




28. Il comando select e l'algebra relazionale

parte seconda

database

il comando select e l'algebra relazionale

Abbiamo detto che le operazioni di interrogazione in SQL, vengono realizzate per mezzo del costrutto **SELECT-FROM-WHERE**:

SELECT <lista_campi>  Quali campi deve visualizzare ?
FROM <lista_tabelle>  Da quali tabelle preleva i dati da visualizzare ?
[**WHERE** <condizione>];  In base a quali criteri sceglie le righe da visualizzare ?

In pratica, questo comando produce una tabella risultato (temporanea) selezionando, tra le righe che appartengono al prodotto cartesiano delle tabelle elencate nella clausola **FROM**, quelle che soddisfano le condizioni espresse nell'argomento della clausola **WHERE**, le cui colonne sono elencate nella clausola **SELECT**.

Sia le colonne della clausola **SELECT** che le tabelle della clausola **FROM** possono essere ridenominate tramite l'uso della parola chiave **AS**.

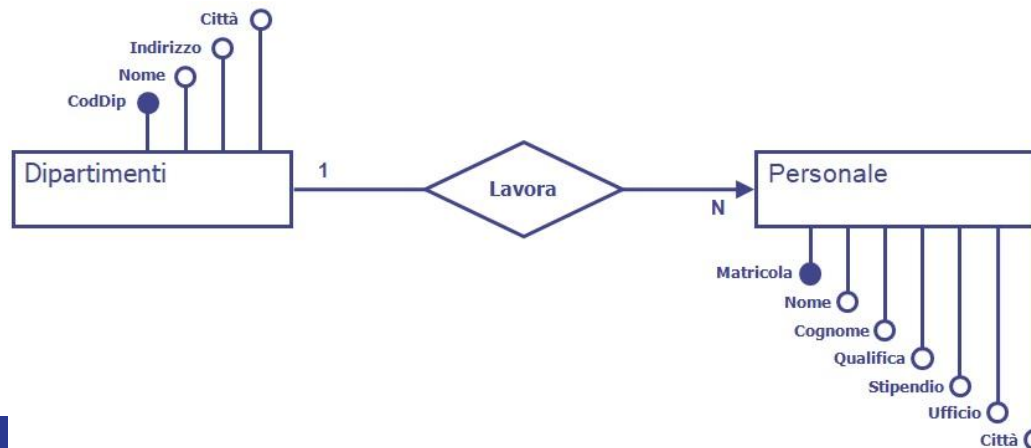
il comando select e l'algebra relazionale

Possiamo analizzarne il funzionamento con un semplice esempio:

Un'azienda è organizzata in dipartimenti aventi ciascuno un certo numero di dipendenti; in ogni dipartimento lavorano più dipendenti, mentre ogni dipendente afferisce a un solo dipartimento, per cui sussiste una associazione 1:N da *Dipartimenti* a *Personale*.

Vediamo un possibile schema E/R col relativo modello relazionale:

Dipartimenti (CodDip, Nome, Indirizzo, Città);
Personale (Matricola, CodDip, Cognome, Nome, Qualifica, Stipendio, Ufficio, Città);



il comando select e l'algebra relazionale

Per estrarre ad es. lo stipendio degli impiegati che hanno la qualifica di 'Tecnico', si deve formulare la seguente query:

```
SELECT Stipendio AS Salario
FROM Personale WHERE Qualifica = 'Tecnico';
```

NB: Nella tabella risultato, il nome del campo *Stipendio* è stato cambiato in *Salario*.

