

PROGETTO S3/L5 - FRUNCILLO RAFFAELE ALESSANDRO

Sei un analista di dati che lavora per un'azienda di e-commerce. La tua azienda ha una tabella di vendite (vendite) che registra ogni transazione. Ogni record nella tabella vendite include un ID transazione, la categoria del prodotto, il costo a cui si è venduto e lo sconto applicato. Inoltre hai una seconda tabella dettagli_vendite che include ID cliente, la data della transazione dell'acquisto e la quantità.

Vendite

id_transazione	categoria	costo_vendita	sconto
1	Elettronica	500.00	20.00
2	Abbigliamento	120.50	10.00
3	Libri	30.00	5.50
4	Casa	250.75	15.00
5	Sport	80.20	12.00
6	Gioielli	350.00	5.00
7	Libri	75.99	50.50
8	Abbigliamento	45.50	3.75
9	Ferramenta	120.25	10.20
10	Elettronica	90.75	67.00
11	Casa	300.00	15.00
12	Elettronica	700.50	30.00
13	Abbigliamento	85.25	7.50
14	Libri	40.00	5.00
15	Sport	120.75	12.50
16	Cucina	90.99	10.25
17	Gioielli	420.00	35.00
18	Ferramenta	150.25	12.75
19	Salute e Bellezza	65.50	6.00
20	Giocattoli	60.75	5.25
21	Elettronica	350.00	15.00
22	Cucina	80.50	8.00
23	Abbigliamento	65.00	5.00
24	Libri	25.75	2.50
25	Sport	110.20	10.00
26	Casa	200.99	12.50
27	Gioielli	420.75	20.00
28	Ferramenta	90.50	7.25
29	Salute e Bellezza	55.25	6.50
30	Giocattoli	75.75	8.25

Dettagli_vendite

id_cliente	data_transaz	quantita	id_transaz
101	2024-01-08	25	1
102	2024-01-09	33	2
103	2024-01-09	49	3
104	2023-01-08	4	4
105	2024-01-08	91	5
106	2023-10-13	1	6
107	2023-02-14	2	7
108	2023-11-15	1	8
109	2023-12-16	3	9
110	2023-09-17	2	10
110	2023-05-18	15	11
112	2023-06-21	10	12
113	2023-07-24	5	13
101	2023-08-27	30	14
115	2023-09-30	20	15
116	2023-10-03	8	16
107	2023-11-06	12	17
108	2023-12-09	18	18
119	2023-01-12	22	19
120	2023-02-15	28	20
101	2023-02-18	15	21
102	2023-03-21	20	22
103	2023-04-24	25	23
101	2023-05-27	10	24
104	2023-06-30	18	25
105	2023-07-03	22	26
106	2023-09-09	12	27
107	2023-09-09	28	28
108	2023-10-12	30	29
109	2023-11-15	35	30

1. Creazione delle Tabelle:

- Scrivi le istruzioni SQL per creare le tabelle vendite e dettagli_vendite.

