



UNIVERSITÀ DEGLI STUDI DI SALERNO

Atzeni, Ceri, Fraternali, Paraboschi, Torlone
Basi di dati *Quinta edizione*
McGraw-Hill Education, 2018

BASI DI DATI

Concetti Base SQL-DML - terza parte



DIEM
DIPARTIMENTO DI
INGEGNERIA DELL'INFORMAZIONE ED ELETTRICA E MATEMATICA APPLICATA

Matteo Gaeta
Full Professor – Senior Member IEEE

Insegnando la matematica agli altri finisci per comprenderla meglio tu. (Ian Stewart)

JOIN

L'operatore JOIN nell'algebra relazionale:

$R_1(X_1), R_2(X_2)$

$R_1 \text{ JOIN } R_2$ è una relazione su X_1X_2

**$\{ t \text{ su } X_1X_2 \mid \text{esistono } t_1 \in R_1 \text{ e } t_2 \in R_2$
con } t[X_1] = t_1 \text{ e } t[X_2] = t_2 \}**

Questa definizione ci dice che le tuple del risultato sono ottenute combinando tuple degli operandi con valori uguali sugli attributi comuni

Esempio Prove scritte in un concorso

- I compiti sono anonimi e ad ognuno è associata una busta chiusa con il nome del candidato
- Ciascun compito e la relativa busta vengono contrassegnati con uno stesso numero

Numero	Voto	Numero	Candidato
1	25	1	Mario Rossi
2	13	2	Nicola Russo
3	27	3	Mario Bianchi
4	28	4	Remo Neri

Numero	Candidato	Voto
1	Mario Rossi	25
2	Nicola Russo	13
3	Mario Bianchi	27
4	Remo Neri	28

L'algebra e il denaro sono essenzialmente livellatori; la prima intellettualmente, l'altro effettivamente. (Simone Weil)

JOIN NATURALE

- Il JOIN naturale essenzialmente utile per riflessioni astratte ha la caratteristica di correlare dati di relazioni diverse sulla basi di valori uguali in **ATTRIBUTI CON LO STESSO NOME**
- operatore binario (generalizzabile)
- produce un risultato
 - sull'unione degli attributi degli operandi
 - con ennuple costruite ciascuna a partire da una ennupla di ognuno degli operandi

Se la gente non crede che la matematica sia semplice, è soltanto perché non si rende conto di quanto complicata sia la vita. (J. V. Neumann)

Esempio

Impiegato	Reparto	Reparto	Capo
Rossi	A	A	Mori
Neri	B	B	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

- ogni ennupla contribuisce al risultato:

- join completo

Prodotto cartesiano

$R_1(X_1)$, $R_2(X_2)$ con i due insiemi di attributi X_1 e X_2 sono disgiunti è delicato

il risultato è sempre X_1X_2 e ciascuna tupla deriva sempre da due tuple degli operandi, ma atteso che X_1 e X_2 non hanno attributi in comune, non viene di fatto richiesta nessuna condizione per partecipare al JOIN (che normalmente ottiene tuple del risultato combinando tuple degli operandi con valori uguali sugli attributi comuni), degenera in una condizione sempre verificata.

In effetti, potremmo affermare che il JOIN degenera nel prodotto cartesiano

Ovvero l'operare prodotto cartesiano è definito esattamente come il JOIN NATURALE su relazioni senza attributi in comune. Ovviamente parliamo di concatenazione di tuple e non di coppie.

contiene sempre un numero di ennuple pari al prodotto delle cardinalità degli operandi (le ennuple sono tutte combinabili)

Esempio Prodotto cartesiano

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Codice	Capo
A	Mori
B	Bruni

Impiegati JOIN Reparti

Impiegato	Reparto	Codice	Capo
Rossi	A	A	Mori
Rossi	A	B	Bruni
Neri	B	A	Mori
Neri	B	B	Bruni
Bianchi	B	A	Mori
Bianchi	B	B	Bruni

Esempio un join non completo

Impiegato	Reparto	Reparto	Capo
Rossi	A	B	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

Esempio Un join vuoto

Impiegato	Reparto	Reparto	Capo
Rossi	A	D	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
-----------	---------	------

Esempio un join completo, con n x m ennuple

Impiegato	Reparto	Reparto	Capo
Rossi	B	B	Mori
Neri	B	B	Bruni

Impiegato	Reparto	Capo
Rossi	B	Mori
Rossi	B	Bruni
Neri	B	Mori
Neri	B	Bruni

Cardinalità del join

- Il join di R_1 e R_2 contiene un numero di ennuple ...
 - **compreso fra zero e il prodotto di $|R_1|$ e $|R_2|$**
- se il join coinvolge una chiave di R_2 , allora il numero di ennuple è ...
 - **compreso fra zero e $|R_1|$**
- se il join coinvolge una chiave di R_2 e un vincolo di integrità referenziale, allora il numero di ennuple è
 - **pari a $|R_1|$**

Cardinalità del join

- $R_1(A,B)$, $R_2(B,C)$
- in generale

$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1| \times |R_2|$$
- se B è chiave in R_2

$$0 \leq |R_1 \text{ JOIN } R_2| \leq |R_1|$$
- se B è chiave in R_2 ed esiste vincolo di integrità referenziale fra B (in R_1) e R_2 :

$$|R_1 \text{ JOIN } R_2| = |R_1|$$

Join, una difficoltà

Impiegato	Reparto	Reparto	Capo
Rossi	A	B	Mori
Neri	B	C	Bruni
Bianchi	B		

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori

- alcune ennuple non contribuiscono al risultato: vengono "tagliate fuori"

Join esterno

- Il join **esterno** estende, con valori nulli, le ennuple che verrebbero tagliate fuori da un join (**interno**)
- esiste in tre versioni:
 - sinistro, destro, completo
 - **sinistro**: mantiene tutte le ennuple del primo operando, estendendole con valori nulli, se necessario
 - **destro**: ... del secondo operando ...
 - **completo**: ... di entrambi gli operandi ...

