

Sia dato il seguente schema di base di dati:

CLIENTI(Codice_Cliente, Nome, Cognome, NomeDitta, Indirizzo, Citta, Telefono)

CATEGORIE_PRODOTTI(Codice_Categoria, Nome)

PRODOTTI(Codice_Prodotto, Nome, Produttore, Cod_Categoria, Scadenza, PrezzoVendita)

VENDITE(Cod_Cliente, Data, Quantita, Cod_Prodotto, Sconto)

con i seguenti vincoli di integrità referenziale:

VENDITE.Cod_Cliente → CLIENTI.Codice_Cliente

VENDITE.Cod_Prodotto → PRODOTTI.Codice_Prodotto

PRODOTTI.Cod_Categoria → CATEGORIE_PRODOTTI.Codice_Categoria

(NB si consideri presente un vincolo per il quale, nella tabella CLIENTI, se sono privati, si debba specificare Nome e Cognome mentre, se sono ditte, si debba specificare il nome della ditta).

1. Scrivere le query SQL per trovare:

1. Il nome dei prodotti con prezzo > 100

```
SELECT nome
FROM prodotti
WHERE PrezzoVendita > 100
```

2. Il cognome e il nome dei clienti che abitano a Parma.

```
SELECT Cognome, Nome
FROM Clienti
WHERE Città='Parma'
```

3. Cognome e nome dei clienti che hanno acquistato prodotti che scadevano nel 2008 (usare la funzione *year(data)* per estrarre l'anno dalla data)

```
SELECT Cognome, C.Nome
FROM Clienti C, Prodotti P, Vendite V
WHERE V.Cod_Cliente = C.Codice_cliente
AND V.Cod_prodotto = Codice_Prodotto
AND year(Scadenza) = 2008
```

4. Nome dei prodotti che appartengono alla categoria Alimentari.

```
SELECT P.Nome, CP.Nome
FROM Prodotti P JOIN Categorie_Prodotti CP ON P.Cod_Categoria = CP.Codice_Categoria
WHERE CP.Nome = 'Alimentari'
```

5. Nome e cognome dei clienti di nome Giuseppe che hanno acquistato prodotti di costo > 150

```
SELECT C.Nome, C. Cognome
FROM Clienti C, Prodotti P, Vendite V
WHERE V.Cod_Cliente = C.Codice_Cliente
AND P.Codice_Prodotto=V.Cod_Prodotto
AND C.Nome='Giuseppe'
AND P.PrezzoVendita > 150
```

6. La massima quantità di un prodotto venduta a un cliente di nome Piero

```
SELECT max(V. Quantita)
FROM Clienti C, Vendite V
WHERE V.Cod_Cliente = C.Codice_Cliente
AND C.Nome='Piero'
```

7. I ricavi totali ottenuti per la vendita di ciascun prodotto dal 1/1/2012 al 31/12/2012

```
SELECT P. Nome, sum(V.Quantita * P.PrezzoVendita)
FROM Prodotti P JOIN Vendite V on P.Codice_Prodotto=V.Cod_Prodotto
WHERE year(V.Data) = 2012
GROUP BY P.Nome
```

8. Il costo del prodotto più caro acquistato da un cliente residente a Parma

```
SELECT max(P.PrezzoVendita)
FROM Clienti C, Prodotti P, Vendite V
WHERE V.Cod_Cliente = C.Codice_cliente
AND V.Cod_prodotto = P.Codice_Prodotto
AND Citta='Parma'
```

9. Il prezzo medio di vendita dei prodotti di ciascuna categoria

```
SELECT CP.Nome, avg(PrezzoVendita)
FROM Prodotti P JOIN Categorie_Prodotti CP ON P.Cod_Categoria = CP.Codice_Categoria
GROUP BY CP.Nome
```

10. La spesa totale dei clienti residenti nelle diverse città (un totale per ogni città).

```
SELECT C.Citta, sum(V.Quantita * P.PrezzoVendita)
FROM Prodotti P JOIN Vendite V ON P.Codice_Prodotto=V.Cod_Prodotto
JOIN Clienti C on C.Codice_Cliente=V.Cod_Cliente
GROUP BY C.Citta
```

11. I ricavi totali anno per anno.

```
SELECT year(V.Data), sum(V.Quantita * P.PrezzoVendita)
FROM Prodotti P JOIN Vendite V on P.Codice_Prodotto=V.Cod_Prodotto
GROUP BY year(V.Data)
```

12. I ricavi totali, anno per anno, per ciascuna categoria di prodotti.

```
SELECT CP.Nome, year(V.Data), sum(V.Quantita * P.PrezzoVendita)
FROM Prodotti P JOIN Vendite V on P.Codice_Prodotto=V.Cod_Prodotto
JOIN Categorie_Prodotti CP ON P.Cod_Categoria = CP.Codice_Categoria
GROUP BY CP.Nome, year(V.Data)
```

