

**INNER JOIN** tra **employees** e **offices** su Foreign Key **employees.officeCode**

(FK derivata dalla associazione **impiegare** del modello Concettuale: "Ogni dipendente **DEVE** essere *impiegato* in **UN SOLO** ufficio")

Per i dipendenti **manager** esporre numero, nome, cognome, mansione (jobTitle) e paese, città e codice dell'ufficio in cui lavorano.

-- forma sconsigliata - SQL	-- forma CONSIGLIATA – SQL2	-- forma ammessa in MySQL
<pre> SELECT   O.officeCode FROM   employees E , offices O WHERE E.officeCode = O.officeCode -- condizione di join AND jobTitle LIKE '%manager%' -- condizione di selezione </pre>	<pre> SELECT   employeeNumber, lastName,   firstName, jobTitle,   country, city,   O.officeCode FROM   employees E INNER JOIN offices O ON E.officeCode = O.officeCode -- condizione di join WHERE jobTitle LIKE '%manager%' -- condizione di selezione </pre>	<pre> SELECT   officeCode FROM   employees INNER JOIN offices USING ( officeCode ) -- condizione di join WHERE jobTitle LIKE '%manager%' -- condizione di selezione </pre>

Solo **3 dipendenti** soddisfano la condizione **jobTitle LIKE '%manager%'**; per ognuno vengono estratte le informazioni sul proprio ufficio.

employeeNumber	lastName	firstName	jobTitle	country	city	officeCode
1088	Patterson	William	Sales Manager (APAC)	Australia	Sydney	6
1102	Bondur	Gerard	Sale Manager (EMEA)	France	Paris	4
1143	Bow	Anthony	Sales Manager (NA)	USA	San Francisco	1

NOTA: senza la **condizione di selezione** **jobTitle LIKE '%manager%'** vengono estratte 23 righe ovvero le informazioni per **tutti** i dipendenti; l'associazione **impiegare** è infatti obbligatoria: "Ogni dipendente **DEVE** essere *impiegato* in **UN SOLO** ufficio" (**employees.officeCode <> NULL**)

**Self JOIN** tra **employees** e **employees** su Foreign Key **employees.reportsTo**

(FK derivata dalla associazione ricorsiva **coordinare** del modello Concettuale: "Ogni dipendente **PUO'** essere *coordinato* da **UN SOLO** dipendente")

Esporre per ogni dipendente il suo capo, selezionando come capi solo i Sales Manager APAC, EMEA e NA (employeeNumber 1088, 1102, 1143 rispettivamente); per dipendente e capo esporre solo le colonne numero, nome, cognome e mansione (jobTitle).

-- forma sconsigliata - SQL	-- forma CONSIGLIATA - SQL2
<pre> SELECT      E.employeeNumber  AS nDipendente,             E.lastName, E.firstName, E.jobTitle,             Capo.employeeNumber AS nCapo,             Capo.lastName, Capo.firstName,             Capo.jobTitle      AS titoloCapo  FROM        employees E             , employees Capo WHERE      E.reportsTo = Capo.employeeNumber -- cond. join AND       Capo.employeeNumber IN (1088, 1102, 1143)             -- condizione di selezione ORDER BY  nCapo, nDipendente; </pre>	<pre> FROM        employees E             INNER JOIN employees Capo ON          E.reportsTo = Capo.employeeNumber -- cond. join WHERE      Capo.employeeNumber IN (1088, 1102, 1143)             -- condizione di selezione ORDER BY  nCapo, nDipendente; </pre>

Sono **estratte 15 righe**:

nDipendente	lastName	firstName	jobTitle	nCapo	lastName	firstName	titoloCapo
1611	Fixter	Andy	Sales Rep	1088	Patterson	William	Sales Manager (APAC)
1612	Marsh	Peter	Sales Rep	1088	Patterson	William	Sales Manager (APAC)
1619	King	Tom	Sales Rep	1088	Patterson	William	Sales Manager (APAC)
1337	Bondur	Loui	Sales Rep	1102	Bondur	Gerard	Sale Manager (EMEA)
...	...	...	...	...	...	...	...

NOTA: per la **self** join è necessario l'uso degli ALIAS altrimenti su ogni colonna si avrebbe ambiguità; inoltre non è possibile usare la forma semplificata USING ammessa da MySQL perché le colonne della condizione di join hanno nomi diversi: **E.reportsTo** e **Capo.employeeNumber**.

**LEFT JOIN** tra **employees** e **employees** su Foreign Key **employees.reportsTo**

Esporre per ogni dipendente il suo capo, estraendo anche il Presidente (mediante **LEFT JOIN**).