Esempio Join Esterno sinistro

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{LEFT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL

Esempio Join Esterno destro

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{RIGHT} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

Esempio Join Esterno completo

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati JOIN_{FULL} Reparti

Impiegato	Reparto	Capo
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

Semijoin

- Operatore su due relazioni $R_1(X_1)$, $R_2(X_2)$
- Restituisce una relazione su X_1 , con le ennuple di R_1 che contribuiscono al join con R_2

Impiegati

Impiegato	Reparto
Rossi	A
Neri	B
Bianchi	B

Reparti

Reparto	Capo
B	Mori
C	Bruni

Impiegati SEMIJOIN Reparti

Impiegato	Reparto
Neri	B
Bianchi	B

SQL

- Sintassi (quasi) generale

- **Select** <elenco attributi>
From <elenco tabelle>
[Where <condizione>
[Group by <attributi di raggruppamento>
[Having <condizione di raggruppamento>
[Order by <elenco attributi>



SQL e Algebra Relazionale

In Pratica dati $R1(A,B)$ e $R2(C,D)$
 Attraverso lo statement SQL realizzo:

SELECT distinct $R1.A, R2.D$ (proiezione)
FROM $R1, R2$ (prodotto cartesiano)
WHERE $R1.B = R2.C$ (selezione)

$PROJ_{A,D} (SEL_{B=C} (R1 JOIN R2))$

SQL e le ridenominazioni

- Possono essere necessarie ridenominazioni
- Esse possono essere eseguite:
 - **nel prodotto cartesiano**
 - e/o**
 - **nella target list (ovvero proiezione)**

```
select X.A1 AS B1, ...  
from   R1 X, R2 Y, R1 Z  
where  X.A2 = Y.A3 AND ...
```

SQL Esecuzione e specifica delle query

- Le espressioni SQL sono dichiarative e noi ne stiamo vedendo la semantica
- i DBMS eseguono le operazioni in modo efficiente, ad esempio:
 - ✓ eseguono le selezioni al più presto
 - ✓ se possibile, eseguono join e non prodotti cartesiani
- La capacità dei DBMS di "ottimizzare" le interrogazioni, rende (di solito) non necessario preoccuparsi dell'efficienza quando si specifica un'interrogazione
- È perciò più importante preoccuparsi della chiarezza (anche perché così è più difficile sbagliare ...)

JOIN

Join (SQL)

Il JOIN è una clausola del linguaggio SQL che serve a combinare le tuple di due o più relazioni di una base di dati.

SELECT [DISTINCT] *lista Attributi*

FROM Tabella

{Tipo **JOIN** Altra Tabella **ON** Condizione di Join}

[WHERE *altra Condizione*]

Lo standard ANSI definisce alcune specifiche per il linguaggio SQL sul tipo di JOIN da effettuare: INNER, FULL, LEFT e RIGHT, alle quali diversi DBMS aggiungono CROSS.

JOIN Implicito e ridenominazione

- Le persone che guadagnano più dei rispettivi padri; mostrare nome, reddito e reddito del padre

- ❖ **Select f.Nome, f.Reddito, p.Reddito**
from Persone p, Paternità, Persone f
where p.Nome = Padre and Figlio = f.Nome and f.Reddito > p.Reddito

- ❖ **Select Figlio, f.Reddito as Reddito,**
p.Reddito as RedditoPadre **from**
Persone p, Paternità, Persone f **where**
p.Nome = Padre and Figlio = f.Nome
and f.Reddito > p.Reddito

Maternità	Madre	Figlio
	Luisa	Maria
	Luisa	Luigi
	Anna	Olga
	Anna	Filippo
	Maria	Andre
	Maria	Aldo

Paternità	Padre	Figlio
	Sergio	Franco
	Luigi	Olga
	Luigi	Filippo
	Franco	Andre
	Franco	Aldo

Persone	Nome	Età	Reddito
	Andrea	27	21
	Aldo	25	15
	Maria	55	42
	Anna	50	35
	Filippo	26	30
	Luigi	50	40
	Franco	60	20
	Olga	30	41
	Sergio	85	35
	Luisa	75	87

JOIN Implicito ed Esplicito

- Elencare Padre e madre di ogni persona

```
select Paternita.Figlio, Padre, Madre
from Maternita, Paternita
where Paternita.Figlio =
Maternita.Figlio
```

```
select Madre, Paternita.Figlio, Padre
from Maternita JOIN Paternita on
Paternita.Figlio = Maternita.Figlio
```

Maternità	Madre	Figlio
	Luisa	Maria
	Luisa	Luigi
	Anna	Olga
	Anna	Filippo
	Maria	Andre
	Maria	Aldo

Paternità	Padre	Figlio
	Sergio	Franco
	Luigi	Olga
	Luigi	Filippo
	Franco	Andre
	Franco	Aldo

Persone	Nome	Età	Reddito
	Andrea	27	21
	Aldo	25	15
	Maria	55	42
	Anna	50	35
	Filippo	26	30
	Luigi	50	40
	Franco	60	20
	Olga	30	41
	Sergio	85	35
	Luisa	75	87

JOIN

Lo standard ANSI definisce alcune specifiche per il linguaggio SQL sul tipo di JOIN da effettuare:

- ✓ INNER JOIN;
- ✓ FULL JOIN;
- ✓ LEFT JOIN;
- ✓ RIGHT JOIN

alle quali diversi DBMS aggiungono CROSS.

In alcuni casi è possibile, spesso è necessario, che una tabella debba essere combinata con se stessa, in questo caso si parlerà di SELF JOIN.

