



Linguaggio SQL: fondamentali

Interrogazioni nidificate



Interrogazioni nidificate

Introduzione

Struttura dell'istruzione SELECT

```
SELECT [DISTINCT] ElencoAttributiDaVisualizzare  
FROM ElencoTabelleDaUtilizzare  
[WHERE CondizioniDiTupla ]  
[GROUP BY ElencoAttributiDiRaggruppamento ]  
[HAVING CondizioniSuAggregati ]  
[ORDER BY ElencoAttributiDiOrdinamento ];
```

Interrogazioni nidificate

- Introduzione
- Operatore IN
- Operatore NOT IN
- Costruttore di tupla
- Operatore EXISTS
- Operatore NOT EXISTS
- Correlazione tra interrogazioni
- Operazione di divisione
- Table functions

- Un'interrogazione nidificata è un'istruzione **SELECT** contenuta all'interno di un'altra interrogazione
 - la nidificazione di interrogazioni permette di suddividere un problema complesso in sottoproblemi più semplici
- È possibile introdurre istruzioni **SELECT**
 - in un predicato nella clausola **WHERE**
 - in un predicato nella clausola **HAVING**
 - nella clausola **FROM**

DB forniture prodotti (1/2)

- P (CodP, NomeP, Colore, Taglia, Magazzino)
- F (CodF, NomeF, NSoci, Sede)
- FP (CodF, CodP, Qta)

DB forniture prodotti (2/2)

P

<u>CodP</u>	<u>NomeP</u>	<u>Colore</u>	<u>Taglia</u>	<u>Magazzino</u>
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

FP

<u>CodF</u>	<u>CodP</u>	<u>Qta</u>
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

F

<u>CodF</u>	<u>NomeF</u>	<u>NSoci</u>	<u>Sede</u>
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

Interrogazioni nidificate (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

F

<u>CodF</u>	<u>NomeF</u>	<u>NSoci</u>	<u>Sede</u>
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

Interrogazioni nidificate (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

FX

<u>CodF</u>	<u>NomeF</u>	<u>NSoci</u>	<u>Sede</u>
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

FY

<u>CodF</u>	<u>NomeF</u>	<u>NSoci</u>	<u>Sede</u>
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

Interrogazioni nidificate (n.1)

- Trovare il codice dei fornitori che hanno sede nella stessa città di F1

- La formulazione mediante interrogazioni nidificate consente di separare il problema in due sottoproblemi
 - sede del fornitore F1
 - codici dei fornitori con la stessa sede

Interrogazioni nidificate (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT Sede  
FROM F  
WHERE CodF='F1'
```

*Sede del
fornitore F1*

Interrogazioni nidificate (n.1)


➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT CodF
FROM F
WHERE Sede = (SELECT Sede
              FROM F
              WHERE CodF='F1');
```

Interrogazioni nidificate (n.1)

- Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT CodF
FROM F
WHERE Sede = (SELECT Sede
              FROM F
              WHERE CodF='F1');
```



- È possibile utilizzare '=' esclusivamente se è noto a priori che il risultato della SELECT nidificata è sempre un solo valore

Formulazione equivalente (n.1)

- Trovare il codice dei fornitori che hanno sede nella stessa città di F1
- È possibile definire una formulazione equivalente con il join

Formulazione equivalente

- La formulazione equivalente con il join è caratterizzata da
- Clausola FROM contenente le tabelle referenziate nelle FROM di tutte le SELECT
 - Opportune condizioni di join nella clausola WHERE
 - Eventuali predicati di selezione aggiunti nella clausola WHERE

Clausola FROM (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT CodF
FROM (F)
WHERE Sede = (SELECT Sede
               FROM (F)
               WHERE CodF='F1');
```

