

Basi di dati

Giuseppe De Giacomo

***Dipartimento di Informatica e Sistemistica "Antonio Ruberti"
Università di Roma "La Sapienza"***

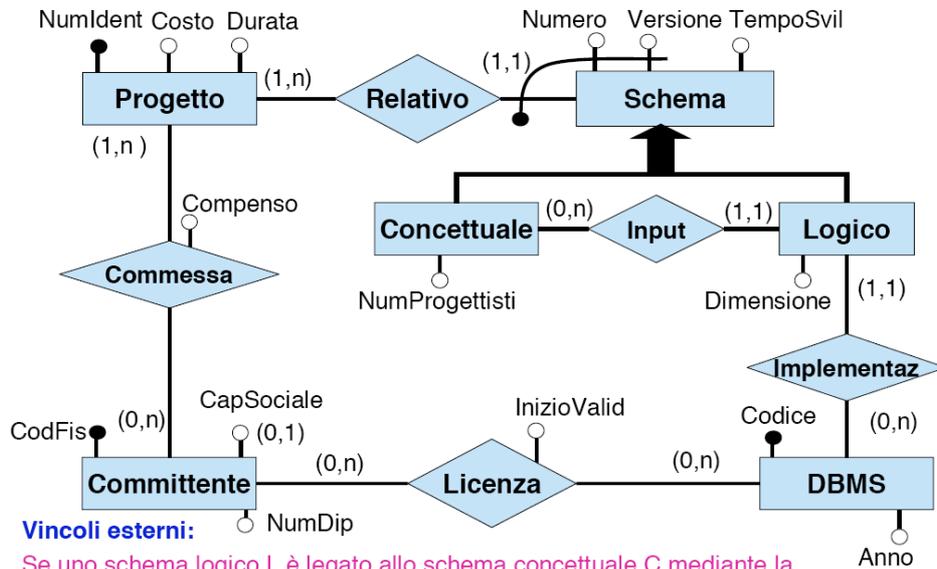
Anno Accademico 2006/07
Canale M-Z

<http://www.dis.uniroma1.it/~degiacomo/didattica/basidati/>

Esercizio d'esame: appello del 19/12/02

Progettare lo schema concettuale Entità-Relazione di un'applicazione relativa ai progetti di basi di dati condotti da una azienda informatica. L'azienda 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, e ogni committente eroga, per ogni progetto, un relativo compenso. Un committente può chiedere uno o più progetti. Per ogni progetto, l'azienda produce uno o più schemi. Ogni schema viene sviluppato per uno ed un solo progetto, ed è caratterizzato da un numero, una versione ed il tempo (in giorni) impiegato per lo sviluppo. La combinazione (numero, versione) di uno schema è unica 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.

Schema ER



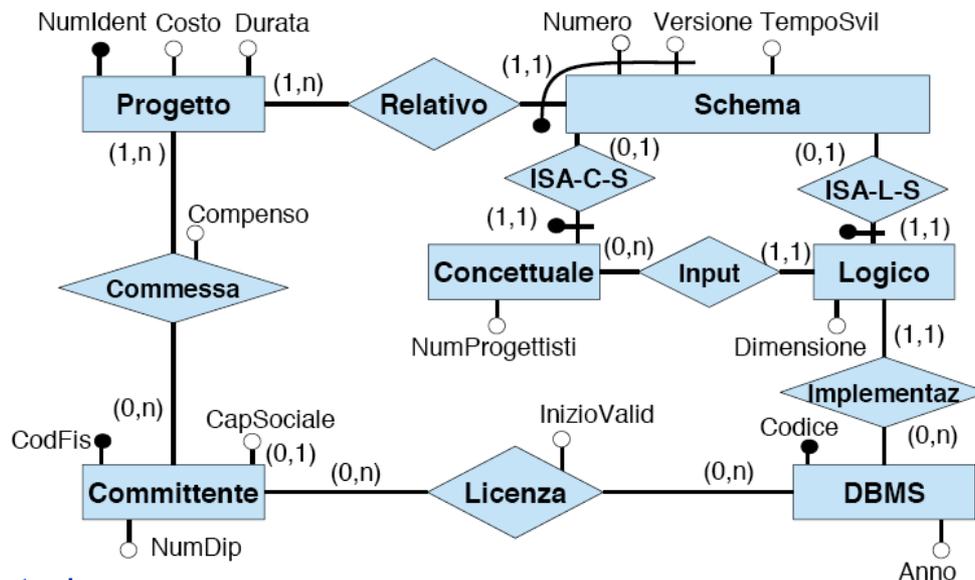
Se uno schema logico L è legato allo schema concettuale C mediante la relazione Input, allora L e C sono legati dalla relazione Relativo allo stesso progetto.

Esercizio d'esame (cont.)

Si progetti lo schema logico dell'applicazione, illustrando lo schema relazionale corrispondente (non necessariamente espresso in SQL), completo di vincoli, così come ottenuto dalla "traduzione diretta".

