

Laurea in Ingegneria Informatica
Nuovo Ordinamento
A.A. 2002/2003

Esame di Basi di Dati

Appello del 19/12/2002
Compito A

Specifica

Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa ai progetti di basi di dati condotti da una azienda informatica *A*. L'azienda *A* in questione vuole tenere traccia di un insieme di informazioni concernenti tali progetti, descritte nel seguito. Ogni progetto ha un numero identificativo, un costo ed una durata (in giorni). Ogni progetto viene effettuato per uno o più committenti (cioè aziende che incaricano *A* di effettuare il progetto), e ogni committente eroga, per ogni progetto, un relativo compenso all'azienda *A*. Un committente può richiedere uno o più progetti. Per ogni progetto, l'azienda *A* produce uno o più schemi. Ogni schema viene sviluppato per uno ed un solo progetto, ed è caratterizzato da un numero e dal tempo (in giorni) impiegato per lo sviluppo. Il numero di uno schema è unico nell'ambito del progetto per il quale lo schema è stato sviluppato. Esistono esattamente due tipi di schemi: gli schemi concettuali e gli schemi logici. Di ogni schema concettuale interessa il numero di progettisti che hanno concorso al suo sviluppo. Di ogni schema logico interessa la dimensione (espressa come un numero intero), ed il DBMS impiegato per la sua implementazione. Ogni schema logico è implementato in uno ed un solo DBMS. Di ogni schema logico *S* interessa anche conoscere quale sia lo schema concettuale utilizzato come input per lo sviluppo dello schema *S* stesso. Ogni schema logico *S* ha infatti uno ed un solo schema concettuale in input, e tale schema concettuale deve essere stato prodotto per lo stesso progetto per il quale è stato prodotto *S*. Ogni DBMS ha un codice identificativo e l'anno di rilascio. Di ogni DBMS interessa conoscere quali sono i committenti che hanno la licenza per usarlo, con l'indicazione dell'anno di inizio di validità della licenza stessa. Infine, di ogni committente interessa il codice fiscale (identificativo), il numero di dipendenti, e, se noto, il capitale sociale.

Specifica (cont.)

Problema 2

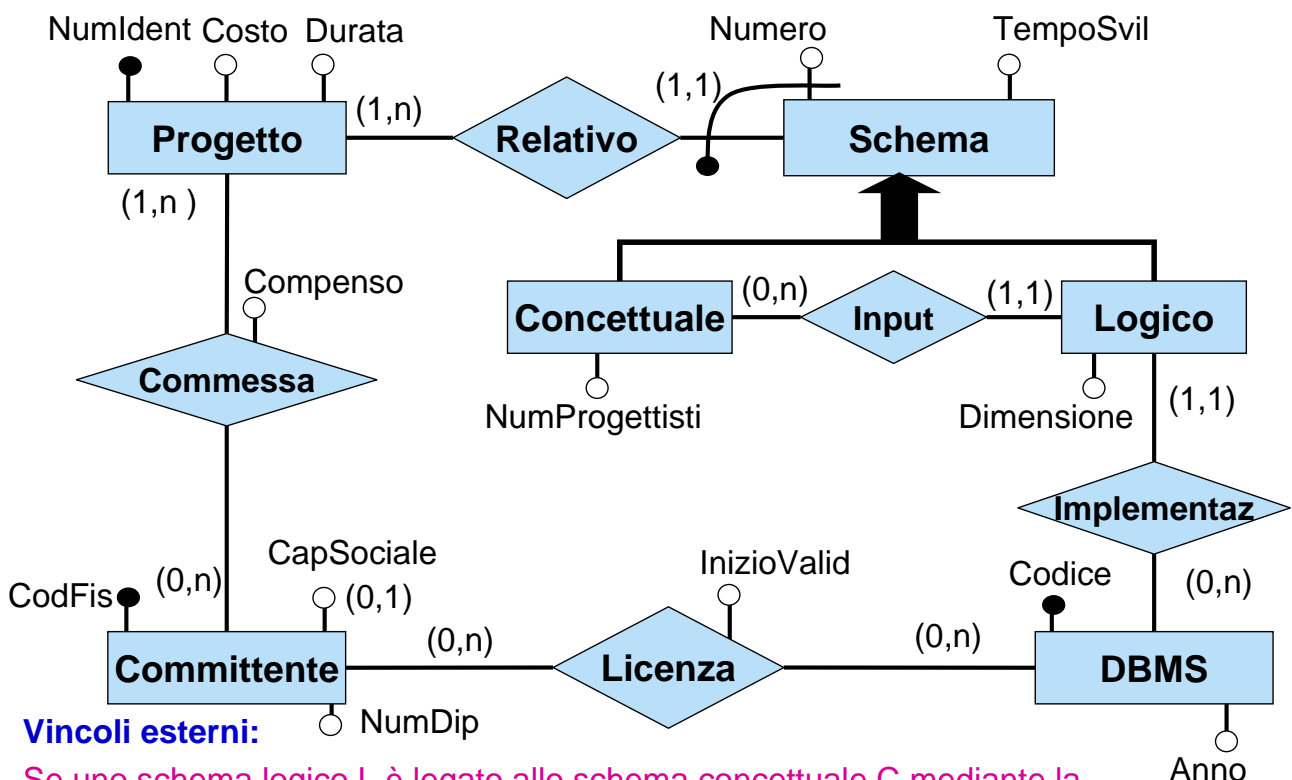
Si progetti lo schema logico dell'applicazione, illustrando lo schema relazionale corrispondente (non necessariamente espresso in SQL), completo di vincoli, e seguendo l'unica indicazione di evitare valori nulli nella base di dati (si noti che le interrogazioni menzionate si seguito **non** sono da considerarsi significative per le indicazioni di progetto).

