

SUBQUERY nella clausola WHERE**subQuery con operatori di confronto: = < <= > >= <> !=**

Individuare i clienti che hanno effettuato il pagamento più alto	<pre>SELECT customerNumber, checkNumber, amount FROM payments WHERE amount = (SELECT MAX(amount) FROM payments)</pre>	1
Elencare i clienti che hanno effettuato pagamenti superiori al valore medio	<pre>SELECT customerNumber, checkNumber, amount FROM payments WHERE amount > (SELECT AVG(amount) FROM payments) ORDER BY amount DESC</pre>	134

subQuery con operatori: IN NOT IN

Individuare i dipendenti che coordinano altri dipendenti

<pre>SELECT * FROM employees WHERE employeeNumber IN (SELECT DISTINCT reportsTo FROM employees)</pre>	6 su 23 dipendenti
---	--------------------

Individuare i dipendenti che **non** coordinano altri dipendenti

<pre>SELECT * FROM employees WHERE employeeNumber NOT IN (SELECT DISTINCT reportsTo FROM employees WHERE reportsTo IS NOT NULL)</pre>	17 su 23 dipendenti
---	---------------------

OPPURE con 2 subQuery annidate

<pre>SELECT * FROM employees WHERE employeeNumber NOT IN (SELECT employeeNumber FROM employees WHERE employeeNumber IN (SELECT DISTINCT reportsTo FROM employees))</pre>	17
--	----

NOTA: la **query interna** senza clausola WHERE estrae anche il valore NULL presente nella riga dell'unico dipendente NON coordinato: 1002 – Murphy - President (i valori estratti sono 7: NULL, 1002, 1056, 1088, 1102, 1143, 1621); per estrarre i dipendenti non coordinatori è quindi necessario escludere il valore NULL perchè nessun dipendente può soddisfare la condizione:

employeeNumber NOT IN (NULL, 1002, ...) che equivale a:

employeeNumber != NULL AND employeeNumber != 1002 AND ... (con 1^condizione sempre falsa! – cfr. p.198)

con le **2 subQuery annidate**, la query intermedia estrae solo gli **employeeNumber** dei coordinatori escludendo il valore NULL:

employeeNumber IN (NULL, 1002, ...) che equivale a:

employeeNumber = NULL OR employeeNumber = 1002 OR ... (la 1^condizione è sempre falsa ma le altre a volte vere)

Altri esempi:

Individuare i clienti che hanno effettuato ordini

```

SELECT customerNumber, customerName
FROM customers
WHERE customerNumber IN (SELECT DISTINCT customerNumber
                                FROM orders) -- clienti CON ordini

```

98 su 122
clientiIndividuare i clienti che **non** hanno effettuato ordini

```

SELECT customerNumber, customerName
FROM customers
WHERE customerNumber NOT IN (SELECT DISTINCT customerNumber
                                FROM orders) -- clienti SENZA ordini

```

24 su 122
clienti**soluzioni alternative SENZA subQuery:**

```

SELECT DISTINCT customerNumber, customerName
FROM customers
      INNER JOIN orders USING (customerNumber)
-- clienti CON ordini

```

98

```

SELECT customerNumber, customerName
FROM customers
      INNER JOIN orders USING (customerNumber)
GROUP BY customerNumber
-- clienti CON ordini

```

98

```

SELECT customerNumber, customerName
FROM customers
      LEFT JOIN orders USING (customerNumber)
WHERE orderNumber IS NULL -- clienti SENZA ordini

```

24

```

SELECT customerNumber, customerName
FROM customers
      LEFT JOIN orders USING (customerNumber)
GROUP BY customerNumber
HAVING COUNT(orderNumber) = 0 -- clienti SENZA ordini

```

24

SUBQUERY nella clausola FROM

Individuare il numero massimo e medio di dettagli inseriti in un ordine

```
SELECT MAX(numDettOrdine) AS numMAXdett,
       FLOOR(AVG(numDettOrdine)) AS numMEDdett
FROM (SELECT orderNumber,
            COUNT(orderLineNumber) AS numDettOrdine
      FROM orderdetails
      GROUP BY orderNumber) AS stats_orderdetails
```

numMAXdett	numMEDdett
18	9

Individuare i dettagli d'ordine con numero linea d'ordine uguale al valore medio di dettagli inseriti

```
SELECT * FROM orderdetails
WHERE orderLineNumber =
      (SELECT FLOOR(AVG(numDettOrdine)) AS numMedDett
      FROM (SELECT orderNumber,
                  COUNT(orderLineNumber) AS numDettOrdine
            FROM orderdetails
            GROUP BY orderNumber) AS stats_orderdetails
      )
```