2. Inserimento dei Dati:

- Crea e inserisci un set di dati di esempio in entrambe le tabelle.

```
CREATE TABLE vendite (  
    id_transazione INT PRIMARY KEY,  
    categoria VARCHAR(255),  
    costo_vendita DECIMAL(10, 2),  
    sconto DECIMAL(5, 2)  
);  
  
CREATE TABLE dettagli_vendite (  
    id_cliente INT ,  
    data_transaz DATE,  
    quantita INT,  
    id_transaz INT,  
    FOREIGN KEY (id_transaz) REFERENCES vendite(id_transazione)  
);  
  
INSERT INTO vendite (id_transazione, categoria, costo_vendita, sconto)  
VALUES  
    (1, 'Elettronica', 500.00, 20.00),  
    (2, 'Abbigliamento', 120.50, 10.00),  
    (3, 'Libri', 30.00, 5.50),  
    (4, 'Casa', 250.75, 15.00),  
    (5, 'Sport', 80.20, 12.00),  
    (6, 'Gioielli', 350.00, 5.00),  
    (7, 'Libri', 75.99, 50.50),  
    (8, 'Abbigliamento', 45.50, 3.75),  
    (9, 'Ferramenta', 120.25, 10.20),  
    (10, 'Elettronica', 90.75, 67.00),  
    (11, 'Casa', 300.00, 15.00),  
    (12, 'Elettronica', 700.50, 30.00),  
    (13, 'Abbigliamento', 85.25, 7.50),  
    (14, 'Libri', 40.00, 5.00),  
    (15, 'Sport', 120.75, 12.50),  
    (16, 'Cucina', 90.99, 10.25),  
    (17, 'Gioielli', 420.00, 35.00),  
    (18, 'Ferramenta', 150.25, 12.75),  
    (19, 'Salute e Bellezza', 65.50, 6.00),  
    (20, 'Giocattoli', 60.75, 5.25),  
    (21, 'Elettronica', 350.00, 15.00),  
    (22, 'Cucina', 80.50, 8.00),  
    (23, 'Abbigliamento', 65.00, 5.00),  
    (24, 'Libri', 25.75, 2.50),  
    (25, 'Sport', 110.20, 10.00),  
    (26, 'Casa', 200.99, 12.50),  
    (27, 'Gioielli', 420.75, 20.00),  
    (28, 'Ferramenta', 90.50, 7.25),  
    (29, 'Salute e Bellezza', 55.25, 6.50),  
    (30, 'Giocattoli', 75.75, 8.25);  
  
INSERT INTO dettagli_vendite (id_cliente, data_transaz, quantita,  
id_transaz)  
VALUES  
    (101, '2024-01-08', 25, 1),  
    (102, '2024-01-09', 33, 2),  
    (103, '2024-01-09', 49, 3),
```

```
(104, '2023-01-08', 4, 4),
(105, '2024-01-08', 91, 5),
(106, '2023-10-13', 1, 6),
(107, '2023-02-14', 2, 7),
(108, '2023-11-15', 1, 8),
(109, '2023-12-16', 3, 9),
(110, '2023-09-17', 2, 10),
(110, '2023-05-18', 15, 11),
(112, '2023-06-21', 10, 12),
(113, '2023-07-24', 5, 13),
(101, '2023-08-27', 30, 14),
(115, '2023-09-30', 20, 15),
(116, '2023-10-03', 8, 16),
(107, '2023-11-06', 12, 17),
(108, '2023-12-09', 18, 18),
(119, '2023-01-12', 22, 19),
(120, '2023-02-15', 28, 20),
(101, '2023-02-18', 15, 21),
(102, '2023-03-21', 20, 22),
(103, '2023-04-24', 25, 23),
(101, '2023-05-27', 10, 24),
(104, '2023-06-30', 18, 25),
(105, '2023-07-03', 22, 26),
(106, '2023-09-09', 12, 27),
(107, '2023-09-09', 28, 28),
(108, '2023-10-12', 30, 29),
(109, '2023-11-15', 35, 30);
```

3. Query Semplici:

- *Seleziona tutte le vendite avvenute in una specifica data.*
- *Elenco delle vendite con sconti maggiori del 50%.*

```
SELECT d.*, v.costo_vendita
FROM dettagli_vendite d
JOIN vendite v ON d.id_transaz = v.id_transazione
WHERE d.data_transaz = "2024-01-08";
```

id_cliente	data_transaz	quantita	id_transaz	costo_vendita
101	2024-01-08	25	1	500.00
105	2024-01-08	91	5	80.20

```
SELECT d.*, v.costo_vendita, v.sconto
FROM dettagli_vendite d
JOIN vendite v ON d.id_transaz = v.id_transazione
WHERE v.sconto > 50;
```

id_cliente	data_transaz	quantita	id_transaz	costo_vendita	sconto
107	2023-02-14	2	7	75.99	50.50
110	2023-09-17	2	10	90.75	67.00

