

SQL – Esercizi DML – Blocco 1

Si consideri il seguente schema di base di dati che vuole tenere traccia di alcune informazioni riguardanti gli studenti di un'università.

Studenti(Matricola, NomeS, CognomeS, CittàRes, Sesso, NumTelefono)

Corsi(CodCorso, NomeC, NumCrediti, MatricolaDocente)

Esami(Matricola, CodCorso, Voto)

Docenti(MatricolaD, NomeD, CognomeD, DipartimentoDiAfferenza, Stipendio)

La tabella Esami contiene l'elenco di esami superati dagli studenti.

Esercizi

- 1) Trovare le informazioni su tutti gli studenti presenti nella tabella Studenti
 - a.

```
select Matricola, NomeS, CognomeS, CittàRes, Sesso, NumTelefono
from studenti;
```

Alternativa
 - b.

```
select *
from studenti;
```
- 2) Trovare la matricola e la città di residenza di tutti gli studenti presenti nella base di dati
 - a.

```
select Matricola, CittàRes
from studenti;
```
- 3) Trovare le città in cui risiede almeno uno studente
 - a.

```
/* Con questa prima soluzione non vengono eliminate le città duplicate/ripetute */
select CittàRes
from studenti;
```

Alternativa
 - b.

```
/* Con questa seconda soluzione vengono eliminate le città duplicate/ripetute grazie alla
presenza della parola chiave DISTINCT */
select DISTINCT CittàRes
from studenti;
```
- 4) Trovare matricola e stipendio mensile per ogni docente (si supponga che l'attributo Stipendio della tabella Docenti contenga lo stipendio annuo e che lo stipendio mensile sia calcolabile come quello annuo diviso 12)
 - a.

```
/* Il valore di Stipendio/12 viene rinominato StipendioMensile */
select MatricolaD, Stipendio/12 as StipendioMensile
from docenti;
```
- 5) Trovare gli studenti di Verona (tutte le informazioni disponibili su tali studenti)
 - a.

```
select *
from studenti
where CittàRes='Verona';
```

6) Trovare gli studenti che non risiedono a Verona

a.

```
select *
from studenti
where CittàRes<>'Verona'
```

7) Trovare gli studenti di Verona di sesso femminile

a.

```
select *
from studenti
where CittàRes='Verona' and Sesso='F';
```

8) Trovare gli studenti di sesso femminile che risiedono a Verona o Parma.

a.

```
select *
from studenti
where Sesso='F' and (CittàRes='Verona' or CittàRes='Parma');
```

9) Trovare le informazioni sui corsi da almeno 5 crediti

a.

```
select *
from corsi
where numCrediti>=5;
```

Esempi d'uso della parola chiave LIKE per il confronto tra stringhe. Si ricorda che il comando **attributo LIKE 'stringa'** permette di confrontare il valore dell'attributo con il contenuto della stringa di caratteri presente dopo la parola chiave LIKE. La stringa di confronto può contenere i caratteri speciali % e _

% = stringa di lunghezza arbitraria (eventualmente zero) con contenuto arbitrario

_ = un singolo carattere arbitrario (uno ed esattamente uno)

10) Trovare gli studenti il cui cognome comincia con la lettera R

a.

```
select *
from studenti
where CognomeS LIKE 'R%';
```

11) Trovare gli studenti il cui cognome comincia con la lettera R e finisce con la lettera i

a.

```
select *
from studenti
where CognomeS LIKE 'R%i';
```

12) Trovare gli studenti il cui cognome ha la lettera a in seconda posizione

a.

```
select *
from studenti
where CognomeS LIKE '_a%';
```

Valore NULL. Si ricorda che per verificare se un attributo di una tupla contiene il valore nullo oppure no è necessario usare i comandi *attributo IS NULL* e *attributo IS NOT NULL*

13) Trovare gli studenti per cui è noto il numero di telefono

a.

```
select *
from studenti
where NumTelefono IS NOT NULL;
```

14) Trovare la matricola degli studenti per cui non è noto il numero di telefono

a.

```
select matricola
from studenti
where NumTelefono IS NULL;
```

Ordinamento dei risultati.