Clausola FROM (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT CodF
FROM (F) ← FX
WHERE Sede = (SELECT Sede
               FROM (F) ← FY
               WHERE CodF='F1');
```

Clausola FROM (n.1)

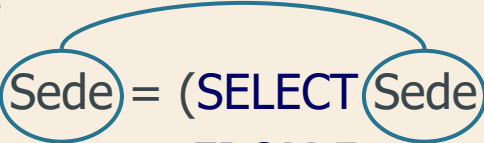
➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT ...  
FROM F AS FX, F AS FY  
...
```

Condizione di join (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT CodF
FROM F
WHERE Sede = (SELECT Sede
               FROM F
               WHERE CodF='F1');
```



Condizione di join (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT ...  
FROM F AS FX, F AS FY  
WHERE FX.Sede=FY.Sede  
...
```


Predicato di selezione (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT CodF
FROM F
WHERE Sede = (SELECT Sede
               FROM F
               WHERE CodF='F1');
```

Predicato di selezione (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT ...  
FROM F AS FX, F AS FY  
WHERE FX.Sede=FY.Sede AND  
       FY.CodF='F1';
```

Clausola SELECT (n.1)

➤ Trovare il codice dei fornitori che hanno sede nella stessa città di F1

```
SELECT FX.CodF
FROM F AS FX, F AS FY
WHERE FX.Sede=FY.Sede AND
      FY.CodF='F1';
```

Interrogazioni nidificate (n.2)

- Trovare il codice dei fornitori il cui numero di soci è minore del numero massimo di soci

Interrogazioni nidificate (n.2)

- Trovare il codice dei fornitori il cui numero di soci è minore del numero massimo di soci

```
SELECT MAX(NSoci)  
FROM F
```

} *Numero
massimo
di soci*

Interrogazioni nidificate (n.2)

- Trovare il codice dei fornitori il cui numero di soci è minore del numero massimo di soci

```
SELECT CodF
FROM F
WHERE NSoci < (SELECT MAX(NSoci)
                FROM F);
```


Formulazione equivalente (n.2)

- Trovare il codice dei fornitori il cui numero di soci è minore del numero massimo di soci

```
SELECT CodF
FROM F
WHERE NSoci < (SELECT MAX(NSoci)
                FROM F);
```

- È possibile definire una formulazione equivalente con il join?

Formulazione equivalente (n.2)

- Trovare il codice dei fornitori il cui numero di soci è minore del numero massimo di soci

```
SELECT CodF
FROM F
WHERE NSoci < (SELECT MAX(NSoci)
                FROM F);
```

- Non è possibile definire una formulazione equivalente con il join



Interrogazioni nidificate

Operatore IN

Operatore IN (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2

Operatore IN (n.1)

- Trovare il nome dei fornitori che forniscono il prodotto P2

- Scomposizione del problema in due sottoproblemi
 - codici dei fornitori del prodotto P2
 - nome dei fornitori aventi quei codici

Operatore IN (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT CodF  
FROM FP  
WHERE CodP='P2'
```

} *Codici
dei
fornitori
di P2*

Operatore IN (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2

FP

<u>CodF</u>	<u>CodP</u>	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400



<u>CodF</u>
F1
F2
F3

```
SELECT CodF  
FROM FP  
WHERE CodP='P2'
```

*Codici
dei
fornitori
di P2*


Operatore IN (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F
WHERE CodF      (SELECT CodF
                  FROM FP
                  WHERE CodP='P2')
```


Operatore IN (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F
WHERE CodF  (SELECT CodF
FROM FP
WHERE CodP='P2')
```

A red arrow points from a red question mark to the red circle, indicating a question about the operator used in the SQL query.

Operatore IN (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2


```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP='P2');
```

Operatore IN (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP='P2');
```

Appartenenza all'insieme



- Esprime il concetto di appartenenza ad un insieme di valori
 - *NomeAttributo* IN (*InterrogazioneNidificata*)

- Permette di scrivere l'interrogazione
 - scomponendo il problema in sottoproblemi
 - seguendo un procedimento "bottom-up"

