

# DATABASE COLLEZIONE MUSICALE

## TESTO

Esempio tratto dal libro di testo.

Un negozio di musica vuole organizzare una base di dati per gestire dischi e cassette musicali allo scopo di rispondere in modo veloce ed efficiente alle richieste dei clienti, in particolare si vuole:

- 1) L'elenco delle registrazioni di un dato formato, acquisito da input, comprendente titolo, etichetta e numero di brani delle registrazioni;
- 2) L'elenco dei brani di durata superiore a 7 minuti completo di titolo del brano, genere, durata, nome dell'artista;
- 3) Il raggruppamento brani per registrazione, visualizzando per ciascuna registrazione il numero di brani e la relativa durata complessiva;
- 4) L'elenco dei brani eseguiti da un artista, il cui nome sia fornito da input, comprensivo di titolo e genere del brano, posizione nella registrazione e identificatore della registrazione;
- 5) L'elenco dei brani contenuti in una registrazione della quale sia noto il titolo comprensivo di titolo e genere del brano e nome dell'artista che esegue lo stesso;
- 6) L'elenco dei brani il cui genere differisca da quello abituale dell'artista, comprensivo di titolo e genere del brano, nome e genere dell'artista che lo esegue.

## ANALISI DEL TESTO

Vista la natura del problema, escludiamo, come ipotesi plausibile, che nel database possano esservi registrazioni "vuote" (cioè prive di brani) e artisti cui non sia associato alcun brano.

Supponiamo inoltre di identificare tramite l'attributo **NomeArtista** una realtà artistica, quella che esegue la musica (singolo, nome o nome d'arte, complesso, orchestra, coro, ecc.), in modo che un brano sia sempre riconducibile ad un'unica entità artista.

## MODELLO E/R



**Nota 1:** scelti gli attributi (vedi *Modello Logico* a pagina seguente), tra essi **NumBraniReg** e **DurataReg** compresi nella tabella **Registrazioni** sono deducibili tramite conteggi e somme dalla tabella **Brani**; a rigore tali campi sarebbero quindi ridondanti ma si rivelano di evidente utilità (*Ridondanza Controllata*).

**Nota 2:** l'unità di misura dell'attributo **Registrazioni.Capacità** è descritta in 2 modi per distinguere tra supporti digitali (byte) e cassette (tempo).

**Nota 3:** giova infine osservare che volendo esprimere gli attributi **DurataBrano** e **DurataReg** (numeri reali) nella forma più tipica *Minuti:Secondi* (sistema sessagesimale) occorrerà operare un'apposita trasformazione. A tale scopo, il linguaggio SQL fornisce apposite *Funzioni* o la possibilità di inserire *Formule* nelle Query (consigliabile ricorrere al *Generatore di Espressioni*).

## MODELLO LOGICO

### TABELLA ARTISTI

ATTRIBUTO	TIPO	LEN	FORMATO	KEY	Ind	DESCRIZIONE
<u>IdArtista</u>	Contatore	-	Intero	PK	si	Gestione automatica
NomeArtista	Testo	30			si	Identificatore univoco
Epoca	Testo	6	Codificato			Allegato Codice Epoca
GenereArtista	Testo	8	Codificato			Allegato Codice Genere
NoteArtista	Memo	-				
Foto	OLE	-				Oggetto OLE

### TABELLA BRANI

ATTRIBUTO	TIPO	LEN	FORMATO	KEY	Ind	DESCRIZIONE
<u>IdBran</u>	Contatore	-	Intero	PK	si	Gestione automatica
TitoloBran	Testo	30			si	
DurataBran	Reale	-				Minuti
GenereBran	Testo	8	Codificato			Allegato Codice Genere
Dimensione	Intero	-				Byte
Posizione	Intero	-				
NoteBran	Memo	-				
<u>IdArtista</u>	Intero	-		FK	si	FK da Tabella Artisti
<u>IdRegistrazione</u>	Intero	-		FK	si	FK da Tabella Registrazioni

### TABELLA REGISTRAZIONI

ATTRIBUTO	TIPO	LEN	FORMATO	KEY	Ind	DESCRIZIONE
<u>IdRegistrazione</u>	Contatore	-	Intero	PK	si	Gestione automatica
TitoloReg	Testo	30			si	
DurataReg	Intero	-				Minuti
TipoReg	Testo	8	Codificato			Allegato Tipo Registrazione
Etichetta	Testo	30				
Capacità	Intero	-				Byte/Tempo
NumBranReg	Intero	-				
DurataReg	Intero	-				Minuti
NoteReg	Memo	-				

## ALLEGATI

### CODICE GENERE

Valido per *Artisti* e *Brani*. Le categorie sono volutamente ampie ed autoesplicative. Si sceglierà in base al genere prevalente dell'artista o al genere più attinente per il brano.