← →				▼	Matricola	▲ 1	CodDip	Cognome	Nome	DataNasc	Qualifica	Stipendio	Ufficio	Citta	
<input type="checkbox"/>		Modifica		Copia		Elimina	AB001	D1	Rossi	Mario	1989-02-01	Tecnico	25000.00	10	Catanzaro
<input type="checkbox"/>		Modifica		Copia		Elimina	AB002	D1	Bianchi	Carlo	1992-12-12	Programmatore	32400.00	20	Cosenza
<input type="checkbox"/>		Modifica		Copia		Elimina	CD001	D2	Verdi	Giovanni	2000-11-07	Analista	28000.00	20	Reggio Calabria
<input type="checkbox"/>		Modifica		Copia		Elimina	CD002	D2	Verdi	Franco	1996-09-02	Tecnico	26000.00	16	Catanzaro
<input type="checkbox"/>		Modifica		Copia		Elimina	CD003	D2	Rossi	Carlo	2001-04-01	Impiegato	22000.00	14	Cosenza
<input type="checkbox"/>		Modifica		Copia		Elimina	EF001	D3	Verdi	Giovanni	1982-10-05	Impiegato	23000.00	7	Reggio Calabria
<input type="checkbox"/>		Modifica		Copia		Elimina	EF002	D3	Rosati	Paola	1995-08-13	Impiegato	24000.00	75	Crotone
<input type="checkbox"/>		Modifica		Copia		Elimina	EF003	D3	Franco	Marco	1999-07-03	Programmatore	30000.00	20	Catanzaro
<input type="checkbox"/>		Modifica		Copia		Elimina	GH001	D4	Tedesco	Matteo	2001-06-13	Programmatore	32000.00	12	Cosenza
<input type="checkbox"/>		Modifica		Copia		Elimina	LM001	D2	Ponsi	Gianni	2002-04-09	Analista	38000.00	7	Cosenza
<input type="checkbox"/>		Modifica		Copia		Elimina	LM002	D3	Vacchi	Marco	1996-04-03	Programmatore	35000.00	16	Cosenza
<input type="checkbox"/>		Modifica		Copia		Elimina	LM003	D2	Segni	Carlo	1998-04-14	Programmatore	28800.00	7	Napoli
<input type="checkbox"/>		Modifica		Copia		Elimina	NP001	D3	Gessi	Giovanni	2003-04-13	Tecnico	25000.00	10	Catanzaro
<input type="checkbox"/>		Modifica		Copia		Elimina	NP002	D2	Voci	Claudio	1999-03-21	Impiegato	18500.00	20	Messina
<input type="checkbox"/>		Modifica		Copia		Elimina	ZZ001	D2	Palmi	Pedi	1993-04-21	Analista	18000.00	14	Catanzaro
<input type="checkbox"/>		Modifica		Copia		Elimina	ZZ002	D4	Ponsi	Mario	1994-05-11	Programmatore	38000.00	7	Reggio Calabria

+ Opzioni									
									Salario
<input type="checkbox"/>		Modifica		Copia		Elimina	25000		
<input type="checkbox"/>		Modifica		Copia		Elimina	26000		
<input type="checkbox"/>		Modifica		Copia		Elimina	25000		

NB: Naturalmente, il tempo di vita del campo virtuale *Salario* si esaurisce con l'esecuzione della query stessa e non viene memorizzato nel database; senza l'uso della parola chiave **AS** il DBMS avrebbe assegnato un nome fittizio a tale campo.

il comando select e l'algebra relazionale

La clausola **SELECT** specifica i campi della tabella risultato; come argomento di questa clausola può anche comparire il carattere “ * ” (asterisco) che rappresenta la selezione di tutti i campi delle tabelle elencate nella clausola **FROM**.

Ad es. per ottenere tutte le informazioni dei dipendenti che hanno la qualifica di tecnico, possiamo modificare la query precedente come segue:

```
SELECT *  
FROM Personale  
WHERE Qualifica = 'Tecnico';
```

Estrai tutti i dati del personale 'Tecnico'.

← T →				Matricola	CodDip	Cognome	Nome	DataNasc	Qualifica	Stipendio	Ufficio	Citta			
<input type="checkbox"/>		Modifica		Copia		Elimina	AB001	D1	Rossi	Mario	1989-02-01	Tecnico	25000	10	Catanzaro
<input type="checkbox"/>		Modifica		Copia		Elimina	CD002	D2	Verdi	Franco	1996-09-02	Tecnico	26000	16	Catanzaro
<input type="checkbox"/>		Modifica		Copia		Elimina	NP001	D4	Gessi	Giovanni	2003-04-13	Tecnico	25000	10	Catanzaro

il comando select e l'algebra relazionale

Nella clausola **SELECT** possono anche comparire generiche espressioni aritmetiche sul valore degli attributi di ciascuna riga selezionata.