Formulazione equivalente

- La formulazione equivalente con il join è caratterizzata da
- clausola FROM contenente le tabelle referenziate nelle FROM di tutte le SELECT
 - opportune condizioni di join nella clausola WHERE
 - eventuali predicati di selezione aggiunti nella clausola WHERE

Operatore IN (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP='P2');
```

Formulazione equivalente (n.1)

➤ Trovare il nome dei fornitori che forniscono il prodotto P2

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF
      AND CodP='P2';
```

Operatore IN (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

Operatore IN (n.2)

- Trovare il nome dei fornitori che forniscono almeno un prodotto rosso
- Scomposizione del problema in sottoproblemi
 - codici dei prodotti rossi
 - codici dei fornitori di quei prodotti
 - nomi dei fornitori aventi quei codici

Operatore IN (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

*Codici dei
prodotti rossi*

```
SELECT CodP  
FROM P  
WHERE Colore='Rosso'
```

Operatore IN (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

*Codici dei fornitori
di prodotti rossi*

```
SELECT CodF
FROM FP
WHERE CodP IN (SELECT CodP
                FROM P
                WHERE Colore='Rosso')
```

Operatore IN (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP IN (SELECT CodP
                               FROM P
                               WHERE Colore='Rosso'));
```

Formulazione equivalente (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP IN (SELECT CodP
                               FROM P
                               WHERE Colore='Rosso'));
```

Clausola FROM (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP IN (SELECT CodP
                               FROM P
                               WHERE Colore='Rosso'));
```

Clausola FROM (n.2)


➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT ...  
FROM F, FP, P  
...
```

Condizioni di join (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT NomeF      1
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP IN (SELECT CodP
                               FROM P
                               WHERE Colore='Rosso')));
```



Condizioni di join (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

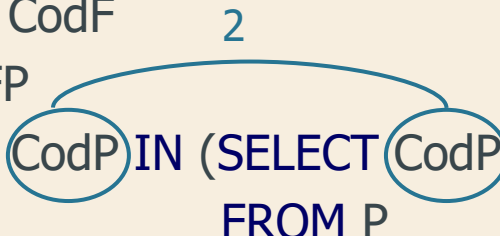
```
SELECT ...  
FROM F, FP, P  
WHERE FP.CodF=F.CodF
```

1

Condizioni di join (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP IN (SELECT CodP
                               FROM P
                               WHERE Colore='Rosso'));
```



Condizioni di join (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT ...  
FROM F, FP, P  
WHERE FP.CodF=F.CodF AND  
      FP.CodP=P.CodP  
...
```

2

Predicato di selezione (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP IN (SELECT CodP
                               FROM P
                               WHERE Colore='Rosso'));
```

Predicato di selezione (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT ...  
FROM F, FP, P  
WHERE FP.CodF=F.CodF AND  
       FP.CodP=P.CodP AND  
       Colore='Rosso'
```

Clausola SELECT (n.2)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto rosso

```
SELECT NomeF
FROM F, FP, P
WHERE FP.CodF=F.CodF AND
      FP.CodP=P.CodP AND
      Colore='Rosso'
```

Esempio complesso (n.3)

- Trovare il nome dei fornitori che forniscono almeno un prodotto fornito da fornitori di prodotti rossi

Esempio complesso (n.3)

- Trovare il nome dei fornitori che forniscono almeno un prodotto fornito da fornitori di prodotti rossi



Esempio complesso (n.3)

- Trovare il nome dei fornitori che forniscono almeno un prodotto fornito da fornitori di prodotti rossi
- La formulazione con il join è difficile
 - è più semplice scomporre il problema in sottoproblemi mediante interrogazioni nidificate

Esempio complesso (n.3)

- Trovare il nome dei fornitori che forniscono almeno un prodotto fornito da fornitori di prodotti rossi

*Codici dei
prodotti rossi*

```
SELECT CodP  
FROM P  
WHERE Colore='Rosso'
```

Esempio complesso (n.3)

- Trovare il nome dei fornitori che forniscono almeno un prodotto fornito da fornitori di prodotti rossi