<b>LEFT JOIN</b>	<b>INNER JOIN</b>
------------------	-------------------

```

SELECT  E.employeeNumber AS nDipendente,
        E.lastName, E.firstName, E.jobTitle,
        Capo.employeeNumber AS nCapo,
        Capo.lastName, Capo.firstName,
        Capo.jobTitle AS titoloCapo
FROM    employees E

```

<b>LEFT JOIN</b> <b>employees Capo</b>	<b>INNER JOIN</b> <b>employees Capo</b>
--	---

```

ON      E.reportsTo = Capo.employeeNumber
ORDER BY nCapo, nDipendente;

```

**Risultato con la INNER JOIN: 22 righe estratte**

Sono estratte solo 22 righe →  
perché il Presidente non ha un capo:  
per il dipendente 1002,  
la colonna **reportsTo** è a NULL  
quindi non soddisfa la condizione di join.

nDipendente	lastName	firstName	jobTitle	nCapo	lastName	firstName	titoloCapo
1056	Patterson	Mary	VP Sales	1002	Murphy	Diane	President
1076	Firrelli	Jeff	VP Marketing	1002	Murphy	Diane	President
1088	Patterson	William	Sales Manager (APAC)	1056	Patterson	Mary	VP Sales
1102	Bondur	Gerard	Sale Manager (EMEA)	1056	Patterson	Mary	VP Sales
...	...	...	...	...	...	...	...

**Risultato con la LEFT JOIN: 23 righe estratte**

nDipendente	lastName	firstName	jobTitle	nCapo	lastName	firstName	titoloCapo
1002	Murphy	Diane	President	NULL	NULL	NULL	NULL
1056	Patterson	Mary	VP Sales	1002	Murphy	Diane	President
1076	Firrelli	Jeff	VP Marketing	1002	Murphy	Diane	President
1088	Patterson	William	Sales Manager (APAC)	1056	Patterson	Mary	VP Sales
1102	Bondur	Gerard	Sale Manager (EMEA)	1056	Patterson	Mary	VP Sales
...	...	...	...	...	...	...	...

← Viene estratto anche il Presidente  
con le informazioni relative al capo tutte a NULL

**LEFT JOIN e RIGHT JOIN tra customers e employees su Foreign Key customers.salesRepEmployeeNumber**

(FK derivata dalla associazione **seguire** del modello Concettuale: "Ogni cliente **PUO'** essere seguito da **UN SOLO** dipendente")

<p><b>LEFT JOIN (122 righe)</b> Sono esposti <b>tutti i clienti</b> anche quelli senza impiegato di riferimento a seguirli (<code>salesRepEmployeeNumber</code> a NULL)</p>	<p><b>INNER JOIN (100 righe)</b> Per ogni cliente, esporre numero, nome, cognome e ufficio dell'impiegato che lo segue solo 100 hanno <code>salesRepEmployeeNumber</code> &lt;&gt; NULL</p>	<p><b>RIGHT JOIN (108 righe)</b> Sono esposti <b>tutti i dipendenti</b> anche quelli che non seguono clienti (non compaiono in <code>salesRepEmployeeNumber</code>)</p>
---	---	---

**SELECT** employeeNumber, lastName, firstName, officeCode, customerNumber, customerName  
**FROM** customers

<b>LEFT JOIN employees</b>	<b>INNER JOIN employees</b>	<b>RIGHT JOIN employees</b>
----------------------------	-----------------------------	-----------------------------

**ON** employeeNumber = salesRepEmployeeNumber  
**ORDER BY** customerNumber

employeeNumber	lastname	firstName	officeCode	customerNumber	customerName
1370	Hernandez	Gerard	4	103	Atelier graphique
1166	Thompson	Leslie	1	112	Signal Gift Stores
1611	Fixter	Andy	6	114	Australian Collectors, Co.
1370	Hernandez	Gerard	4	119	La Rochelle Gifts
1504	Jones	Barry	7	121	Baane Mini Imports
1165	Jennings	Leslie	1	124	Mini Gifts Distributors Ltd.
NULL	NULL	NULL	NULL	125	Havel & Zbyszek Co
1504	Jones	Barry	7	128	Blauer See Auto, Co.
...	...	...	...	...	...

