

Rapporto Finale el equipo Fantasy

PREMESSA:

Il seguente report, che è stato realizzato dopo un confronto di gruppo e con il riscontro di diversi punti critici e punti salienti, presenta il risultato che abbiamo ritenuto più ottimale. Partendo da un DATASET che ci è stato fornito dal docente Rizzo in formato csv, abbiamo importato inizialmente i dati su MySQL Workbench, notoriamente utile per la gestione di database e relazioni tra tabelle. In questo modo è stato possibile analizzare i dati distribuiti su più tabelle, relativi nello specifico a un'azienda di E-Commerce, ottenendo i risultati che seguono: Analisi sulle vendite, sulle recensioni del sito, sui clienti etc.

In un secondo momento, abbiamo analizzato alcune delle query scritta prima in linguaggio SQL, elaborando le analisi su EXCEL, principalmente con l'utilizzo delle tabelle pivot e la creazione di grafici intuitivi, che mostrassero direttamente a video il risultato.

LEGENDA:

GROUP BY= Raggruppa le righe del DATASET in base a delle colonne che decidiamo di usare come parametro discriminante. Quando si utilizza GROUP BY, è necessario includere nell'elenco SELECT tutte le colonne che non sono oggetto di funzioni di aggregazione (come SUM, COUNT, AVG, etc.).

ORDER BY= Funzione SQL tramite la quale viene praticato un ordinamento o ascendente o decrescente, rispetto a uno o più parametri (columns) da noi stabiliti.

LAG= Funzione che restituisce il valore di una colonna in una riga specificata rispetto alla riga corrente in un risultato di query, all'interno di un determinato ordine di ordinamento.

CAST + AS + tipo di dato in cui convertire= Converte un dato, dal formato attuale al formato desiderato.

CONCAT= concatena gli argomenti specificati all'interno della funzione, utilizzando come separatore la virgola e dando la possibilità di inserire all'interno: risultati di una query o stringhe/caratteri da noi direttamente inseriti (in quest'ultimo caso, tramite apici).

Build Week 1 - Esercizi SQL

RESOCONTO e SPIEGAZIONE:

1. Analisi delle Vendite Mensile

→ *Domanda:* Trova il totale delle vendite per ogni mese.

Query utilizzata:

```
SELECT YEAR(DataTransazione) AS Anno, MONTH(DataTransazione) AS Num, MONTHNAME(DataTransazione) AS Mese,
       CONCAT(CAST(SUM(ImportoTransazione) AS DECIMAL(10,2)), ' EURO') AS Totale
FROM transazioni
GROUP BY Anno, Num, Mese
ORDER BY Anno, Num;
```

Risultato:

Risultato Anno	Num	Mese	Totale
2022	1	January	2.517,32
2022	2	February	2.083,31
2022	3	March	3.013,55
2022	4	April	3.096,66
2022	5	May	2.583,50
2022	6	June	2.191,73
2022	7	July	1.556,35
2022	8	August	2.460,69
2022	9	September	2.127,31
2022	10	October	1.852,54
2022	11	November	2.591,98
2022	12	December	2.355,87

Spiegazione:

Essendo la mole di dati ignota, al momento dell'esecuzione della prima query, abbiamo deciso di discriminare le righe del dataset, non solo per mese, ma anche per anno (potenzialmente, i dati avrebbero potuto essere distribuiti su più anni). Non solo, nel caso in cui il nostro dataset venga aggiornato con l'inserimento di altre righe, nella fattispecie con valore di anno diverso da "2022", la query appena descritta sarebbe sempre valida. Nel caso in cui discriminiamo solo per mese, è valida se abbiamo un unico valore di Anno.

La nostra richiesta, che era di calcolare il totale delle vendite per mese, lo si ottiene semplicemente raggruppando per anno e per mese (**GROUP BY**), che cronologicamente, viene eseguita dal programma prima della (**SELECT**), le righe della tabella transazioni. Una volta raggruppate, la funzione SELECT SUM degli importi singoli delle transazioni ci restituirà il totale delle vendite di ciascun mese.

2. Prodotti più Venduti:

→ *Domanda:* Identifica i tre prodotti più venduti e la loro quantità venduta

Query utilizzata:

```
SELECT p.ProdottoID AS Codice, p.NomeProdotto AS Prodotto, SUM(t.QuantitaAcquistata) AS Quantita
FROM prodotti p
JOIN transazioni t ON t.ProdottoID= p.ProdottoID
GROUP BY Codice
ORDER BY Quantita DESC
LIMIT 3;
```

Risultato:

Codice	Prodotto	Quantità
3577	Prodotto 3477	17
856	Prodotto 756	16
4023	Prodotto 3923	13

Spiegazione:

Come struttura della query tutti i membri del gruppo abbiamo concordato, mentre per quanto riguarda la determinazione del concetto di “vendita”, sono sovvenuti due pareri differenti: il primo considerava la vendita come transazione con consegna riuscita, mentre il secondo considerava la vendita in quanto avvenuto pagamento (indipendentemente dalla consegna o meno). Anche qui è stato fatto uso del **GROUP BY**, per distinguere i differenti prodotti. Abbiamo associato al **GROUP BY**, un **ORDER BY** DESC (decrescente) per ottenere poi come risultato, tramite la somma delle quantità acquistate, i prodotti più venduti. Dato che ci interessano nello specifico i tre più venduti, usiamo la funzione LIMIT 3 (che restituisce le prime 3 righe), dato che la funzione TOP è appannaggio esclusivamente di SQL server.

3. Analisi Cliente:

→ *Domanda:* Trova il cliente che ha effettuato il maggior numero di acquisti.

Query utilizzata:

```
SELECT c.ClienteID AS Codice, c.NomeCliente AS Nominativo, COUNT(*) AS Acquisti, c.DataRegistrazione AS reg
FROM clienti c
INNER JOIN transazioni t ON t.ClienteID= c.ClienteID
GROUP BY Codice
ORDER BY Acquisti DESC, reg
LIMIT 10;
```

Risultato:

Codice	Nominativo	Acquisti	DATA Registrazione
4063	Cliente 4063	3	2022-08-08
700	Cliente 700	3	2022-09-02
3350	Cliente 3350	2	2022-01-08
4055	Cliente 4055	2	2022-01-27
202	Cliente 202	2	2022-02-07
3751	Cliente 3751	2	2022-02-20
1473	Cliente 1473	2	2022-04-26

2664	Cliente 2664	2	2022-05-08
4213	Cliente 4213	2	2022-05-08
2105	Cliente 2105	2	2022-05-31

Spiegazione:

Dato che ci interessa sapere chi è il cliente che ha effettuato più acquisti, vogliamo conoscere anche eventuali specifiche (oltre l'ID, anche nome o eventualmente e-mail). Abbiamo così effettuato una **INNER JOIN** per associare i clienti alle relative transazioni da loro effettuate, quindi eseguendo il join sulla chiave cliente. Anche qui come nelle query precedenti l'utilizzo di GROUP BY, per dividere i vari clienti. Abbiamo poi ordinato con un ORDER BY decrescente sulla base degli acquisti effettuati (conteggio delle righe nella tabella transazioni rispetto al codice cliente) → **COUNT(*)**. Per non assegnare un ordinamento casuale stabilito dalla macchina, abbiamo ritenuto opportuno, nel caso in cui più clienti avessero effettuato lo stesso numero di acquisti, di dare priorità a quelli con maggiore anzianità di iscrizione.

