



3. Intersezione e differenza con il predicato IN. I predicati ANY, ALL, EXISTS

Costruzione di intersezione e differenza con query annidate

IN e **NOT IN** possono essere utilizzati, sotto opportune condizioni, per costruire le operazioni insiemistiche di **intersezione** e di **differenza** di tabelle. La condizione, oltre a quella ovvia che le tabelle abbiano la medesima struttura, è che entrambe le tabelle abbiano una chiave. Mostriamo con un esempio come si può procedere.

Accanto alla tabella *Impiegati*, si consideri la tabella *NuoviDipendenti* di schema identico:

NuoviDipendenti (ID, Nome, Cognome, Residenza, *Dipartimento*)

Per ragioni organizzative alcuni dipendenti della tabella *NuoviDipendenti* sono stati inseriti in *Impiegati* e altri no. Si desidera sapere chi sono i nuovi assunti presenti nella tabella *Impiegati* e quali non lo sono.

La risposta, avendo a disposizione le operazioni insiemistiche di intersezione e differenza, è semplice.

I dipendenti presenti in entrambe le tabelle sono dati dall'intersezione delle due tabelle:

```
Impiegati INTERSECT NuoviDipendenti;
```

mentre l'elenco dei dipendenti presenti in *NuoviDipendenti*, ma non in *Impiegati*, si ottiene con la seguente operazione di differenza tra le tabelle:

```
NuoviDipendenti EXCEPT Impiegati;
```

L'intersezione *Impiegati INTERSECT NuoviDipendenti* può essere rappresentata con un'interrogazione nidificata in questo modo:

```
SELECT *
FROM Impiegati
WHERE ID IN ( SELECT ID
              FROM Nuovi Dipendenti );
```

mentre la differenza *NuoviDipendenti EXCEPT Impiegati* è costruita con la seguente interrogazione nidificata:

```
SELECT *
FROM NuoviDipendenti
WHERE ID NOT IN ( SELECT ID
                 FROM Impiegati );
```

Si deve osservare che i due comandi *Select* precedenti producono risultati corretti solamente se i due campi *ID* che si corrispondono rappresentano una forma di identificazione universale dell'oggetto identificato (si pensi, per esempio, al Codice Fiscale di una persona): non basta che siano dello stesso tipo.

I predicati ANY, ALL, EXISTS

L'elenco di valori prodotti da una sottoquery può essere confrontato, oltre che con predicati **IN** e **NOT IN**, anche con i seguenti predicati.

• ANY

ANY viene utilizzato in una clausola *Where* in espressioni del tipo: $x > ANY$ *Elenco*.

Il predicato **ANY** è vero se il confronto è vero per almeno uno dei valori dell'elenco. La condizione di ricerca è falsa se la sottoquery restituisce un insieme vuoto oppure se il confronto è falso per ciascuno dei valori restituiti dalla sottoquery.

La seguente interrogazione serve per ottenere le informazioni dei dipendenti del dipartimento *Amministrazione* con lo stipendio superiore a quello di almeno un impiegato del *Marketing*:

```
SELECT Cognome, Nome
FROM Impiegati
WHERE Dipartimento = 'Amm' AND
      Stipendio > ANY ( SELECT Stipendio
                       FROM Impiegati
                       WHERE Dipartimento = 'Mkt');
```

• ALL

ALL viene utilizzato in una clausola *Where* in espressioni del tipo: $x \leq ALL$ *Elenco*.

Il predicato **ALL** restituisce vero se il confronto è vero per ciascuno dei valori in *Elenco*. La condizione di ricerca è falsa se il confronto è falso per almeno uno tra i valori dell'elenco restituito dalla sottoquery.

Sostituendo *All* ad *Any* nella precedente interrogazione, si possono estrarre tutte le righe dei dipendenti del dipartimento *Amministrazione* che hanno lo stipendio maggiore di quello di tutti gli impiegati del *Marketing*.

```
SELECT Cognome, Nome
FROM Impiegati
WHERE Dipartimento = 'Amm' AND
      Stipendio > ALL ( SELECT Stipendio
                       FROM Impiegati
                       WHERE Dipartimento = 'Mkt');
```

Valgono le seguenti equivalenze:

Attributo **IN** (SELECT...) equivale a: *Attributo* = **ANY** (SELECT...)

Attributo **NOT IN** (SELECT...) equivale a: *Attributo* <> **ALL** (SELECT...)

• EXISTS

EXISTS viene utilizzato in una clausola *Where* in espressioni del tipo: *EXISTS* *Tabella*.

Il predicato **EXISTS** controlla se vengono restituite righe dall'esecuzione della sottoquery: la condizione di ricerca è vera se la *Select* nidificata produce una o più righe come risultato, è falsa se la subquery restituisce un insieme vuoto. Il predicato *Exists* può essere negato nella costruzione della condizione di ricerca inserendo la parola *Not* prima di *Exists*.

Per esempio, se si vuole ottenere l'elenco dei dipendenti, con cognome e nome e scostamento dalla media degli stipendi del dipartimento *Marketing*, solo se esistono dipendenti che lavorano in quel dipartimento, si può usare il comando *Select* nel seguente formato:

```
SELECT Cognome, Nome, Stipendio - (SELECT AVG(Stipendio)
                                   FROM Impiegati
                                   WHERE Dipartimento = 'Mkt')
FROM Impiegati
WHERE EXISTS (SELECT *
              FROM Impiegati
              WHERE Dipartimento = 'Mkt');
```

Il predicato *Exists* è il solo che non confronta un valore con uno o più altri valori. Le colonne utilizzate nella sottoquery di una clausola *Exists* sono irrilevanti: quindi, per brevità, comunemente si utilizza la forma *Select ** nella sottoquery.