**Esercizi da Risolvere con SQL:**

1. **Analisi delle Vendite Mensili:**
   * *Domanda:* Trova il totale delle vendite (calcolato prezzo\*quantità) per ogni mese.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

SELECT

MONTH(DataTransazione) AS Mese,

ROUND(SUM(p.Prezzo \* t.QuantitaAcquistata),0) AS Importo\_VenditeTotali

FROM

`2.0schema`.prodotti p

JOIN

`2.0schema`.transazioni t ON p.ProdottoID = t.ProdottoID

GROUP BY

Mese;

1. **Prodotti più Venduti:**
   * *Domanda:* Identifica i tre prodotti più venduti e la loro quantità venduta.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

SELECT

ProdottoID,

SUM(QuantitaAcquistata) AS QuantitaTotale

FROM

`2.0schema`.transazioni

GROUP BY

ProdottoID

ORDER BY

QuantitaTotale DESC

LIMIT 3;

1. **Analisi Cliente:**
   * *Domanda:* Trova il cliente che ha effettuato il maggior numero di acquisti.

Maggior numero di transazioni per 1 cliente

SELECT

ClienteID,

COUNT(\*) AS NumeroAcquisti

FROM

`2.0schema`.transazioni

GROUP BY

ClienteID

ORDER BY

NumeroAcquisti DESC

LIMIT 1;

Immagine che contiene testo, Carattere, linea, schermata

Descrizione generata automaticamente

Maggior numero di acquisti per 1 cliente

SELECT

ClienteID,

SUM(QuantitaAcquistata) AS QuantitaTotaleAcquistata

FROM

`2.0schema`.transazioni

GROUP BY

ClienteID

ORDER BY

QuantitaTotaleAcquistata DESC

LIMIT 1;



1. **Valore medio della transazione:**
   * *Domanda:* Calcola il valore medio per trimetre di ogni transazione

SELECT

YEAR(DataTransazione) AS Anno,

QUARTER(DataTransazione) AS Trimestre,

ROUND(AVG(ImportoTransazione),2) AS ValoreMedioTrimestre

FROM

`2.0schema`.transazioni

GROUP BY

Anno, Trimestre;

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

SELECT

YEAR(DataTransazione) AS Anno,

QUARTER(DataTransazione) AS Trimestre,

ROUND(AVG(p.Prezzo \* t.QuantitaAcquistata),0) AS Importo\_VenditeTotali

FROM

`2.0schema`.prodotti p

JOIN

`2.0schema`.transazioni t ON p.ProdottoID = t.ProdottoID

GROUP BY

Anno, Trimestre;

SELECT

YEAR(DataTransazione) AS Anno,

QUARTER(DataTransazione) AS Trimestre,

ROUND(SUM(p.Prezzo \* t.QuantitaAcquistata),0) AS Importo\_VenditeTotali

FROM

`2.0schema`.prodotti p

JOIN

`2.0schema`.transazioni t ON p.ProdottoID = t.ProdottoID

GROUP BY

Anno, Trimestre;

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

1. **Analisi Categoria Prodotto:**
   * *Domanda:* Determina la categoria di prodotto con il maggior numero di vendite.

SELECT

Categoria,

SUM(t.QuantitaAcquistata) AS QuantitaTotaleVenduta

FROM `2.0schema`.transazioni t

JOIN `2.0schema`.prodotti p ON t.ProdottoID = p.ProdottoID

GROUP BY Categoria

ORDER BY QuantitaTotaleVenduta DESC

LIMIT 3;

Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamente

1. **Cliente Fedele:**
   * *Domanda:* Identifica il cliente con il maggior valore totale di acquisti.



SELECT

c.NomeCliente,

SUM(p.Prezzo \* t.QuantitaAcquistata) AS TotaleVendite

FROM

`2.0schema`.transazioni t

JOIN

`2.0schema`.clienti c ON t.ClienteID = c.ClienteID

JOIN

`2.0schema`.prodotti p ON t.ProdottoID = p.ProdottoID

GROUP BY

c.ClienteID, c.NomeCliente

ORDER BY

TotaleVendite DESC

LIMIT 1;

1. **Spedizioni Riuscite:**
   * *Domanda:* Calcola la percentuale di spedizioni con "Consegna Riuscita".



