

BASI DI DATI I

- RECAP su SQL
- Assertion



STRUTTURA DI UN'INTERROGAZIONE

SELECT <elenco attributi e funzioni>

FROM <elenco delle tabelle o viste>

[**WHERE** <condizioni>]

[**GROUP BY** <attributo o attributi di raggruppamento>]

[**HAVING** <condizione di raggruppamento>]

[**ORDER BY** <elenco attributi>]



NELLA CLAUSOLA WHERE

- WHERE attributo op.rel. Valore
- WHERE attributo op.rel. Attributo
- WHERE attributo IS [NOT] NULL
- WHERE attributo IN (vall, val2, ...)
- WHERE attributo BETWEEN vall AND val2
- WHERE attributo LIKE ‘pattern’
- WHERE salario < 30000 (<>, =, <, <=, >, >=)
- WHERE M.salario > E.salario
- WHERE salario IS NOT NULL
- WHERE sesso IN ('M', 'F')
- WHERE salario BETWEEN 300 AND 400
- WHERE nome LIKE '_Mario%



DOVE ANNIDARE LE QUERY

SELECT <elenco attributi e funzioni> NO

FROM <elenco delle tavelle o viste> SI

[**WHERE** <condizioni>] SI

[**GROUP BY** <attributo o attributi di raggruppamento>] NO

[**HAVING** <condizione di raggruppamento>] SI

[**ORDER BY** <elenco attributi>] NO



RAGIONARE SULLA QUERY

- Le query possono essere di diverse complessità
- In tutti i casi, la richiesta è chiara e ci è molto semplice andare ad identificare ciò di cui abbiamo bisogno all'interno dello schema logico
- Risulta però complicato andare a ragionare su COME costruire la query che restituiscia effettivamente l'output richiesto
- È SEMPRE importante andare a ragionare bene sulla richiesta, ed eventualmente trovare un altro modo per riscriverla, in modo che risulti più semplice andare a trasformarla in una query.



ESEMPIO

- Schema logico

STUDENTE (MATRICOLA, NOME COGNOME)
ESAME(MATRICOLA, CORSO)

- Query

Gli studenti che hanno sostenuto almeno tutti gli esami sostenuti dallo studente con matricola 100

PROVIAMO A RISCRIVERLA?



ESEMPIO

- Schema logico

STUDENTE (MATRICOLA, NOME COGNOME)
ESAME(MATRICOLA, CORSO)

- Query

Gli studenti che hanno sostenuto almeno tutti gli esami sostenuti dallo studente con matricola 100

Trovo tutti gli studenti per cui NON ESISTA un esame che hanno sostenuto e che lo studente 100 non ha sostenuto

L'insieme degli esami sostenuti dallo studente 100 e non sostenuti dallo studente x è vuoto



ESEMPIO

```
SELECT *  
FROM Studenti AS S  
WHERE NOT EXISTS
```

```
(SELECT E.CORSO  
FROM Esami AS E
```

```
WHERE E.Matricola = '100' AND E.CORSO NOT IN
```

```
(SELECT E2.CORSO  
FROM Esami AS E2  
WHERE E2.Matricola = S.Matricola))
```



ESEMPIO

```
SELECT *
FROM Studenti AS S
WHERE NOT EXISTS
    (SELECT E.CORSO
     FROM Esami AS E
     WHERE E.Matricola = '100' AND E.CORSO NOT IN
          (SELECT E2.CORSO
           FROM Esami AS E2
           WHERE E2.Matricola = S.Matricola))
```

Insieme dei corsi sostenuti
dallo studente S.Matricola



ESEMPIO

Insieme dei corsi sostenuti da 100, ma non sostenuti dallo studente S.Matricola

```
SELECT *  
FROM Studenti AS S  
WHERE NOT EXISTS
```

```
(SELECT E.CORSO  
FROM Esami AS E  
WHERE E.Matricola = '100' AND E.CORSO NOT IN
```

```
(SELECT E2.CORSO  
FROM Esami AS E2  
WHERE E2.Matricola = S.Matricola))
```



