

**Risolvere i seguenti esercizi:**

1. Trovare numero e descrizione di tutti gli articoli.

**> select nroart,descrizione from articoli;**

2. Trovare cognome e nome di tutti i clienti che hanno un fido di almeno 800.

**>select cognome,nome from clienti where fido>=800;**

3. Ricavare il numero d'ordine per tutti gli ordini emessi dal cliente '124' il 5/9/2002.

**>select nroordine from ordini where codcliente=124 and data='2002/09/05'**

4. Trovare numero, cognome e nome di tutti i clienti che sono rappresentati dal rappresentante '03' o '12'.

**>select codcliente,cognome,nome from clienti where codrapp=03 or codrapp=12**

5. Trovare numero e descrizione di tutti gli articoli che non appartengono alla classe 'CS'.

**>select nroart,descrizione from articoli where categoria != 'cs';**

6. Trovare numero e descrizione di tutti gli articoli che hanno fra 100 e 200 pezzi disponibili.

**>select nroart,descrizione from articoli where giacenza>=100 and giacenza>=200;**

7. Ricavare numero, descrizione e valore a magazzino (dato da pezzi disponibili\*prezzo) di ciascun articolo in classe 'EL'.

**>select nroart,descrizione,giacenza\*prezzounitario as valore from articoli where categoria='el';**

8. Mostrare numero, descrizione e valore a magazzino di ciascun articolo il cui valore a magazzino sia almeno 1000.

**>>select nroart,descrizione,giacenza\*prezzounitario as valore from articoli where giacenza\*prezzounitario>1000;**

9. Mostrare numero e descrizione di ciascun articolo la cui classe sia 'CS' o 'SP'.

**>select nroart,descrizione from articoli where categoria='cs' or categoria='sp';**

10. Trovare numero, cognome e nome di ogni cliente il cui nome inizi con la lettera 'D'.

**>select cognome,nome from clienti where nome like 'D%';**

11. Mostrare tutti i dettagli degli articoli, ordinando rispetto alla descrizione.

**>select \* from articoli order by descrizione;**

12. Mostrare tutti i dettagli degli articoli, ordinando rispetto a classe articolo e numero articolo.

```
>select * from articoli order by categoria,nroart;
```

Nei seguenti esercizi usare il join sia esplicito che implicito:

1. Per ogni ordine, mostrare numero e data dell'ordine insieme a numero, cognome e nome del cliente che ha emesso l'ordine.

```
>select o.nroordini,o.data,c.codcliente,c.cognome,c.nome from ordini o,clienti c  
where c.codcliente=o.codcliente;
```

```
>select o.nroordini,o.data,c.codcliente,c.cognome,c.nome from ordini o join clienti c  
on c.codcliente=o.codcliente;
```

2. Per ogni ordine emesso il 5/09/2002, mostrare numero e data dell'ordine insieme al numero, cognome e nome del cliente che ha emesso l'ordine.

```
>select o.nroordini,o.data,c.codcliente,c.cognome,c.nome from ordini o,clienti c  
where c.codcliente=o.codcliente and data='2002/09/05';
```

```
>select o.nroordini,o.data,c.codcliente,c.cognome,c.nome from ordini join clienti c on  
c.codcliente=o.codcliente and data='2002/09/05';
```

3. Per ogni ordine, mostrare numero e data dell'ordine insieme a numero articolo, numero dei pezzi ordinati e prezzo richiesto per ciascuna riga che compone l'ordine.

```
>select o.nroordine,o.data,a.nroart,k.qtaordinata,k.przrichiesto from ordini o,articoli  
a,dettordini k where o.nroordine=k.nroordine and a.nroart=k.nroarticolo;
```

```
>select o.nroordine,o.data,a.nroart,k.qtaordinata,k.przrichiesto from ordini o join  
articoli a join dettordini k on o.nroordine=k.nroordine and a.nroart=k.nroarticolo;
```

**Risolvere i seguenti esercizi:**

1. Trovare quanti clienti hanno un saldo inferiore al loro fido.

**>select count(\*) from clienti where fido>saldo;**

2. Trovare il totale di tutti i saldi di tutti i clienti rappresentati dal rappresentante '12' e che hanno un saldo inferiore al loro fido.

**>select count(\*) from clienti where codrappr='12' and saldo<fido;**

3. Elencare numero, descrizione e pezzi disponibili di tutti gli articoli per i quali il numero di pezzi disponibili è superiore alla media.