177

NOTA: la prima *subQuery* è posta nella clausola **WHERE** mentre la *subQuery* più interna è posta nella clausola **FROM**
In modo analogo si procede per elencare i dettagli d'ordine con numero linea uguale al valore massimo di dettagli (trova 11 righe)

Determinare il numero massimo di volte che un prodotto è presente nei dettagli d'ordine

```
SELECT MAX(numOrdini) AS maxNumProdOrdini
FROM (SELECT productCode,
            COUNT(orderNumber) AS numOrdini
      FROM orderdetails
      GROUP BY productCode) AS ordiniXprodotto
```

maxNumProdOrdini
53

Individuare il prodotto ordinato più volte (presente nel maggior numero di ordini)

```
SELECT productCode, productName, productLine, COUNT(orderNumber) AS numOrdProd
FROM products INNER JOIN orderdetails USING (productCode)
GROUP BY productCode
HAVING numOrdProd = (SELECT MAX(numOrdini) AS maxNumProdOrdini
                   FROM (SELECT productCode,
                               COUNT(orderNumber) AS numOrdini
                         FROM orderdetails
                         GROUP BY productCode) AS ordiniXprodotto
                   )
```

productCode	productName	productLine	numOrdProd
S18_3232	1992 Ferrari 360 Spider red	Classic Cars	53

NOTA: la prima *subQuery* è posta nella clausola **HAVING** mentre la *subQuery* più interna è posta nella clausola **FROM**

Esporre per ogni cliente gli importi degli ordini evasi (status = 'shipped')

```
SELECT country, city, customerNumber, customerName, orderNumber,
       orderDate, SUM(quantityOrdered * priceEach) AS importo
FROM customers INNER JOIN orders USING (customerNumber)
      INNER JOIN orderdetails USING (orderNumber)
WHERE status = 'shipped'
GROUP BY country, city, customerNumber, orderNumber
ORDER BY city, importo
```

Q.1)

303 righe

	country	city	customerNumber	customerName	orderNumber	orderDate	importo
▶	USA	Allentown	157	Diecast Classics Inc.	10272	2004-07-20	23715.70
	USA	Allentown	157	Diecast Classics Inc.	10318	2004-11-02	35152.12
	USA	Allentown	157	Diecast Classics Inc.	10281	2004-08-19	39641.43
	New Zealand	Auckland	323	Down Under Souvenirs, Inc	10132	2003-06-25	2880.00
	New Zealand	Auckland	357	GiftsForHim.com	10202	2003-12-02	20220.04
	New Zealand	Auckland	496	Kelly's Gift Shop	10399	2005-04-01	30253.75

Determinare per ogni cliente gli ordini evasi di massimo importo

(la query precedente può essere utilizzata come query interna della seguente subQuery)

```
SELECT country, city, customerNumber, customerName, MAX(importo) AS impMAX
FROM ( SELECT country, city, customerNumber, customerName, orderNumber,
            orderDate, SUM(quantityOrdered * priceEach) AS importo
      FROM customers INNER JOIN orders USING (customerNumber)
            INNER JOIN orderdetails USING (orderNumber)
      WHERE status = 'shipped'
      GROUP BY country, city, customerNumber, orderNumber
    ) AS cust_details
GROUP BY country, city, customerNumber
ORDER BY city, impMAX
```

Q.2)

98 righe

	country	city	customerNumber	customerName	impMAX
▶	USA	Allentown	157	Diecast Classics Inc.	39641.43
	New Zealand	Auckland	357	GiftsForHim.com	36442.34
	New Zealand	Auckland	323	Down Under Souvenirs, Inc	41426.81

Determinare per l'anno 2004 e per ogni paese (country)

l'importo totale degli ordini evasi e numero di clienti e di ordini

(la prima query può essere utilizzata come query interna anche per la seguente subQuery)

```
SELECT country, YEAR(orderDate) AS anno, COUNT(customerNumber) AS numCli,
       COUNT(orderNumber) AS ordini, SUM(importo) AS impTOT
FROM ( SELECT country, city, customerNumber, customerName, orderNumber,
            orderDate, SUM(quantityOrdered * priceEach) AS importo
      FROM customers INNER JOIN orders USING (customerNumber)
            INNER JOIN orderdetails USING (orderNumber)
      WHERE status = 'shipped'
      GROUP BY country, city, customerNumber, orderNumber
    ) AS cust_details
WHERE YEAR(orderDate) = 2004
GROUP BY country, anno
ORDER BY impTOT DESC
```

Q.3)

20 righe

	country	anno	numCli	ordini	impTOT
▶	USA	2004	52	52	1485054.44
	France	2004	19	19	506660.01
	Spain	2004	13	13	392816.48
	Australia	2004	6	6	204213.18
	New Zealand	2004	5	5	195592.89