ESEMPIO

```
SELECT *
FROM Studenti AS S
WHERE NOT EXISTS
    (SELECT E.CORSO
     FROM Esami AS E
     WHERE E.Matricola = '100' AND E.CORSO NOT IN
```

Insieme degli studenti S per cui l'insieme degli esami sostenuti da 100, ma non sostenuti da S, è vuoto.

```
(SELECT E2.CORSO
     FROM Esami AS E2
     WHERE E2.Matricola = S.Matricola))
```



ATTENZIONE ALLA VISIBILITÀ DEI NOMI

```
SELECT *  
FROM Studenti AS S  
WHERE NOT EXISTS
```

```
(SELECT E.CORSO  
FROM Esami AS E
```

```
WHERE E.Matricola = '100' AND E.CORSO NOT IN
```

```
(SELECT E2.CORSO  
FROM Esami AS E2  
WHERE E2.Matricola = S.Matricola))
```



ASSERTION

- Vincoli che conosciamo
 - Vincoli di dominio: i valori dell'attributo devono appartenere al relativo dominio
 - Vincoli di integrità referenziale: FOREIGN KEY
 - Vincoli intrarelazionale: NOT NULL, UNIQUE, PRIMARY KEY
- Altri vincoli
 - Vincoli interrelazionali: vincoli che coinvolgono attributi di più tabelle
 - Nello standard di SQL viene messo a disposizione un costrutto che si chiama ASSERTION
 - La Assertion è un vincolo che la base di dati deve rispettare

CREATE ASSERTION <nome vincolo> espressione booleana



ASSERTION

CREATE ASSERTION <nome_vincolo> espressione booleana

[NOT] EXIST (SELECT

<Cerco la violazione del vincolo>

)



ESEMPIO

ALBERO(CodAlbero, CodRadice)

NODE(CodNodo, Label, CodAlbero)

ARCO(CodArco, NodoSorgente, NodoTarget, Label, CodAlbero)

Vincoli

- *Sorgente e target di un albero sono diversi*



ESEMPIO

ALBERO(CodAlbero, CodRadice)

NODE(CodNodo, Label, CodAlbero)

ARCO(CodArco, NodoSorgente, NodoTarget, Label, CodAlbero)

Vincoli

- *Sorgente e target di un albero sono diversi (vincolo intrarelazionale, in particolare è un vincolo di ennupla) → non ho bisogno di un vincolo di asserzione, ma posso inserire un vincolo nella tabella.*



ESEMPIO

ALBERO(CodAlbero, CodRadice)

NODE(CodNodo, Label, CodAlbero)

ARCO(CodArco, NodoSorgente, NodoTarget, Label, CodAlbero)

Vincoli

1) *Sorgente e target di un albero sono diversi (vincolo intrarelazionale, in particolare è un vincolo di ennupla) → non ho bisogno di un vincolo di asserzione, ma posso inserire un vincolo nella tabella.*

ALTER TABLE ARCO

ADD CONSTRAINT C1

CHECK NodoSorgente <> Nodo Target



ESEMPIO

ALBERO(CodAlbero, CodRadice)

NODE(CodNodo, Label, CodAlbero)

ARCO(CodArco, NodoSorgente, NodoTarget, Label, CodAlbero)

Vincoli

2) *L'identificativo della radice è un nodo dell'albero*



ESEMPIO

ALBERO(CodAlbero, CodRadice)

NODE(CodNodo, Label, CodAlbero)

ARCO(CodArco, NodoSorgente, NodoTarget, Label, CodAlbero)

Vincoli

2) L'identificativo della radice è un nodo dell'albero (interrelazionale)

CREATE ASSERTION A1

*CHECK NOT EXIST (SELECT * FROM ALBERO AS T*

WHERE T.CodRadice NOT IN

(SELECT N.CodNodo FROM NODO N

WHERE N.CodNodo= T.CodRadice)



ALTRI VINCOLI INTERRELAZIONALI

- Un nodo ha un solo padre
- Una radice non ha padri
- Tutti i nodi sono connessi





FINE

Per eventuali domande: (in ordine di preferenza personale)

- Ora.
- Chat di Teams
- Mail: silvio.barra@unina.it