4. Aggregazione dei Dati:

- Calcola il totale delle vendite per categoria.
- Trova il numero totale di prodotti venduti per ogni categoria.

```
SELECT v.categoria, SUM(v.costo_vendita * d.quantita) as TOT_VENDITE  
FROM vendite v  
JOIN dettagli_vendite d ON v.id_transazione = d.id_transaz  
GROUP BY v.categoria;
```

categoria	TOT_VENDITE
Elettronica	24936.50
Abbigliamento	6073.25
Libri	3079.48
Casa	9924.78
Sport	11696.80
Gioielli	10439.00
Ferramenta	5599.25
Cucina	2337.92
Salute e Bellezza	3098.50
Giocattoli	4352.25

```
SELECT v.categoria, SUM(dettagli_vendite.quantita) as TOT_PRODOTTI FROM  
vendite v  
JOIN dettagli_vendite ON dettagli_vendite.id_transaz = v.id_transazione  
GROUP BY v.categoria;
```

categoria	TOT_PRODOTTI
Elettronica	52
Abbigliamento	64
Libri	91
Casa	41
Sport	129
Gioielli	25
Ferramenta	49
Cucina	28
Salute e Bellezza	52
Giocattoli	63

5. Funzioni di Data:

- Seleziona le vendite dell'ultimo trimestre.
- Raggruppa le vendite per mese e calcola il totale delle vendite per ogni mese.

```
SELECT DATE_FORMAT(d.data_transaz, '%d-%m-%Y') AS DATA, v.*
FROM vendite v
JOIN dettagli_vendite d ON v.id_transazione = d.id_transaz
WHERE d.data_transaz >= DATE_ADD(CURRENT_DATE(), INTERVAL -3 MONTH);
```

DATA	id_transazione	categoria	costo_vendita	sconto
08-01-2024	1	Elettronica	500.00	20.00
09-01-2024	2	Abbigliamento	120.50	10.00
09-01-2024	3	Libri	30.00	5.50
08-01-2024	5	Sport	80.20	12.00
13-10-2023	6	Gioielli	350.00	5.00
15-11-2023	8	Abbigliamento	45.50	3.75
16-12-2023	9	Ferramenta	120.25	10.20
06-11-2023	17	Gioielli	420.00	35.00
09-12-2023	18	Ferramenta	150.25	12.75
12-10-2023	29	Salute e Bellezza	55.25	6.50
15-11-2023	30	Giocattoli	75.75	8.25

```
SELECT MAX(DATE_FORMAT(d.data_transaz, '%Y')) as ANNO,
MAX(DATE_FORMAT(d.data_transaz, '%M')) as MESE, SUM(v.costo_vendita *
d.quantita) as TOT_VENDITE
FROM vendite v
JOIN dettagli_vendite d ON v.id_transazione = d.id_transaz
GROUP BY YEAR(d.data_transaz), MONTH(d.data_transaz)
ORDER BY YEAR(d.data_transaz), MONTH(d.data_transaz);
```

ANNO	MESE	TOT_VENDITE
2023	January	2444.00
2023	February	7102.98
2023	March	1610.00
2023	April	1625.00
2023	May	4757.50
2023	June	8988.60
2023	July	4848.03
2023	August	1200.00
2023	September	10179.50
2023	October	2735.42
2023	November	7736.75
2023	December	3065.25
2024	January	25244.70

Ho raggruppato anche per anno poiché ho più di un anno di dati (2023-2024)

7. Analisi degli Sconti:

- Trova la categoria con lo sconto medio più alto.

```
SELECT categoria, MAX(SCONTO_MEDIO) AS MAX_SCONTO_MEDIO
FROM (
    SELECT categoria, AVG(sconto) as SCONTO_MEDIO
    FROM vendite
    GROUP BY categoria
) AS t_sconto_medio
GROUP BY categoria
LIMIT 1;
```

