



Politecnico  
di Torino



# Interrogazioni avanzate

---

Linguaggio SQL

# Linguaggio SQL: interrogazioni avanzate

---

- Tabelle derivate
- CTE

# Tabelle derivate

- Definisce una tabella temporanea che può essere utilizzata per ulteriori operazioni di calcolo
- La tabella derivata
  - ha la struttura di una **SELECT**
  - è definita all'interno di una clausola **FROM**
  - può essere referenziata come una normale tabella
- La tabella derivata permette di
  - calcolare più livelli di aggregazione
  - formulare in modo equivalente le interrogazioni che richiedono la correlazione

# Calcolo di aggregati a due livelli (n.1)

---

- Trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Risoluzione in 2 passi
  - trovare la media per ogni studente
  - trovare il valore massimo della media

# Calcolo di aggregati a due livelli (n.1)

---

- Trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

Passo 1: trovare la media per ogni studente

```
SELECT Matricola, AVG(Voto) AS MediaStudente  
FROM ESAME-SUPERATO  
GROUP BY Matricola
```

# Calcolo di aggregati a due livelli (n.1)

- Trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

Passo 2: trovare il valore massimo della media

```
SELECT MAX(MediaStudente)
```

```
FROM (SELECT Matricola, AVG(Voto) AS MediaStudente  
      FROM ESAME-SUPERATO  
      GROUP BY Matricola) AS MEDIE;
```

*Tabella derivata*

# Calcolo di aggregati a due livelli (n.2)

---

- Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Risoluzione in 2 passi
  - trovare la media per ogni studente
  - raggruppare gli studenti per anno di iscrizione e calcolare la media massima

# Calcolo di aggregati a due livelli (n.2)

- Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Passo 1: trovare la media per ogni studente

```
(SELECT Matricola, AVG(Voto) AS MediaStudente  
FROM ESAME-SUPERATO  
GROUP BY Matricola) AS MEDIE
```



# Calcolo di aggregati a due livelli (n.2)

- Per ogni anno di iscrizione, trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Passo 2: raggruppare gli studenti per anno di iscrizione e calcolare la media massima

```
SELECT Anno Iscrizione, MAX(MediaStudente)
FROM STUDENTE,
```

```
(SELECT Matricola, AVG(Voto) AS MediaStudente
FROM ESAME-SUPERATO
GROUP BY Matricola) AS MEDIE
```

← *Tabella derivata*

```
WHERE STUDENTE.Matricola=MEDIE.Matricola
```

← *Condizione  
di join*

```
GROUP BY AnnoIscrizione
```

# Correlazione con tabella derivata

- Per ogni prodotto, trovare il codice del fornitore che ne fornisce la quantità massima

F (CodF, NomeF, NSoci, Sede)

P (CodP, NomeP, Colore, Taglia, Magazzino)

FP (CodP, CodF, Qta)

- Risoluzione in 2 passi
  - Calcolare la Qta massima fornita per ogni prodotto
  - Selezionare i fornitori che forniscono la Qta massima, prodotto per prodotto

# Correlazione con tabella derivata

- Per ogni prodotto, trovare il codice del fornitore che ne fornisce la quantità massima

F (CodF, NomeF, NSoci, Sede)

P (CodP, NomeP, Colore, Taglia, Magazzino)

FP (CodP, CodF, Qta)

```
SELECT CodP, CodF
FROM FP AS FPX
WHERE Qta = (SELECT MAX(Qta)
             FROM FP AS FPY
             WHERE FPY.CodP=FPX.CodP);
```

*Condizione di correlazione*

*Quantità  
massima  
per il  
prodotto  
corrente*

# Correlazione con tabella derivata

- Per ogni prodotto, trovare il codice del fornitore che ne fornisce la quantità massima

F (CodF, NomeF, NSoci, Sede)

P (CodP, NomeP, Colore, Taglia, Magazzino)

FP (CodP, CodF, Qta)

- Passo 1: calcolare la Qta massima fornita per ogni prodotto

```
SELECT CodP, MAX(Qta) AS MQta  
FROM FP  
GROUP BY CodP
```