13. Le categorie per le quali, nel 2012, si sono avuti ricavi maggiori di 20000.

```
SELECT CP.Nome, sum(V.Quantita * P.PrezzoVendita) AS Ricavitali
FROM Prodotti P JOIN Vendite V on P.Codice_Prodotto=V.Cod_Prodotto
JOIN Categorie_Prodotti CP ON P.Cod_Categoria = CP.Codice_Categoria
WHERE year(V.Data)=2012
GROUP BY CP.Nome
HAVING Ricavitali > 20000
```

14. L'elenco dei prodotti che sono più cari di almeno un prodotto della categoria Alimentari.

```
SELECT Nome
FROM Prodotti
WHERE PrezzoVendita > ANY ( SELECT PrezzoVendita
                             FROM Prodotti P, CategorieProdotti CP
                             WHERE CP.Codice_Categoria=P.Cod_Categoria
                             AND CP.Nome='Alimentari' )
```

15. Tutti i prodotti in vendita, in ordine di quantità venduta nel 2010.

```
SELECT P.Nome, sum(Quantita) as Quantitatotale
FROM Prodotti P, Vendite V
WHERE P.Codice_Prodotto=V.Cod_Prodotto
AND year(V.data) = 2010
GROUP BY P.Nome
ORDER BY Quantitatotale
```

16. Il prodotto più caro fra quelli acquistati da Piero Neri nel 2010 (usando una query nidificata).

```
SELECT P.Nome
FROM Prodotti P, Vendite V, Clienti C
WHERE P.Codice_Prodotto=V.Cod_Prodotto
AND V.Cod_Cliente=C.Codice_Cliente
AND C.Nome='Piero'
AND C.Cognome='Neri'
AND year(V.Data)=2010
AND P.PrezzoVendita = ( SELECT max(PrezzoVendita)
                        FROM Prodotti P1, Vendite V1, Clienti C1
                        WHERE P1.Codice_Prodotto=V1.Cod_Prodotto
                        AND V1.Cod_Cliente=C1.Codice_Cliente
                        AND C1.Nome='Piero'
                        AND C1.Cognome='Neri'
                        AND year(V1.Data)=2010 )
```

17. L'elenco dei prodotti più cari per ogni categoria, oltre alla la categoria e il prezzo corrispondenti, ordinati in ordine decrescente di prezzo.

```
SELECT CP.Nome, max(PrezzoVendita) AS PrezzoMax
FROM Prodotti P, CategorieProdotti CP
WHERE CP.Codice_Categoria=P. Cod_Categoria
GROUP BY CP.Nome
ORDER BY PrezzoMax DESC
```

18. La categoria di prodotti per cui i ricavi totali, nel 2011, sono stati i più bassi.

```
SELECT CP.Nome, sum(V.Quantita * P.PrezzoVendita) as RicaviTotali
FROM Prodotti P JOIN Vendite V on P.Codice_Prodotto=V.Cod_Prodotto
      JOIN Categorie_Prodotti CP ON P.Cod_Categoria = CP.Codice_Categoria
WHERE year(V.Data)=2011
GROUP BY CP.Nome
HAVING RicaviTotali <= ALL (SELECT sum(V1.Quantita * P1.PrezzoVendita)
                          FROM Prodotti P1 JOIN Vendite V1 on P1.Codice_Prodotto=V1.Cod_Prodotto
                          JOIN Categorie_Prodotti CP1 ON P1.Cod_Categoria = CP1.Codice_Categoria
                          WHERE year(V1.Data)=2011
                          GROUP BY CP1.Nome )
```

19. Il cliente che ha acquistato merce per il valore complessivo più alto dal 1/1/2011 a oggi.

```
SELECT C.Nome, C.Cognome, sum(V.Quantita * P.PrezzoVendita) AS TotAcquisti
FROM Prodotti P JOIN Vendite V ON P.Codice_Prodotto=V.Cod_Prodotto
      JOIN Clienti C on C.Codice_Cliente=V.Cod_Cliente
WHERE V.Data>='2011-1-1'
GROUP BY C.Codice_Cliente, C.Nome, C. Cognome
HAVING TotAcquisti>=ALL ( SELECT sum(V1.Quantita * P1.PrezzoVendita)
                        FROM Prodotti P1 JOIN Vendite V1 ON P1.Codice_Prodotto=V1.Cod_Prodotto
                        JOIN Clienti C1 on C1.Codice_Cliente=V1.Cod_Cliente
                        WHERE V1.Data>='2011-1-1'
                        GROUP BY C1.Cod_Cliente, C1.Nome, C1. Cognome )
```