categoria	MAX_SCONTO_MEDIO
Elettronica	33.000000

8. Variazioni delle Vendite:

- Confronta le vendite mese per mese per vedere l'incremento o il decremento delle vendite. Calcola l'incremento o decremento mese per mese

```
CREATE VIEW TOT_VENDITE_PER_MESE AS
SELECT
    MONTH(d.data_transaz) as MESE,
    MAX(DATE_FORMAT(d.data_transaz, '%M')) as MESE_NOME,
    SUM(v.costo_vendita * d.quantita) as TOT_VENDITE
FROM vendite v
JOIN dettagli_vendite d ON v.id_transazione = d.id_transaz
GROUP BY MESE
ORDER BY MESE;

SELECT
    T1.MESE,
    T1.MESE_NOME,
    T1.TOT_VENDITE AS VENDITE_CORR,
    T2.TOT_VENDITE AS VENDITE_PREC,
    COALESCE(T1.TOT_VENDITE - T2.TOT_VENDITE, 0) AS VARIAZIONE,
    CASE
        WHEN T1.TOT_VENDITE - T2.TOT_VENDITE > 0 THEN "+"
        WHEN COALESCE(T1.TOT_VENDITE - T2.TOT_VENDITE, 0) = 0 THEN "N.D."
        ELSE "-"
    END AS SEGNO
FROM TOT_VENDITE_PER_MESE T1
LEFT JOIN TOT_VENDITE_PER_MESE T2 ON T1.MESE = T2.MESE + 1
ORDER BY T1.MESE;
```

Per questo esercizio ho creato una View in modo da avere da parte una tabella con i mesi e le vendite per mese. Ho eseguito una left join sulla stessa tabella facendo unire i valori del mese della tabella 1 con i valori del mese successivo (MESE+1) della tabella 2.

MESE	MESE_NOME	VENDITE_CORR	VENDITE_PREC	VARIAZIONE	SEGNO
1	January	27688.70	NULL	0.00	N.D.
2	February	7102.98	27688.70	-20585.72	-
3	March	1610.00	7102.98	-5492.98	-
4	April	1625.00	1610.00	15.00	+
5	May	4757.50	1625.00	3132.50	+
6	June	8988.60	4757.50	4231.10	+
7	July	4848.03	8988.60	-4140.57	-
8	August	1200.00	4848.03	-3648.03	-
9	September	10179.50	1200.00	8979.50	+
10	October	2735.42	10179.50	-7444.08	-
11	November	7736.75	2735.42	5001.33	+
12	December	3065.25	7736.75	-4671.50	-

9. Analisi Stagionale:

- Confronta le vendite totali in diverse stagioni.

```
SELECT SUM(v.costo_vendita * d.quantita) as TOT_VENDITE,
       CASE
         WHEN DATE_FORMAT(d.data_transaz, '%m%d') between '0321' and
'0621' then 'PRIMAVERA'
         WHEN DATE_FORMAT(d.data_transaz, '%m%d') between '0622' and
'0922' then 'ESTATE'
         WHEN DATE_FORMAT(d.data_transaz, '%m%d') between '0923' and
'1220' then 'AUTUNNO'
         ELSE 'INVERNO'
       END AS STAGIONE
FROM vendite v
JOIN dettagli_vendite d ON v.id_transazione = d.id_transaz
GROUP BY STAGIONE;
```

TOT_VENDITE	STAGIONE
34791.68	INVERNO
15952.42	AUTUNNO
15796.13	ESTATE
14997.50	PRIMAVERA

10. Clienti Fedeli:

•Supponendo di avere una tabella clienti con i campi IDCliente e IDVendita, scrivi una query per trovare i top 5 clienti con il maggior numero di acquisti.

Si possono trovare i top 5 clienti senza la tabella Clienti:

```
SELECT d.id_cliente, SUM(v.costo_vendita * d.quantita) as TOT_VENDITE
FROM vendite v
JOIN dettagli_vendite d ON v.id_transazione = d.id_transaz
GROUP BY d.id_cliente
ORDER BY TOT_VENDITE DESC
LIMIT 5;
```

id_cliente	TOT_VENDITE
101	19207.50
105	11719.98
107	7725.98
112	7005.00
102	5586.50

Supponendo di avere una tabella clienti la query potrebbe essere:

```
SELECT c.id_cliente, SUM(v.costo_vendita * d.quantita) as TOT_VENDITE
FROM vendite v
JOIN clienti c ON v.id_transazione = c.id_vendita
JOIN dettagli_vendite d ON d.id_cliente = c.id_cliente
GROUP BY c.id_cliente
ORDER BY TOT_VENDITE DESC
LIMIT 5;
```