In questo modo, se volessimo ricavare nome, cognome e stipendio mensile (riportato invece nella tabella *Personale* come stipendio annuale) dei dipendenti con qualifica di programmatore, potremmo usare la query seguente:

```
SELECT Cognome, Nome, Stipendio/12 AS Mensile  
FROM Personale  
WHERE Qualifica = 'Programmatore'
```

In ogni riga del risultato, il valore del campo *Stipendio* viene diviso per 12 e riportato col nome *Mensile*.

				Cognome	Nome	Mensile
<input type="checkbox"/>		Modifica		Copia		Elimina
		Bianchi		Carlo		2700.000000
<input type="checkbox"/>		Modifica		Copia		Elimina
		Franco		Marco		2500.000000
<input type="checkbox"/>		Modifica		Copia		Elimina
		Tedesco		Matteo		2666.666667
<input type="checkbox"/>		Modifica		Copia		Elimina
		Vacchi		Marco		2916.666667
<input type="checkbox"/>		Modifica		Copia		Elimina
		Segni		Carlo		2400.000000
<input type="checkbox"/>		Modifica		Copia		Elimina
		Ponsi		Mario		3166.666667

il comando select e l'algebra relazionale

Così, per ottenere un elenco relativo a matricola, cognome ed età dei dipendenti del dipartimento D1, si può realizzare la seguente query:







SELECT

YEAR(CURRENT_DATE()) - YEAR(DataNasc) **AS** eta

FROM Personale

WHERE CodDip = 'D1'

Matricola.

+ Opzioni				Matricola	Cognome	eta
← T →						
<input type="checkbox"/>		Modifica		Copia		Elimina
				AB001	Rossi	33
<input type="checkbox"/>		Modifica		Copia		Elimina
				AB002	Bianchi	30

La funzione YEAR() restituisce l'anno relativo alla data usata come argomento, data restituita dalla funzione CURRENT_DATE(); in *Microsoft Access* usiamo la funziona NOW() al posto di CURRENT_DATE().

NB: Si poteva usare anche: **DATEDIFF**(CURRENT_DATE(), DataNasc)

Mentre per visualizzare matricola, cognome, stipendio e un'eventuale gratifica del 10% da corrispondere ai dipendenti del dipartimento D3 con qualifica di impiegato:

SELECT Matricola, Cognome, Stipendio, Stipendio*0.1 **AS** Gratifica

FROM Personale

WHERE CodDip = 'D3' AND Qualifica='Impiegato'

il comando select e l'algebra relazionale

Quando si desidera formulare una query che coinvolge dati appartenenti a più di una tabella, si pone come argomento della clausola **FROM** l'insieme delle tabelle alle quali si vuole accedere.

Come già visto in precedenza, se omettiamo di riportare la clausola **WHERE** otteniamo il prodotto cartesiano delle tabelle indicate nella clausola **FROM**.

Sul prodotto cartesiano così ottenuto, verranno poi applicate le condizioni specificate nella clausola **WHERE**, se presente.

Quindi, un'operazione di join può essere realizzata indicando in modo esplicito, nella clausola WHERE, le condizioni che esprimono il legame tra le diverse tabelle, cioè l'uguaglianza dei valori del campo (o dei campi) in comune tra le tabelle interessate.

il comando select e l'algebra relazionale

Ad es. per estrarre i nomi degli impiegati insieme con le città in cui sono localizzati i dipartimenti dove questi lavorano, dobbiamo recuperare dati registrati nelle tabelle *Personale* e *Dipartimenti*:

```
SELECT Personale.Cognome, Personale.Nome,  
         Dipartimenti.Citta  
FROM Dipartimenti, Personale  
WHERE Dipartimenti.CodDip=Personale.CodDip
```

Condizione di join

Cognome	Nome	Citta
Rossi	Mario	Catanzaro
Bianchi	Carlo	Catanzaro
Verdi	Giovanni	Cosenza
Verdi	Franco	Cosenza
Rossi	Carlo	Cosenza
Ponsi	Gianni	Cosenza
Segni	Carlo	Cosenza
Voci	Claudio	Cosenza
Palmi	Pedi	Cosenza
Verdi	Giovanni	Reggio Calabria
Rosati	Paola	Reggio Calabria
Franco	Marco	Reggio Calabria
Vacchi	Marco	Reggio Calabria
Gessi	Giovanni	Reggio Calabria
Tedesco	Matteo	Catanzaro
Ponsi	Mario	Catanzaro

NB: Si nota l'uso dell'operatore *punto* per identificare le tabelle da cui vengono estratti i campi.