4. Valore medio della transazione:

→ **Domanda: Calcola il valore medio di ogni transazione (per metodo spedizione)**

Query utilizzata:

#A

```
SELECT MetodoSpedizione AS Spedizione, CONCAT(CAST(AVG(ImportoTransazione) AS DECIMAL(10,2)), ' EURO') AS ImportoMedio
FROM transazioni
GROUP BY Spedizione
ORDER BY ImportoMedio DESC;
```

#B

```
SELECT AVG(ImportoTransazione) as ValorMedioTransazione
FROM transazioni;
```

Risultato:

Spedizione	ImportoMedio
Corriere Express	57.74 EURO
Posta Prioritaria	55.80 EURO

Spiegazione:

A una prima lettura, ci viene subito in mente il concetto di MEDIA applicata a tutti gli importi delle transazioni, ma, dato che una media, per grandi scostamenti risulta essere un'operazione poco significativa, abbiamo deciso di rapportarla al tipo di spedizione, generando un importo medio per ciascun tipo di spedizione. La media viene calcolata tramite una funzione chiamata **AVG()** che in questo caso, avendo discriminato prima per spedizione e ordinato poi per importo medio decrescente, restituirà prima la media di uno (maggiore) e poi dell'altro.

5. Analisi Categoria Prodotto:

→ **Domanda:** Determina la categoria di prodotto con il maggior numero di vendite

Query utilizzata:

```
SELECT p.Categoria AS Categoria, SUM(t.QuantitaAcquistata) AS Vendite
FROM prodotti p
INNER JOIN transazioni t ON t.ProdottoID= p.ProdottoID
GROUP BY Categoria
ORDER BY Vendite DESC
LIMIT 1;
```

Risultato:

Categoria	Vendite
Libri	865

Spiegazione:

La query n°5 richiedeva di determinare la categoria di prodotto con il maggior numero di vendite. Se per vendite intendiamo la quantità di prodotti venduti, abbiamo bisogno dei dati compresi nella colonna QuantitaAcquistata, presente nella tabella transazioni. Possiamo disporre dalla FROM della relativa tabella ed includere un collegamento alla tabella prodotti con un JOIN sulla colonna ProdottoID, così da stabilire infine qual'è la categoria alla quale appartiene il maggior numero di prodotti venduti.

6. Cliente Fedele:

→ **Domanda:** Identifica il cliente con il maggior valore totale di acquisti

Query utilizzata:

```
SELECT c.ClienteID AS Codice, c.NomeCliente AS Nome, SUM(t.ImportoTransazione) AS TotaleAcquisti
FROM clienti c
INNER JOIN transazioni t ON t.ClienteID= c.ClienteID
GROUP BY Codice
ORDER BY TotaleAcquisti DESC
LIMIT 1;
```

Risultato:

Codice	Nome	Totale Speso
2105	Cliente 2105	191.1

Spiegazione:

Come criterio per stabilire chi fosse il miglior cliente, abbiamo scelto l'importo speso.

7. Spedizioni Riuscite:

→ **Domanda:** Calcola la percentuale di spedizioni con "Consegna Riuscita"

Query utilizzata:

```
SELECT CONCAT(  
    CAST(  
        ((SELECT COUNT(*)  
          FROM transazioni  
          WHERE StatusConsegna='Consegna Riuscita')  
        /  
        COUNT(*) * 100  
        AS DECIMAL(10,2)),  
        '%') AS Percentuale_Consegne_Riuscite  
FROM transazioni;
```

Risultato:

Consegne RIUSCITE
48.80%

Spiegazione:

Essendo la query suddetta complementare della query n°19 rispetto alla colonna parametro Status Consegna, l'interpretazione del codice qui allegato sarà lo stesso presente nella 19, variando unicamente la condizionale gestita dalla WHERE. Anche in questo caso la FROM prende i valori presenti nella tabella transazioni, e tramite una subquery al numeratore con la condizione specificata nel WHERE, dove si richiedono i conteggi delle Consegne Riuscite, dividendo per i conteggi complessivi della tabella transazioni: si ottiene un numero decimale, il quale operando con la funzione CAST è stato limitato nel numero di cifre presente dopo la virgola a 2, moltiplicato per 100 e concatenato con il simbolo % al fine di visualizzare il conto in forma percentuale.

8. Prodotto con la Migliore Recensione:

→ **Domanda:** Trova il prodotto con la recensione media più alta

Query utilizzata:

```

SELECT p.NomeProdotto as NOME, CAST(AVG(r.Rating) AS DECIMAL(2,1)) AS Valutazione, COUNT(r.ProductID) AS Voti
FROM recensioni r
/*FROM transazioni t
INNER JOIN recensioni r ON t.ProdottoID=r.ProductID*/
INNER JOIN prodotti p ON p.ProdottoID=r.ProductID
GROUP BY p.ProdottoID
ORDER BY Valutazione DESC, Voti DESC
LIMIT 10;

```

Risultato:

NOME	Valutazione	Voti
Prodotto 1493	5.0	2
Prodotto 820	5.0	2
Prodotto 1241	5.0	2
Prodotto 3755	5.0	2
Prodotto 4322	5.0	2
Prodotto 459	5.0	2
Prodotto 2339	5.0	2
Prodotto 1520	5.0	1
Prodotto 615	5.0	1
Prodotto 3306	5.0	1

Spiegazione:

Inizialmente avevamo pensato di rendere la recensione una funzione che avesse per parametri, quelle che erano le colonne che più avrebbero potuto influenzare il concetto di recensione stessa. Ma nel momento in cui c'è stato fornito il dataset recensioni, abbiamo impostato il codice selezionando il nome del Prodotto, il numero di Voti assegnati a ciascuno con il conteggio del numero di comparse di ogni ID di Prodotto e la media del rating ovvero la media delle Valutazioni associate a ciascuna Votazione per ciascun Prodotto, alla quale abbiamo abbinato un **CAST** con un cifra decimale per mostrare la ragione dell'arrotondamento.

9. Analisi Temporale:

→ **Domanda:** Calcola la variazione percentuale nelle vendite rispetto al mese precedente

Query utilizzata:

```

SELECT
    DATE_FORMAT(DataTransazione, '%Y-%m') AS Mese,
    CAST(SUM(ImportoTransazione) AS DECIMAL(10,2)) AS Totale,
    CAST(LAG(SUM(ImportoTransazione)) OVER (ORDER BY DATE_FORMAT(DataTransazione, '%Y-%m')) AS DECIMAL(10,2)) AS Precedente,
    CONCAT(CAST(((SUM(ImportoTransazione)/LAG(SUM(ImportoTransazione)) OVER (ORDER BY DATE_FORMAT(DataTransazione, '%Y-%m')))*100-100) AS DECIMAL(7,2)), '%') AS
FROM
    transazioni
GROUP BY Mese
ORDER BY Mese;

```

Risultato:

