Studente**, e **Appartiene**, ottenendo una nuova relazione **Studente** definita come segue:

Studente(Matricola, AnnoIscrizione, #esami, Media, CDA)

include: **Studente**[Matricola] \subseteq **Domanda**[Studente]

foreign key: **Studente**[CDA] \subseteq **CDA**[Nome]

Come prima, il resto dello schema logico rimane immutato (anche se ovviamente ora si fa riferimento alla nuova tabella **Studente**)

Query

Problema 3 – Interrogazioni SQL

1. Per ogni domanda, restituire la matricola, la media, il numero di esami superati, l'anno di iscrizione ed il consiglio d'area dello studente che l'ha presentata.

```
select
D.BorsaSede, D.BorsaSettoreDisciplinare, D.BorsaResponsabile,
D.Studente, S.Media, S.#esami, S.AnnoIscrizione, S.CDA
from Domanda D, Studente S,
where D.Studente = S.Matricola
```

2. Restituire le domande per le quali tutti i corsi che lo studente intende frequentare sono nel settore disciplinare della borsa alla quale si riferisce la domanda .

```
select D.BorsaSede, D.BorsaSettoreDisciplinare,
D.BorsaResponsabile, D.Studente
from Domanda D
```

minus

```
select C.BorsaSede, C.BorsaSettoreDisciplinare,
C.BorsaResponsabile, C.Studente
form Comprende C, Del CSD
where C.Corso = CSD.Corso and
      CSD.SettoreDisciplinare <> C.BorsaSettoreDisciplinare
```

Problema 3 – Interrogazioni SQL (cont.)

3. Un consiglio d'area viene detto "rilevante per Erasmus" se sono state presentate almeno 10 domande di borse Erasmus da studenti appartenenti al consiglio d'area stesso. Per ciascun consiglio d'area rilevante per Erasmus, calcolare il numero di domande presentate da studenti ad esso appartenenti.

```
select S.CDA, count(*)  
from Domanda D, Studente S  
where D.Studente = S.Matricola  
group by S.CDA  
having count(*) >= 10
```