

Operazioni insiemistiche

1 Operazioni insiemistiche

Il risultato di una query è una relazione, cioè un insieme di tuple. Può essere utile applicare su di esso le operazioni insiemistiche di unione, intersezione e differenza, ma la `SELECT` da sola non permette di farlo. Per questo, SQL prevede dei costrutti ad-hoc:

- `UNION`;
- `INTERSECT`;
- `EXCEPT / MINUS`.

Questi operatori si applicano con la sintassi

- `A UNION B`
- `A INTERSECT B`
- `A EXCEPT B` (oppure `A MINUS B`)

dove `A` e `B` sono espressioni relazionali, che possono essere generate da istruzioni `SELECT`.

Per poter eseguire le operazioni insiemistiche, è necessaria la compatibilità di schema tra `A` e `B`: entrambe le relazioni devono avere lo *stesso numero di attributi* e gli *stessi domini*. Non è invece necessario che gli attributi abbiano gli stessi nomi: la relazione risultato utilizza i nomi degli attributi della relazione `A`.

2 UNION

`A UNION B` esegue l'unione delle due espressioni relazionali `A` e `B`, cioè produce una relazione contenente tutte le tuple di `A` e tutte quelle di `B`.

Di default, `UNION` rimuove le tuple duplicate dal risultato. Si può invece usare la sintassi `UNION ALL` per mantenere i duplicati.

3 INTERSECT

A `INTERSECT B` esegue l'intersezione delle espressioni relazionali A e B, cioè seleziona le tuple che sono presenti in entrambe le relazioni.

L'operazione di intersezione può essere eseguita anche mediante:

- il `join`;
- l'operatore `IN` (che si applica a una sotto-interrogazione).

4 EXCEPT

A `EXCEPT B` (oppure `A MINUS B`) sottrae l'espressione relazionale B all'espressione relazionale A: il risultato è costituito dalle tuple di A che non sono presenti anche in B.

L'operazione di differenza può essere eseguita anche mediante una sotto-interrogazione, usando l'operatore `NOT IN`.

5 Esempi

Determinare i nomi di tutti i genitori:

```
SELECT Madre
FROM Maternita
UNION
SELECT Padre
FROM Paternita;
```

La relazione risultato di quest'interrogazione ha una singola colonna: essa contiene i nomi di tutti i genitori, ma si chiama `Madre`, perché viene preso il nome dell'attributo nel primo operando della `UNION`. Per dare un nome più significativo alla colonna del risultato, si può rinominare l'attributo `Madre` nella prima `SELECT`:

```
SELECT Madre AS Genitore
FROM Maternita
UNION
SELECT Padre
FROM Paternita;
```

Determinare gli anni in cui sono usciti sia film di Tim Burton sia film di Quentin Tarantino:

```

SELECT anno
FROM Film
WHERE regista = 'tim burton'
INTERSECT
SELECT anno
FROM Film
WHERE regista = 'quentin tarantino';
-- oppure, con il join:
SELECT anno
FROM Film F1, Film F2
WHERE F1.anno = F2.anno
      AND F1.regista = 'tim burton'
      AND F2.regista = 'quentin tarantino';

```

Determinare gli anni in cui sono usciti film di Tim Burton ma non film di Quentin Tarantino:

```

SELECT anno
FROM Film
WHERE regista = 'tim burton'
EXCEPT
SELECT anno
FROM Film
WHERE regista = 'quentin tarantino';

```

6 Notazione posizionale

Gli operatori insiemistici operano sugli attributi in base alle loro posizioni nelle tuple, trascurandone i nomi. Di conseguenza, è importante l'ordine con cui gli attributi vengono specificati nella SELECT.

Ad esempio, la query

```

SELECT Padre, Figlio
FROM Paternita
UNION
SELECT Figlio, Madre
FROM Maternita;

```

è sbagliata, perché il risultato conterrà alcune tuple genitore-figlio e altre coppie figlio-genitore. Anche con la ridenominazione, si avrebbe comunque lo stesso effetto:

```
SELECT Padre AS Genitore, Figlio
FROM Paternita
UNION
SELECT Figlio, Madre AS Genitore
FROM Maternita;
```

L'unica soluzione corretta è elencare gli attributi nello stesso ordine in entrambe le SELECT:

```
SELECT Padre, Figlio
FROM Paternita
UNION
SELECT Madre, Figlio
FROM Maternita;
-- oppure, con la ridenominazione:
SELECT Padre AS Genitore, Figlio
FROM Paternita
UNION
SELECT Madre AS Genitore, Figlio -- basta anche solo rinominare Padre
FROM Maternita;
```