



PEGASO
Università Telematica



Indice

1. INTRODUZIONE.....	3
2. CLAUSOLA WHERE	4
2.1 INTERROGAZIONI DI ESEMPIO	5
3. OPERATORE LIKE.....	7
4. GESTIONE OPERATORI NULLI	8
5. RIEPILOGO	10
BIBLIOGRAFIA	11

1. Introduzione

In questa dispensa illustriamo in modo approfondito la clausola WHERE, di fondamentale importanza nel momento in cui si vogliono inserire filtri al recupero dei dati da un DBMS. Successivamente si passa ad illustrare l'operatore LIKE il quale consente di fare un ulteriore passo in avanti, consentendo di inserire filtri che riguardano le stringhe, con tutte le varianti date dai caratteri speciali. Infine, si espone un punto delicato ossia la gestione dei valori NULL i quali vanno comunque gestiti con attenzione attraverso l'operatore IS NULL. Ovviamente si suggerisce al lettore di approfondire tali concetti anche sul libro di testo.

2. Clausola WHERE

La clausola WHERE viene utilizzata per filtrare i record. Solo i record che soddisfano una condizione specificata. Ammette come argomento una espressione booleana costruita combinando predicati semplici attraverso i seguenti gli operatori logici AND, OR e NOT. Inoltre, ciascun predicato usa gli operatori: =, <, >, <>, <= e >= per confrontare espressioni, le quali nel caso più semplice, sono rappresentate dal nome di un attributo. Si possono avere i seguenti casi di utilizzo:

- Predicati separati dall'operatore **AND**:
 - Sono selezionate le righe per cui **TUTTI** i predicati sono veri
 - Predicati separati dall'operatore **OR**
 - Sono selezionate tutte le righe per cui **ALMENO UNO** dei predicati risulta vero

L'operatore logico **NOT** è unario e inverte il valore di verità del predicato.

Nel caso di espressione con più operatori, l'operatore NOT ha la precedenza. AND e OR sono paritetici (mettere le parentesi). Riprendiamo adesso lo schema composto dalla tabella Impiegato di Figura 1 e svolgiamo su di essa alcune interrogazioni nei prossimi paragrafi.

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Bianchi	Produzione	20	36	Torino
Giovanni	Verde	Amministrazione	20	40	Roma
Franco	Neri	Distribuzione	16	45	Napoli
Carlo	Rossi	Direzione	14	80	Milano
Lorenzo	Gialli	Direzione	7	73	Genova
Paola	Rosati	Amministrazione	75	40	Venezia
Marco	Franco	Produzione	20	46	Roma

Figura 1: tabella Impiegato.

2.1 Interrogazioni di esempio

Per consolidare le conoscenze sulla clausola WHERE, ecco alcune interrogazioni da costruire come validi esempi di utilizzo. Seguiremo la numerazione iniziata nella UDA precedente.

Interrogazione 6: estrarre il nome ed il cognome dalla tabella Impiegati, degli impiegati che lavorano nell'ufficio 20 del dipartimento Amministrazione.

Istruzione SQL:

- SELECT nome,cognome
- FROM Impiegati
- WHERE Nome='Amministrazione' AND Ufficio=20

Il risultato è illustrato in Figura 2.

Nome	Cognome
Giovanni	Verdi

Figura 2: risultato dell'interrogazione 6.

Interrogazione 7: estrarre i nomi ed i cognomi degli impiegati che lavorano nel dipartimento Amministrazione o nel dipartimento Produzione, sempre dalla tabella Impiegati. La soluzione SQL è:

- SELECT Nome, Cognome
- FROM Impiegato
- WHERE Dipartimento ='Amministrazione' OR
- Dipartimento='Produzione'

Il risultato (resource-set) è quello di Figura 3.

Nome	Cognome
Mario	Rossi
Carlo	Bianchi
Giovanni	Verdi
Paola	Rosati
Marco	Franco

Figura 3: resource set risultato della query 7.

Interrogazione 8: estrarre i nomi propri degli impiegati di cognome 'Rossi' che lavorano nei dipartimenti Amministrazione o Produzione.

Istruzione SQL:

- SELECT nome
- FROM Impiegati
- WHERE Cognome ='Rossi' AND (Dipart='Amministrazione OR Dipart='Produzione')

Il resource-set è illustrato in Figura 4.

Nome
Mario

Figura 4: resource-set interrogazione 8.