20. Definire una vista ProdottiAlimentari che contenga le tuple contenute nel risultato della query 4.

```
CREATE VIEW ProdottiAlimentari(Codice_Prodotto, Nome, Produttore, Cod_Categoria, Scadenza, PrezzoVendita)
AS
( SELECT *
  FROM Prodotti P JOIN Categorie_Prodotti CP ON P.Cod_Categoria = CP.Codice_Categoria
  WHERE CP.Nome = 'Alimentari' )
```

21. Formulare le query 14, 16, 18, 19 definendo un'opportuna vista che poi semplifichi il codice dell'interrogazione.
Osservare come cambia la leggibilità del codice.

(14.)

```
SELECT Nome
FROM Prodotti
WHERE PrezzoVendita >= ANY ( SELECT PrezzoVendita
                              FROM ProdottiAlimentari )
```

(16.)

```
CREATE VIEW AcquistiNeri2010(CodiceProdotto, Nome, Quantita, PrezzoVendita)
AS
(
    SELECT P.CodiceProdotto, P.Nome, V.Quantita, P.PrezzoVendita)
FROM Prodotti P, Vendite V, Clienti C
WHERE P.Codice_Prodotto=V.Cod_Prodotto
AND V.Cod_Cliente=C.Codice_Cliente
AND C.Nome='Piero'
AND C.Cognome='Neri'
AND year(V.Data)=2010 )
```

```
SELECT Nome
FROM AcquistiNeri2010
WHERE PrezzoVendita = (    SELECT max(PrezzoVendita)
                        FROM AcquistiNeri2010 )
```

(18.)

```
CREATE VIEW VenditeCategorie2011(CodiceCategoria, Nome, Ricavi)
AS
(
    SELECT CP.CodiceCategoria, CP.Nome, sum(V.Quantita * P.PrezzoVendita)
FROM Prodotti P JOIN Vendite V on P.Codice_Prodotto=V.Cod_Prodotto
JOIN Categorie_Prodotti CP ON P.Cod_Categoria = CP.Codice_Categoria
WHERE year(V.Data)=2011
GROUP BY CP.Nome )
```

```
SELECT Nome
FROM VenditeCategorie2011
WHERE Ricavi = (    SELECT min(Ricavi)
                  FROM VenditeCategorie2011 )
```

(19.)

```
CREATE VIEW Acquisticlienti2011oggi(CodiceCliente, Nome, Cognome, TotaleAcquisti)
AS
(
    SELECT C.CodiceCliente, C.Nome, C.Cognome, sum(V.Quantita * P.PrezzoVendita)
FROM Prodotti P JOIN Vendite V ON P.Codice_Prodotto=V.Cod_Prodotto
JOIN Clienti C on C.Codice_Cliente=V.Cod_Cliente
WHERE V.Data>'2011-1-1'
GROUP BY C.Codice_Cliente, C.Nome, C. Cognome )
```

```
SELECT Nome, Cognome, TotaleAcquisti
FROM Acquisticlienti2011oggi
WHERE TotaleAcquisti = (    SELECT max(TotaleAcquisti)
                        FROM Acquisticlienti2011oggi )
```

22. Aumentare il prezzo di tutti i prodotti alimentari del 10%. Verificare come 'reagisce' MySQL se proviamo ad aggiornare la vista ProdottiAlimentari invece della tabella Prodotti.

```
UPDATE Prodotti
SET PrezzoVendita=PrezzoVendita * 1.1
WHERE Cod_Categoria = ( SELECT Codice_Categoria
                        FROM CategorieProdotti
                        WHERE Nome='Alimentari' )
```

(aggiornando la vista, se consentito)

```
UPDATE ProdottiAlimentari
SET PrezzoVendita = PrezzoVendita * 1.1
```

23. Introdurre nella tabella Prodotti anche nuovi prodotti della categoria Cosmetici

```
INSERT INTO Categorie_Prodotti
VALUES ('CSM', 'Cosmetici')
```

```
INSERT INTO Prodotti      PRODOTTI(Codice_Prodotto, Nome, Produttore, Cod_Categoria, Scadenza, PrezzoVendita)
VALUES      ('CSM001', 'Rossetto Redlips', 'XXY', 'CSM', '2015-12-31', 15.50),
            ('CSM002', 'Ombretto Blackeyes', 'ZZX', 'CSM', '2014-06-08', 10.30)
```

24. Eliminare le tuple relative ai cosmetici dalla tabella Prodotti.

```
DELETE FROM Prodotti
WHERE Cod_categoria = ( SELECT Codice_Categoria
                       FROM CategorieProdotti
                       WHERE Nome='Cosmetici' )
```