SQL 2 lezione interrogazioni

Francesca Gasparini gasparini@disco.unimib.it

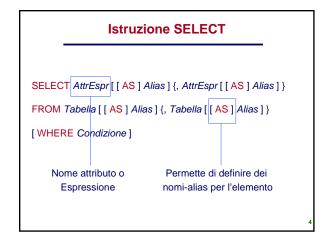
Generalità

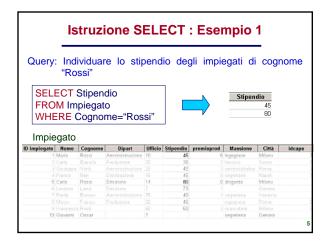
SQL permette di effettuare interrogazioni (query) in modo dichiarativo: si specifica obiettivo di interrogazione, non il modo per effettuarla.

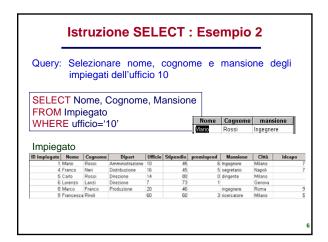
Ogni interrogazione SQL è passata da DBMS a Query Optimizer, che la traduce in linguaggio procedurale interno al sistema per eseguirla in modo efficiente (passaggio trasparente a utente).

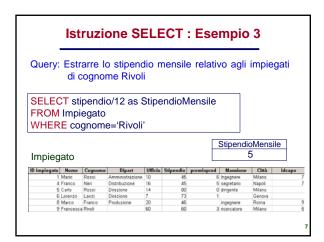
Le query devono essere costruite per essere leggibili e modificabili, non necessariamente efficienti.

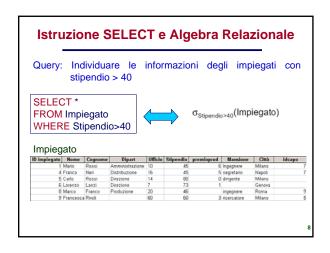
Le interrogazioni avvengono usando l'istruzione SELECT: SELECT ListaAttributi FROM ListaTabelle [WHERE Condizioni] TARGET LIST Cosa si vuole come risultato Dove reperire le informazioni Che condizioni devono essere soddisfatte

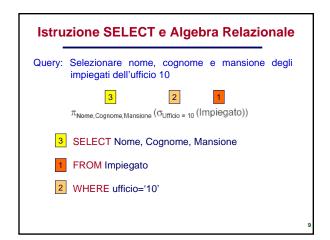










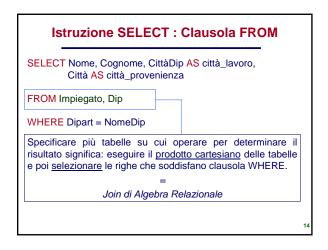


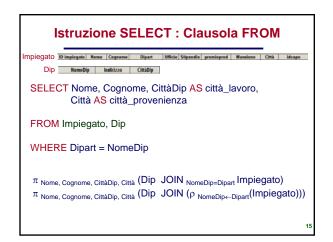


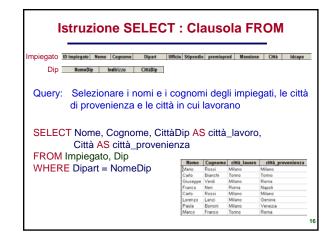


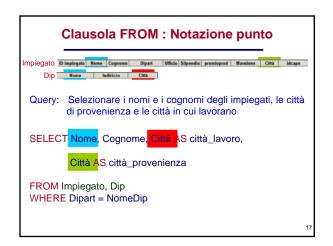


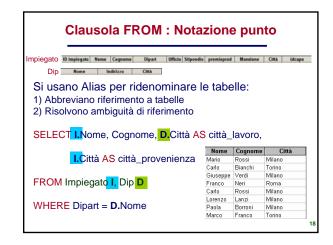


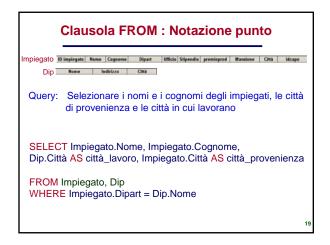








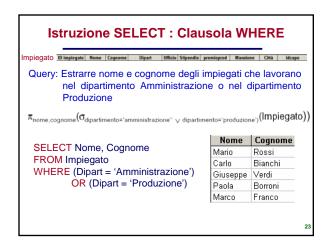














Clausola WHERE: Operatore LIKE

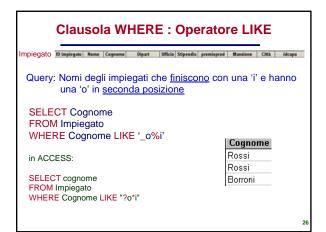
L'operatore LIKE permette di esprimere dei "pattern" su stringhe mediante "wildcard":

(un carattere arbitrario) (in ACCESS ?)