```
SELECT CodF
FROM FP
WHERE CodP IN
      (SELECT CodP
       FROM P
       WHERE Colore='Rosso')
```

*Codici dei fornitori
di prodotti rossi*

Esempio complesso (n.3)

➤ Trovare il nome dei fornitori che forniscono almeno un prodotto fornito da fornitori di prodotti rossi

*Codici dei prodotti
forniti da fornitori
di prodotti rossi*

```
SELECT CodP
FROM FP
WHERE CodF IN
    (SELECT CodF
     FROM FP
     WHERE CodP IN
         (SELECT CodP
          FROM P
          WHERE Colore='Rosso'))
```

Esempio complesso (n.3)

```
SELECT CodF
FROM FP
WHERE CodP IN
    (SELECT CodP
     FROM FP
     WHERE CodF IN
        (SELECT CodF
         FROM FP
         WHERE CodP IN
            (SELECT CodP
             FROM P
             WHERE Colore='Rosso'))))
```

*Codici dei fornitori
di prodotti forniti
da fornitori di
prodotti rossi*

Interrogazione completa (n.3)

```
SELECT NomeF
FROM F
WHERE CodF IN
    (SELECT CodF
     FROM FP
     WHERE CodP IN
         (SELECT CodP
          FROM FP
          WHERE CodF IN
              (SELECT CodF
               FROM FP
               WHERE CodP IN
                   (SELECT CodP
                    FROM P
                    WHERE Colore='Rosso'))));
```

Formulazione con il join (n.3)



Formulazione con il join (n.3)

```
SELECT NomeF
FROM F
WHERE CodF IN
    (SELECT CodF
     FROM FP
     WHERE CodP IN
         (SELECT CodP
          FROM FP
          WHERE CodF IN
              (SELECT CodF
               FROM FP
               WHERE CodP IN
                   (SELECT CodP
                    FROM P
                    WHERE Colore='Rosso'))));
```


Clausola FROM (n.3)

```
SELECT NomeF
FROM (F)
WHERE CodF IN
      (SELECT CodF
       FROM (FP)
       WHERE CodP IN
            (SELECT CodP
             FROM (FP)
             WHERE CodF IN
                  (SELECT CodF
                   FROM (FP)
                   WHERE CodP IN
                        (SELECT CodP
                         FROM (P)
                         WHERE Colore='Rosso'))));
```

Clausola FROM (n.3)

```
SELECT NomeF
FROM (F)
WHERE CodF IN
  (SELECT CodF
   FROM (FP)
   WHERE CodP IN
     (SELECT CodP
      FROM (FP)
      WHERE CodF IN
        (SELECT CodF
         FROM (FP)
         WHERE CodP IN
           (SELECT CodP
            FROM (P)
            WHERE Colore='Rosso'))));
```

FPA

FPB

FPC

Clausola FROM (n.3)

SELECT ...

FROM F, FP AS FPA, FP AS FPB, FP AS FPC, P

...

Condizioni di join (n.3)

```
SELECT NomeF
FROM F
WHERE CodF IN
  (SELECT CodF
   FROM FP
   WHERE CodP IN
     (SELECT CodP
      FROM FP
      WHERE CodF IN
        (SELECT CodF
         FROM FP
         WHERE CodP IN
           (SELECT CodP
            FROM P
            WHERE Colore='Rosso'))));
```

Diagram illustrating the join conditions in the SQL query:

- The outer query's `CodF IN` condition is circled in blue, with a blue line and the number `1` pointing to the inner query's `CodF`.
- The inner query's `FP` table reference is circled in red, with a red arrow pointing to the label `FPA`.

Condizioni di join (n.3)

```
SELECT ...  
FROM F, FP AS FPA, FP AS FPB, FP AS FPC, P  
WHERE F.CodF=FPA.CodF ①  
...
```

Condizioni di join (n.3)