È necessario specificare il nome della tabella quando le tabelle nella clausola **FROM** presentano più campi con lo stesso nome; qualora non vi sia possibilità di ambiguità, è possibile specificare il campo senza dichiarare la tabella di appartenenza.

il comando select e l'algebra relazionale

I campi per i quali può sorgere ambiguità sono *Nome* e *Città*, per cui l'interrogazione precedente può anche essere espressa facendo uso degli **alias** per le tabelle, allo scopo di abbreviare i riferimenti ad esse.

Un alias stabilisce una sorta di nome alternativo per le tabelle per fare riferimento ad esse nell'ambito di una interrogazione.

```
SELECT P.Cognome, P.Nome, D.Città
FROM Dipartimenti AS D, Personale AS P
WHERE D.CodDip = P.CodDip
```



In pratica, è come se venissero realizzate due diverse tabelle “D” e “P” contenenti rispettivamente le stesse righe delle tabelle *Dipartimenti* e *Personale*.

Cognome	Nome	Città
Rossi	Mario	Catanzaro
Bianchi	Carlo	Catanzaro
Verdi	Giovanni	Cosenza
Verdi	Franco	Cosenza
Rossi	Carlo	Cosenza
Ponsi	Gianni	Cosenza
Segni	Carlo	Cosenza
Voci	Claudio	Cosenza
Palmi	Pedi	Cosenza
Verdi	Giovanni	Reggio Calabria
Rosati	Paola	Reggio Calabria
Franco	Marco	Reggio Calabria
Vacchi	Marco	Reggio Calabria
Gessi	Giovanni	Reggio Calabria
Tedesco	Matteo	Catanzaro
Ponsi	Mario	Catanzaro

il comando select e l'algebra relazionale

Utilizzando gli alias è possibile fare riferimento a più esemplari della stessa tabella.

Tutte le volte che si introduce un alias per una tabella, si dichiara in effetti una variabile che rappresenta le righe della tabella di cui è alias.

Quando una tabella compare una sola volta in una interrogazione, non c'è differenza tra l'interpretare l'alias come uno pseudonimo della tabella o come una nuova variabile.

Quando una tabella compare invece più volte (come ad es. nel caso di un *self join*), è necessario considerare l'alias come una nuova variabile.

il comando select e l'algebra relazionale

Ad es. per estrarre le persone che hanno omonimi:

```
SELECT P1.*  
FROM Personale AS P1, Personale AS P2  
WHERE P1.Cognome=P2.Cognome  
      AND P1.Nome=P2.Nome  
      AND P1.Matricola<>P2.Matricola
```

Matricola	CodDip	Cognome	Nome	DataNasc	Qualifica	Stipendio	Ufficio	Citta
AB001	D1	Rossi	Mario	1989-02-01	Tecnico	25000.00	10	Catanzaro
AB002	D1	Bianchi	Carlo	1992-12-12	Programmatore	32400.00	20	Cosenza
CD001	D2	Verdi	Giovanni	2000-11-07	Analista	28000.00	20	Reggio Calabria
CD002	D2	Verdi	Franco	1996-09-02	Tecnico	26000.00	16	Catanzaro
CD003	D2	Rossi	Carlo	2001-04-01	Impiegato	22000.00	14	Cosenza
EF001	D3	Verdi	Giovanni	1982-10-05	Impiegato	23000.00	7	Reggio Calabria

Matricola	CodDip	Cognome	Nome	DataNasc	Qualifica	Stipendio	Ufficio	Citta
EF002	D3	Rosati	Paola	1995-08-13	Impiegato	24000.00	75	
EF003	D3	Franco	Marco	1999-07-03	Programmatore	30000.00	20	
GH001	D4	Tedesco	Matteo	2001-06-13	Programmatore	32000.00	12	
LM001	D2	Ponsi	Gianni	2002-04-09	Analista	38000.00	7	
LM002	D3	Vacchi	Marco	1996-04-03	Programmatore	35000.00	16	

il comando select e l'algebra relazionale

La clausola **WHERE** usa come argomento un'espressione booleana costituita, nel caso più semplice, dal nome di un campo, il cui valore viene messo a confronto (tramite gli operatori classici di confronto) con un valore costante o con un'altra espressione.