# Correlazione con tabella derivata

- Per ogni prodotto, trovare il codice del fornitore che ne fornisce la quantità massima

F (CodF, NomeF, NSoci, Sede)

P (CodP, NomeP, Colore, Taglia, Magazzino)

FP (CodP, CodF, Qta)

- Passo 2: selezionare i fornitori che forniscono la Qta massima, prodotto per prodotto

```
SELECT CodP, CodF
```

```
FROM FP,
```

```
(SELECT CodP, MAX(Qta) AS MQta  
FROM FP  
GROUP BY CodP) AS TMax
```

← *Tabella derivata*

```
WHERE FP.CodP = TMax.CodP
```

← *Condizione di join*

```
AND FP.Qta = TMax.MQta;
```

← *Correlazione*

# Correlazione tra interrogazioni (n.2)

- Trovare il codice dei viaggi che hanno una durata inferiore alla durata media dei viaggi sullo stesso percorso (caratterizzato dallo stesso luogo di partenza e di arrivo)

VIAGGIO (CodV, LuogoPartenza, LuogoArrivo,  
OraPartenza, OraArrivo)

```
SELECT CodV
FROM VIAGGIO AS VA
WHERE OraArrivo-OraPartenza <
      (SELECT AVG(OraArrivo-OraPartenza)
       FROM VIAGGIO AS VB
        WHERE VB.LuogoPartenza=VA.LuogoPartenza
              AND VB.LuogoArrivo=VA.LuogoArrivo);
```

*Condizioni di correlazione*

*Durata  
media  
dei viaggi  
sul percorso  
corrente*

# Common Table Expression

- Definisce una tabella temporanea che può essere utilizzata per ulteriori operazioni di calcolo
- La CTE
  - ha la struttura di una **SELECT**
  - è definita mediante la clausola **WITH**
  - può essere referenziata come una normale tabella
- La CTE è usata per
  - calcolare più livelli di aggregazione
  - formulare in modo equivalente le interrogazioni che richiedono la correlazione
- Riferimenti
  - a CTE *precedentemente* definite nella stessa clausola **WITH**
  - ricorsivo

# CTE vs Tabelle derivate

---

- La CTE è preferita quando
  - è necessario fare riferimento a una tabella derivata più volte in una singola query
  - è necessario eseguire lo stesso calcolo più volte in più parti della query
  - si vuole aumentare la leggibilità di query complesse



# Sintassi per definizione di CTE

WITH

cte\_1 [(campo\_A, ...)] AS

(CTE query 1)

{, cte\_X AS (CTE query X) }

SELECT campo\_A, campo\_B, ...  
FROM cte\_1

*Nome della CTE*

*Query associata alla CTE*

*Query*

# Calcolo di aggregati a due livelli (n.1)

---

- Trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

- Risoluzione in 2 passi
  - trovare la media per ogni studente
  - trovare il valore massimo della media

# Calcolo di aggregati a due livelli

---

- Trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

Passo 1: trovare la media per ogni studente

WITH MEDIE AS

```
(SELECT Matricola, AVG(Voto) AS MediaStudenti  
FROM ESAME-SUPERATO  
GROUP BY Matricola)
```

# Calcolo di aggregati a due livelli

- Trovare la media massima (conseguita da uno studente)

STUDENTE (Matricola, AnnoIscrizione)

ESAME-SUPERATO (Matricola, CodC, Data, Voto)

Passo 2: trovare il valore massimo della media

WITH MEDIE AS

(SELECT Matricola, AVG(Voto) AS MediaStudenti

FROM ESAME-SUPERATO

GROUP BY Matricola)

SELECT MAX(MediaStudenti)

FROM MEDIE;

# Calcolo di aggregati con granularità diversa

- Trovare tutte le compagnie aeree in cui il salario medio di tutti i piloti di quella compagnia è superiore alla media dei salari totali di tutti i piloti del database

PILOTI (CodP, Nome, Cognome, Compagnia, Salario)