```
SELECT NomeF  
FROM F  
WHERE CodF IN
```

```
(SELECT CodF  
FROM FP  
WHERE CodP IN  
(SELECT CodP  
FROM FP  
WHERE CodF IN  
(SELECT CodF  
FROM FP  
WHERE CodP IN  
(SELECT CodP  
FROM P  
WHERE Colore='Rosso'))));
```

The diagram illustrates the execution flow of the SQL query. Red arrows labeled 'FPA' and 'FPB' point to the 'FP' tables in the inner and middle subqueries, respectively. A blue arrow labeled '2' points from the 'FP' table in the inner subquery to the 'FP' table in the middle subquery, indicating a join or data flow. Blue circles highlight the 'FP' tables and the 'CodP' fields in the subqueries.

Condizioni di join (n.3)

```
SELECT ...  
FROM F, FP AS FPA, FP AS FPB, FP AS FPC, P  
WHERE F.CodF=FPA.CodF AND  
      FPA.CodP=FPB.CodP  
...
```

2

Condizioni di join (n.3)

```
SELECT NomeF  
FROM F  
WHERE CodF IN
```

```
(SELECT CodF  
FROM FP  
WHERE CodP IN
```

```
WHERE CodP IN
```

```
(SELECT CodP  
FROM FP
```

```
WHERE CodF IN
```

```
(SELECT CodF  
FROM FP
```

```
WHERE CodP IN
```

FPB

3

FPC

```
(SELECT CodP
```

```
FROM P
```

```
WHERE Colore='Rosso'))));
```


Condizioni di join (n.3)

```
SELECT ...  
FROM F, FP AS FPA, FP AS FPB, FP AS FPC, P  
WHERE F.CodF=FPA.CodF AND  
      FPA.CodP=FPB.CodP AND  
      FPB.CodF=FPC.CodF  
...
```

3

Condizioni di join (n.3)

```
SELECT NomeF
FROM F
WHERE CodF IN
  (SELECT CodF
   FROM FP
   WHERE CodP IN
     (SELECT CodP
      FROM FP
      WHERE CodF IN
        (SELECT CodF
         FROM FP
         WHERE CodP IN
           (SELECT CodP
            FROM P
            WHERE Colore='Rosso'))));
```

The diagram illustrates the execution of the nested query. A red arrow points from the label 'FPC' to the 'FP' table in the innermost query. A blue arrow points from the number '4' to the 'FP' table in the next level up. The 'FP' and 'CodP' tables in the innermost query are circled in blue.

Condizioni di join (n.3)

```
SELECT ...  
FROM F, FP AS FPA, FP AS FPB, FP AS FPC, P  
WHERE F.CodF=FPA.CodF AND  
      FPA.CodP=FPB.CodP AND  
      FPB.CodF=FPC.CodF AND  
      FPC.CodP=P.CodP  
...
```

4

Predicato di selezione (n.3)

```
SELECT NomeF
FROM F
WHERE CodF IN
    (SELECT CodF
     FROM FP
     WHERE CodP IN
         (SELECT CodP
          FROM FP
          WHERE CodF IN
              (SELECT CodF
               FROM FP
               WHERE CodP IN
                   (SELECT CodP
                    FROM P
                    WHERE Colore='Rosso'))));
```

Predicato di selezione (n.3)

```
SELECT ...  
FROM F, FP AS FPA, FP AS FPB, FP AS FPC, P  
WHERE F.CodF=FPA.CodF AND  
       FPA.CodP=FPB.CodP AND  
       FPB.CodF=FPC.CodF AND  
       FPC.CodP=P.CodP AND  
       Colore='Rosso'
```

Clausola SELECT (n.3)

