

Table expressions

- Tra le caratteristiche più rilevanti di SQL vi è quella di poter usare il risultato di una interrogazione come argomento della clausola from.
- In questo caso la clausola from è rappresentata da una tabella derivata dinamicamente che prende il nome di table expression (o derived table, supportate da MySQL 5.0).

257

Table expressions

Esempio: valutare per ogni sede quale è lo stipendio massimo e quanti impiegati lo percepiscono:

```
SELECT SM.Sede,SM.MaxStip,COUNT(*) AS
      NumImpWMaxStip
FROM Imp I,(
      SELECT Sede, MAX(Stipendio)
      FROM Imp
      GROUP BY Sede) AS SM(Sede,MaxStip)
WHERE I.Sede = SM.Sede AND I.Stipendio =
      SM.MaxStip
GROUP BY SM.Sede,SM.MaxStip
```

258

Table expressions

- L'interrogazione prevede la definizione della tabella SM(Sede, Stipendio) derivata dinamicamente come risultato di una SelectSQL.
- All'interno della table expression possono anche comparire variabili definite esternamente ad essa.

259

Table expressions

- Esempio: calcolare, per ogni sede, la somma degli stipendi percepiti dagli analisti.

```
select S.Sede, Stip.TotStip
from Sedi S, (select sum(Stipendio)
              from Imp I
              where I.Sede=S.Sede and
                    I.Ruolo='Analista') as
              Stip(TotStip)
```

260

Table expressions

- Lo stesso risultato si poteva ottenere anche senza l'uso delle table expression:

```
select Sede, sum(Stipendio) as  
StipendiAnalisti  
from Imp  
where Ruolo='Analista'  
group by Sede
```

261

Common table expressions ed interrogazioni ricorsive (SQL-99)

- Le table expressions possono anche essere usate per definire delle viste temporanee da essere usate all'interno di interrogazioni come fossero a tutti gli effetti delle viste.
- Si parla in questo caso di common table expressions.

262

Common table expressions ed interrogazioni ricorsive (SQL-99)

```
with SediStip(Sede, TotStip)
as (select Sede, sum(Stipendio)
     from Imp
     group by Sede)
select Sede
from SediStip
where TotStip =
     select max(TotStip)
     from SediStip
```

263

Common table expressions ed interrogazioni ricorsive (SQL-99)

- Attraverso il costrutto `with`, le common table expressions possono essere impiegate per formulare interrogazioni ricorsive (quindi interrogazioni non esprimibili in algebra relazionale in quanto aventi bisogno di un numero di self-join non noto a priori).

264

Common table expressions ed interrogazioni ricorsive (SQL-99)

- Esempio: data la relazione Genitori(Figlio, Genitore) sotto mostrata, trovare tutti antenati (genitori, nonni, bisnonne, ecc.) di Anna.

Genitori

Figlio	Genitore
Anna	Luca
Luca	Carla
Luca	Giovanni
Carla	Luisa
Giovanni	Augusto
Giorgio	Susanna

265

Common table expressions ed interrogazioni ricorsive (SQL-99)

- La soluzione prevede la definizione di una vista temporanea ricorsiva Antenati(Persona, Avo) che risulta dall'unione di:
 - una sub-query di base non ricorsiva impiegata per inizializzare Antenati(Persona, Avo)
 - una sub-query ricorsiva che ad ogni iterazione aggiorna Antenati(Persona, Avo)

266

Common table expressions ed interrogazioni ricorsive (SQL-99)

with Antenati(Persona, Avo) as(

```
select Figlio, Genitore  
from Genitori
```

Inizializzazione:
subquery di base

```
union all
```

```
select G.Figlio, A.Avo  
from Genitori G, Antenati A  
where G.Genitore=A.Persona)
```

Ricorsione:
subquery ricorsiva

```
select Avo  
from Antenati  
where Persona='Anna'
```

267

Common table expressions ed interrogazioni ricorsive (SQL-99)

- Apparentemente la sintassi sembra suggerire che ad ogni iterazione venga fatto il join (implicito) tra Genitori ed Antenati con Genitore=Persona.
- Tuttavia questo non è vero: infatti, il join in questione non coinvolge l'intera tabella Antenati, ma solo quella parte di tale tabella che è stata aggiunta nell'ultima iterazione.

```
with Antenati(Persona, Avo) as(  
select Figlio, Genitore  
from Genitori  
union all  
select G.Figlio, A.Avo  
from Genitori G, Antenati A  
where G.Genitore=A.Persona)  
select Avo  
from Antenati  
where Persona='Anna'
```

268

Common table expressions ed interrogazioni ricorsive (SQL-99)

- In questo modo, il contenuto della tabella Antenati viene così costruito:

Genitori

Figlio	Genitore
Anna	Luca
Luca	Carla
Luca	Giovanni
Carla	Luisa
Giovanni	Augusto
Giorgio	Susanna

Antenati

Persona	Avo
Anna	Luca
Luca	Carla
Luca	Giovanni
Carla	Luisa
Giovanni	Augusto
Giorgio	Susanna
Anna	Carla
Anna	Giovanni
Luca	Luisa
Luca	Augusto
Anna	Luisa
Anna	Augusto

Inizializzazione

1^a Iterazione

2^a Iterazione

```
with Antenati(Persona, Avo) as(
select Figlio, Genitore
from Genitori
union all
select G.Figlio, A.Avo
from Genitori G, Antenati A
where G.Genitore=A.Persona)
```

Common table expressions ed interrogazioni ricorsive (SQL-99)

- Per le common table expressions valgono le seguenti restrizioni:
 - possono essere usate solo nel blocco più esterno di una query
 - nella subquery ricorsiva non si possono usare funzioni aggregate: group by, select distinct