

Altri predicati

Wednesday, 17 May 2023

10:42

- WHERE EXISTS (sottoespressione)

- Funzione booleana true/false

- True se la subquery non è vuoto, aka abbiamo almeno 1 riga

- Le sottoquery sono utili quando vogliamo correlare query interna con query esterna

- Esempio:

Studenti che hanno degli omonimi (cioè studenti con stesso nome e cognome ma matricole diverse)

```
SELECT * FROM Studente S1
```

```
WHERE EXISTS (
```

```
    SELECT * FROM Studente S2
```

```
    WHERE S1.nome = S2.nome AND S1.cognome = S2.cognome
```

```
    AND S1.matricola <> S2.matricola )
```

Il fatto che abbiamo una correlazione tra query interna ed esterna si chiama "predicato di correlazione"

Abbiamo una correlazione di 2 query

- Bisogna applicare semantica top-down

- Regole di visibilità:

- La visibilità delle variabili SQL segue la regola che

- Una sottoquery può vedere le variabili della query esterna

- Però la query esterna non può vedere le variabili della query interna

- Query ad uno stesso livello di profondità non possono vedere le variabili a

Esempio:

```
SELECT * FROM Studente JOIN Esame
```

```
WHERE Matricola_st IN (SELECT matricola_studente FROM Esame E1)
```

```
OR Matricola IN (SELECT Matricola_Studente
```

```
    FROM Esame E2
```

```
    Where E1.data = E2.data)
```

E1 non è definita nella seconda query

- Esercizi riepilogativi istruzione select

1) Studente(matricola_st, nome, corso)

Esame(codice_e, matricola_studente, codice_corso, data, voto)

Corso(codice_c, nome, ore_lezioni, ore_esercitazioni, crediti_lezione, crediti_es

PianoDiStudio(Matricola_st, codice_corso)

Personale_docente(matricola_d, cognome, nome, ruolo, classe_stipendio)

Trovare lo studente con la media ponderata più alta.

Ci serve:

esercizi.

- Esame -> codice, matricola_studente, voto
- Corso -> crediti

Noi vogliamo affiancare la nostra tabella, aggiungere colonne

E per farlo dobbiamo fare un join:

```
SELECT * FROM Esame E, Corso C  
WHERE E.codice_corso=C.codice_c
```

Ora dobbiamo trovare Voto e CFU totali

```
SELECT Matricola_studente as MATr, voto, crediti-lezione+crediti_esercitazioni  
FROM Esame E, Corso C  
WHERE E.codice_corso=C.codice_c
```

Ora proviamo a calcolare la media ponderata della matricola 149 (passo temporaneo)

```
SELECT SUM(Voto*(crediti_lezione+crediti_esercitazione))/SUM(Crediti_lezione)  
FROM Esame E, Corso C  
WHERE E.codice_corso=C.codice_c  
AND Matricola_studente = 149
```

Ora da 1 studente N studenti

```
SELECT Matricola_studente,  
        SUM(Voto*(crediti_lezione+crediti_esercitazione))/SUM(Crediti_lezione)  
FROM Esame E, Corso C  
WHERE E.codice_corso=C.codice_c  
GROUP BY Matricola_studente -> Dividiamo per blocchi di studenti
```

Ora proviamo con la media ponderata maggiore di 25

```
SELECT Matricola_studente,  
        SUM(Voto*(crediti_lezione+crediti_esercitazione))/SUM(Crediti_lezione)  
FROM Esame E, Corso C  
WHERE E.codice_corso=C.codice_c  
GROUP BY Matricola_studente  
HAVING media > 25
```

Ora invece maggiore di tutti

```
SELECT Matricola_studente,  
        SUM(Voto*(crediti_lezione+crediti_esercitazione))/SUM(Crediti_lezione)  
FROM Esame E, Corso C  
WHERE E.codice_corso=C.codice_c  
GROUP BY Matricola_studente  
HAVING media >= ALL(
```

```

SELECT SUM(Voto*(crediti_lezione+crediti_esercitazione))/SUM(Crediti_lezione)
FROM Esame E, Corso C
WHERE E.codice_corso=C.codice_c
GROUP BY Matricola_studente
)

```

Con max:

```

SELECT Matricola_studente,
       SUM(Voto*(crediti_lezione+crediti_esercitazione))/SUM(Crediti_lezione)
FROM Esame E, Corso C
WHERE E.codice_corso=C.codice_c
GROUP BY Matricola_studente
HAVING media = (
    SELECT max(media_ponderata)
    FROM (
        SELECT SUM(Voto*(crediti_lezione+crediti_esercitazione))/SUM(Crediti_lezione)
        FROM Esame E, Corso C
        WHERE E.codice_corso=C.codice_c
        GROUP BY Matricola_studente
    ) AS media_voti
)

```

2) Trovare gli studenti con più di 25 in tutti gli esami

```

SELECT DISTINCT Matricola_st, Nome, Cognome
FROM Studente S JOIN Esame E ON E.matricola_studente=S.matricola_st
WHERE Matricola_St NOT IN (
    SELECT Matricola_studente
    FROM Esame
    WHERE Voto < 20
)

```

- Viste in SQL:

- **CREATE VIEW** Nome_Vista **AS** (query)
\-----/-> Crea la lista

- Insert

- Insert into TABELLA (attributi) values (valori) -> permette di inserire singoli valori
Se non specifico attributi, è implicito tutti gli attributi

- DISTINCT nella SELECT permette di prendere solo quelli univoci

- UNION/INTERSECT/MINUS è la ... nel mr

Es.

SELECT A, B FROM R1 UNION SELECT C, D FROM R2

R1

A	B	C

R2

C	D	E

Unione

A	B

**SELECT A,B FROM R1
UNION
SELECT C,D FROM R2**

Intersezione

A	B

**SELECT A,B FROM R1
INTERSECT
SELECT C,D FROM R2**

Differenza

A	B

**SELECT A,B FROM R1
MINUS
SELECT C,D FROM R2**