Problema 3

Dato lo schema relazionale prodotto, esprimere le seguenti interrogazioni in SQL:

1. Per ogni schema concettuale che ha richiesto più di 30 giorni per lo sviluppo, si vogliono conoscere i dati relativi al progetto ed al numero.
2. Per ogni schema logico di dimensione maggiore di 100, si vogliono conoscere il tempo di sviluppo, ed il costo del relativo progetto.
3. Fornire la lista dei progetti per i quali è stato prodotto almeno uno schema logico implementato in un DBMS per il quale almeno un committente del relativo progetto non ha la licenza.
4. Produrre la lista di tutti i progetti il cui costo è inferiore al compenso totale che hanno determinato per l'azienda, dove il compenso totale che un progetto determina per l'azienda è semplicemente la somma dei compensi erogati dai relativi committenti per quel progetto.

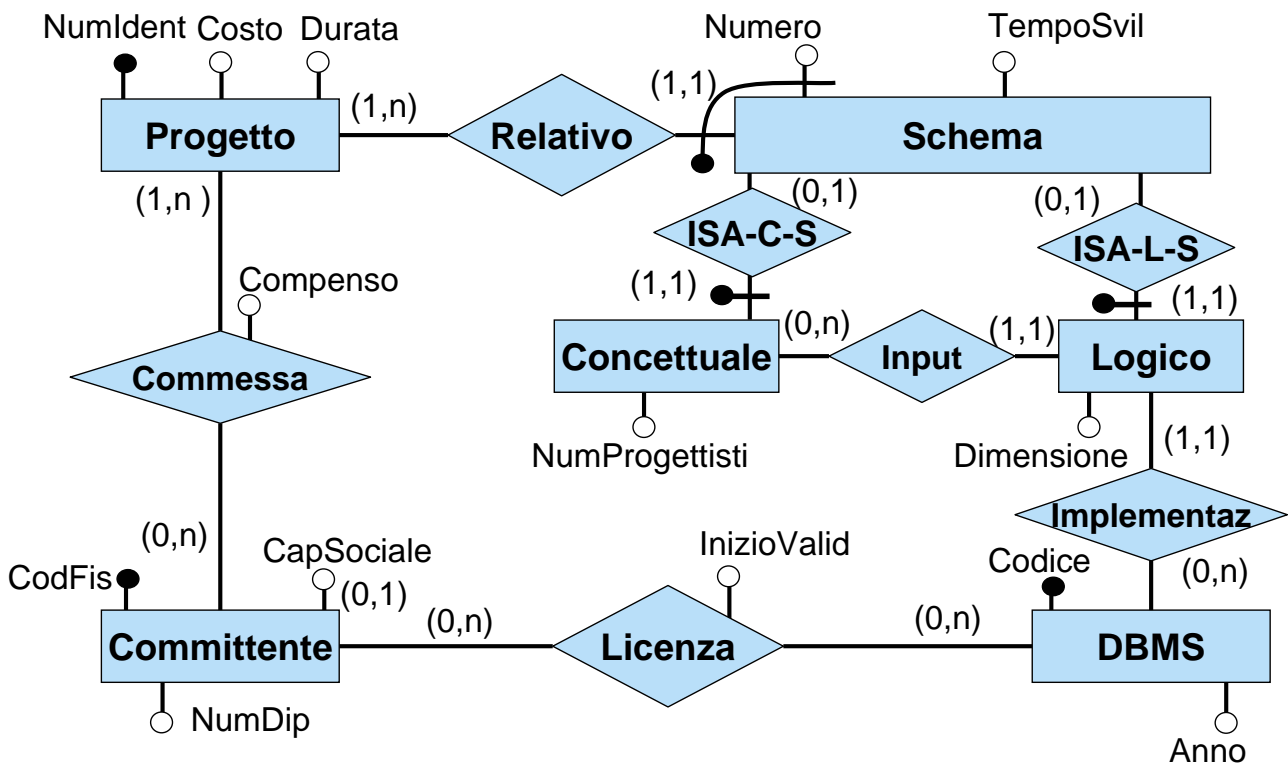
Problema 1 – Schema concettuale



Vincoli esterni:

Se uno schema logico L è legato allo schema concettuale C mediante la relazione Input, allora L e C sono legati dalla relazione Relativo allo stesso progetto.