```
SELECT NomeF
FROM F, FP AS FPA, FP AS FPB, FP AS FPC, P
WHERE F.CodF=FPA.CodF AND
      FPA.CodP=FPB.CodP AND
      FPB.CodF=FPC.CodF AND
      FPC.CodP=P.CodP AND
      Colore='Rosso';
```



Interrogazioni nidificate

Operatore NOT IN

Concetto di esclusione (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

Concetto di esclusione (n.1)

- Trovare il nome dei fornitori che *non* forniscono il prodotto P2
 - è possibile esprimere l'interrogazione mediante il join?

Concetto di esclusione (n.1)

- Trovare il nome dei fornitori che *non* forniscono il prodotto P2
- è possibile esprimere l'interrogazione mediante il join?

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF ...
```

Concetto di esclusione (n.1)

- Trovare il nome dei fornitori che *non* forniscono il prodotto P2
- è possibile esprimere l'interrogazione mediante il join?

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF
      AND CodP<>'P2';
```

Soluzione errata (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

F

<u>CodF</u>	<u>NomeF</u>	<u>NSoci</u>	<u>Sede</u>
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

FP

<u>CodF</u>	<u>CodP</u>	<u>Qta</u>
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

Soluzione errata (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

F

<u>CodF</u>	NomeF	NSoci	Sede
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF
      AND CodP <> 'P2';
```

FP

<u>CodF</u>	<u>CodP</u>	Qta
<i>F1</i>	<i>P1</i>	<i>300</i>
F1	P2	200
<i>F1</i>	<i>P3</i>	<i>400</i>
<i>F1</i>	<i>P4</i>	<i>200</i>
<i>F1</i>	<i>P5</i>	<i>100</i>
<i>F1</i>	<i>P6</i>	<i>100</i>
<i>F2</i>	<i>P1</i>	<i>300</i>
F2	P2	400
F3	P2	200
<i>F4</i>	<i>P3</i>	<i>200</i>
<i>F4</i>	<i>P4</i>	<i>300</i>
<i>F4</i>	<i>P5</i>	<i>400</i>

Soluzione errata (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

F

<u>CodF</u>	NomeF	NSoci	Sede
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

FP

<u>CodF</u>	<u>CodP</u>	Qta
<i>F1</i>	<i>P1</i>	<i>300</i>
F1	P2	200
<i>F1</i>	<i>P3</i>	<i>400</i>
<i>F1</i>	<i>P4</i>	<i>200</i>
<i>F1</i>	<i>P5</i>	<i>100</i>
<i>F1</i>	<i>P6</i>	<i>100</i>
<i>F2</i>	<i>P1</i>	<i>300</i>
F2	P2	400
F3	P2	200
<i>F4</i>	<i>P3</i>	<i>200</i>
<i>F4</i>	<i>P4</i>	<i>300</i>
<i>F4</i>	<i>P5</i>	<i>400</i>

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF
AND CodP<>'P2';
```

R



NomeF
Andrea
Luca
Gabriele

Soluzione errata (n.1)

- Trovare il nome dei fornitori che *non* forniscono il prodotto P2
- non è possibile esprimere l'interrogazione mediante il join

```
SELECT NomeF  
FROM F, FP  
WHERE F.CodF=FP.CodF  
AND CodP<>'P2';
```

Soluzione errata (n.1)

```
SELECT NomeF
FROM F, FP
WHERE F.CodF=FP.CodF
      AND CodP<> 'P2';
```

➤ A che interrogazione corrisponde?

Soluzione errata (n.1)

```
SELECT NomeF  
FROM F, FP  
WHERE F.CodF=FP.CodF  
      AND CodP<> 'P2';
```



Trovare il nome dei fornitori che forniscono almeno un prodotto diverso da P2

Concetto di esclusione (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

Concetto di esclusione (n.1)

- Trovare il nome dei fornitori che *non* forniscono il prodotto P2

- Occorre escludere dal risultato
 - i fornitori che forniscono il prodotto P2

Concetto di esclusione (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

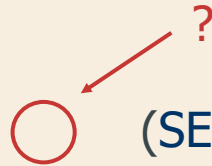
```
SELECT CodF  
FROM FP  
WHERE CodP='P2'
```

*Codici dei fornitori
che forniscono P2*

Concetto di esclusione (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