Mese	Totale €	Precedente €	Incremento %
2022-01	2.517,32	NULL	NULL
2022-02	2.083,31	2.517,32	-17,24
2022-03	3.013,55	2.083,31	44,65
2022-04	3.097,06	3.013,55	2,76
2022-05	2.583,50	3.097,06	-16,57
2022-06	2.192,13	2.583,50	-15,16
2022-07	1.556,35	2.192,13	-28,99
2022-08	2.461,09	1.556,35	58,11
2022-09	2.127,31	2.461,09	-13,55
2022-10	1.852,54	2.127,31	-12,92
2022-11	2.592,38	1.852,54	39,91
2022-12	2.356,27	2.592,38	-9,11

Spiegazione:

Dato che veniva richiesto di calcolare la variazione percentuale delle vendite (su importo totale) di ogni mese, rispetto al mese precedente, si rende necessario, in primo luogo, il calcolo del totale del mese. Secondo, dobbiamo calcolare il totale del mese precedente. Per tal motivo utilizziamo la funzione **DATE_FORMAT** che estrae l'anno e il mese dalla colonna DataTransazione abbinandola all'utilizzo della funzione **LAG**, che restituisce il valore della somma degli importi totali, raggruppati per mese e anno, **MA** del valore di anno e mese precedenti. Una volta ottenuti i due valori, dividiamo il primo per il secondo, e moltiplicando per cento e sottraendo cento, otteniamo l'incremento o il decremento dell'importo rispetto al mese precedente.

10. Quantità di Prodotti Disponibili:

→ **Domanda:** Determina la quantità media disponibile per categoria di prodotto

Query utilizzata:


```
SELECT Categoria, CAST(AVG(QuantitaDisponibile)AS DECIMAL(10,2)) AS Disponibili
FROM prodotti
GROUP BY Categoria
ORDER BY Disponibili DESC;
```

Risultato:

Categoria	Disponibilità media (pezzi)
Elettronica	51,22
Libri	51,07
Abbigliamento	50,36

Spiegazione: Richiedendo di determinare la quantità media di prodotti disponibile per Categoria, agendo attraverso la sola tabella prodotti nella select, abbiamo dichiarato l'intenzione di visualizzare la disponibilità specificata come alias e basandosi su di un raggruppamento ed ordinamento attraverso la **GROUP BY** tramite la colonna Categoria ed ordinando con la **ORDER BY** per nuova colonna Disponibili, la quale è stata convertita in formato decimale attraverso l'operatore **CAST** con il fine sempre di visualizzare la ragione di un eventuale arrotondamento per eccesso o per difetto, abbiamo realizzato la media per ciascuna Categoria avvalendosi infine della funzione di aggregazione **AVG()**, ottenendo i suddetti risultati rappresentati in forma tabellare.

11. Analisi Spedizioni:

→ **Domanda:** Trova il metodo di spedizione più utilizzato

Query utilizzata:

```
SELECT CONCAT('Il metodo di spedizione piu' usato e' ',MetodoSpedizione) AS Risultato,
       CONCAT(CAST((COUNT(*)/(SELECT COUNT(*) FROM transazioni))*100 AS DECIMAL(10,2)), '%') AS Utilizzo
FROM transazioni
GROUP BY MetodoSpedizione
ORDER BY Utilizzo DESC
LIMIT 1;
```

Risultato:

Risultato	Utilizzo
Il metodo di spedizione piu' usato e' Corriere Express	54.80%

Spiegazione:

In questo caso, essendo presenti solo due MetodoSpedizione, utilizzando un **GROUP BY** rispetto a quest'ultimo, otteniamo due raggruppamenti. Successivamente, viene eseguito, con l'utilizzo di una select annidata in un'altra, il rapporto tra il conteggio delle transazioni rispetto al metodo di spedizione (numeratore) e nella select annidata, non essendo questa dipendente dai parametri della select esterna, ritorna il conteggio delle transazioni totali (denominatore). Otteniamo così il valore frazionario delle transazioni di ogni metodo di spedizione sul totale. L'**ORDER BY** successivamente le ordina in modo decrescente, e tramite il **LIMIT** ci restituisce il primo risultato, quindi il metodo più utilizzato.

12. Analisi dei Clienti:

→ **Domanda:** Calcola il numero medio di clienti registrati al mese

Query utilizzata:

```
SELECT DATE_FORMAT(DataRegistrazione, '%Y-%m') AS Mese, COUNT(*) AS Registrazioni
FROM clienti
WHERE YEAR(DataRegistrazione)= 2022
GROUP BY Mese
ORDER BY Mese DESC;
```

Risultato:

Mese	Registrazioni
2022-12	431
2022-11	381
2022-10	406
2022-09	435
2022-08	407
2022-07	452
2022-06	419
2022-05	415
2022-04	448
2022-03	438
2022-02	358
2022-01	393

Query utilizzata:

```

SELECT
    AVG(Registrazioni) as MediaRegistrazioniMensileClienti
FROM( SELECT DATE_FORMAT(DataRegistrazione, '%Y-%m') AS Mese, COUNT(*) AS Registrazioni
FROM clienti
WHERE YEAR(DataRegistrazione)= 2022
GROUP BY Mese
ORDER BY Mese DESC) AS N°RegistrazioniTotaliPerMese;

```

Risultato:

MEDIA REG	415,25
------------------	---------------

Spiegazione: Al fine di avere a video un risultato più globale e comprensibile, abbiamo optato per fare due query. La prima che restituisse i registrati per mese e la seconda che fornisse la media di registrati, calcolata sui conteggi dei registrati di ogni mese. Inoltre, avendo notato che il dataset riportava registrazione del solo anno 2022 (eccezion fatta per gennaio 2023), avendo generato un codice valido per una media mensile su un numero anche maggiore di 1 di anni, abbiamo deciso di inserire una condizione sul where che specificasse un anno in particolare, rispetto al quale operare la media mensile, specificando nel caso proprio l'anno '2022'.

13. Prodotti Rari:

→ **Domanda: Identifica i prodotti con una quantità disponibile inferiore alla media**

Query utilizzata:

```

SELECT NomeProdotto, QuantitaDisponibile, (SELECT CAST(AVG(QuantitaDisponibile) AS DECIMAL(10,2)) FROM prodotti) AS Media
FROM prodotti
WHERE QuantitaDisponibile < (SELECT AVG(QuantitaDisponibile) FROM prodotti)
ORDER BY QuantitaDisponibile
LIMIT 10000;

```

Risultato:

NomeProdotto	QuantitaDisponibile	Media
Prodotto 1	8	51,01
Prodotto 1001	4	51,01
Prodotto 1004	17	51,01
Prodotto 1005	4	51,01
Prodotto 1007	36	51,01

Prodotto 1009	16	51,01
Prodotto 101	37	51,01
Prodotto 1010	29	51,01
Prodotto 1011	22	51,01

Spiegazione: La query per il calcolo della media è stata impostata in modo didattico, dato che accanto ad ogni valore di disponibilità, viene fatto vedere il valore della media totale. Nel caso più pratico, avremmo semplicemente restituito le quantità disponibili inferiori alla media per prodotto. In tal caso, la media delle quantità disponibili generale, sarebbe stata solo calcolata nella clausola **WHERE**. I prodotti inferiori alla media, sono risultati essere **2484**, ma per motivi di spazio abbiamo restituito qui a video solo i primi 10.