Problema 2 – Schema concettuale ristrutturato



Problema 2 – Vincoli sullo schema concettuale ristrutturato

- Se uno schema logico L è legato allo schema concettuale C mediante la relazione Input, allora L e C sono legati dalla relazione Relativo allo stesso progetto.
- **Ogni istanza di Schema partecipa ad ISA-C-S oppure ad ISA-L-S ma non ad entrambe** (dovuto all'eliminazione della generalizzazione).

Problema 2 – Schema logico

Progetto(NumIdent, Costo, Durata)

inclusione: Progetto[NumIdent] \subseteq Schema[Progetto]

inclusione: Progetto[NumIdent] \subseteq Commessa[Progetto]

Schema(Progetto, Numero, TempoSvil)

foreign key: Schema[Progetto] \subseteq Progetto[NumIdent]

Concettuale(Progetto, Numero, NumProgettisti)

foreign key: Concettuale[Progetto,Numero] \subseteq Schema[Progetto,Numero]

Logico(Progetto, Numero, Dimensione)

foreign key: Logico[Progetto,Numero] \subseteq Schema[Progetto,Numero]

foreign key: Logico[Progetto,Numero] \subseteq Input[NumeroSL, VersioneSL]

Input(ProgettoSL, NumeroSL, ProgettoSC, NumeroSC)

foreign key: Input[ProgettoSL,NumeroSL] \subseteq Logico[Progetto, Numero]

foreign key: Input[ProgettoSC,NumeroSC] \subseteq Concettuale[Progetto, Numero]

Implementazione(ProgettoSL, NumeroSL, DBMS)

foreign key: Implementazione[ProgettoSL,NumeroSL] \subseteq
Logico[Progetto,Numero]

foreign key: Implementazione[DBMS] \subseteq DBMS[Codice]

Problema 2 – Schema logico (cont.)

DBMS(Codice, Anno)

Licenza(DBMS, Committente, InizioValid)

foreign key: Licenza[Committente] \subseteq Committente[CodFis]

foreign key: Licenza[DBMS] \subseteq DBMS[Codice]

Committente(CodFis, NumDip, CapSociale*)

Commessa(Committente, Progetto, Compenso)

foreign key: Commessa[Committente] \subseteq Committente[CodFis]

foreign key: Commessa[Progetto] \subseteq Progetto[NumIdent]

Vincoli di generalizzazione:

Logico[Progetto, Numero] \cap Concettuale[Progetto, Numero] = \emptyset

Schema[Progetto, Numero] \subseteq

Logico[Progetto, Numero] \cup Concettuale[Progetto, Numero]

Vincoli esterni:

per ogni tupla $t \in$ Input: $t[\text{ProgettoSL}] = L[\text{ProgettoSC}]$

Problema 2 – Ristrutturazione dello schema logico

- Per eliminare i valori nulli di **CapSociale** in **Committente**, effettuiamo una decomposizione mista di **Committente** in:

Committente(CodFis, NumDip)

CommConCS(CodFis, CapSociale)

foreign key: **CommConCS**[CodFis] \subseteq **Committente**[CodFis]

- Dato il vincolo esterno che uguaglia i valori di **ProgettoSL** e **ProgettoSC** nelle tuple di **Input**, uno dei due attributi può essere eliminato. La relazione **Input** diventa:

Input(Progetto, NumeroSL, NumeroSC)

foreign key: **Input**[Progetto, NumeroSL] \subseteq **Logico**[Progetto, Numero]

foreign key: **Input**[Progetto, NumeroSC] \subseteq **Concettuale**[Progetto, Numero]

Problema 3 – Interrogazioni SQL

1.

```
select C.Progetto, C.Numero
from Concettuale C, Schema S
where C.Progetto = S.Progetto and
      C.Numero = S.Numero and
      S.TempoSvil > 30
```
2.

```
select P.Costo, S.TempoSvil
from Progetto P, Schema S, Logico L,
where P.NumIdent = S.Progetto and
      S.Progetto = L.Progetto and
      S.Numero = L.Numero and
      L.Dimensione > 100
```

Problema 3 – Interrogazioni SQL (cont.)

```
3. select L.Progetto
   from Logico L, Commessa C, Implementazione I
  where L.Progetto = C.Progetto and
        L.Progetto = I.Progetto and L.Numero = I.NumeroSL
        and I.DBMS not in
          (select DBMS
           from Licenza LI
          where  LI.Committente = C.Committente)
```

```
4. select P.NumIdent
   from Progetto P, Commessa C
  where P.NumIdent = C.Progetto
 group by P.NumIdent, P.Costo
 having P.Costo < sum(C.Compenso)
```