SELECT

COUNT(\*) AS NumeroTotaleSpedizioni,

SUM(CASE WHEN s.StatusConsegna = 'Consegna Riuscita'

THEN 1 ELSE 0 END) AS NumeroSpedizioniRiuscite,

ROUND((SUM(CASE WHEN s.StatusConsegna = 'Consegna Riuscita'

THEN 1 ELSE 0 END) / COUNT(\*)) \* 100 ,0) AS PercentualeRiuscite

FROM `2.0schema`.spedizioni s;

1. **Prodotto con la Migliore Recensione:**
   * *Domanda:* Trova il prodotto con la recensione media più alta.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

La media per categoria

SELECT

p.Categoria,

AVG(r.Rating) AS MediaRecensioniCategoria

FROM `2.0schema`.prodotti p

LEFT JOIN `2.0schema`.ratings r ON p.ProdottoID = r.ProductID

GROUP BY p.Categoria

ORDER BY MediaRecensioniCategoria DESC;

Immagine che contiene testo, schermata, numero, menu

Descrizione generata automaticamente

La media di ciascun mese per ogni categoria

SELECT

p.Categoria,

YEAR(t.DataTransazione) AS Anno,

MONTH(t.DataTransazione) AS Mese,

AVG(r.Rating) AS MediaRecensioni

FROM `2.0schema`.prodotti p

LEFT JOIN `2.0schema`.ratings r ON p.ProdottoID = r.ProductID

LEFT JOIN `2.0schema`.transazioni t ON r.ProductID = t.ProdottoID

GROUP BY p.Categoria, Anno, Mese

HAVING Mese IS NOT NULL

ORDER BY p.Categoria, Anno, Mese;

1. **Analisi Temporale:**
   * *Domanda:* Calcola la variazione percentuale nelle vendite rispetto al mese precedente.

SELECT

Anno,

Mese,

Categoria,

ROUND(AVG(VenditeTotali), 0) AS MediaVenditeTotali,

ROUND((AVG(VenditeTotali) - LAG(AVG(VenditeTotali)) OVER (PARTITION BY Categoria ORDER BY Anno, Mese)) / LAG(AVG(VenditeTotali)) OVER (PARTITION BY Categoria ORDER BY Anno, Mese) \* 100, 0) AS VariazionePercentuale,

CASE WHEN (AVG(VenditeTotali) - LAG(AVG(VenditeTotali)) OVER (PARTITION BY Categoria ORDER BY Anno, Mese)) > 0 THEN '+' ELSE '-' END AS SegnoVariazione

FROM (

SELECT

p.Categoria,

YEAR(t.DataTransazione) AS Anno,

MONTH(t.DataTransazione) AS Mese,

AVG(p.Prezzo \* t.QuantitaAcquistata) AS VenditeTotali

FROM `2.0schema`.prodotti p

LEFT JOIN `2.0schema`.transazioni t ON p.ProdottoID = t.ProdottoID

GROUP BY p.Categoria, Anno, Mese

) AS VenditeTotaliPerMese

GROUP BY Categoria, Anno, Mese

HAVING MediaVenditeTotali IS NOT NULL

ORDER BY Categoria, Anno, Mese;

Immagine che contiene testo, schermata, numero

Descrizione generata automaticamente

1. **Quantità di Prodotti Disponibili:**
   * *Domanda:* Determina la quantità media disponibile per categoria di prodotto.

SELECT

p.Categoria,

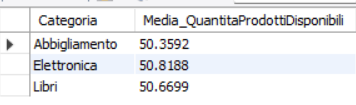
AVG(p.QuantitaDisponibile) AS Media\_QuantitaProdottiDisponibili

FROM

`2.0schema`.prodotti p

GROUP BY

p.Categoria;



1. **Analisi Spedizioni:**
   * *Domanda:* Trova il metodo di spedizione più utilizzato.

SELECT

MetodoSpedizione,

COUNT(\*) AS NumeroTransazioni,

ROUND((COUNT(\*) / (SELECT COUNT(\*) FROM `2.0schema`.transazioni)) \* 100, 2) AS PercentualeTotale

FROM

`2.0schema`.transazioni

GROUP BY

MetodoSpedizione

ORDER BY

NumeroTransazioni DESC

LIMIT 1;

Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamente

SELECT

MetodoSpedizione,

COUNT(\*) AS NumeroSpedizioni,

ROUND((COUNT(\*) / (SELECT COUNT(\*) FROM `2.0schema`.spedizioni)) \* 100, 2) AS PercentualeTotale

FROM

`2.0schema`.spedizioni

GROUP BY

MetodoSpedizione

ORDER BY

NumeroSpedizioni DESC

LIMIT 1;



1. **Analisi dei Clienti:**
   * *Domanda:* Calcola il numero medio di clienti registrati al mese.