JOIN – Tabelle di riferimento

```
DROP TABLE IF EXISTS DIPARTIMENTI CASCADE;
CREATE TABLE DIPARTIMENTI (
  ID_DIP CHAR(4) NOT NULL,
  NOME VARCHAR(20) NOT NULL UNIQUE,
  DIP_IND VARCHAR (50) ,
  CITTA VARCHAR (20) ,
  CONSTRAINT PK_DIPARTIMENTI PRIMARY KEY (ID_DIP) );
```

```
/* INSERIRE OCCORRENZE IN UNA TABELLA DIPARTIMENTI */
INSERT INTO DIPARTIMENTI (ID_DIP, NOME, DIP_IND, CITTA)
VALUES ('AMMZ', 'AMMINISTRAZIONE', 'VIA TITO LIVIO, 27', 'MILANO' );
INSERT INTO DIPARTIMENTI (ID_DIP, NOME, DIP_IND, CITTA)
VALUES ('PROD', 'PRODUZIONE', 'P.LE LAVATER, 3', 'TORINO' );
INSERT INTO DIPARTIMENTI (ID_DIP, NOME, DIP_IND, CITTA)
VALUES ('DIST', 'DISTRIBUZIONE', 'VIA SEGRE, 9', 'ROMA' );
INSERT INTO DIPARTIMENTI (ID_DIP, NOME, DIP_IND, CITTA)
VALUES ('DIRE', 'DIREZIONE', 'VIA TITO LIVIO, 27', 'MILANO' );
INSERT INTO DIPARTIMENTI (ID_DIP, NOME, DIP_IND, CITTA)
VALUES ('RICE', 'RICERCA', 'VIA VENOSA, 6', 'MILANO' );
```

JOIN – Tabelle di riferimento

```
40 SELECT *
41 FROM DIPARTIMENTI;
42
```

Data Output Explain Messages Notifications

	id_dip [PK] character (4)	nome character varying (20)	dip_ind character varying (50)	citta character varying (20)
1	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
2	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
3	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
4	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
5	RICE	RICERCA	VIA VENOSA, 6	MILANO

JOIN – Tabelle di riferimento

```

/* CREARE UNA TABELLA IMPIEGATI */
DROP TABLE IF EXISTS IMPIEGATI CASCADE;
CREATE TABLE IMPIEGATI (
    MATRICOLA CHAR(6) PRIMARY KEY,
    NOME VARCHAR(20),
    COGNOME VARCHAR (20),
    ID_DIP CHAR(4),
    UFFICIO NUMERIC (3),
    STIPENDIO NUMERIC (9) DEFAULT 0,
    CITTA CHARACTER VARYING(50),
    UNIQUE (COGNOME, NOME),
    CONSTRAINT FK_DIPART
    FOREIGN KEY (ID_DIP) REFERENCES DIPARTIMENTI (ID_DIP)
    ON DELETE SET NULL ON UPDATE RESTRICT );

```

JOIN – Tabelle di riferimento

```

/* INSERIRE OCCORRENZE IN UNA TABELLA IMPIEGATI */
INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000001', 'MARIO', 'ROSSI', 'AMMZ', 10, 45, 'MILANO');
INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000002', 'CARLO', 'BIANCHI', 'PROD', 20, 36, 'TORINO');
INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000003', 'GIOVANNI', 'VERDI', 'AMMZ', 20, 40, 'ROMA');
INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000004', 'FRANCO', 'NERI', 'DIST', 16, 45, 'NAPOLI');
INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000005', 'CARLO', 'ROSSI', 'DIRE', 14, 80, 'MILANO');
INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000006', 'LORENZO', 'GIALLI', 'DIRE', 7, 73, 'GENOVA');
INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000007', 'PAOLA', 'ROSATI', 'AMMZ', 75, 40, 'VENEZIA');
INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000008', 'MARCO', 'FRANCO', 'PROD', 20, 46, 'ROMA');

INSERT INTO IMPIEGATI (MATRICOLA, NOME, COGNOME, ID_DIP, UFFICIO, STIPENDIO, CITTA)
VALUES ('000009', 'MATTEO', 'GAETA', NULL, 20, 46, 'SALERNO');

```

JOIN – Tabelle di riferimento

```

99  SELECT *
100 FROM IMPIEGATI;
101
102

```

Data Output Explain Messages Notifications

	matricola [PK] character	nome character va	cognome character var	id_dip character	ufficio numeric	stipendio numeric (9)	citta character v
1	000001	MARIO	ROSSI	AMMZ	10	45	MILANO
2	000002	CARLO	BIANCHI	PROD	20	36	TORINO
3	000003	GIOVANNI	VERDI	AMMZ	20	40	ROMA
4	000004	FRANCO	NERI	DIST	16	45	NAPOLI
5	000005	CARLO	ROSSI	DIRE	14	80	MILANO
6	000006	LORENZO	GIALLI	DIRE	7	73	GENOVA
7	000007	PAOLA	ROSATI	AMMZ	75	40	VENEZIA
8	000008	MARCO	FRANCO	PROD	20	46	ROMA
9	000009	MATTEO	GAETA	[null]	20	46	SALERNO

JOIN – Tabelle di riferimento

Nella Tabella IMPIEGATI vi possono essere tra zero e N tuple che fanno riferimento alla chiave primaria della tabella DIPARTIMENTI.

Infatti un valore di ID_DIP in tabella IMPIEGATI può ripetersi più volte, ma deve sempre essere presente nella Tabella DIPARTIMENTI e corrispondere ad un valore presente nella chiave primaria di DIPARTIMENTI oppure può assumere il valore NULL (l'impiegato non è stato ancora assegnato a nessun dipartimento)

Nota: Il Dipartimento "Ricerca" ID_DIP="RICE" della tabella "DIPARTIMENTI" non ha alcuna corrispondenza nella tabella "IMPIEGATI". Mentre l'impiegato "Gaeta" non è stato assegnato ad alcun DIPARTIMENTO, infatti ID_DIP=NULL.

INNER JOIN

L'Inner Join si distingue da tutti gli altri tipi di Join per il suo fornire un **risultato minimale**.

Il risultato di una Inner Join è costituito solamente dai record di dati (Tuple, Righe) dell'unione incrociata che soddisfano la condizione di selezione.

Considerando che l'Inner Join è il Join SQL più importante, si può omettere la parola chiave "inner".

INNER JOIN

Realizzando una Inner Join tra due Tabelle, se si confronta la tabella dei risultati con le due tabelle di output, si noterà che possono mancare alcune Tuple per ogni tabella coinvolta nel Join.

Si tratta di quei record di dati che nei campi dichiarati dalla condizione di Inner Join non contengono valori corrispondenti tra le Tabelle coinvolte nel Join.