Ristrutturare lo schema logico ottenuto 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)

Schema ER ristrutturato (degradato)



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.
- **Ogni istanza di Schema partecipa ad ISA-C-S oppure ad ISA-L-S ma non ad entrambe** (dovuto all'eliminazione della generalizzazione).

Schema logico

Progetto(NumIdent, Costo, Durata)

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

inc: Progetto[NumIdent] \subseteq Compressa[Progetto]

Schema(Progetto, Numero, Versione, TempoSvil)

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

Concettuale(Progetto, Numero, Versione, NumProgettisti)

fk: Concettuale[Progetto, Numero, Versione] \subseteq Schema[Progetto, Numero, Versione]

Logico(Progetto, Numero, Versione, Dimensione)

fk: Logico[Progetto, Numero, Versione] \subseteq Schema[NumIdent, Numero, Versione]

fk: Logico[Progetto, Numero, Versione] \subseteq Input[ProgettoLog, NumeroLog, VersioneLog]

fk: Logico[Progetto, Numero, Versione] \subseteq Implementazione[ProgettoLog, NumeroLog, VersioneLog]

Input(ProgettoLog, NumeroLog, VersioneLog, ProgConc, NumConc, VerConc)

fk: Input[ProgettoLog, NumeroLog, VersioneLog] \subseteq Logico[Progetto, Numero, Versione]

fk: Input[ProgConc, NumConc, VerConc] \subseteq Concettuale[Progetto, Numero, Versione]

Implementazione(ProgettoLog, NumeroLog, VersioneLog, CodDBMS)

fk: Implementazione[ProgettoLog, NumeroLog, VersioneLog] \subseteq Logico[Progetto, Numero, Versione]

fk: Implementazione[CodDBMS] \subseteq DBMS[Codice]

DBMS(Codice, Anno)

...

Schema logico (cont.)

...

DBMS(Codice, Anno)

Licenza(DBMS, Committente, InizioValid)

fk: Licenza[Committente] \subseteq Committente[CodFis]

fk: Licenza[DBMS] \subseteq DBMS[Codice]

Committente(CodFis, NumDip, CapSociale*)

Commessa(Committente, Progetto, Compenso)

fk: Commessa[Committente] \subseteq Committente[CodFis]

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

Vincoli esterni:

$\forall P, NL, VL, PC, NC, VC. \text{Input}(P, NL, VL, PC, NC, VC) \Rightarrow PC = PL$

Vincoli per catturare generalizzazione completa:

$\text{Logico}[\text{Progetto}, \text{Numero}, \text{Versione}] \cap \text{Concettuale}[\text{Progetto}, \text{Numero}, \text{Versione}] = \emptyset$

$\text{Schema}[\text{Progetto}, \text{Numero}, \text{Versione}] \subseteq$

$\text{Logico}[\text{Progetto}, \text{Numero}, \text{Versione}] \cup \text{Concettuale}[\text{Progetto}, \text{Numero}, \text{Versione}]$

Schema logico: un commento

Vincoli esterni:

$\forall P, NL, VL, PC, NC, VC. \text{Input}(P, NL, VL, PC, NC, VC) \Rightarrow PC = PL$

create table Input (-- traduzione in SQL letterale

ProgettoLog varchar(30) not null,

NumeroLog varchar(30) not null,

VersioneLog varchar(30) not null,

ProgettoConc varchar(30) not null,

NumeroConc varchar(30) not null,

VersioneConc varchar(30) not null,

primary key (ProgettoLog, NumeroLog, VersioneLog),

foreign key (ProgettoLog, NumeroLog, VersioneLog) references Logico(Progetto, Numero, Versione),

foreign key (ProgettoConc, NumeroConc, VersioneConc) references Concettuale(Progetto, Numero, Versione),

check (ProgettoLog=ProgettoConc)

)

create table Input (-- traduzione in SQL un po' più pensata

Progetto varchar(30) not null,

NumeroLog varchar(30) not null,

VersioneLog varchar(30) not null,

NumeroConc varchar(30) not null,

VersioneConc varchar(30) not null,

primary key (Progetto, NumeroLog, VersioneLog),

foreign key (**Progetto**, NumeroLog, VersioneLog) references Logico(Progetto, Numero, Versione),

foreign key (**Progetto**, NumeroConc, VersioneConc) references Concettuale(Progetto, Numero, Versione)

)

Ristrutturazione schema logico

Per eliminare i valori nulli di **CapSociale** in **Committente**, effettuiamo una **decomposizione mista** di **Committente** ottenendo al posto della vecchia tabella **Committente**: quanto segue (*il resto dello schema logico rimane immutato*):

Committente(CodFis, NumDip)

CommConCS(CodFis, CapSociale)

fk: CommConCS[CodFis] \subseteq Committente[CodFis]