SELECT

YEAR(DataRegistrazione) AS Anno,

MONTH(DataRegistrazione) AS Mese,

COUNT(DISTINCT ClienteID) AS NumeroClientiRegistrati

FROM

`2.0schema`.clienti

GROUP BY

Anno, Mese;

Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente

SELECT

ROUND(COUNT(\*) / 12, 0) AS MediaClientiMensile

FROM

`2.0schema`.clienti

WHERE

YEAR(DataRegistrazione) = 2022;

Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente

1. **Prodotti Rari:**
   * *Domanda:* Identifica i prodotti con una quantità disponibile inferiore alla media.

SELECT

Categoria,

AVG(QuantitaDisponibile) AS MediaQuantitaCategoria

FROM

`2.0schema`.prodotti

GROUP BY

Categoria;

SELECT

NomeProdotto,

Categoria,

QuantitaDisponibile,

AVG(QuantitaDisponibile) OVER (PARTITION BY Categoria) AS MediaQuantitaCategoria,

CASE

WHEN QuantitaDisponibile > AVG(QuantitaDisponibile) OVER (PARTITION BY Categoria) THEN '+'

WHEN QuantitaDisponibile < AVG(QuantitaDisponibile) OVER (PARTITION BY Categoria) THEN '-'

ELSE '='

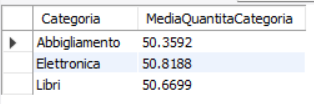
END AS ConfrontoConMedia

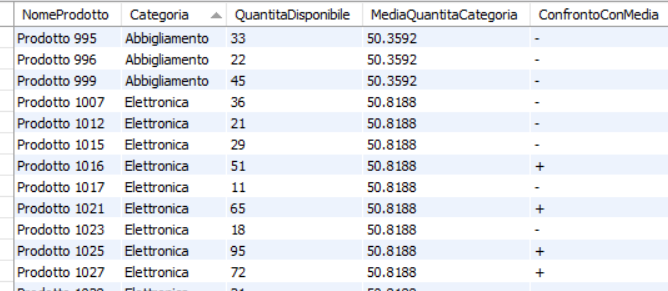
FROM

`2.0schema`.prodotti

ORDER BY

Categoria, NomeProdotto;





SELECT

COUNT(\*) AS NumeroTotaleProdotti,

SUM(CASE WHEN QuantitaDisponibile < COALESCE((SELECT AVG(QuantitaDisponibile) FROM `2.0schema`.prodotti), 0) THEN 1 ELSE 0 END) AS NumeroProdottiInferioriMedia

FROM

`2.0schema`.prodotti;

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

1. **Analisi dei Prodotti per Cliente:**
   * *Domanda:* Per ogni cliente, elenca i prodotti acquistati e il totale speso.

Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente

SELECT a.ClienteID, a.ProdottoID,

ROUND(SUM(a.QuantitaAcquistata\*b.Prezzo),2) AS Spesa\_acquisti,

a.QuantitaAcquistata

FROM `2.0schema`.transazioni a

JOIN `2.0schema`.prodotti b

ON a.ProdottoID = b.ProdottoID

GROUP BY a.ClienteID, a.ProdottoID

ORDER BY a.ClienteID ASC;

1. **Miglior Mese per le Vendite:**
   * *Domanda:* Identifica il mese con il maggior importo totale delle vendite.

SELECT

YEAR(t.DataTransazione) AS Anno,

MONTH(t.DataTransazione) AS Mese,

ROUND(SUM(p.Prezzo \* t.QuantitaAcquistata),0) AS Importo\_VenditeTotali

FROM

`2.0schema`.prodotti p

JOIN

`2.0schema`.transazioni t ON p.ProdottoID = t.ProdottoID

GROUP BY

Anno, Mese

ORDER BY

Importo\_VenditeTotali DESC

LIMIT 1;

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

1. **Analisi dei Prodotti in Magazzino:**
   * *Domanda:* Trova la quantità totale di prodotti disponibili in magazzino.

SELECT

SUM(QuantitaDisponibile) AS QuantitaTotaleInMagazzino

FROM

`2.0schema`.prodotti;

Immagine che contiene testo, Carattere, schermata, design

Descrizione generata automaticamente

1. **Clienti Senza Acquisti:**
   * *Domanda:* Identifica i clienti che non hanno effettuato alcun acquisto.