Quando vogliamo rilevare esattamente tali irregolarità e renderle visibili nell'interrogazione, dobbiamo utilizzare **una outer join al posto di una inner join**.

INNER JOIN

Una INNER JOIN crea una nuova tabella combinando i valori delle due tabelle di partenza (DIPARTIMENTI, IMPIEGATI) basandosi su una certa regola di confronto.

La query compara ogni riga della tabella DIPARTIMENTI con ciascuna riga della tabella IMPIEGATI cercando di soddisfare la regola di confronto definita.

Quando la REGOLA DI JOIN viene soddisfatta, i valori di tutte le colonne delle tabelle DIPARTIMENTI e IMPIEGATI vengono combinate in un'unica riga nella costruzione della tabella risultante.

La INNER JOIN è la forma di JOIN usata più di frequente nelle applicazioni e rappresenta la modalità predefinita.

Il termine INNER si può omettere.

INNER JOIN

EFFETTUARE UN INNER JOIN TRA LA TABELLA DIPARTIMENTI E LA TABELLA IMPIEGATI SULLE COLONNE

DIPARTIMENTI.ID_DIP

E

IMPIEGATI.ID_DIP

CON LA REGOLA JOIN

<< DIPARTIMENTI.ID_DIP = IMPIEGATI.ID_DIP >>

Esempio di INNER JOIN in forma esplicita:

```
SELECT * FROM IMPIEGATI INNER JOIN DIPARTIMENTI  
ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;
```

INNER JOIN

```

102 SELECT *
103 FROM IMPIEGATI INNER JOIN DIPARTIMENTI
104      ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;
105

```

Data Output Explain Messages Notifications

	matricola	nome	cognome	id_dip	ufficio	stipendio	citta	id_dip	nome	dip_ind	citta
	character (6)	character v	character v	character	numeric (5)	numeric (9)	character v	character	character varying (20)	character varying (5)	character v
1	000001	MARIO	ROSSI	AMMZ	10	45	MILANO	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
2	000002	CARLO	BIANCHI	PROD	20	36	TORINO	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
3	000003	GIOVANNI	VERDI	AMMZ	20	40	ROMA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
4	000004	FRANCO	NERI	DIST	16	45	NAPOLI	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
5	000005	CARLO	ROSSI	DIRE	14	80	MILANO	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
6	000006	LORENZO	GIALLI	DIRE	7	73	GENOVA	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
7	000007	PAOLA	ROSATI	AMMZ	75	40	VENEZIA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
8	000008	MARCO	FRANCO	PROD	20	46	ROMA	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO

INNER JOIN

```

SELECT * FROM IMPIEGATI INNER JOIN DIPARTIMENTI ON
IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;

```

EQUIVALE A:

```

SELECT * FROM IMPIEGATI, DIPARTIMENTI WHERE
IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;

```

```

106 SELECT *
107 FROM IMPIEGATI, DIPARTIMENTI
108 WHERE IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;

```

Data Output Explain Messages Notifications

	matricola	nome	cognome	id_dip	ufficio	stipendio	citta	id_dip	nome	dip_ind	citta
	character	character v	character v	character	numeri	numeric (9)	character	character	character varying (20)	character varying (5)	character
1	000001	MARIO	ROSSI	AMMZ	10	45	MILANO	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
2	000002	CARLO	BIANCHI	PROD	20	36	TORINO	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
3	000003	GIOVANNI	VERDI	AMMZ	20	40	ROMA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
4	000004	FRANCO	NERI	DIST	16	45	NAPOLI	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
5	000005	CARLO	ROSSI	DIRE	14	80	MILANO	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
6	000006	LORENZO	GIALLI	DIRE	7	73	GENOVA	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
7	000007	PAOLA	ROSATI	AMMZ	75	40	VENEZIA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
8	000008	MARCO	FRANCO	PROD	20	46	ROMA	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO

INNER JOIN

Osserviamo che l'impiegato "Gaeta" e il Dipartimento "Ricerca" non sono presenti nel risultato della Query

La tupla di «Gaeta» ha NULL come valore nella colonna ID_DIP nella Tabella IMPIEGATI,

mentre la tupla di «Ricerca» pur essendo presente in tabella DIPARTIMENTI e correttamente valorizzata in ogni suo campo non ha nessun valore corrispondente in nessuna tupla della Tabella IMPIEGATI alla colonna ID_DIP.

Quando come risultato finale si desidera avere anche i record che non hanno corrispondenza si usa una query con **OUTER JOIN**.

INNER JOIN

Le inner join si possono realizzare sotto forma di theta join, equi join, non equi join e natural join.

Theta join, equi join e non equi join

L'inner join della terminologia SQL corrisponde al theta join dell'algebra relazionale.

La **theta join** si differenzia da equi join e non equi join, in quanto fornisce agli utenti un numero illimitato di operatori di confronto tra cui scegliere.

Le **equi join**, invece, limitano la condizione di selezione per le query sull'uguaglianza dei valori delle colonne.

Le **non equi join**, invece, ammettono tutti gli operatori di confronto ad eccezione del segno uguale.

THETA JOIN – Operatori di Confronto

Tipo di

join

Operatori di confronto ammessi

Theta join = (uguale) < (minore) > (maggiore) ≤ (minore o uguale) ≥ (maggiore o uguale) <> (disuguale) != (disuguale)

Equi join = (uguale)

Non equi join < (minore) > (maggiore) ≤ (minore o uguale) ≥ (maggiore o uguale) <> (disuguale) != (disuguale)

THETA JOIN – Operatori di Confronto

Se due tabelle sono collegate da colonne con lo stesso nome, le inner join vengono solitamente convertite in natural join.

Le natural join sono una sottocategoria di equi join.

Come anche l'equi join, la natural join prevede l'uguaglianza dei valori delle due colonne come condizione di selezione.

L'operatore natural join collega automaticamente le tabelle utilizzando colonne con lo stesso nome.

La condizione di selezione non deve essere definita esplicitamente.

