**Build Week 1 - SFIDA Esercizi SQL**

*Gruppo Fantasy*

**#1. Quali prodotti vendono meglio in determinati periodi dell’anno?**

SELECT Tab.Mese, MAX(Tab.Spesa), Tab.Nome

FROM (SELECT MONTH(t.DataTransazione) AS Mese, p.NomeProdotto as Nome, SUM(t.ImportoTransazione) as Spesa

FROM prodotti p

INNER JOIN transazioni t ON p.ProdottoID=t.ProdottoID

GROUP BY Mese, Nome

ORDER BY Mese ASC, Spesa DESC) AS Tab

GROUP BY Tab.Mese

ORDER BY Tab.Mese ASC;

**#2. Selezione i primi 3 clienti che hanno il prezzo medio di acquisto più alto in ogni categoria di prodotto.**

SELECT spesaclienti.Categoria as Categoria, spesaclienti.Cliente as Nome, CAST(MAX(spesaclienti.Spesa) AS DECIMAL(10,2)) AS Spesa

FROM (SELECT p.Categoria as Categoria,c.NomeCliente AS Cliente, AVG(t.ImportoTransazione) AS Spesa

FROM clienti c

INNER JOIN transazioni t ON t.ClienteID=c.ClienteID

INNER JOIN prodotti p ON p.ProdottoID=t.ProdottoID

GROUP BY Categoria) AS spesaclienti

GROUP BY Categoria

LIMIT 3;

**#3. Numero di prodotti con una quantità disponibile inferiore alla media.**

SELECT COUNT(\*)

FROM prodotti

WHERE QuantitaDisponibile<(SELECT AVG(QuantitaDisponibile) FROM prodotti);

**#4. Media delle recensioni dei clienti il cui tempo di elaborazione dell'ordine è inferiore a 30gg**

SELECT CAST(AVG(r.Rating) AS DECIMAL (10,2)) AS MediaRecensioni

FROM recensioni r

INNER JOIN clienti c ON r.CustomerID=c.ClienteID

INNER JOIN transazioni t ON t.ClienteID=c.ClienteID

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

**#5. Valutazione del tempo in anni in cui viene gestita una spedizione con visualizzazione di "Più di un anno" o "Meno di un anno" in una colonna calcolata.**

SELECT ProdottoID, DataTransazione, DataSpedizione,

CASE

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

ELSE 'Meno di un anno'

END AS Tempo\_Elaborazione

FROM

transazioni;

**#6. Totale delle disponibilità in magazzino dei prodotti divisi per categoria**

*Query utilizzata:*

*SELECT*

*Categoria,*

*SUM(QuantitaDisponibile) AS TotaleDisponibilita*

*FROM*

*data\_analyst.prodotti\_dataset*

*GROUP BY*

*Categoria;*

**#7. Totale delle disponibilità in magazzino dei prodotti divisi per categoria**

*Query utilizzata:*

*SELECT*

*c.NomeCliente,*

*t.ImportoTransazione,*

*p.NomeProdotto,*

*r.Rating,*

*CASE*

*WHEN t.ImportoTransazione > m.MediaTransazioni THEN 'Sopra La Media'*

*ELSE 'Sotto la Media'*

*END AS OUTPUT*

*FROM*

*data\_analyst.`transazioni new` t*

*JOIN*

*data\_analyst.clienti\_dataset c ON t.ClienteID = c.ClienteID*

*JOIN*

*data\_analyst.prodotti\_dataset p ON t.ProdottoID = p.ProdottoID*

*LEFT JOIN*

*data\_analyst.ratings\_dataset r ON t.ProdottoID = r.ProductID AND t.ClienteID = r.CustomerID*

*JOIN*

*MediaTransazioniAnno m ON YEAR(t.DataTransazione) = m.AnnoTransazione;*

**#8. Trovare tutti i clienti che si sono registrati nel mese con più profitto**

Query utilizzata:

WITH MeseMaggioreProfitto AS (

SELECT

YEAR(t.DataTransazione) AS AnnoTransazione,

MONTH(t.DataTransazione) AS MeseTransazione,

SUM(t.ImportoTransazione \* t.QuantitaAcquistata) AS ProfittoMese

FROM

data\_analyst.`transazioni new`t

GROUP BY

YEAR(t.DataTransazione), MONTH(t.DataTransazione)

ORDER BY

ProfittoMese DESC

LIMIT 1

)

SELECT

c.NomeCliente,

c.DataRegistrazione

FROM

data\_analyst.clienti\_dataset c

WHERE

YEAR(c.DataRegistrazione) = (SELECT AnnoTransazione FROM MeseMaggioreProfitto)