14. Analisi dei Prodotti per Cliente:

→ **Domanda: Per ogni cliente, elenca i prodotti acquistati e il totale speso**

Query utilizzata:

```
SELECT c.NomeCliente AS Cliente, COUNT(p.NomeProdotto) AS Prodotti, CONCAT(CAST(SUM(t.ImportoTransazione) AS DECIMAL(10,2)), ' EURO') AS Spesa_TOT
FROM transazioni t
INNER JOIN clienti c ON t.ClienteID=c.ClienteID
INNER JOIN prodotti p ON t.ProdottoID=p.ProdottoID
GROUP BY Cliente
ORDER BY Prodotti DESC;
```

Risultato:

Cliente	Prodotti	Spesa_TOT
Cliente 3875	18	111,13
Cliente 2105	15	191,10
Cliente 1215	14	113,76
Cliente 3350	14	159,71
Cliente 4213	13	136,28
Cliente 3231	13	138,86
Cliente 4114	12	98,55
Cliente 3751	11	37,00
Cliente 4055	11	67,30
Cliente 4835	10	81,62

Spiegazione:

Qui abbiamo raggruppato per Cliente, in modo da vedere quanti prodotti ogni cliente avesse acquistato. Abbiamo calcolato il numero dei prodotti come conteggio di questi nella tabella transazioni, avendo fatto un **INNER JOIN** in base al codice di prodotto. Per quanto riguarda la spesa totale per cliente, abbiamo fatto una somma dei singoli Importo Transazione. La seconda **INNER JOIN** invece serve semplicemente per restituirci il nome del cliente, non presente invece nella tabella transazioni. I clienti che risulta abbiano fatto acquisti sono **475**, ma anche qui, per ragioni di spazio, abbiamo visualizzato solo i primi 10, ordinandoli in ordine decrescente per numero di prodotti acquistati.

15. Miglior Mese per le Vendite:

→ **Domanda: Identifica il mese con il maggior importo totale delle vendite**

Query utilizzata:

```
SELECT CONCAT('Il mese con importo totale delle vendite maggiore e'' il ',DATE_FORMAT(DataTransazione, '%Y-%m')) AS Risultato, CONCAT(CAST(SUM(ImportoTransazione)AS DECIMAL(10,2)), ' EURO') AS Totale
FROM transazioni
GROUP BY DATE_FORMAT(DataTransazione, '%Y-%m')
ORDER BY Totale DESC
LIMIT 1;
```

Risultato:

Risultato	Totale
Il mese con importo totale delle vendite maggiore e' il 2022-04	3097.06 EURO

Spiegazione:

Il codice è stato ingegnerizzato in funzione della LIMIT, in quanto restituisce così una tabella 1x2, nella quale è riportato tramite SELECT per primo il mese soggetto indicato da una stringa testuale associata con la CONCAT e per secondo l'importo totale ricavato come somma delle vendite, con SUM, e riportato in formato decimale con CAST, dopo aver disposto i dati della tabella transazioni attraverso GROUP BY e ORDER BY..

16. Analisi dei Prodotti in Magazzino:

→ **Domanda: Trova la quantità totale di prodotti disponibili in magazzino**

Query utilizzata:

```
SELECT Categoria, SUM(QuantitaDisponibile) AS Pezzi, (SELECT SUM(QuantitaDisponibile) FROM prodotti) AS TOTALE
FROM prodotti
GROUP BY Categoria;
```

Risultato:

Categoria	Pezzi	TOTALE
Abbigliamento	86215	253065
Elettronica	84410	253065
Libri	82440	253065

Spiegazione:

Qui si richiede il totale dei pezzi presenti nel magazzino dell'azienda e-commerce. Ma poiché il dato somma totale numero di prodotti distinti per codice prodotto, come ammontare, ci sembrava poco interessante preso da solo ed in senso generico, abbiamo voluto restituire anche l'ammontare di pezzi per categoria nella colonna a fianco.

17. Clienti Senza Acquisti:

→ **Domanda: Identifica i clienti che non hanno effettuato alcun acquisto**

Query utilizzate:

```
# 17. Clienti Senza Acquisti:
#* Domanda: Identifica i clienti che non hanno effettuato alcun acquisto.
#a.
SELECT COUNT(*) AS Inattivi
FROM clienti c
LEFT JOIN transazioni t ON c.ClienteID=t.ClienteID
WHERE t.ClienteID is NULL;

#b. --> mi faccio restituire anche la data di iscrizione, per capire quanto tempo è rimasto inattivo e magari sollecitarlo con delle mail.
SELECT c.ClienteID as Cod, c.NomeCliente AS Nominativo, c.Email AS Email, DATE_FORMAT(c.DataRegistrazione, '%Y-%m-%d') AS Iscrizione
FROM clienti c
LEFT JOIN transazioni t ON c.ClienteID=t.ClienteID
WHERE t.ClienteID is NULL
ORDER BY Iscrizione ASC
LIMIT 5000;
```

Risultati:

QUERY A)

INATTIVI	4525
-----------------	-------------

QUERY B)

Cod	Nominativo	Email	Iscrizione
1479	Cliente 1479	cliente1479@email.com	2022-01-02
1653	Cliente 1653	cliente1653@email.com	2022-01-02
1666	Cliente 1666	cliente1666@email.com	2022-01-02

1934	Cliente 1934	cliente1934@email.com	2022-01-02
2305	Cliente 2305	cliente2305@email.com	2022-01-02
2967	Cliente 2967	cliente2967@email.com	2022-01-02
2986	Cliente 2986	cliente2986@email.com	2022-01-02
4346	Cliente 4346	cliente4346@email.com	2022-01-02
4699	Cliente 4699	cliente4699@email.com	2022-01-02

Spiegazione:

Abbiamo suddiviso in due query il risultato dell'esercizio. Nella QUERY A abbiamo semplicemente estratto il numero totale degli inattivi. Utilizzando una **LEFT JOIN** su codice cliente, laddove questo campo fosse **NULL** e quindi non comparisse nella tabella transazioni.

Nella QUERY B volevamo il numero di clienti senza acquisti, perciò tutti coloro che non risultavano dalla tabella transazioni, al che abbiamo operato attraverso la **LEFT JOIN** della clienti con la transazioni specificando di visualizzare i clienti che risultavano solo nella tabella clienti. A differenza della QUERY A, abbiamo qui stampato in dettaglio i dati relativi agli inattivi, ordinandoli in modo crescente per data di registrazione, in modo da capire da quanto fossero inattivi.

18. Analisi Annuale delle Vendite:

→ **Domanda: Calcola il totale delle vendite per ogni anno**

Query utilizzata:

```
SELECT YEAR(DataTransazione) AS Anno, CAST(SUM(ImportoTransazione) AS DECIMAL(10,2)), ' EURO' AS TOTALE_ANNO
FROM transazioni
GROUP BY Anno
ORDER BY Anno DESC;
```

Risultato:

Anno	TOTALE_ANNO
2022	28430.81 EURO

Spiegazione:

Qui semplicemente veniva richiesto di restituire il totale delle vendite dell'anno, per cui è stato impostato un raggruppamento per anno e nel caso in cui ci siano vendite relative ad altri anni, avremmo ordinato a partire dall'anno più recente. Il risultato non è altro che la somma dei singoli importo transazione sulla tabella transazione fatto in base al valore di YEAR della data della transazione.

19. Spedizioni in Ritardo:

→ **Domanda:** Trova la percentuale di spedizioni con "In Consegna" rispetto al totale.