THETA JOIN – Operatori di Confronto

OSSERVAZIONE IMPORTANTE

La tabella dei risultati della natural join si differenzia da quella della classica inner join, in quanto le colonne con lo stesso nome nelle tabelle di output non sono elencate due volte, ma vengono unite in una colonna comune.

Per le natural join è disponibile una notazione abbreviata che non richiede una clausola USING.

Si utilizza invece l'operatore natural join.

EQUI JOIN

La EQUI JOIN, ha un particolare tipo di comparatore, detto **THETA JOIN**, che utilizza come metodo di verifica, **solamente l'uguaglianza matematica come regola di confronto**.

La query riportata di seguito utilizza una equi-join:

```
102 SELECT *
103 FROM IMPIEGATI INNER JOIN
104 DIPARTIMENTI ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;
105
```

	matricola character (6)	nome character var	cognome character var	id_dip character (4)	ufficio numeric (1)	stipendio numeric (9)	citta character var	id_dip character (4)	nome character varying (20)	dip_ind character varying (50)	citta character
1	000001	MARIO	ROSSI	AMMZ	10	45	MILANO	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
2	000002	CARLO	BIANCHI	PROD	20	36	TORINO	PROD	PRODUZIONE	PLE LAVATER, 3	TORINO
3	000003	GIOVANNI	VERDI	AMMZ	20	40	ROMA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
4	000004	FRANCO	NERI	DIST	16	45	NAPOLI	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
5	000005	CARLO	ROSSI	DIRE	14	80	MILANO	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
6	000006	LORENZO	GIALLI	DIRE	7	73	GENOVA	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
7	000007	PAOLA	ROSATI	AMMZ	75	40	VENEZIA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
8	000008	MARCO	FRANCO	PROD	20	46	ROMA	PROD	PRODUZIONE	PLE LAVATER, 3	TORINO

EQUI JOIN

SQL fornisce altre opzioni attraverso la keyword **USING**

SELECT * FROM IMPIEGATI **INNER JOIN** DIPARTIMENTI **USING (id_dip);**

PS: Ricordarsi delle parentesi quando si utilizza la USING

USING è supportata da Microsoft, MySQL, Oracle, PostgreSQL, SQLite, e DB2/400.

```
106 SELECT *
107 FROM IMPIEGATI INNER JOIN DIPARTIMENTI
108 USING (id_dip);
109
```

Data Output Explain Messages Notifications									
id_dip	matricola	nome	cognome	ufficio	stipendio	citta	nome	dip_ind	citta
character	character	character v	character v	numeric	numeric (8)	character	character varying (20)	character varying (50)	character
1	AMMZ	000001	MARIO	ROSSI	10	45	MILANO	AMMINISTRAZIONE	VIA TITO LIVIO, 27
2	PROD	000002	CARLO	BIANCHI	20	36	TORINO	PRODUZIONE	P.LE LAVATER, 3
3	AMMZ	000003	GIOVANNI	VERDI	20	40	ROMA	AMMINISTRAZIONE	VIA TITO LIVIO, 27
4	DIST	000004	FRANCO	NERI	16	45	NAPOLI	DISTRIBUZIONE	VIA SEGRE, 9
5	DIRE	000005	CARLO	ROSSI	14	80	MILANO	DIREZIONE	VIA TITO LIVIO, 27
6	DIRE	000006	LORENZO	GIALLI	7	73	GENOVA	DIREZIONE	VIA TITO LIVIO, 27
7	AMMZ	000007	PAOLA	ROSATI	75	40	VENEZIA	AMMINISTRAZIONE	VIA TITO LIVIO, 27
8	PROD	000008	MARCO	FRANCO	20	46	ROMA	PRODUZIONE	P.LE LAVATER, 3
									TORINO

Natural Join

il **natural join** offre ulteriori specializzazioni di equi-join.

Solitamente il JOIN confronta colonne di tabelle diverse che hanno lo stesso nome.

il natural join effettua questa operazione.

Se considerate che nel nostro caso, le Tabelle IMPIEGATI e DIPARTIMENTI hanno i seguenti campi in comune: **IN TABELLA DIPARTIMENTI** ID_DIP CHAR(4) NOT NULL, NOME VARCHAR(20) – contiene il nome del Dipartimento; CITTA VARCHAR (20) contiene la città sede del Dipartimento, mentre in **TABELLA IMPIEGATI** NOME VARCHAR(20) – contiene il nome dell'impiegato; CITTA VARCHAR(20) contiene magari la città dove risiede -, osserverete che se si effettua un Natural Join sulle tabelle precedentemente definite il risultato sarà:

Natural Join

1 / 2

```
173 SELECT *
174 FROM Impiegati NATURAL JOIN Dipartimenti;
```

Data Output Explain Messages Notifications

nome	id_dip	citta	matricola	cognome	ufficio	stipendio	dip_ind
character varying	character (4)	character varying	character (6)	character varying	numeric	numeric (9)	character

Provate a MODIFICARE i nomi delle colonne con lo stesso nome in entrambe le tabelle, ovvero modificate il nome delle colonne NOME e CITTA in tabella DIPARTIMENTI ad es. NOME in NOME_D e CITTA in CITTA_D.

In tal modo esse avranno solo il campo ID-DIP in comune. Poi verificate che questo natural join equivale alla precedente equi-join (USING) che abbiamo realizzato

Natural Join

```
113 ALTER TABLE DIPARTIMENTI RENAME NOME TO NOME_D;
114 ALTER TABLE DIPARTIMENTI RENAME CITTA TO CITTA_D;
115
116 SELECT *
117 FROM Impiegati NATURAL JOIN Dipartimenti;
```

Data Output Explain Messages Notifications

	id_dip	matricola	nome	cognome	ufficio	stipendio	citta	nome_d	dip_ind	citta_d
	character	character	character	character	numeric	numeric (9)	character	character varying (20)	character	character
1	AMMZ	000001	MARIO	ROSSI	10	45	MILANO	AMMINISTRAZIONE	VIA TITO...	MILANO
2	PROD	000002	CARLO	BIANCHI	20	36	TORINO	PRODUZIONE	P.LE LAV...	TORINO
3	AMMZ	000003	GIOVANNI	VERDI	20	40	ROMA	AMMINISTRAZIONE	VIA TITO...	MILANO
4	DIST	000004	FRANCO	NERI	16	45	NAPOLI	DISTRIBUZIONE	VIA SEG...	ROMA
5	DIRE	000005	CARLO	ROSSI	14	80	MILANO	DIREZIONE	VIA TITO...	MILANO
6	DIRE	000006	LORENZO	GIALLI	7	73	GENOVA	DIREZIONE	VIA TITO...	MILANO
7	AMMZ	000007	PAOLA	ROSATI	75	40	VENEZIA	AMMINISTRAZIONE	VIA TITO...	MILANO
8	PROD	000008	MARCO	FRANCO	20	46	ROMA	PRODUZIONE	P.LE LAV...	TORINO

Una natural join viene implementata automaticamente come inner join.

