

Esercitazione 4

Temi: ragionamenti vari e query complesse su un database parecchio strutturato.

L'archivio musicale

Proviamo a realizzare un database più complesso. Il database conterrà 5 tabelle variamente collegate fra loro, contenenti dati relativi ad un archivio musicale.

SPECIFICA: vogliamo essere in grado di memorizzare i dati di ciascun *album* nel nostro archivio; ogni album è prodotto da un *produttore* ed interpretato da un *interprete*; ciascun interprete e ciascun produttore sono associati ad una *nazione*. Infine, ogni *album* contiene un insieme di brani musicali (la tabella si chiama *brano*, coerentemente con le altre).

PROGETTAZIONE: strutturiamo il nostro database come segue. Le tabelle sono queste:

- *nazione*(nome)
- *interprete*(nome,*nazione*)
- *produttore*(nome,*indirizzo*,*nazione*)
- *album*(*idalbum*,titolo,*anno*,*produttore*,*prezzo*,interprete)
- *brano*(*titolo*,album,posizione,*durata*)

Al solito, le chiavi sono sottolineate; imponiamo i relativi vincoli di chiave (unicità, non-nullità) su ciascuna tabella. Notate come la progettazione rifletta le esigenze “intuitive” della specifica; ad esempio,

- la chiave della tabella *album* è la coppia {*titolo*,*interprete*}, dal momento che presumibilmente non esistono due album al mondo con lo stesso titolo, interpretati dallo stesso interprete. Notate che possono bensì esistere due album con lo stesso titolo ma di due interpreti differenti (quanti gruppi al mondo hanno pubblicato “The best of” oppure “Greatest hits”...) oppure due album con titoli diversi dello stesso interprete (e ci mancherebbe altro);
- la chiave della tabella *brano* è la coppia {*album*,*posizione*}, dal momento che ogni brano è univocamente individuato data la sua posizione in un certo album. Questo non pregiudica il fatto che possono esserci brani con lo stesso titolo interpretati da autori diversi, naturalmente.

Inoltre, nella tabella *album*, imponiamo che *idalbum* sia univoco e indicizzato, di fatto rendendolo un’ulteriore chiave, indipendente da {*titolo*,*interprete*} – una scelta intelligente consiste nell’usare il tipo dati *contatore* di Access.

Le *relationship* fra tabelle sono le seguenti:

- *nazioni.nome* è in relazione uno-a-molti con *interpreti.nazione*
- *nazioni.nome* è in relazione uno-a-molti con *produttori.nazione*
- *produttori.nome* è in relazione uno-a-molti con *album.produttore*
- *interpreti.nome* è in relazione uno-a-molti con *album.interprete*
- *album.idalbum* è in relazione uno-a-molti con *brano.album*

Imporre i vincoli di integrità referenziale fra tabelle quando è possibile e verificare che, per quanto ci interessa, non vi siano ridondanze e/o anomalie. Anche queste relationship riflettono le esigenze del progettista: le nazioni, sia degli interpreti che dei produttori, sono indicizzate in una tabella a parte; ogni album ha uno ed un solo produttore e uno ed un solo interprete; ed infine, ad ogni album corrispondono uno o più brani.

INTERROGAZIONE: Una volta popolate decentemente le tabelle del database, provate a realizzare le seguenti query (in ordine crescente di difficoltà):

1. (*query su più tabelle*) quali album di artisti italiani sono prodotti da produttori italiani?
2. (*query su più tabelle*) in generale, quali album sono prodotti nel paese dell'artista?
3. (*query a somma*) qual è il valore totale dell'archivio musicale?
4. (*query a somma con raggruppamento*) e per nazionalità dell'interprete? per esempio: quanto vale tutta la musica italiana che c'è nell'archivio?
5. (*prodotto cartesiano*) quali brani appaiono in più di un disco suonati da artisti diversi (cover)?
6. (*prodotto cartesiano*) e quali invece suonati dallo stesso, per esempio perché riappaiono in una compilation?
7. (*due query in cascata*) chi interpreta il brano più lungo dell'archivio?
8. (*tre query in cascata*) e chi l'album più lungo dell'archivio?