Query utilizzata:

```
SELECT CONCAT('Le spedizioni in ritardo sono il ',
              CAST(
                ((SELECT COUNT(*) FROM transazioni WHERE StatusConsegna='In Consegna')
                 /COUNT(*))*100
                AS DECIMAL(10,2))
              ,'%') AS Risultato
FROM transazioni
```

Risultato:

Risultato
Le spedizioni in ritardo sono il 51.20%

Spiegazione:

La spiegazione della seguente query è identica a quella della query numero 7. Pertanto rimandiamo a quella.

Build Week 1 - Esercizi EXCEL

Per il dettaglio degli esercizi in excel abbiamo realizzato gli esercizi anche su google sheet e riportiamo di seguito il link per la sua visualizzazione:

<https://docs.google.com/spreadsheets/d/1NmfdGhdrYmAq704tWWFO8NbRhKP2rGj2gb8J7e71MZI/edit?usp=sharing>

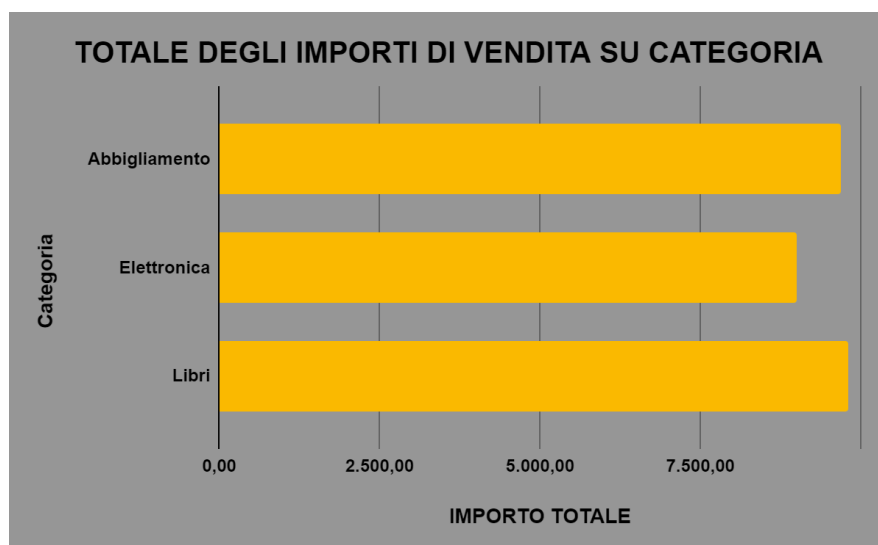
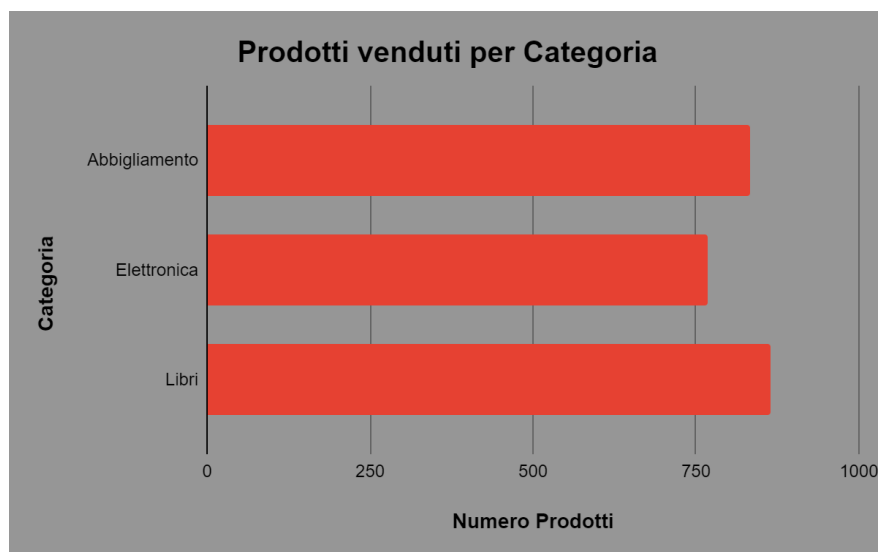
1. Analisi Categoria Prodotto in Excel: (1 punto)

- **Domanda:** Importa i dati delle transazioni e dei prodotti in Excel. Crea una tabella pivot per visualizzare le vendite per categoria di prodotto e un grafico a barre per rappresentare i risultati.

Risultato:

Categoria	Prodotti Venduti	TOTALE Importi €
Abbigliamento	834	9.697,08
Elettronica	768	8.992,02
Libri	865	9.814,11

Totale generale	2467	28.503,21
------------------------	-------------	------------------



Spiegazione:

Abbiamo introdotto nel foglio transazioni una nuova colonna, grazie alla funzione CERCA.VERT/CERCA.X: =CERCA.X(C2;prodotti!A:A;prodotti!C:C;;0). Come primo campo si inserisce il valore da confrontare (come se fosse il codice su cui facciamo la JOIN in sql), nel nostro caso l'ID del prodotto nella tabella transazioni, confrontandolo con la colonna A del foglio contenente la tabella prodotti e facendoci restituire le categorie, presenti nella colonna C. Nella quinta condizione specifichiamo che vogliamo corrispondenza ESATTA, 0. Infine facciamo la Tabella Pivot, Inserendo come Riga la categoria e come campo valore: SUM di quantità acquistata e SUM di importo transazione. Altrimenti si poteva usare lo strumento Power Pivot importando tutti i dataset in un modello dati per poi creare tabelle pivot partendo da tutte le colonne presenti all'interno di tale modello.

2. Spedizioni Riuscite in Excel: (1 punto)

- **Domanda:** Importa i dati delle spedizioni in Excel. Utilizza la funzione CONT.SE per contare il numero di spedizioni con "Consegna Riuscita" e "In Consegna". Crea un grafico a torta per visualizzare la percentuale di spedizioni riuscite.

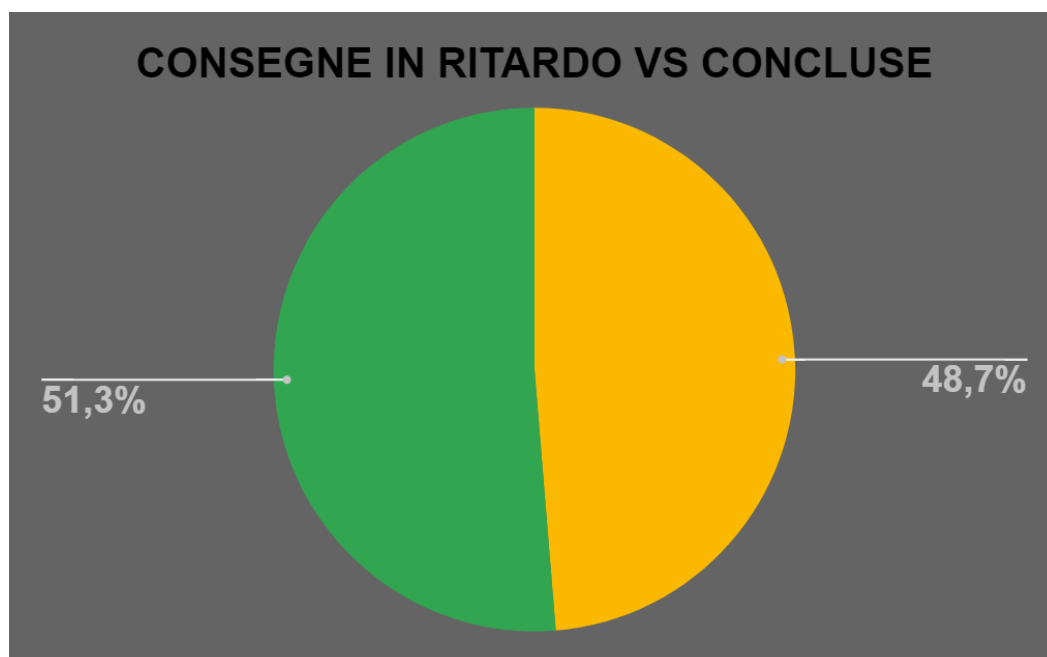
Formule utilizzate:

