

## 6c - Recap sui vincoli nel modello relazionale

### Chiave primaria

Il vincolo di chiave primaria è il più importante vincolo di integrità intra-relazionale (all'interno di una singola relazione) che può essere definito nelle basi di dati relazionali. Per capire cosa si intende per tale vincolo, iniziamo dal concetto più generale di *superchiave*.

- Sia  $K$  un sottoinsieme degli attributi di uno schema di relazione  $R$ , avente attributi  $A_1, \dots, A_n$ , vale a dire:  $R(A_1, \dots, A_n)$ .
  - Ad esempio: *automobile(marca, modello, n\_posti, n\_porte, consumo)*, tabella di relazione che memorizza informazioni riguardanti modelli di automobili, dove assumiamo che ciascuna automobile sia identificata dalla sua marca e modello, es., {Fiat, 124 Spider 1400 Lusso}
- $K$  è una **superchiave** di  $R$  se i valori di  $K$  sono sufficienti per identificare in modo univoco una tupla in ogni possibile istanza di relazione di  $R$ . In altre parole, data  $r$  istanza di relazione di  $R$ , non vi possono essere due tuple in  $r$  che assumono gli stessi valori sugli attributi appartenenti a  $K$ . Ogni relazione ha una superchiave naturale, composta da tutti gli attributi della relazione stessa.
  - Ad esempio, una superchiave per la relazione *automobile* è: {marca, modello, n\_posti}
- Una superchiave  $K$  è detta **chiave candidata** se  $K$  è minimale, vale a dire, rimuovendo un ulteriore attributo, si perde la proprietà espressa al punto precedente.
  - Ad esempio, una chiave candidata per la relazione *automobile* è: {marca, modello}
- Una delle chiavi candidate, tipicamente quella con meno attributi, viene scelta come **chiave primaria**.

Il vincolo di chiave primaria, definito su uno o più attributi, comporta in modo automatico l'imposizione di due vincoli su di essi:

- vincolo di univocità (unique): come detto in precedenza, non vi possono essere due tuple nella relazione che assumono gli stessi valori sugli attributi appartenenti alla chiave primaria;
- vincolo di integrità dell'entità: gli attributi che fanno parte della chiave primaria non possono assumere valori nulli (altrimenti si non avrebbero tutte le informazioni necessarie per identificare una tupla).

Queste ultime richieste sono piuttosto naturali, dato il ruolo svolto dalla chiave primaria.

*Cosa accade nel caso in cui in una relazione siano identificabili più chiavi candidate?* Su una di esse viene posto il vincolo di chiave primaria, mentre sulle altre è possibile porre, ad esempio, un vincolo di univocità. In tal modo, è possibile garantire il rispetto delle proprietà legate alle diverse chiavi candidate.

Si osservi, infine, che il concetto di chiave candidata è strettamente legato al dominio (applicazione, ambito di utilizzo) in cui viene sviluppata la base di dati, e alle regole che vogliamo imporre su di esso. Si consideri il seguente esempio:

*albergo(p\_iva, nome, citta, indirizzo, n\_camere, stelle)*

La relazione *albergo* memorizza informazioni riguardanti la locazione e le caratteristiche di un insieme di alberghi. Dal momento che fra gli attributi della relazione è presente *p\_iva*, che registra la partita iva dei vari alberghi, possiamo scegliere in modo piuttosto naturale questo attributo come chiave candidata. Se volessimo imporre che, data una città, non vi possano essere due alberghi con lo stesso nome, un'ulteriore chiave candidata diventerebbe la coppia di attributi {*nome, citta*}. A questo punto, potremmo scegliere di porre come chiave primaria della tabella {*p\_iva*}, e di soddisfare il secondo requisito ponendo un vincolo di tipo unique su {*nome, citta*}.

## Chiave esterna

Il vincolo di chiave esterna è il più importante vincolo di integrità inter-relazionale che può essere definito nelle basi di dati relazionali. Esso identifica una o più colonne di una tabella, detta tabella *referenziante*, che fa riferimento ad una o più colonne di un'altra tabella, detta *referenziata*. I valori delle colonne referenzianti di una tupla devono essere presenti in un'unica tupla della tabella referenziata. Ciò implica che un record nella tabella referenziante non può contenere valori che non esistono nella tabella referenziata (eccetto nel caso particolare di valori NULL nella colonna referenziante). Più tuple della tabella referenziante possono puntare alla stessa tupla della tabella referenziata.

Si osservi che l'insieme degli attributi di una chiave esterna costituisce sempre una chiave candidata nella tabella referenziata (tipicamente, la chiave primaria). Ciò è naturale, in quanto attraverso la chiave esterna si va a fare riferimento ad una *specifica* tupla della tabella referenziata.

