

Laboratorio di Basi di dati 1

Soluzioni della IV esercitazione - 1 aprile 2002

La sintassi dei comandi T-SQL non è rigorosa, poiché permette di ottenere lo stesso risultato con diverse combinazioni di argomenti. Di seguito saranno indicate solo le possibilità principali che permettono di rispondere a quanto richiesto nell'esercitazione.

Uso di colonne virtuali e controlli sui valori nulli

1.

```
select denominazione, count(*) as numeroIscritti
from studenti join corsiDiLaurea on corsodilaurea=id
group by facolta, denominazione
```
2.

```
select matricola
from studenti
where relatore is null and iscrizione ='2000/2001'
```
3.

```
select cognome, nome, year(getdate()) - year(datanascita) as eta
from studenti
```
4.

```
select 'Dott. ' + Nome + ' ' + Cognome
from studenti
where laurea is not null
order by cognome, nome
```
5.

```
select studente, avg(voto) as mediaMatematica
from esami
where voto > 17
group by studente
```
6.

```
select studente, (sum(voto)-max(voto)-min(voto))/(count(*)-2) as mediaPesata
from esami
where voto > 17
group by studente
having count(*)>2
```
7.

```
select CORSO, AVG(VOTO)
from esami join corsi on corso=id
where professore is NOT null AND ATTIVATO = 1 and voto > 17
GROUP BY CORSO
```
8.

```
SELECT STUDENTE, AVG(VOTO) AS MEDIA
FROM ESAMI, STUDENTI, CORSIDILAUREA
WHERE ESAMI.STUDENTE = STUDENTI.MATRICOLA
AND STUDENTI.CORSODILAUREA = CORSIDILAUREA.ID
AND DENOMINAZIONE = 'INFORMATICA'
AND DATA BETWEEN '06/01/2000' AND '06/30/2000'
GROUP BY STUDENTE
HAVING COUNT(*)>1
```

oppure

```
SELECT STUDENTE, AVG(VOTO) AS MEDIA
FROM ESAMI JOIN STUDENTI ON ESAMI.STUDENTE = STUDENTI.MATRICOLA JOIN CORSIDILAUREA ON
STUDENTI.CORSODILAUREA = CORSIDILAUREA.ID
WHERE DENOMINAZIONE = 'INFORMATICA'
AND DATA BETWEEN '06/01/2000' AND '06/30/2000'
GROUP BY STUDENTE
HAVING COUNT(*)>1
```

Sottointerrogazioni

1.

```
select professori.id from professori
where professori.id not in (select professori.id
from corsi join professori on professore=professori.id
where attivato = 1)
```
2.

```
select cognome, nome
from professori
where (cognome,nome) IN (select cognome, nome from studenti)
```

3.

```
select corsidilaurea.id
from corsi join professori on professore=professori.id join corsidilaurea on corsodilaurea=corsidilaurea.id
where attivato =1
group by corsidilaurea.id
having count(*) > (select count(*) as profeChimica
from corsi join professori on professore=professori.id
join corsidilaurea on corsodilaurea=corsidilaurea.id
where corsidilaurea.denominazione like 'chimica%')
```
4.

```
select id, count(*) as niscr
from studenti join corsidilaurea on corsodilaurea=id
where iscrizione='2000/2001'
group by id, matricola
having count(*) < (select count(*) as iscrInfo
from studenti join corsidilaurea on corsodilaurea=id
where iscrizione='2000/2001' and denominazione='informatica')
```
5.

```
select distinct Studenti.cognome, studenti.nome
from studenti, pianidistudio, corsi, corsidilaurea, professori
where studenti.CorsoDiLaurea=corsidilaurea.id
and matricola=pianidistudio.studente
and pianidistudio.anno=5
and corsidilaurea.denominazione='informatica'
and annoaccademico='2001/2002'
and professori.id=studenti.relatore
and relatore in (select professori.id from professori, corsi, corsidilaurea
where professori.id=corsi.professore
and corsi.CorsoDiLaurea=corsidilaurea.id
and corsi.attivato = 1
and CorsiDiLaurea.denominazione='matematica')
```
6.

```
SELECT STUDENTE
FROM ESAMI, STUDENTI, CORSIDILAUREA, CORSI
WHERE ESAMI.STUDENTE = STUDENTI.MATRICOLA
AND STUDENTI.CORSODILAUREA = CORSIDILAUREA.ID
AND ESAMI.CORSO = CORSI.ID
AND CORSIDILAUREA.DENOMINAZIONE = 'INFORMATICA'
AND DATA BETWEEN '06/01/2000' AND '06/30/2000'
AND CORSI.DENOMINAZIONE = 'BASI DI DATI 1'
AND STUDENTE IN (SELECT MATRICOLA
FROM ESAMI, STUDENTI, CORSIDILAUREA, CORSI
WHERE ESAMI.STUDENTE = STUDENTI.MATRICOLA
AND STUDENTI.CORSODILAUREA = CORSIDILAUREA.ID
AND ESAMI.CORSO = CORSI.ID
AND CORSIDILAUREA.DENOMINAZIONE = 'INFORMATICA'
AND DATA BETWEEN '06/01/2000' AND '06/30/2000'
AND CORSI.DENOMINAZIONE = 'INTERFACCE GRAFICHE')
```