=CONTA.SE(D:D;"Consegna Riuscita")

=CONTA.SE(D:D;"In Consegna")

Risultato:

IN RITARDO	2435
CONCLUDE	2565



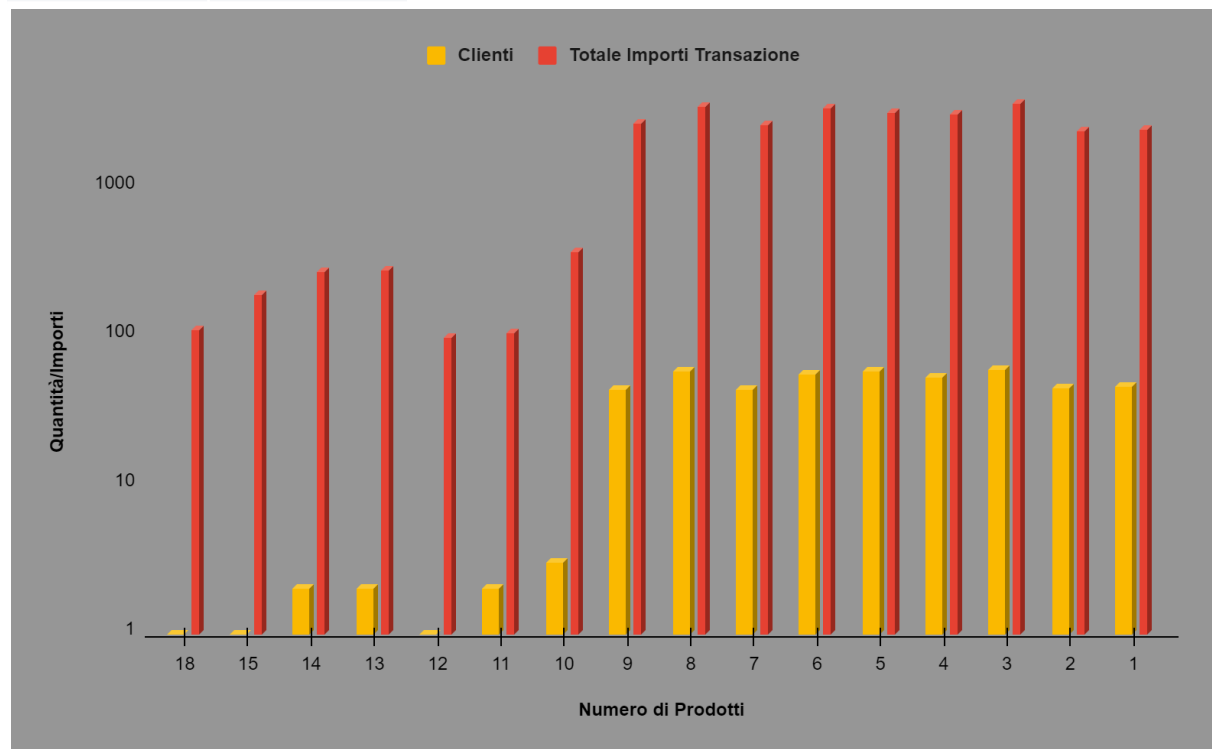
Spiegazione: Abbiamo fatto i precedenti due CONTA SE, e una volta impostato il grafico a torta, abbiamo messo l'opzione di visualizzazione delle percentuali.

3. Analisi dei Prodotti per Cliente in Excel: (2 punto)

- **Domanda:** Importa i dati delle transazioni in Excel. Crea una tabella pivot per visualizzare i prodotti acquistati e il totale speso per ogni cliente. Utilizza un grafico a barre raggruppato per rappresentare i risultati.

Risultato:

<i>ClienteID</i>	<i>ProdottoID</i>	Prodotti Acquistati	Spesa della Transazione
3875	151	9	35,35
	619	9	76,18
Totale 3875		18	111,53
2105	1777	6	98,36
	316	9	93,14
Totale 2105		15	191,50
3350	5067	5	87,52
	4225	9	72,19
Totale 3350		14	159,71
1215	2651	6	46,46
	4226	8	67,30
Totale 1215		14	113,76
4213	3227	4	75,27
	1784	9	61,01
Totale 4213		13	136,28
3231	3545	5	58,13
	5012	8	81,13
Totale 3231		13	139,26
4114	2358	5	45,42
	4893	7	53,13



N.B. La visualizzazione dell'asse delle ordinate è in scala logaritmica (dato che le due colonne hanno un ordine di grandezza di differenza)

Spiegazione:

Nella tabella pivot, abbiamo raccolto i dati degli acquisti effettuati da ogni cliente: cosa ha acquistato, in quale quantità e quanto ha speso.

Provando a realizzare un grafico che mostrasse gli acquisti, raggruppando per ID Cliente, il risultato era di difficile lettura. Di conseguenza abbiamo optato per raggruppare per tipologia d'acquisto (basandosi sulla quantità totale di prodotti acquistati per cliente). Le barre gialle rappresentano quanti clienti hanno effettuato acquisti con una determinata quantità di prodotti e la barra rossa, rappresenta l'utile generato per ogni categoria d'acquisto.

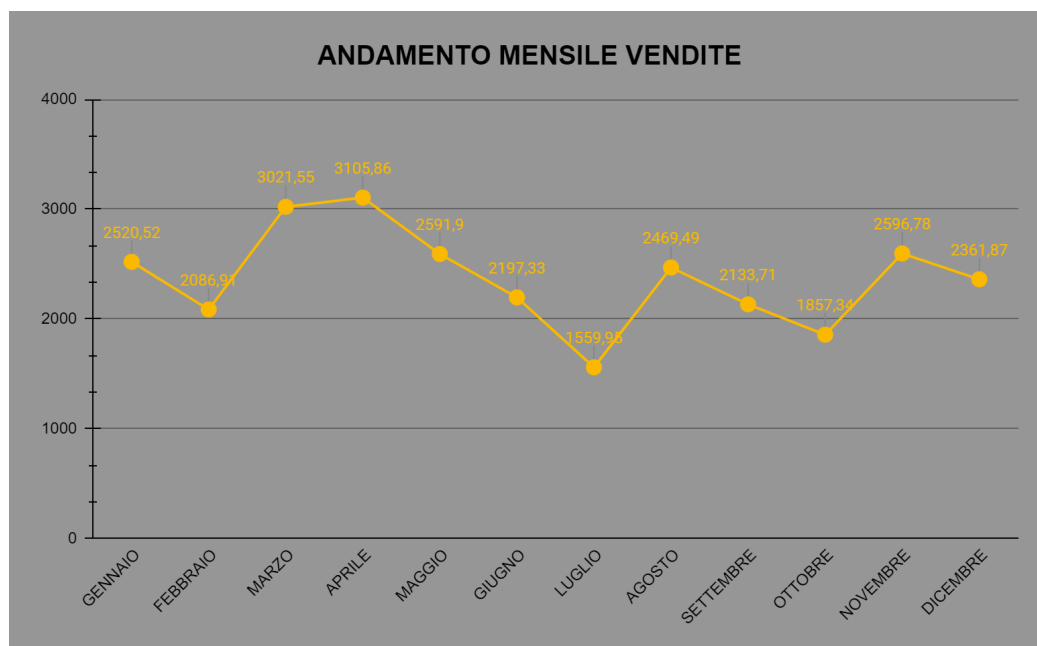
4. Analisi Temporale delle Vendite in Excel: (1 punto)