Codici: **Classica - Leggera - Jazz - Opera - Folk - Altro**

### CODICE EPOCA

Valido per *Artisti*. Le categorie sono volutamente ampie ed autoesplicative. Si sceglierà in base all'epoca più significativa per l'artista in caso di dubbio.

Codici: **Antica - 700 - 800 - Nov1 - Nov2 - 2000**

### TIPO REGISTRAZIONE

Valido per *Registrazioni*. Codici: **CD - DVD - Cassetta - Vinile - Altro**

## Quesito 1

Tutti i dati di interesse per la Query, che è *Parametrica*, si trovano nella tabella Registrazioni. Le operazioni da fare e il corrispondente codice SQL sono elencate di seguito.

### $\sigma_P$ Registrazioni

Temp = Selezione di Registrazioni per TipoReg = [Inserire Tipo di Registrazione: ]

### $\Pi_L$ Temp

Proiezione di Temp su TitoloReg, Etichetta, NumBraniReg

```
SELECT    TitoloReg AS Titolo Album, Etichetta, NumBraniReg AS Numero Brani
FROM      Registrazioni
WHERE     TipoReg = [Inserire Tipo di Registrazione: ]
ORDER BY  TitoloReg;
```

## Quesito 2

I dati di interesse per la Query si trovano nella tabella Brani, escluso l'attributo NomeArtista, raggiungibile dalla tabella Brani tramite la FK IdArtista che la collega alla tabella Artisti ove è il nome richiesto.

Le operazioni da fare e il corrispondente codice SQL sono elencate di seguito.

### $\sigma_P$ Brani

Temp1 = Selezione di Brani per DurataBrano > 7

### Temp1<sub>FK</sub> $\bowtie$ Artisti<sub>PK</sub>

Temp2 = Congiunzione di Temp1 e Artisti su IdArtista

### $\Pi_L$ Temp2

Proiezione di Temp2 su TitoloBrano, GenereBrano, DurataBrano, NomeArtista

```
SELECT    TitoloBrano, GenereBrano, DurataBrano, NomeArtista
FROM      Brani, Artisti
WHERE     DurataBrano > 7 AND Brani.IdArtista = Artisti.IdArtista;
```

**Nota 4:** assumendo che nella tabella Artisti la PK **IdArtista** rappresenti il nome dell'artista (eliminando così l'attributo NomeArtista), nella Query si eliminerebbe la Join (in tal caso infatti tutti i dati necessari sarebbero nella tabella Brani, tramite la FK IdArtista). La scelta migliore sarà effettuata in base a considerazioni di efficienza (p.es. si evita la Join ma si rinuncia al campo di tipo Contatore a gestione automatica, invece che un numero occorre immettere un nome anche sulla FK IdArtista con maggiori rischi di incongruenza dei dati). In questa ipotesi la Query si semplificherebbe come segue.

```
SELECT TitoloBrano, GenereBrano, DurataBrano, IdArtista AS NomeArtista
FROM Brani
WHERE DurataBrano > 7;
```

### Versione della Query in Microsoft Access

$\Pi_L$ Temp2	SELECT	TitoloBrano, GenereBrano, DurataBrano, NomeArtista
Temp1 <sub>FK</sub> $\triangleright \triangleleft$ Artisti <sub>PK</sub>	FROM	Registrazioni INNER JOIN Artisti ON Registrazioni.IdArtista = Artisti.IdArtista
$\sigma_P$ Brani	WHERE	DurataBrano > 7;

**Nota 5:** in questa versione viene meglio evidenziata sia la sequenza delle operazioni implicate nella Query che la struttura della Select e l'operazione di Join (in pratica, sarà eseguita prima la Selezione, poi la Join, quindi la Proiezione).