Sottointerrogazioni correlate

1.

```
SELECT CORSO, STUDENTE, VOTO
FROM ESAMI X
WHERE VOTO < (SELECT AVG(VOTO)
FROM ESAMI WHERE X.CORSO = CORSO
GROUP BY CORSO)
ORDER BY CORSO, STUDENTE, VOTO
```
2.

```
SELECT DENOMINAZIONE, COGNOME, NOME, VOTO
FROM CORSI JOIN ESAMI X ON CORSO=ID JOIN STUDENTI ON MATRICOLA=STUDENTE
WHERE VOTO < (SELECT AVG(VOTO) AS VOTOMEDIO
FROM ESAMI WHERE
X.CORSO = CORSO
GROUP BY CORSO)
order by denominazione
```
3.

```
SELECT MATRICOLA, AVG(VOTO) AS MEDIA
FROM ESAMI, STUDENTI, CORSIDILAUREA
WHERE ESAMI.STUDENTE = STUDENTI.MATRICOLA
AND STUDENTI.CORSODILAUREA = CORSIDILAUREA.ID
AND CORSIDILAUREA.DENOMINAZIONE = 'INFORMATICA'
AND STUDENTE IN (SELECT MATRICOLA
FROM ESAMI X, STUDENTI, CORSIDILAUREA
WHERE ESAMI.STUDENTE = STUDENTI.MATRICOLA
```

```

        AND STUDENTI.CORSODILAUREA = CORSIDILAUREA.ID
        AND CORSIDILAUREA.DENOMINAZIONE = 'INFORMATICA'
        AND MONTH(DATA)=MONTH(X.DATA) AND YEAR(DATA)=YEAR(X.DATA))
GROUP BY MATRICOLA
HAVING COUNT(DISTINCT CORSO) > 1
4. SELECT *
FROM STUDENTI X
WHERE RELATORE IS NULL
AND NOT EXISTS ( SELECT *
                 FROM CORSI
                 WHERE CORSODILAUREA=X.CORSODILAUREA
                 AND ID NOT IN (SELECT CORSO
                                FROM ESAMI
                                WHERE STUDENTE=X.MATRICOLA))
5. SELECT CORSO, YEAR(DATA), MONTH(DATA), STUDENTE, VOTO
FROM ESAMI X
WHERE VOTO = (SELECT MAX(VOTO)
              FROM ESAMI
              WHERE X.CORSO=CORSO AND YEAR(X.DATA)=YEAR(DATA)
              AND MONTH(X.DATA)=MONTH(DATA)
              GROUP BY CORSO, YEAR(DATA), MONTH(DATA))
ORDER BY CORSO, DATA
oppure utilizzando il casting:
SELECT CORSO, cast(YEAR(DATA) as char)+ cast (MONTH(DATA) as char) as sessione, STUDENTE, VOTO
FROM ESAMI X
WHERE VOTO = (SELECT MAX(VOTO)
              FROM ESAMI
              WHERE X.CORSO=CORSO AND YEAR(X.DATA)=YEAR(DATA)
              AND MONTH(X.DATA)=MONTH(DATA)
              GROUP BY CORSO, YEAR(DATA), MONTH(DATA))
ORDER BY CORSO, DATA
6. - MEDIA DEI VOTI DI OGNI CORSO
SELECT CORSO, AVG(VOTO) AS MEDIA
FROM ESAMI
GROUP BY CORSO -

SELECT COGNOME, NOME, PROFESSORI.ID
FROM PROFESSORI JOIN CORSI ON PROFESSORE=PROFESSORI.ID
JOIN ( SELECT CORSO, AVG(VOTO) AS MEDIA
       FROM ESAMI
       GROUP BY CORSO) AS R1 ON R1.CORSO=CORSI.ID
WHERE NOT EXISTS (SELECT CORSO
                  FROM ESAMI
                  GROUP BY CORSO
                  HAVING AVG(VOTO)> R1.MEDIA)
7. Soluzione utilizzando la sotto query correlata (si noti l'utilizzo di una select all'interno della clausola FROM)
----- NUMERO DI BOCCIATI PER OGNI CORSO
SELECT CORSO, COUNT(*) AS NBOCCIATI
FROM ESAMI
WHERE VOTO < 18
GROUP BY CORSO -----

SELECT DENOMINAZIONE, R1.NBOCCIATI
FROM CORSI, (SELECT CORSO, COUNT(*) AS NBOCCIATI
            FROM ESAMI
            WHERE VOTO < 18
            GROUP BY CORSO) R1
WHERE CORSI.ID=R1.CORSO
AND NOT EXISTS (SELECT CORSO
                FROM ESAMI
                WHERE VOTO < 18
                GROUP BY CORSO
                HAVING COUNT(*)>R1.NBOCCIATI)
Altra soluzione che fa uso di una semplice sotto query
select denominazione, count(*) As NBOCCIATI
from esami join corsi on esami.corso = corsi.id
where voto < 18
group by corsoDiLaurea, denominazione