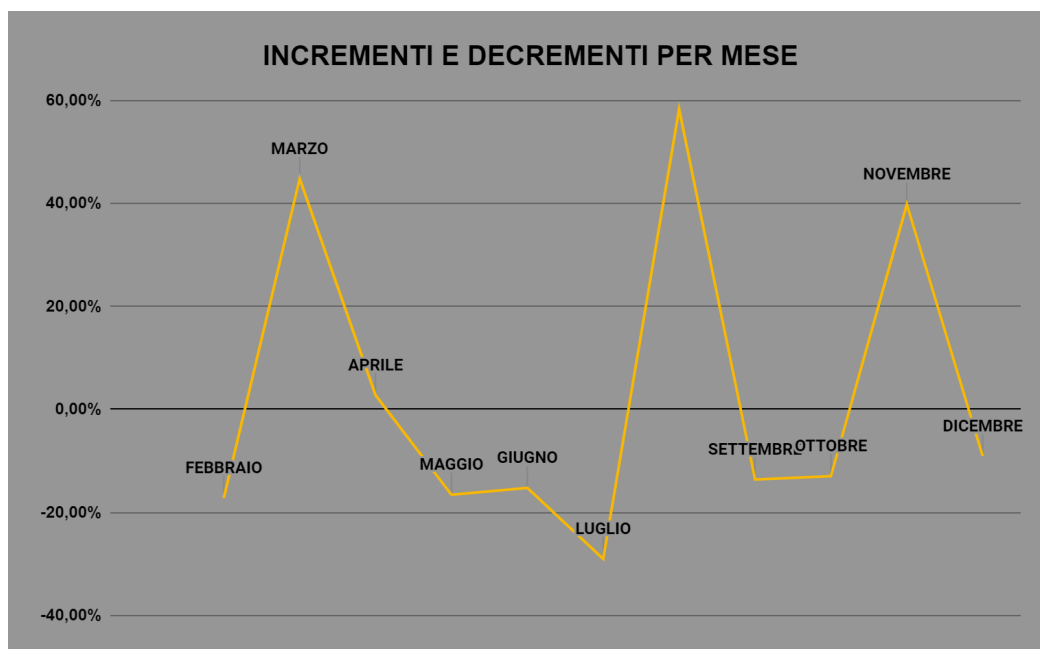
- **Domanda:** Importa i dati delle transazioni in Excel. Crea un grafico a linee che mostri la variazione mensile nelle vendite nel corso del tempo.

Formule utilizzate:

=SOMMA.SE(transazioni!K:K;A2;transazioni!F:F)
=(C3/C2)-1

Risultato:





Spiegazione: Per l'analisi temporale delle vendite mese per mese abbiamo creato un primo grafico che mostra la distribuzione del monte vendite per ciascun mese ed un secondo grafico che mostra la distribuzione degli incrementi o decrementi tra il mese precedente ed il mese corrente. Nel primo caso ci siamo avvalsi della funzione Excel SOMMA.SE per sommare tutti gli elementi importo(vendite) della colonna K secondo il criterio mobile sulla colonna, ovvero i mesi; potevamo comunque avvalerci sempre di Power Pivot altrimenti; Per generare il secondo grafico invece abbiamo operato un FRAZIONE mese corrente su precedente sottratto moltiplicato 100, sottratto 100 e trasformato in formato percentuale.

5. Analisi Distribuzione dei Prodotti in Excel: (2 punto)

- **Domanda:** Importa i dati delle transazioni e dei prodotti in Excel. Utilizza una tabella pivot per analizzare la distribuzione delle vendite per categoria e sottocategoria di prodotto.

Risultato:

Categoria	ProdottoID	SUM di ImportoTransazione
Totale Abbigliamento		9.697,08
Totale Elettronica		8.992,02
Totale Libri		9.814,11
Totale generale		28.503,21

Spiegazione:

Questa tabella pivot, risponde anche qui a una query simile precedentemente vista in SQL. Sulla tabella prodotti creiamo una pivot, dove mettiamo come prima riga la categoria e come seconda i prodotti appartenenti a tale categoria. Infine, come campo valore introduciamo la somma degli importi totali per categoria. Nell'ultima riga, tramite l'opzione "mostra totali" ci viene anche mostrato il totale delle vendite, come somma dei totali per categoria.

6. Analisi degli Acquisti dei Clienti nel Tempo in Excel: (2 punto)

- **Domanda:** Importa i dati delle transazioni in Excel. Utilizza una tabella pivot per analizzare gli acquisti dei clienti nel tempo, suddivisi per mese e anno.

Risultato:

ANNO	MESE	ClienteID	ProdottoID	Spesa	
2022	1	76	3778	42,15	
		Totale 76			42,15
		511	3442	63,34	
		Totale 511			63,34
		599	2348	30,11	
		Totale 599			30,11
		720	3182	76,11	
		Totale 720			76,11
		739	3226	53,08	
		Totale 739			53,08
		826	1290	27,06	
		Totale 826			27,06
		1044	1668	19,27	
		Totale 1044			19,27
		1171	3529	83,12	
		Totale 1171			83,12

Riducendo le righe

ANNO	MESE	ClienteID	ProdottoID	Spesa
2022	Totale 1			2.520,52
	Totale 2			2.086,91
	Totale 3			3.021,55
	Totale 4			3.105,86
	Totale 5			2.591,90
	Totale 6			2.197,33
	Totale 7			1.559,95

	Totale 8	2.469,49
	Totale 9	2.133,71
	Totale 10	1.857,34
	Totale 11	2.596,78
	Totale 12	2.361,87
Totale 2022		28.503,21
Totale generale		28.503,21

Spiegazione:

