



SQL: definizione schema

Si scriva il codice SQL che definisce il seguente schema relazionale

Clienti(codice, nome, indirizzo, p_iva)

Prodotti(codice, nome, descrizione, prezzo)

Fatture(codice, cliente, data)

RigheFattura(codice, fattura, prodotto, quantità, prezzo)

Vincoli di integrità referenziale

RigheFattura.fattura -> Fatture.codice

RigheFattura.prodotto -> Prodotti.codice

Fatture.cliente -> Cliente.nome

1



Schema relazionale con vincoli in forma grafica

CREATE SCHEMA fatture

```
CREATE TABLE clienti(  
  codice serial PRIMARY KEY,  
  nome varchar(20) NOT NULL,  
  indirizzo varchar(40) NOT NULL,  
  p_iva char(11) unique)
```

```
CREATE TABLE prodotti(  
  codice serial PRIMARY KEY,  
  nome varchar(20) NOT NULL,  
  descrizione varchar(256) NOT NULL,  
  prezzo decimal(8,2) NOT NULL)
```

2

Schema relazionale con vincoli in forma grafica

```
CREATE TABLE fatture(  
    codice serial PRIMARY KEY,  
    cliente integer NOT NULL REFERENCES Clienti(codice),  
    data date NOT NULL)
```

```
CREATE TABLE RigheFattura(  
    codice serial PRIMARY KEY,  
    fattura integer NOT NULL REFERENCES Fatture(codice),  
    prodotto integer NOT NULL REFERENCES Prodotti(Codice),  
    quantità integer NOT NULL,  
    prezzo decimal(8,2))
```

3

SQL: Interrogazioni

Date le tabelle precedentemente definite, si scriva il codice SQL che implementi le seguenti interrogazioni

1. Ricercare tutti i prodotti acquistati da 'pippo', mostrando la data della fattura, il nome del prodotto e il prezzo come da fattura
2. Ricercare tutti i prodotti con le rispettive fatture, mostrando il nome del prodotto, la data della fattura e il prezzo a cui è stato venduto e avendo cura di visualizzare anche i prodotti che non sono mai stati venduti
3. Calcolare il totale delle vendite sommando il prezzo di tutte le righe di tutte le fatture. Supponendo che l'azienda guadagni all'incirca il 10% su ogni vendita si calcoli anche il guadagno atteso.
4. Visualizzare per ogni fattura, il codice, la data, il numero di righe e il totale ottenuto sommando il prezzo associato a ciascuna riga

4



SQL: Interrogazioni

5. Ripetere l'interrogazione precedente ordinando i dati per data.
6. Visualizzare il nome di ogni cliente con il numero totale di fatture che gli sono state emesse includendo nel conto solo le fatture del 2009, visualizzando solo i clienti che hanno almeno 3 fatture e ordinando le fatture per nome del cliente.

5



SQL: Interrogazioni

1. Ricercare tutti i prodotti acquistati da 'pippo', mostrando la data della fattura, il nome del prodotto e il prezzo come da fattura

```
SELECT fatture.data, prodotti.nome, righefatture.prezzo
FROM clienti, fatture, righefatture, prodotti
WHERE clienti.codice=fatture.cliente
      AND fatture.codice=righefatture.fattura
      AND righefatture.prodotto=prodotti.codice
      AND clienti.nome='pippo'
```

6



SQL: Interrogazioni

2. Ricercare tutti i prodotti con le rispettive fatture, mostrando il nome del prodotto, la data della fattura e il prezzo a cui è stato venduto e avendo cura di visualizzare anche i prodotti che non sono mai stati venduti

```
SELECT fatture.data, prodotti.nome, righefatture.prezzo
FROM prodotti LEFT JOIN righefatture
      ON righefatture.prodotto=prodotti.codice
LEFT JOIN fatture ON fatture.codice=righefatture.fattura
```

7



SQL: Interrogazioni

3. Calcolare il totale delle vendite sommando il prezzo di tutte le righe di tutte le fatture. Supponendo che l'azienda guadagni all'incirca il 10% su ogni vendita si calcoli anche il guadagno atteso.

```
SELECT sum(righefatture.prezzo) AS "Totale vendite",
      sum(righefatture.prezzo)*0.1 AS "Guadagno atteso"
FROM righefatture
```

8



SQL: Interrogazioni

4. Visualizzare per ogni fattura, il codice, la data, il numero di righe e il totale ottenuto sommando il prezzo associato a ciascuna riga

```
SELECT fatture.codice, fatture.data,  
       sum(righefatture.prezzo) AS "Totale fattura"  
FROM   righefatture, fatture  
WHERE  righefatture.fattura=fatture.codice  
GROUP BY fatture.codice, fatture.data
```

9



SQL: Interrogazioni

5. Ripetere l'interrogazione precedente ordinando i dati per data.

```
SELECT fatture.codice, fatture.data,  
       sum(righefatture.prezzo) AS "Totale fattura"  
FROM   righefatture, fatture  
WHERE  righefatture.fattura=fatture.codice  
GROUP BY fatture.codice, fatture.data  
ORDER BY fatture.data
```

10



SQL: Interrogazioni

6. Visualizzare il nome di ogni cliente con il numero totale di fatture che gli sono state emesse includendo nel conto solo le fatture del 2009, visualizzando solo i clienti che hanno almeno 3 fatture e ordinando le fatture per nome del cliente.

```
SELECT clienti.nome, count(*) AS "Numero righe"  
FROM clienti, fatture  
WHERE fatture.cliente=clienti.codice  
      AND date_part('year',fatture.data)=2009  
GROUP BY clienti.nome  
HAVING count(*) >1  
ORDER BY clienti.nome
```