L'argomento della clausola **WHERE** può essere composto da più espressioni booleane insieme con gli operatori **AND**, **OR** e **NOT**.

La sintassi assegna la precedenza alla valutazione dell'operatore **NOT** ma non definisce una relazione di precedenza tra gli operatori **AND** e **OR**; se necessario, conviene esplicitare l'ordine di valutazione mediante l'uso delle parentesi.

il comando select e l'algebra relazionale

Estrarre nome, cognome e qualifica dei dipendenti con qualifica di tecnico che lavorano nell'ufficio n. 20:

```
SELECT Nome, Cognome, Qualifica  
FROM Personale  
WHERE Qualifica = 'Programmatore' AND Ufficio = '20';
```

+ Opzioni

					Nome	Cognome	Qualifica
<input type="checkbox"/>		Modifica		Copia		Elimina	Carlo Bianchi Programmatore
<input type="checkbox"/>		Modifica		Copia		Elimina	Marco Franco Programmatore

il comando select e l'algebra relazionale

Estrarre nome, cognome e qualifica dei dipendenti con qualifica di programmatore o di analista:

```
SELECT Nome, Cognome, Qualifica
FROM Personale
WHERE Qualifica = 'Programmatore' OR
        Qualifica = 'Analista';
```

+ Opzioni

← T →			Nome	Cognome	Qualifica				
<input type="checkbox"/>		Modifica		Copia		Elimina	Carlo	Bianchi	Programmatore
<input type="checkbox"/>		Modifica		Copia		Elimina	Giovanni	Verdi	Analista
<input type="checkbox"/>		Modifica		Copia		Elimina	Marco	Franco	Programmatore
<input type="checkbox"/>		Modifica		Copia		Elimina	Matteo	Tedesco	Programmatore
<input type="checkbox"/>		Modifica		Copia		Elimina	Gianni	Ponsi	Analista
<input type="checkbox"/>		Modifica		Copia		Elimina	Marco	Vacchi	Programmatore
<input type="checkbox"/>		Modifica		Copia		Elimina	Carlo	Segni	Programmatore
<input type="checkbox"/>		Modifica		Copia		Elimina	Pedi	Palmi	Analista
<input type="checkbox"/>		Modifica		Copia		Elimina	Mario	Ponsi	Programmatore

il comando select e l'algebra relazionale

Estrarre nome, cognome e qualifica dei dipendenti con cognome Bianchi e che abbiano la qualifica di programmatore o di analista:

```
SELECT Nome, Cognome, Qualifica
FROM Personale
WHERE Cognome = 'Bianchi' AND
  (Qualifica = 'Programmatore' OR Qualifica = 'Analista');
```



... poi l'operatore **AND** (selezionando tra queste ultime, le righe in cui il cognome del dipendente è 'Bianchi').

L'ordine di valutazione imposto dalle parentesi prevede che venga valutato prima l'operatore **OR** (selezionando tutte le righe relative ai programmatori oppure agli analisti) ...

+ Opzioni

	Nome	Cognome	Qualifica
<input type="checkbox"/>  Modifica  Copia  Elimina	Carlo	Bianchi	Programmatore

NB: Senza le parentesi, la query avrebbe estratto i dipendenti con cognome 'Bianchi' e qualifica 'Programmatore' oppure i dipendenti con qualifica 'Analista'.

il comando select e l'algebra relazionale

Per quanto riguarda le condizioni espresse su campi di tipo stringa il linguaggio SQL mette a disposizione l'operatore **LIKE** che confronta il valore di un attributo di tipo stringa di caratteri con un modello di stringa che può contenere caratteri jolly.

I caratteri jolly sono:

- “ _ ” (*underscore*) per indicare un singolo carattere qualsiasi in quella posizione della stringa;
- “ % ” (*percento*) per indicare una sequenza qualsiasi di caratteri in quella posizione della stringa;

il comando select e l'algebra relazionale

- `LIKE "R%"`: vengono ricercate tutte le stringhe che iniziano con il carattere "R";
- `LIKE "%a"`: serve per ricercare tutte le stringhe che terminano con il carattere "a";
- `LIKE "%eta%"`: per le stringhe che contengono al loro interno i caratteri "eta";
- `LIKE "_ossi"`: controlla le stringhe di 4 caratteri che finiscono con "ossi".