3. Operatore LIKE

L'operatore LIKE viene utilizzato in una clausola WHERE per cercare una informazione specifica in una colonna. È un operatore di confronto tra stringhe e permette di fare effettuare confronti con stringhe in cui compaiono i caratteri speciali: `_` e `%`. Il primo carattere rappresenta un carattere qualunque. Il secondo carattere rappresenta una stringa di caratteri arbitrari (anche nullo).

Esempio:

- LIKE `'ab%ba_'` indica quindi una stringa che:
 - Inizia per `'ab'`
 - Continua con un insieme arbitrario di caratteri qualunque
 - Termina con `'ab'` + un carattere qualunque. Es. `'abpippobat'`

Esempi

Facciamo ora un paio di esempi.

Interrogazione 9: estrarre gli impiegati che hanno un cognome che ha una `'o'` in seconda posizione e termina per `'i'`.

L'istruzione SQL è la seguente:

- SELECT *
- FROM Impiegato
- WHERE cognome LIKE `'_o%i'`

Il risultato è quello illustrato in Figura 5.

Da notare che MS ACCESS utilizza il carattere `'*'` al posto del carattere `'%'` e il carattere `'?'` al posto del carattere `'_'`.

Nome	Cognome	Dipart	Ufficio	Stipendio	Città
Mario	Rossi	Amministrazione	10	45	Milano
Carlo	Rossi	Direzione	14	80	Milano
Paola	Rosati	Amministrazione	75	40	<u>venezia</u>

Figura 5: risultato della query con LIKE.

L'istruzione SQL: `SELECT * FROM Impiegati WHERE Cognome LIKE 'a%'` darebbe come risultato tutte le righe che hanno l'impiegato con il cognome che inizia per `'a'`. In questo caso quindi insieme vuoto.

4. Gestione operatori nulli

Un valore NULL in un attributo può significare che un certo attributo:

- Non è applicabile
- Il valore è applicabile ma non è conosciuto
- Non si sa quale delle due situazioni precedenti vale

Per selezionare i termini con valori nulli, SQL fornisce il predicato IS NULL:

- Attributo IS [NOT] NULL

Il risultato risulta vero solo se l'attributo ha un valore NULL. Il predicato IS NOT NULL è la sua negazione.

Impiegati

Matricola	Cognome	Filiale	Età
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

Figura 6: la tabella Impiegati.

Prendiamo la seguente tabella di Figura 6.

Vogliamo conoscere gli impiegati la cui età è o potrebbe essere maggiore di 40. L'istruzione SQL è la seguente:

- SELECT *
- FROM Impiegati
- WHERE (Età>40) OR (Età IS NULL)

Il resource-set è quello di Figura 7. Come si può notare, è stato recuperato anche il record con il valore NULL nell'attributo Età.

Matricola	Cognome	Filiale	Età
5998	Neri	Milano	45
9553	Bruni	Milano	NULL

Figura 7: risultato dell'operatore IS NULL

Quindi, La condizione IS NULL è soddisfatta se il termine che precede immediatamente la parola chiave IS specifica uno dei seguenti valori indefiniti:

- Il nome di una colonna che contiene un valore nullo.
- Un'espressione che restituisce null.

Viceversa, se si utilizza l'operatore IS NOT NULL, la condizione è soddisfatta quando la colonna contiene un valore che non è nullo o quando l'espressione che precede immediatamente le parole chiave IS NOT NULL non viene valutata nulla.

5. Riepilogo

In Figura 8 lo schema di un database sul quale proviamo ad effettuare queries un po' più complesse, utilizzando tutte le clausole.

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

Figura 8: schema di database di prova.

Interrogazione: Si vogliono recuperare i nomi delle persone che hanno reddito 26, Madre di nome Anna e padre di nome Luigi. La soluzione è la seguente istruzione:

- SELECT *
- FROM Persone
- WHERE (Età=26 AND Maternità='Anna' AND Paternità='Luigi')

La clausola WHERE presenta tre condizioni ed il risultato è quello illustrato in Tabella 1.

Tabella 1: risultato della query.

Nome	Età	Reddito
Filippo	26	30

Bibliografia

- Atzeni P., Ceri S., Fraternali P., Paraboschi S., Torlone R. (2018). Basi di Dati. McGraw-Hill Education.
- Batini C., Lenzerini M. (1988). Basi di Dati. In Cioffi G. and Falzone V. (Eds). Calderini. Seconda Edizione.