### Quesito 3

Avendo optato per la scelta di comprendere gli attributi **NumBraniReg** e **DurataReg** nella tabella Registrazioni (*Ridondanza Controllata*), i dati di interesse per la Query si trovano tutti nella tabella Registrazioni, pertanto la Query è la seguente.

```
SELECT IdRegistrazione, NumBraniReg AS Brani, DurataReg AS Durata
FROM Registrazioni;
```

Escludendo invece i campi **NumBraniReg** e **DurataReg** dalla tabella Registrazioni occorrerà operare appositi raggruppamenti, conteggi e somme riferendosi alla tabella Brani; in tal caso il quesito si risolverebbe ricorrendo a raggruppamenti (GROUP BY) ed alle Funzioni di Aggregazione SUM e COUNT.

Siccome le Funzioni di Aggregazione e i Raggruppamenti costituiscono estensioni delle operazioni relazionali, nel nostro caso la Query non comporta nessuna sequenza di operazioni relazionali.

Il codice SQL è riportato di seguito.

```
SELECT IdRegistrazione, COUNT(IdBrano) AS Brani, SUM(DurataBrano) AS Durata
FROM Brani
GROUP BY IdRegistrazione;
```

## Quesito 4

I dati di interesse per la Query, che è *Parametrica*, si trovano nelle tabelle Artisti e Brani. Le operazioni da fare e il corrispondente codice SQL sono elencate di seguito.

### $\sigma_P$ Artisti

Temp1 = Selezione di Artisti per NomeArtista = [ Immettere Nome Artista: ]

### $Temp1_{PK} \triangleright \triangleleft Brani_{FK}$

Temp2 = Congiunzione di Temp1 e Brani su IdArtista

### $\Pi_L$ Temp2

Proiezione di Temp2 su TitoloBranò, GenereBranò, IdRegistrazione, Posizione

```
SELECT    TitoloBranò, GenereBranò, IdRegistrazione, Posizione
FROM      Brani, Artisti
WHERE     NomeArtista = [ Immettere Nome Artista: ] AND
          Brani.IdArtista = Artisti.IdArtista;
```

## Quesito 5

I dati di interesse per la Query coinvolgono tutte le tabelle del database. Per maggiore generalità consideriamo la query parametrica in relazione al campo TitoloReg richiesto come noto.

Le operazioni da fare e il corrispondente codice SQL sono elencate di seguito.

### $\sigma_P$ Registrazioni

Temp1 = Selezione di Registrazioni per TitoloReg = [ Immettere Titolo Album: ]

### $Temp1_{PK} \triangleright \triangleleft Brani_{FK}$

Temp2 = Congiunzione di Temp1 e Brani su IdRegistrazione

### $Temp2_{FK} \triangleright \triangleleft Artisti_{PK}$

Temp3 = Congiunzione di Temp2 e Artisti su IdArtista

### $\Pi_L$ Temp3

Proiezione di Temp3 su TitoloBranò, GenereBranò, NomeArtista

```
SELECT    TitoloBranò, GenereBranò, NomeArtista
FROM      Brani, Artisti, Registrazioni
WHERE     TitoloReg = [ Immettere Titolo Album: ] AND
          Registrazioni.Id Registrazione = Brani.Id Registrazione AND
          Brani.IdArtista = Artisti.IdArtista;
```

## Quesito 6

I dati di interesse per la Query coinvolgono le tabelle Artisti e Brani.

Le operazioni da fare e il corrispondente codice SQL sono elencate di seguito.

$\text{Artisti}_{PK} \triangleright \triangleleft \text{Brani}_{FK}$

Temp1 = Congiunzione di Artisti e Brani su IdArtista

$\sigma_P \text{Temp1}$

Temp2 = Selezione di Temp1 GenereArtista  $\triangleleft$  GenereBranò

$\Pi_L \text{Temp2}$

Proiezione di Temp2 su TitoloBranò, GenereBranò, NomeArtista, GenereArtista

```
SELECT TitoloBranò, GenereBranò, NomeArtista, GenereArtista
FROM Brani, Artisti
WHERE GenereArtista  $\triangleleft$  GenereBranò AND
      Brani.IdArtista = Artisti.IdArtista;
```