**>select nroart,descrizione,giacenza from articoli where giacenza >(select avg(giaccenza) as totale from articoli);**

4. Trovare l'articolo meno costoso nel database.

**>select \* from articoli where prezzounitario=(select max(prezzounitario) from articoli);**

5. Trovare l'articolo più costoso nel database.

**>select \* from articoli where prezzounitario=(select min(prezzounitario) from articoli);**

6. Stabilire quanti clienti vi sono nel database.

**>select count(\*) as Clienti\_nel\_database from clienti;**

7. Elencare la somma dei saldi di tutti i clienti per ogni rappresentante,ordinando e raggruppando i risultati rispetto al numero del rappresentante.

**>select codrappr,sum(saldo) as totale from clienti group by codrappr order by codrappr;**

8. Elencare le classi degli articoli e la somma del valore dei pezzi disponibili, raggruppando i risultati per categoria di articolo.

**>select categoria,giacenza\*prezzounitario as valore from articoli group by categoria;**

9. Trovare articoli per i quali la descrizione non sia nota.

**>select nroart,descrizione from articoli where descrizione is null;**

10. Trovare numero cliente, cognome e nome per ogni cliente che ha emesso un ordine il 5 settembre 2002 (risolvere utilizzando l'annidamento).

**>select \* from clienti where codcliente in (select distinct c.codcliente from ordini o, clienti c where c.codcliente = o.codcliente and o.data='2002/09/05');**

11. Risolvere l'esercizio 10, utilizzando l'operatore EXISTS.

**>select \* from clienti where exists (select \* from ordini o where o.data='2002/09/05' and clienti.codcliente = o.codcliente);**

12. Trovare numero, cognome e nome di ogni cliente che non ha emesso ordini il 5 settembre 2002.

```
>select * from clienti where codcliente in (select distinct c.codcliente from ordini o, clienti c where c.codcliente = o.codcliente and o.data!='2002/09/05');
```

13. Per ogni ordine, elencate numero ordine, data ordine, numero articolo, descrizione articolo, e categoria per ciascun articolo che compone l'ordine.

```
>select o.nroordine,k.data,a.nroart,a.descrizione,a.categoria from dettordini o, ordini k , articoli a where a.nroart=o.nroarticolo and o.nroordine=k.nroordine;
```

14. Risolvere l'esercizio 13, ma questa volta ordinando le righe per categoria di articolo e quindi per numero d'ordine.

```
>select o.nroordine,k.data,a.nroart,a.descrizione,a.categoria from dettordini o, ordini k , articoli a where a.nroart=o.nroarticolo and o.nroordine=k.nroordine order by a.categoria,o.nroordine;
```

15. Trovare numero, cognome e nome di ogni rappresentante che abbia almeno un cliente con un fido di \$2.000 (risolvere utilizzando una query con annidamento)

```
>select codrappr,cognome,nome from rappresentanti where codrappr in (select codrappr from clienti where fido>=2000);
```

16. Risolvere l'esercizio 15 senza utilizzare una query annidata.

```
>select r.codrappr,r.cognome,r.nome from rappresentanti r,clienti c where r.codrappr=c.codrappr and c.fido>=2000;
```

17. Trovare numero, cognome e nome di ogni cliente che al momento ha nel file un ordine per un ferro da stiro.

```
>select c.codcliente,c.cognome,c.nome from clienti c,ordini o ,dettordini k,articoli a where c.codcliente=o.codcliente and o.nroordine=k.nroordine and a.nroart=nroarticolo and a.descrizione='ferrostiro';
```

-18. Elencare numero, descrizione e categoria articolo per ogni coppia di articoli che appartengono alla stessa categoria e che hanno lo stesso numero di magazzino.

```
>select a.nroart,a.descrizione,a.categoira from categoria a,categoria z where a.categoria=z.categoria and a.giacenza=z.giacenza and a.nroart=z.nroart;
```

19. Elencare descrizione e numero dell'articolo, nonché numero e data ordine per ogni ordine emesso da Mary Nelson che includa una TavolaPesi.

```
>select a.descrizione,a.nroart,o.nroordine,o.data from articoli a, ordini o,clienti c,dettordini k where c.codcliente = o.codcliente and c.nome='Mary' and c.cognome='Nelson' and a.descrizione='Tavolapesi' and o.nroordine=k.nroordine and a.nroart=k.nroarticolo;
```

**20. Elencare descrizione e numero dell'articolo, nonché numero e data ordine per ogni ordine emesso da Mary Nelson che non includa una TavolaPesi.**

```
>>select a.descrizione,a.nroart,o.nroordine,o.data from articoli a, ordini o,clienti c,dettordini k where c.codcliente = o.codcliente and c.nome='Mary' and c.cognome='Nelson' and a.descrizione!='Tavolapesi' and o.nroordine=k.nroordine and a.nroart=k.nroarticolo;
```