15) Trovare i corsi, restituendoli ordinati in numero di crediti decrescente.

a.

```
select *
from corsi
order by NumCrediti DESC;
```

16) Trovare i corsi, restituendoli ordinati in numero di crediti decrescente. A parità di numero di crediti ordinare i corsi alfabeticamente in base al nome

a.

```
select *
from corsi
order by NumCrediti DESC, NomeC;
```

Interrogazioni che coinvolgono più tabelle in join tra di loro.

17) Trovare matricola, cognome e nome degli studenti di Verona che hanno preso almeno un 30

a.

```
select Studenti.Matricola, Studenti.CognomeS, Studenti.NomeS
from studenti, esami
where studenti. Matricola=Esami. Matricola
and Voto=30;
```

Soluzione alternativa in cui solo per comodità di scrittura rinomino le istanze delle tabelle STUDENTI e CORSI rispettivamente come S e C (vedremo in esercizi successivi che l'assegnazione di alias/soprannomi alle diverse istanze delle tabelle è necessario quando si hanno più istanze della stessa tabella nella stessa interrogazione per distinguere le diverse istanze)

b.

```
select S.Matricola, S.CognomeS, S.NomeS
from studenti S, corsi C
where S.Matricola=C.Matricola and Voto=30;
```

Altra sintassi possibile per eseguire il natural join (non spiegata in aula) basata sull'operatore JOIN

- c.

```
select Studenti.Matricola, Studenti.CognomeS, Studenti.NomeS
from studenti JOIN esami ON studenti. Matricola=Esami. Matricola
where Voto=30;
```

18) Trovare codice e nome dei corsi per i quali almeno uno studente di Verona ha preso 30 all'esame.

- a.

```
select C.CordCorso, C.NomeC
from studenti S, esami E, Corsi C
where S. Matricola=E. Matricola
and C.CordCorso=E.CodCorso
and Voto=30
and CittàRes='Verona';
```

19) Trovare codice, nome e cognome degli studenti che ha preso un voto superiore a 25 ad almeno uno tra i corsi 'Fondamenti Inf 1' e 'Fondamenti Inf 2'

- a.

```
select Studenti.Matricola, Studenti.CognomeS, Studenti.NomeS
from Esami, Corsi
where Corsi.CordCorso=Esami.CodCorso
and Voto>25
and (NomeC='Fondamenti Inf 1' OR NomeC='Fondamenti Inf 2');
```

Funzioni aggregate

20) Contare quanti studenti sono presenti nella tabella studenti

- a.

```
select count(*)
from studenti;
```
- b.

```
/* In questa versione alternativa associo il nome/alias numeroStudenti a count(*) */
select count(*) as numeroStudenti
from studenti;
```

21) Contare da quante città diverse provengono gli studenti presenti nella base di dati

- a.

```
/* La parola chiave DISTINCT all'interno dell'operatore COUNT(..) serve per specificare che
voglio contare quanti valori diversi assume CittàRes nelle tuple selezionate */
select count(DISTINCT CittàRes)
from studenti;
```

22) Contare per quanti studenti è noto il numero di telefono

- a.

```
/* L'operatore COUNT(attributo) considera solo le tuple in cui il campo specificato non
assume valore NULL */
select count(NumTelefono)
from studenti;
```

23) Selezionare il voto più alto e quello più basso preso dallo studente con matricola 'M1000'

- a.

```
select MAX(Voto) as VotoMassimo, MIN(Voto) as VotoMinimo
from esami
where Matricola='M1000';
```

24) Calcolare la media voti dello studente con matricola 'M1000'

a.

```
select AVG(Voto) as MediaVoti
from esami
where Matricola='M1000';
```

25) Contare quanti studenti di Verona hanno superato l'esame di Basi di dati con un voto superiore a 25

a.

```
select COUNT(*) as StudentiVerora_EsamePassato
from Esami, Studenti
where Esami.Matricola=Studenti.Matricola
and Studenti.CittàRes='Verona' and Voto>25;
```

26) Calcolare la somma degli stipendi dei docenti del Dipartimento di Fisica

a.

```
select SUM(Stipendio) as SommaStipendi
from Docenti
where DipartimentoDiAfferenza='Dipartimento di Fisica';
```