SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ing. dell'Informazione, Informatica e Statistica
Laurea in Informatica

Basi di Dati, Modulo 2

Prof. Toni Mancini, Prof. Federico Mari
Dipartimento di Informatica
http://tmancini.di.uniroma1.it
http://mari.di.uniroma1.it

Slides B.3.1.3 (S.B.3.1.3)

Basi di Dati Relazionali
La Fase di Progettazione
Progettazione della Base Dati
Vincoli di Integrità sullo Schema Relazionale

Versione 2016-06-16

Vincoli di Integrità sullo Schema Relazionale

Oltre a progettare lo schema relazionale della base dati, bisogna decidere come implementare i vincoli esterni in vincoli di integrità supportati dal DBMS in uso.

I vincoli esterni da considerare sono:

- Vincoli esterni al diagamma ER ristrutturato
  (provenienti da vincoli dello schema concettuale, oppure introdotti durante la ristrutturazione)
- Vincoli introdotti durante la produzione dello schema relazionale:
  vincoli di chiave, di foreign key, di inclusione, di ennupla, di dominio).

Vincoli di Integrità sullo Schema Relazionale (2)

I DBMS permettono una facile implementazione delle seguenti tipologie di vincoli di integrità:

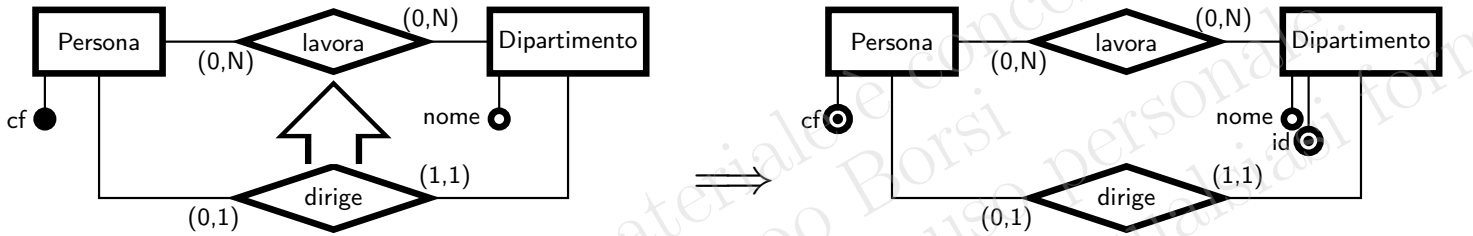
- vincoli di chiave (costrutto **unique**)
- vincoli di chiave primaria (costrutto **primary key**)
- vincoli di foreign key (costrutto **foreign key**)
- vincoli di ennupla e di dominio (costrutto **check**).

I vincoli non traducibili con le tecniche di cui sopra (in particolare i vincoli inter-relazionali non di tipo foreign key) vanno imposti nel DBMS con approcci più complessi.

Esempio

Alcuni vincoli dovuti alla ristrutturazione possono essere espressi agevolmente.

Esempio: ristrutturazione delle is-a tra relationship



Vincolo: $\forall p, d\ \text{dirige}(p, d) \rightarrow \text{lavora}(p, d)$

- Definizione delle seguenti relazioni con vincoli:
  - **Persona** (cf:char(16))
  - **Dipartimento** (id:integer, nome:varchar(100))
    Vincolo inclusione: id $\subseteq$ dirige(dip) $\Longrightarrow$ foreign key!
  - **lavora** (persona:char(16), dip:integer)
    Vincolo foreign key: persona references Persona(cf)
    Vincolo foreign key: dip references Dipartimento(id)
  - **dirige** (persona:char(16), dip:integer)
    Vincolo foreign key: (persona, dip) refer. lavora(persona, dip)
    Vincolo chiave: persona

Vincoli di Integrità: Trigger

Come imporre vincoli di integrità generali sullo Schema Relazionale della base dati?

I DBMS permettono di definire trigger, che possono essere usati per:

1. Intercettare operazioni, come inserimento, cancellazione, modifica di una ennupla in una tabella
2. Eseguire una funzione che ha accesso al più a due argomenti (in base all'evento intercettato):
   - inserimento: la ennupla da inserire (argomento new)
   - cancellazione: ennupla da cancellare (argomento old)
   - modifica: ennupla prima e dopo la modifica (argomenti old e new)

   La funzione può:
   - eseguire comandi SQL arbitrari
   - generare errori che impediscono l'operazione intercettata.

Vincoli di Integrità: Trigger (2)

La progettazione di un vincolo esterno mediante trigger è la seguente:

1. Definire tutti gli eventi (inserimento, cancellazione o modifica di ennuple) che possono portare la base dati a violare il vincolo esterno.
2. Decidere se valutare il vincolo esterno subito prima o subito dopo che l'evento accada.
3. Definire una funzione (sugli argomenti old e/o new) che:
   - valuta il vincolo (usando query SQL) e genera un errore se il livello estensionale della base dati al termine dell'operazione lo viola oppure
   - effettua ulteriori modifiche al livello estensionale della base dati (usando comandi SQL) per soddisfare il vincolo.