% (una stringa arbitraria) (in ACCESS *)

Query: Nomi degli impiegati che <u>finiscono</u> con una 'i' e hanno una 'o' in <u>seconda posizione</u>

25



Clausola WHERE: Operatore BETWEEN

L'operatore BETWEEN permette di esprimere condizioni di appartenenza a un intervallo

Query: Nome e stipendio degli impiegati che hanno uno stipendio compreso tra 38 e 60 (estremi inclusi)

SELECT Nome, Stipendio FROM Impiegato

WHERE Stipendio BETWEEN 38 AND 60

SELECT Nome, Stipendio FROM Impiegato

WHERE Stipendio>=38 AND Stipendio<=60

 Cognome
 stipendio

 Rossi
 45

 Verdi
 40

 Neri
 45

 Borroni
 40

 Franco
 46

Clausola WHERE: Operatore IN

L'operatore IN permette di esprimere condizioni di

Query: Cognome e ufficio degli impiegati degli uffici 10 e 20

SELECT Cognome, ufficio FROM Impiegato

appartenenza a un insieme

WHERE ufficio IN ('10','20')

SELECT Cognome, ufficio FROM Impiegato

WHERE ufficio = '10' OR ufficio = '20'

28

Cognome ufficio

Bianchi

Franco

20

20

Clausola WHERE: Valori nulli

SQL-2 usa una <u>logica a tre valori</u> per valutare il valore di verità di una clausola WHERE: True (T), False (F), Unknown (?)

- 1) Un predicato semplice valutato su un attributo a valore nullo dà come risultato della valutazione ?.
- Una tupla per cui il valore di verità è ? <u>non</u> viene restituita dalla query.
- Se la valutazione del predicato di un constraint è ? il constraint non è violato

Clausola WHERE : Valori nulli

Il valore di verità di un predicato complesso viene calcolato in base alle seguenti tabelle di verità:

AND T F ?
T T F ?
F F F F
? ? F ?

OR	Т	F	?
Т	Т	Т	Т
F	Т	F	?
?	Т	?	?

NOT
T F
F T
? ?

30

29

Clausola WHERE: Valori nulli

A B C t1 a ? c1 t2 a1 b c2 t3 a2 ? ?

SELECT * FROM Tabella WHERE A=a OR B=b

Il valore di verità della condizione per ogni tupla è il seguente:

t1: $T OR ? \rightarrow T$ t2: $F OR T \rightarrow T$ t3: $F OR ? \rightarrow ?$

31

Clausola WHERE: Valori nulli

A B C t1 a ? c1 t2 a1 b c2 t3 a2 ? ?

SELECT * FROM Tabella WHERE A=a AND B=b

Il valore di verità della condizione per ogni tupla è il seguente:

t1: $T \text{ AND } ? \rightarrow ?$ t2: $F \text{ AND } T \rightarrow F$ t3: $F \text{ AND } ? \rightarrow F$

Nessuna tupla verifica la query.

__

Clausola WHERE: Valori nulli

A B C
t1 a ? c1
t2 a1 b c2
t3 a2 ? ?

SELECT * FROM Tabella WHERE NOT C=c1

Il valore di verità della condizione per ogni tupla è il seguente:

t1: NOT T \rightarrow F t2: NOT F \rightarrow T t3: NOT ? \rightarrow ?

33

Clausola WHERE: Operatore IS [NOT] NULL

Come valutare attributi nulli?

Attributo IS [NOT] NULL

IS NULL: vero se attributo ha valore nullo, falso altrimenti

IS NOT NULL: vero se attributo ha valore specificato, falso altrimenti

Clausola WHERE: Operatore IS NULL

A B C t1 a ? c1 t2 a1 b c2 t3 a2 ? ?

SELECT * FROM Tabella WHERE B IS NULL

Restituisce le tuple t1 e t3

SELECT * FROM Tabella WHERE B IS NULL AND C IS NULL Restituisce la tupla t3

SELECT * FROM Tabella WHERE B IS NULL OR C IS NULL Restituisce le tuple t1 e t3

35

Clausola WHERE: Operatore IS NULL

A B C t1 a ? c1 t2 a1 b c2 t3 a2 ? ?