- Risoluzione in 3 passi
  - trovare il salario medio per ogni compagnia
  - trovare il salario medio considerando tutti i piloti
  - trovare le compagnie con salario medio maggiore del salario medio globale

# Calcolo di aggregati con granularità diversa

---

- Passo 1: trovare il salario medio per ogni compagnia

```
WITH salarioMedioCompagnia AS  
  (SELECT Compagnia, AVG(Salario) AS AvgSalComp  
   FROM PILOTI  
   GROUP BY Compagnia)
```

# Calcolo di aggregati con granularità diversa

- Passo 2: trovare il salario medio del database

```
WITH salarioMedioCompagnia AS
    (SELECT Compagnia, AVG(Salario) AS AvgSalComp
     FROM PILOTI
     GROUP BY Compagnia),
mediaSalario AS
    (SELECT AVG(Salario) AS MediaSal
     FROM PILOTI )
```

# Calcolo di aggregati con granularità diversa

- Passo 3: trovare le compagnie con salario medio maggiore del salario medio globale

```
WITH salarioMedioCompagnia AS
    (SELECT Compagnia, AVG(Salario) AS AvgSalComp
     FROM PILOTI
     GROUP BY Compagnia),
mediaSalario AS
    (SELECT AVG(Salario) AS MediaSal
     FROM PILOTI )
SELECT Compagnia
FROM salarioMedioCompagnia, mediaSalario
WHERE salarioMedioCompagnia. AvgSalComp >
mediaSalario.MediaSal;
```



# CTE referenziate

- Considerando le distanze medie percorse per ciascuna città, calcolare la distanza massima percorsa per ciascuna provincia

CITTA (CodC, NomeC, Provincia)

AUTISTA (CodA, NomeA, Cognome, CodC)

CORSA\_GIORNALIERA (Data, CodA, Importo, Distanza)

- Risoluzione in 3 passi
  - calcolare la *distanza totale* percorsa per ogni città da ogni autista
  - calcolare la *distanza media* per ogni città
  - calcolare la *distanza media massima* per provincia

# CTE referenziate

---

- Passo 1: calcolare la distanza percorsa per ogni città da ogni autista

WITH totDistanzaAutista AS

```
( SELECT SUM(Distanza) AS distanzaTot, CG.CodA, CG.CodC, NomeC, Provincia  
  FROM CORSA_GIORNALIERA CG, CITTA C, AUTISTA A  
 WHERE CG.CodA=A.CodA AND A.CodC=C.CodC  
 GROUP BY CG.CodA, CG.CodC, NomeC, Provincia )
```

# CTE referenziate

- Passo 2: calcolare la distanza media per ogni città

WITH totDistanzaAutista AS

```
(SELECT SUM(Distanza) AS distanzaTot, CG.CodA, CG.CodC, NomeC, Provincia
FROM CORSA_GIORNALIERA CG, CITTA C, AUTISTA A
WHERE CG.CodA=A.CodA AND A.CodC=C.CodC
GROUP BY CG.CodA, CG.CodC, NomeC, Provincia),
```

distanzaMedia AS

```
( SELECT AVG(distanzaTot) AS avgDist, CodC, NomeC, Provincia
FROM totDistanzaAutista
GROUP BY CodC, NomeC, Provincia )
```

# CTE referenziate

- Passo 3: calcolare la distanza massima per provincia

WITH totDistanzaAutista AS

```
(SELECT SUM(Distanza) AS distanzaTot, CG.CodA, CG.CodC, NomeC, Provincia
FROM CORSA_GIORNALIERA CG, CITTA C, AUTISTA A
WHERE CG.CodA=A.CodA AND A.CodC=C.CodC
GROUP BY CG.CodA, CG.CodC, NomeC, Provincia),
```

distanzaMedia AS

```
( SELECT AVG(distanzaTot) AS avgDist, CodC, NomeC, Provincia
FROM totDistanzaAutista
GROUP BY CodC, NomeC, Provincia )
```

```
SELECT MAX(avgDist), Provincia
FROM distanzaMedia
GROUP BY Provincia
```