Esempio: Disgiunzione

Diagramma ER concettuale

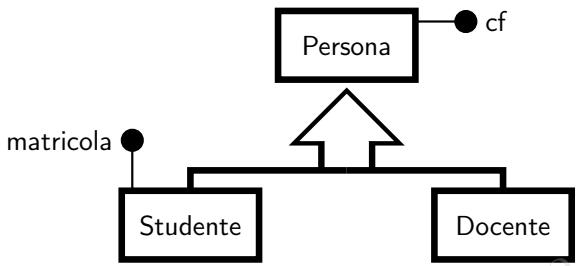
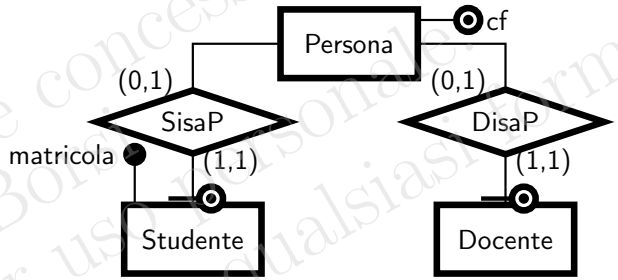


Diagramma ER ristrutturato



V.Persona.isa.disj:

$$\forall p\ \text{Persona}(p) \rightarrow [\ (\exists s\ \text{SisaP}(s,p)) \rightarrow \neg(\exists d\ \text{DisaP}(d,p))\ ]$$

Schema Relazionale

Persona(cf:char(16))

Studente(persona:char(16), matricola:varchar(10))

Vincolo foreign key: persona references Persona(cf)
Vincolo altra chiave: matricola

Docente(persona:char(16))

Vincolo foreign key: docente references Persona(cf)

Esempio: Disgiunzione (2)

Trigger per V.Persona.isa.disj:

- Operazioni: inserimento o modifica in Studente o Docente
- Istante di invocazione: prima dell'operazione intercettata
- Funzione:
  1. Sia isError = **FALSE**;
  2. Sia new l'ennupla che si sta inserendo oppure l'ennupla risultato della modifica;
  3. Se si sta inserendo o modificando una ennupla in Studente:

```
isError := exists (select * from Docente d
where d.persona = new.persona);
```

  4. Altrimenti (inserimento/modifica di una ennupla in Docente):

```
isError := exists (select * from Studente s
where s.persona = new.persona);
```

  5. Se isError = **TRUE** blocca l'operazione;
  6. Altrimenti permetti l'operazione.

Esempio: Dati Ridondanti

Diagramma ER concettuale

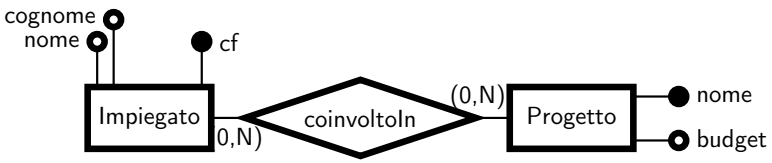
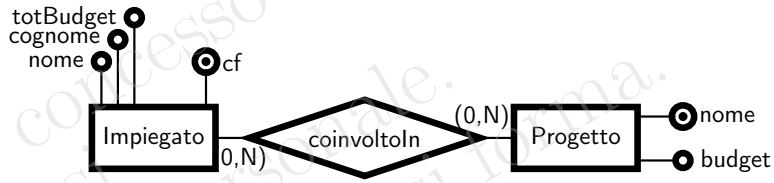


Diagramma ER ristrutturato



V.Impiegato.totBudget:

$$\forall i, t \ \text{Impiegato}(i) \wedge \text{totBudget}(i, t) \rightarrow t = \sum_{(p,b) \in B_i} (b)$$

dove $B_i = \{(p, b) | \text{Progetto}(p) \wedge \text{coinvolto}(i, p) \wedge \text{budget}(p, b)\}$.

Schema Relazionale

Impiegato(cf:char(16), ..., totBudget:PosInteger)

Progetto(nome:varchar(100), budget:PosInteger)

coinvoltoIn(impiegato:char(16), progetto:PosInteger)

Vincolo foreign key: impiegato references Impiegato(cf)
Vincolo foreign key: progetto references Progetto(nome)

Esempio: Dati Ridondanti (2)

Trigger per V.Impiegato.totBudget

- Operazioni:
  - inserimento, modifica, cancellazione in coinvoltoIn
  - modifica in Progetto
- Istante di invocazione: dopo dell'operazione intercettata
- Funzione:
  1. Sia deltaBudget = 0;
  2. Se l'operazione ha agito su coinvoltoIn:
     - 2.1. Se è stata inserita l'ennupla new in coinvoltoIn:

```
deltaBudget := select budget from
    Progetto
where nome = new.progetto;
```

Esempio: Dati Ridondanti (3)

```
update into Impiegato
set totBudget = totBudget + deltaBudget
where cf = new.impiegato

...
```

Esempio: Dati Ridondanti (4)

2.2. Altrimenti, se è stata cancellata l'ennupla old in coinvoltoIn:

```
deltaBudget := select budget
from Progetto
where nome = old.progetto;

update into Impiegato
set totBudget = totBudget - deltaBudget
where cf = old.impiegato
...
```

Esempio: Dati Ridondanti (5)

2.3. Altrimenti, se l'ennupla old in coinvoltoIn è stata modificata in new:

```
deltaBudget := select budget from
    Progetto
where nome = old.progetto;

update into Impiegato
set totBudget = totBudget - deltaBudget
where cf = old.impiegato

deltaBudget := select budget from
    Progetto
where nome = new.progetto;
```

Esempio: Dati Ridondanti (6)

```
update into Impiegato
set totBudget = totBudget + deltaBudget
where cf = new.impiegato
```

Esempio: Dati Ridondanti (7)

3. Se l'ennupla old in Progetto è stata modificata in new:
   3.1. Se new.nome <> old.nome blocca l'operazione;
   3.2. Sia deltaBudget := new.budget − old.budget;
   3.3.
```
update Impiegato
set totBudget = totBudget + deltaBudget
from coinvoltoIn ci
where ci.progetto = new.progetto
  and ci.impiegato = Impiegato.cf
```

Inoltre: togli il diritto di modifica diretta della colonna totBudget di Impiegato a tutti gli utenti.