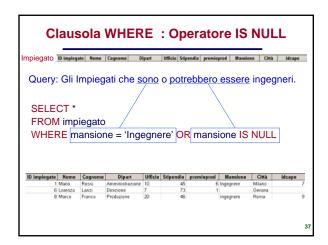
SELECT * FROM Tabella WHERE B IS NOT NULL Restituisce la tupla t2

Si noti che le interrogazioni

SELECT * FROM Tabella WHERE B = 'b' OR B <> 'b' SELECT * FROM Tabella WHERE B IS NOT NULL

sono equivalenti

36



Istruzione SELECT: DISTINCT

In algebra relazionale i risultati delle interrogazioni non contengono elementi duplicati.

In SQL, le tabelle prodotte dalle interrogazioni possono contenere più righe identiche tra loro.

I duplicati possono essere rimossi usando la parola chiave DISTINCT.

SELECT [DISTINCT] AttrExpr FROM Tabella [WHERE Cond]

20

Istruzione SELECT : DISTINCT

Query: Ritornare i cognomi degli impiegati e le città di provenienza.

SELECT DISTINCT Cognome, Città FROM Impiegato WHERE stipendio>40

Il risultato della query SQL contiene righe duplicate:

Cognome	Città	
Franco	Roma	
Lanzi	Genova	
Neri	Napoli	
Rossi	Milano	

Espressioni e Funzioni

I predicati usati nelle interrogazioni possono coinvolgere, oltre a nomi di colonna, anche espressioni.

Le espressioni sono formulate applicando operatori ai valori delle colonne delle tuple.

Esempi di espressioni e funzioni sono quelle aritmetiche, su stringhe, su date e tempi.

Le espressioni possono comparire nelle clausole SELECT, WHERE e UPDATE (si vedrà più avanti)

Espressioni e Funzioni

Una espressione usata nella clausola SELECT dà luogo ad una colonna, detta virtuale, non presente nella relazione su cui si effettua l'interrogazione.

Le colonne virtuali non sono fisicamente memorizzate, ma sono materializzate (esistono) come risultato delle interrogazioni

Anche alle colonne virtuali è possibile assegnare un alias.

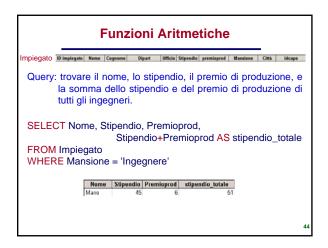
Espressioni e Funzioni

SQL mette a disposizione funzioni predefinite:

su stringhe (SUBSTR, LENGTH, || ,....)
su date, intervalli (+, -, DATE, DAYOFWEEK,...)
matematiche (* , + , - , / , TAN, SQRT, SIN,...)
informazioni di sistema (USER, CURRENT_DATE,...)

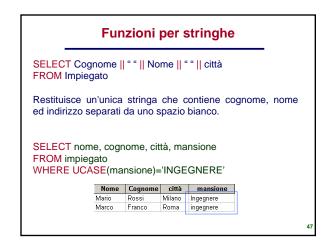
42













Operatore di JOIN

L'operatore di JOIN rappresenta un'importante funzionalità in quanto permette di correlare dati in relazioni diverse.

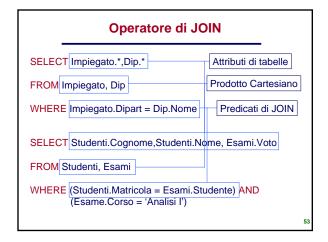
In SQL il JOIN è espresso tramite un prodotto cartesiano tra tabelle a cui sono applicati uno o più predicati di join.

Un predicato di join esprime una condizione che deve essere verificata dalle tuple del risultato dell'interrogazione.









Operatore di JOIN: join esplicito Sintassi alternativa per Join, introdotta da SQL-2: SELECT AttrExpr [[AS] Alias] {, AttrExpr [[AS] Alias] } FROM Tabella [[AS] Alias] { (TipoJoin] JOIN Tabella [[AS] Alias] ON CondizJoin } [WHERE Cond] Esplicita il JOIN e separa le condizioni di JOIN dalle condizioni di query. TipoJoin può essere inner, RIGHT [OUTER], LEFT [OUTER] oppure FULL [OUTER], consentendo la rappresentazione dei join esterni.