NB: L'operatore **LIKE** utilizzato con un modello di stringa che non contiene caratteri jolly è del tutto equivalente all'operatore di uguaglianza "`=`".

Volendo estrarre tutti i dati di tutti i dipendenti il cui cognome inizia per «Ros» («Rossi», e «Rosati») si usa la seguente query:

```
SELECT *  
FROM Personale  
WHERE Cognome LIKE 'Ros%';
```

+ Opzioni															
			Matricola	CodDip	Cognome	Nome	DataNasc	Qualifica	Stipendio	Ufficio	Citta				
<input type="checkbox"/>		Modifica		Copia		Elimina	AB001	D1	Rossi	Mario	1989-02-01	Tecnico	25000.00	10	Catanzaro
<input type="checkbox"/>		Modifica		Copia		Elimina	CD003	D2	Rossi	Carlo	2001-04-01	Impiegato	22000.00	14	Cosenza
<input type="checkbox"/>		Modifica		Copia		Elimina	EF002	D3	Rosati	Paola	1995-08-13	Impiegato	24000.00	75	Crotone

il comando select e l'algebra relazionale

Allo stesso modo, la query:

```
SELECT *  
FROM Personale  
WHERE Nome LIKE '%aria';
```

permette di estrarre i dati di coloro in cui il contenuto del campo *Nome* termina per «aria» (Maria, Ilaria, ecc.).

Infine la query:

```
SELECT *  
FROM Personale  
WHERE Cognome LIKE '%erto%';
```


permette di estrarre i dati di coloro in cui il campo *Cognome* contenga in un punto qualsiasi la sequenza «erto» (Roberto, Bertolini, ecc.).

il comando select e l'algebra relazionale

Estrarre tutti i dati dei dipendenti che hanno un cognome con una "o" in seconda posizione e termina per "i":

```
SELECT *  
FROM Personale  
WHERE Cognome LIKE '_o%i';
```

NB: Utilizzando *Microsoft Access*, bisogna usare il carattere jolly "?" al posto di "_" e "*" al posto di "%".



Personale		Dipartimenti		Query_9					
Matricola	CodDip	Cognome	Nome	DataNasc	Qualifica	Stipendio	Ufficio	Città	
AB001	D1	Rossi	Mario	01/02/1989	Tecnico	€ 25.000,00	10	Catanzaro	
CD003	D2	Rossi	Carlo	01/04/2001	Impiegato	€ 22.000,00	14	Cosenza	
EF002	D3	Rosati	Paola	13/08/1995	Impiegato	€ 24.000,00	75	Crotone	
*									

il comando select e l'algebra relazionale

Un'ultima considerazione riguarda il valore NULL.

Come abbiamo accennato in precedenza, un valore nullo (**NULL**) in un attributo può significare che quell'attributo non è applicabile, o che il valore è applicabile ma non è conosciuto, o anche che non si sa quale delle due situazioni vale.

Quando una riga di una tabella presenta un valore NULL per un certo attributo, vuol dire semplicemente che non è stato registrato alcun valore per quell'attributo, e quindi non va confuso col valore '0', nel caso si tratti di un campo numerico, o con la stringa vuota, nel caso sia di tipo stringa o carattere.

il comando select e l'algebra relazionale

Il predicato **IS NULL** confronta il valore di un campo con il valore NULL; l'uso di questo predicato è il solo modo per controllare la presenza del valore NULL in un campo.

Ad es. la query che segue:

```
SELECT *  
FROM Personale  
WHERE Qualifica IS NULL;
```

equivalente a:

```
SELECT *  
FROM Personale  
WHERE ISNULL(Qualifica);
```

permette la selezione delle righe della tabella *Personale* relative agli impiegati a cui non sia stata assegnata una qualifica specifica.

Ovviamente, per selezionare invece le righe della tabella relative agli impiegati a cui è già stata assegnata una qualifica specifica, bisogna utilizzare il predicato **IS NOT NULL**.