SELECT

MAX(NumeroClientiTransazioni) AS NumeroClientiTransazioni,

MAX(NumeroClientiClienti) AS NumeroClientiClienti,

MAX(ClientiSoloInTransazioni) AS ClientiSoloInTransazioni,

MAX(ClientiSoloInClienti) AS ClientiSoloInClienti

FROM (

SELECT

COUNT(DISTINCT t.ClienteID) AS NumeroClientiTransazioni,

0 AS NumeroClientiClienti,

COUNT(DISTINCT CASE WHEN c.ClienteID IS NULL THEN t.ClienteID END) AS ClientiSoloInTransazioni,

0 AS ClientiSoloInClienti

FROM

`2.0schema`.transazioni t

LEFT JOIN

`2.0schema`.clienti c ON t.ClienteID = c.ClienteID

UNION ALL

SELECT

0 AS NumeroClientiTransazioni,

COUNT(DISTINCT c.ClienteID) AS NumeroClientiClienti,

0 AS ClientiSoloInTransazioni,

COUNT(DISTINCT CASE WHEN t.ClienteID IS NULL THEN c.ClienteID END) AS ClientiSoloInClienti

FROM

`2.0schema`.transazioni t

RIGHT JOIN

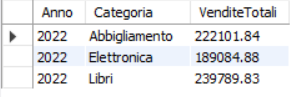
`2.0schema`.clienti c ON t.ClienteID = c.ClienteID

) AS Result;



* NumeroClientiTransazioni restituisce il numero di clienti distinti presenti solo nella tabella transazioni.
* NumeroClientiClienti restituisce il numero di clienti distinti presenti solo nella tabella clienti.
* ClientiSoloInClienti restituisce il numero di clienti distinti presenti solo nella tabella clienti.
* ClientiSoloInTransazioni restituisce il numero di clienti distinti presenti solo nella tabella transazioni.

1. **Analisi Annuale delle Vendite:**
   * *Domanda:* Calcola il totale delle vendite per ogni anno.



SELECT

YEAR(t.DataTransazione) AS Anno,

p.Categoria,

ROUND(SUM(p.Prezzo \* t.QuantitaAcquistata), 2) AS VenditeTotali

FROM

`2.0schema`.prodotti p

JOIN

`2.0schema`.transazioni t ON p.ProdottoID = t.ProdottoID

GROUP BY

Anno, p.Categoria

UNION ALL

SELECT

YEAR(t.DataTransazione) AS Anno,

'Totale' AS Categoria,

ROUND(SUM(p.Prezzo \* t.QuantitaAcquistata), 2) AS VenditeTotali

FROM

`2.0schema`.prodotti p

JOIN

`2.0schema`.transazioni t ON p.ProdottoID = t.ProdottoID

GROUP BY

Anno;

1. **Spedizioni in Ritardo:**
   * *Domanda:* Trova la percentuale di spedizioni con "In Consegna" rispetto al totale.

SELECT

Round((COUNT(StatusConsegna)/5000)\*100 , 0) AS percentuale\_ritardo

FROM `2.0schema`.spedizioni

WHERE StatusConsegna = 'In Consegna';

Immagine che contiene testo, Carattere, schermata, linea

Descrizione generata automaticamente

SELECT \*

FROM `2.0schema`.spedizioni

WHERE StatusConsegna = 'In Consegna';

SELECT

\*,

CASE

WHEN DATEDIFF(CURDATE(), DataSpedizione) > 60 THEN 'In Ritardo'

ELSE 'Puntuale'

END AS StatoSpedizione

FROM

`2.0schema`.spedizioni

WHERE

StatusConsegna = 'In Consegna';

Immagine che contiene testo, schermata, numero, Carattere

Descrizione generata automaticamente

Query per la sfida

SET SQL\_SAFE\_UPDATES = 0;

UPDATE 2.0schema.transazioni t

JOIN (

SELECT

p.ProdottoID,

p.NomeProdotto,

p.QuantitaDisponibile - COALESCE(SUM(t.QuantitaAcquistata), 0) AS DifferenzaQuantita

FROM

2.0schema.prodotti p

LEFT JOIN

2.0schema.transazioni t ON p.ProdottoID = t.ProdottoID

GROUP BY

p.ProdottoID, p.NomeProdotto, p.QuantitaDisponibile

HAVING

DifferenzaQuantita < 0

) AS ProdottiConDifferenzaNegativa ON t.ProdottoID = ProdottiConDifferenzaNegativa.ProdottoID

SET t.StatusConsegnaAggiornato = 'ANNULLATO'

WHERE t.TransazioneID IS NOT NULL AND t.TransazioneID > 0;

SET SQL\_SAFE\_UPDATES = 1;