```

```
having count(*) >= ALL (select count(*)
by corso)
from esami
where voto < 18
group
```

8. Soluzione utilizzando la sotto query correlata

```
SELECT COGNOME, NOME, R1.NCORSI
FROM PROFESSORI, (SELECT PROFESSORE, COUNT(*) AS NCORSI
FROM CORSI
WHERE PROFESSORE IS NOT NULL
GROUP BY PROFESSORE) R1
WHERE PROFESSORI.ID=R1.PROFESSORE
AND NOT EXISTS (SELECT PROFESSORE, COUNT(*) AS NCORSI
FROM CORSI
WHERE PROFESSORE IS NOT NULL
GROUP BY PROFESSORE
HAVING COUNT(*)>NCORSI)
```

Altra soluzione che fa uso di una semplice sotto query

```
select cognome,nome, count(*) as numeroCorsi
from corsi join professori on professori.id=professore
where professore is not null
group by professore, cognome, nome
having count(*) >= ALL (select count(*)
from corsi
where professore is not null
group by professore)
```

9. – NUMERO DI ISCRITTI PER SESSIONE

```
SELECT MONTH(DATA), COUNT(*) AS NISCRITTI,
FROM ESAMI
GROUP BY MONTH(DATA)
```

– NUMERO DI BOCCIATI

```
SELECT MONTH(DATA), COUNT(*) AS NBOCCIATI,
FROM ESAMI
WHERE VOTO<18
GROUP BY MONTH(DATA)
```

– PERCENTUALE DI BOCCIATI, PER SESSIONE

```
SELECT R1.SESSIONE, R2.NBOCCIATI/R1.NISCRITTI*100 AS PERC
FROM (SELECT MONTH(DATA) AS SESSIONE, COUNT(*) AS NISCRITTI
FROM ESAMI
GROUP BY MONTH(DATA)) R1
JOIN (SELECT MONTH(DATA) AS SESSIONE, COUNT(*) AS NBOCCIATI
FROM ESAMI
WHERE VOTO<18
GROUP BY MONTH(DATA)) R2 ON R1.SESSIONE=R2.SESSIONE
```

oppure utilizzando il casting:

– NUMERO DI ISCRITTI PER SESSIONE

```
SELECT MONTH(DATA), COUNT(*) AS NISCRITTI,
FROM ESAMI
GROUP BY MONTH(DATA)
```

– NUMERO DI BOCCIATI

```
SELECT MONTH(DATA), COUNT(*) AS NBOCCIATI,
FROM ESAMI
WHERE VOTO<18
GROUP BY MONTH(DATA)
```

– PERCENTUALE DI BOCCIATI, PER SESSIONE

```
SELECT R1.SESSIONE, (CAST(R2.NBOCCIATI AS FLOAT)/CAST (R1.NISCRITTI AS FLOAT))*100 AS PERC
FROM (SELECT MONTH(DATA) AS SESSIONE, COUNT(*) AS NISCRITTI
FROM ESAMI
GROUP BY MONTH(DATA)) R1
JOIN (SELECT MONTH(DATA) AS SESSIONE, COUNT(*) AS NBOCCIATI
FROM ESAMI
WHERE VOTO<18
GROUP BY MONTH(DATA)) R2 ON R1.SESSIONE=R2.SESSIONE
```

10. — come prima