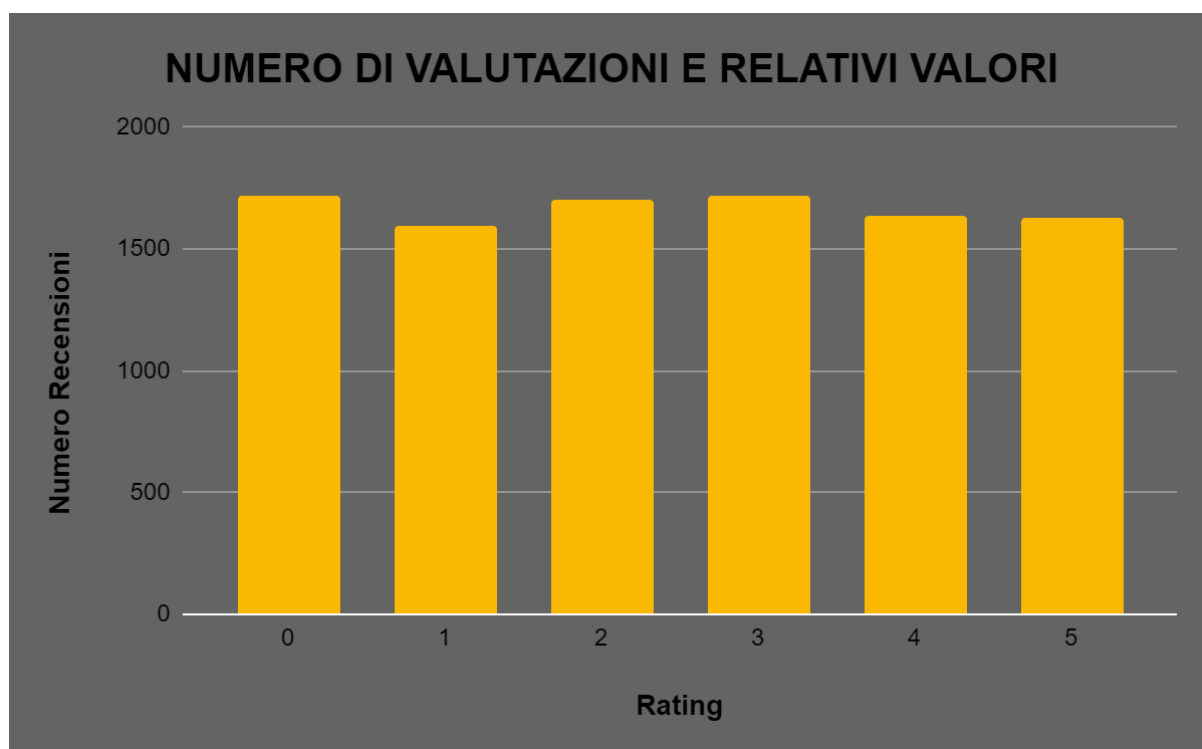
Qui vediamo per ogni Anno/Mese, quali clienti hanno effettuato acquisti, che prodotti hanno acquistato e l'ammontare degli importi. Si poteva volendo anche operare sulla somma della quantità acquistata dal cliente.

7. Analisi di Feedback Clienti in Excel:(1 punto)

- **Domanda:** Importa i dati delle recensioni dei prodotti in Excel. Crea un grafico a barre che mostri la distribuzione delle recensioni dei clienti.

Risultato:

Rating	Numero Recensioni	
	0	1717
	1	1595
	2	1703
	3	1720
	4	1636
	5	1629
Totale generale		10000



Spiegazione:

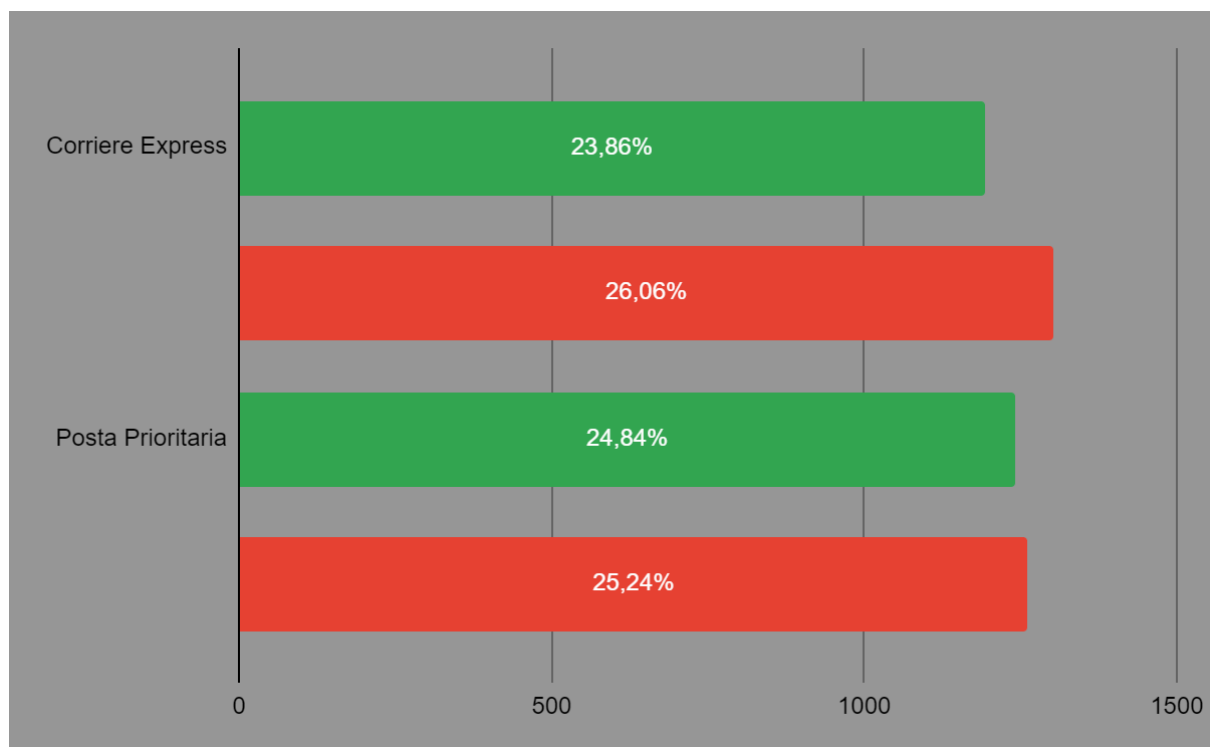
Ecco dunque operando di nuovo attraverso una pivot rating/ numero di conteggi per rating, dalla tabella recensioni, abbiamo infine allegato la tabella e suo grafico.

8. Analisi Incidenza di Spedizioni Riuscite per Metodo in Excel: (2 punti)

- **Domanda:** Importa i dati delle spedizioni in Excel. Utilizza una tabella pivot per analizzare la percentuale di spedizioni riuscite per ciascun metodo di spedizione.

Risultato:

MetodoSpedizi one	StatusConsegn a	Spedizioni	percentuale
Corriere Express	Consegna Riuscita	1193	23,86%
	In Consegna	1303	26,06%
Posta Prioritaria	Consegna Riuscita	1242	24,84%
	In Consegna	1262	25,24%
Totale generale		5000	100,00%



Spiegazione:

Creiamo la pivot dalla tabella spedizioni, inserendo come riga principale il metodo di spedizione e come sottoriga lo status della consegna. Dopodiché metto come campo valore un conta, che mi va a vedere quante spedizioni appartengono a ciascun sottotipo. Infine un campo percentuale sempre delle spedizioni appartenenti ai sottotipi, sul totale.

Così da poter poi creare il grafico precedente sulla riuscita/ritardo delle rispettive tipologie di consegna.