AND MONTH(c.DataRegistrazione) = (SELECT MeseTransazione FROM MeseMaggioreProfitto);

**#9. Trovare i prodotti con il rating medio più alto, insieme al nome del cliente che ha dato il rating più alto a ciascun prodotto.**

SELECT recens.prodotto, c.NomeCliente AS cliente, recens.recensione AS recensione

FROM (SELECT p.ProdottoID as prodotto, AVG(r.Rating) as recensione FROM prodotti p

INNER JOIN recensioni r ON r.ProductID=p.ProdottoID

GROUP BY prodotto

ORDER BY recensione DESC) AS recens

INNER JOIN clienti c ON recens.prodotto=c.ClienteID

ORDER BY recensione DESC

LIMIT 3;

**/\*10. Trova le informazioni dettagliate sulla spedizione,**

**inclusi il cliente, il prodotto, la quantità acquistata e l'importo totale per ciascuna spedizione.**

**Modifica lo stato degli ordini in base a determinate condizioni,**

**ad esempio, se un prodotto è esaurito, imposta lo stato dell'ordine su "Annullato". \*/**

*-- Dettagli sulla spedizione*

SELECT

c.NomeCliente,

pd.NomeProdotto,

td.QuantitaAcquistata,

td.ImportoTotaleTransazione,

s.MetodoSpedizione,

s.DataSpedizione,

s.StatusConsegna

FROM

build\_week\_1\_lucasilvi.transazioni\_dataset td

JOIN

build\_week\_1\_lucasilvi.clienti\_dataset c ON td.ClienteID = c.ClienteID

JOIN

build\_week\_1\_lucasilvi.prodotti\_dataset pd ON td.ProdottoID = pd.ProdottoID

JOIN

build\_week\_1\_lucasilvi.spedizioni\_dataset s ON td.IDSpedizione = s.IDTransazione;

-- Modifica dello stato degli ordini

UPDATE

build\_week\_1\_lucasilvi.spedizioni\_dataset s

JOIN

build\_week\_1\_lucasilvi.transazioni\_dataset td ON s.IDTransazione = td.IDSpedizione

JOIN

build\_week\_1\_lucasilvi.prodotti\_dataset pd ON td.ProdottoID = pd.ProdottoID

SET

s.StatusConsegna = 'Annullato'

WHERE

pd.QuantitaDisponibile = 0;

**/\*11. Considerando i prodotti della Categoria Abbigliamento nel periodo Primavera-Estate**

**e Autunno-Inverno, trova la quantità totale di prodotti acquistati e totale speso,**

**arrotondando laddove necessario (Suggerimento/N.B: il totale speso, non il totale transazione)\*/**

SELECT

SUM(td.QuantitaAcquistata) AS QuantitaTotaleAutunnoInverno,

ROUND(SUM(CAST(pd.Prezzo AS DECIMAL(10,2)) \* td.QuantitaAcquistata), 2) AS TotaleSpeso

FROM

build\_week\_1\_lucasilvi.transazioni\_dataset td

JOIN

build\_week\_1\_lucasilvi.prodotti\_dataset pd ON td.ProdottoID = pd.ProdottoID

WHERE

pd.Categoria = 'Abbigliamento'

AND

((MONTH(td.DataTransazione) BETWEEN 10 AND 12) + (MONTH(td.DataTransazione) BETWEEN 1 AND 3));

SELECT

SUM(td.QuantitaAcquistata) AS QuantitaTotalePrimaveraEstate,

ROUND(SUM(CAST(pd.Prezzo AS DECIMAL(10,2)) \* td.QuantitaAcquistata), 2) AS TotaleSpeso

FROM

build\_week\_1\_lucasilvi.transazioni\_dataset td

JOIN

build\_week\_1\_lucasilvi.prodotti\_dataset pd ON td.ProdottoID = pd.ProdottoID

WHERE

pd.Categoria = 'Abbigliamento'

AND

(MONTH(td.DataTransazione) BETWEEN 4 AND 9);

**/\*12. Trova il nome del prodotto, la data di transazione e la data di spedizione degli ultimi dieci prodotti acquistati.**

**Elenca anche quanti giorni siano trascorsi dalla data di transazione a quella di spedizione\*/**

SELECT

pd.NomeProdotto,

td.DataTransazione,

td.DataSpedizione,

DATEDIFF(DATE(td.DataSpedizione), DATE(td.DataTransazione)) AS GiorniTrascorsi

FROM

build\_week\_1\_lucasilvi.transazioni\_dataset td

JOIN

build\_week\_1\_lucasilvi.prodotti\_dataset pd ON td.ProdottoID = pd.ProdottoID

ORDER BY

td.DataTransazione DESC

LIMIT

10;

