

SQL → linguaggio DICHIARATIVO

INTERROGAZIONE:

Comando Principale è SELECT

~~Un~~ SELECT ~~da~~ From [TABELLE SORGENTI] WHERE [condizioni]

SELECT

Posso Specificare espressioni che determinano gli Attributi Separati da una Virgola

From

dico da Dove faccio la SELECT

WHERE

Condizione BOOLEANA per Selezionare i Valori degli Attributi ma con Alcune Clauses come:

- HAVING

- GROUP BY

- ORDER BY

SELECT mi definisce la Struttura dello Schema Risultante



Π → un proiezione

→ tutti gli Attributi di una Tabella

~~Un~~ SELECT [{EXPRESSION | *}, {...}]

Posso fare anche un ALIAS: (è la RIDENOMINAZIONE)

[{EXPRESSION [[AS] nome_output] | * }]

DISTINCT permette di SELEZIONARE solo le RIGHE con Valori \neq

↓

elimina così i duplicati

→ Visto che SQL non lo fa in Automatico

Dato INSEGNAMENTO (CODICE, NOME, NUM-CREDITI)

CODICE	NOME	NUM-CREDITI
INF01	PROG 1	12
INF02	DB	12
INF03	ANALISI	6

SELECT * FROM Insegnamento; \Rightarrow RESTITUISCE tutto

SELECT Nome FROM Insegnamento \Rightarrow Nome dei corsi

Codice e nome in Maiuscolo:

SELECT Codice, Upper(Nome) FROM Insegnamento

funzioni Comode:

- ⌘ Lower
- ⌘ Upper

CONCATENAZIONE DELLE STRINGHE \Rightarrow [Codice][Nome + CFU]

SELECT codice, nome || 'CFU' || num-creditil FROM Insegnamento

Codice	Nome_Num_Crediti
INF01	progr 1 CFU 12
db	CFU 12
Analisi	CFU 6

Ho Questa parte fissa

CFU

Se volessi dare un Nome Significativo
dovrei mettere AS nome-che-voglio

From Nome_Tabella \rightarrow singola

From Tab1, ..., TabN \rightarrow multipla

Posso Indicare delle SOTTO-QUERY

Se Nel FROM Ho più Tabelle viene fatto il prodotto Cartesiano

From A,B

Avrà dimensione $|A| \cdot |B|$

Ma combinerò Ogni Riga di A con ogni Riga di B

Se ci Sono Attributi con Nome Uguale vengono Visti Come 2 Attributi \neq e Troveremo $[Nome_TAB.Nome_Attributo]$

From Nome_Tab as Nuova_Nome sarà Solo una RIDENOMINAZIONE
all' Interno della QUERY

ESEMPIO:

TAB-A (ID, A₁)

TAB-B (ID, B₁)

■ SELECT * FROM TAB-A, TAB-B ed Avrò N Righe dove
N = |A| · |B| ed ogni Riga ha 4 Colonne

■ SELECT TAB-A.ID, B₁ FROM TAB-A, TAB-B e così ID non è
ambiguo vista che ha Specificato la Tabella dove l'Attributo
è in Comune

■ SELECT A.ID, B₁ FROM TAB-B, TAB-A AS A ed ha Ridenominato
- Tabella per Comodità

JOIN:

Scelgo Solo le RIGHE che mi Soddisfano una CONDIZIONE

■ SELECT * FROM A, B WHERE [Condition]

■ SELECT * FROM A <JOIN-TYPE> ON <JOIN-CONDITION>

ESEMPIO:

IMPIEGATO (MATRICOLA, nome, cognome, reparto)

REPARTO (CODICE, nome, Tel.)