employeeNumber	lastName	firstName	officeCode	customerNumber	customerName
1002	Murphy	Diane	1	NULL	NULL
1056	Patterson	Mary	1	NULL	NULL
1076	Firrelli	Jeff	1	NULL	NULL
1088	Patterson	William	6	NULL	NULL
1102	Bondur	Gerard	4	NULL	NULL
1143	Bow	Anthony	1	NULL	NULL
1619	King	Tom	6	NULL	NULL
1625	Kato	Yoshimi	5	NULL	NULL
1370	Hernandez	Gerard	4	103	Atelier graphique
1166	Thompson	Leslie	1	112	Signal Gift Stores
...	...	...	...	...	...

risultato  
**INNER JOIN**  
v

risultato **LEFT JOIN**

I **clienti** con impiegato di riferimento a NULL (`salesRepEmployeeNumber`) presentano a NULL tutte le colonne selezionate da **employees**

employeeNumber	lastname	firstName	officeCode	customerNumber	customerName
1370	Hernandez	Gerard	4	103	Atelier graphique
1166	Thompson	Leslie	1	112	Signal Gift Stores
1611	Fixter	Andy	6	114	Australian Collectors, Co.
1370	Hernandez	Gerard	4	119	La Rochelle Gifts
1504	Jones	Barry	7	121	Baane Mini Imports
1165	Jennings	Leslie	1	124	Mini Gifts Distributors Ltd.
1504	Jones	Barry	7	128	Blauer See Auto, Co.
1165	Jennings	Leslie	1	129	Mini Wheels Co.
...	...	...	...	...	...

risultato **RIGHT JOIN**

Gli 8 **dipendenti** che non seguono clienti presentano a NULL tutte le colonne relative a **customers**; gli altri 15 dipendenti seguono uno o più **clienti**: ("Ogni dipendente **PUO'** seguire **UNO o PIU'** clienti")

**JOIN di 3 tabelle: employees , customers e orders**

su Foreign Key **customers.salesRepEmployeeNumber** e **orders.customerNumber**

(FK derivate rispettivamente dalle associazioni **seguire** e **compilare** del modello Concettuale)

Per i clienti di Frankfurt e di New Haven esporre nome e ufficio dei dipendenti che li seguono e numero e stato degli ordini che hanno compilato.

-- forma sconsigliata - SQL (6 righe estratte)	-- forma CONSIGLIATA – SQL2 (6 righe estratte)
<pre> SELECT      employeeNumber, lastName, firstName, officeCode,             C.customerNumber, customerName, city,             orderNumber, status FROM        employees , customers C , orders O WHERE      employeeNumber = salesRepEmployeeNumber -- cond.ni AND       C.customerNumber = O.customerNumber -- di join AND       city IN ('New Haven', 'Frankfurt') -- cond. di selez. ORDER BY  city, customerNumber, orderNumber; </pre>	<pre> INNER JOIN  customers C ON         employeeNumber = salesRepEmployeeNumber -- cond. join INNER JOIN  orders O ON         C.customerNumber = O.customerNumber -- cond. join WHERE      city IN ('New Haven', 'Frankfurt') -- cond. di selez. ORDER BY  city, customerNumber, orderNumber; </pre>

NOTA: per la seconda INNER JOIN è possibile utilizzare la dichiarazione **USING ( customerNumber )** ed evitare l'uso degli ALIAS.

employeeNumber	lastName	firstName	officeCode	customerNumber	customerName	city	orderNumber	status
1504	Jones	Barry	7	128	Blauer See Auto, Co.	Frankfurt	10101	Shipped
1504	Jones	Barry	7	128	Blauer See Auto, Co.	Frankfurt	10230	Shipped
1504	Jones	Barry	7	128	Blauer See Auto, Co.	Frankfurt	10300	Shipped
1504	Jones	Barry	7	128	Blauer See Auto, Co.	Frankfurt	10323	Shipped
NULL	NULL	NULL	NULL	247	Messner Shopping Network	Frankfurt	NULL	NULL
1286	Tseng	Foon Yue	3	168	American Souvenirs Inc	New Haven	NULL	NULL
1286	Tseng	Foon Yue	3	455	Super Scale Inc.	New Haven	10196	Shipped
1286	Tseng	Foon Yue	3	455	Super Scale Inc.	New Haven	10245	Shipped

NULL dovuti alla **RIGHT JOIN employees customers**

NULL per la **LEFT JOIN** tra **customers** e **orders**

Per estrarre anche i clienti che **non sono seguiti da un dipendente specifico** e/o che **non hanno ancora compilato ordini** (2 clienti in più) vanno utilizzati:

```

RIGHT JOIN FROM employees
+ RIGHT JOIN customers
ON employeeNumber = salesRepEmployeeNumber
LEFT JOIN orders
LEFT JOIN USING ( customerNumber )

```