Grazie all'uso delle chiavi esterne, è possibile specificare dei legami fra diverse relazioni. Si consideri il seguente esempio:

*dipendente*(*cf*, *nome*, *cognome*, *data\_di\_nascita*, *dipartimento*)  
*dipartimento*(*codice*, *tipologia*, *città*)

Le relazioni memorizzano informazioni riguardanti i dipendenti di un'azienda. Assumiamo che ogni dipendente sia identificato in modo univoco dal suo codice fiscale. Dunque, la chiave primaria della tabella *dipendente* è {*cf*}. Per quanto riguarda i dipartimenti, ciascuno di essi è identificato univocamente dal suo codice. Un dipartimento è di una specifica tipologia (es., vendite, marketing, ricerca, ...) e assumiamo che in una data città possa esserci al più una tipologia di dipartimento (es., non vi possono essere due dipartimenti di ricerca a Milano). Dati tali requisiti, possiamo identificare due chiavi candidate: {*codice*} (che poniamo come chiave primaria) e {*tipologia*, *città*}, insieme di attributi su cui poniamo un vincolo di univocità.

Osserviamo ora l'attributo *dipartimento* all'interno della tabella *dipendente*. In modo piuttosto naturale, per garantire la consistenza dei dati, possiamo richiedere che i dipartimenti che compaiono in tale colonna facciano parte di quelli previsti dalla tabella *dipartimento*. A tal fine, poniamo un vincolo di chiave esterna che va da *dipendente.dipartimento* a *dipartimento.codice*. In tal modo, intuitivamente, abbiamo specificato che ciascun dipendente afferisce ad al più un dipartimento, e che un dipartimento ha, in generale, più dipendenti. In letteratura, si parla in questo caso di legame *uno-a-molti* (o *molti-a-uno*), ad indicare il fatto che una data tupla della prima relazione fa riferimento ad al più una tupla della seconda relazione; viceversa, una tupla della seconda relazione può essere collegata a più tuple della prima.

Un vincolo di chiave esterna può essere specificato su una singola tabella. Si consideri il seguente esempio:

*corso*(*codice*, *titolo*, *descrizione*, *CFU*, *prerequisito*)

Assumiamo che ciascun corso sia identificato in modo univoco da un codice (chiave primaria). Inoltre, un corso può avere al più un corso come prerequisito. Per imporre quest'ultima condizione, possiamo definire un vincolo di chiave esterna dall'attributo *corso.prerequisito* all'attributo *corso.codice*. Focalizziamoci ora sulla frase "*al più un corso come prerequisito*". Questo significa che l'attributo *corso.prerequisito* può assumere valore NULL, quando un corso non ha alcun prerequisito. Se invece vogliamo assumere che ciascun corso abbia esattamente un requisito, andremo a porre un vincolo not-null su *corso.prerequisito*.

## Esempio

Vediamo ora un ultimo esempio. Si consideri il seguente schema di base di dati:

*piatto*(nome, descrizione, nazionalita)  
*ingrediente*(nome, provenienza, piccante?)  
*usato\_in*(ingrediente, provenienza, piatto)

Assumiamo i seguenti fatti riguardanti lo schema appena definito:

- un piatto è identificato univocamente dal suo nome;
- un ingrediente è identificato univocamente dal suo nome e dalla provenienza. Ad esempio, il sale dell'Himalaya è distinto rispetto al sale di Pirano;
- ciascun ingrediente può essere utilizzato in più piatti; inoltre, un dato piatto può essere composto da più ingredienti.

Date tali assunzioni, possiamo definire i seguenti vincoli. Vi sono due chiavi esterne:

1.  $\{usato\_in.ingrediente, usato\_in.provenienza\} \rightarrow \{ingrediente.nome, ingrediente.provenienza\}$
2.  $\{usato\_in.piatto\} \rightarrow \{piatto.nome\}$

La chiave candidata (e primaria) di *piatto* è *nome*. La chiave candidata (e primaria) di *ingrediente* è  $\{nome, provenienza\}$ . Dal momento che un ingrediente può far parte di più piatti ed un piatto può avere più ingredienti, poniamo come chiave primaria di *usato\_in* l'insieme di attributi  $\{ingrediente, provenienza, piatto\}$ . Si osservi che, nel presente caso, in cui il legame fra due relazioni è di tipo *molti-a-molti* (una tupla della prima relazione può essere legata a più tuple della seconda relazione, e viceversa), per rappresentare tale legame è per forza necessario fare uso di una terza tabella (es., *usato\_in* specifica il legame fra *piatto* e *ingrediente*).

Come dovremmo cambiare la chiave primaria di *usato\_in*, volendo imporre che un ingrediente possa essere utilizzato in al più un piatto? In tal caso, in tale tabella, vogliamo che ciascuna coppia di valori di  $\{ingrediente, provenienza\}$  compaia associata ad al più un unico valore di  $\{piatto\}$ . Per garantire ciò, poniamo come chiave primaria  $\{ingrediente, provenienza\}$ . Si osservi che, intuitivamente, ci troviamo ora di fronte ad una situazione uno-a-molti/molti-a-uno, che può essere modellata facendo uso delle seguenti tabelle (non è più necessaria *usato\_in*):

*piatto*(nome, descrizione, nazionalita)  
*ingrediente*(nome, provenienza, piccante?, piatto)

Dove è presente la chiave esterna  $\{ingrediente.piatto\} \rightarrow \{piatto.nome\}$ .