Se, invece, volete implementare una outer join come natural join, sono necessarie parole chiave aggiuntive (ad esempio una natural left outer join).

Cross Join

Una cross join, cartesian join o product fornisce le strutture attraverso cui tutti i tipi di inner join operano.

Il risultato di una cross join è il prodotto cartesiano di tutte le righe delle tabelle che concorrono alla query di join.

È come dire che stiamo facendo una inner join senza impostare la regola di confronto o in cui la regola di confronto ritorna sempre vero.

Cross Join

Esempio di cross join esplicito:

SELECT * FROM Impiegati CROSS JOIN Dipartimenti;

matricola	nome	cognome	id_dip	ufficio	stipendio	citta	id_dip	nome_dip	id_dip	citta_dip
id_cliente	character	character	character	number	number	character	character	character	number	character
1	000001	MARIO	ROSSI	AMMZ	10	45 MILANO	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
2	000002	CARLO	BIANCHI	PROD	20	36 TORINO	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
3	000003	GIOVANNI	VERDI	AMMZ	20	40 ROMA	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
4	000004	FRANCO	NERI	DIST	16	45 NAPOLI	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
5	000005	CARLO	ROSSI	DIRE	14	80 MILANO	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
6	000006	LORENZO	GALLI	DIRE	7	73 GENOVA	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
7	000007	PAOLA	ROSATI	AMMZ	75	40 VENEZIA	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
8	000008	MARCO	FRANCO	PROD	20	46 ROMA	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
9	000009	MATTEO	GAETA	[null]	20	46 SALER...	AMMZ	AMMINISTRAZ...	VIA TITO LIVIO...	MILA...
10	000001	MARIO	ROSSI	AMMZ	10	45 MILANO	PROD	PRODUZIONE	FILE LAVATER...	TORINO
11	000002	CARLO	BIANCHI	PROD	20	36 TORINO	PROD	PRODUZIONE	FILE LAVATER...	TORINO
12	000003	GIOVANNI	VERDI	AMMZ	20	40 ROMA	PROD	PRODUZIONE	FILE LAVATER...	TORINO
13	000004	FRANCO	NERI	DIST	16	45 NAPOLI	PROD	PRODUZIONE	FILE LAVATER...	TORINO
14	000005	CARLO	ROSSI	DIRE	14	80 MILANO	PROD	PRODUZIONE	FILE LAVATER...	TORINO
15	000006	LORENZO	GALLI	DIRE	7	73 GENOVA	PROD	PRODUZIONE	FILE LAVATER...	TORINO
16	000007	PAOLA	ROSATI	AMMZ	75	40 VENEZIA	PROD	PRODUZIONE	FILE LAVATER...	TORINO
17	000008	MARCO	FRANCO	PROD	20	46 ROMA	PROD	PRODUZIONE	FILE LAVATER...	TORINO
18	000009	MATTEO	GAETA	[null]	20	46 SALER...	PROD	PRODUZIONE	FILE LAVATER...	TORINO
19	000009	MATTEO	GAETA	[null]	20	46 SALER...	PROD	PRODUZIONE	FILE LAVATER...	TORINO
20	000001	MARIO	ROSSI	AMMZ	10	45 MILANO	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
21	000002	CARLO	BIANCHI	PROD	20	36 TORINO	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
22	000003	GIOVANNI	VERDI	AMMZ	20	40 ROMA	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
23	000004	FRANCO	NERI	DIST	16	45 NAPOLI	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
24	000005	CARLO	ROSSI	DIRE	14	80 MILANO	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
25	000006	LORENZO	GALLI	DIRE	7	73 GENOVA	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
26	000007	PAOLA	ROSATI	AMMZ	75	40 VENEZIA	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
27	000008	MARCO	FRANCO	PROD	20	46 ROMA	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
28	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
29	000001	MARIO	ROSSI	AMMZ	10	45 MILANO	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
30	000002	CARLO	BIANCHI	PROD	20	36 TORINO	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
31	000003	GIOVANNI	VERDI	AMMZ	20	40 ROMA	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
32	000004	FRANCO	NERI	DIST	16	45 NAPOLI	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
33	000005	CARLO	ROSSI	DIRE	14	80 MILANO	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
34	000006	LORENZO	GALLI	DIRE	7	73 GENOVA	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
35	000007	PAOLA	ROSATI	AMMZ	75	40 VENEZIA	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
36	000008	MARCO	FRANCO	PROD	20	46 ROMA	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
37	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
38	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
39	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
40	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
41	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
42	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
43	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
44	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
45	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
46	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
47	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
48	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
49	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
50	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
51	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
52	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
53	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
54	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
55	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
56	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
57	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
58	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
59	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
60	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
61	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
62	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
63	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
64	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
65	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
66	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
67	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
68	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
69	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
70	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
71	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
72	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
73	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
74	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
75	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
76	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
77	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
78	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
79	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
80	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
81	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
82	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
83	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
84	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
85	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
86	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
87	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
88	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
89	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
90	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
91	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
92	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
93	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
94	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
95	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
96	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
97	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
98	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
99	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...
100	000009	MATTEO	GAETA	[null]	20	46 SALER...	DIRE	DIREZIONE	VIA TITO LIVIO...	MILA...

Outer Join

Una Outer Join non richiede che ci sia corrispondenza esatta tra le righe di due tabelle.