IMPIEGATO.REPARTO → REPARTO

■ A. Impiegato il Nome, Cognome e Telefono del Reparto:

SELECT Cognome, Impiegato.Nome, Reparto.Tel FROM Impiegato, Reparto
WHERE Impiegato.reparto = Reparto.codice

WHERE è Sempre una Condizione BOOLEANA ricavata combinando condizioni Semplici che SONO:

- ATTRIB. = COSTANTE
- AND
- ATTR = ATTR +
- NOT
- OR

Dove Posso Usare tutti gli OPERATORI DI CONFRONTO
Facendo Attenzione che la CLAUSOLA DI ESSERE \neq è $<$ o $>$

SELECT Nome FROM Insegnamento WHERE num-Cfu ≥ 6

CONSIDERANDO Ora: STUD (matricola, Nome, Cognome, Indirizzo, città, media)

MATRICOLA	NOME	COGNOME	INDIRIZZO	CITTÀ	MEDIA
IN01	Mario	Rossi	Via Nobile	Verona	27.50
IN03	Simone	Mattioli	Via Storto	Rovereto	28.6666

Media Nota che non può Essere Ne Numeric Ne decimal

■ SELECT * FROM stud WHERE cognome = 'Mattioli';

■ SELECT * FROM stud WHERE cognome = 'Mattioli' AND città = 'Verona'

NON case Sensitive

Questa Invece è CASE SENSITIVE

Per Evitare ERRORE/problemsi:

'Verona' \neq 'VERONA' \neq 'verona'

LOWER(città) = 'verona' oppure
UPPER(città) = 'VERONA'

SELECT * FROM stud WHERE NOT cognome = 'Mattioli'

Booleani Sono a 3 Valori in SQL: - TRUE
- FALSE

Qualsiasi Condizione mi Restituirà NULL - NULL
a Meno che io non INSERISCA:
IS [NOT] NULL

PATTERN MATCHING:

[A] LIKE [PATTERN]

↓
2 caratteri Speciali:

%. ⇒ φ....N caratteri Qualsiasi

_- ⇒ 1 carattere Qualsiasi

[A] NOT LIKE [PATTERN]

Città LIKE 'V%.R%.' ⇒ Città che Iniziano con V e
che Contengono la lettera R

Città LIKE 'V%.R-' e Trovo le Città che Iniziano con V
e Terminano con R seguito da un carattere Qualsiasi

C'è Anche L' OPERATORE SIMILAR TO

BETWEEN \Rightarrow Testa l'Appartenenza ad un Intervallo, ad Esempio:

- SELECT * FROM Stud WHERE media BETWEEN (21, 27)
- SELECT * FROM Stud WHERE media > 21 AND media < 27

Posso Sempre Negare e quindi avere il NOT BETWEEN

IN \Rightarrow Permette di Verificare se il Valore di un Attributo è Incluso in un Insieme di Valori

- SELECT * FROM Stud WHERE nome IN ('Simone', 'Omer', 'David')

ORDINAMENTO:

Una Volte Otenuto il Risultato, posso Richiedere di averlo rispetto un ORDINAMENTO Ben Specifico tramite la ORDER BY dove Inserisco 1...N Attributi e dire se

- ASCENDENTE
- DECRESCENTE

↓
se ho più Attributi allora a Parte di primo eseguo l'ordinamento sul secondo

AGGREGAZIONE:

Permettono di determinare un Valore considerando i Valori Otenuti dalla SELECT con:

- MIN \rightarrow
- COUNT \rightarrow #righe Risultato
- MAX \rightarrow

- **Avg** →
- **Sum** →

COUNT(*) ⇒ conta Tutte le Righe del Risultato

■ **SELECT count(*)** FROM Stud ⇒ ed è un **CONTATORE**

■ **Count (expression)**

↓
<nome-attribute>

■ **SELECT count(città)** ⇒ righe in cui Attributo città non è null.

■ **Count (DISTINCT <expression>)** ⇒ Conta i Valori, SENZA duplicati dove è Vera l'Espressione

MIN/MAX/AVG

■ **SELECT** \max \min $\frac{\text{avg}}{\text{sum}}$ (media) **FROM** Stud

Sum Somma i Valori Mentre COUNT conta le Righe