1.

SELECT

MONTH(DataTransazione) AS Mese,

NomeProdotto,

SUM(QuantitaAcquistata) AS QuantitaTotale

FROM

transazioni

JOIN prodotti ON transazioni.ProdottoID = prodotti.ProdottoID

GROUP BY

Mese, NomeProdotto

ORDER BY

Mese, QuantitaTotale DESC;

2.

WITH PrezzoMedioPerClienteCategoria AS (

SELECT

t.ClienteID,

p.Categoria,

AVG(p.Prezzo \* t.QuantitaAcquistata) AS PrezzoMedioAcquisto,

ROW\_NUMBER() OVER (PARTITION BY p.Categoria ORDER BY AVG(p.Prezzo \* t.QuantitaAcquistata) DESC) AS RowNumPerCategoria

FROM

transazioni t

JOIN

prodotti p ON t.ProdottoID = p.ProdottoID

GROUP BY

t.ClienteID, p.Categoria

)

SELECT

ClienteID,

Categoria,

PrezzoMedioAcquisto

FROM

PrezzoMedioPerClienteCategoria

WHERE

RowNumPerCategoria <= 3

ORDER BY

Categoria, PrezzoMedioAcquisto DESC;

3.

WITH QuantitaMedia AS (

SELECT

AVG(QuantitaDisponibile) AS MediaQuantita

FROM

prodotti

)

SELECT

COUNT(\*) AS NumeroProdottiConQuantitaInferioreMedia

FROM

prodotti

JOIN

QuantitaMedia qm ON 1=1 -- Unisce con la CTE QuantitaMedia, che ha una sola riga

WHERE

QuantitaDisponibile < qm.MediaQuantita;

4.

-- SELECT \* FROM buildweek.ratings\_dataset;

SELECT

AVG(rd.Rating) AS MediaRecensioni

FROM

ratings\_dataset rd

JOIN transazioni t ON rd.CustomerID = t.ClienteID AND rd.ProductID = t.ProdottoID

WHERE

DATEDIFF(t.DataSpedizione, t.DataTransazione) < 30;

5.

SELECT

TransazioneID,

DataTransazione,

DataSpedizione,

CASE

WHEN DATEDIFF(DataSpedizione, DataTransazione) > 365 THEN 'Più di un anno'

ELSE 'Meno di un anno'

END AS TempoSpedizione

FROM Transazioni

WHERE DataTransazione IS NOT NULL AND DataSpedizione IS NOT NULL;

6.

SELECT

Categoria,

SUM(QuantitaDisponibile) AS TotaleDisponibilita

FROM Prodotti

GROUP BY Categoria;

7.

8.

SELECT

c.NomeCliente,

c.Email,

c.DataRegistrazione

FROM

Clienti c

JOIN

Transazioni t ON c.ClienteID = t.ClienteID

WHERE

MONTH(c.DataRegistrazione) =

(SELECT MONTH(DataTransazione)

FROM Transazioni

GROUP BY MONTH(DataTransazione)

ORDER BY SUM(ImportoTransazione) DESC

LIMIT 1)

ORDER BY

c.DataRegistrazione;

9.

WITH ValutazioniProdotti AS (

SELECT

p.ProdottoID,

AVG(r.Rating) AS RatingMedio

FROM prodotti p

JOIN ratings\_dataset r ON p.ProdottoID = r.ProdottoID

GROUP BY p.ProdottoID

)

SELECT

vp.ProdottoID,

p.NomeProdotto,

vp.RatingMedio,

c.NomeCliente AS ClienteConRatingPiùAlto

FROM ValutazioniProdotti vp

JOIN ratings\_dataset r ON vp.ProdottoID = r.ProdottoID AND vp.RatingMedio = r.Rating

JOIN clienti\_dataset c ON r.ClienteID = c.ClienteID

JOIN prodotti p ON vp.ProdottoID = p.ProdottoID;

10.

11.