La tabella risultante da una Outer Join trattiene tutti quei record che non hanno alcuna corrispondenza tra le tabelle.

Le Outer Join si suddividono in Left Outer Join, Right Outer Join, e Full Outer Join, in base a quale sia la tabella di cui intendiamo trattenere i valori in caso di mancata corrispondenza della regola di confronto da (sinistra, destra, o entrambi).

In questo caso left (sinistra) e right (destra) si riferiscono ai due lati della keyword JOIN.

Si sottolinea come esista un ordine di esecuzione tra le condizioni specificate nella WHERE e quelle presenti nella ON.

Specificatamente le clausole presenti nella ON sono valutate ANTERIORMENTE all'esecuzione del join mentre le clausole nella where sono valutate SUCCESSIVAMENTE all'esecuzione del join.

Left Outer Join e Right Outer Join

Il risultato di una query **LEFT OUTER JOIN** (o semplicemente left join) per le tabelle IMPIEGATI e DIPARTIMENTI contiene sempre tutti i record della tabella di sinistra ("left") IMPIEGATI, mentre vengono estratti dalla tabella di destra ("right") DIPARTIMENTI solamente le righe che trovano corrispondenza nella regola di confronto della join **ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip**.

Questo significa che la JOIN è "GUIDATA" dalla Tabella LEFT e che se la clausola ON trova zero righe nella tabella di DESTRA, la join mostrerà una riga risultante con valore NULL in tutte le colonne corrispondenti al risultato per le colonne di DESTRA.

Nel nostro caso ciò accade per la tupla Gaeta della Tabella IMPIEGATI.

In pratica la query con **RIGHT OUTER JOIN** (o right join) semplicemente ripropone il funzionamento della **LEFT OUTER JOIN**, a tabelle invertite.

Left Outer Join

```

119 SELECT *
120 FROM Impiegati LEFT OUTER JOIN Dipartimenti
121 ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;

```

	matricola	nome	cognome	id_dip	ufficio	stipendio	citta	id_dip	nome_d	dip_ind	citta_d
	character	character var	character var	character	numeric	numeric (9)	character	character	character varying (20)	character varying (50)	character
1	000001	MARIO	ROSSI	AMMZ	10	45	MILANO	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
2	000002	CARLO	BIANCHI	PROD	20	36	TORINO	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
3	000003	GIOVANNI	VERDI	AMMZ	20	40	ROMA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
4	000004	FRANCO	NERI	DIST	16	45	NAPOLI	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
5	000005	CARLO	ROSSI	DIRE	14	80	MILANO	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
6	000006	LORENZO	GIALLI	DIRE	7	73	GENOVA	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
7	000007	PAOLA	ROSATI	AMMZ	75	40	VENEZIA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
8	000008	MARCO	FRANCO	PROD	20	46	ROMA	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
9	000009	MATTEO	GAETA	[null]	20	46	SALERNO	[null]	[null]	[null]	[null]

Right Outer Join

Viceversa con il **RIGHT OUTER JOIN** la query è "GUIDATA" dalla Tabella RIGHT è quindi ciò accadrà per la tabella di SINISTRA, nel nostro caso ciò accade per la tupla Ricerca della Tabella DIPARTIMENTI

```

123 SELECT *
124 FROM Impiegati RIGHT OUTER JOIN Dipartimenti
125 ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;

```

	matricola	nome	cognome	id_dip	ufficio	stipendio	citta	id_dip	nome_d	dip_ind	citta_d
	character	character varying	character varying	character	numeric	numeric (9)	character	character	character varying (20)	character varying (50)	character varying
1	000001	MARIO	ROSSI	AMMZ	10	45	MILANO	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
2	000002	CARLO	BIANCHI	PROD	20	36	TORINO	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
3	000003	GIOVANNI	VERDI	AMMZ	20	40	ROMA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
4	000004	FRANCO	NERI	DIST	16	45	NAPOLI	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
5	000005	CARLO	ROSSI	DIRE	14	80	MILANO	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
6	000006	LORENZO	GIALLI	DIRE	7	73	GENOVA	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
7	000007	PAOLA	ROSATI	AMMZ	75	40	VENEZIA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
8	000008	MARCO	FRANCO	PROD	20	46	ROMA	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
9	[null]	[null]	[null]	[null]	[null]	[null]	[null]	RICE	RICERCA	VIA VENOSA, 6	MILANO

Full Outer Join

Una Query con Full Outer Join combina i risultati delle due tabelle di Sinistra e Destra tenendo conto di tutte le righe delle tabelle, anche di quelle che non hanno corrispondenza tra di loro.

Il risultato di un Full Outer Join per le tabelle IMPIEGATI e DIPARTIMENTI **contiene tutte** le righe della tabella left IMPIEGATI, estraendo dalla tabella right DIPARTIMENTI solamente le righe che trovano corrispondenza nella regola di confronto della join;

inoltre

verranno estratti tutti i record della tabella di left IMPIEGATI che non trovano corrispondenza nella tabella right DIPARTIMENTI con NULL come valori di tutte le colonne della tabella DIPARTIMENTI

e

tutti i record della tabella di destra right DIPARTIMENTI che non trovano corrispondenza nella tabella di sinistra left IMPIEGATI con NULL come valori di tutte le colonne della tabella IMPIEGATI.