**21. Elencare numero e data per ogni ordine che o è stato emesso da Mary Nelson o include un ferro da stiro.**

```
>select c.nome,c.cognome,o.nroordine,o.data,a.descrizione from ordini o,clienti,c dettordini d,articoli a where o.codcliente =c.codcliente and o.nroordine=d.nroordine and d.nroarticolo=a.nroart and ((a.descrizione='ferrostiro') or (c.nome='mary' and c.cognome='nelson'));
```

**22. Elencare numero e data per ogni ordine che è stato emesso da Mary Nelson e che non include un ferro da stiro.**

```
>select c.nome,c.cognome,o.nroordine,o.data,a.descrizione from ordini o,clienti,c dettordini d,articoli a where o.codcliente =c.codcliente and o.nroordine=d.nroordine and d.nroarticolo=a.nroart and ((a.descrizione!='ferrostiro') and (c.nome='mary' and c.cognome='nelson'));
```

**24. Elencare numero articolo, descrizione, prezzo unitario e categoria per ogni articolo che abbia un prezzo unitario maggiore del prezzo unitario di ogni articolo nella categoria 'CS'. Risolvere utilizzando l'operatore ALL .**

```
>select a.nroart, a.descrizione, (a.prezzounitario) from articoli a where a.prezzounitario >= all (select prezzounitario from articoli where categoria = 'cs');
```

**25. Risolvere l'esercizio 24, utilizzando l'operatore ANY.**

```
>select a.nroart, a.descrizione, (a.prezzounitario) from articoli a where a.prezzounitario < any (select prezzounitario from articoli where categoria = 'cs');
```

**26. Risolvere l'esercizio 24, utilizzando un operatore aggregato.**

```
>select a.nroart, a.descrizione, (a.prezzounitario) from articoli a where a.prezzounitario >= (select max(prezzounitario) from articoli where categoria = 'cs');
```

**Risolvere i seguenti esercizi di creazione delle viste:**

1. Definire una vista chiamandola PiccoliClienti con schema (nro\_cliente, cognome, nome, via, saldo, fido) che contenga i clienti il cui fido sia di \$1.000 o inferiore.

**>create view piccoliclienti(nro\_cliente,cognome,nome,via,saldo,fido)as select codcliente,cognome,nome,via,saldo,fido where fido>=1000;**

2. Scrivere una query per reperire numero, nome e cognome di ogni cliente nella vista il cui saldo supera il fido.

**>select nro\_cliente,cognome,nome from piccoliclienti where saldo>fido;**

3. Scrivere la query che il DBMS di fatto esegue.

**>select codcliente,cognome,nome from clienti where saldo>fido and fido>=1000;**

4. L'aggiornamento del database tramite questa vista crea qualche problema? Se così fosse, quali sono? E se no, perché no?

**>**

5. Definire una vista OrdiniClienti con schema (nro\_cliente, cognome, nome, saldo, nro\_ordine, data\_ordine) che contenga ogni ordine che al momento è su file.

**>create view ordiniclienti(nro\_cliente,cognome,nome,saldo,nro\_ordine,data\_ordine) as select c.codcliente,c.cognome,c.nome,c.saldo,o.nroordine,o.data from clienti c,ordini o where c.codcliente=o.codcliente;**

6. Scrivere una query per reperire numero del cliente, nome, cognome, numero ordine e data ordine per ogni ordine nella vista per ogni cliente il cui saldo è maggiore di \$500.

**>select nro\_cliente,nome,cognome,nro\_ordine from ordiniclienti where saldo>500;**

7. Scrivere la query che il DBMS di fatto esegue.

**>select c.codcliente,c.nome,c.cognome,o.nroordine from clienti c,ordini o where saldo>500 and c.codcliente=o.codcliente;**

8. L'aggiornamento del database tramite questa vista crea qualche problema? Se così fosse, quali sono? E se no, perché no?

**>**

9. Definire una vista chiamandola TotaleOrdini con schema (nro\_ordine, tot\_ordine) di ogni ordine che al momento è su file. (Il totale ordine è la somma del numero di pezzi ordinati moltiplicato per il prezzo richiesto in ciascuna riga ordine per quell'ordine.)

**>create view totaleordini(nro\_ordine,tot\_ordine)as select nroordine,sum(qtaordinata)\*przrichiesto from dettordini group by nroordine,qtaordinata,nroarticolo;**

10. Scrivere una query per reperire il numero ordine e il totale ordine per ogni ordine nella vista il cui totale è maggiore di \$500, ordinando i risultati per numero ordine.

**>select nro\_ordine,tot\_ordine from totaleordini where tot\_ordine>500 order by nro\_ordine;**

11. Scrivere la query che il DBMS di fatto esegue.

**>**

**12. L'aggiornamento del database tramite questa vista crea qualche problema? Se così fosse, quali sono? E se no, perché no?**

>

F P R