SELECT

p.Categoria,

SUM(CASE WHEN t.DataTransazione BETWEEN '2022-03-21' AND '2022-09-22' THEN t.QuantitaAcquistata ELSE 0 END) AS QuantitaPrimaveraEstate,

SUM(CASE WHEN t.DataTransazione BETWEEN '2022-09-23' AND '2023-03-20' THEN t.QuantitaAcquistata ELSE 0 END) AS QuantitaAutunnoInverno,

SUM(CASE WHEN t.DataTransazione BETWEEN '2022-03-21' AND '2022-09-22' THEN p.Prezzo \* t.QuantitaAcquistata ELSE 0 END) AS TotaleSpesoPrimaveraEstate,

SUM(CASE WHEN t.DataTransazione BETWEEN '2022-09-23' AND '2023-03-20' THEN p.Prezzo \* t.QuantitaAcquistata ELSE 0 END) AS TotaleSpesoAutunnoInverno

FROM

Transazioni t

JOIN

Prodotti p ON t.ProdottoID = p.ProdottoID

WHERE

p.Categoria = 'Abbigliamento'

GROUP BY

p.Categoria;

12.

select p.nomeprodotto,

t.datatransazione,

t.dataspedizione,

datediff(t.dataspedizione, t.datatransazione) as giornitrascorsi

from transazioni t

join prodotti p on t.prodottoID = p.prodottoID

order by

t.datatransazione desc

limit 10;

13.

WITH VenditeProdotti AS (

SELECT

t.ProdottoID,

p.NomeProdotto,

SUM(t.QuantitaAcquistata) AS QuantitaVenduta

FROM

transazioni t

JOIN

prodotti p ON t.ProdottoID = p.ProdottoID

GROUP BY

t.ProdottoID, p.NomeProdotto

HAVING

SUM(t.QuantitaAcquistata) >= 9

)

SELECT

vp.NomeProdotto,

vp.QuantitaVenduta,

AVG(rd.Valutazione) AS MediaRecensioni

FROM

VenditeProdotti vp

JOIN

ratings\_dataset rd ON vp.ProdottoID = rd.ProdottoID

GROUP BY

vp.ProdottoID, vp.NomeProdotto, vp.QuantitaVenduta

ORDER BY

QuantitaVenduta DESC, MediaRecensioni DESC;

14.

SELECT

CASE DAYOFWEEK(DataTransazione)

WHEN 1 THEN 'Domenica'

WHEN 2 THEN 'Lunedì'

WHEN 3 THEN 'Martedì'

WHEN 4 THEN 'Mercoledì'

WHEN 5 THEN 'Giovedì'

WHEN 6 THEN 'Venerdì'

WHEN 7 THEN 'Sabato'

END AS NomeGiorno,

COUNT(\*) AS NumeroTransazioni,

(COUNT(\*) / (SELECT COUNT(\*) FROM Transazioni)) \* 100 AS PercentualeTransazioni

FROM Transazioni

GROUP BY NomeGiorno

ORDER BY MIN(DAYOFWEEK(DataTransazione));

15.

select

c.clienteid,

c.nomecliente

from clienti c

join transazioni t on c.clienteid = t.clienteid

join prodotti p on t.prodottoID = p.prodottoID

where

t.importotransazione > 3

group by

c.clienteid, c.nomecliente

having

count(distinct p.categoria) >= 2;

16.

SELECT

p.ProdottoID,

p.NomeProdotto,

p.QuantitaDisponibile,

c.NomeCliente,

c.Email

FROM

Prodotti p

JOIN

Transazioni t ON p.ProdottoID = t.ProdottoID

JOIN

Clienti c ON t.ClienteID = c.ClienteID

WHERE

p.QuantitaDisponibile = (SELECT MIN(QuantitaDisponibile) FROM Prodotti)

ORDER BY

p.ProdottoID;

17.

-- SELECT \* FROM buildweek.prodotti; domanda 17

SELECT

p.NomeProdotto,

p.Categoria,

SUM(t.QuantitaAcquistata) AS QuantitaTotale

FROM

transazioni t

JOIN prodotti p ON t.ProdottoID = p.ProdottoID

GROUP BY

p.NomeProdotto, p.Categoria

ORDER BY

QuantitaTotale DESC

LIMIT 3;

18.

SELECT

p.NomeProdotto,

p.Categoria,

SUM(t.ImportoTransazione) AS SpesaTotale

FROM

transazioni t

JOIN prodotti p ON t.ProdottoID = p.ProdottoID

GROUP BY

p.NomeProdotto, p.Categoria;