**/\*13. Identifica i prodotti che hanno venduto più di nove unità,**

**e calcola la media delle recensioni per ciascun prodotto,**

**ordinando il risultato in modo decrescente in base al numero di unità vendute e alla media delle recensioni.\*/**

SELECT

pd.ProdottoID,

pd.NomeProdotto,

SUM(td.QuantitaAcquistata) AS UnitàVendute,

AVG(r.Rating) AS MediaRecensioni

FROM

build\_week\_1\_lucasilvi.transazioni\_dataset td

JOIN

build\_week\_1\_lucasilvi.prodotti\_dataset pd ON td.ProdottoID = pd.ProdottoID

JOIN

build\_week\_1\_lucasilvi.ratings\_dataset r ON pd.ProdottoID = r.ProductID

GROUP BY

pd.ProdottoID, pd.NomeProdotto

HAVING

UnitàVendute > 9

ORDER BY

UnitàVendute DESC, MediaRecensioni DESC;

**/\*14. Mostra come sono distribuite le transazioni sui giorni della settimana.**

**Mostra sia il numero di transazioni per ogni giorno della settimana,**

**che la percentuale divisa per giorno della settimana\*/**

SELECT

DAYNAME(td.DataTransazione) AS GiornoSettimana,

COUNT(\*) AS NumeroTransazioni,

ROUND((COUNT(\*) / (SELECT COUNT(\*) FROM build\_week\_1\_lucasilvi.transazioni\_dataset td)) \* 100, 2) AS Percentuale

FROM

build\_week\_1\_lucasilvi.transazioni\_dataset td

GROUP BY

GiornoSettimana, DAYOFWEEK(td.DataTransazione)

ORDER BY

DAYOFWEEK(td.DataTransazione);

**/\*15: Trova tutti i clienti che hanno effettuato transazioni**

**con importo totale superiore a una certa soglia e**

**che hanno speso in almeno due categorie di prodotti diverse.\*/**

SELECT DISTINCT c.ClienteID

FROM clienti\_dataset c

JOIN transazioni\_dataset t ON c.ClienteID = t.ClienteID

JOIN prodotti\_dataset p ON t.ProdottoID = p.ProdottoID

WHERE t.ImportoTotaleTransazione > 30

GROUP BY c.ClienteID

HAVING COUNT(DISTINCT p.Categoria) >= 2;

**/\*16: Trova i prodotti con la quantità disponibile più bassa e**

**il relativo cliente che li ha acquistati.\*/**

SELECT p.\*, c.NomeCliente

FROM prodotti\_dataset p

JOIN transazioni\_dataset t ON p.ProdottoID = t.ProdottoID

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

WHERE p.QuantitaDisponibile = (

SELECT MIN(QuantitaDisponibile)

FROM prodotti\_dataset

);

**/\*17. Identifica i tre prodotti più venduti e la loro quantità venduta.**

**Trova la spesa totale (Prezzo\*QuantitaAcquistata) per ogni prodotto e identifica la sua categoria\*/**

SELECT

pd.ProdottoID,

pd.NomeProdotto,

pd.Categoria,

SUM(td.QuantitaAcquistata) AS QuantitaVenduta,

SUM(CAST(pd.Prezzo AS DECIMAL(10,2)) \* td.QuantitaAcquistata) AS SpesaTotale

FROM

build\_week\_1\_lucasilvi.transazioni\_dataset td

JOIN

build\_week\_1\_lucasilvi.prodotti\_dataset pd ON td.ProdottoID = pd.ProdottoID

GROUP BY

pd.ProdottoID, pd.NomeProdotto, pd.Categoria

ORDER BY

QuantitaVenduta DESC

LIMIT 3;

**/\*18. Trova la quantità totale e il prezzo medio di prodotti**

**(arrotonda a tre cifre decimali) disponibili in magazzino per ogni prodotto.**

**Identifica la sua categoria.\*/**

SELECT

pd.ProdottoID,

pd.NomeProdotto,

pd.Categoria,

SUM(pd.QuantitaDisponibile) AS QuantitaTotale,

ROUND(AVG(CAST(pd.Prezzo AS DECIMAL(10, 3))), 3) AS PrezzoMedio

FROM

build\_week\_1\_lucasilvi.prodotti\_dataset pd

GROUP BY

pd.ProdottoID, pd.NomeProdotto, pd.Categoria

ORDER BY

pd.ProdottoID;