```
SELECT NomeF  
FROM F  
WHERE CodF
```



```
(SELECT CodF  
FROM FP  
WHERE CodP='P2');
```

*Codici dei fornitori
che forniscono P2*

Operatore NOT IN (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

```
SELECT NomeF
```

```
FROM F
```

```
WHERE CodF NOT IN (SELECT CodF  
                    FROM FP  
                    WHERE CodP='P2');
```

*Codici dei fornitori
che forniscono P2*

Operatore NOT IN (n.1)

➤ Trovare il nome dei fornitori che *non* forniscono il prodotto P2

```
SELECT NomeF  
FROM F
```

```
WHERE CodF NOT IN (SELECT CodF  
FROM FP  
WHERE CodP='P2');
```

Non appartiene

*Codici dei fornitori
che forniscono P2*

Operatore NOT IN

- Esprime il concetto di esclusione da un insieme di valori
 - *NomeAttributo* NOT IN (*InterrogazioneNidificata*)

- Richiede di individuare in modo appropriato *l'insieme da escludere*
 - definito dall'interrogazione nidificata

Operatore NOT IN (n.2)

➤ Trovare il nome dei fornitori che forniscono *solo* il prodotto P2

Operatore NOT IN (n.2)

➤ Trovare il nome dei fornitori che forniscono *solo* il prodotto P2



Trovare il nome dei fornitori di P2 che non hanno mai fornito prodotti diversi da P2

Operatore NOT IN (n.2)

➤ Trovare il nome dei fornitori che forniscono *solo* il prodotto P2



Trovare il nome dei fornitori di P2 che non hanno mai fornito prodotti diversi da P2

- Insieme da escludere
- fornitori di prodotti diversi da P2

Operatore NOT IN (n.2)

➤ Trovare il nome dei fornitori che forniscono solo il prodotto P2

```
SELECT CodF  
FROM FP  
WHERE CodP <> 'P2'
```

*Codici dei fornitori
che forniscono
almeno un
prodotto diverso
da P2*

Operatore NOT IN (n.2)

- Trovare il nome dei fornitori che forniscono solo il prodotto P2

```
SELECT NomeF
FROM F
WHERE CodF NOT IN (SELECT CodF
                   FROM FP
                   WHERE CodP<>'P2')
```

...

Soluzione alternativa (n.2)

➤ Trovare il nome dei fornitori che forniscono solo il prodotto P2

```
SELECT NomeF
FROM F
WHERE CodF NOT IN (SELECT CodF
                   FROM FP
                   WHERE CodP<>'P2')
AND CodF IN (SELECT CodF
             FROM FP);
```

Operatore NOT IN (n.2)

➤ Trovare il nome dei fornitori che forniscono solo il prodotto P2

```
SELECT NomeF
FROM F, FP
WHERE F.CodF NOT IN (SELECT CodF
                     FROM FP
                     WHERE CodP<>'P2')
AND F.CodF=FP.CodF;
```

Operatore NOT IN (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

Operatore NOT IN (n.3)

- Trovare il nome dei fornitori che *non* forniscono prodotti rossi
- Insieme da escludere?

Operatore NOT IN (n.3)

- Trovare il nome dei fornitori che *non* forniscono prodotti rossi
- Insieme da escludere?
 - i fornitori di prodotti rossi, identificati dal loro codice

Operatore NOT IN (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

*Codici dei fornitori
di prodotti rossi*

```
(SELECT CodF  
FROM FP  
WHERE CodP IN (SELECT CodP  
FROM P  
WHERE Colore='Rosso'))
```

Operatore NOT IN (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

```
SELECT NomeF
FROM F
WHERE CodF NOT IN (SELECT CodF
                    FROM FP
                    WHERE CodP IN (SELECT CodP
                                   FROM P
                                   WHERE Colore='Rosso'));
```

Alternativa (corretta?) (n.3)

- Trovare il nome dei fornitori che *non* forniscono prodotti rossi

Alternativa (corretta?) (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP NOT IN (SELECT CodP
                                    FROM P
                                    WHERE Colore='Rosso'));
```