Full Outer Join

```
127 SELECT * FROM Impiegati FULL OUTER JOIN Dipartimenti
128 ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;
129
```

Data Output Explain Messages Notifications

	matricola	nome	cognome	id_dip	ufficio	stipendio	citta	id_dip	nome_d	dip_ind	citta_d
	character	character	character	character	numeric	numeric	character	character	character varying (20)	character varying (50)	character
1	000001	MARIO	ROSSI	AMMZ	10	45	MILANO	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
2	000002	CARLO	BIANCHI	PROD	20	36	TORINO	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
3	000003	GIOVAN...	VERDI	AMMZ	20	40	ROMA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
4	000004	FRANCO	NERI	DIST	16	45	NAPOLI	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
5	000005	CARLO	ROSSI	DIRE	14	80	MILANO	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
6	000006	LOREN...	GIALLI	DIRE	7	73	GENOVA	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
7	000007	PAOLA	ROSATI	AMMZ	75	40	VENEZIA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
8	000008	MARCO	FRANCO	PROD	20	46	ROMA	PROD	PRODUZIONE	P.LE LAVATER, 3	TORINO
9	000009	MATTEO	GAETA	[null]	20	46	SALERNO	[null]	[null]	[null]	[null]
10	[null]	[null]	[null]	[null]	[null]	[null]	[null]	RICE	RICERCA	VIA VENOSA, 6	MILANO

Full Outer Join

```

127 SELECT * FROM Impiegati FULL OUTER JOIN Dipartimenti
128 ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip;
129

```

Data Output Explain Messages Notifications

	matricola	nome	cognome	id_dip	ufficio	stipendio	citta	id_dip	nome_d	dip_ind	citta_d
	character	character	character	character	numeric	numeric	character	character	character varying (20)	character varying (50)	character
1	000001	MARIO	ROSSI	AMMZ	10	45	MILANO	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
2	000002	CARLO	BIANCHI	PROD	20	36	TORINO	PROD	PRODUZIONE	P.L. LAVATER, 3	TORINO
3	000003	GIOVAN...	VERDI	AMMZ	20	40	ROMA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
4	000004	FRANCO	NERI	DIST	16	45	NAPOLI	DIST	DISTRIBUZIONE	VIA SEGRE, 9	ROMA
5	000005	CARLO	ROSSI	DIRE	14	80	MILANO	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
6	000006	LOREN...	GIALLI	DIRE	7	73	GENOVA	DIRE	DIREZIONE	VIA TITO LIVIO, 27	MILANO
7	000007	PAOLA	ROSATI	AMMZ	75	40	VENEZIA	AMMZ	AMMINISTRAZIONE	VIA TITO LIVIO, 27	MILANO
8	000008	MARCO	FRANCO	PROD	20	46	ROMA	PROD	PRODUZIONE	P.L. LAVATER, 3	TORINO
9	000009	MATTEO	GAETA	[null]	20	46	SALERNO	[null]	[null]	[null]	[null]
10	[null]	[null]	[null]	[null]	[null]	[null]	[null]	RICE	RICERCA	VIA VENOSA, 6	MILANO

Full Outer Join

Alcuni database (MySQL) non supportano direttamente questa funzionalità, ma la si può emulare attraverso la combinazione di left e right outer join per utilizzando UNION.

SELECT * FROM Impiegati LEFT JOIN Dipartimenti

ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip

UNION

SELECT * FROM Impiegati RIGHT JOIN Dipartimenti

ON IMPIEGATI.id_dip=DIPARTIMENTI.id_dip

WHERE Impiegati.id_dip IS NULL;

Full Outer Join

Allo stesso modo senza utilizzare nemmeno la query di **RIGHT OUTER JOIN**:

SELECT Impiegati.*, Dipartimenti.*

**FROM Impiegati LEFT JOIN Dipartimenti ON
Impiegati.id_dip=Dipartimenti.id_dip**

UNION

SELECT Impiegati.*, Dipartimenti.*

**FROM Dipartimenti LEFT JOIN Impiegati ON
Impiegati.id_dip=Dipartimenti.id_dip**

WHERE Impiegati.id_dip IS NULL;

BASI DI DATI

Materiale utilizzato e bibliografia

➤ Le slide utilizzate dai docenti per le attività frontali sono in gran parte riconducibili e riprese dalle slide originali (con alcuni spunti parziali ripresi dai libri indicati) realizzate da:

- ✓ autori del libro Basi di Dati (Atzeni e altri) testo di riferimento del corso Basi di Dati e sono reperibili su internet su molteplici link oltre che laddove indicato dagli stessi autori del libro;
- ✓ Prof.ssa Tiziana Catarci e dal dott. Ing. Francesco Leotta – corso di Basi di Dati dell'Università degli Studi La Sapienza di Roma al seguente link ed altri: <http://www.dis.uniroma1.it/~catarci/basidatGEST.html> (molto Interessanti anche le lezioni su YouTube).
- ✓ Proff. Luca Allulli e Umberto Nanni, Libro Fondamenti di basi di dati, editore HOEPLI (testo di facile lettura ed efficace).

➤ Diverse slide su specifici argomenti utilizzate dai docenti per le attività frontali sono anche in parte riconducibili e riprese dalle slide originali – facilmente reperibili e accessibili su internet - realizzate da:

Prof.ssa Roberta Aiello – corso Basi di Dati dell'Università di Salerno

Prof. Dario Maio - corso Basi di Dati dell'Università di Bologna al seguente link ed altri: <http://bias.csr.unibo.it/maio>

Prof. Marco Di Felice - corso Basi di Dati dell'Università di Bologna al seguente link ed altri: <http://www.cs.unibo.it/difelice/dbsi/>

Prof. Marco Maggini e prof. Franco Scarselli - corso Basi di Dati dell'Università di Siena ai seguenti link ed altri: [http://staff.icar.cnr.it/pontieri/didattica/LabSI/lezioni/_preliminari-DB1%20\(Maggini\).pdf](http://staff.icar.cnr.it/pontieri/didattica/LabSI/lezioni/_preliminari-DB1%20(Maggini).pdf)

Prof.ssa Raffaella Gentilini - corso Basi di Dati dell'Università di Perugia al seguente link ed altri: <http://www.dmi.unipg.it/raffaella.gentilini/BD.htm>

Prof. Enrico Giunchiglia - corso Basi di Dati dell'Università di Genova al seguente link ed altri: <http://www.star.dist.unige.it/~enrico/BasiDiDati/>

Prof. Maurizio Lenzerini - corso Basi di Dati dell'Università degli Studi La Sapienza di Roma al seguente link ed altri: <http://didattica.info.altervista.org/Quinta/Database2.pdf>

➤ The PostgreSQL Global Development Group - PostgreSQL nn.xx Documentation

➤ PostgreSQL (appendice - scaricabile dal sito del libro (area studenti) e www.postgresql.org)