Alternativa (corretta?) (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

*Codici dei
prodotti
rossi*

```
SELECT CodP  
FROM P  
WHERE Colore='Rosso'
```

Alternativa (corretta?) (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

*Codici dei fornitori
che forniscono
almeno un
prodotto non rosso*

```
SELECT CodF
FROM FP
WHERE CodP NOT IN (SELECT CodP
                    FROM P
                    WHERE Colore='Rosso')
```


Alternativa errata (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

```
SELECT NomeF  
FROM F  
WHERE CodF IN (SELECT CodF  
                FROM FP  
                WHERE CodP NOT IN (SELECT CodP  
                                    FROM P  
                                    WHERE Colore='Rosso')));
```

Alternativa errata (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

```
SELECT NomeF
FROM F
WHERE CodF IN ((SELECT CodF
                FROM FP
                WHERE CodP NOT IN (SELECT CodP
                                   FROM P
                                   WHERE Colore='Rosso')));
```

*Codici dei fornitori
di prodotti
non rossi*

Alternativa errata (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

P

CodP	NomeP	Colore	Taglia	Magazzino
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

F

CodF	NomeF	NSoci	Sede
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

FP

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

Alternativa errata (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

P

CodP	NomeP	Colore	Taglia	Magazzino
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

F

CodF	NomeF	NSoci	Sede
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

FP

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

Alternativa errata (n.3)

➤ Trovare il nome dei fornitori che *non* forniscono prodotti rossi

```
SELECT NomeF
FROM F
WHERE CodF IN (SELECT CodF
                FROM FP
                WHERE CodP NOT IN (SELECT CodP
                                    FROM P
                                    WHERE Colore='Rosso'));
```

➤ L'insieme di elementi da escludere non è corretto



Interrogazioni nidificate

Costruttore di tupla

Costruttore di tupla

- Permette di definire la struttura temporanea di una tupla
 - si elencano gli attributi che ne fanno parte tra ()

(NomeAttributo₁, NomeAttributo₂, ...)

- Permette di estendere il poter espressivo degli operatori **IN** e **NOT IN**

Esempio (n.1)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,
OraPartenza, OraArrivo)

- Trovare le coppie luogo di partenza e luogo di arrivo per cui nessun viaggio dura più di 2 ore

Esempio (n.1)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,
OraPartenza, OraArrivo)

➤ Trovare le coppie luogo di partenza e luogo di arrivo per cui nessun viaggio dura più di 2 ore

```
(SELECT LuogoPartenza, LuogoArrivo  
FROM VIAGGIO  
WHERE OraArrivo-OraPartenza>2)
```

*Percorsi per
cui esistono
viaggi che
durano
più di 2 ore*

Esempio (n.1)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,
OraPartenza, OraArrivo)

➤ Trovare le coppie luogo di partenza e luogo di arrivo per cui nessun viaggio dura più di 2 ore

```
SELECT LuogoPartenza, LuogoArrivo
FROM VIAGGIO
WHERE (LuogoPartenza, LuogoArrivo) NOT IN
      (SELECT LuogoPartenza, LuogoArrivo
       FROM VIAGGIO
       WHERE OraArrivo-OraPartenza>2);
```

Esempio (n.1)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,
OraPartenza, OraArrivo)

➤ Trovare le coppie luogo di partenza e luogo di arrivo per cui nessun viaggio dura più di 2 ore

```
SELECT LuogoPartenza, LuogoArrivo
FROM VIAGGIO
WHERE (LuogoPartenza, LuogoArrivo) NOT IN
      (SELECT LuogoPartenza, LuogoArrivo
       FROM VIAGGIO
       WHERE OraArrivo-OraPartenza>2);
```

*Costruttore
di tupla*